How Self-Attention Improves Rare Class Performance in a Question-Answering Dialogue Agent

Adam Stiff, Qi Song, Eric Fosler-Lussier
Department of Computer Science and Engineering
The Ohio State University
stiff.4@osu.edu, song.1602@osu.edu,
fosler-lussier.1@osu.edu

Abstract

Contextualized language modeling using deep Transformer networks has been applied to a variety of natural language processing tasks with remarkable success. However, we find that these models are not a panacea for a question-answering dialogue agent corpus task, which has hundreds of classes in a long-tailed frequency distribution, with only thousands of data points. Instead, we find substantial improvements in recall and accuracy on rare classes from a simple one-layer RNN with multi-headed self-attention and static word embeddings as inputs. While much research has used attention weights to illustrate *what* input is important for a task, the complexities of our dialogue corpus offer a unique opportunity to examine *how* the model represents what it attends to, and we offer a detailed analysis of how that contributes to improved performance on rare classes. A particularly interesting phenomenon we observe is that the model picks up implicit meanings by splitting different aspects of the semantics of a single word across multiple attention heads.

1 Introduction

Many semantic classification tasks have seen a huge boost in performance in recent years (Wang et al., 2018, 2019), thanks to the power of contextualized language models such as BERT (Devlin et al., 2019), which uses a Transformer (Vaswani et al., 2017) architecture to produce context-specific word embeddings for use in downstream classification tasks. These large, data-hungry models are not always well suited to tasks that have a large number of classes or relatively small data sets (Mahabal et al., 2019).

One task having both of these inauspicious properties is the Virtual Patient corpus (Jin et al., 2017), a collection of dialogues between medical students and a virtual patient experiencing back pain. The corpus contains examples of nearly 350 questions that the virtual patient knows how to answer, and the interaction is modeled as a text-based conversation in which the human, as the interviewer of the patient, always has the conversational initiative. Thus, the corpus represents a question identification task from the perspective of the dialogue agent, in which natural language inputs must be mapped to semantically equivalent classes, so that the appropriate fixed response can be returned to the user to achieve the desired pedagogical objectives.[1]

Many of the classes in this task are distinguished in subtle ways, e.g., in degree of specificity ("Are you married?" vs. "Are you in a relationship?") or temporal aspect ("Do you [currently] have any medical conditions?" vs. "Have you ever had a serious illness?"). A few classes are very frequent, but many appear only once in the data set, with almost three quarters of the classes comprising only 20 percent of the examples (Jin et al., 2017).

The current best approach to this task uses an ensemble of Text CNNs (Kim, 2014) combined with a rule-based dialogue manager (Wilcox, 2019) via a logistic regression model, to leverage complementary performance characteristics of each system on the rare classes (Jin et al., 2017). This approach naïvely treats all classes as orthogonal, so the semantic similarity of the classes above can be problematic. Ideally, a model should be able to learn the semantic contributions of common linguistic substructures from frequent classes, and use that knowledge to improve performance when those structures appear in infrequent classes.

We hypothesize that multi-headed attention mechanisms may help with this kind of generalization, because each head is free to specialize, but should be encouraged to do so cooperatively to

[1] We are currently working to anonymize this corpus, and we will release code and data at https://github.com/OSU-slatelab/ when it is available.

, pages 307 - 313
July 1-3, 2020

maximize performance. Three different methods of utilizing BERT-based architectures for this task surprisingly did not improve upon the performance of the CNN models of Jin et al. (2017). In contrast, a very simple RNN equipped with a multi-headed self-attention mechanism improves performance substantially, especially on rare classes. We assess the reasons for this using several techniques, chiefly, visualization of severely constrained intermediate representations from within the network, and agglomerative clustering of full representations. We find evidence that independent attention heads: 1) represent the same concepts similarly when they appear in different classes; 2) learn complementary information; and 3) may learn to attend to the same word for different reasons. This last behavior leads to discovery of idiomatic meanings of some words.

2 Related Work

Self-attention, in which a model examines some hidden representation to determine which portions of that representation should be passed along for further processing, became prominent relatively recently (Vaswani et al., 2017; Lin et al., 2017). These models have been very successful for some tasks (Wang et al., 2019), but other approaches may work better for classification tasks with many classes and few examples (Mahabal et al., 2019). We explore two types of self-attentive models for a virtual patient dialogue task (Danforth et al., 2013; Jaffe et al., 2015), which has many classes and scarce data. Previous authors have used memory networks (Weston et al., 2015) to improve performance on rare classes for this task (Jin et al., 2018).

