Neural Skill Transfer from Supervised Language Tasks to Reading Comprehension

Todor Mihaylov [1,3], Zornitsa Kozareva[2], and Anette Frank[1,3]

[1]Institute for Computational Linguistics, Heidelberg University, Germany
[2]Amazon, AWS Deep Learning, Palo Alto, CA
[3]Research Training Group AIPHES
{mihaylov,frank}@cl.uni-heidelberg.de, zornitsa@kozareva.com

Abstract

Reading comprehension is a challenging task in natural language processing and requires a set of skills to be solved. While current approaches focus on solving the task as a whole, in this paper, we propose to use a neural network 'skill' transfer approach. We transfer knowledge from several lower-level language tasks (skills) including textual entailment, named entity recognition, paraphrase detection and question type classification into the reading comprehension model. We conduct an empirical evaluation and show that transferring language skill knowledge leads to significant improvements for the task with much fewer steps compared to the baseline model. We also show that the skill transfer approach is effective even with small amounts of training data. Another finding of this work is that using token-wise deep label supervision for text classification improves the performance of transfer learning.

1 Introduction

Reading comprehension (RC) is a language understanding task, typically evaluated in a question answering setting, where a system is expected to read a given passage of text (document D) and answer questions (Q) about it. Recent work has introduced several datasets for reading comprehension which gained a lot of attention such as the 'CNN/Daily Mail' [13], Children Book Test [14], Who Did What [33], bAbI [50] and before that MCTest [39]. Most recently SQuAD [38], NewsQA [47] and TriviaQA[17] were created using crowd-sourcing. Reading comprehension has been shown [45, 4, 38] to require different sets of skills such as paraphrase detection, recognition of named entities, natural language inference, etc. The common approach to tackling a higher-level task such as Reading Comprehension is to build a complex neural model that reads a large-scale dataset and tries to learn all required skills at once.

We propose learning the 'skills' required for the task of reading comprehension from existing supervised language tasks. We evaluate the performance of several learned lower-level 'skills' for reading comprehension on the SQuAD [38] dataset by integrating them in a simple neural model. This is in contrast to [8] who propose learning sentence compression representations from a large supervised corpus and transfer the learned knowledge to a set of smaller tasks. Our approach is similar to [27] who used weights pre-trained on machine translation to boost the performance of a very good RC system [52]. Instead of solving a single complex task, we propose using the knowledge learned from multiple supervised, possibly low-scale, language tasks as 'skills'. We propose a simple model that allows to inject learned 'skill' representations easily and analyze the learning behavior of this skill transfer model for reading comprehension. We also experiment with training on smaller parts of the training data (2%, 5%, 10%) to examine the impact of 'skill' transfer on smaller datasets.

2 Method

In this work, we tackle the task of reading comprehension using lower-level supporting 'skill' tasks. To do that, we implement a baseline model to represent the relation between a given question and the story context and enrich the representation by reusing encoder weights from the chosen 'skill' tasks.

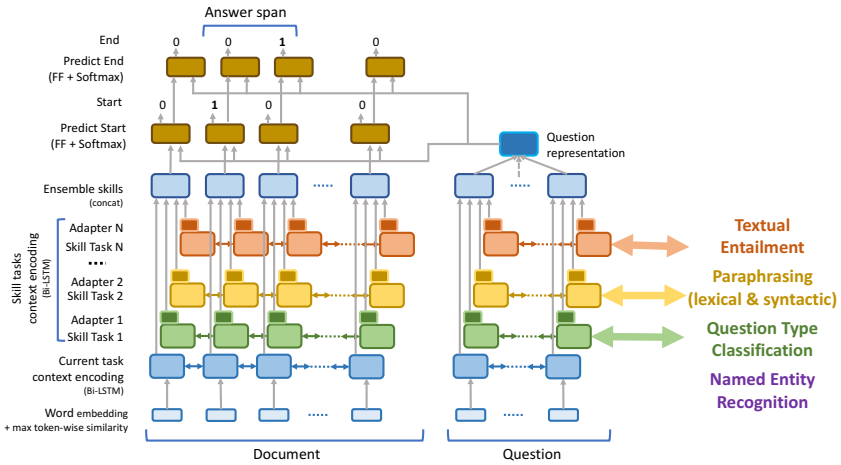

Figure 1: Skillful Reader: Architecture for transferring knowledge from 'skill' language tasks to a reading comprehension model.

