Importance of Self-Attention for Sentiment Analysis

Gaël Letarte[*], **Frédérik Paradis**[*], **Philippe Giguère**, **François Laviolette**
Department of Computer Science and Software Engineering
Université Laval, Québec, Canada
{gael.letarte, frederik.paradis}.1@ulaval.ca

Abstract

Despite their superior performance, deep learning models often lack interpretability. In this paper, we explore the modeling of insightful relations between words, in order to understand and enhance predictions. To this effect, we propose the Self-Attention Network (SANet), a flexible and interpretable architecture for text classification. Experiments indicate that gains obtained by self-attention is task-dependent. For instance, experiments on sentiment analysis tasks showed an improvement of around 2% when using self-attention compared to a baseline without attention, while topic classification showed no gain. Interpretability brought forward by our architecture highlighted the importance of neighboring word interactions to extract sentiment.

1 Introduction

Deep neural networks have achieved great successes on numerous tasks. However, they are often seen as black boxes, lacking interpretability. Research efforts in order to solve this issue have steadily increased (Simonyan et al., 2013; Zeiler and Fergus, 2014; Bach et al., 2015; Ribeiro et al., 2016; Fong and Vedaldi, 2017). In language modeling, interpretability often takes place via an attention mechanism in the neural network (Bahdanau et al., 2014; Xu et al., 2015; Sukhbaatar et al., 2015; Choi et al., 2017). In this context, attention essentially allows a network to identify which words in a sentence are more relevant. Beyond interpretability, this often results in improved decision making by the network.

Recently, Vaswani et al. (2017) proposed the *Transformer* architecture for machine translation. It relies only on attention mechanisms, instead of making use of either recurrent or convolutional neural networks. This architecture contains layers called self-attention (or intra-attention) which allow each word in the sequence to pay attention to other words in the sequence, independently of their positions. We modified this architecture, resulting in the following contributions:

- A novel architecture for *text classification* called Self-Attention Network (SANet) that models the interactions between all input word pairs. It is sequence length-agnostic, thanks to a global max pooling layer.
- A study on the impact of this self-attention mechanism on large scale datasets. In particular, we empirically demonstrate the positive impact of self-attention in terms of performance and interpretability for sentiment analysis, compared to topic classification. In the study, we make use of two quantitative metrics (Gini coefficient and diagonality) that exhibit particular behaviors for attention mechanisms in sentiment analysis.

2 Related Work

The majority of text classification techniques either use convolutional or recurrent neural networks on the words or the characters of the sentence (Zhang et al., 2015, 2017; Yang et al., 2016; Conneau et al., 2017; Johnson and Zhang, 2016, 2017; Howard and Ruder, 2018). One notable exception is the *fastText* architecture (Joulin et al., 2016) which essentially employs a bag-of-words approach with word embeddings of the sentence.

Attention mechanisms are a way to add interpretability in neural networks. They were introduced by Bahdanau et al. (2014), where they achieved state-of-the-art in machine translation. Since then, attention mechanisms have been used in other language modeling tasks such as image captioning (Xu et al., 2015), question answer-

[*] Authors contributed equally to this work.

Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pages 267–275
Brussels, Belgium, November 1, 2018.

ing (Sukhbaatar et al., 2015; Choi et al., 2017), and text classification (Yang et al., 2016). The concept of *self-attention* (Cheng et al., 2016; Parikh et al., 2016), central to our proposed approach, has shown great promises in natural language processing; It produced state-of-the-art results for machine translation (Vaswani et al., 2017).

In text classification, the focus on interpretability has thus far been limited. Lee et al. (2018) used a convolutional neural network (CNN) with Class Activation Mapping (CAM) (Oquab et al., 2015) to do sentiment analysis. CAM basically uses the weights of the classification layer to derive a heatmap on the input. Wang et al. (2018) used a densely connected CNN (Huang et al., 2017) to apply attention to n-grams. However, their approach limits the range and acuteness of the interactions between the words in the text. Lin et al. (2017) and Yang et al. (2016) both combined an attention mechanism with a recurrent neural network. The main difference with our work is, while being interpretable, these approaches do not perform true word-on-word attention across a whole sequence such as our self-attention layer.

3 SANet: Self-Attention Network

Inspired by the *Transformer* architecture (Vaswani et al., 2017) which performed machine translation without recurrent or convolutional layers, we propose the Self-Attention Network (SANet) architecture targeting instead text classification. One key difference between our approach and Vaswani et al. (2017)'s is that we only perform *input-input* attention with self-attention, as we do not have sequences as output but a text classification. Moreover, we employ global max pooling at the top, which enables our architecture to process input sequences of arbitrary length.