Despite the contrast presented above, our self-attentive model actually shares characteristics with the work by Mahabal et al. (2019), as we find that individual word tokens carry parallel meanings.

We present a detailed analysis of our model's behavior using clustering and visualization techniques; this bears a resemblance to the analysis by Tenney et al. (2019), although they use internal representations to make predictions for linguistic tasks, rather than examining correlations between representations and individual input tokens.

3 Task and Data

As described above, our task is a text-based question-answering task for an agent that has a fixed set of responses. The goal is to classify input queries as paraphrases of canonical questions that the agent knows how to answer, so we call this a question identification task.

Data are collected from actual user interactions with a virtual patient, which is a graphical avatar with a text input interface and subtitles as output. After collection, the system's responses are annotated as correct or not, and if not, annotated with the correct label. Jin et al. (2017) used a data set consisting of 4,330 inputs, comprising 359 classes. We extended this data set by replicating the hybrid system described in their work, and deploying it to collect more data. This resulted in a combined data set of 9,626 examples over 259 dialogues.

We noticed that the annotation method for the data used by Jin et al. (2017) introduced a bias for classifications that produce acceptable *responses*, since only examples deemed to be incorrect were reviewed to identify the correct class. Since our evaluation metrics are on the basis of classes and not the agent's responses, we re-annotated every example, with the aim of maximizing the semantic equivalence of members of the same class. This resulted in the elimination and addition of some classes, leaving 348 in the re-annotated set. The long-tailed distribution is no less a problem in the re-annotated set than in the original, but our baseline outperforms theirs since we use cleaner data.

We hold out 2,799 examples from the combined set as a test set, and perform tenfold cross-validation on the training set for development. The test set only contains 268 classes, but fifteen are unseen in the training data (other than the canonical question, see Appendix A).

4 Experimental Design and Results

We start from a **Text-CNN** baseline for this task (Jin et al., 2017), utilizing a single stream system for comparisons. This system convolves GloVe word embeddings with 300 filters of widths 3, 4, and 5; the max of each filter over the sequence serves as input to a fully connected leaky ReLU layer (Nair and Hinton, 2010), followed by a softmax layer.

We compare this against two contextual models: the relatively well known **Fine-tuned BERT** (Devlin et al., 2019) using the pretrained base model [2], as well as a variant of a simpler **RNN model with**

[2] https://github.com/google-research/bert

System	Acc. (%)	F1
Baseline CNN	80.7	55.6
BERT Fine-tune	79.8	46.6
Self-attention RNN	**82.6**	**61.4**
BERT Static CNN	76.9	49.4
BERT Contextual CNN	75.3	45.2
Mean-pool RNN	81.8	59.4
Bottleneck RNN	80.8	57.2

Table 1: Dev set results comparing different models (top, Sec. 4), word embeddings (middle, Sec. 5.1), and attentional mechanisms (bottom, Sec. 5.2).

self-attention (Lin et al., 2017).[3] Note that despite extensive experimentation, only minor modifications of the work of Lin et al. (2017) proved beneficial for our task, so the architecture we describe here is not a novel contribution.

The self-attentive RNN is a single-layer BiGRU (Cho et al., 2014) equipped with a two-layer perceptron that takes hidden states as inputs, and produces one attention score for each of eight attention heads, for each input step. These scores are then softmaxed over the input, and the attention-weighted sum of the corresponding hidden states serves as the value of the attention head. These values are concatenated and fed into a fully connected layer with tanh activations, and a softmax output determines the class. We use dropout of 0.5 in the attention module and in the fully connected layer. The size of hidden states in the BiGRU is 500 dimensions (in each direction), the size of the hidden layer in the attention module is 350 units, and the fully connected classification layer has 500 dimensions. The original model utilizes an orthogonality constraint on the attention vectors for each attention head, but we find that this is detrimental to our task, so we disable it.

Training parameters for all three models are provided in Appendix A.

