NEURAL TEXT UNDERSTANDING WITH ATTENTION SUM READER

Rudolf Kadlec, Martin Schmid, Ondřej Bajgar & Jan Kleindienst
IBM Watson, V Parku 4, Prague 4, Czech Republic
{rudolf_kadlec,martin.schmid,obajgar,jankle}@cz.ibm.com

ABSTRACT

Two large-scale cloze-style context-question-answer datasets have been introduced recently: i) the CNN and Daily Mail news data and ii) the Children's Book Test. Thanks to the size of these datasets, the associated task is well suited for deep-learning techniques that seem to outperform all alternative approaches. We present a new, simple model that is tailor made for such question-answering problems. Our model directly sums attention over candidate answer words in the document instead of using it to compute weighted sum of word embeddings. Our model outperforms models previously proposed for these tasks by a large margin.

1 INTRODUCTION

Most of the information humanity has gathered up to this point is stored in the form of plain text. Hence the task of teaching machines how to understand this data is of utmost importance in the field of Artificial Intelligence. One way of testing the level of text understanding is simply to ask the system questions the answer to which can be inferred from the text. A well known example of a system that could make use of a huge collection of unstructured documents to answer questions is for instance IBM's Watson system used for the Jeopardy challenge (Ferrucci et al., 2010).

Cloze style questions, i.e. tasks of filling in a missing phrase into a sentence, are an appealing form of such questions. While the task is easy to evaluate, one can vary the context, the source of the question sentence or the missing phrase to dramatically change the task structure and difficulty.

Formal Task Description. The task consists of answering a cloze style question, the answer to which is dependent on the understanding of a context document provided with the question. The model is also provided with a set of possible answers from which the correct one is to be selected. Thus, the training data consist of tuples $(\mathbf{q}, \mathbf{d}, a, A)$ where $\mathbf{q}$ is a question, $\mathbf{d}$ is a document that contains an answer to the question $\mathbf{q}$, A is a set of possible answers and $a \in A$ is the ground truth answer. Both $\mathbf{q}$ and $\mathbf{d}$ are sequences of words from vocabulary V. We assume that all possible answers are words from the vocabulary and that the ground truth a appears in the document.

Datasets. Three large scale datasets have recently been introduced for training and evaluating models that fit the above task. The first two datasets (Hermann et al., 2015) were constructed from a large number of news articles from the CNN and Daily Mail websites. The main body of each article forms a context, while the cloze style question is formed using the short summary of the story. Specifically, the question is created by replacing a named entity from the summary sentence. (*e.g. "Producer X will not press charges against Jeremy Clarkson, his lawyer says."*). The third dataset, the Children's Book Test (CBT) (Hill et al., 2015), is built from books that are freely available thanks to Project Gutenberg. Each context is formed by 20 consecutive sentences taken from a children's book story. The cloze style question is constructed from the subsequent (21^{st}) sentence.

2 OUR MODEL — ATTENTION SUM READER

Our model called the *Attention Sum Reader (AS Reader)* is tailor-made to leverage the fact that the answer is a phrase from the context document. This is a double-edged sword. While it achieves state-of-the art results on all three datasets (for which this assumption holds), it cannot produce an answer which is not contained in the document. Intuitively, our model is structured as follows: i) we

compute a vector embedding of the query; ii) we compute a vector embedding of each individual word in the context; iii) using a similarity measure between the question embedding and the words in the context, we select the most likely candidate answer

