Generalizing word vectors in a multi model approach

Kareem Hegazy
Physics Department
Stanford University
Stanford, CA 94305
khegazy@stanford.edu

Abstract

Question answering tasks have greatly benefited from improved datasets and advance techniques borrowed from similar fields, or developed to better encode contextual information. Many of these models, however, generally focus on a single methodology, whereas combining multiple approaches often occurs while ensembling trained models. In this paper we combine different methodologies by reapplying a generalization of representing and contextualizing words. This generalization allows us to combine different methods at varying contextual granularities to exploit the benefits these methods.

1 Introduction

One of the primary tasks to natural language processing is teaching a computer to answer questions given by a human in their native language. Such an achievement provides new ways humans can interact with computers in their daily lives. Question answering (QA) has benefited throughout the past few years from the development of advanced models, adapting techniques from similar fields, and the creation of quality datasets, such as the Stanford question answering dataset (SQuAD) [8], which we use to evaluate our model on. One such model that has become ubiquitous in QA is attention [1], whereas models like convolution neural networks (CNN) [4, 5, 11, 10, 2] are borrowed from computer vision. While all of these methods have their own merits, these networks often heavily rely on a single methodology. One way to mitigate such a focused approach is to create ensembles of models, however, these models are generally trained separately and consequently lack the knowledge basis that comes from directly correlating the models while training. Such correlation learning may benefit contextualizing words since we rely on word correlations to build our own context and understanding.

Consequently, we propose a model that uses various methodologies simultaneously to exploit their strengths and correlations to build different contextual representations. The choice of methods and their order is intuited from a single generalization of a common NLP technique, which we will call the generalized unit. This unit consists of two steps:

1. Create a representation of each word to encode its context and semantics (i.e. word vectors)
2. Contextualize these words in the given sentence (i.e. long short term memory (LSTM) [3]).

We focus on the optimal representations through multiple levels of generalization and the use of various methods to exploit their correlation.

2 Related Work

In order to make computations based on linguistic systems we represent the language as a vector space, where a word's meaning is encoded in the vector's coordinates. Global vectors for word

representation (GloVe), is the standard representation due to its high performance [7]. It is intuited by assuming word-word co-occurrences encode a word's meaning. GloVe uses the ratio of co-occurrences to train the word vectors since the noise from non-discriminant words cancels out so that ratios of words that correlate stand out, encoding meaning within the word vectors. The loss function is derived using two primary assumptions. The first being that all word vector representations used in the derivation should be the same, enforcing symmetry in the equation. The second being that the word vector representation is linear. These assumptions result in a log-bilinear model with a weighted least square loss function.

One of the attention models used to compute an attention representation is bi-directional attention flow (BIDAF). BIDAF is a hierarchical multi-stage attention model that encodes contextual representations of the paragraph with varying degrees of granularity at each stage of the model [9]. BIDAF creates a query-aware context layer by computing the attention for each time step. This attention, along with the attended vector and representations from previous layers, are subsequently sent through a modeling layer consisting of a bi-directional LSTMs in order to "*flow*". In our model we use one of the six BIDAF layers in order to compute the query-aware attention that we use as a basis of our attention representation.

The second attention model we use is the dynamic coattention network (DCN), which is an end-to-end neural network for question answering tasks [12]. DCN consists of an encoder that correlates between the question and the document. The encoder attends to the document and query simultaneously and combines them in the end. We use the coattention encoder as a second method to form an attention representation.

Convolutional neural networks (CNN) are heavily used in computer vision and have become a staple for image related machine learning techniques. CNN models have been shown to improve task-specific word vectors [5], sentence modeling [4], semantic parsing [11], search query retrieval [10], and various other NLP tasks [2]. Training a filter to correspond to a specific semantic entity (either at the word or n-gram level) and convolving it with the sentence allow us to find the location of this semantic entity within the sentence. This technique can be very useful when evaluating the SQuAD dataset as we search for the start and end position of the answer within the document.

