Entity Tracking Improves Cloze-style Reading Comprehension

Luong Hoang **Sam Wiseman** **Alexander M. Rush**
School of Engineering and Applied Sciences
Harvard University
Cambridge, MA, USA
lhoanger@gmail.com, {swiseman,srush}@seas.harvard.edu

Abstract

Reading comprehension tasks test the ability of models to process long-term context and remember salient information. Recent work has shown that relatively simple neural methods such as the Attention Sum-Reader can perform well on these tasks; however, these systems still significantly trail human performance. Analysis suggests that many of the remaining hard instances are related to the inability to track entity-references throughout documents. This work focuses on these hard entity tracking cases with two extensions: (1) additional entity features, and (2) training with a multi-task tracking objective. We show that these simple modifications improve performance both independently and in combination, and we outperform the previous state of the art on the LAMBADA dataset, particularly on difficult entity examples.

1 Introduction

There has been tremendous interest over the past several years in Cloze-style (Taylor, 1953) reading comprehension tasks, datasets, and models (Hermann et al., 2015; Hill et al., 2016; Kadlec et al., 2016; Dhingra et al., 2016; Cui et al., 2016). Many of these systems apply neural models to learn to predict answers based on contextual matching, and have inspired other work in long-form generation and question answering. The extent and limits of these successes have also been a topic of interest (Chen et al., 2016; Chu et al., 2017). Recent analysis by Chu et al. (2017) suggests that a significant portion of the errors made by standard models, especially on the LAMBADA dataset (Paperno et al., 2016), derive from the inability to correctly track entities or speakers, or a failure to handle various forms of reference.

This work targets these shortcomings by designing a model and training scheme targeted towards *entity tracking*. Specifically we introduce two simple changes to a stripped down model: (1) simple, entity-focused features, and (2) two multi-task objectives that target entity tracking. Our ablation analysis shows that both independently improve entity tracking, which is the primary source of overall model's improvement. Together they lead to state-of-the-art performance on LAMBADA dataset and near state-of-the-art on CBT dataset (Hill et al., 2016), even with a relatively simple model.

> usually , that teen behavior was directed toward a parent , but since [julie] had somewhat filled the role of both parents for amy , [she] knew that **amy** was now directing that rebellion at [her] . `` i 'm sorry , marsh , " [**julie**] glanced at [her] friend out of the corner of [her] eye as [she] drove . `` do n't be . " `` do n't apologize for me , [**julie***]

Figure 1: A LAMBADA example where the final word "julie" (with reference chain in brackets) is the answer, y, to be predicted from the preceding context x. A system must know the two speakers and the current dialogue turn, simple context matching is not sufficient. Here, our model's predictions before and after adding multi-task objective are shown.

2 Background and Related Work

Cloze-style reading comprehension uses a passage of word tokens $x = x_{1:n}$ (the *context*), with one token x_j masked; the task is to fill in the masked word y, which was originally at position j. These datasets aim to present a benchmark challenge requiring some understanding of the context to select the correct word. This task is a prerequisite for problems like long-form generation and document-based question answering.

A number of datasets in this style exist with dif-

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 1049–1055
Brussels, Belgium, October 31 - November 4, 2018.

ferent focus. Here we considered the LAMBADA dataset and the named entity portion of the Children's Book Test dataset (CBT-NE). LAMBADA uses novels where examples consist of 4-5 sentences and the last word to be predicted is masked, x_n. The dataset is constructed carefully to focus on examples where humans needed the context to predict the masked word. CBT-NE examples, on the other hand, include 21 sentences where the masked word is a named entity extracted from the last sentence, with $j \leq n$, and is constructed in a more automated way. We show an example from LAMBADA in Figure 1. In CBT, as well as the similar CNN/Daily Mail dataset (Hermann et al., 2015), the answer y is always contained in x whereas in LAMBADA it may not be. Chu et al. (2017) showed, however, that training only on examples where y *is* in x leads to improved overall performance, and we adopt this approach as well.