Our 'skill' transfer method is visualized in Figure 1 and can be summarized in two main steps:

- Skill Learning: Train context encoder-based (Bi-LSTM) models for several language skill tasks and save the learned encoder weights.
- Neural Skill Transfer: Reuse the learned context encoder skill weights to encode the text context of document and question, in a simple model for the higher-level task (QA/RC).

An overview of our model is shown in Figure 1. It can be considered similar to *progressive neural networks* [42] without the notion of sequential learning of the tasks. We refer to the underlying tasks as skills, following [45], who show that complex tasks like RC require a set of language analysis skills. We show that using such skills, learned from specialized corpora, boosts the performance of a good baseline RC system (i) early in training and (ii) when training on smaller portions (2, 5, and 10 percent) of the original training data.

2.1 Skill Learning

For encoding the skill knowledge from lower-level tasks we first implement simple context encoder models for each low-level task. In this work we implement three types of models for encoding language skill tasks: Sequence Labeling, Text Classification, and Relation Classification.

Sequence Labeling is applied for labeling each token of a given text with a specific category. For this type of encoder model we use a vanilla Bi-directional Long Short-Term Memory [10] architecture, that uses word embeddings as input with a label projection layer with Softmax to predict the sequence labels (2a). While this does not lead to a supreme performance in any sequence-labeling task, it is a stable baseline [25, 21]. We hypothesize that by using a simple architecture for the skill model, we can encode the skill knowledge in the context layer. As a sequence labeling skill, we choose the task of Named Entity Recognition (NER) based on the CoNLL 2012 NER dataset [37]. We use the BIO schema for label encoding, as shown in Figure 2a.

Text Classification is applied in order to categorize a given word token sequence. Given that our RC task is cast as a QA problem, we propose to employ the skill of Question Type Classification, using the TREC Question Classification dataset [23] with 50 classes for training. The task is to classify a given question according to the type of the answer phrase. To learn text classification skills we employ a simple model with Bi-LSTM context encoder, where we apply label supervision on the **token** level. The model is shown in Fig. 2b). That is, instead of retrieving a single vector representation of the

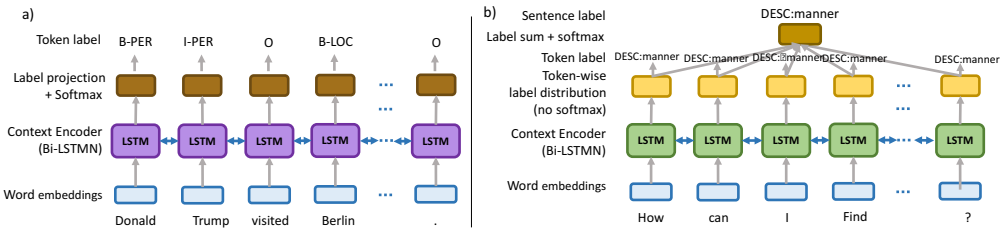

Figure 2: a) Vanilla Bi-LSTM for sequence labeling (NER). b) Text classification (Question Type Classification) with Bi-LSTM context encoder and token-wise label supervision.

sentence (with avg- or max-pooling, etc.) and predicting the label, we project the token context representation $c_{t_{1..n}}$ to the label space (50 classes) $c^{lbl}_{t_{1..n}}$ and sum the label representation predicted for each token, to obtain the label for the sentence $r^{lbl}_{sent} = softmax(\sum c^{lbl}_{t_{1..n}})$. We hypothesize that with lower-level label supervision we can propagate the knowledge expressed by the label to the context representations of specific tokens. This is a form of deep supervision [22], similar to [24].

Relation Classification is used to classify the relation between two arguments represented as text. We implement relation classification skills following the exact *Bi-LSTM max-out* model from Conneau et al. [8], that has been shown to be successful for learning sentence representations.

