R-NET with BiDAF for Reading Comprehension

Zibo Gong **Jingwei Ji**
Department of Electrical Engineering, Stanford University
{zibo, jingweij}@stanford.edu

Abstract

In our project, we focus on the problem of reading comprehension in the form of question answering. As the mainstream setup, we aim to answer question given a passage or document. Inspired by the the Bi-Directional Attention Flow in BiDAF, the gated self-matching attention mechanism and the pointer network in R-NET, we combine these ideas in an end-to-end model leveraging different input modalities including words and characters. Our experiments are performed on the Stanford Question Answering Dataset (SQuAD), and without engineering tricks and ensembling, we show promising quantitative and qualitative question answering results on a single model. Our single model of R-NET + BiDAF has achieved **75.602% F1** and **64.730% EM** on dev set of SQuAD, and we expect higher performance with model ensembling and on test set.

1 Introduction

In this work, we focus on reading comprehension task in the form of question answering. Given a passage or a document as context, and a question related to the context, the task is to answer this question leveraging information provided in the context.

In the recent years, reading comprehension has been a popular task and an important indicator of the development of natural language processing. Its main competition on the Stanford Question Answering Dataset (SQuAD) (1) has stimulated many fundamental architectures in natural language processing with deep learning. In the process of exploring better algorithms for reading comprehension, people has found the importance of attention mechanism (2). Among the variety of deep models tackling this problem, BiDAF (3) and R-NET (4) has been proved to be two of the most successful models tackling the task, and both of them have proposed their mechanisms for attention. BiDAF is famous for its coupled attention from both *context to question* and *question to context*, which inspires us in designing module to connect pieces of information between context and question. Besides bi-directional attention, another important attention mechanism that has been proposed in R-NET is self-matching in context features, which can effectively aggregate evidence from the whole passage to infer the answer.

Holding the philosophy of "attention is all you need", we combine both attention mechanisms in this work. We call our architecture **R-NET+BiDAF**, which first leverages bi-directional attention flow between context and question, then utilizes gated self-matching attention to aggregate extracted context features, finally, employ a pointer network as in (5; 6; 4) for generating answer. We train our model on SQuAD train split, and evaluate it on dev and test splits. We show that even without engineering tricks and model ensembling, we achieve 75.12% F1 and 64.26% EM on the dev split of SQuAD.

2 Related Work

SQuAD. Stanford Question Answering Dataset (SQuAD) (1) is consist of more than 100,000 question-answer pairs from 500+ articles, and it is one of the biggest reading comprehension datasets.

A publicly accessible leaderboard (https://rajpurkar.github.io/SQuAD-explorer/) shows the fierce competition of state-of-the-art approaches, and some of them have even exceeded human performance according to certain metric.

State-of-the-Art Works on SQuAD. Among the renowned methods on SQuAD leaderboard, BiDAF (3), R-NET (4) and their inheritors have shown their superior performance in this task. We start from these two models, combine their different attention mechanisms, and form our final model.

Low level language feature embedding. Our model considers both word and character as units of language. Prior works (7; 8; 9; 10) have demonstrated powerful representation learning over low level language feature. In this work, we utilize such techniques for word and character embeddings.

3 Problem Statement

The problem statement follows the mainstream works on reading comprehension task on SQuAD. Given a word sequence of context with length T, $P = \{p_1, p_2, \ldots, p_T\}$ and a question with length N, $Q = \{q_1, q_2, \ldots, q_N\}$, the question answering problem is to find a function $f : (P, Q) \rightarrow (A_s, A_e)$, where (A_s, A_e) is the answer to the question Q in the form of a start and an end position in the context P, such that $1 \leq A_s \leq A_e \leq T$.

4 Model

As illustrated in Figure 2, we combine R-NET framework with BiDAF attention module in the following way: we replace the gated attention-based recurrent network in R-NET by BiDAF attention module. The following subsections describe in detail every module's function.

4.1 Word and Character Embeddings

Our model takes two modalities of sentences as inputs: words and characters. Word embedding has been well-developed by models including Word2Vec (7) and GloVe (8), and similarly, (9; 10) has provided general character embedding techniques. Following this line, we also use GloVe word embedding as the main input to our network. Besides, we also utilize character embedding in the network as another source of low level language feature. The character embedding could relieve the problem of Out-of-Vocabulary (OOV). During testing, if one word is never seen, we can still dissemble the word into characters and embed them, such that informative features could be extracted even if a work is OOV.

