Dynamic Memory Network based Dual Attentive Network

Shihui Li, Sida Wang, Kangyan Zhou
Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
shihuil,sidaw1,kangyanz@andrew.cmu.edu

1 Introduction

Machine comprehension has gained popularity over the recent years in natural language processing field. Various kinds of technique have been proposed to solve this problem, among which memory networks[20] and its variants show good performance due to its ability to effectively process, store, and extract the supporting facts that can be used to answer the questions. Dynamic Memory Network Plus(DMN+) is one of the successful examples of the memory networks, demonstrating exceptional performance in bAbI task[21].

In this project, we propose the Dynamic Memory Network based Dual Attentive Network (DMN-DuAN) and experiment how to use DMN+ and a variant of character embedding, CharWNN[11], to model a certain kind of commonsense knowledge as well as the supporting facts to answer machine comprehension problems. The task we work on is *SemEval2018 Task 11: Machine Comprehension using Commonsense Knowledge*, and the commonsense knowledge we use is script knowledge, which describes steps to complete a simple daily action. We choose script knowledge since most passages in the task are very similar to script knowledge.

Our choice of DMN+ and CharWNN originates from our error analysis on the results produced by Three-way Attentive Network (TriAN), the second place solution in the competition. By analyzing results from the TriAN model, we found that errors can be categorized into four types. The main error type (48%) is that the model is unable to infer the correct answer from the passage, which is mostly a sequence of actions. The second dominating error type (20%) points to the model's incapability to borrow commonsense knowledge from external knowledge source. The remaining errors expect a better matching mechanism between the answer and the passage. We experiment building different neural network structures based on TriAN, and the results show that incorporating script knowledge and character embedding into TriAN indeed help the model in making better judgments.

2 Related Work

Memory networks[20] refer to a type of network structure that consists of four parts: Input, which converts input text to embeddings; Generalization, which determines which memory slot the input embeddings belong to; Output, which takes the questions and selective memory slots to produce feature vectors for answers; and Response, which aims to generate final answers. [17] proposed an end-to-end trainable memory network by replace the hard attention part in [20] with a softmax layer. [7] proposed dynamic memory network that have a more complex component for the generalization and output parts. [22] improved the dynamic memory network by slightly modifying the structure of the network and discussing its application in visual question answering(VQA). We will have a detailed introduction of this model(DMN+) below. Key-Value Memory Networks[8] use the hash of the input questions representation to perform a similarity search among all the key embeddings, and the result is used to fetch the embeddings of the value corresponding to the keys. It is considered a general structure that is independent of tasks.

Besides memory networks, some other methods are also proposed to solve machine comprehension problems. Dynamic coattention network[22] combines the co-dependent representations of the question and the document and use a combination of Highway Network[16] and Maxout Network[3] iteratively compute the starting and ending position of an answer based on the co-attention output. It demonstrates a stable performance regardless of the length of documents and questions. The model became the state-of-the-art soon after the release of the SQuAD dataset. In the field of cloze-style QA, the Attention Sum Reader model[4] selects the most likely answer simply based on the dot product between the question embedding and the contextual embedding of each token in the answer. The downside of these models is that all of them can only extract instead of compose answers from the given documents and cannot give a summary based on several documents. The complexity of these models also restricts their application in small datasets. Another popular model is BiDAF[13], where it computes the attentions between questions and context, and uses the attention to enhance both representations.

Not specific to machine comprehension tasks, people also try to include character level awareness in the network structure. The advantages of this fine grained method are that it increases the model's ability to utilize subword information for open vocabulary problems and that it saves the effort to perform feature engineering on the word type such as capitalization. Applications can be found such as text classification [24], language modelling [5], part-of-speech tagging [11], etc.

3 Task Overview

3.1 Competition Data Overview

In *SemEval-2018 Task 11: Machine Comprehension using Commonsense Knowledge*, the passages are presented with narrative texts about everyday activities. The models should be able to answer multiple-choice questions based on the texts. The question usually focuses on particular actions in the sequence. For each question, two candidate answers are provided and only one of them is correct. Different from common machine comprehension tasks, some questions in the task can only be answered with commonsense knowledge instead of using information from the given passage. An example of the passage, question, and text is shown in table 1. The statistics for the data is shown in table 2.

Passage	I first worked on the counter tops. I used a special marble cleaner to clean them... I mopped the floor of the bathroom, washed out the bathtub with shower cleaner...
Question	How did they clean the tub?
Candidate Answer 1	Wash out with shower cleaner *
Candidate Answer 2	With marble cleaner

Table 1: An example of the passage, question, and text for SemEval-2018 Task 11. Only the most related parts of the text to the question is shown. The answer marked with * in the end is the correct answer.