Related Work The first popular neural network reading comprehension models were the Attentive Reader and its variant Impatient Reader (Hermann et al., 2015). Both were the first to use bidirectional LSTMs to encode the context paragraph and the query separately. The Stanford Reader (Chen et al., 2016) is a simpler version with fewer layers for inference. These models use an encoder to map each context token x_i to a vector $\boldsymbol{u}_i$. Following the terminology of Wang et al. (2017), *explicit reference models* calculate a similarity measure $s_i = s(\boldsymbol{u}_i, \boldsymbol{q})$ between each context vector $\boldsymbol{u}_i$ and a query vector $\boldsymbol{q}$ derived for the masked word. These similarity scores are projected to an attention distribution $\boldsymbol{\alpha} = \mathrm{softmax}(\{s_i\})$ over the context positions in $1, \ldots, n$, which are taken to be candidate answers.

The Attention Sum Reader (Kadlec et al., 2016) is a further simplified version. It computes $\boldsymbol{u}_i$ and $\boldsymbol{q}$ with separate bidirectional GRU (Chung et al., 2014) networks, and s_i with a dot-product. It is trained to minimize:

$$\begin{aligned}\mathcal{L}^0(\boldsymbol{\theta}) &= -\ln p(y \mid x, \boldsymbol{q}) \\ &= -\ln \sum_{i:x_i=y} p(x_i \mid \boldsymbol{q}) = -\ln \sum_{i:x_i=y} \alpha_i,\end{aligned}$$

where $\boldsymbol{\theta}$ is the set of all parameters associated with the model, and y is the correct answer. At test time, a *pointer sum attention* mechanism is used to predict the word type with the highest aggregate attention as the answer. The Gated Attention Reader (Dhingra et al., 2016) leverages the same mechanism for prediction and introduces an attention gate to modulate the joint context-query information over multiple hops.

The Recurrent Entity Networks (Henaff et al., 2016) uses a custom gated recurrent module, Dynamic Memory, to learn and update entity representations as new examples are received. Their gate function is combined of (1) a similarity measure between the input and the hidden states, and (2) a set of trainable "key" vectors which could learn any attribute of an entity such as its location or other entities it is interacting with in the current context. The Query Reduction Networks (Seo et al., 2016) is also a gated recurrent network which tracks state in a paragraph and uses a hidden query vector to keep pointing to the answer at each step. The query is successively transformed with each new sentence to a reduced state that's easier to answer given the new information.

Model In this work, we were particularly interested in the shortcomings of simple models and exploring whether or how much entity tracking could help, since Chu et al. (2017) has pointed out this weakness. As a result, we adapt a simplified Attention Sum (AttSum) reader throughout all experiments. Our version uses only a single bidirectional GRU for both $\boldsymbol{u}_i$ and $\boldsymbol{q}$. This GRU is of size $2d$, using the first d states for the context and second d for the query. Formally, let $\overrightarrow{\boldsymbol{h}}_i$ and $\overleftarrow{\boldsymbol{h}}_i$ (both in $\mathbb{R}^{2d}$) represent the forward and backward states of a bidirectional GRU run over x, and let $\overrightarrow{\boldsymbol{h}}_{i,\uparrow}$ and $\overrightarrow{\boldsymbol{h}}_{i,\downarrow}$ be the first and second d states respectively, and define $||$ as the concatenation operator. The context vectors are constructed as $\boldsymbol{u}_i = \overrightarrow{\boldsymbol{h}}_{i,\uparrow} || \overleftarrow{\boldsymbol{h}}_{i,\uparrow}$. For datasets using the last word, the query is constructed as $\boldsymbol{q} = \overrightarrow{\boldsymbol{h}}_{n,\downarrow} || \overleftarrow{\boldsymbol{h}}_{1,\downarrow}$. When the masked word can be anywhere, the query is constructed as $\boldsymbol{q} = \overrightarrow{\boldsymbol{h}}_{j-1,\downarrow} || \overleftarrow{\boldsymbol{h}}_{j+1,\downarrow}$.

Our main contribution is the extension of this simple model to incorporate entity tracking. Other authors have explored extending neural reading comprehension models with linguistic features, particularly Dhingra et al. (2017) who use a modified GRU with knowledge such as coreference relations and hypernymy. In Dhingra et al. (2018), the most recent coreferent antecedent for each token is incorporated into the update equations of the GRU unit to bias the reader towards coreferent recency. In this work, we instead use a much sim-

1	Sentence Index, POS Tag, NER Tag
2	Is among last 3 PERSON words in $\boldsymbol{x}$
3	Is a PERSON word in the last sentence
4	Is a PERSON word identical to previous PERSON word
5	Is a PERSON word identical to next PERSON word
6	Quoted-speech Index
7	Speaker

Table 1: Word-level features used in AttSum-Feat model.

pler set of features and compare to this and several other models as baseline approaches.