3 Approach

The fundamental ideas behind this network is to combine different methods and apply them within our generalized unit. By stacking different methods we exploit their predictive strengths, whereas by stacking generalized units we refine the context of each word and produce contextualized representations with different granularity. The entire model can be seen in Figure 3, where each section of the model is color coded. In the following paragraphs we provide an overview of the model. We refer to "Relu activation" and "linear layer", which are defined as

$$f(\mathbf{x}) = \text{Relu}\left(\mathbf{W}\mathbf{x} + \mathbf{b}\right) \quad \text{and} \quad g(\mathbf{x}) = \mathbf{W}\mathbf{x} + \mathbf{b}$$

respectively, where Relu is the rectified linear unit.

The network begins at the first step of the generalized unit by creating a word embedding layer in which we map each word in our dictionary into a vector space using GloVe [7], called the "word-representation". We then execute the second step of the generalized unit with a bi-directional LSTM to contextualize our word-representation. Although we can stack this exact unit again, we chose a different route.

We again execute the first step of the generalized unit by forming a new representation, however, in this step we form a representation specific to the document/query pair by using attention. In the attention layer we transform our contextualized word-representation into an attention-representation using the output of BIDAF and DCN. This shift in representation is a strong summarization method since each word becomes a weighted summary of either the document or the query.

Next, we continue to give further context to each word in the attention-representation by stacking a modified fully connected layer and a convolution layer. We choose these methods since they are significantly different from what has previously been used and have been shown to produce quality results in various fields. The reason we stack these methods is to correlate the learning between these two methods and our attention methods. Stacking will exploit the advantages they bring to this

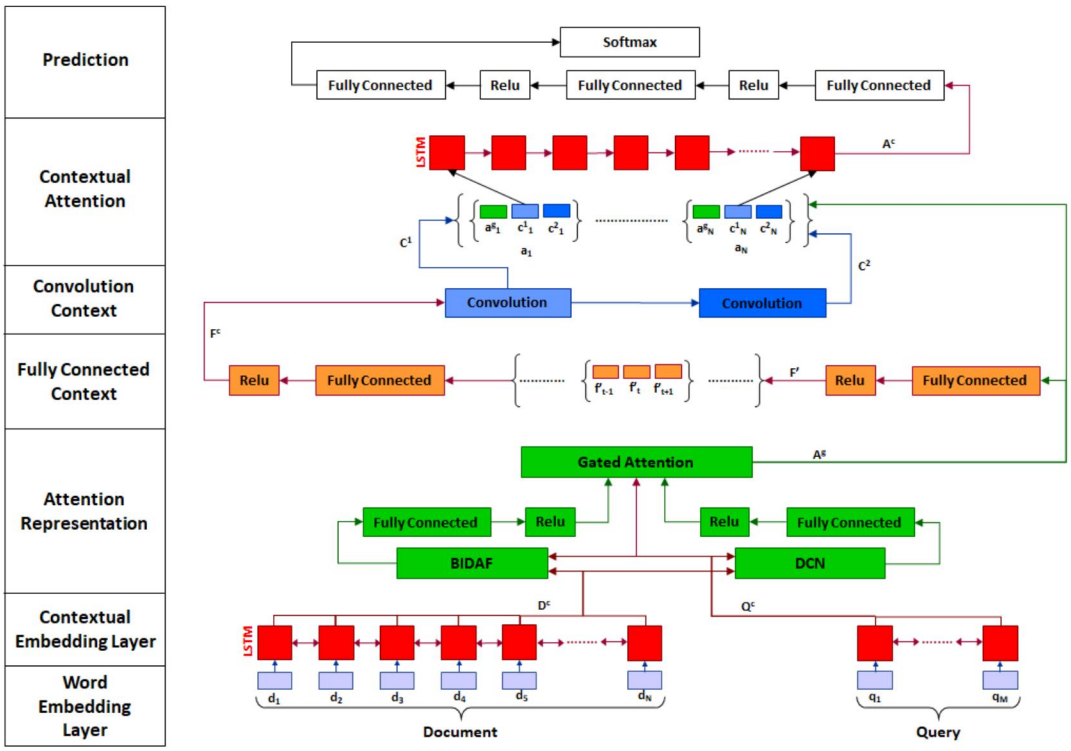

Figure 1: The schematic of the complete neural network contains shows the information flow between layers and illustrates where information with varying degrees of context enter into each layer.

