Comparative Analysis of BiDAF & DCN Research Papers in Machine Comprehension on SQuAD Data.

This article draws a comparison between papers
BI-DIRECTIONAL ATTENTION FLOWFOR MACHINE COMPREHENSION (Seo et al. (v5. 2017)
and
DYNAMIC COATTENTION NETWORKS FOR QUESTION ANSWERING (Xiong et al.(v4 2018)

Hari Prasad – ID:20173074
meethariprasad@gmail.com

Contents

Introduction

Reading a paragraph or document, understanding its contents, answering a question related to the read content is a facet of comprehension, which as humans we do regularly with great amount of accuracy. It has been a challenge, till the advent of deep learning algorithms and infrastructure to do this task with significant accuracy using AI domain.

Stanford Question Answering Dataset(Rajpurkar et al. 2016) (hereby referred as **SQuAD**) is a reading comprehension dataset, consisting of questions posed by crowd workers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading passage. With 100,000+ question-answer pairs on 500+ articles, SQuAD is significantly larger than previous reading comprehension datasets.

Bidirectional Attention Flow for Machine Comprehension (Seo et al. (v5. 2017) (here by referred as **BiDAF**) and Dynamic Coattention Networks For Question Answering (Xiong et al. (v4. 2018) (hereby referred as **DCN**) are two papers published parallelly to solve the SQuAD problem.

This paper is intended to provide the comparative analysis of these two papers. The flow of this comparative analysis consists of

1. Architectural Level Comparison
2. Quantitative Comparison

Architectural Level Comparison

Before we dwell deep in to the deeper level of architectural comparison, we will attempt here to generalize the solution approaches and we will understand and compare how BiDAF and DCN achieve these objectives in their own ways. We will not be trying to explain the model in depth here, but to provide the comparative analysis with respect to how they achieve the parts of the solution in their own way.

In a highly simplified solution outline , here is what both BiDAF and DCN are trying to achieve in a nutshell.

1. Understand the Content with respect to Question.
2. Understand the Question with respect to Content.
3. By the combined understanding of the content in the light of question find the start word of the answer in content.
4. By the combined understanding of the content in the light of question find the end word of the answer in content.

In terms of the architectural terms, again at high level, both BiDAF & DCN solution consists of following steps, which are sequential in nature.

1. Embedding Layer: A layer which embeds the Document and Questions.
2. Attention Layer: This layer consists of the architecture through which we will get our document context with respect to question.

3. Pointer Layer: A layer where start word and end word of the span in the content are pointed out by the architecture.

We have placed the figures from the BIDAF & DCN paper respectively, which we will refer in further comparative analysis together.

BiDAF Architecture (BiDAF Figure 1)

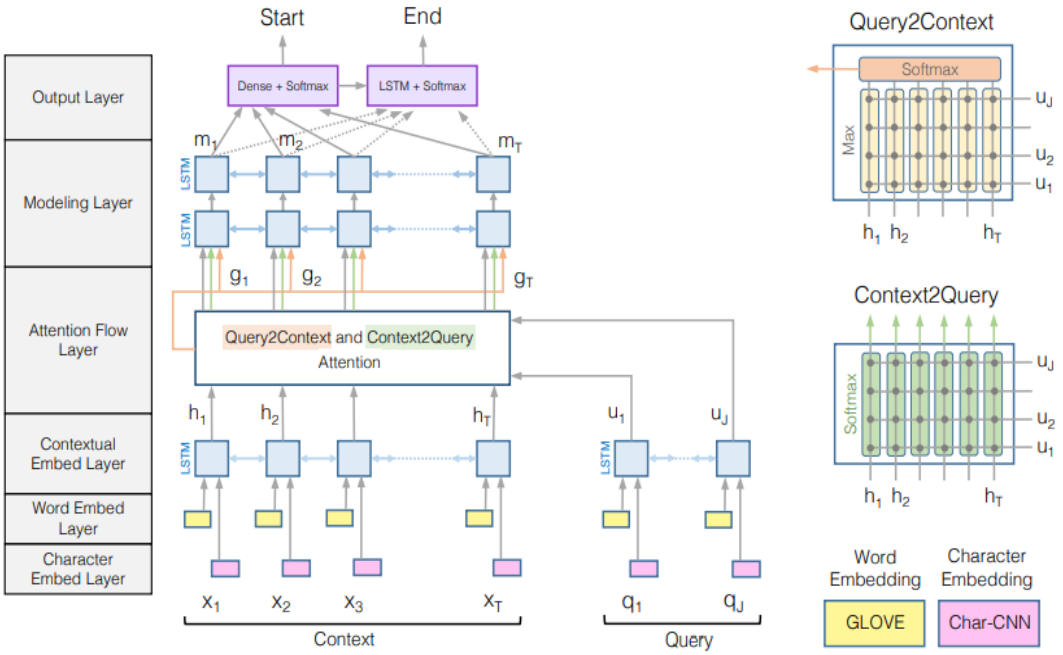


Figure 1: BiDirectional Attention Flow Model *(best viewed in color)*

Dynamic Co Attention Architecture (Figure: DCN1)