4.2 BiDAF Attention Module

Instead of using the gated attention-based recurrent network in R-NET, we resort to use BiDAF attention. The bi-directional attention flow offers a query-aware context representation, which is proved to be more effective in our model.This attention doesn't summarize all question words or context words to a simple feature. Instead, it allows feature vectors at each time step to flow through the subsequent modeling layer. That reduces the information loss.

We represent the context by H and the query by U. We first calculate the similarity matrix, S between the contextual embeddings of the context and the query. The S_{tj} indicates the similarity between $t-$th context word and $j-$th query word. The similarity matrix is computed by

$$\mathbf{S}_{tj} = \alpha(\mathbf{H}_{:t}, \mathbf{U}_{:j}) \tag{1}$$

$$\alpha(\mathbf{h}, \mathbf{u}) = \mathbf{w}_{(S)}^{T}[\mathbf{h}; \mathbf{u}; \mathbf{h} \circ \mathbf{u}] \tag{2}$$

Now, we use **S** to obtain the attentions and the attended vectors in both directions.
Context-to-query Attention Context-to-query Attention signifies which query words are most relevant to each context word. $\mathbf{a}_t = softmax(\mathbf{S}_{t:})$, and subsequently each attended query vector is $\tilde{\mathbf{U}}_{:t} = \sum_j \mathbf{a}_{tj}\mathbf{U}_{:j}$ **Query-to-context Attention** Query-to-context Attention signifies which context words have the closest similarity to one of the query words. $\mathbf{b} = softmax(max_{col}(\mathbf{S}))$. Then the