As a relation classification skill we employ the tasks of Textual Entailment (TE) learned from the Stanford Natural Language Inference (SNLI) corpus [3]. TE is a task that requires a model to classify the entailment relation between two sentences: hypothesis and premise. For instance, the premise *'Dogs like eating food.'* entails the hypothesis *'Animals like eating.'*. Another task that we consider useful for our target task is paraphrase detection over the PPDB 2.0 [35] where the model is required to detect the relation between two phrases in one of the given 6 fine-grained paraphrase classes.

2.2 Model for Reading Comprehension with Skills

We build a simple neural model that uses pre-trained embeddings and word-matching features as input to a bi-directional LSTM context-encoder of document and question and two Bi-LSTM layers for predicting start and end of the answer span. The architecture of the model is shown in Figure 1.

Word embedding input. As an input to the neural model, we use pre-trained 100d Glove [36] word embeddings (WE). We also use two features for each token: the exact word matching feature (em) [49] [6] between each token in the document and the question and the maximum similarity between the word embedding vector of each of the document tokens and each token in the question ($maxsim(w_{d_i}, w_{q_{1..m}}) = \max(\cos(w_{d_i}, w_{q_{1..m}}))$). The WE *maxsim* between two texts has been shown to be helpful for community question answering [5] and discourse relation sense classification [28]. For each token we concatenate the WE and the two features ($w^r_{p_{1..N}} = concat(w^p_{e_i}, maxsim, em)$, *r* means input representation, *p* is a token sequence that can be *d*(document) or *q*(question)) and use them as an input to the context-encoder. For the question, the two features above are set to 1 as in [49].

Context encoding. In particular, we use a Bi-LSTM context encoder represented as $c_{p_{1..N}} = BiLSTM(w^r_{p1..N})$. We refer to a task-specific context-encoder as Enc_{task}.

Context encoder for the current (main) task. For the target task of reading comprehension, we initialize an encoder Enc_{RC} with random weights.

Skill task context encoders. For each skill task, we train a context-encoder model as described in Sec. 2.1. We use the learned weights to initialize the task-specific encoders Enc_{skill}. For the tasks where we employ token label prediction (NER and Question Type Classification), we also concatenate the soft label prediction vectors with the context encoder states: $Enc_{NER/QTC} = concat(c_{p_{1..N}}, c^{lbl}_{p_{1..N}})$.

Adapted representations. Each output from the skill context encoder is projected to a lower dimension using adapter weights [42]: $c^{1..n}_{task} = Enc_{task}(w_{1..n})A_{task} + b^a_{task}$, where A_{task} is a weight matrix for the current task (skill task or target task (RC)) and b^a_{task} is a bias vector.

Ensemble representation. For each token in the document d and question q we concatenate all adapted skill representations c_{task} to the main task representation c_{rc} to obtain the ensemble repre-

sentation $e_p = concat(c_{rc}, c_{ner}, c_{qtc}, c_{te}, c_{ppdb})$, where p is d or q. We represent the question by a weighted representation of its ensemble token vectors: $r_q = sum(e_{q_{1..m}} * softmax(e_{q_{1..m}} W_{qw}))$, where W_{qw} is a weight matrix. We then model interaction between the question representation r_q and each document token e_{d_i} as $r_{d_i 2q} = concat(e_{d_i}, r_q, e_{d_i} * r_q)$.

Answer span prediction. To predict the answer span we predict start and end pointers in the document context. We model the probability of the document tokens being the start of the answer span as $ans_i^{start} = softmax(W_{start} FF(r_{d_i 2q}) + b_{start})$, where W_{start} is a weight matrix and b_{start} is bias. We then model the probability of the document tokens being the end of the answer span as $ans_i^{end} = softmax(W_{end} FF(concat(r_{d_i 2q}, ans_i^{start}, ans_i^{start} * e_{d_i})) + b_{end})$, where W_{end} is a weight matrix and b_{end} is a bias vector. We use dynamic programming to find the answer span (i,j) that maximizes $ans_i^{start} * ans_j^{end}$. FF is a 2-layer (size 100) feed-forward network with $ReLU$[31].

Training details. For all skill tasks and the RC task we use pre-trained Glove word embeddings with size 100. For all tasks, including the target RC task, we train the bi-directional LSTM encoder with output size 256. For the adaption layer we use output size of 100 for the skills and 128 for RC.

