Multiway Attention with Densely-connected Recurrent and Co-attentive Networks for Sentence Matching

Hubert Au, Jay Haran, and Joel Sandler

Department of Computer Science, University College London

Abstract

Densely-connected recurrent and co-attentive networks (DRCN) have achieved state-of-the-art performances in natural language inference, as well as other tasks. However, the DRCN only applies one attentive function of cosine similarity that has no learnable weights. Our contribution is an enhancement of the DRCN using multiway attention, inspired by previous work that has found multiway attention networks to be successful in the same tasks. We achieved an increase in performance from our baseline model by 5.4%, from 77.46% to 82.86% on the Stanford Natural Language Inference dataset. In particular, we found the best combination to be bilinear, concatenation, cosine, and minus attention, aggregated using a BiLSTM. We believe there is ample scope for future work: there is strong evidence that less time restriction would return much higher performance, and our contribution is highly modular, allowing easily for future research into more varied combinations of attention functions.

1 Introduction

The main research question studied was whether densely-connected recurrent and co-attentive neural networks (DRCN) can be combined with multiple attention mechanisms to increase accuracy for semantic sentence matching, specifically for the task of natural language inference. The attention mechanisms being used are bilinear attention, concatenation attention, cosine attention, element-wise dot product and minus attention. Further, it has been shown that annotation artefacts in our dataset of choice - the Stanford Natural Language Inference corpus (Bowman et al., 2014) - allows for approximately 67% accuracy to achieved on the test set solely using the hypothesis (Gururangan et al., 2018). Given that the nature of our experiments concerns word-level attention matching, we also qualitatively examined whether our models make decisions on key words and phrases identified in Gururangan et al. (2018).

Semantic sentence matching is a core research area in natural language processing. Its applications include natural language inference, i.e., determining if a given statement (the premise) semantically entails another given statement (the hypothesis) and question answering, which is used for a variety of tasks such as information retrieval and chat bots. Ongoing developments in deep learning along with large new datasets such as the Stanford Natural Language Inference SNLI corpus (570k sentence pairs) (Bowman et al., 2014) and the Multi-Genre Natural Language Inference MultiNLI corpus (433k sentence pairs) (Williams et al., 2018) facilitate advancements in semantic sentence matching learning.

There are two traditional approaches for learning semantics for sentence matching. The first is sentence-encoding-based where each sentence is encoded to a vector of fixed size and both vectors for their corresponding sentences are used to predict the degree of matching (Conneau et al., 2017; Nie and Bansal, 2017; Shen et al., 2018). The limitation of this first approach is the lack of interaction between the sentences during the encoding. The second approach utilises interactive features such as attentive information between the sentences. It applies the attention mechanism at the word level, which improves the matching between words in the two sentences. This matching information is then aggregated to the sentence level (Wang et al., 2017; Gong et al., 2017; Chen et al., 2016).

Our experiment built upon Kim et al. (2018), which used a DRCN (defined formally in section 3 below) and utilised the increased representational power of deeper recurrent networks with attentive

information. We added four additional attention mechanisms to the framework: bilinear attention, concatenation attention, element wise dot product, and minus attention (similarly, to be defined in section 3).

Our hypothesis was the following: applying a multiway attention mechanism in conjunction with DCRN will lead to an improvement in performance compared to the base DRCN with a singular cosine attention mechanism. In particular, we hypothesised that the full combination of the five proposed attention functions will perform the best. 'Best' was measured as the classification accuracy on the test set. To test our hypothesis, we first replicated the base DRCN with its cosine attention, then modified it to contain the full combination of the five functions proposed. Finally, we performed an ablation study to obtain some measure of how each attention function contributes to performance. This totalled to seven models trained, for which we found that within our time and computational constraints our hypothesis was partially true. Although there was improvement on the baseline DRCN with the full model, the best test accuracy was achieved with the model combining bilinear, concatenation, cosine, and minus attention, with the best validation multi-class cross entropy loss across all epochs of all models at 0.479, achieving a test set classification accuracy of 82.86%.

The structure of this paper is as follows. In section 2, we review related work and hence provide motivation for our research. In section 3, we explain the key features of our baseline model and state our original contribution. In section 4, we give implementation details and explain the details of the experiments run. In section 5, we display and critically analyse our results, and give the aforementioned qualitative analysis. Finally, in section 6, we give suggestions for future work.

2 Related Work

Attention mechanisms have been used to capture interaction between sentence pairs to deliver better performance in natural language inference (Vaswani et al., 2017). Kim et al. (2018) and their DRCN belong to the second approach mentioned above, known as the matching-aggregation framework as seen in Tan et al. (2016) and Wang et al. (2017). Previously, these frameworks have been used to improve LSTM-based recurrent neural networks, employing word-by-word attention as seen in Rocktäschel et al. (2015). Crucially, the DRCN incorporates substantially deeper recurrent networks than before. These are a preferred architecture for longer sequences and in general, have outperformed shallower ones. Kim et al. (2018) serves as the focus for the rest of this section.

Their architecture utilises five layers of densely-connected recurrent cells and uses an autoencoder to suppress the number of dimensions, which they observe to also play a role of regularisation The attention mechanism that is employed is a cosine similarity between word vectors in P (henceforth 'P' is used to refer to the premise in an example) and word vectors in Q (henceforth 'Q' is used to refer to the hypothesis in an example). This is displayed in Figure 1.

The key shortcoming we observed with Kim et al. (2018) is that they did not comment on or try different attention mechanisms, or an aggregation of the outputs of multiple attention mechanisms, which has been successful in the past (Tan et al., 2018; Man et al., 2017) for the same task. This served as the basis for our project; to experiment with multiple attention mechanisms.

We draw inspiration for the attention functions we employ from Tan et al. (2018). They emphasised the importance of word-level interactions between sentences and produced architectures that perform well using multiple attention mechanisms. They then made decisions by aggregating the information from each attention mechanism using a complex structure of multiple GRUs and further attention.

The crucial part of the multiway attention network (MwAN) in Tan et al. (2018) is that, in their respective ablation studies, all of their attention mechanisms contributed to an improvement to their final score. Therefore, we conjectured that the expansion of the DRCN architecture with respect to attention functions should result in improved performance on the SNLI dataset.

We further drew inspiration from Man et al. (2017) for the BiLSTM as a mechanism for attention aggregation. In their multiway attention experiments, they also found each attention function to contribute to an improvement to their final performance. We did not use their attention functions due to time limitations.