The development set results (top 3 lines of Table 1) were a bit surprising to us: while we expected that contextual models would outperform the baseline CNN, fine-tuned BERT performed comparatively poorly. The Self-attention RNN, however, performed significantly better than the baseline CNN, which carries over to a smaller degree to the test set (CNN: 76.2% accuracy, 51.9% F1; RNN: 79.1% accuracy, 54.7% F1).[4] A breakdown of accuracy by class frequency quintiles for the test results is shown in Figure 1, to emphasize the relationship between F1 and rare class performance.

In particular, the BERT model has a very low F1, likely because of the large number of subtly distinguished classes, the relatively small data set, and the high degree of freedom in the BERT model. That is, BERT may be representing semantically similar sentences in nearby regions of the representation space, but with enough variation within those regions that our training set does not permit enough examples for the classifier to learn good boundaries for those regions. Alternatively, the masked language modeling task may simply not induce the grammatical knowledge required to distinguish some classes well.

The success of one attention-based contextual model (Self-attention RNN) and the failure to improve of another (Fine-tuned BERT) led us to ask two analytical questions: first, are the BERT representations not as appropriate for the Virtual Patient dialog domain compared to GloVe embeddings? Second, is there something that we can learn about how the attention-based method is helping over the CNN (and particularly on F1)?

5 Analysis

5.1 Why did BERT perform less well?

The difference in accuracy from the baseline CNN model to the BERT fine-tuning result is fairly small, while the drop in F1 is substantial. Since there are many more infrequent than frequent classes, this suggests that BERT is seriously underperforming in the least frequent quintiles, and making up for it in the most frequent. That, in turn, supports the interpretation that small numbers of examples are inadequate to train a classifier to handle the variation in representations that come out of a contextualized model. This would be consistent with other research showing poor performance of BERT in low-data regimes (Mahabal et al., 2019). Some of the discrepancy may also be explained by a domain mismatch. The BERT base model is trained on book and encyclopedia data (Devlin et al., 2019), to provide long, contiguous sequences of text. In contrast, our inputs are short, conversational, and full of typos. GloVe.42B, trained on web data (Pennington et al., 2014), may simply be a better fit for

[3] https://github.com/ExplorerFreda/Structured-Self-Attentive-Sentence-Embedding

[4] We only tested on the baseline and best system in this paper to minimize use of the test set for future work.

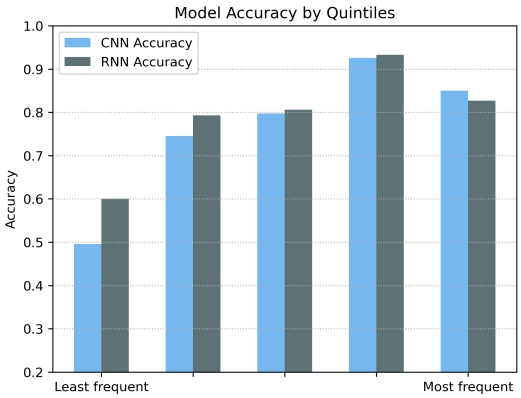

Figure 1: Quintile accuracies for the tested RNN and CNN baseline

our corpus.

To try to tease apart the contributions of model architecture and learned representations, we utilized two different embeddings within the CNN: the **contextual BERT** embeddings from the first layer[5] of the BERT model, and a **static BERT** embedding for each token calculated from the average contextual embedding over all instances of the token in our corpus.

The worst of our BERT-based models is the full contextualized embeddings fed into the baseline CNN. Since the classification architecture is the same as the baseline, this suggests that a significant contributor to the reduced performance of the BERT-based models is the contextualized representations themselves. It seems that stable representations of lexical items are beneficial for generalizing to unseen sentences when few training examples are available. Consistent with this, the static BERT CNN result, despite a lower accuracy than the fine-tuning result, shows a gain in F1. Again, this supports the idea that variation is harmful for rare classes, since stable representations of informative words for those classes help.

5.2 Analyzing the Self-attention RNN

One question is how much attention versus recurrency is playing a role in the Self-attention RNN's improvements. We replaced the attention mechanism with **mean pooling** over the input, controlling for parameter counts by replicating the mean hidden state once for each attention head; Table 1 shows that performance is intermediate between the CNN and the self-attentive RNN, suggesting that the attention does play a role.