representation that is already built upon models that are well known to produce quality results. We begin to contextualize the attention-representation by mixing the BIDAF and DCN results. Finally we encode nearest neighbor context through a Relu activation layer. The convolution layers provide further context to each word by mixing itself with its neighbors to search for specific semantic meanings. The final contextualized attention-representation is the concatenation of two convolution layers and the input attention layer.

Having completed the first step of the generalized unit by improving the attention-representation through contextualizing it, we perform the second step: contextualize the attention-representation. Therefore, we send the contextualized attention-representation through a LSTM in the attention attention context layer. We use the final output states to find the answer to the query within the document text, this is done by predicting the beginning and end word of the answer within the document. In the following paragraphs we describe the details of each section.

3.1 Word embedding layer

The word embedding layer is responsible for building an E dimensional linear representation of words: the word-representation. This embedding is a pre-trained GloVe [7] representation trained on 6 billion words from Wikipedia and Gigaword. As mentioned before, this is the first step of our generalized unit.

3.2 Contextual embedding layer

Here we take the second step in the generalized unit by contextualizing the GloVe representation. We create a $2H$ dimensional representation of each word in the document and query by encoding contextual information with a bi-directional LSTM [3]. We combine the forward and backward states to form the contextual embedding that we feed into the attention layers $\mathbf{d}_i^c, \mathbf{q}_j^c \in \mathbb{R}^{2H}$.

3.3 Attention representation

After completing the first iteration of the generalized unit by computing the contextualized the word-representation we reapply the generalized unit by building the attention-representation. BIDAF and DCN are the transformation methods we use to shift our representation into the attention-representation. These models were chosen due to their success in addition to their differences. In particular, BIDAF begins with a linear transformation of both $\mathbf{d}^c$, $\mathbf{q}^c$, and their inner product, whereas DCN begins with a nonlinear transformation of $\mathbf{d}^c$. Furthermore, BIDAF ends by fusing together weighted summaries of the document and query, but DCN puts uses a bi-directional LSTM. When trained simultaneously we expect these two models to contextualize words differently, resulting in a more diverse attention-representation. For example, we may find that BIDAF is better at answer "what" and "where" questions, while DCN is better at answering "who" and "when". The resulting BIDAF and DCN attentions are each sent through single Relu activation, resulting in $\mathbf{a}_i^{BDF}$ and $\mathbf{a}_i^{DCN} \in \mathbb{R}^H$.

3.3.1 Gated attention

To encourage BIDAF and DCN to specialize in attending to different question types we introduce a gate $G \in \mathbb{R}^{N \times 2H}$. This gate controls which variables from BIDAF and DCN will be important for the query $(\mathbf{q}^c)$. Thus, the gate depends on the contextualized query $(\mathbf{q}^c)$, as well as the BIDAF and DCN attentions.

To compute the gate we first summarize the contextualized question by projecting onto learned variables $\mathbf{S} \in \mathbb{R}^{H \times G}$. The idea of $\mathbf{S}$ is to create G learned vectors to represent key query words we want to attend to. We do not normalize the projections because we want to ignore the contribution from non key query words, as they will not project strongly onto $\mathbf{S}$. Our final question summary $(\mathbf{q}^s)$ is a sum of the learned key query words weighted by the unnormalized projections, followed by a Relu activation layer and a linear layer.