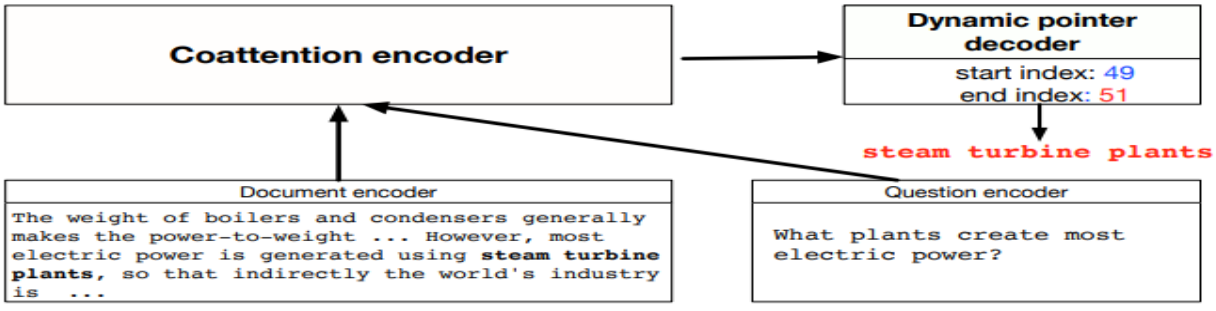


Figure 1: Overview of the Dynamic Coattention Network.

Dynamic Co Attention Encoder (Figure: DCN2)

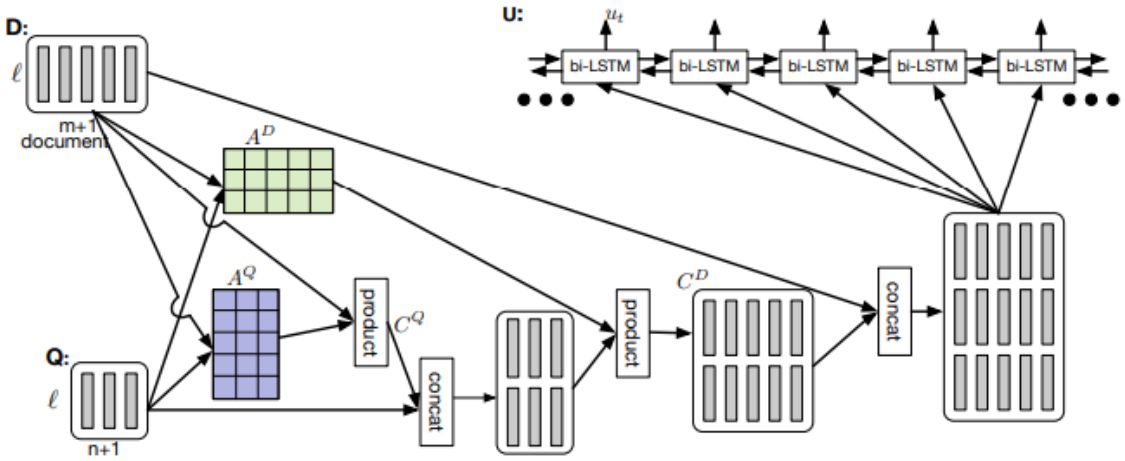


Figure 2: Coattention encoder. The affinity matrix L is not shown here. We instead directly show the normalized attention weights A^D and A^Q.

Dynamic Decoder (Figure: DCN3)