Formally, let $X = [x_1^T; x_2^T; \ldots; x_n^T]$ be the concatenation of a sequence of n vectors giving a matrix $X \in \mathbb{R}^{n \times d}$ such that $x_i \in \mathbb{R}^d$. Vaswani et al. (2017) defined attention as a function with as input a triplet containing queries Q, keys K with associated values V.

$$\text{Att}(Q, K, V) = \text{softmax}\left(QK^T\right)V$$

In the case of self-attention, Q, K and V are linear projections of X. Thus, we define the dot-product self-attention mechanism as follows.

$$\begin{aligned}\text{Self-Att}(X) &= \text{Att}(XW_Q, XW_K, XW_V) \\ &= \text{softmax}\left(XW_{QK}X^T\right)XW_V\end{aligned}$$

Where $W_Q, W_K, W_V, W_{QK} \in \mathbb{R}^{d \times d}$ and $W_{QK} = W_Q W_K^T$. Hence, W_{QK} and W_V are learned parameters.

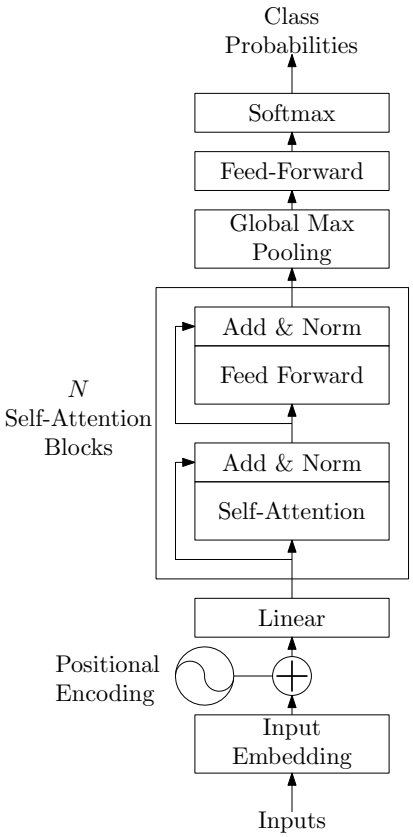


Figure 1: Our Self-Attention Network (SANet), derived from the Transformer architecture (Vaswani et al., 2017). The self-attention block is repeated N times.

Our network (depicted in Figure 1) first encodes each word to its embedding. Pre-trained embeddings, like GloVe (Pennington et al., 2014), may be used and fine-tuned during the learning process. Next, to inject information about the order of the words, the positional encoding layer adds location information to each word. We use the positional encoding vectors that were defined by Vaswani et al. (2017) as follows.

$$\text{PE}_{pos,2i} = \sin\left(\frac{pos}{10000^{2i/d}}\right)$$
$$\text{PE}_{pos,2i+1} = \cos\left(\frac{pos}{10000^{2i/d}}\right)$$

Where pos is the position of the word in the sequence and $1 \leq i \leq d$ is the dimension in the positional encoding vector.