It is from these papers that we determined our research hypothesis. Given previous findings, we

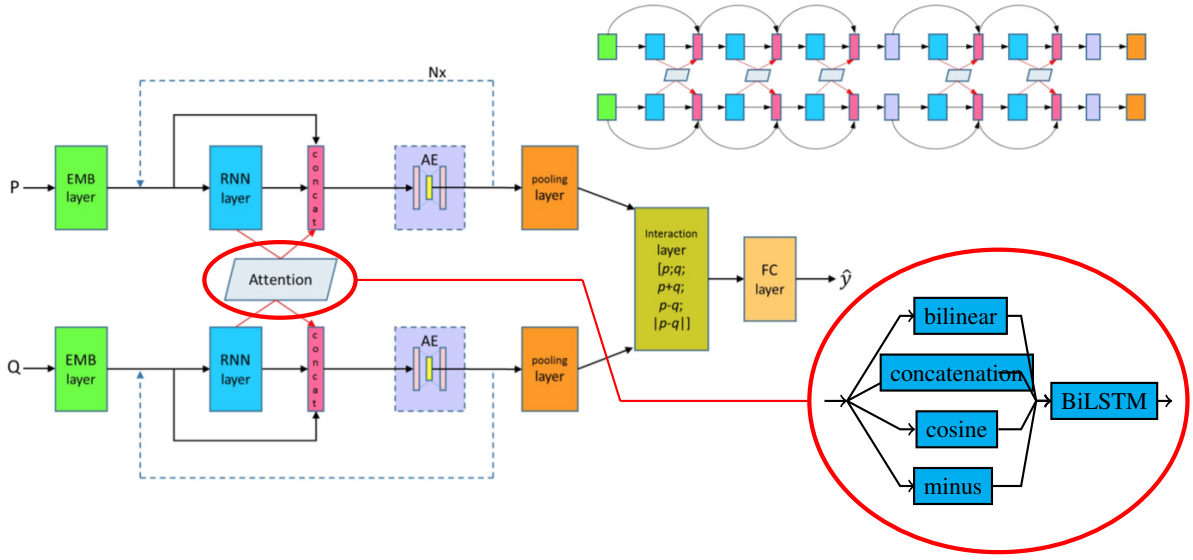

Figure 1: Architecture from Kim et al. (2018), with an added illustration of our contribution. In their words,'General architecture of our Densely-connected Recurrent and Co-attentive Neural Network (DRCN). Dashed arrows indicate that a group of RNN-layer, concatenation and AE can be replaced multiple (N) times (like a repeat mark in a music score). The bottleneck component denoted as AE, inserted to prevent the ever-growing size of a feature vector, is optional for each repetition. The upper right diagram is our specific architecture for experiments with 5 RNN layers (N=4)'.

expect that each attention function we propose to use will improve best test score performance when aggregated.

3 Methods

In this section we describe the DRCN architecture of the model our research was based on, and detail how our sentence matching architecture differs. In the first subsection, we do not give a full replication of the DRCN model, but instead highlight a few of its key features. In the second subsection we detail our contribution.

3.1 DRCN

3.1.1 Word Representation Layer

The DRCN utilises a word representation of four components, combined through concatenation. The first two components are initialised to pre-trained word embeddings; namely GloVe embeddings. One copy is fixed in the embedding layers, and one copy is allowed to be mutable during training time. The motivation for this is that trainable word embeddings have been found to capture particular characteristics of a task well but can be prone to overfitting. Conversely, fixed word embeddings lack this adaptability. Therefore, combining them in this way is a simple but effective method, as demonstrated in the original paper. The third component is a character-level representation of each word, as introduced by Zhang et al. (2015). The final component is an exact match flag that is activated whenever the exact word appears in the other sentence (e.g. in the sentences 'Hello World' and 'Bye World', the 'World' word in both sentences would have this flag activated in their representations).

3.1.2 RNN Stack and Attention

Kim et al. (2018) utilise an RNN stack, inspired by Densenet (Huang et al., 2016). The RNN stack works as follows: the base layer is a BiLSTM that takes in the output of the word representation layer. The following layers are all RNNs and each take the output of the previous layer, concatenated with the input to the previous layer, as input. This is to encourage proper gradient flow backwards. For a time-step t, a regular ordinal RNN is specified as:

$$\begin{aligned} h_t^l &= H_l(x_t^l, h_{t-1}^l) \\ x_t^l &= h_t^{l-1} \end{aligned} \tag{1}$$

where h_t^l is the output of layer l at time-step t. In a Densenet structure, it is instead specified as,

using ';' as the concatenation operation:

$$h_t^l = H_l(x_t^l, h_{t-1}^l)$$
$$x_t^l = [x_t^{l-1}; h_t^{l-1}] \quad (2)$$

That is, for a layer with input dimension 200 and an output dimension of 100, the input dimension to the next layer is $200 + 100 = 300$.

Further, the DRCN incorporates co-attention. That is, for the vector representation of each word (time-step) in a layer in sentence P, $h_{p_i}^l$ and conversely $h_{q_j}^l$, attention is calculated both ways and the results are concatenated with the output of each RNN layer, forming the input into the next layer. The output of attention of $h_{p_i}^l$ on $h_{q_j}^l$ is fed back into the P stream, and the output of attention of $h_{q_j}^l$ on $h_{p_i}^l$ is fed back into the Q stream.

The full expression for the Densenet-inspired RNN stack therefore is:

$$h_t^l = H_l(x_t^l, h_{t-1}^l)$$
$$x_t^l = [x_t^{l-1}; h_t^{l-1}; a_t^{l-1}] \quad (3)$$

where a_t^{l-1} is the output of the attention function. In the base DRCN, the following cosine attention function is used, where (dropping layer notation) h_{p_i} is the representation of the *i*-th word of sentence P, similarly for Q, and the output is a_{p_i}:

$$e_{i,j} = cos(h_{p_i}, h_{q_j})$$
$$\alpha_{i,j} = \frac{exp(e_{i,j})}{\sum_{k=1}^{J} exp(e_{i,k})} \quad (4)$$
$$a_{p_i} = \sum_{j=1}^{J} \alpha_{i,j} h_{q_j}$$

3.1.3 Max-Pooling, Interaction and Prediction Layers

Finally, the outputs from the RNN stack is max-pooled over the dimension of (max) sentence length. That is, for a 50-word length sentence representation output of 50×600, the output of the max-pooling operation gives a 600 length output.

These are then combined using

$$v = [p; q; p + q; p - q; |p - q|] \quad (5)$$

and passed into two fully connected layers into a three-class log-softmax output.

3.2 Contribution

Our contribution is two-fold: multiway attention and a modified exact match flag.

3.2.1 Multiway Attention

We employ a multiway attention in place of the cosine attention used in the base DRCN. As discussed above our implementation is inspired by MwAN and CALYPSO. Given the same inputs as the base DRCN, our model replaces the cosine attention with the following:

$$x = [a_{bilinear}; a_{concat}; a_{cosine}; a_{dot}; a_{minus}]$$
$$a_{combined} = BiLSTM(x) \quad (6)$$

where initial hidden and cell states are set to zero. The output $a_{combined}$ has the same dimension as a_p as in the base DRCN. The four additional functions are taken from MwAN. Notably, however, we observe a mistake in their mathematical representation of the concat attention. Where Huang et al. (2016) specify:

$$e_{i,j} = v_{Ca}^T tanh(W_{Ca}^1 h_{p_i} + W_{Ca}^2 h_{q_j}) \quad (7)$$

the operation within the nonlinearity is manifestly not one of concatenation. The correct implementation, and the one we use, is

$$e_{i,j} = v_{Ca}^T tanh(W_{Ca}[h_{p_i}; h_{q_i}]) \quad (8)$$

The full specification for attention functions used is as follows:

Bilinear (also known as 'weighted soft attention'):

$$e_{i,j} = v_B^T tanh(h_{q_j}^T W_B h_{p_i})$$
$$\alpha_{i,j} = \frac{exp(e_{i,j})}{\sum_{k=1}^{J} exp(e_{i,k})} \quad (9)$$
$$a_{p_i} = \sum_{j=1}^{J} \alpha_{i,j} h_{q_j}$$

Concatenation:

$$e_{i,j} = v_{Ca}^T tanh(W_{Ca}[h_{p_i}; h_{q_i}])$$
$$\alpha_{i,j} = \frac{exp(e_{i,j})}{\sum_{k=1}^{J} exp(e_{i,k})} \quad (10)$$
$$a_{p_i} = \sum_{j=1}^{J} \alpha_{i,j} h_{q_j}$$

Dot:

$$
\begin{aligned}
e_{i,j} &= v_D^T tanh(W_D(h_{p_i} \odot h_{q_j})) \\
\alpha_{i,j} &= \frac{exp(e_{i,j})}{\sum_{k=1}^{J} exp(e_{i,k})} \\
a_{p_i} &= \sum_{j=1}^{J} \alpha_{i,j} h_{q_j}
\end{aligned}
\tag{11}
$$

Minus:

$$
\begin{aligned}
e_{i,j} &= v_M^T tanh(W_M[h_{p_i} - h_{q_j}]) \\
\alpha_{i,j} &= \frac{exp(e_{i,j})}{\sum_{k=1}^{J} exp(e_{i,k})} \\
a_{p_i} &= \sum_{j=1}^{J} \alpha_{i,j} h_{q_j}
\end{aligned}
\tag{12}
$$

The BiLSTM aggregation is taken from CALYPSO. We decided not to employ the more complex multi-way GRU aggregation that MwAN uses to keep the independent variable of the actual attention functions used as isolated as possible.

3.2.2 Exact Match Flag

The base DRCN activates the exact match flag for a word when the identical token appears in the other sentence. However, we employ a match flag as used in the work they cite (Gong et al., 2017), and as was shown to be useful in previous work (Chen et al., 2017). This is a binary flag that is activated when either the stem or lemma of a word is present in the comparison sentence and improves performance. We believe this should pass into the network an indicator that is much more robust to minor changes in words that nevertheless play an important part in logical reasoning (consider the stem 'play' in the example: 'a dog plays with a bone' and 'a dog is playing with an object').

4 Experiments

4.1 Implementation Details

Within the SNLI dataset, only examples that had a golden label were included, meaning examples without a clear majority annotation were not included. A batch size of 64 was used; the batch size of 350 that Kim et al. use was not possible due to memory limitations. Each of the models were run to 6 epochs each due to computation time limitations; the original authors ran to 30 epochs. We used the SpaCy tokeniser, and NLTK with WordNet for stems and lemmas. We use the RMSProp optimiser with a learning rate of 0.001, reduced by a factor of 0.15 (i.e. 85% drop) when dev. loss (not accuracy) stops increasing, using an optimiser learning scheduler. The models were trained using multi-class cross entropy loss, implemented as a Log Softmax with Negative Log Likelihood loss. 300d GloVe vectors trained on Common Crawl were used to initialise word embeddings. Out-of-vocab words were initialised randomly, using a standard normal distribution with the random seed set to 1. Hidden dimension of the RNN stack was 100. ReLU activations were used after the convolutional layer and after each fully connected layer. Character embeddings were similarly randomly initialised to 16d embeddings, and 32d representations were extracted through the character convolution. Character embeddings and character convolution weights were jointly learned during training time. All parameters in the models were subject to L2 weight of 1e-6. Dropout was applied to each RNN layer output (before passing to attention function) with a dropout probability of 0.5. Dropout was applied after the interaction and pooling layer with a dropout probability of 0.2. Batch normalisation was applied to both fully connected layers.

4.2 Experiments

Initial runs indicated that batch sizes of 128 and 96 caused memory errors on a K80 GPU and it was found that 64 was a stable batch size to use. We collected results for the base DRCN to serve as a baseline. We ran the full 5-attention combination, and an ablation study, dropping each of the 5 attention functions respectively.

5 Results and Discussion

5.1 Experiment and Ablation Study Results

Table 1 shows the results of our experiment with the full model (all five attention mechanisms) and the results of the ablation study. It also shows the performance of the model only using a cosine attention mechanism, which was our baseline. The best performing model on the test set was *not* the full model as hypothesised, but instead the model containing all but the dot attention, with a test accuracy of 82.86%, achieved on the 6th epoch (out of 6 in total), compared to the baseline model best accuracy of 77.46%. The full model achieved a best test accuracy of 81.26% on the 5th epoch. This best performing model (B-Ca-Cs-M) also had

the best validation loss of 0.457. The worst performing model was the one without the bilinear attention (Ca-Cs-D-M) with a best test accuracy of 74.56% which was achieved on the 5th epoch, resulting in a worse accuracy than the baseline model.

Figure 2 shows the bias-variance graph for the best performing model. Both training loss and validation loss decrease monotonically as expected, with validation loss approaching training loss as epoch number increases.

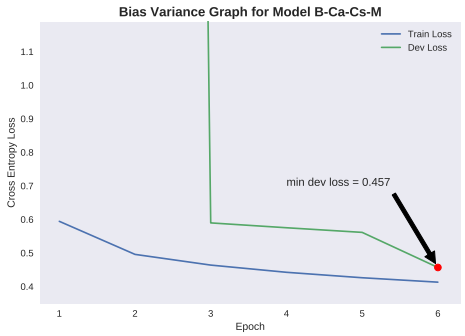

Figure 2: Bias-variance graph for model (B-Ca-Cs-M). It stands to reason more epochs will deliver even better performance.

Figure 3 shows, for an example of a contradiction premise/hypothesis pair, weights for the attention functions used in the best performing model. The example had the label 'contradiction' and was correctly predicted by the model. Further examples are available in the Appendix and will be analysed in the following section.

5.2 Discussion

The results indicate that our original hypothesis was partially correct: the full model B-Ca-Cs-D-M did improve over our baseline by $81.26 - 77.46 = 3.8\%$. However, the best improvement came not from the full model but from model 3 (B-Ca-Cs-M) in the ablation study that drops the dot attention. It achieved a test accuracy improvement over the baseline of $82.86 - 77.46 = 5.4\%$, and also achieved the lowest validation loss of any model at 0.457.

We make a few observations here. First, the DRCN model (Cs) we use as a baseline achieved 88.9% on the SNLI test set in the original paper, which was not achieved by our replication. Our attempt was made to the best of our ability and in liaison with Seonhoon Kim. The two most significant differences in our implementations were batch size (original authors used a batch size of 350) and training time (original authors trained for 30 epochs, which was not possible here timewise). We expect that a larger batch size may help training with more accurate gradients. Moreover, upon examination of the bias-variance graph (see Figure 9) for the base DRCN in our experiments, we expect that further training will give further decreases in training and validation loss, with an associated increase in test score performance.