3 Related work

Reading comprehension [15] has gained a lot of attention in the last years thanks to large-scale datasets [13][14][33]. More recently the SQuAD [38] dataset offered over 100 thousand crowd-sourced questions to answer questions about Wikipedia. Some of the best performing single models (F1 75-84) on the SQuAD dataset propose token-wise interaction between documents and question Bi-DAF [43], Dynamic-Coattention Networks [52], R-NET [48]. Some models [44][30][46] try to perform reasoning more explicitly using an approach based on memory networks [51, 11]. Some simple neural models [6][49][9] incorporate features to achieve better performance. It has been shown that a big enough dataset [1] can provide enough knowledge to allow a simple neural model [19] to achieve human performance. However, in practice, having a huge dataset is not always an option. So another approach can be to transfer knowledge [18] from another dataset of the same task or from a less related task such as machine translation [27]. Indeed almost all recent neural models use a form of transfer learning by incorporating word embeddings, such as [29][36], as input. Some recent models [34] even use the task of question answering to learn better embeddings. Transfer Learning with neural models has been proposed in NLP initially by [7] and has been encouraged as a way of sharing representations between tasks [2]. It can be performed jointly on multiple tasks [40] which includes learning linguistic tasks in a hierarchical fashion [41] on many levels [12] or even perform the knowledge transfer between tasks from different modalities [20]. In this work we propose a generic and modular approach to learning a set of relevant 'skill' tasks and transferring this knowledge to a target task, here the problem of reading comprehension.

4 Experiments and results

In this work we examine the impact of transferring knowledge from several 'skill' tasks to the task of Reading Comprehension. The assumption is that the transfer of skill knowledge should improve the learning of the target task (RC) and allows for using smaller training sets and fewer training steps. To examine this impact we run several experiments: adding single skill tasks to the RC task, adding all tasks, and ablation of tasks.

Training on the full training set. We use the SQuAD [38] train dataset for training and the publicly available dev set for evaluation. We do not aim for state-of-the art performance but focus on the impact of injecting skill knowledge. In Table 1 we show evaluation results with single tasks and ablation of tasks, w/ and w/o fine tuning of the skill parameters. Figure 3 shows the results in different training steps, with different skills. It shows that individual skills and all skills jointly show a noticeable impact in the early training stages compared to a model without skills.

Training on small parts of the train data. Figure 5 shows results with training on different sizes of the train data. 2% of the train data contains 378 paragraphs, 2512 questions, with 88k tokens in total. We show that with less data (2%, 5% of the full train set), employing skill tasks shows high impact, reaching the best result compared to 'No skills' or 'Random skill weights' setups in only 1000 steps.

Token-wise label supervision. Figure 4 analyzes the impact of token-wise label prediction vs. sentence-wise label prediction with Question Type Classification. We show that token supervision clearly outperforms sentence label supervision in early training phases.

Setup	Skill fine-tuning F-score	EM	No skill fine-tuning F-Score	EM
RC only (no skills)	59.41	46.90	**59.66**	**46.80**
+ only TE	**61.67**	**49.12**	59.40	46.47
+ only QC	60.94	48.68	57.80	44.39
+ only PPDB	60.82	48.71	58.23	45.25
+ only NER	60.65	48.45	58.17	44.70
all (RC + all skills)	60.92	48.70	58.30	45.51
all - PPDB	61.11	48.83	57.87	45.19
all - QC	60.91	48.52	57.28	45.17
all - TE	60.86	48.81	58.48	45.73
all - NER	60.81	48.55	56.99	44.07

Table 1: Results for transferring knowledge from skill tasks. 'Fine-tuning': parameters of the skill tasks are fine-tuned during training.'EM' shows the results for Exact Match with the gold answers. We show evaluation results on the dev set of SQuAD [38]. Trained with 51k steps on the train set.