The input into the gate computation is the question summary $(\mathbf{q}^s)$, the BIDAF attention $(\mathbf{a}_i^{BDF})$, and the DCN attention $(\mathbf{a}_i^{DCN})$. We compute the gate with a single Relu activation layer followed by a single linear layer and a sigmoid activation. Once the gate is computed we compute the gated attention-representation, $\mathbf{A}^g \in \mathbb{R}^{N \times 2H}$ through an element wise multiplication between $\mathbf{G}$ and the attentions $\left[\mathbf{a}_i^{BDF}; \mathbf{a}_i^{DCN}\right] \in \mathbb{R}^{2H}\ \forall i \in N$.

3.4 Fully connected context

After transforming into the gated attention-representation we perform our first contextual layer to further develop the attention-representation through contextualizing it. The objective of this layer is to create a general context layer from which we can build more complicated representations on top of. We firstly apply a Relu activation to the gated attention-representation $(\mathbf{A}^g)$. This activation layer does not change the representation's size and does not mix representations of different words, it instead allows us to mix the gated BIDAF and gated DCN attentions into a preferred representation, which we call $\mathbf{f}_i'$. To contextualize this mixed representation we first concatenate together each word's nearest neighbors with itself and send this through a single Relu activation layer.

$$\mathbf{f}_i^c = \text{Relu}\left(\mathbf{W}^{att}\left[\mathbf{f}_{i-1}' \oplus \mathbf{f}_i' \oplus \mathbf{f}_{i+1}'\right] + \mathbf{b}^{att}\right) \in \mathbb{R}^{3H} \tag{1}$$

This activation layer contextualizes the attention-representation by encoding each document word with context information from its nearest neighbor.

3.5 Convolution context

We continue contextualizing the attention-representation by stacking two convolution layer on top of the fully connected context layer to further. The first convolution layer uses H filters, each filter spans five words, to perform a 1d convolution with a Relu activation at the end. This convolution results in a representation $(\mathbf{c}_i^1 \in \mathbb{R}^{N \times H})$ that encodes information from each word's nearest and next nearest neighbors. If we consider the context from the fully connected context layer, then each filter contains information from seven words in total. The second convolution layer follows the same procedure to produce $\mathbf{c}_i^2 \in \mathbb{R}^{N \times H}$, but convolves over the output of the first convolution

layer. In this layer each filter contains information from 11 words. These convolutions produce H entries corresponding to different semantic entities. By stacking these layers we further generalize the semantics into broader categories which will help in deciding the start and end location of answer within the document.

3.6 Contextual Attention

At this point we have finished the first step in the generalization unit by computing multiple representations: gated attention-representation, as well as contextualized attention-representations generated through fully connected layers and convolution layers. These layers all offer a different contextual representations of the document at various network depth and consequently provide different granularity of information. Furthermore, these representations are specific to the given document/query pair. We naturally continue to the second step of the generalized unit, which is to contextualize this representation using a LSTM.

To derive our final context layer, we feed the LSTM context attention-representations with different contextual granularity in order to help the LSTM understand short and long range relationships. Hence, the LSTM input is the concatenation of the gated attentions and the output of each convolution context layer. We chose to include the gated attention since this forms the basis of the attention-representation, and to provide stronger gradient flow into the attention calculations. The LSTM produces the output $\mathbf{a}_i^c \in \mathbb{R}^{2H}$ which we use to make our final prediction.

3.7 Prediction

Although the previous layers contextualize the words, such a representation is not necessarily optimal for making our prediction. Therefore, before predicting the location of the start and end word we send the output of the attention context layer through three Relu activation layers with a final linear layer. These few layers are added to find a better representation to make the final prediction in. We evaluate the prediction of the beginning and end word of the answer by applying softmax to the output of the last linear layer.

3.8 Training

This model was trained to minimize the cross-entropy loss of predicting both the start and end word. The parameters were updated using the Adam optimization method [6] and regularized using dropout. We additionally applied gradient clipping at each training step in order to avoid exploding gradients.

