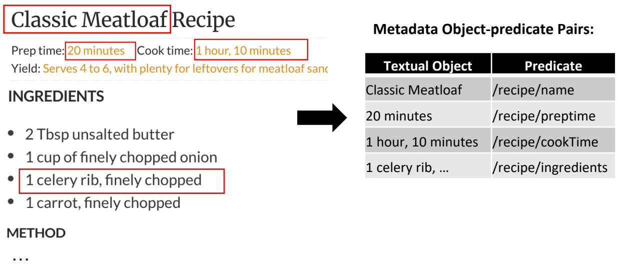
Leveraging Structured Metadata for Improving Question Answering on the Web

Xinya Du[1]* **Adam Fourney**[2] **Robert Sim**[2]
Claire Cardie[1] **Paul N. Bennett**[2] **Ahmed Hassan Awadallah**[2]
[1]Department of Computer Science, Cornell University, Ithaca, NY
{xdu, cardie}@cs.cornell.edu
[2]Microsoft Research, Redmond, WA
{adamfo, rsim, paul.n.bennett, hassanam}@microsoft.com

Abstract

We show that leveraging metadata information from web pages can improve the performance of models for answer passage selection/reranking. We propose a neural passage selection model that leverages metadata information with a fine-grained encoding strategy, which learns the representation for metadata predicates in a hierarchical way. The models are evaluated on the MS MARCO (Nguyen et al., 2016) and Recipe-MARCO datasets. Results show that our models significantly outperform baseline models, which do not incorporate metadata. We also show that the fine-grained encoding's advantage over other strategies for encoding the metadata.



Classic Meatloaf Recipe

Prep time: 20 minutes Cook time: 1 hour, 10 minutes
Yield: Serves 4 to 6, with plenty for leftovers for meatloaf sanc

INGREDIENTS

- 2 Tbsp unsalted butter
- 1 cup of finely chopped onion
- 1 celery rib, finely chopped
- 1 carrot, finely chopped

METHOD

...

Metadata Object-predicate Pairs:

Textual Object	Predicate
Classic Meatloaf	/recipe/name
20 minutes	/recipe/preptime
1 hour, 10 minutes	/recipe/cookTime
1 celery rib, ...	/recipe/ingredients

Figure 1: Metadata Example from SimplyRecipes.

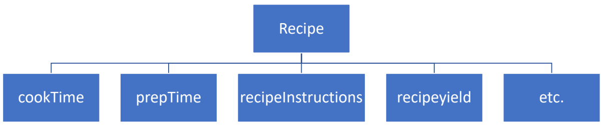


Figure 2: Hierarchy diagram showing properties of "recipe" from schema.org/recipe.

1 Introduction

Question answering (QA) is a long-standing task in NLP and IR. Having QA systems that perform well on real-world questions is of significant value for search engines and intelligent assistants. While some of the earliest work tackled the task of answering questions based on a large corpus (Voorhees and Tice, 2000; Voorhees, 2003; Wang et al., 2007) (albeit mostly focusing on simple fact-oriented questions), much of the recent work on QA has focused on answering questions in a less realistic setting – drawing the answer from a paragraph of text (Rajpurkar et al., 2016; Joshi et al., 2017), which is commonly referred to as machine reading comprehension (MRC).

In this work, we tackle the more realistic problem — candidate answers passages selection/reranking for real-world questions on the **web**. In contrast to both MRC and early work on QA from a large corpus, web pages often provide an additional source of knowledge. In particular, and thanks in part to the Semantic Web initiative (Berners-Lee et al., 2001), it is estimated that a non-trivial portion of web pages contain metadata annotations that provide a deeper understanding of the website content. The Web Data Commons project (Mühleisen and Bizer, 2012) estimates that 0.9 billion HTML pages out of the 2.5 billion pages (37.1%) in the Common Crawl web corpus[1] contain structured metadata. Figure 1 shows an example of this metadata which comes in the form of object-predicate pairs annotated with schema.org tags – a set of tags/predicates defined in the schema.org[2] hierarchy. In the example, the hierarchical metadata is used to add more structure to the web page of a recipe, providing meaning to the otherwise unstructured content. This makes several aspects of the recipe explicit – the preparation time (PREPTIME), cooking time (COOKTIME), ingredients (INGREDIENTS), etc. Figure 2 shows the "recipe" object in schema.org; it contains several properties such as COOKTIME, PREPTIME,

*Work conducted during internship at Microsoft Research.