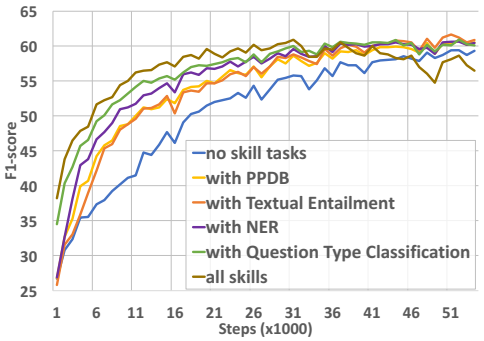

Figure 3: Results for single skill tasks combined with the QA-encoder. (w/ skill fine-tuning)

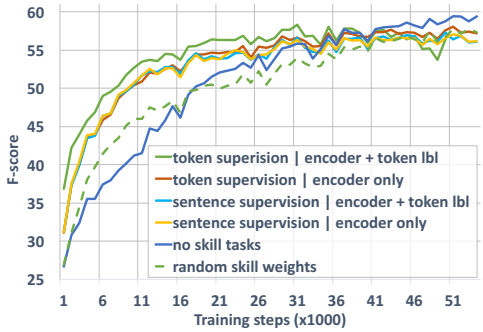

Figure 4: Results with QTC skill, w/ and w/o token label supervision. (w/o fine-tuning)

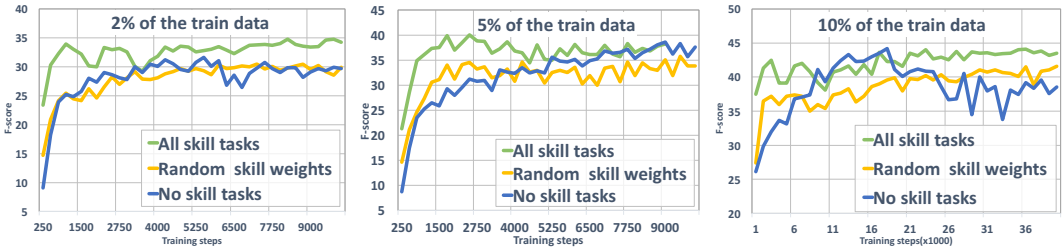

Figure 5: Results for training with different sizes of the training data (2%, 5%, 10%) and evaluated on the dev set. 'Rand. skill weights' is 'All skill tasks' model with random weights. (w/ fine-tuning)

5 Conclusion and future work

In this work, we show the impact of injecting knowledge from supervised language skill tasks into a reading comprehension model. We observe noticeable gains of performance in both, early training stages and when using small training data. While for some domains, currently large training sets are being built, in others such as [39] this is not the case. Beyond performance issues, using skill tasks as proposed in this work can be applied as a tool for analyzing which specific skills are required for reading comprehension (or other tasks) and also the contribution of specific skills for a particular dataset and problem formulation, without having to conduct manual annotation as in [45]. Another finding is that token-wise deep label supervision for QTC is beneficial for reading comprehension in a QA setting. In future work we plan to transfer knowledge from other tasks i.a. Discourse Relations [16] [32], Semantic Role Labeling [26]. We also want to experiment with different models of integrating the learned skills, also for other tasks. We also plan to train all the tasks jointly, in multi-task fashion, where shared parameters are fine-tuned on the skill tasks and the target task.

Acknowledgments

Most of this work is performed during Todor Mihaylov's internship at Amazon, AWS Deep Learning. This work is partly supported by the German Research Foundation as part of the Research Training Group "Adaptive Preparation of Information from Heterogeneous Sources" (AIPHES) under grant No. GRK 1994/1.

References

[1] Ondrej Bajgar, Rudolf Kadlec, and Jan Kleindienst. Embracing data abundance: BookTest Dataset for Reading Comprehension. 2016. URL http://arxiv.org/abs/1610.00956.

[2] Yoshua Bengio. Deep Learning of Representations for Unsupervised and Transfer Learning. *JMLR: Workshop and Conference Proceedings 7*, 7:1–20, 2011.

[3] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. pages 632–642, 2015. doi: 10.18653/v1/D15-1075. URL http://www.aclweb.org/anthology/D15-1075.

[4] Danqi Chen, Jason Bolton, and Christopher D. Manning. A thorough examination of the CNN/Daily Mail reading comprehension task. 2016.

[5] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-1171. URL http://www.aclweb.org/anthology/P17-1171.