To better understand the behavior of the self-attentive RNN, we employ a relatively novel method of analyzing attention: we insert **bottleneck layers** of just eight dimensions after each attention head, with sigmoid activations and no dropout. This adds another nonlinearity into the model, but reduces the total number of parameters substantially. Color coding gives an easily interpretable representation of both *what* each head is attending to, as well as *how* it represents it. Examples are shown in Figure 2. The bottleneck RNN and CNN have similar overall performance (Table 1), but the RNN's performance on the least frequent classes is still superior.

By finding the greatest Jensen-Shannon divergence between predictions made by the baseline CNN and the RNN, as well as the largest change in class recall between the systems, we can identify interesting cases illlustrating the benefit of the RNN system. One compelling case is the difference between *Do you drink [alcohol]?*, *Do you drink coffee?*, and *Do you drink enough fluid?* (classes 85, 86, and 87 in development data). The *Do you drink?* class is very frequent, while the other two are in the least frequent quintile. Since *drink* by itself implies alcohol, the trigram *do you drink* is highly predictive of the alcohol class, and the CNN almost always errs on the other classes.

The RNN, on the other hand, handles this distinction quite well. In all cases, *drink* is attended by multiple heads (Figure 2), but across the set most of the heads are focused on representing the verb itself, while the magenta and tan representations (third and last row, respectively) are representing the object of the drinking. In the absence of an object, the object-focused head lands on the verb itself, and learns the implicit meaning of alcohol from the supervision.

We confirm that this behavior persists in the full model by performing agglomerative clustering on the full head representation in the test RNN. We see that the head that attends most strongly to *water* and *coffee* also often represents *alcohol* and *drink* in the same cluster. Meanwhile, other heads attend to the verbal meaning of *drink*, and encouragingly, these representations cluster nearby to similar consumption verbs such as *use* in the context of illegal drugs (Stiff, 2020). This may be expected due to the pretrained word vectors, but we also observe clusterings of apparently unrelated words like *take*

[5]Empirically, and surprisingly, this worked better than other layers.

Figure 2: Example inputs with bottleneck attention head representations. The colored underlines show the foci of the attention heads, with opacity reflecting attention weights. The activation patterns in the correspondingly-colored rows of the grid representations reflect *how* the attended tokens are represented by each head. Note that heads consistently attending to "drink" (e.g. yellow and green) have similar representations across classes, while heads attending to the object of drinking (e.g. magenta and tan) have distinct representations for each class; further, the object-focused heads accept the verb as a stand-in for its implicit object when alcohol is not explicitly mentioned.

and *on*, which are similarly predictive of questions about prescribed medication (e.g. "Are you *on* any prescriptions?"), but which word senses are unlikely to converge representationally from pretraining on a general domain corpus. We take this as evidence of the BiGRU's ability to disambiguate word senses based on context, especially since we occasionally observe the same word types in different clusters within the same head. Finally, we observe some very broad concepts being captured by some attention heads that generalize across many classes, such as the notion of temporal existential quantifiers (*ever, before, experienced*).

6 Conclusion

In some sense, our analysis is unsurprising. Words having the same input representations should cluster together in model-internal representations, and members of the same class should similarly cluster. However, we have shown evidence that the self-attentive RNN does some amount of word sense disambiguation that generalizes across classes, and this behavior is driven only by semantic classification. From a human perspective, it makes sense that learning the most generalizable representation should be effective, but it's not clear that a model would need to learn those generalizations in order to perform the classification task. Clearly it benefits from doing so, so it seems the multi-headed self-attention at least allows for learning these generalizable concepts and the corresponding better optimum.

There are some interesting questions and open issues that should be addressed with future work. Additional experiments should do more to control for parameter counts; these should be matched for comparisons of the Bottleneck RNN to the full Self-attentive RNN, to more robustly characterize the effects of the additional nonlinearity in the bottleneck model. The Bottleneck representations also seem to reflect something like rudimentary "concepts," insofar as similar semantics often cluster together in the representation space. This raises the intriguing possibility that "metacognitive" processes could improve performance, for example with deductive or abductive inferences about relationships between representations across attention heads.

Overall, our analysis supports the claim that representations learned in frequent classes are transferring to, and improving performance on, rare classes, and further supports the value of a data set with a large number of subtly distinct classes.