[1] http://commoncrawl.org
[2] http://schema.org

Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pages 551–556
December 4 - 7, 2020. ©2020 Association for Computational Linguistics

Is selected	URL	Passage Text
✗	allrecipes.com ...	Preheat oven to 350 degrees F and lightly grease a ... instructions
✓	simplyrecipes.com ...	... Bake for **1 hour and 10 min** cookTime or until a meat thermometer inserted ...
✗	thekitchn.com ...	Any ground meat can be used to make meatloaf: beef, pork, veal ingredients ...
✗	livestrong.com ...	... loaf to stand for 10 to 15 min cookTime before slicing and serving it to 4-6 yield ...

Table 1: Example of answer passage selection on the Web. There are 4 candidate passages the query *"How long should I cook ground beef meat loaf in the oven?"*

RECIPEINSTRUCTIONS, etc.

We hypothesize that leveraging this metadata, in addition to the textual content, will improve the performance of QA systems on the Web. Table 1 presents an example of a query and several candidate passages. The candidate answer passages are decorated by colored spans that denote a corresponding schema.org predicate property. The correct answer (*"1 hour and 10 min"*) could be inferred from the metadata tag COOKTIME. While it seems clear from the example that the hierarchical schema.org metadata can be exploited in web QA, it will only be of true benefit if the use of metadata is prevalent in web pages. Luckily, this is the case as shown by Guha et al. who studied a sample of 10 billion web pages and showed that one third (**31.3%**) of the pages have schema.org markup.

To date, the end-to-end web QA systems have not made use of this metadata information. We first explore how to incorporate (and the effect of incorporating) semantic web hierarchical metadata into statistical NLP models for web-based QA. More specifically, we introduce a fine-grained encoding method for metadata predicates, to better leverage the semantic information in it. We evaluate the models on the answer passage selection/re-ranking task of MS MARCO (Nguyen et al., 2016), that contains real user queries sampled from the Bing search engine, with the answer passages extracted from real-world web pages. Results show that our approaches outperform the baseline systems substantially, with more significant gains on the subset of queries whose candidate passages contain richer metadata tags. Our work demonstrates the importance of encoding metadata information for QA, and verifies our hypothesis that the metadata knowledge can significantly benefit the performance of the neural models. We also provide qualitative analysis that includes performance comparisons across domains. Our findings further provide motivation for webmasters to annotate their web pages with semantic schema.org markup and for question answering systems developer to leverage them.

2 Related Work

Our work is related to several directions of work in semantic web, NLP and ML.

Metadata for NLP and ML Metadata like time stamp (Blei and Lafferty, 2006) and rating (Mcauliffe and Blei, 2008) have been successfully incorporated in document modeling. In community question answering, metadata is often used as hard features to improve the model performance – category metadata (Cao et al., 2010; Zhou et al., 2015) and user-level information and question- and answer-specific data (Joty et al., 2018; Xu et al., 2018). For answer quality prediction, author information (Burel et al., 2012; Suggu et al., 2016) has been often incorporated. In our work, we investigate how to leverage the *general* metadata knowledge from schema.org in web answer passage selection. Our metadata schema used, as compared to prior work mentioned, is structural and hierarchical, and applies to general web pages. The metadata could provide rich information to better understand the textual content on the web.