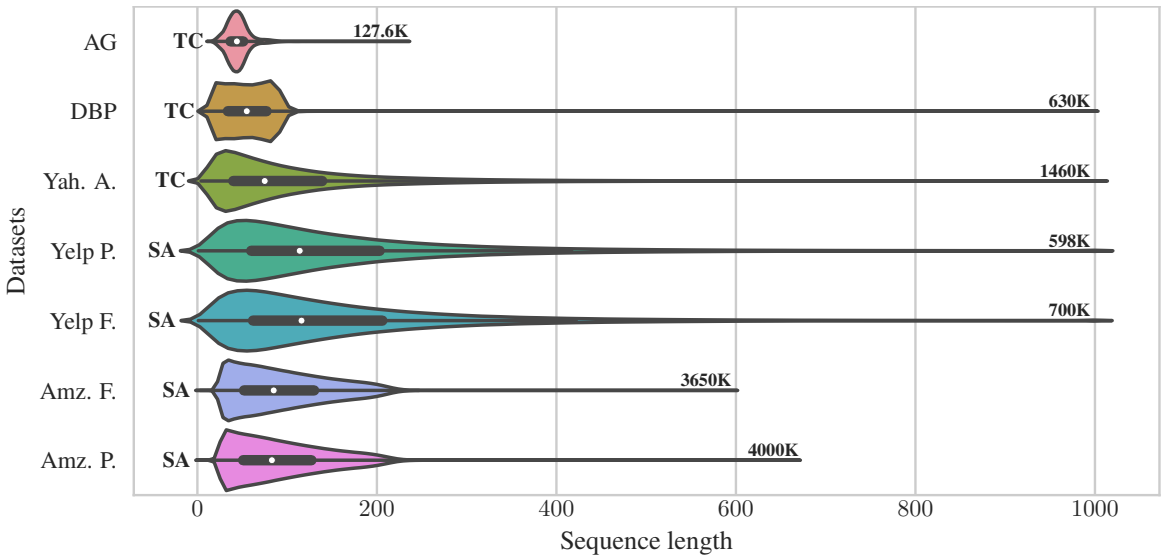


Figure 2: Visualization of sequences length distributions. For each dataset, the total number of examples is presented on the right and task semantics are identified on the left: Topic Classification (TC) or Sentiment Analysis (SA).

A linear layer then performs dimensionality reduction/augmentation of the embedding space to a vector space of dimension d, which is kept constant throughout the network. It is followed by one or several "self-attention blocks" stacked one onto another. These blocks are comprised of a self-attention layer followed by a feed-forward network, both with residual connections. Contrary to Vaswani et al. (2017), we only use a single attention head, with attention performed on the complete sequence with constant d-dimensional inputs.

The feed-forward network consists of a single hidden layer with a ReLU.

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

Where $W_1, W_2 \in \mathbb{R}^{d \times d}$ are learned parameters. The "Add & Norm" layer is a residual connection defined by $\text{LayerNorm}(x + \text{SubLayer}(x))$, where $\text{SubLayer}(x)$ is the output of the previous layer and LayerNorm is a layer normalization method introduced by Ba et al. (2016). Let x_i be the vector representation of an element in the input sequence. The normalization layer simply normalizes x_i by the mean and the variance of its elements. Throughout this paper, dropout of 0.1 is applied to the output of $\text{SubLayer}(x)$

Finally, since we restrict ourselves to classification, we need a fixed-size representation of the sequence before the classification layer. To achieve this, we apply a global max pooling operation for each dimension across all the n words of the sequence. That is, if $X \in \mathbb{R}^{n \times d}$, then the max pooling on X outputs a vector in $\mathbb{R}^d$. This technique was inspired by global average pooling introduced by Lin et al. (2013) for image classification in CNNs. Global max pooling allows us to handle sequences of any length (up to memory limitations). Thus, our approach is length-agnostic contrary to some approaches based on CNN, where sequences are truncated or padded to obtain a fixed-length representation.

4 Experiments

We evaluated our model on seven large scale text classification datasets introduced by Zhang et al. (2015), grouped into two kinds of tasks. The first one is topic classification: *AG's News* with 4 classes of news articles, *DBPedia* with 14 classes of the Wikipedia ontology and *Yahoo! Answers* containing 10 categories of questions/answers. *Yelp* and *Amazon* reviews involve sentiment analysis with ratings from 1 to 5 stars. Two versions are derived from those datasets: one for predicting the number of stars, and the other involving the polarity of the reviews (negative for 1-2 stars, positive for 4-5 stars).

Each text entry was split into sentences and tokenized using NLTK (Bird et al., 2009). Sequences longer than 1000 tokens were truncated to accommodate GPU memory limitations, only affecting a negligible portion of the texts. See Figure 2 for

Table 1: Test error rates (%) for text classification. In **bold**, the state-of-the-art and in *italic*, our best model. Lin et al. (2017)'s results provided by Wang et al. (2018). Stars (*) indicate attention mechanisms.