2.1 FORMAL DESCRIPTION

AS Reader uses two encoder functions. The first is a document encoder function f that encodes every word from the document $\mathbf{d}$ in the context of the whole document (*contextual embedding*). We denote the contextual embedding of i-th word in $\mathbf{d}$ by $f_i(\mathbf{d})$. The second encoder g is used to translate the query $\mathbf{q}$ into a fixed length representation as $g(\mathbf{q})$. We use these vectors to compute dot product between all contextual embedding and the query embedding. This weight might be viewed as an attention over the document $\mathbf{d}$, and is normalized to form a probability distribution by applying the *softmax* function $s_i \propto \exp\left(f_i(\mathbf{d}) \cdot g(\mathbf{q})\right)$ Finally, we compute the probability of word w being the correct answer as $P(w|\mathbf{q}, \mathbf{d}) = \sum_{i \in I(w,\mathbf{d})} s_i$, where $I(w, \mathbf{d})$ is a set of positions where w appears in the document $\mathbf{d}$ (note that it is often the case that one word appears at multiple locations in the document). We call this mechanism *pointer sum attention* since we use attention as a pointer over discrete tokens in the context document and then we directly sum attention weights of the same words. This differs from usual use of attention in sequence-to-sequence models (Bahdanau et al., 2015) where attention is used to blend representations of words into one new embedding vector. Our use of attention was inspired by Pointer Networks (Vinyals et al., 2015). Both f and g share a single word embedding matrix.

In our model the document encoder f is implemented by a bidirectional Gated Recurrent Unit (GRU) network Cho et al. (2014) where hidden states are used as contextual word embeddings, that is $f_i(\mathbf{d}) = \overrightarrow{f_i}(\mathbf{d}) \mid\mid \overleftarrow{f_i}(\mathbf{d})$, where $\mid\mid$ denotes vector concatenation and $\overrightarrow{f_i}$ and $\overleftarrow{f_i}$ denote forward and backward contextual embeddings from respective recurrent networks. The query encoder g is implemented by another bidirectional GRU network. This time the last hidden state of the forward network is concatenated with the last hidden state of the backward network to form the query embedding, that is $g(\mathbf{q}) = \overrightarrow{g_{|\mathbf{q}|}}(\mathbf{q}) \mid\mid \overleftarrow{g_1}(\mathbf{q})$.

3 RELATED WORK

Attentive and Impatient Reader were proposed by Hermann et al. (2015), where the Attentive Reader model is very similar to our architecture. In contrast to the Attentive Reader, we select the answer from the context by directly using the computed attention rather than using such attention to blend the individual representations. While such blending allows the model to potentially produce an answer that is not included in the document, we hypothesize it actually harms the model if the answer is indeed always contained in the document. Adding blending into our implementation did indeed result into a significant drop in accuracy.

Memory Networks (Weston et al., 2014) were applied to the task of text comprehension by Hill et al. (2015). Due to the fact that the model uses a fixed (8) length windows centered around the candidate words, it's unable to capture longer dependencies. Furthermore, the representation for such window is computed simply as the sum of the words' embeddings inside that window. Finally, a heuristic approach called *self supervision* was necessary in order to get the model beyond 50% accuracy. In contrast our model i) can capture context from the entire document ii) the embedding computation via a recurrent network is more flexible compared to a simple sum iii) no heuristic supervision is necessary

4 EXPERIMENTAL RESULTS

Method. We trained our model using stochastic gradient descent with the ADAM update rule (Kingma & Ba, 2015) with learning rate 0.001 and gradient clipping threshold of 10. The embedding matrices were initialized randomly. Analogically to (Hermann et al., 2015), we shuffled assignments of named entities to corresponding word embedding vectors (for CNN and Daily Mail tasks only). We evaluated progress of training after each epoch and stopped when validation error started increasing (usually after two epochs). Our model was implemented using Theano (Bastien et al., 2012) and Blocks (van Merrienboer et al., 2015).

Results. We evaluated the proposed model both as a single model and using ensemble averaging. For single models, we are reporting results for the best model as well as the average of accuracies for the best 20% of models according to validation performance. We find that more informative since single models can display considerable variation of results which can then prove difficult to reproduce. As for the ensembles, we used simple averaging of the answer probabilities predicted by ensemble members. For the averaging ensemble we used the top 70% of all trained models (14, 16, 84 and 53 models for CNN, Daily Mail and CBT CN and NE respectively).