	Passage Count	Question Count
Training Data	1536	10164
Development Data	218	1411
Test Data	429	2797

Table 2: The statistics for the data.

3.2 Script Data Overview

Script knowledge is a body of knowledge that describes a typical sequence of actions people do in a particular situation[12]. An example of script knowledge is shown in table 3. We will also define some terms to ease the explanation of the usage of script knowledge. *Sequence* refers to the name of script knowledge, such as 'baking a cake'. *Description* refers to one description of the sequence . *Action* refers to the actions in the description, such as 'Get your recipe'. Please refer to 1 for an illustration of the script knowledge structure.

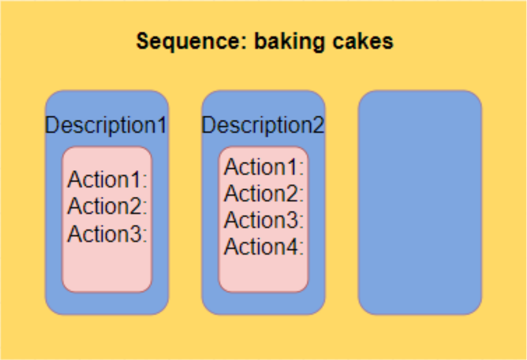

Figure 1: Structure of script knowledge: each sequence is followed by multiple descriptions and in each description, there is a list of actions to accomplish this task.

Gather your cooking materials for baking a cake
Get you cooking supplies
Get your recipe
Begin the process of making your cake
When finished cut and eat
Store any leftovers in the fridge

Table 3: The script knowledge of baking a cake.

number of sequences in the script knowledge base	219
average number of descriptions for a sequence	57.75
average lengths (number of actions in a description) of descriptions for a sequence	5.11
average utterance length of each action in a description	4.88

Table 4: Statistics for script knowledge

The script knowledge we use is the combination of DeScript[19], RKP[10], and OMCS stories[14]. Some statistics for our script knowledge base is shown in Table 4. We notice that most of the sequences of actions in the passages can be found in the script knowledge sources, however there is no ground truth mapping from a passage to the script knowledge.

4 Method

4.1 Some Common Notations

We will first list some of the common notations. Each training example consist of a passage P, a question Q, and a answer A, a label Y. Each symbol represents a sequence of word indices. The passage, question, and answer indices will first feed into a glove embedding layer: the output is the passage representation w_{P_i}, the question representation w_{Q_i}, and the answer representation w_{A_i}.

4.2 Review of TriAN

Our model is based on the state-of-the-art model for the same task, which is called TriAN[18]. The high-level idea for TriAN is very similar to the BiDAF: the key point of the model is to compute question-aware passage representation $\{w^q_{P_i}\}_{i=1}^{|P|}$, passage-aware question representation $\{w^p_{Q_i}\}_{i=1}^{|Q|}$ and question-aware answer representations $\{w^q_{A_i}\}_{i=1}^{|A|}$. These representations are computed via weighted sum of sequence attention score following [1]:

$$Att_{seq}(u, \{v_i\}_{i=1}^{n}) =(u, \{v_i\}_{i=1}^{n}) = \sum_{i=1}^{n} \alpha_i v_i \tag{1}$$

$$\alpha_i =softmax_i(f(W_1 u)^T f(W_1 v_i)) \tag{2}$$

$$\{w_{P_i}^{q}\}_{i=1}^{|P|} =Att_{seq}(w_{P_i}, \{q_i\}_{i=1}^{n}) \tag{3}$$

$$\{w_{Q_i}^{p}\}_{i=1}^{|Q|} =Att_{seq}(w_{Q_i}, \{p_i\}_{i=1}^{n})) \tag{4}$$

$$\{w_{A_i}^{q}\}_{i=1}^{|A|} =Att_{seq}(w_{A_i}, \{q_i\}_{i=1}^{n})) \tag{5}$$

$$\tag{6}$$

These representation $\{w_{P_i}^{q}\}_{i=1}^{|P|}$, $\{w_{Q_i}^{p}\}_{i=1}^{|Q|}$, $\{w_{A_i}^{q}\}_{i=1}^{|A|}$ are then fed into a BiLSTM to model the temporal dependencies. The outputs from BiLSTM $\{h_{P_i}^{q}\}_{i=1}^{|P|}$, $\{h_{Q_i}^{p}\}_{i=1}^{|Q|}$, $\{h_{A_i}^{q}\}_{i=1}^{|A|}$ are then summarized with self attention as in [23]