4 Experiments

To illustrate the strength of the generalized unit, as well as the effectiveness of stacking different methods we evaluate three separate models, each at a different layer of generalization. These models were trained and evaluated on the SQuAD dataset with a dropout probability of 15% and with the following parameters: $H = 200$, $G = 400$, $N = 600$, $M = 30$, and an initial learning rate of 0.001. We evaluate our model using the exact match (EM) score and the F1 score.

	Dev Dataset		**Test Dataset**	
Model	**F1**	**EM**	**F1**	**EM**
Attention Only	56.438	45.582	N/A	N/A
Context Attention-Representation	63.861	53.519	63.577	53.782
Full Model	70.922	60.274	71.5	61.576

4.1 Attention only

Our benchmark model makes predictions using the gated attention-representation without any further context layers. The predictions are made by concatenating the gated attention $(\mathbf{a}_i^g)$ with $\mathbf{d}_i^c$ and sending this through two Relu activation layers and a final linear layer of sizes $2H$, H, and H. We apply a softmax to the result of the final layer to predict the beginning and end word of the answer.

This model was trained for 15 epochs and received a F1 score of 56.4% and an EM score of 45.6% on the development set. These numbers are quite low given state of the art models achieve F1 scores around 89% and EM scores around 82%. Thus we can see that attention alone is not a sufficient predictor alone.

Qualitative inspection of the results reveal four common errors. The most common error comes from predicting the end word to be before the start word. This indicates that the model has a terrible understanding of the document, in particular the spatial dependence of words. This is most likely due to the lack of context in the attention-representation. The second most common mistake occurs when part of the question should have been paraphrased in the answer, as shown below.

- QUESTION: relegation to secondary status for abc resulted in viewership how much lower than their competitors , according to goldenson ?
- TRUE/PREDICTED ANSWER: five times lower viewership / five times

In many cases we see that the model's dependence on the overlap of the query and document, which causes the network to answer the query with a phrase that includes a key query words.

- QUESTION: which theory states that slow geological processes are still occurring today , and have occurred throughout earth 's history ?
- TRUE/PREDICTED ANSWER: uniformitarianism / this theory

The final most common error is that the network has a vague understanding of the query and document and gives an answer that may seem suitable, but is not correct, as shown below.

- QUESTION: besides the arguments with rome and his own fellow reformers , what scandal contributed to luther 's failing health ?
- TRUE/PREDICTED ANSWER: bigamy of the philip of hesse / kidney and bladder stones , and arthritis

4.2 Context attention-representation

The second experiments predicts the answer based upon the context attention-representation. This experiment reveals the information gain through contextualizing the attention-representation using the fully connected layer and the convolution layer. After the convolution layer we concatenate the results of both convolution layers with the gated attention results, which we would have fed into the bi-direction LSTM. Instead, we send these representations through two Relu activation layers and a single linear layer of sizes $4H$, $2H$, and H before we predict the beginning and end word using softmax. This model was trained for roughly 10 epochs and received a F1 score of 63.9% and an EM score of 53.5% on the development set as well as a F1 score of 63.6% and an EM score of 53.8% on the test set. Adding these contextualizing layers resulted in a relative improvement of 13% for the F1 score and 18% for the EM score. This indicates that the attention only model had a poor representation of the context, and that further context can significantly improve our representation of the words.

Qualitatively, the context attention-representation does a better job at finding the answers, however, in many of its mistakes it is selecting a large swath of text that does include the answer. Furthermore, the model still suffers from predicting the end word before the start word, as well as paraphrasing the query in the answer.

- QUESTION: previous to isotopic dating sections of rocks had to be dated using fossils and stratigraphic correlation relative to what ?
- TRUE/PREDICTED ANSWER: to one another / one another

As before, the model still suffers from selecting answer that are nearby or include a key query word, such as in the example below.

- QUESTION: which theory states that slow geological processes are still occurring today , and have occurred throughout earth 's history ?