3 Learning to Track Entities

Analysis on reading comprehension has indicated that neural models are strong at matching local context information but weaker at following entities through the discourse (Chen et al., 2016; Chu et al., 2017). We consider two straightforward ways for extending the Attention Sum baseline to better track entities.

Method 1: Features We introduce a short-list of features in Table 1 to augment the representation of each word in x. These features are meant to help the system to identify and use the relationships between words in the passage.[1] Features 2-5 apply only to words tagged PERSON by the NER tagger. Features 6-7 apply only to words between opening and closing quotation marks. Feature 6 indicates the index of the quote in the document, and Feature 7 gives the assumed speaker of the quote using some simple rules; we provide the rules in the Supplementary Material. Though most of these features are novel, they are motivated by recent analysis (Wang et al., 2015; Chen et al., 2016; Wang et al., 2017).

All features are incorporated into a word's representation by embedding each discrete feature into a vector of the same size as the original word embedding, adding the vectors as well as a bias, and applying a $\tanh$ nonlinearity.

Method 2: Multitasking We additionally encourage the neural model to keep track of entities by multitasking with simple auxiliary entity-tracking tasks. Examples such as Figure 1 suggest that keeping track of which entities are currently in scope is useful for answering reading comprehension questions. There, *amy* and *julie* are conversing, and being able to track that *amy* is the speaker of the final quote helps to rule her out as a candidate answer. We consider two tasks:

For Task 1 ($\mathcal{L}^1$) we train the same model to predict repeated named entities. For all named entities x_j such that there is a $x_i = x_j$ with $i < j$, we attempt to mask and predict the word type x_j. This is done by introducing another Cloze prediction, but now setting the target $y = x_j$, reducing the context to preceding words $x_{1:j-1}$ with $\mathbf{u_i} = \overrightarrow{\boldsymbol{h}}_i$, and the query $\boldsymbol{q} = \overrightarrow{\boldsymbol{h}}_{j-1}$. (Note that unlike above, both of these only use the forward states of the GRU). We use a bilinear similarity score $s_i = \boldsymbol{q}^\mathsf{T} \boldsymbol{Q}\, \boldsymbol{u}_i$, for this prediction where $\boldsymbol{Q}$ is a learned transformation in $\mathbb{R}^{2d \times 2d}$. This task is inspired by the antecedent ranking task in coreference (Wiseman et al., 2015, 2016).

For Task 2 ($\mathcal{L}^2$) we train to predict the order index in which a named entity has been introduced. For example, in Figure 1, *julie* would be 1, *amy* would be 2, *marsh* would be 3, etc. The hope here is that learning to predict when entities reappear will help the model track their reoccurences. For the blue labeled *julie*, the model would aim to predict 1, even though it appears later in the context. This task is inspired by the One-Hot Pointer Reader of Wang et al. (2017) on the Who-did-What dataset (Onishi et al., 2016). Formally, letting $\widehat{\mathrm{idx}}(x_j)$ be the predicted index for x_j, we minimize:

$$\begin{aligned}\mathcal{L}^2(\boldsymbol{\theta}) &= -\ln p(\widehat{\mathrm{idx}}(x_j) = \mathrm{idx}(x_j) \mid x_{1:j-1}) \\ &= -\ln \mathrm{softmax}(\boldsymbol{W}\overrightarrow{\boldsymbol{h}}_j)_{\mathrm{idx}(x_j)},\end{aligned}$$

where $\boldsymbol{W} \in \mathbb{R}^{|\mathcal{E}| \times 2d}$ and $\mathcal{E}$ is the set of entity word types in the document. Note that this is a simpler computation, requiring only $O(|\mathcal{E}| \times n)$ predictions per $\boldsymbol{x}$, whereas $\mathcal{L}^1$ requires $O(n^2)$.