Semantic Web Berners-Lee et al. (2001) described the vision of the *Semantic Web*. The authors envisioned an extension of the World Wide Web, in which information is given well-defined meaning by bringing structure to the content of web pages. Ten years later, several major search engines have come together to launch the schema.org initiative, that to focus on creating, maintaining and promoting a common set of schemas for structured data markup on web pages. Webmasters use this schema to add metadata tags to their websites in order to help search engines understand the content. The use of such metadata has gained more popularity over the years.

3 Leveraging Metadata for Answer Passage Selection

In our setting of answer passage selection, the input to the system is a set of candidate passages $p_1, ..., p_n$, and a query q, the goal is to identify the passage that best answers the question.

For each candidate passage p_i, we have the URL_i of the web page from where it is extracted. The web document from URL_i, may contain a list of metadata object-predicate pairs $(\text{obj}_1, \text{pred}_1), ..., (\text{obj}_m, \text{pred}_m)$. The detailed approach of obtaining the pairs is presented in Section (3.1). Each predicate pred_j consists of a root r_j and a property pro_j (e.g., RECIPE and COOKTIME for /RECIPE/COOKTIME, respectively). We denote the path between r_j and pro_j as pt_j.

3.1 Generate Metadata-Decorated Passages

Algorithm 1 generates the decorated answer passages with metadata. The example for a decorated passage is shown immediately after the algorithm. The spans are marked up with the metadata predicate features. The decorated results are later used as input for our models. To be more specific, given the queryPsgExample (including query, candidate answer passage, URL, label of whether is selected) and metadata object-predicate pairs as input, we aim to obtain the queryPsgExamples whose candidate answer passages are decorated. We first obtain all the metadata pairs (matchingMetaPairs) for the URL where the passage text appears (line 1). Then, for each metadata pair in matchingMetaPairs, we employ a similarity function (MetaSim in line 6) to first compute the similarity between all possible text spans of the passage and the object text in the metadata object-predicate pair; afterwards the function records the start and end offset of the text spans which have a similarity score higher than the threshold. In our case, we use BLEU-4 (Papineni et al., 2002) as MetaSim. It calculates a score for up to 4-grams overlap using uniform weights. A metadata-decorated candidate passage with the algorithm is presented in Table 2.

Algorithm 1: How to obtain for metadata for each URL and generate metadata-decorated passage

```
  Data: queryPsgEg (query, psgText, URL, label),
        metaPairs (subj, pred, obj, URL);
1 matchingMetaPairs←Join(queryPsgEg[URL] ==
    metaPairs[URL]);
2 for each pair ∈ matchingMetaPairs do
3     if pair[obj] is not text then
4         continue;
5     else
6         startOffsets, endOffsets, score ←
            MetaSim(queryPsgEg[psgText], pair[obj]);
7         Decorate(queryPsgEg[psgText],
            startOffsets, endOffsets, pred);
8     end
9 end
```

word	Rinse	**tilapia**	**fillets**	in	cold	water	...
	Season	both	sides	with	**salt**	**and**	**pepper**
pred.	`O`	`B_R_ING`	`I_R_ING`	`O`	`O`	`O`	`O`
feature	`O`	`O`	`O`	`O`	`B_R_ING`	`I_R_ING`	`I_R_ING`

Table 2: Metadata-Decorated Candidate Passage

3.2 Neural Passage Selection with Fine-grained Metadata Encoding

We propose a simple but effective neural network structure for building our base neural passage selector (NPS). Similar to the neural reader (Hermann et al., 2015; Chen et al., 2017) for MRC, we first obtain a feature-rich (including the fine-grained encoding of the metadata) contextualized representation for each token in the passage and query. The output layer takes the passage and query representations as input and makes the prediction.