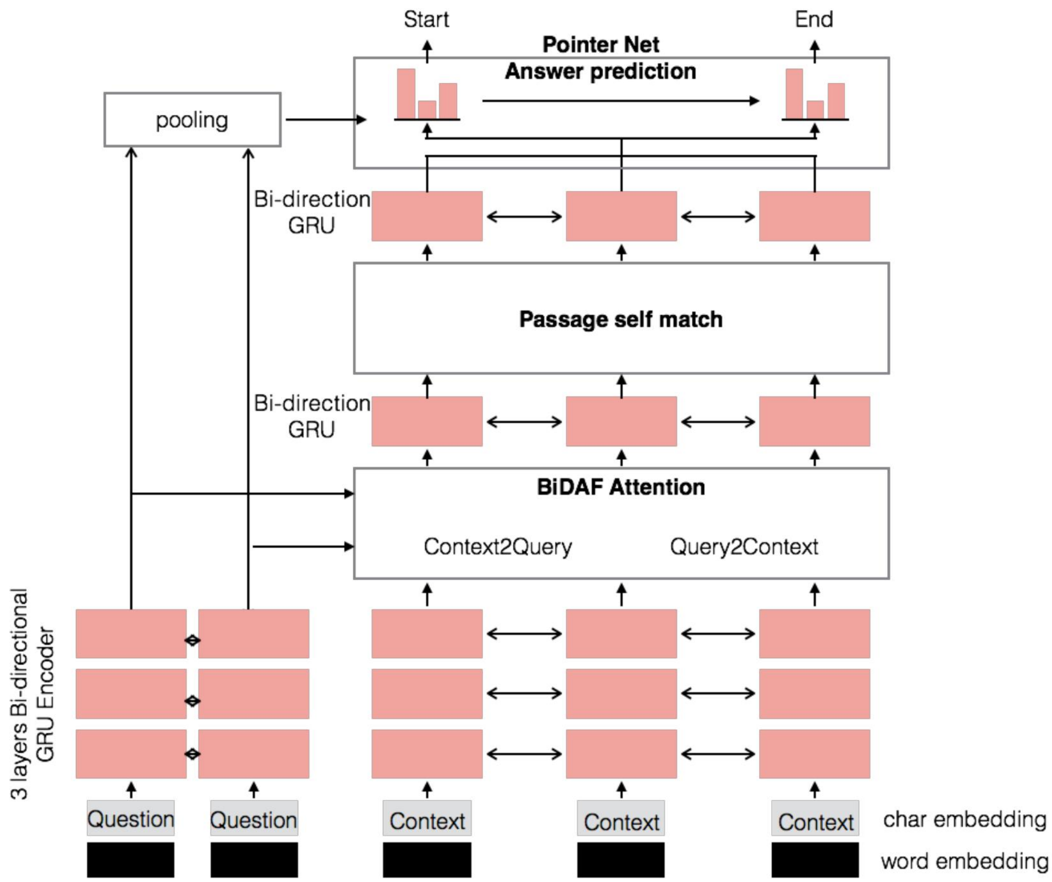


Figure 1: The architecture demonstration of our model. Our model mainly follows the R-NET architecture, with key components of word and character embeddings, bi-directional GRU encoders, passage self match attention module, and pointer net for answer prediction. Differently, we use BiDAF attention after word and character embedding, which captures the mutual attention relationships between context and query question.

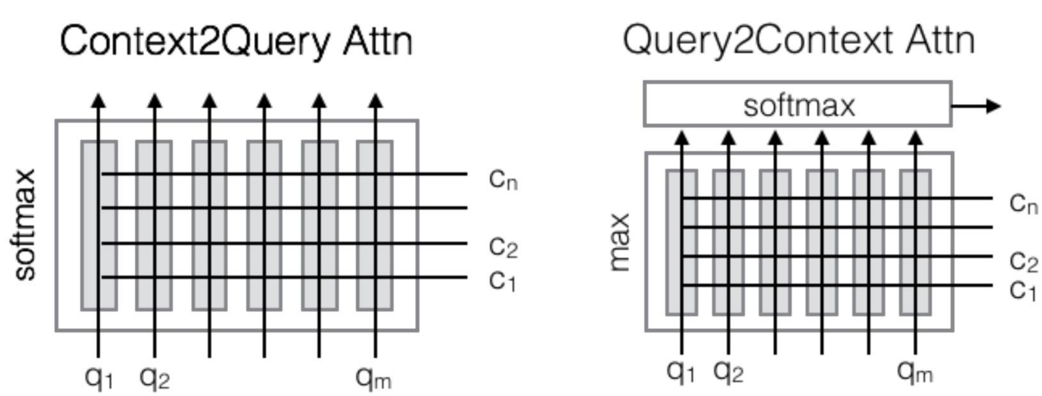


Figure 2: Illustration of BiDAF attention module. BiDAF attention contains both context-to-query and query-to context attentions.

attended context vector is $\tilde{\mathbf{h}} = \sum_t \mathbf{b}_t \mathbf{H}_{:t}$.
Finally, combine them to yield $\mathbf{G}$,

$$\mathbf{G}_{:t} = \beta(\mathbf{H}_{:t}, \tilde{\mathbf{U}}_{:t}, \tilde{\mathbf{H}}_{:t}) \tag{3}$$

In our model, $\beta = [h; \widetilde{u}; h \circ \widetilde{u}; h \circ \widetilde{h}]$.

4.3 Self-Matching Attention Module

After the BiDAF attention module, question-aware context representation $\{v_t^P\}$ is generated. The problem of this representation is that it's highly focused on the relation between context and question, but has limited knowledge of the context itself, especially the relation between parts of context. Therefore, R-NET proposes a self-matching attention mechanism that matches the question-aware context representation $\{v_t^P\}$ against itself. Formally, it generates another level of passage representation h_t^P:

$$h_t^P = BiRNN(h_{t-1}^P, [v_t^P, c_t]) \tag{4}$$

where $c_t = att(v^P, v_t^P)$ is an attention-pooling vector of the whole context v^P. Besides, an additional gate is applied to $[v_t^P, c_t]$ to control the input of RNN. Please refer to (4) for more mathematical details of the attention-pooling computation and the control gate.

4.4 Pointer Network

We use pointer networks(5) to predict the start and end position of the answer. We use an attention-pooling over the question to generate the initial hidden vector for the pointer network.Given passage representation h_t^P, start position p^1 and end position p^2. Please refer the implementation to (4)

5 Experimental Results

5.1 Implementation and Training Details

Hyperparameters. For GloVe word embedding, we choose embedding size of 300, which in general captures the word features the best. In all of our RNN cells, we keep hidden size of 75. We keep dropout rate as 0.15 during training and testing.