Second, we surmise that not only would further epochs allow the base model a better estimate of the original author's achievement, it would also greatly benefit our models with added attention functions. This is because the cosine attention used in the original paper has no learnable parameters, and the introduction of these new attention functions add a significant number of learnable parameters. Given that there is already a significant increase in performance over 6 epochs with the introduction of new attention mechanisms, it stands to reason that more epochs could plausibly accelerate this improvement (see Figure 2).

Third, consider Figure 3. This is an example of a correctly predicted premise/hypothesis pair in the validation set; the graph shows attention weights for each attention function at layers 1, 3, and 5 respectively. The reverse co-attention of the hypothesis on the premise is available in the appendix as Figure 4, along with other examples of entailment and neutral pairs (Figure 5, Figure 6, Figure 7, Figure 8). Immediately upon examination of this example and others we find that the first layer (leftmost graphically) typically seems to carry the most intuitively accessible information: in the 1st layer bilinear weights, 'stands' is heavily picked up by most words in the premise, and 'tall' is picked up by the minus attention function. In this example's co-attention weights (Figure 6), 'stands' picks up 'crouches' and 'tall' picks up 'low' in the 1st layer bilinear weights. Typically, as information moves beyond the first layer, most of the time two of the four attention functions pick up the word(s) necessary for the logical relationship recognition very strongly. In particular, it seems all input words pick up on them very strongly. In this example we see this most identifiably in the 3rd layer weights for the concatenation attention (middle graph of the second row). This begs the question: why do the attention weights lose intuitive understanding as we move

Model No.	Model	Best epoch	Train loss	Dev. loss	Best test accuracy (%)
1	B-Ca-Cs-D-M	5	0.413	0.479	81.26
2	B-Ca-Cs-D (no M)	6	0.399	0.514	81.38
3	B-Ca-Cs-M (no D)	6	0.413	**0.457**	**82.86**
4	B-Ca-D-M (no Cs)	2	0.488	0.491	80.64
5	B-Cs-D-M (no Ca)	4	0.436	0.504	80.01
6	Ca-Cs-D-M (no B)	5	0.411	0.627	74.56
7	Cs	6	0.370	0.574	77.46

Table 1: Experiment results. Models are delineated as follows. 'B' indicates bilinear attention, 'Ca' indicates concatenation attention, 'Cs' indicates cosine attention, 'D' indicates dot attention, and 'M' indicates minus attention. Hyphenation indicates the inclusion of the attentions listed, and from models 2-6 the brackets indicate the attention function that was dropped from the full model (model 1).

through the network? Is this a problem? To the first question: we propose that the main reason the weights look less intuitive after the 1st layer is that just as the base layer of the RNN stack is a BiLSTM and not an RNN, our architecture has the first layer's attention functions storing different learned weights than layers 2-5, which share the same weights. This shared nature may mean that during backpropagation each layer's gradients compete in the attention functions, causing a jumbled behaviour. Conversely, however, as stated we do observe a trend that key words become more uniformly picked up later in the network. To the second question then, this may not be a problem and may be the result of more and more aggregated contextual information as the tensors move through the RNN stack, pointing to a small subset of key words. We believe this provides ripe ground for future work: attempt separated weights for each layer's attention functions.

Fourth, we were aware of potential annotation artefacts from the SNLI dataset, i.e., patterns arising from when crowd workers authored the hypotheses. It has been shown that a simple classification model can correctly label the hypothesis alone in 67% of the SNLI dataset (Gururangan et al., 2018). In particular, they find that entailment hypotheses often contained generic words such as 'outdoors', 'instrument' and 'animal', which may have been chosen to generalize over more specific words in the premise such as 'park', 'trombone' or 'cat'. These hypotheses also often contain the words 'at least' or 'some', which replace exact numbers with approximates. Furthermore, the words 'human' and 'person' are very common, which removes the gender from the premise. Neutral hypotheses tend to contain modifiers for example 'tall' or 'sad', or superlatives such as 'favourite' or 'most'. They also often contain purpose clauses such as 'because' or 'in order to'. Contradiction hypotheses often include negation, i.e., 'not', 'never' and 'nothing'. Also, the word 'sleeping' is common as it contradicts any activity and so too the word 'naked' as this contradicts the mention of any clothing. We found that our results are not inconsistent with their analysis. Upon examination of the attention weights of a random sample of approximately $1\% = 100$ examples within the test set, we find that (a) for contradiction, 'sleep' is quite heavily picked up by at least one function towards later layers, (b) adjectives such as 'good' and 'bad', along with modifiers that Gururangan et al. (2018) note such as 'tall' and 'sad' do appear to be picked up heavily by at least one function, (c) though for entailment we observed less use of particular general words and more repetition of uni, bi, or tri-grams that were the indicator of their logical relationship - this was not always obviously picked up by the attention functions.

6 Conclusion and Future Work

We believe that the results presented support our hypothesis that multiway attention improves DRCN performance on the natural language inference task. Given limited time and resources we were not able to replicate the results of the DRCN, but instead achieved 77.46% accuracy on the test set. From this baseline we found improved performance with multiway attention using the four attention functions of bilinear, concat, cosine, and