[6] Fisch Adam Chen, Danqi, Jason Weston, and Antoine Bordes. Reading Wikipedia to Answer Open-Domain Questions. 2016.

[7] Ronan Collobert and Jason Weston. A unified architecture for natural language processing. *Proceedings of the 25th international conference on Machine learning - ICML '08*, 20(1):160–167, 2008. ISSN 07224028. doi: 10.1145/1390156.1390177. URL http://portal.acm.org/citation.cfm?id=1390177{%}5Cnhttp://portal.acm.org/citation.cfm?doid=1390156.1390177.

[8] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. may 2017. URL http://arxiv.org/abs/1705.02364.

[9] Bhuwan Dhingra, Hanxiao Liu, Zhilin Yang, William Cohen, and Ruslan Salakhutdinov. Gated-attention readers for text comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1832–1846. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-1168. URL http://www.aclweb.org/anthology/P17-1168.

[10] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610, 2005.

[11] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural Turing Machines. *Arxiv*, 2014. URL http://arxiv.org/abs/1410.5401.

[12] Kazuma Hashimoto, caiming xiong, Yoshimasa Tsuruoka, and Richard Socher. A joint many-task model: Growing a neural network for multiple nlp tasks. pages 446–456, 2017. URL http://www.aclweb.org/anthology/D17-1046.

[13] Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. 2015. URL http://arxiv.org/abs/1506.03340.

[14] Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. The Goldilocks Principle: Reading Children's Books with Explicit Memory Representations. *International Conference on Learning Representations*, 2016. URL http://arxiv.org/abs/1511.02301.

[15] Lynette Hirschman, Marc Light, Eric Breck, and John D. Burger. Deep read: A reading comprehension system. 1999. URL http://www.aclweb.org/anthology/P99-1042.

[16] Yacine Jernite, Samuel R Bowman, and David Sontag. Discourse-Based Objectives for Fast Unsupervised Sentence Representation Learning. URL https://arxiv.org/pdf/1705.00557.pdf.

[17] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics (ACL) 2017. URL http://aclweb.org/anthology/P17-1147.

[18] Rudolf Kadlec, Ondrej Bajgar, and Jan Kleindienst.

[19] Rudolf Kadlec, Martin Schmid, Ondřej Bajgar, and Jan Kleindienst. Text understanding with the attention sum reader network. pages 908–918, 2016. doi: 10.18653/v1/P16-1086. URL http://www.aclweb.org/anthology/P16-1086.

[20] Lukasz Kaiser, Aidan N. Gomez, Noam Shazeer, Ashish Vaswani, Niki Parmar, Llion Jones, and Jakob Uszkoreit. One model to learn them all. *CoRR*, abs/1706.05137, 2017. URL http://arxiv.org/abs/1706.05137.

[21] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. pages 260–270, 2016. doi: 10.18653/v1/N16-1030. URL http://www.aclweb.org/anthology/N16-1030.

[22] Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. pages 562–570, 2015.

[23] Xin Li and Dan Roth. Learning question classifiers. 2002. URL http://www.aclweb.org/anthology/C02-1150.

[24] Zachary C Lipton, David C Kale, Charles Elkan, and Randall Wetzell. Learning to diagnose with lstm recurrent neural networks. URL https://arxiv.org/pdf/1511.03677.pdf.

[25] Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. pages 1064–1074, 2016. doi: 10.18653/v1/P16-1101. URL http://www.aclweb.org/anthology/P16-1101.

[26] Ana Marasović and Anette Frank. SRL4ORL: Improving Opinion Role Labelling using Multi-task Learning with Semantic Role Labeling. nov 2017. URL http://arxiv.org/abs/1711.00768.

[27] Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. Learned in translation: Contextualized word vectors. *CoRR*, abs/1708.00107, 2017. URL http://arxiv.org/abs/1708.00107.

[28] Todor Mihaylov and Anette Frank. Discourse relation sense classification using cross-argument semantic similarity based on word embeddings. In *Proceedings of the CoNLL-16 shared task*, pages 100–107. Association for Computational Linguistics, 2016. doi: 10.18653/v1/K16-2014. URL http://www.aclweb.org/anthology/K16-2014.

[29] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '13, pages 746–751, Atlanta, Georgia, USA, 2013. URL http://www.aclweb.org/anthology/N13-1090.