Table 1: Results of our AS Reader on CNN and Daily Mail datasets. Results for models marked with † are taken from (Hermann et al., 2015), results of models marked with ‡ are taken from (Hill et al., 2015). Performance of ‡ models was reported only on CNN dataset.

	CNN		Daily Mail	
	valid	test	valid	test
Attentive Reader †	61.6	63.0	70.5	69.0
Impatient Reader †	61.8	63.8	69.0	68.0
MemNNs (single model) ‡	63.4	66.8	NA	NA
MemNNs (ensemble) ‡	66.2	69.4	NA	NA
AS Reader (single model)	68.6	69.5	74.9	73.7
AS Reader (avg for top 20%)	68.4	69.9	74.5	73.5
AS Reader (avg ensemble)	**73.9**	**75.4**	**78.0**	**77.1**

Table 2: Results of our AS Reader on CBT datasets. Results marked with ‡are taken from (Hill et al., 2015). $^{(*)}$Human results were collected on 10% of the test set.

	Named entity		Common noun	
	valid	test	valid	test
Humans (context+query) $^{\ddagger(*)}$	NA	***81.6***	NA	***81.6***
MemNNs (window memory + self-sup.) ‡	70.4	66.6	64.2	63.0
AS Reader (single model)	73.8	68.6	68.8	63.4
AS Reader (avg for top 20%)	73.3	68.4	67.7	63.2
AS Reader (avg ensemble)	**74.6**	**70.6**	**71.2**	**69.0**

4.1 Discussion

CNN. Even our single best models outperform the previous best reported results (including ensembles). Furthermore, simple averaging ensemble performs 6% absolute better than the previous best reported ensemble (Hill et al., 2015).

Daily Mail. Our single model outperforms the previous best result achieved by Attentive Reader (Hermann et al., 2015) by 4.7% absolute and our averaging ensemble further improves this to 8.1% absolute improvement.

CBT. For named entity prediction our best single model performs 2% absolute better than the MemNN with self supervision, the averaging ensemble performs 4% absolute better than the best previous result. For common noun prediction, our single model accuracy is 0.4% absolute higher and the ensemble accuracy 6% absolute higher than the one of MemNN.

5 Conclusion

We presented a new neural network model for text comprehension which is simpler than previously proposed models (Hermann et al., 2015; Hill et al., 2015), however through a new efficient use of an attention mechanism it does perform significantly better on relevant datasets.

REFERENCES

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. *ICLR*, 2015. URL http://arxiv.org/abs/1409.0473v3.

Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian J. Goodfellow, Arnaud Bergeron, Nicolas Bouchard, and Yoshua Bengio. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *EMNLP*, jun 2014. URL http://arxiv.org/abs/1406.1078v3.

David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya a. Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, Nico Schlaefer, and Chris Welty. Building Watson: An Overview of the DeepQA Project. *AI Magazine*, 31(3):59–79, 2010. ISSN 0738-4602. doi: 10.1609/aimag.v31i3.2303. URL http://www.aaai.org/ojs/index.php/aimagazine/article/view/2303.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pp. 1684–1692, 2015.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. The goldilocks principle: Reading children's books with explicit memory representations. *arXiv preprint arXiv:1511.02301*, 2015.

Diederik P. Kingma and Jimmy Lei Ba. Adam: a Method for Stochastic Optimization. *International Conference on Learning Representations*, pp. 1–13, 2015.

Bart van Merrienboer, Dzmitry Bahdanau, Vincent Dumoulin, Dmitriy Serdyuk, David Warde-farley, Jan Chorowski, and Yoshua Bengio. Blocks and Fuel : Frameworks for deep learning. pp. 1–5, 2015.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Advances in Neural Information Processing Systems*, pp. 2674–2682, 2015.

Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2014.