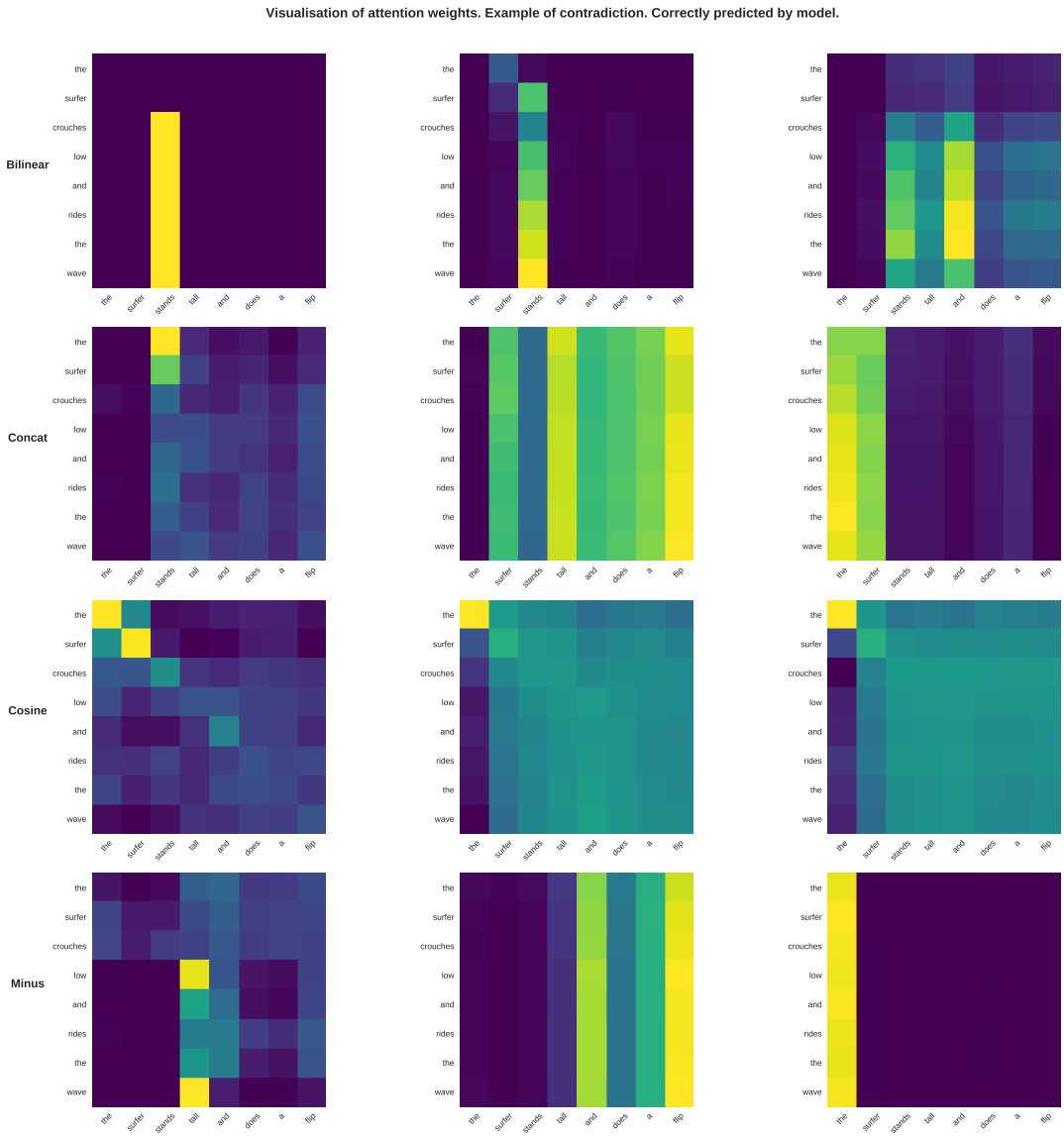

Figure 3: Visualisation of attention matrices for an example of contradiction. Premise is 'The surfer crouches low and rides the wave' and the hypothesis is 'The surfer stands tall and does a flip'. Each row represents the attention weights of one function, and from left to right they are the outputs of the 1st, 3rd, and 5th layers of the DRCN stack respectively.

minus, achieving a best accuracy of 82.86% and less validation loss than in the base model. We believe that more time will only improve our model further as the introduction of multiple attention functions with learnable parameters, as well as the aggregation BiLSTM, should require more time to properly train. Some suggestions for future work are as follows. First, more epochs and a larger batch size. Second, as we have noted above in our analysis of attention weight visualisations, we believe that a reasonable modification to our proposed architecture is to have individually learned attention weights for each function for each of the DRCN layers. Third, more attention functions may be employed. For example, self-attention has been shown to be effective in natural language processing tasks (Vaswani et al., 2017).

We will be continuing this research out of personal interest.

7 Acknowledgements

We greatly thank Sebastian Riedel, Tim Rocktäschel, and Qiang Zhang. We are also indebted to Seonhoon Kim for his kind assistance.

References

Samuel R Bowman, Gabor Angeli, Christopher Potts, Christopher D Manning, and Stanford Linguistics. 2014. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to Answer Open-Domain Questions.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2016. Enhanced LSTM for Natural Language Inference.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data.

Yichen Gong, Heng Luo, and Jian Zhang. 2017. Natural Language Inference over Interaction Space.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R Bowman, and Noah A Smith. 2018. Annotation Artifacts in Natural Language Inference Data.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. 2016. Densely Connected Convolutional Networks.

Seonhoon Kim, Inho Kang, and Nojun Kwak. 2018. Semantic Sentence Matching with Densely-connected Recurrent and Co-attentive Information.

Colin Man, Kenny Xu, and Kat Gregory. 2017. CALYPSO: A Neural Network Model for Natural Language Inference. Technical report.

Yixin Nie and Mohit Bansal. 2017. Shortcut-Stacked Sentence Encoders for Multi-Domain Inference.

Tim Rocktäschel, Edward Grefenstette, and Karl Moritz Hermann. 2015. Reasoning about Entailment with Neural Attention. Technical report.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Sen Wang, and Chengqi Zhang. 2018. Reinforced Self-Attention Network: a Hybrid of Hard and Soft Attention for Sequence Modeling.

Chuanqi Tan, Furu Wei, Wenhui Wang, Weifeng Lv, and Ming Zhou. 2018. Multiway attention networks for modeling sentence pairs. In *IJCAI International Joint Conference on Artificial Intelligence*, volume 2018-July, pages 4411–4417.

Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2016. Improved Representation Learning for Question Answer Matching. pages 464–473.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral Multi-Perspective Matching for Natural Language Sentences.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics*, New Orleans, Louisiana. Association for Computational Linguistics.

Xiang Zhang, Yann LeCun, and Junbo Zhao. 2015. Character-level Convolutional Networks for Text Classification.