Overfitting Avoidance. Benefiting from the use of dropout layers in all RNN cells, training our network does not show significant overfitting. As shown in Figure 3, even when the network is trained for 20000 iterations, the performance gaps between train and dev sets have been kept around 0.10, on both F1 and EM metrics. Moreover, the performance gaps have not significantly enlarged during training, which indicates the robust generalizability of our network.

Padding Strategy. The raw contexts and questions are of different lengths. To fit all of them into the fixed size of batches, we pad zeros to context and question embedding to align them with the same length. While in computing losses and gradients, the values related to padded positions won't be computed so that the padded part will not interfere training process. In practice, we pad the context length to be 450, and question length to be 30. Besides, similar strategy is utilized in character processing, and we pad or truncate each word's character length to be 15.

OOV handling. For Out-of-Vocabulary problem, we do not resort to hand-designed engineering trick. Instead, our extra character embedding could handle this problem by splitting unseen word into characters. In the rare case that some character is not seen in the training, in (11) there is a special UNKNOWN token feature, which will be used as the embedding vector for any unknown character.

Ensembling Strategy. Since we use dropout layers during training and testing, ensembling different models help "rebuild" the missed neuron connections during inference. We use a late fusion style of ensembling to boost the performance with our designed model. Specifically, we average different models' predicted probabilities for start position $\{p_{s,t}\}$'s, and for end position $\{p_{e,t}\}$'s, to be the ensemble probabilities. Then argmax gives out the prediction of A_s and A_e of the ensemble model.

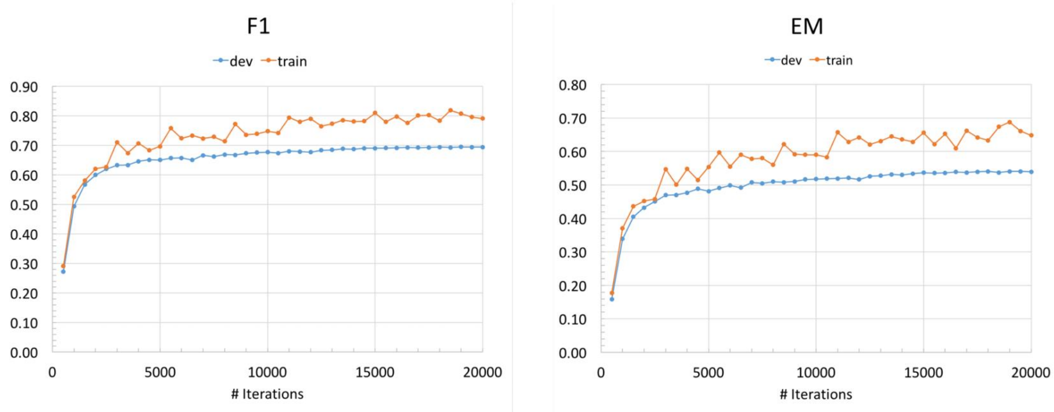


Figure 3: The learning curve of F1 and EM on train and dev set. The evaluations on dev set is computed over the whole dev set, while the values on train set are generated from only a part of the split, thus the curve on train set is more jittered.

Table 1: The quantitative results on SQuAD dev set. We evaluate different models on the commonly used metrics of F1 and EM. By comparison, we find that our R-NET+BiDAF (single model) outperforms both BiDAF and R-NET respectively. Note that up until this point, we haven't tested our model on the test set. The results on test set will be shown on leaderboard soon.

Model	**EM-dev**	**F1-dev**
Baseline	34.749	43.752
BiDAF	61.864	72.421
BiDAF (3 layers)	62.791	73.059
R-NET	62.623	72.294
Ours (Single Model)	**64.730**	**75.602**
Ours (Ensemble of 3 Models)	**66.537**	**77.030**

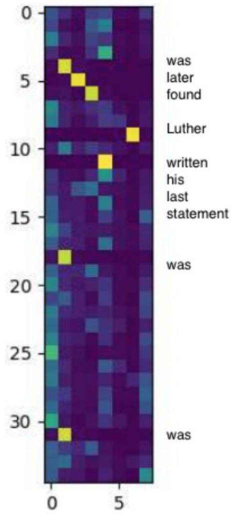
5.2 Ablation Study

Baseline. The baseline model is provided by CS224N Course Assistant team, which uses GloVe word embedding as a start point, and stack one layer of bi-directional GRU encoder for leveraging information between words. A simple dot attention is used to find attention between context and question, and finally fully-connected layers are utilized to generate predictions for answer's start and end points. We report our results on the baseline model in Table 1.