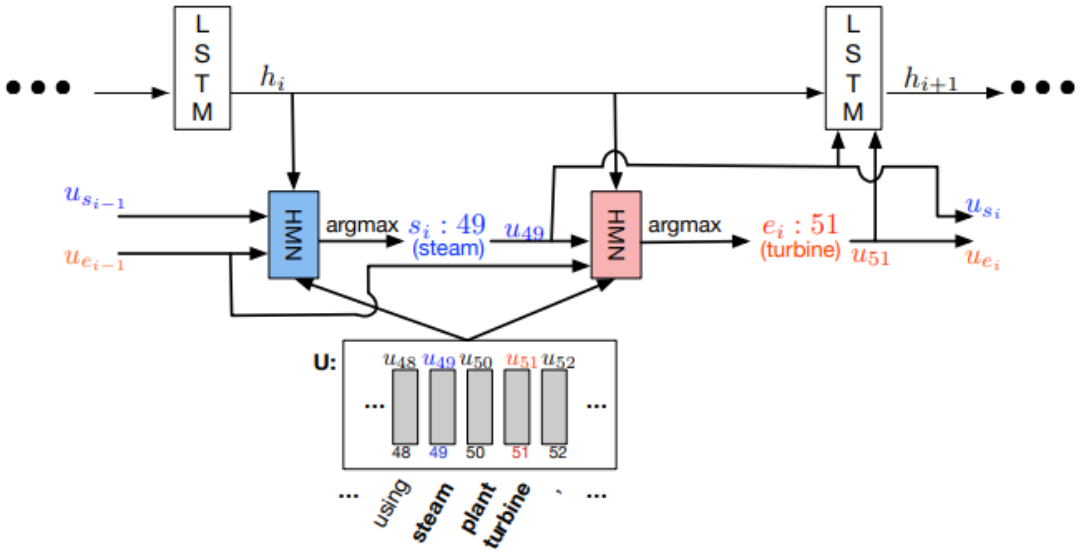


Figure 3: Dynamic Decoder. Blue denotes the variables and functions related to estimating the start position whereas red denotes the variables and functions related to estimating the end position.

Let us look back in to our solution layers and see how BIDAF & DCN works out, keeping in reference to above architecture images.

1. **Embedding Layer:** A layer which embeds the Document and Questions.
   Comparisons:
   BiDAF achieves this task by doing a 3 layer approach. For questions and content, it gets Character Level Embedding using Convolution Layer which are maxpoold to obtain the vector for each word, which is concatenated with word embeddings obtained by GloVE, which is passed further to Highway Networks. Now it also wants to get contextual & temporal interaction between words within, which is achieved through a BiLSTM layer, for both Query and Content. You can refer the character, word and contextual embedding layer part of *BiDAF Figure 1*.

   DCN performs this task more or less similar to DCN, but if we look closely it has some differences in the architecture. At the Document & Question encoding part of the paper, it doesn't mention about having character level encodings, which may be one of the reason because of which DCN might be loosing certain amount of representational opportunity compared to BiDAF. Further they input this word embeddings in a LSTM encoder. Interesting observation arrives when they use same LSTM encoder to share representation power for query also. Overall sense, the DCN can be considered as end to end neural network compared to BiDAF.

2. **Attention Layer**: This layer consists of the architecture through which we will get our document context with respect to question. This is where these two networks excel from earlier uni-directional approaches of single representation of context, by doing a parallel attention to query and context and getting a representation for the each of words in the document, as a result of this.

   The better way to understand and compare this is to understand how each of these architectures get Context to Query attention and Query To Context attention.

   C2Q & Q2C : Context-to-query (C2Q) attention signifies which query words are most relevant to each context word where as Query-to-context (Q2C) attention signifies which context words have the closest similarity to one of the query words.
   a. **BiDAF:** Before calculating the C2Q or Q2C, BiDAF learns a similarity matrix S, which takes the embeddings from previous layer from Query and Document. Now C2Q is achieved as shown in the C2Q section of diagram: *BiDAF Figure 1.* It is a softmax over rows of S further attended, where as Q2C is softmax over columns furher attended. Finally contextual embeddings along with Attention are combined to produce the query aware word matrix. The output of this layer is passed to a bi directional LSTM layer as shown in the BIDAF figure, which created a query aware embedding for the query aware representation of context words.
   b. **DCN:** Approach from DCN for this is similar compared to BiDAF, but appears to computationally more feasibile as it need not learn a scalar for S, (Similiary matrix). It creates affinity scores (. Product of Document Embeddings transposed with Query embeddings). The affinity matrix, as they call will be normalized across column and rows to get the attention weights for each word in the question and

document, respectively. Then it calculates Context matrix (Similar of Context to Query) of Coattention with respect to query by doing dot product of Query embedding with Attention weights of query, similiary Context to query, here by using Attention weights of Query and Query embeddings dot product. Then a co attention matrix is calculated using the mapping of question encoding into space of document encodings as shown below, which is further gets concatenated with document vector to get the temoral information along with the Coattention matrices..