$$Att_{self}(\{u_i\}_{i=1}^{n}) = \sum_{i=1}^{n} \alpha_i u_i \tag{7}$$

$$\alpha_i = softmax_i(W_2^T u_i) \tag{8}$$

$$p_{final} = Att_{self}(\{h_{P_i}^{q}\}_{i=1}^{|P|}) \tag{9}$$

$$q_{final} = Att_{self}(\{h_{Q_i}^{p}\}_{i=1}^{|Q|}) \tag{10}$$

$$a_{final} = Att_{self}(\{h_{A_i}^{q}\}_{i=1}^{|A|}) \tag{11}$$

$$\tag{12}$$

The final output is computed by the bilinear interaction between question/answer and passage/answer

$$y = \sigma(p_{final} W_3 a_{final} + q_{final} W_4 a_{final}) \tag{13}$$

Notice in this model the passage representation is also enhanced with embeddings from part-of-speech and named-entity in the passage.

4.3 Review of DMN+

We also borrow the idea from DMN+. There are four modules in the DMN+: the input module, the question module, the episodic memory module, and the answer module. We will only discuss the input module and the episodic memory module, which we directly use in our model.

4.3.1 Input Module

The input module is a hierarchical encoder for the input sequences of sentences. Each sentence s_i = $[w_{i1}, w_{i2}, ...w_{iM}]$, where M is the length of the sentences and w is the representation of each word, is first encoded with sentence encoding[17]: f_i = $\sum_{j=1}^{M} l_j \circ w_{ij}$, where $\circ$ is elementwise multiplication, and l_j is the column vector of the matrix e $l_{kj} = (1 - j/M) - (k/d)(1 - 2j/M)$, where d is the embedding dimension. Then a bidirectional GRU[2] is used to model the interaction between the sentences. The final representation of each sentence, c_i, is the sum of the output of forward GRU and backward GRU.

4.3.2 Episodic Memory Module

Episodic memory module is designed to allow for interactions between input facts and logical reasoning over ordered inputs for multiple passes. It consists of two separate components: the attention mechanism and the memory update mechanism.
In each pass, the attention mechanism is responsible for producing a contextual vector c^t, where $c^t \in \mathcal{R}^{n_H}$ is a piece of question-aware information derived from episode memory of last pass m^{t-1}.

The initial memory vector is set to the question vector $m^0 = q$. The memory update mechanism takes the context vector c_t and memory from previous pass m^{t-1} as input and generates an up-to-date episodic memory m^t. The memory update mechanism can be modeled by various functions, such as MLP and bilinear functions. The episodic memory from the last pass m^T should contain all the information from script required to answer the question q.

4.4 Review of CharWNN

CharWNN is employed to encode the morphological and n-gram level information of words. To obtain the character-level embedding for a word, several steps are taken. Assume the word has length m. Each character in the word will first go through the character embedding matrix to obtain the vector $cEmb_i \in R^{1\times n}$. Then the concatenation of these vectors $CEemb \in R^{m\times n}$ is fed into a convolution layer. Finally a max pooling layer is applied to produce the embedding for the word. The size of the final embedding is equal to the number of filters in the convolution layer.

4.5 Dynamic Memory Network based Dual Attentive Network

We propose a Dynamic Memory Network based Dual Attentive Network (DMN-DuAN) as shown in figure 2. We use the input module and the episodic memory module from the previous section. The input for the input module is the sequences of all the sentences from the given passage and all the actions from one random description of the sequence described in that passage, if there is a match. The query of the input of the episodic memory module is the passage-aware question representation $\{w^p_{Q_i}\}_{i=1}^{|Q|}$. The output of the memory module, m, is used to replace the p_{final} in the 13.