Model	Topic Classification			Sentiment Analysis			
	AG	DBP.	Yah. A.	Yelp P.	Yelp F.	Amz. F.	Amz. P.
ngrams/CNN (Zhang et al., 2015)	7.64	1.31	28.26	4.36	37.95	40.43	4.98
fastText (Joulin et al., 2016)	7.5	1.4	27.7	4.3	36.1	39.8	5.4
word-CNN (Johnson and Zhang, 2016)	6.57	0.84	24.85	2.90	32.39	36.24	3.79
HN-ATT* (Yang et al., 2016)	-	-	**24.2**	-	-	36.4	-
VDCNN (Conneau et al., 2017)	8.67	1.29	26.57	4.28	35.28	37.00	4.28
DCNN (Zhang et al., 2017)	-	1.17	25.82	3.96	-	-	-
DPCNN (Johnson and Zhang, 2017)	6.87	0.88	23.90	2.64	30.58	**34.81**	**3.32**
SA-Embedding* (Lin et al., 2017)	8.5	1.7	-	5.1	36.6	40.2	-
ULMFiT (Howard and Ruder, 2018)	**5.01**	**0.80**	-	**2.16**	**29.98**	-	-
DCCNN-ATT* (Wang et al., 2018)	6.4	**0.8**	-	3.5	34.0	37.0	-
Baseline (base model)	7.34	1.30	26.87	6.39	39.98	41.80	6.38
SANet* (base model)	7.86	1.27	26.99	6.26	38.16	40.08	5.55
Baseline (big)	*7.20*	*1.25*	25.90	6.42	38.92	40.58	5.82
SANet* (big)	7.42	1.28	*25.88*	*4.77*	*36.03*	*38.67*	*4.52*

a visualization of the resulting sequences length distribution and the total number of examples per dataset.

We used 20% of the training texts for validation. The vocabulary was built using every word appearing in the training and validation sets. The words embeddings were initialized using pre-trained word vectors from GloVe (Pennington et al., 2014) when available, or randomly initialized otherwise.

We experimented with two configurations for our proposed SANet. The base model used $N = 1$ self-attention blocks, an embedding size of 100 and a hidden size of $d = 128$. The big model doubled these numbers, i.e. $N = 2$ self-attention blocks, embedding size of 200 and hidden size $d = 256$. For each configuration, we also trained a baseline network without any attention mechanisms, replacing each self-attention layer with a feed forward layer.

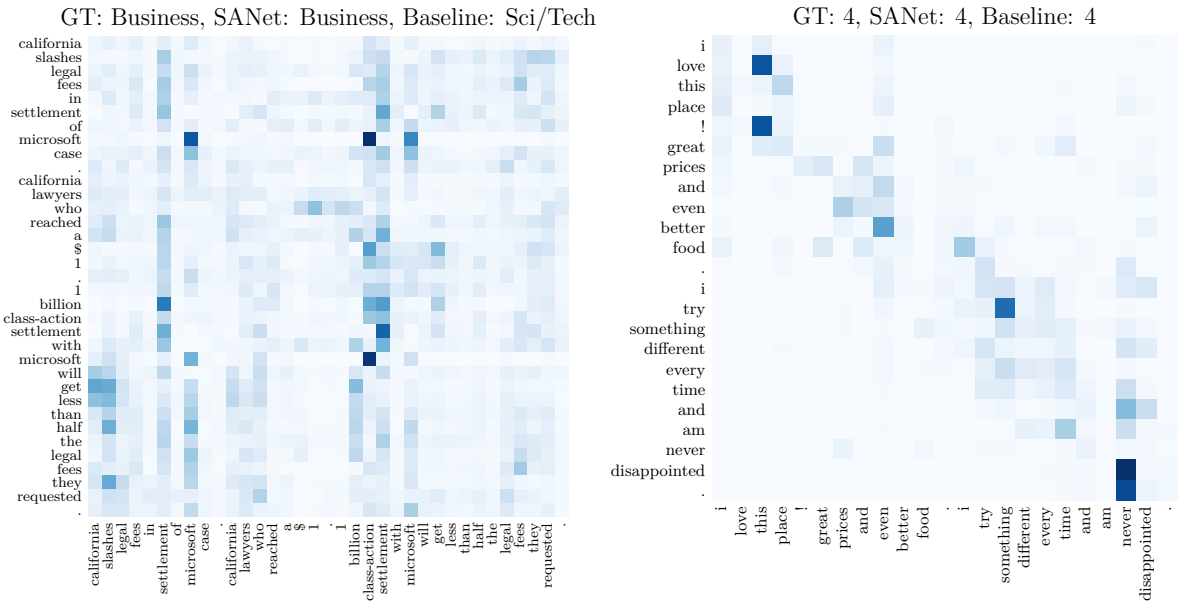
Training was performed using SGD with a momentum of 0.9, a learning rate of 0.01 and mini-batches of size 128. For the embeddings, a learning rate of 0.001 was applied without momentum. All learning rates were halved for the big model. We trained for 40 epochs and selected the best epoch, based on validation accuracy.