$$C^D = \left[Q; C^Q\right] A^D \in \mathbb{R}^{2\ell \times (m+1)}.$$

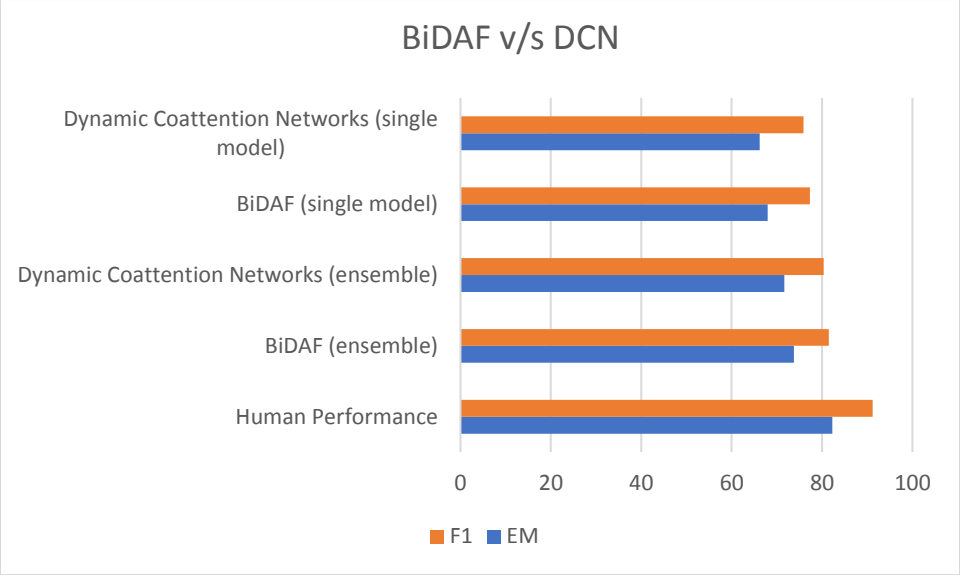
This concatenation is fed to BiLSTM layer further, resulting document word dimensional vector with a vector for each word.

3. **Pointer Layer**: A layer where start word and end word of the span in the content are pointed out by the architecture.
    a. **BiDAF:** BiDAF refers this as output layer.Here it has two separate networks, each calculating probability index over all the words for start and end index, as shown in the BiDAF diagram, by having a sofmax over trainable matrix over the concatenation of model layer M1 (First BiLSTM layer) and M2 (BiLSTM over M1) outputs with output vector obtained by combined representation of C2Q and Q2C (Refered in paper as G), for start and end simultaneously.
    b. **DCN:** DNN implementation a more complex Dynamic Decoding network comprising of Highway Maxout Networks, by implementing a pointer network to pointout the start and end indexes over the argmax on pointers(weights on each words) received by HMN's. Interesting aspect of the paper is usage of HMN networks, with the intuition that there can be multiple model each explaining a pattern and getting the max out of these models to select the model giving least error. This is because authors taken multiple document feed with variation in question in to account and felt it is necessary to select the one which gives maximum relevant answer. At the same time it is quite evident that, they increased the computation complexity of the entire architecture, compared to BiDAF. You can refer figure 3 to visualize. Dynamic Decoder (Figure: DCN3)
4. **Training:**
    a. **BiDAF:** BiDAF defines training loss for minimization as the sum of the negative log probabilities of the true start and end indices by the predicted distributions, averaged over number of samples.
    b. **DCN:** DCN training procedure minimizes cumulative softmax cross entropy of the start and end points across all iterations. Intuitively similar to BiDAF objective, if not mathematically similar.

2. Quantitative Comparison

We will compare the results from BiDAF and DCN on SQuAD dataset using following tables and provide our observations, as of today (Apr 9 -2018). Source: SQuAD. We have also provided chart for better visualization.

Date	Rank	Model	EM	F1
		Human Performance	82.304	91.221
22-Feb-17	36	BiDAF (ensemble)	73.744	81.525
1-Nov-16	43	Dynamic Coattention Networks (ensemble)	71.625	80.383
28-Nov-16	50	BiDAF (single model)	67.974	77.323
1-Nov-16	53	Dynamic Coattention Networks (single model)	66.233	75.896



Observations:

As per above tables, it appears the BiDAF single as well as Ensemble have slight advantage over F1 Score as well as Exact Match score. The difference is not significant if you consider only the numerical differences between scores, but as this is a challenge, even a slight variation can result in significant raise or drop in ranks, as you can see between ensemble models. As we can see that F and EM score pretty much shows stable variations (2 points on average), showing stability of models.

Conclusion and Opportunity for further work

We hypothize that DCN is more computationally intensive model compared to BiDAF by analyzing the architecture. Provided more time, it would be authors effort to implement these two papers and provide the analysis in depth at code and executional level as given by Stanford researchers in other comparative analysis effort.(Ref. 3)

It is evident that as closer as the contextual representation of corpus with respect to question moves without information loss to get the span, the model get better ability to score high. Maybe a better feature engineering can make models, more robust and give ability to score high.

References:

1. BI-DIRECTIONAL ATTENTION FLOW FOR MACHINE COMPREHENSION by Minjoon Seo,Aniruddha Kembhavi Ali Farhadi,Hananneh Hajishirzi: https://arxiv.org/pdf/1611.01603.pdf
2. DYNAMIC COATTENTION NETWORKS FOR QUESTION ANSWERING by Caiming Xiong , Victor Zhong , Richard Socher: https://arxiv.org/pdf/1611.01604.pdf
3. Reading Comprehension using Bidirectional Attention Network. By Neelmani Singh et al. http://web.stanford.edu/class/cs224n/reports/6908966.pdf