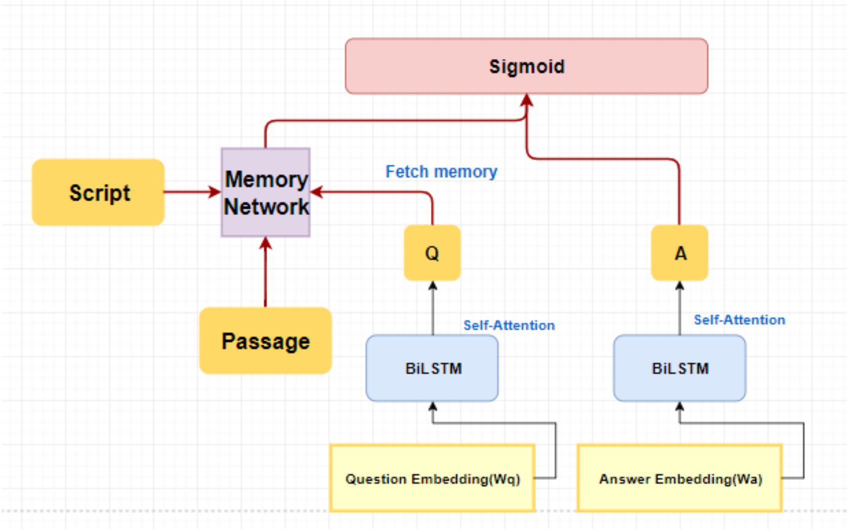

Figure 2: Proposed Model Architecture

5 Experiment and Analysis

5.1 Choosing Script Knowledge

Since there is no ground truth matching between the passages in the training examples, we develop a simple heuristic to match the passage to a sequence in the script knowledge base, if any. Each sequence is encoded as key-value pairs, where each key is one of the 5 most frequent words in all the descriptions, and value is the count of that word. Given a passage, the first step is rule based: if some keywords are present in the passage, such as ’vending’, the output would be ’buy from vending machine’. The next step is to pick two candidate sequences from the script knowledge, by ranking the sum of the value of the intersections of all the words in the passage and the keys in the key-value pairs. If all the size of intersections equal to 0, then there does not exist a match. Then the three candidates are re-ranked by the size of intersections. This simple heuristic gives roughly over 80%

classification accuracy, by our human evaluation[1], thus we do not explore any advanced technique, such as NER.

After getting the sequence name, during each batch, we randomly pick one description from the 5 longest descriptions in that sequence. Our assumption here is that longer descriptions will provide more details for the sequence, but only using the longest one will have some bias.

5.2 Experiment Setup

Each passage, script knowledge description, question, and answer will first go through Spacy tokenizer[2] to tokenize the input text. We use Glove embedding[9] for the word embeddings. We do not have any pretraining for the model. The dropout rate is set to be 0.4 for both embedding dropout and BiLSTM output[15]. The character embedding size is set to be 200. The optimizer we choose is Adamax[6], with the initial learning rate set to be 0.001, and decreased by half over every 5 epochs. The model is trained for 15 epochs. The batch size is set to be 32. The number of hop in the memory network is set to be 1.

5.3 Result

Model	Dev Accuracy	Test Accuracy
TriAN	82.71%	80.51%
TriAN with charemb	82.77%	81.12%
DMN-DuAN w/o script knowledge w/o charemb	83.13%	**82.59%**
DMN-DuAN w/o script knowledge	83.91%	82.05%
DMN-DuAN w/o charemb	**84.06%**	81.65%
DMN-DuAN	83.98%	82.05%

Table 5: Experiment Result: in DMN-DuAN, we pass the sentences from the passage as well as the sentences from the description to the input module while in DMN-DuAN w/o script knowledge, we use only the sentences from the passage as the input.

The results are shown in Table 5. Several interesting findings are:

- DMN+ is more capable of capturing information in the passage, compared to the original passage encoding method. From all combinations of modifications, we can see at least 1% improvement on the test accuracy. This validates the effectiveness of hierachical structure and the iterative attention which bases its attention on the input and the previous results.
- CharWNN is not always helpful. Improvements can be observed when using character embedding on TriAN and on our model with script knowledge. However, the best results are achieved *without* character embedding. This could result from the relatively small size of the corpus (thus fewer unknown words). Since we only use a fixed window size (5), it may restrict the model's capability to capture n-gram features with different n.
- Script knowledge does not consistently improve the model performance though it works well on development dataset. It could be due to several reasons. One is that for the incorrect predictions, there are no corresponding scripts that answer the question. Another possibility is that a certain level of inference is still needed to extract the correct answer from the script knowledge.

5.4 Error Analysis & Future Work

We perform a comparative study on the errors pruduced by TriAN and DMN-DuAN. We found that 40.7% errors (represented by questions) in the TriAN do not appear in DMN-DuAN and 35.5% errors do not appear in TriAN, which means though resolving 40.7% errors in TriAN, DMN-DuAN produces a certain number of other errors.

By manually evaluating the error types of 10 error samples specific to each model (shown in 6), we found no statistical difference between the error type distribution of the two models.

[1] We manually look into the first 50 examples of the development set, and compute the precision.

[2] https://spacy.io/api/tokenizer

We also study the question word distribution of the errors specific to each model in 3. We found that DMN-DuAN makes fewer mistakes in answering the *what* and *who* questions, which may indicate its ability to keep the actor and object consistency between sentences. While higher error rate on *why* question shows it is less capable of reasoning.