The full model minimizes a multi-task loss: $\mathcal{L}^0(\boldsymbol{\theta}) + \gamma_1 \mathcal{L}^1(\boldsymbol{\theta}) + \gamma_2 \mathcal{L}^2(\boldsymbol{\theta})$. Using $\mathcal{L}^1$ and $\mathcal{L}^2$ simultaneously did not lead to improved performance however, and so either γ_1, γ_2 is always 0. We believe that this is because, while the learning objectives for $\mathcal{L}^1$ and $\mathcal{L}^2$ are mathematically different, they are both designed to similarly track the entities mentioned so far in the document and thus do not provide complementary information to each other.

We found it useful to have two hyperparameters per auxiliary task governing the number of distinct

[1] POS tags are produced with the NLTK library (Bird et al., 2009), and NER tags with the Stanford NER tagger (Finkel et al., 2005). We additionally found it useful to tag animate words as PERSONs on the CBT-NE data, using the animate word list of Bergsma and Lin (2006).

named entity word *types* and *tokens* used in defining the losses $\mathcal{L}^1$ and $\mathcal{L}^2$. In particular, per document these hyperparameters control in a top-to-bottom order the number of distinct named entity word types we attempt to predict, as well as the number of tokens of each type considered.

4 Experiments

Methods This section highlights several aspects of our methodology; full hyperparameters are given in the Supplementary Material. For the training sets, we exclude examples where the answer is not in the context. The validation and test sets are not modified however and the model with the highest accuracy on the validation set is chosen for testing. For both tasks, the context words are mapped to learned embeddings; importantly, we initialize the first 100 dimensions with the 100-dimensional GLOVE embeddings (Pennington et al., 2014). Named entity words are anonymized, as is done in the CNN/Daily Mail corpus (Hermann et al., 2015) and in some of the experiments of Wang et al. (2017). The model is regularized with dropout (Srivastava et al., 2014) and optimized with ADAM (Kingma and Ba, 2014). For all experiments we performed a random search over hyperparameter values (Bergstra and Bengio, 2012), and report the results of the models that performed best on the validation set. Our implementation is available at `https://github.com/harvardnlp/readcomp`.

Results and Discussion Table 2 shows the full results of our best models on the LAMBADA and CBT-NE datasets, and compares them to recent, best-performing results in the literature.

For both tasks the inclusion of either entity features or multi-task objectives leads to large statistically significant increases in validation and test score, according to the McNemar test ($\alpha = 0.05$) with continuity correction (Dietterich, 1998). Without features, AttSum + $\mathcal{L}^2$ achieves the best test results, whereas with features AttSum-Feat + $\mathcal{L}^1$ performs best on CBT-NE. The results on LAMBADA indicate that entity tracking is a very important overlooked aspect of the task. Interestingly, with features included, AttSum-Feat + $\mathcal{L}^2$ appears to hurt test performance on LAMBADA and leaves CBT-NE performance essentially unchanged, amounting to a negative result for $\mathcal{L}^2$. On the other hand, the effect of AttSum-Feat + $\mathcal{L}^1$ is pronounced on CBT-NE, and while our simple models do not increase the state-of-the-art test performance on CBT-NE, they outperform "attention-over-attention" in addition to reranking (Cui et al., 2016), and is outperformed only by architectures supporting "multiple-hop" inference over the document (Dhingra et al., 2016). Our best model on CBT-NE test set, AttSum-Feat + $\mathcal{L}^1$, is very close to the current state-of-the-art result. On the validation sets for both LAMBADA and CBT-NE, the improvements from adding features to AttSum + $\mathcal{L}^i$ are statistically significant (for full results refer to our supplementary material). On LAMBADA, the $\mathcal{L}^1$ multi-tasked model is a 3.5-point increase on the state of the art.