Fine-grained metadata embedding each predicate feature *pred* (e.g., /RECIPE/COOKTIME) includes the root r (RECIPE) and the property pro (COOKTIME). To leverage this information, we propose to leverage the hierarchy present on the predicate by learning the root embedding $\mathbf{E}_r$, the property embedding $\mathbf{E}_{pro}$, *as well as* the path embedding $\mathbf{E}_{pt}$ (RECIPE→COOKTIME), instead of only learning an embedding of the entire predicate (/RECIPE/COOKTIME). Thus, the final predicate feature encoding for token t_i is the concatenation of the three components: $\mathbf{E}_{pred}(pred_i) = \text{concat}(\mathbf{E}_r(r_i), \mathbf{E}_{pro}(pro_i), \mathbf{E}_{pt}(pt_i))$.

Passage & Query encoding We first represent each token t_i in the **passage** with a vector representation and pass it through a multi-layer BiLSTM (Hochreiter and Schmidhuber, 1997) network to get the contextualized representation for each token ($\mathbf{t}_1$, $\mathbf{t}_2$,...), where $\mathbf{t}_i$ is the concatenation of:

- *(Contextualized) word embedding:* GloVe `840B.300d` (Pennington et al., 2014) embeddings is used to initialize the embedding layer and is fine-tuned during training, we denote it as $\tilde{\mathbf{t}}_i$ for token t_i. Besides, we also use the pretrained contextualized representations produced by BERT (Devlin et al., 2019), $\hat{\mathbf{q}}_\mathbf{1}, ..., \hat{\mathbf{q}}_\mathbf{m}, ..., \hat{\mathbf{t}}_\mathbf{1}, ..., \hat{\mathbf{t}}_\mathbf{n} = \texttt{BERT}([\text{CLS}], q_1, ..., q_m, [\text{SEP}], t_1, ..., t_n)$. For the i^{th} token, the word embedding $\mathbf{E}(t_i)$ is the concatenation of the two.

- *Metadata predicate embedding:* We use the fine-grained predicate encoding of metadata pair ($\mathbf{E}_{pred}(pred_i)$), as described above. Embedding for beginning (B_) and intermediate (I_) tokens of a decorated span are different and learned during training; For the other passage tokens that are not metadata-decorated, their predicate (O) embedding are filled with zero vectors.

- *Aligned query embedding:* Similar to (Chen et al., 2017), we also incorporate the aligned query embedding. This feature is intended to capture the similarity between t_i and each query word q_j. For the i^{th} token t_i. It is calculated as: $\sum_j \mathbf{E}(q_j) * sim(\mathbf{E}(t_i), \mathbf{E}(q_j))$.

The encoding $\mathbf{p}_k$ for candidate passage k is the sum of the token representations after the BiLSTM. Similarly, **query** token embedding $\mathbf{q}_j$ is the concatenation of its contextualized word embedding ($\hat{\mathbf{q}}_\mathbf{j}$) and the GloVe embedding. We pass it through another BiLSTM, and use the sum operation to obtain the query encoding $\mathbf{q}$.

Prediction Finally, the "Is_selected" score for passage k is calculated as a function of the passage encoding $\mathbf{p}_k$ and the query encoding $\mathbf{q}$: $\text{score}(k) = \text{softmax}(\mathbf{p}_k W \mathbf{q})$. At test time, we calculate $\text{score}(1)$, ..., $\text{score}(n)$ for all the candidate answer passages, and select the passage with highest score: $\text{argmax}_k(\text{score}(k))$.

4 Experiments and Analysis

This section first presents the QA dataset that is used for evaluation, and then describe results comparing different methods (with or without leveraging the metadata information).

4.1 Datasets and Models

We evaluate our models on the passage selection task of **MS MARCO** (Nguyen et al., 2016), to our knowledge, this is currently the only large-scale real-world QA/MRC dataset on general web pages, that is paired with URLs from which the candidate passages are extracted. To measure how the models perform when trained and tested on a subset of queries from a focused domain, where the usage of schema.org metadata is more prevalent, we extract the QA pairs of the recipes domain from MS MARCO dataset and extend it with extra QA pairs in this domain (**Recipe-MARCO**). Table 3 shows the number of queries for the datasets. Although WikiQA (Yang et al., 2015) and Natural Questions (Kwiatkowski et al., 2019) also contain queries from real users, their answer candidates are restricted to be from Wikipedia. However, the adoption of schema.org tags in Wikipedia pages is very low (< **2.2%**[3]). This is significantly less than general web pages where the adoption rate of schema.org metadata is around **31.3%**. Thus we do not use these datasets for evaluation.