For more detailed analysis, we expect to further label each question with relevant scripts, which would help us better understand how many scripts are useful for answering questions and how to employ the script knowledge in a more effective way. In the meanwhile, since different models are better at certain question types, knowing their strengths and weakness might help us design models with better performance by ensembling existing models.

Error Type	TriAN	DMN-DuAN
Passage Inference	5	4
Commonsense	0	2
Simple Matching	4	3
Answer Confusion	1	0
Other	0	1

Table 6: Manual Categorization of 10 Model Specific Error Samples: no statistical difference was found.

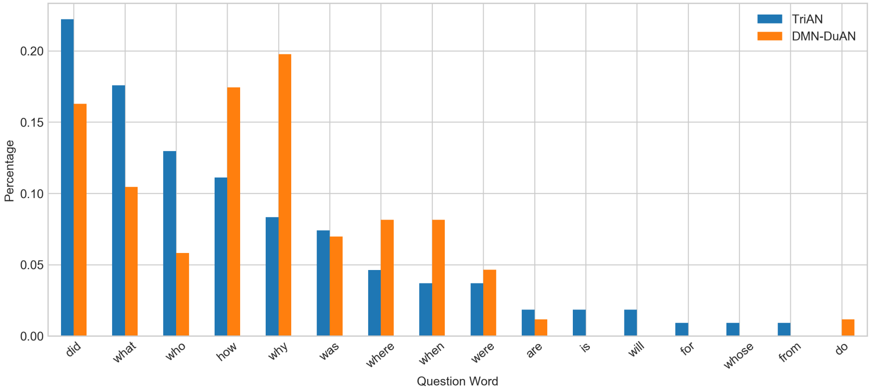

Figure 3: Question Word Distribution of Model Specific Error Samples in the Validation Set.

6 Conclusion

Using script knowledge proves to be hard in machine comprehension and no leading team participating in SemEval Task 11 succeeded in improving their model with script knowledge. We propose a two-step architecture enabling our model to logically reason over script knowledge so as to correctly answer multiple-choice questions: first, we use IR techniques to retrieve the relevant description from a large script knowledge base; second, we adopt a dynamic memory network to allow the model to learn critical information from the description text as well as the passage in the question. Our model does show some improvements especially on development dataset. Our architecture is not restricted to multiple-choice question answering tasks: it can easily be extended to general question answering tasks.

References

[1] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*, 2017.

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

[3] Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. *arXiv preprint arXiv:1302.4389*, 2013.

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

[5] Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. Character-aware neural language models. 2016.

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

[7] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. In *International Conference on Machine Learning*, pages 1378–1387, 2016.

[8] Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. Key-value memory networks for directly reading documents. *arXiv preprint arXiv:1606.03126*, 2016.

[9] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[10] Michaela Regneri, Alexander Koller, and Manfred Pinkal. Learning script knowledge with web experiments. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 979–988. Association for Computational Linguistics, 2010.

[11] Cicero D Santos and Bianca Zadrozny. Learning character-level representations for part-of-speech tagging. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1818–1826, 2014.

[12] Roger C. Schank and Robert P. Abelson. *Scripts, Plans, Goals and Understanding: an Inquiry into Human Knowledge Structures*. L. Erlbaum, Hillsdale, NJ, 1977.

[13] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*, 2016.

[14] Push Singh, Thomas Lin, Erik T Mueller, Grace Lim, Travell Perkins, and Wan Li Zhu. Open mind common sense: Knowledge acquisition from the general public. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 1223–1237. Springer, 2002.

[15] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[16] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

[17] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448, 2015.

[18] Liang Wang. Yuanfudao at semeval-2018 task 11: Three-way attention and relational knowledge for commonsense machine comprehension. *arXiv preprint arXiv:1803.00191*, 2018.

[19] Lilian DA Wanzare, Alessandra Zarcone, Stefan Thater, and Manfred Pinkal. Descript: A crowdsourced corpus for the acquisition of high-quality script knowledge. In *The International Conference on Language Resources and Evaluation. http://www. lrec-conf. org/proceedings/lrec2016/pdf/913_Paper. pdf*, 2016.

[20] J. Weston, S. Chopra, and A. Bordes. Memory Networks. *ArXiv e-prints*, October 2014.

[21] Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*, 2015.

[22] Caiming Xiong, Victor Zhong, and Richard Socher. Dynamic coattention networks for question answering. *arXiv preprint arXiv:1611.01604*, 2016.

[23] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, 2016.

[24] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015.