Our method also employs fewer parameters than other richer models such as the GA Reader in (Dhingra et al., 2016). More specifically, in terms of number of parameters, our models are very similar to a 1-hop GA Reader. In contrast, all published experiments of the latter use 3 hops where each hop requires 2 separate Bi-GRUs, one

LAMBADA	Val	Test
GA Reader (Chu et al., 2017)	-	49.00
MAGE (48) (Dhingra et al., 2017)	51.10	51.60
MAGE (64) (Dhingra et al., 2017)	52.10	51.10
GA + C-GRU (Dhingra et al., 2018)	-	55.69
AttSum	56.03	55.60
AttSum + $\mathcal{L}^1$	58.35	56.86
AttSum + $\mathcal{L}^2$	58.08	57.29
AttSum-Feat	59.62	59.05
AttSum-Feat + $\mathcal{L}^1$	**60.22**	**59.23**
AttSum-Feat + $\mathcal{L}^2$	60.13	58.47
CBT-NE		
GA Reader (Dhingra et al., 2016)	78.50	**74.90**
EpiReader (Trischler et al., 2016)	75.30	69.70
DIM Reader (Liu et al., 2017)	77.10	72.20
AoA (Cui et al., 2016)	77.80	72.0
AoA + Reranker (Cui et al., 2016)	**79.60**	74.0
AttSum	74.35	69.96
AttSum + $\mathcal{L}^1$	76.20	72.16
AttSum + $\mathcal{L}^2$	76.80	72.60
AttSum-Feat	77.80	72.36
AttSum-Feat + $\mathcal{L}^1$	78.40	74.36
AttSum-Feat + $\mathcal{L}^2$	79.40	72.40

Table 2: Validation & Test results on all datasets. AttSum* are our models, including variants with features and multi-task loss. Others indicate previous best published results. All improvements over AttSum are statistically significant (α = 0.05) according to the McNemar test with continuity correction (Dietterich, 1998).

LAMBADA	All	Entity	Speaker	Quote
AttSum	56.03	75.17	74.81	73.31
AttSum + $\mathcal{L}^1$	58.35	78.51	78.38	79.42
AttSum + $\mathcal{L}^2$	58.08	78.17	77.96	76.76
AttSum-Feat	59.62	79.40	80.34	79.68
AttSum-Feat + $\mathcal{L}^1$	60.22	82.00	82.98	81.67
AttSum-Feat + $\mathcal{L}^2$	60.14	82.06	83.06	82.60
CBT-NE				
AttSum	74.35	76.28	75.08	74.96
AttSum + $\mathcal{L}^1$	76.20	78.03	76.98	77.33
AttSum + $\mathcal{L}^2$	76.80	77.45	76.27	76.48
AttSum-Feat	77.80	80.58	79.84	79.61
AttSum-Feat + $\mathcal{L}^1$	78.40	80.44	79.68	79.78
AttSum-Feat + $\mathcal{L}^2$	79.40	82.41	81.51	81.39

Table 3: Ablation results on validation sets, see text for definitions of the numeric columns and models.

after a few more minutes axel left patrick at the wall and went for a walk around the **village** . he stopped wherever he saw people and chatted . lifting spirits , cracking jokes , giving advice . the **vicar** was holding a service in the village square . a short and simple one . ` when you go out to war against your enemies , ' said the [vicar]

as the piece of rock fell off and was accelerating its way to the ground , he kicked it toward the robot and destroyed it upon contact . **lance** watched on the screen through the point of **view** of the last robot as the rock flew speedily toward it and then his screen cut into black . a soldier quickly walked in and went up to [lance]

Figure 2: LAMBADA examples where AttSum incorrectly predicts a non-entity answer whereas AttSum-Feat and AttSum + $\mathcal{L}^i$ choose correctly.

to model the document and one for the query. This constitutes the largest difference in model size between the two approaches.

Table 3 considers the performance of the different models based on a segmentation of the data. Here we consider examples where: (1) Entity - if the answer is a named entity; (2) Speaker - if the answer is a named entity and the speaker of quote; (3) Quote - if the answer is found within a quoted speech. Note that Speaker and Quote categories, while mutually exclusive, are subsets of the overall Entity category. We see that both the additional features and multi-task objectives independently result in a clear improvement in all categories, but that the gains are particularly pronounced for named entities and specifically for Speaker and Quote examples. Here we see sizable increases in performance, particularly in the Speaker category. We see larger increases in the more dialog heavy LAMBADA task.