References

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

Douglas Danforth, A. Price, K. Maicher, D. Post, B. Liston, D. Clinchot, C. Ledford, D. Way, and H. Cronau. 2013. Can virtual standardized patients be used to assess communication skills in medical students. In *Proceedings of the 17th Annual IAMSE Meeting, St. Andrews, Scotland.*

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Evan Jaffe, Michael White, William Schuler, Eric Fosler-Lussier, Alex Rosenfeld, and Douglas Danforth. 2015. Interpreting questions with a log-linear ranking model in a virtual patient dialogue system. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 86–96.

Lifeng Jin, David King, Amad Hussein, Michael White, and Douglas Danforth. 2018. Using paraphrasing and memory-augmented models to combat data sparsity in question interpretation with a virtual patient dialogue system. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 13–23.

Lifeng Jin, Michael White, Evan Jaffe, Laura Zimmerman, and Douglas Danforth. 2017. Combining cnns and pattern matching for question interpretation in a virtual patient dialogue system. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 11–21.

Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 1746–1751.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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*.

Abhijit Mahabal, Jason Baldridge, Burcu Karagol Ayan, Vincent Perot, and Dan Roth. 2019. Text classification with few examples using controlled generalization. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3158–3167.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML 2010)*, 3, pages 807–814.

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.

Adam Stiff. 2020. *Mitigation of Data Scarcity Issues for Semantic Classification in a Virtual Patient Dialogue Agent*. Ph.D. thesis, The Ohio State University, Columbus, Ohio, USA.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. Bert rediscovers the classical nlp pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601.

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 5998–6008.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, pages 3261–3275.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2015. Memory Networks. In *ICLR*, pages 1–15.

Bruce Wilcox. 2019. Chatscript. [Online; accessed 23-July-2019].

Matthew D. Zeiler. 2012. ADADELTA: An Adaptive Learning Rate Method. *CoRR*.

A Model Training

A.1 CNN Baseline

During cross-validation, we take ten percent of the data as test, and another ten percent for validation. We train on the remainder using Adadelta (Zeiler, 2012) for up to 25 epochs, and the model that produces the best validation accuracy is tested on the test set. Each training fold is augmented with the canonical questions for each class, so that no class is entirely unseen at test time. At test time, we take ten percent of the training data as a validation set, train on the other 90 percent, and use the same method of choosing which model to test. We use batch sizes of 50, and use GloVe.42B (Pennington et al., 2014) pretrained word vectors as input. We follow (Jin et al., 2017) for initialization and optimization parameters.

A.2 BERT Fine-tuning

We follow the recommended procedure for fine-tuning BERT to our task. We used the uncased base pretrained BERT model as input to a dense layer followed by a softmax for classification. All parameters were tuned jointly. The grid search optimized hyperparameters were a max sequence length of 16, a batch size of 2, 10 training epochs, and a learning rate of 2e-5.

A.3 CNN with Static BERT Embeddings

We expect that BERT model may be over-parameterized and under-trained for our relatively small data set. Thus, we collect non-contextual representations for the words in our dataset from the pretrained model. We then feed these as input to the baseline CNN model instead of the GloVe vectors.

We collect these static BERT embeddings by running the training set through the BERT model, and taking the state of the first layer from the BERT model as the embedding of the correspond token. We then average these representations for each word type in the data set, and use that as the input wherever the word occurs. Note that since BERT is trained with positional embeddings instead of ordering, representations from this layer likely retain a lot of positional information, which could be an important source of noise in the averaged representations. Training the CNN is otherwise the same as in the baseline experiment.

A.4 CNN with Contextual BERT Embeddings

Finally among our experiments with BERT, we feed the fully contextual representations into the baseline CNN architecture. Here, we again take the representation extracted from bottom layer of the BERT model.

A.5 RNN Training

The RNN with self-attention is trained using the same fold splits and canonical query augmentation as the CNN baseline. Here we use the Adam optimizer (Kingma and Ba, 2014) with default parameters. We initialize layer weights uniformly at random in the range $[-0.1, 0.1]$, and tokenize inputs using default SpaCy tokenization (Honnibal and Montani, 2017). We use GloVe.42B vectors again, and batch sizes of 32. We train for 40 epochs with an initial learning rate of 0.001, take the best model, reinitialize an optimizer with learning rate of 2.5×10^{-4}, and train for another 20 epochs, taking the best model of all 60 epochs to test.