	MARCO	Recipe MARCO
Train	82,326	7515
Dev	10,047	835
Test	9650	846

Table 3: Statistics of Datasets.

We follow previous work (Yang et al., 2015; Tan et al., 2018) on reporting precision@1 (P@1) and Mean Reciprocal Rank (MRR). P@1 measures whether the highest scoring answer passage returned matches the correct passage. MRR (Voorhees and Tice, 2000) evaluates the relative rank of the correct passage in the candidate passages. We compare our models to several baselines, **S-Net** (Tan et al., 2018) is a prior state-of-the-art model on MS MARCO, it also produces synthetic answers and use text generation metrics (e.g., BLEU and ROUGE-L). In this work, we only compare to its capability of passage re-ranking. **NPS** is the baseline "neural passage selector" which does not encode metadata information. It's similar to the implementation in Dai and Callan (2019). **B-NPS** is a version of our model which builds upon NPS and *directly* encodes the entire predicate. **F-NPS** is our main model – fine-grained metadata encoding enriched neural passage selector. We also report the results of selecting the **first** and a **random** passage.

4.2 Results and Analysis

Table 4 shows the comparison of different methods on the candidate passage selection task. We see that: (1) By leveraging the metadata, both versions of our model (B-NPS and F-NPS) outperform the baseline NPS model; (2) With fine-grained encoding, F-NPS significantly outperforms all models in both P@1 and MRR. Particularly, F-NPS achieves higher P@1 than NPS by around 2%; (3) From the ablation study, we see the BERT pretrained representations consistently improve the performance, and leveraging the metadata information further improves it. We also present the results of different methods when trained and tested on Recipe-

[3] `http://webdatacommons.org/structureddata/2018-12/stats/stats.html`

	MARCO		Recipe-MARCO	
	P@1	MRR	P@1	MRR
First Passage	13.89	-	15.13	-
Random	13.76	34.76	11.35	30.67
S-Net (Tan et al., 2018)	28.30	-	-	-
NPS	32.80	51.72	41.68	59.73
w/o BERT	29.57	50.10	40.24	58.39
B-NPS	33.52	52.83	43.58	61.37
F-NPS	**34.70***	**54.21**	**44.37***	**62.46**
w/o BERT	33.01	52.96	43.42	61.13

Table 4: Evaluation results on datasets. Statistic significance is indicated with $^*(p < 0.05)$.

	Prop. (%)	NPS	F-NPS
book	6.37	29.06	**32.81**
medical	13.20	30.69	**34.46**
person	11.75	29.30	**32.51**
organization	13.32	30.12	**33.86**
review	3.09	27.42	**35.48**

Table 5: Analysis of P@1 performance for models w/ and w/o metadata information in diverse domains.

MARCO. We see that the relative increase of performances for F-NPS is more substantial.

Finally, we provide analysis on both the models and the effect of encoding metadata. Since not all web pages come with metadata, we turn our attention to the results describing the model performance on the portion of queries of MS MARCO that come with *at least one* metadata item ("**M-Rich-MARCO**"). We first perform analysis to understand how often the web pages in the dataset contain markup and how it affects the models performance. We see that for each query in MS MARCO, there are around 7.9 metadata pairs for its candidate passages; and 31.6 for queries in M-Rich-MARCO. On M-Rich-MARCO, the results we get on P@1 (F-NPS: 33.13, NPS 28.79) demonstrate that the performance gap between the model that leverages the metadata is larger than the general case. This, once again, demonstrates the effect of encoding metadata knowledge.