5 Results and Discussion

From a performance perspective, as shown in Table 1, our model based entirely on attention is competitive while offering high level interpretability. There is a notable exception with *Yelp Review Polarity* that will be discussed. Our results also indicate that the increase in depth and representation size in the big model is beneficial, compared to the simpler base model. Most noteworthy, we noticed considerably different behaviors of the attention mechanism depending on the type of task. We offer an analysis below.

5.1 Topic Classification Tasks

On the topic classification task, the self-attention behavior can be described as looking for interactions between important concepts, without considering relative distance. As such, it acts similarly to a bag-of-word approach, while highlighting key elements and their associations. Thus, the attention matrix takes shape of active columns, one per concept. One such matrix is depicted in Figure 3a, where the attention is focused on distanced pairs such as (`microsoft`, `class-action`) or (`settlement`, `billions`) to help SANet predict the *Business* category, while the baseline wrongfully predicts *Sci/Tech*. We observed this column-based structure for attention matrix for every topic classification dataset, see Figure 4 for



(a) Topic Classification.

(b) Sentiment Analysis. Label 4 means 5-star review

Figure 3: Self-attention different behavior for each text classification task. The attention matrices were extracted from the SANet base model applied on the testing set of each dataset. Words on the y-axis are attending to the words on the x-axis. GT refers to the ground truth.

multiple examples. Although it adds interpretability to the model, our results seem to indicate that self-attention does not improve performances for topic classification, compared to the baseline.

5.2 Sentiment Analysis Tasks

For sentiment analysis tasks, self-attention improves accuracy for every dataset and model configurations that we tested. For *Yelp Review Polarity*, although attention helps, the overall performances remain subpar.

Noticeably for the other datasets, SANet is able to extract subtle interactions between words, with a strong focus on neighboring relation. Hence, the attention matrices are close to being band matrices, with interest concentrated on very small regions near the diagonal. This is observable in Figure 5 where multiple examples from all sentiment analysis datasets are presented. Concentration of the attention around the diagonal indicates that the useful features learned by the attention mechanism consist essentially of skip-bigrams with relatively small gaps. Of note, Wang and Manning (2012) previously observed consistent gains when including word bigram features to extract sentiment. Thus, our model corroborates this intuition about sentiment analysis while yielding interpretable insights on relevant word pairs across all possible skip-bigrams.

Figure 3b is a typical example of such matrix with a band diagonal structure, for a 5-star Yelp review. A number of positive elements are highlighted by the self-attention mechanism such as *i)* the initial strong sentiment with the interaction between `this` with `love` and `!` *ii)* the favorable comparison with `even` and `better` *iii)* the enticing openness to experiences with `try` and `something` and *iv)* the positive combination of two negative words with `never` and `disappointed`.

Positional encoding helps the self-attention mechanism when interpreting words repetitions, in order to extract sentiment gradation. When repeating three times an adjective before the modified noun, attention on the adjective increases with their proximity to the noun: **horrible horrible horrible service**. Punctuation repetitions exhibit a similar behavior, as in the sentence "love this place!!!", where the words `love` and all three exclamation points apply attention to `this` with varying intensities: **love this** `place` **! ! !**. This particular behavior of the model reinforces our belief that it learns intricate knowledge for the task of sentiment analysis. Entire attention heatmaps for complete sequences can be found in Figure 6.