A Appendix

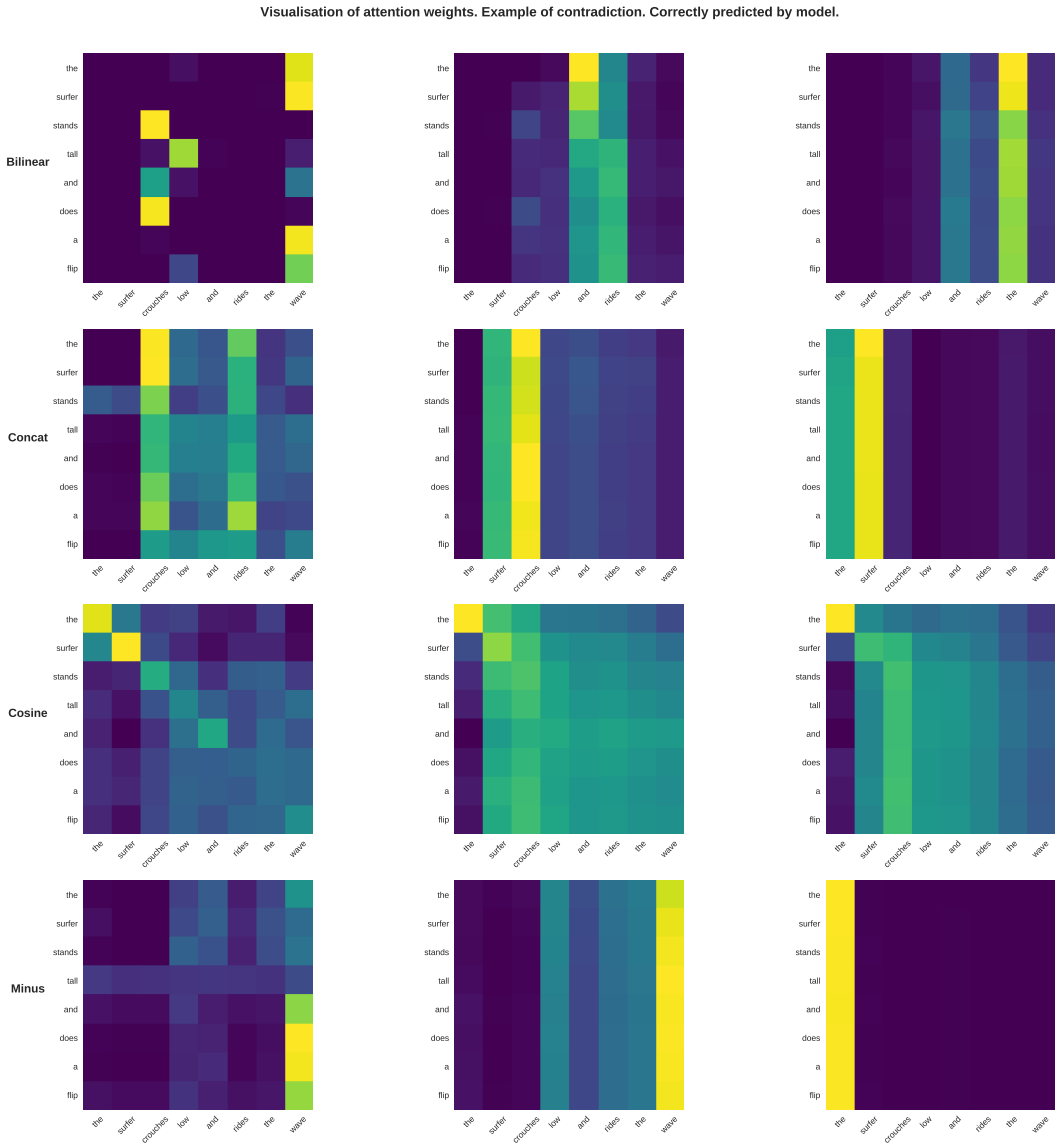

Figure 4: Visualisation of **co-attention** matrices for an example of contradiction (i.e. Hypothesis on Premise). Premise is 'The surfer crouches low and rides the wave' and the hypothesis is 'The surfer stands tall and does a flip'. Each row represents the attention weights of one function, and from left to right they are the outputs of the 1st, 3rd, and 5th layers of the DRCN stack respectively.

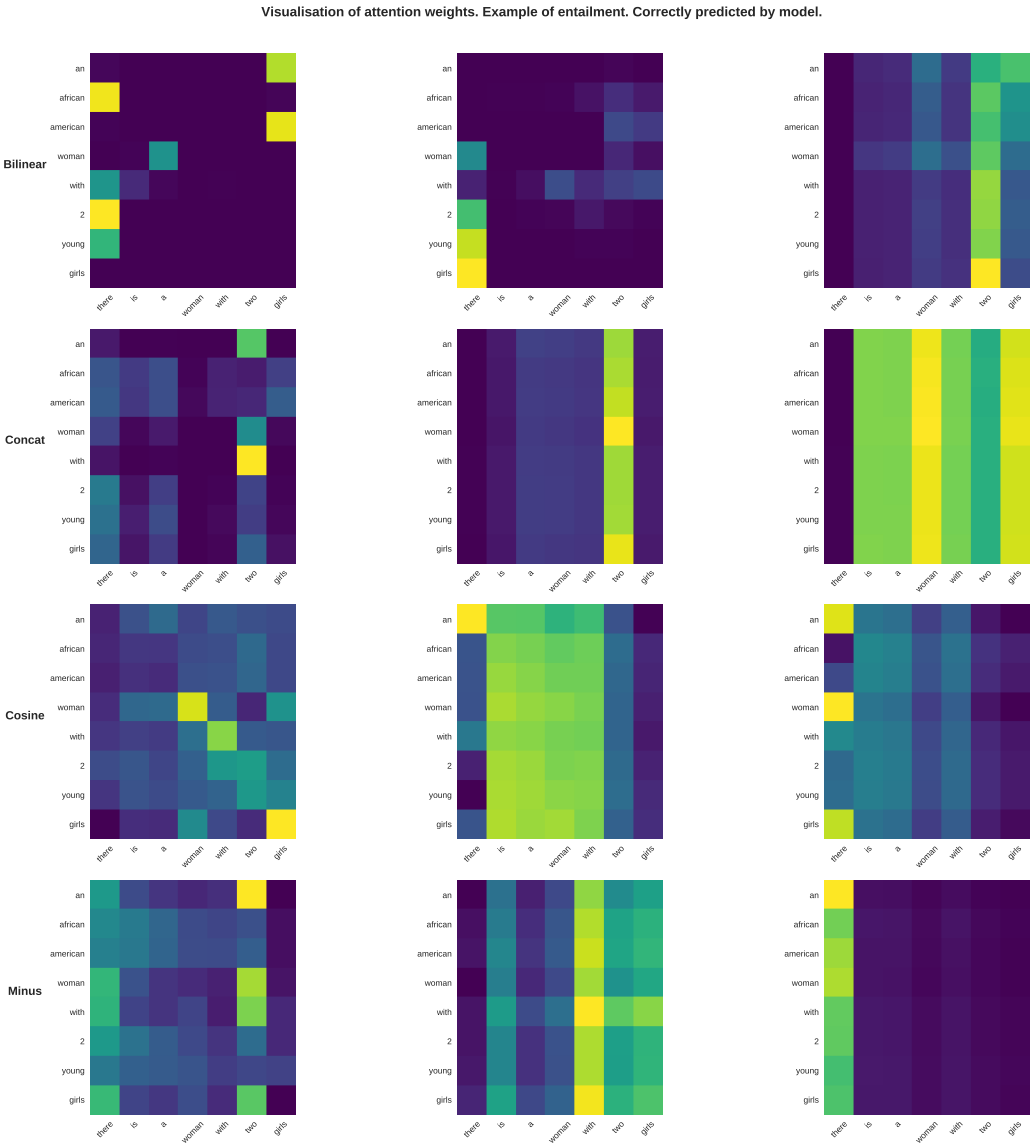

Figure 5: Visualisation of attention weights for an example of entailment. Premise is ‘An African American woman with 2 young girls’ and hypothesis is ‘There is a woman with two girls’.