To better understand how the models perform and the effect of metadata on specific web domains, we report in Table 5 P@1 of models (trained on *entire* MS MARCO) on domains that are richer with metadata (i.e., book, medical, person, organization and review). We observe that queries in "medical", "person" and "organization" domains have a larger presence in the dataset (> 10%). The table also shows the performance of NPS and F-NPS on each domain. We see that F-NPS outperform NPS across all these domains. And the improvement is more substantial as compared to evaluating on the entire test set (the second column of Table 4).

5 Conclusion

We demonstrate benefits of incorporating metadata information from web pages for improving answer passage selection model. We describe methods for obtaining metadata and decorating passages with metadata object-predicate pairs, and a fine-grained encoding strategy for leveraging metadata information in neural models. For future work, we'll investigate metadata for other tasks such as web entity linking and extraction.

Acknowledgments

We thank the anonymous reviewers for helpful feedback and comments.

References

Tim Berners-Lee, James Hendler, and Ora Lassila. 2001. The semantic web. *Scientific american*, 284(5):34–43.

David M Blei and John D Lafferty. 2006. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120.

Grégoire Burel, Yulan He, and Harith Alani. 2012. Automatic identification of best answers in online enquiry communities. In *Extended Semantic Web Conference*.

Xin Cao, Gao Cong, Bin Cui, and Christian S Jensen. 2010. A generalized framework of exploring category information for question retrieval in community question answer archives. In *Proceedings of the 19th international conference on World wide web*, pages 201–210.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. 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, Vancouver, Canada. Association for Computational Linguistics.

Zhuyun Dai and Jamie Callan. 2019. Deeper text understanding for ir with contextual neural language modeling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 985–988.

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*, Minneapolis, Minnesota. Association for Computational Linguistics.

Ramanathan V Guha, Dan Brickley, and Steve Macbeth. 2016. Schema. org: evolution of structured data on the web. *Communications of the ACM*, 59(2):44–51.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *NeurIPS*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. 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. Association for Computational Linguistics.

Shafiq Joty, Lluís Màrquez, and Preslav Nakov. 2018. Joint multitask learning for community question answering using task-specific embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Jon D Mcauliffe and David M Blei. 2008. Supervised topic models. In *Advances in neural information processing systems*, pages 121–128.

Hannes Mühleisen and Christian Bizer. 2012. Web data commons-extracting structured data from two large web corpora. *LDOW*, 937.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *Workshop on Cognitive Computation (CoCo)*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. 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, Doha, Qatar. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Sai Praneeth Suggu, Kushwanth Naga Goutham, Manoj K. Chinnakotla, and Manish Shrivastava. 2016. Hand in glove: Deep feature fusion network architectures for answer quality prediction in community question answering. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*.

Chuanqi Tan, Furu Wei, Nan Yang, Bowen Du, Weifeng Lv, and Ming Zhou. 2018. S-net: From answer extraction to answer synthesis for machine reading comprehension. In *AAAI*.

Ellen M. Voorhees. 2003. Evaluating the evaluation: A case study using the TREC 2002 question answering track. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 260–267.

Ellen M. Voorhees and Dawn M. Tice. 2000. The TREC-8 question answering track. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece. European Language Resources Association (ELRA).

Mengqiu Wang, Noah A. Smith, and Teruko Mitamura. 2007. What is the jeopardy model? A quasi-synchronous grammar for QA. In *EMNLP-CoNLL*.

Steven Xu, Andrew Bennett, Doris Hoogeveen, Jey Han Lau, and Timothy Baldwin. 2018. Preferred answer selection in stack overflow: Better text representations... and metadata, metadata, metadata. In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, pages 137–147.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal. Association for Computational Linguistics.

Guangyou Zhou, Tingting He, Jun Zhao, and Po Hu. 2015. Learning continuous word embedding with metadata for question retrieval in community question answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 250–259, Beijing, China. Association for Computational Linguistics.