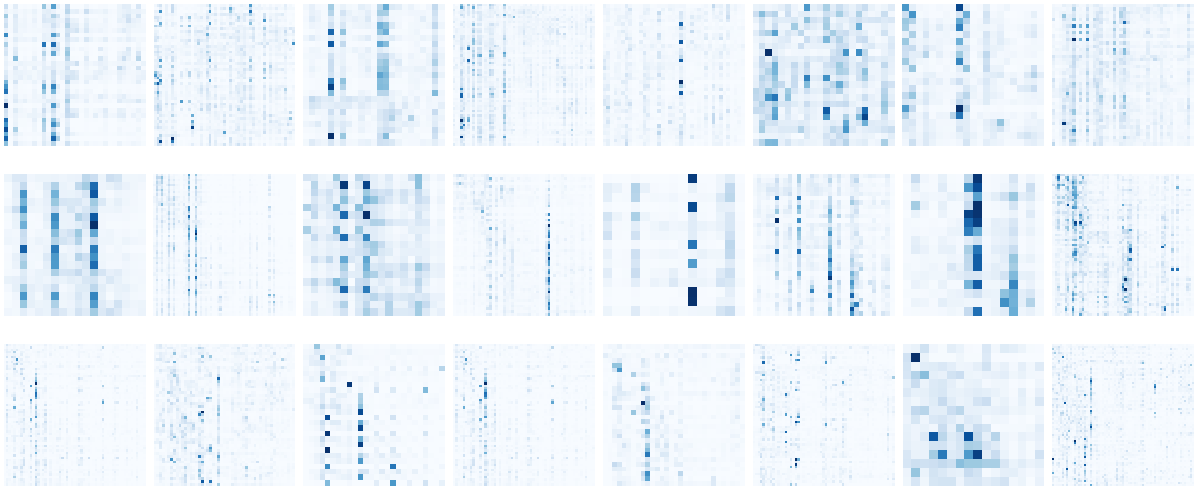


Figure 4: Randomly selected attention matrices for topic classification task. Each row corresponds to a different dataset in this order: *AG's News*, *DBPedia* and *Yahoo! Answers*. The column-based pattern is clearly present in the attention mechanism for topic classification.

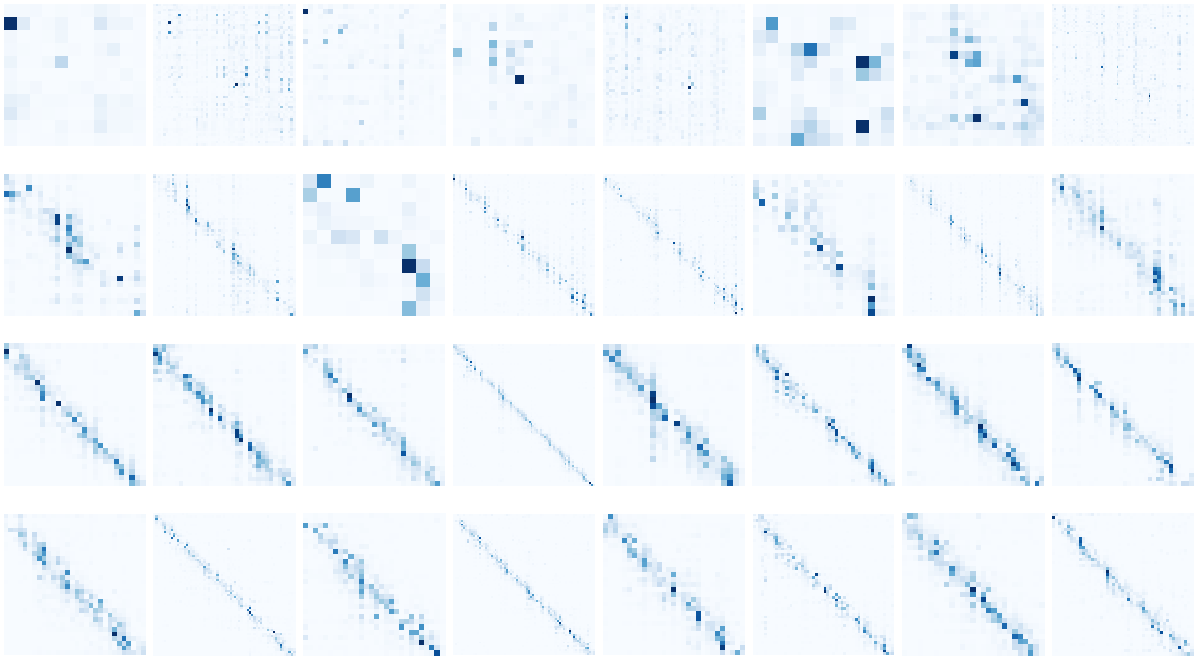
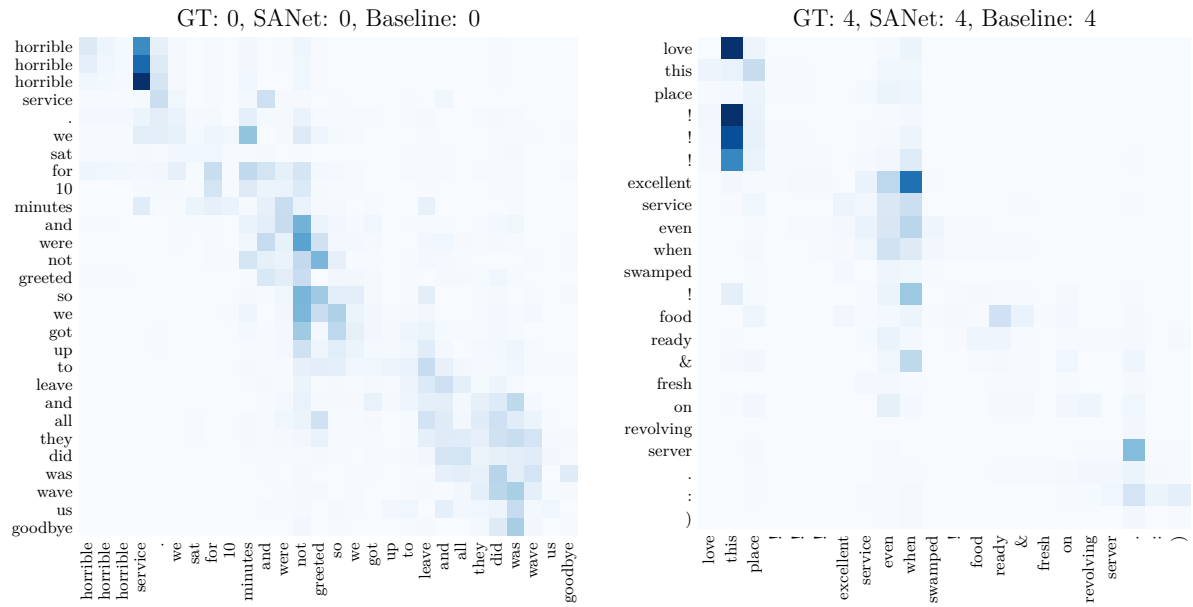