- TRUE/PREDICTED ANSWER: uniformitarianism / this theory

Moreover, it still continues to make mistakes by selecting an answer that may seem sensible, but is not the correct answer, indicating that the model did not fully understand the context in which its answer and the true answer were given in.

- QUESTION: what division offers more then one branch of studies that don't fit in with the other four ?
- TRUE/PREDICTED ANSWER: the new collegiate division / biological sciences collegiate division

4.3 Full model

The final experiment evaluates every layer in Section 3. The importance of this test is to validate the final generalization that we can treat the context attention-representation as analogous to the embedding layer and contextualize it with a LSTM. This model was trained for roughly 13 epochs and received a F1 score of 70.9% and an EM score of 60.3% on the development set, as well as a F1 score of 71.5% and an EM score of 61.6% on the test set. This is a relative increase of 11% for the F1 score and a 13% increase in the EM score. This significant increase appears to validate our generalized unit model and the assumption that we can treat the contextualized attention-representation as analogous to the word-representation.

The most common mistake is still predicting the end word before the start word, indicating that improvements on the attention based representation may not be enough to understand certain document and query pairs. However, this error accounts for 25% more of the total errors as it did in the attention only model. Moreover, the full model does not make mistakes based upon vague understandings as often as it did before, and when it does we see that these mistakes are not as simple as including a key query word. Instead the model is able to abstract the meaning of key query words based on the context, such as in the example below,

- QUESTION: what method is used to intuitively assess or quantify the amount of resources required to solve a computational problem ?
- TRUE/PREDICTED ANSWER: mathematical models of computation / the theory formalizes this intuition

where the network has confused "model" for "the theory". Another indication that we further contextualized the model comes from the example below where both previous models failed to answer by locking onto a key query word, whereas the full model was able to find a reasonable answer that included the correct answer.

- QUESTION: which theory states that slow geological processes are still occurring today , and have occurred throughout earth 's history ?
- TRUE/PREDICTED ANSWER: uniformitarianism / charles darwin , successfully promoted the doctrine of uniformitarianism

This is an improvement over both attention representation models where the network would select answers with the exact word in it. From these improvements, we may infer that the full model either has a good contextual understanding of the document and query pair to answer the query, or has very little understanding of the context and location dependence. Furthermore, the full model performs better at answering queries where the query is paraphrased in the answer.

- QUESTION: relegation to secondary status for abc resulted in viewership how much lower than their competitors , according to goldenson ?
- TRUE/PREDICTED ANSWER: five times lower viewership / five times lower

However, these mistakes still account for a large portion of the errors this model makes. This indicates that the improved context of the full model can now answer more questions, but still cannot answer some questions it was originally confused on.

5 Conclusion

These experiments have shown that applying the generalized unit to contextualize our representation is beneficial to the machine's understanding and ability to solve the QA problem. Furthermore, we also see that stacking various methods and training them together can improve the models understanding as well. To further test our assumption we would like to apply another generalized unit to the output of the attention context layer. This would give us further insight into the effectiveness of the generalized unit. We believe that taking this approach of applying the generalized unit, and stacking various methodologies within this unit, will lead to more advanced and models that may outperform traditional ensembling.

Acknowledgments

We would like to acknowledge Professor Richard Socher and the CS224n TA team for the great class and all the work they put into it. We would also like the thank Microsoft Azure for the ability to perform our experiments on their cloud.

References

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

[2] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

[3] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[4] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014.

[5] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[6] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[7] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[8] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250, 2016.

[9] Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *CoRR*, abs/1611.01603, 2016.

[10] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. A latent semantic model with convolutional-pooling structure for information retrieval. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 101–110. ACM, 2014.

[11] Wen tau Yih, Xiaodong He, and Christopher Meek. Semantic parsing for single-relation question answering. In *ACL*, 2014.

[12] Caiming Xiong, Victor Zhong, and Richard Socher. Dynamic coattention networks for question answering. *CoRR*, abs/1611.01604, 2016.