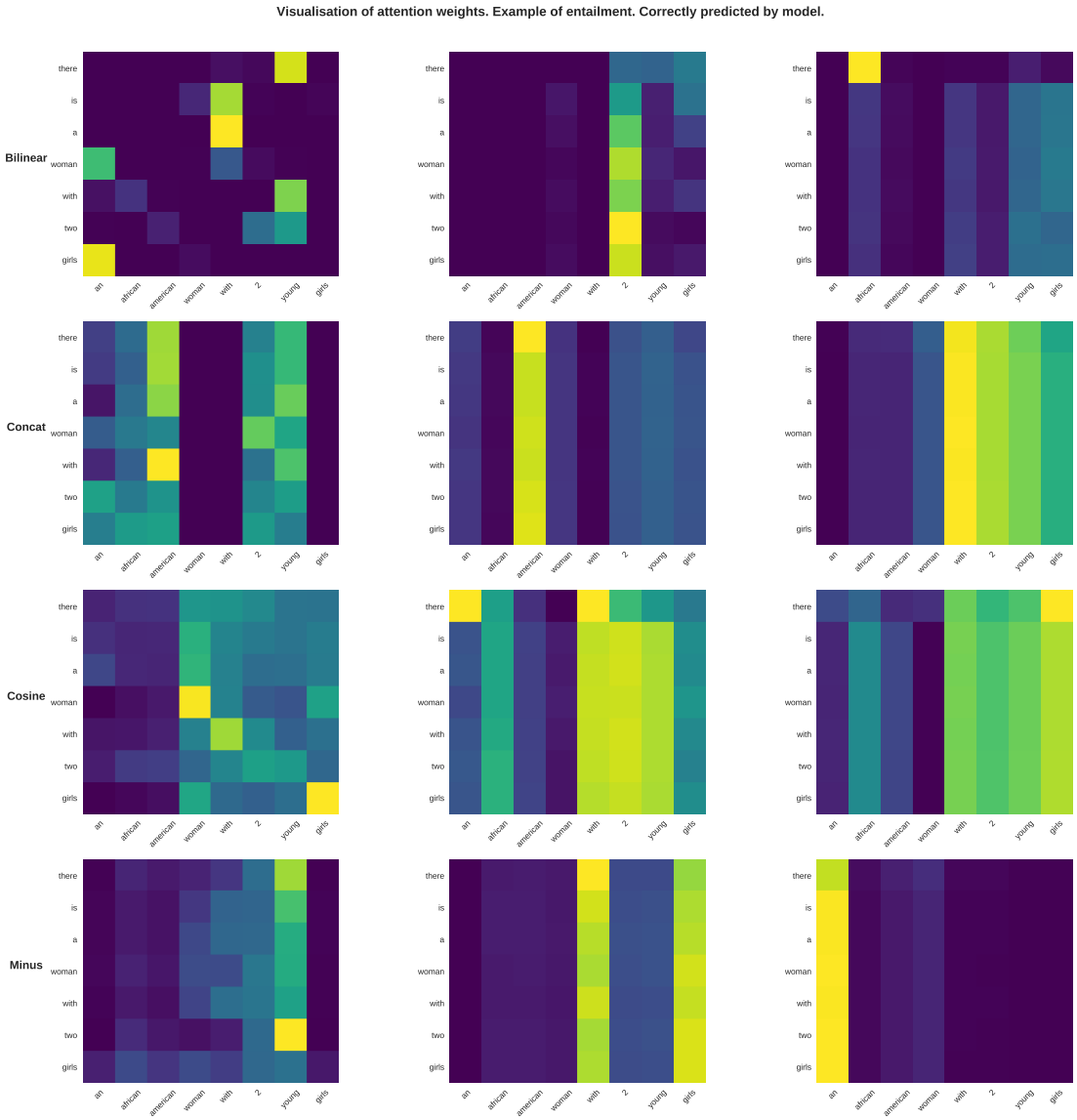

Figure 6: Visualisation of **co-attention** weights for an example of entailment (i.e. Hypothesis on Premise). Premise is ‘An African American woman with 2 young girls’ and hypothesis is ‘There is a woman with two girls’.

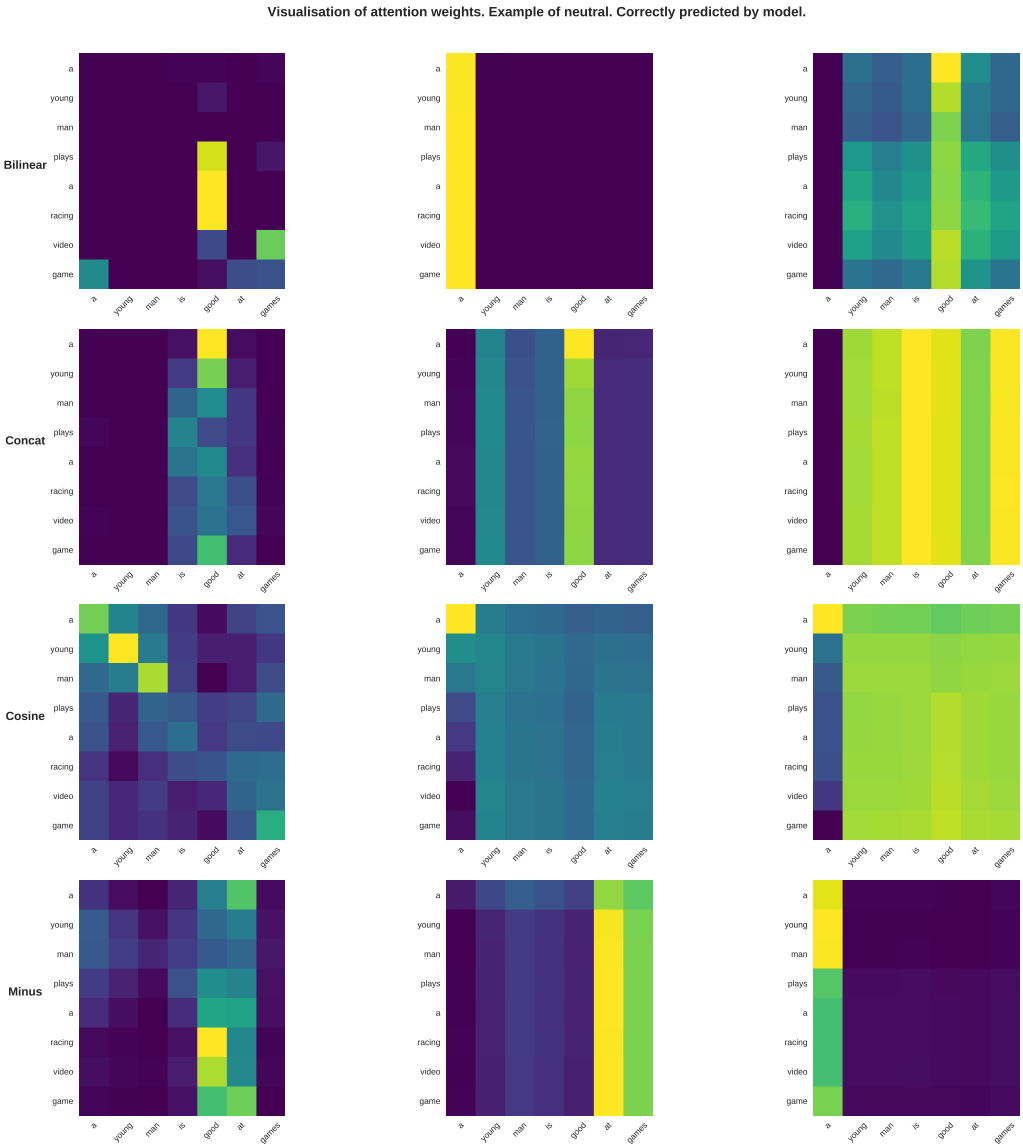

Figure 7: Visualisation of attention weights for an example of neutral. Premise is 'A young man plays a racing video game' and hypothesis is 'a youg man is good at games'