As a qualitative example of the improvement afforded by multi-task training, in Figure 1 we show the different predictions made by our model with and without $\mathcal{L}^1$ (colored as blue and red, respectively). Note that *amy* and *julie* are both entities that have been repeated twice in the passage. In addition to the final answer, our model with the $\mathcal{L}^1$ loss was also able to predict these entities (at the colored locations) given preceding words. Further qualitative analysis reveals that these augmentations improved the model's ability to eliminate non-entity choices from predictions. Some examples are shown in Figure 2.

5 Conclusion

This work demonstrates that learning to track entities with features and multi-task learning significantly increases the performance of a baseline reading comprehension system, particularly on the difficult LAMBADA dataset. This result indicates that higher-level word relationships may not be modeled by simple neural systems, but can be incorporated with minor additional extensions. This work hints that it is difficult for vanilla models to learn long-distance entity relations, and that these may need to be encoded directly through features or possibly with better pre-trained representations.

Acknowledgments

SW gratefully acknowledges the support of a Siebel Scholars award. AMR gratefully acknowledges the support of NSF CCF-1704834, Intel Research, and Amazon AWS Research grants.

References

Shane Bergsma and Dekang Lin. 2006. Bootstrapping path-based pronoun resolution. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 33–40. Association for Computational Linguistics.

James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyz-*

ing text with the natural language toolkit. " O'Reilly Media, Inc.".

Danqi Chen, Jason Bolton, and Christopher D. Manning. 2016. A thorough examination of the cnn/daily mail reading comprehension task. In *ACL*.

Zewei Chu, Hai Wang, Kevin Gimpel, and David A. McAllester. 2017. Broad context language modeling as reading comprehension. In *EACL*, pages 52–57.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. 2016. Attention-over-attention neural networks for reading comprehension. *arXiv preprint arXiv:1607.04423*.

Bhuwan Dhingra, Qiao Jin, Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. 2018. Neural models for reasoning over multiple mentions using coreference. *arXiv preprint arXiv:1804.05922*.

Bhuwan Dhingra, Hanxiao Liu, Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. 2016. Gated-attention readers for text comprehension. *arXiv preprint arXiv:1606.01549*.

Bhuwan Dhingra, Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. 2017. Linguistic knowledge as memory for recurrent neural networks. *arXiv preprint arXiv:1703.02620*.

Thomas G Dietterich. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, 10(7):1895–1923.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 363–370. Association for Computational Linguistics.

Mikael Henaff, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann LeCun. 2016. Tracking the world state with recurrent entity networks. *arXiv preprint arXiv:1612.03969*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2016. The goldilocks principle: Reading children's books with explicit memory representations. In *ICLR*.

Rudolf Kadlec, Martin Schmid, Ondrej Bajgar, and Jan Kleindienst. 2016. Text understanding with the attention sum reader network. *arXiv preprint arXiv:1603.01547*.

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

Zhuang Liu, Degen Huang, Kaiyu Huang, and Jing Zhang. 2017. *DIM Reader: Dual Interaction Model for Machine Comprehension*. Springer International Publishing, Cham.

Takeshi Onishi, Hai Wang, Mohit Bansal, Kevin Gimpel, and David McAllester. 2016. Who did what: A large-scale person-centered cloze dataset. *arXiv preprint arXiv:1608.05457*.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *ACL*.

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.

Minjoon Seo, Sewon Min, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Query-reduction networks for question answering. *arXiv preprint arXiv:1606.04582*.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958.

Wilson L Taylor. 1953. cloze procedure: a new tool for measuring readability. *Journalism Bulletin*, 30(4):415–433.

Adam Trischler, Zheng Ye, Xingdi Yuan, and Kaheer Suleman. 2016. Natural language comprehension with the epireader. *arXiv preprint arXiv:1606.02270*.

Hai Wang, Mohit Bansal, Kevin Gimpel, and David McAllester. 2015. Machine comprehension with syntax, frames, and semantics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 700–706.

Hai Wang, Takeshi Onishi, Kevin Gimpel, and David McAllester. 2017. Emergent predication structure in hidden state vectors of neural readers. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 26–36.

Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. 2016. Learning global features for coreference resolution. In *NAACL HLT*, pages 994–1004.

Sam Wiseman, Alexander M. Rush, Stuart M. Shieber, and Jason Weston. 2015. Learning anaphoricity and antecedent ranking features for coreference resolution. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1416–1426.