BiDAF. Built upon the baseline model, we introduces BiDAF attention module into the network replacing dot attention, where we observe significant performance boost as shown in the 2nd row in Table 1. Also, we stack 3 layers of bi-directional GRU encoders to better extract language feature through RNN structure, and we observe further performance improvement as shown in the 3rd row in Table 1.

R-NET. Although BiDAF exploits the mutual or "outside" attention between context and question, however, context itself should also has "inside" attention. For example, as human we often refer to the semantic context when we are reading materials and trying to understand concepts. Such behavior is not well represented in the BiDAF network. We resort to R-NET, which introduces a self-matching mechanism which is functioned to find attention inside context itself. We observe that R-NET's performance is also much higher than the baseline, indicating the importance of modules in R-NET architecture.

Full model: R-NET+BiDAF. Finally, we combine these two attention mechanisms together to formulate our full model. In Table 1, we show the results on dev set on the single end-to-end model, and also the performance of ensemble model, both of which outperform single BiDAF or R-NET on F1 and EM metrics. We are currently training more models with the same hyperparameter setup, and we expect higher performance of ensemble results when more trained models available.



Context word	Question word
was	was
later	later
found	discovered
luther	luther
written	written
his	written by
last	later discovered
statement	discovered, ?

Figure 4: Context_to_question Attention visualization and corresponding words between context word and question word.

5.3 Attention Analysis

In this section we visualize and analyze Bidaf attention matrix.

Example:

- **Context:** a piece of paper was later found on which Luther had written his last statement. the statement was in Latin , apart from " we are beggars , " which was in German.
- **Question:** what was later discovered written by luther ?
- **Prediction:** his last statement
- **Answer:** his last statement

In Figure 4, we visualize the heatmap of Context_to_question Attention score. In the context, the word "was", "later" and "Luther" are matched to the same words in the question. In addition, for the phrase "his last statement", the model focuses on "later discovered written by".

In Figure 5, we visualize the Question_to_context Attention score for this example. We find that for that question, the model focuses on the word "written", the beginning of the true answer.

From this simple example, we find that the model effectively match the important features in context and question.

5.4 Error Analysis

Wrong attention

- **Question:** who funds the ipcc 's deputy secretary ?
- **Context:** the ipcc receives funding through the ipcc trust fund , established in 1989 by the united nations environment programme (unep) and the world meteorological organization (wmo) , costs of the secretary and of housing the secretariat are provided by the wmo , while unep meets the cost of the depute secretary