[30] Tsendsuren Munkhdalai and Hong Yu. Reasoning with Memory Augmented Neural Networks for Language Comprehension. *International Conference on Learning Representations (ICLR) 2017*, pages 1–13, oct 2016. URL http://arxiv.org/abs/1610.06454.

[31] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In Johannes Fürnkranz and Thorsten Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814. Omnipress, 2010. URL http://www.icml2010.org/papers/432.pdf.

[32] Allen Nie, Erin D. Bennett, and Noah D. Goodman. Dissent: Sentence representation learning from explicit discourse relations. *CoRR*, abs/1710.04334, 2017. URL http://arxiv.org/abs/1710.04334.

[33] Takeshi Onishi, Hai Wang, Mohit Bansal, Kevin Gimpel, and David McAllester. Who did What: A Large-Scale Person-Centered Cloze Dataset. *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2016*, pages 2230–2235, 2016. URL http://arxiv.org/abs/1608.05457.

[34] Boyuan Pan, Hao Li, Zhou Zhao, Bin Cao, Deng Cai, and Xiaofei He. MEMEN: Multi-layer Embedding with Memory Networks for Machine Comprehension. 2017.

[35] Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. Ppdb 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. pages 425–430, 2015. doi: 10.3115/v1/P15-2070. URL http://www.aclweb.org/anthology/P15-2070.

[36] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. pages 1532–1543, 2014. doi: 10.3115/v1/D14-1162. URL http://www.aclweb.org/anthology/D14-1162.

[37] Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. pages 1–40. Association for Computational Linguistics, 2012. URL http://www.aclweb.org/anthology/W12-4501.

[38] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ Questions for Machine Comprehension of Text. *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2016*, 2016. URL http://arxiv.org/abs/1606.05250.

[39] Matthew Richardson, Christopher J C Burges, and Erin Renshaw. MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text. *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2013*, pages 193–203, 2013.

[40] Sebastian Ruder. An Overview of Multi-Task Learning in Deep Neural Networks. (May), 2017.

[41] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *CoRR*, abs/1706.05098, 2017. URL http://arxiv.org/abs/1706.05098.

[42] Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive Neural Networks. 2016. URL http://arxiv.org/abs/1606.04671.

[43] Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *CoRR*, abs/1611.01603, 2016. URL http://arxiv.org/abs/1611.01603.

[44] Yelong Shen, Po-Sen Huang, Jianfeng Gao, and Weizhu Chen. Reasonet: Learning to stop reading in machine comprehension. *CoRR*, abs/1609.05284, 2016. URL http://arxiv.org/abs/1609.05284.

[45] Saku Sugawara, Hikaru Yokono, and Akiko Aizawa. Prerequisite skills for reading comprehension: Multi-perspective analysis of mctest datasets and systems. pages 3089–3096, 2017.

[46] Sainbayar Sukhbaatar, arthur szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. pages 2440–2448, 2015. URL http://papers.nips.cc/paper/5846-end-to-end-memory-networks.pdf.

[47] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset. *CoRR*, abs/1611.09830, 2016. URL http://arxiv.org/abs/1611.09830.

[48] Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. Gated self-matching networks for reading comprehension and question answering. pages 189–198, 2017. doi: 10.18653/v1/P17-1018. URL http://www.aclweb.org/anthology/P17-1018.

[49] Dirk Weissenborn, Georg Wiese, and Laura Seiffe. Making neural qa as simple as possible but not simpler. pages 271–280, 2017. doi: 10.18653/v1/K17-1028. URL http://www.aclweb.org/anthology/K17-1028.

[50] Jason Weston, Antoine Bordes, Sumit Chopra, Tomas Mikolov, and Alexander M. Rush. Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks. *arXiv Prepr.*, 2015. ISSN 1502.05698. doi: 10.1016/j.jpowsour.2014.09.131.

[51] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *International Conference on Learning Representations (ICLR), 2015*, 2015.

[52] Caiming Xiong, Victor Zhong, and Richard Socher. Dynamic coattention networks for question answering. *CoRR*, abs/1611.01604, 2016. URL http://arxiv.org/abs/1611.01604.