Figure 5: Randomly selected attention matrices for sentiment analysis task. Each row corresponds to a different dataset in this order: *Yelp Review Polarity*, *Yelp Review Full*, *Amazon Review Full* and *Amazon Review Polarity*. The diagonal band pattern of the matrices is clearly present in the attention mechanism for sentiment analysis except for the *Yelp Review Polarity* dataset.

5.3 Quantitative Analysis

We now present a quantitative analysis of the attention matrices to support the qualitative intuition stated previously. Two metrics are used in order to assess the properties of the matrices; the first one (Gini coefficient) quantifies the sparsity of the attention, whereas the second one (diagonality) focuses on the diagonal concentration. These two properties are relevant for interpretability issues. The results are presented in Table 2.

The Gini coefficient which measures the inequality in the attention weights distribution is first computed. For topic classification datasets, the mean of the Gini coefficient is 63.57%, whereas, for sentiment analysis datasets, it raises at 87.15% without considering *Yelp Review Polarity*. Thus, for topic classification it reveals that every word



(a) Adjective repetition.

(b) Punctuation emphasis.

Figure 6: Positional encoding impact for sentiment gradation through self-attention mechanism. Both examples are extracted from the testing set of the *Yelp Review Full* dataset. Label 4 means 5-star review and label 0 means 1-star review. Words on the y-axis are attending to the words on the x-axis. GT refers to the ground truth.

Table 2: Quantitative statistics of the self-attention mechanism behavior for the two text classification tasks. Metrics are computed on the testing sets using the SANet base model.

Metric	Topic Classification			Sentiment Analysis			
	AG	DBP.	Yah. A.	Yelp P.	Yelp F.	Amz. F.	Amz. P.
Gini coefficient	55.31	67.94	67.45	65.16	84.18	89.50	87.76
Diagonality (bandwidth = 1)	7.44	8.49	6.34	5.02	23.54	41.77	40.01
Diagonality (bandwidth = 2)	11.86	13.80	9.83	7.89	36.89	62.35	60.34
Diagonality (bandwidth = 3)	16.21	18.88	13.28	10.62	45.49	73.53	71.43
Diagonality (bandwidth = 4)	20.42	23.74	16.59	13.19	50.90	79.49	77.21
Diagonality (bandwidth = 5)	24.48	28.25	19.65	15.62	54.54	83.09	80.56

interacts with multiple other words in the sequence. On the other hand, for sentiment analysis, the attention is focused on a fewer number of word pairs. The second metric will also point out that the sentiment analysis attention is sparse and specifically based on pair of words that are close in the sentence. This structurally corresponds to an attention matrix concentrated near the diagonal and justifies the introduction of the following metric.

This new metric evaluates the resemblance with a band matrix by computing the proportion of attention weights which occur inside the band diagonal of a given bandwidth b, thus the band diagonality or diagonality for short. It expresses the interactions of every element with itself, and the b elements before and after in the sequence. This metric of diagonality was computed for a bandwidth of $b = 1, 2, \ldots, 5$ as presented in Table 2. Results clearly reveal that sentiment analysis attention matrices are structurally close to being band matrices. Notably, with a bandwidth of $b = 3$ for topic classification, 16.12% of the weights occur inside the band diagonal, as for sentiment analysis without considering *Yelp Review Polarity*, 63.48% is located inside the band diagonal.