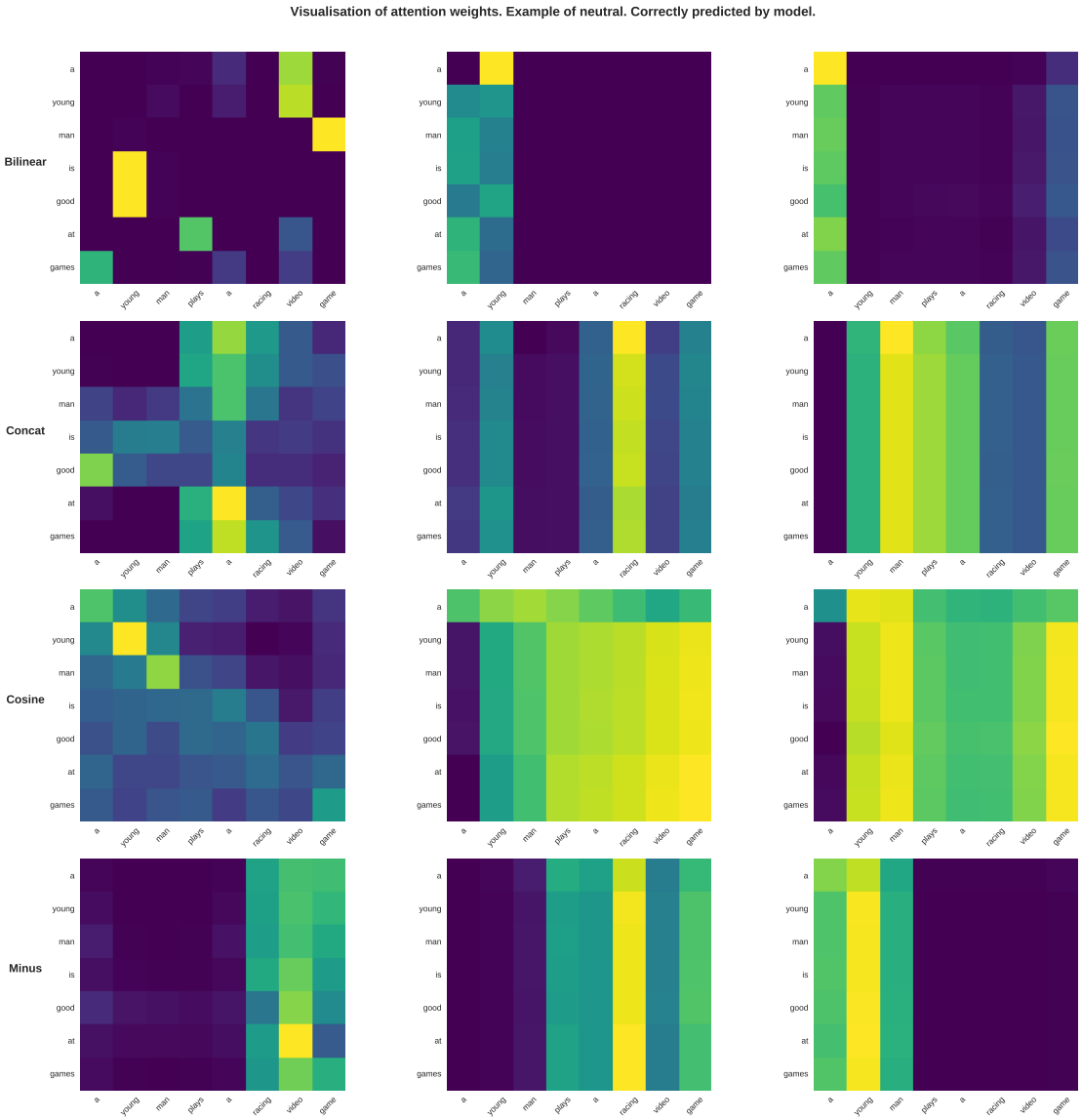

Figure 8: Visualisation of **co-attention** weights for an example of entailment (i.e. Hypothesis on Premise). Premise is 'A young man plays a racing video game' and hypothesis is 'a youg man is good at games'

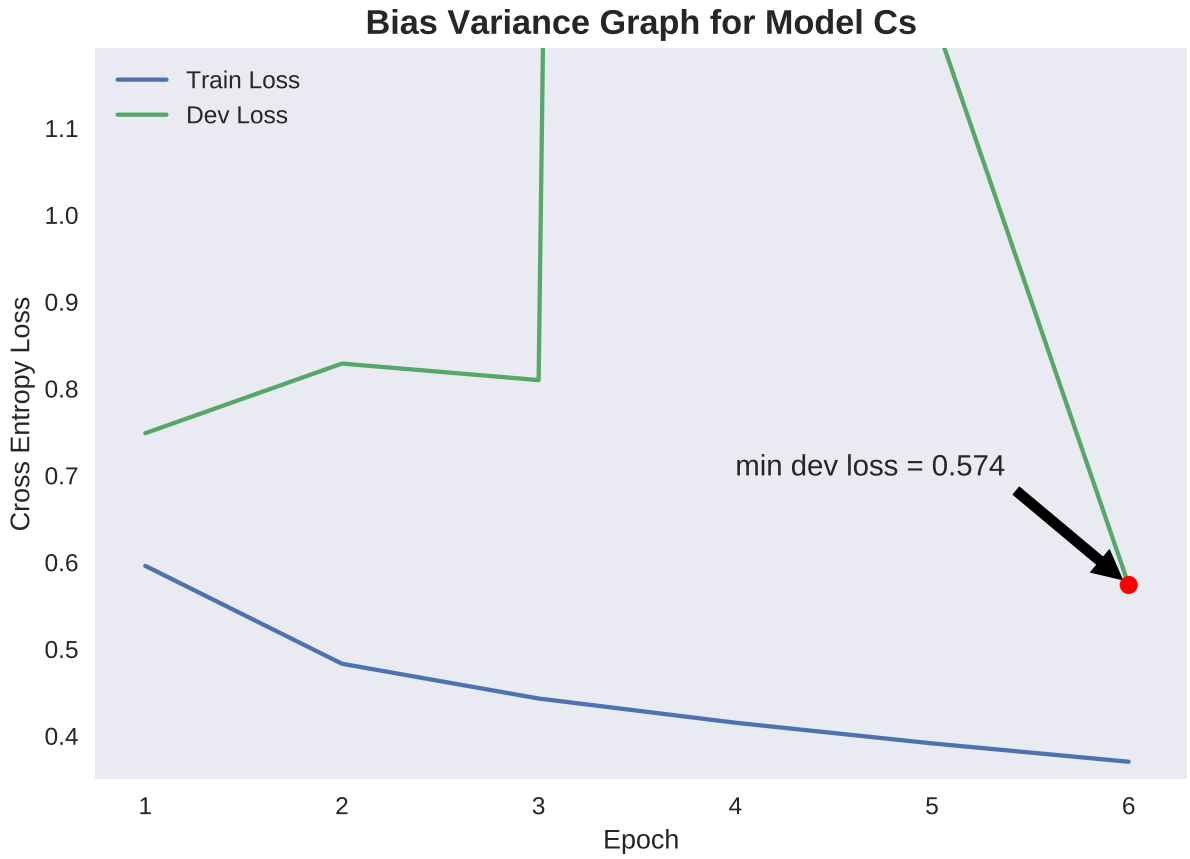

Figure 9: Bias-Variance graph for baseline DRCN model. It stands to reason that performance would improve with more training time.