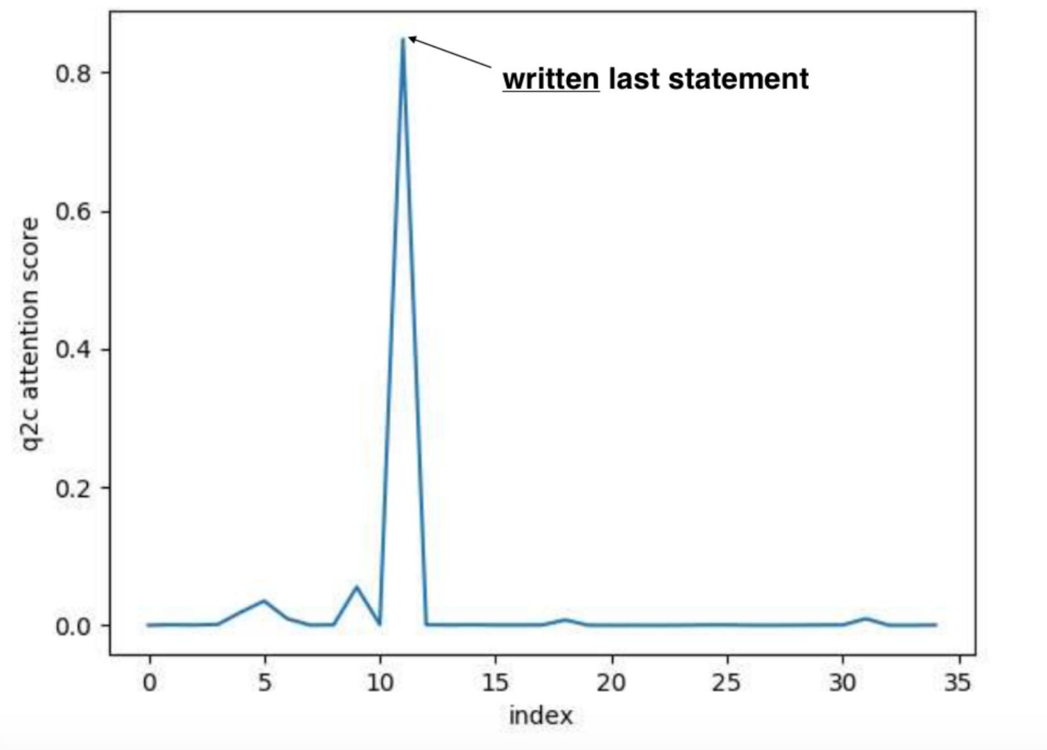


Figure 5: Question_to_context attention visualization

- **Prediction:** ipcc trust fund
- **Answer:** united nations environment programme

In this case, the model match "funds" in question to "receives funding" in context. This incorrect attention produces the wrong answer. In addition, it needs logic and reasoning, which requires knowledge not only in the scope of natural language processing.

Syntactic ambiguities

- **Question:** what was produced at tesla 's company ?
- **Context:** after leaving edison 's company tesla partnered with two businessmen in 1886 , robert lane and benjamin vail , who agreed to finance an electric lighting company in tesla 's name , tesla electric light and manufacturing . the company installed electrical arc light based illumination systems designed by tesla and also had designs for dynamo electric machine commutators , the first patents issued to tesla in the us .
- **Prediction:** illumination systems
- **Answer:** electric machine commutators

In this case, there is nothing wrong with the prediction. As the illumination systems are also designed and produced at tesla's company. However, human may more focus on the "electirc machine commutators", which is more important from a human's perspectives. We need more information or training data to help the model to learn that.

6 Conclusion and Future Work

In this work, we tackle the problem of reading comprehension on SQuAD dataset. We propose a model combining the well-known R-NET and BiDAF models for this task, leveraging their different attention mechanisms, *bi-direction attention flow* and *self matching attention*. Our experiments have shown promising results of our model, that we achieve 75.602% F1 and 64.730% EM on dev set, simply with a single model. We inspect the results through ablation study, attention analysis, and error analysis. Currently, we are not using engineering tricks to boost performance, while some of them have been shown helpful especially for the quantitative evaluation. In the future, we would like to explore such techniques for better performance. More importantly, we would love to better compare different attention mechanisms' functions, which is currently missing in our report due to time constraints. Such analysis would help us thoroughly understand the role of *attention* in natural language processing.

References

[1] Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: Squad: 100,000+ questions for machine comprehension of text. arXiv preprint arXiv:1606.05250 (2016)

[2] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems. (2017) 6000–6010

[3] Seo, M., Kembhavi, A., Farhadi, A., Hajishirzi, H.: Bidirectional attention flow for machine comprehension. arXiv preprint arXiv:1611.01603 (2016)

[4] Wang, W., Yang, N., Wei, F., Chang, B., Zhou, M.: Gated self-matching networks for reading comprehension and question answering. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Volume 1. (2017) 189–198

[5] Vinyals, O., Fortunato, M., Jaitly, N.: Pointer networks. In: Advances in Neural Information Processing Systems. (2015) 2692–2700

[6] Wang, S., Jiang, J.: Machine comprehension using match-lstm and answer pointer. arXiv preprint arXiv:1608.07905 (2016)

[7] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. (2013) 3111–3119

[8] Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). (2014) 1532–1543

[9] Kim, Y., Jernite, Y., Sontag, D., Rush, A.M.: Character-aware neural language models. In: AAAI. (2016) 2741–2749

[10] Santos, C.D., Zadrozny, B.: Learning character-level representations for part-of-speech tagging. In: Proceedings of the 31st International Conference on Machine Learning (ICML-14). (2014) 1818–1826

[11] Woolf, M.: char-embeddings. https://github.com/minimaxir/char-embeddings (2017) [Online].