In our opinion, the combination of these two metrics supports our qualitative observations of

the attention matrices. It strengthens the difference in attention behavior between the topic classification and sentiment analysis task. Moreover, this quantitative analysis clearly exposes SANet inability to learn the appropriate attention behavior for sentiment analysis with *Yelp Review Polarity*. Its failure to adequately exploit the self-attention mechanism coincide with its poor performance to extract sentiment. Interestingly, *Yelp Review Polarity* examples are a subset of *Yelp Review Full* with merged classes, for which SANet performs well with the expected attention behavior. The cause of this discrepancy with the Yelp datasets is unknown and left for future work as is some linguistic investigation of the impact of close interacting words in sentiment analysis.

6 Conclusion

In this paper, we introduced the Self-Attention Network (SANet), an attention-based length-agnostic model architecture for text classification. Our experiments showed that self-attention is important for sentiment analysis. Moreover, the improved interpretability of the model through attention visualization enabled us to discover considerably different behaviors of our attention mechanism between the topic classification and sentiment analysis tasks. The interpretable perspective of this work gives insights on the importance of modeling interaction between neighboring words in order to accurately extract sentiment, as noted by (Wang and Manning, 2012) for bigrams. It highlights how interpretability can help us understand models behavior to guide future research. In the future, we hope to apply our Self-Attention Network to other datasets such as bullying detection on social network data and tasks from various fields, such as genomic data in bioinformatics. Finally, we wish to study the properties of the introduced global max pooling layer as a complementary tool for interpretability in a similar way that was done with CAM (Oquab et al., 2015) for global average pooling. The outcome will be some attention on individual words that can take into account the context given by the self-attention mechanism. This contrast with the approach of this paper which focuses on interaction between elements as pairs. Thus we are allowed to expect that these two mechanisms will act in a complementary way to enrich interpretability.

Acknowledgments

We would like to thank Nicolas Garneau, Alexandre Drouin and Jean-Samuel Leboeuf for their advice and insightful comments. This work is supported in part by NSERC. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

This work was done using the PyTorch library (Paszke et al., 2017) with the PyToune framework (Paradis and Garneau).

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*.

Eunsol Choi, Daniel Hewlett, Jakob Uszkoreit, Illia Polosukhin, Alexandre Lacoste, and Jonathan Berant. 2017. Coarse-to-fine question answering for long documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 209–220.

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2017. Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 1107–1116.

Ruth C Fong and Andrea Vedaldi. 2017. Interpretable explanations of black boxes by meaningful perturbation. *arXiv preprint arXiv:1704.03296*.

Jeremy Howard and Sebastian Ruder. 2018. Fine-tuned language models for text classification. *arXiv preprint arXiv:1801.06146*.

Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, volume 1, page 3.

Rie Johnson and Tong Zhang. 2016. Convolutional neural networks for text categorization: Shallow word-level vs. deep character-level. *arXiv preprint arXiv:1609.00718*.

Rie Johnson and Tong Zhang. 2017. Deep pyramid convolutional neural networks for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 562–570.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Gichang Lee, Jaeyun Jeong, Seungwan Seo, CzangYeob Kim, and Pilsung Kang. 2018. Sentiment classification with word localization based on weakly supervised learning with a convolutional neural network. *Knowledge-Based Systems*.

Min Lin, Qiang Chen, and Shuicheng Yan. 2013. Network in network. *arXiv preprint arXiv:1312.4400*.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.

Maxime Oquab, Léon Bottou, Ivan Laptev, and Josef Sivic. 2015. Is object localization for free?-weakly-supervised learning with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 685–694.

Frédérik Paradis and Nicolas Garneau. PyToune. http://pytoune.org.

Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Empirical Methods in Natural Language Processing*, pages 2249–2255.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

Shiyao Wang, Minlie Huang, and Zhidong Deng. 2018. Densely connected cnn with multi-scale feature attention for text classification. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*.

Sida Wang and Christopher D Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 90–94. Association for Computational Linguistics.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057.

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.

Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

Yizhe Zhang, Dinghan Shen, Guoyin Wang, Zhe Gan, Ricardo Henao, and Lawrence Carin. 2017. Deconvolutional paragraph representation learning. In *Advances in Neural Information Processing Systems*, pages 4172–4182.