SQuAD 2.0 QA with BiDAF++ & QANet

Stanford CS224N {Default} Project**, (IID SQuAD track)**

Mei Tan
Graduate School of Education
Stanford University
mxtan@stanford.edu

Raymond Zhang
Graduate School of Education
Stanford University
zhan1087@stanford.edu

Abstract

This project explores different implementations of a question answering system that performs well on the official SQuAD 2.0 dataset. To this end, implementations of the character-level embedding, additional input features, and transformer-based QANet architecture were tested. QANet is the best performing model with F1,EM scores of 64.098 and 60.49 respectively. Analysis on the predicted outputs of the different implementations revealed that models tend to perform better on "When" and "Who" questions then any other question type. The task of question answering is an key area in educational research thus the learning experience was the primary motivation for this project, as additionally we hope to apply our findings to relative educational domains in the future.

1 Key Information to include

- Mentor: Michihiro Yasunaga
- Late Days: 1 (1 from Mei and and 1 from Raymond)

2 Introduction

The task of question answering (QA) is an interesting and meaningful area of investigation not only because of the popularity in machine learning research communities but also in the potential of applying such systems across domains into the social sciences. Through this project we hope to gain an understanding of state-of-the-art neural network architectures, effects of different tuning parameters, model evaluation, and input feature decisions. Our ultimate goal is to be able to gain the competency to apply such techniques appropriately to the education domain.

The particular task of this project is to produce a QA system that works well on SQUaD 2.0. The human task we are aiming to replicate is that given a context paragraph and a question the person should be able to produce the answer or realize there is not an reasonable answer. Similar to evaluating if students authentically comprehended material being taught, understanding if the model truly "learned" or "understood" the text is nuanced. This makes the task of QA and machine comprehension to be a fruitful area of research and application if the goal is to imitate human learning. Furthermore, analysis of machine performance on middle school and high school on reading comprehension problems still shows a significant gap between humans and natural language processing (NLP) models[1].

The baseline model we compare against is the lookup based word embedding Bidirection Attention Flow Model (BiDAF) described in the project handout trained on the given SQuAD 2.0 dataset. The baseline model has low performance as indicated by its placement on the official SQUaD leader board. Our first improvement was to implement character-level embeddings replicating the original BiDAF model [2]. The addition of character-level embeddings showed an substantial improvement over the baseline in EM, F1, and AvNA score.

The second iteration of the model included the implementation of QANet[3] which utilizes Transformer architecture with convolution and self attention encoding blocks, and improvement through replacing the previous Recurrent Neural Networks (RNN) LSTM encoding layers. The QANet model further improved the EM, F1, and AvNA score, underscoring the power of Transformer-inspired architectures.

Finally, for our third iteration of the model, we implemented additional token features appended to the word and character embeddings for context words. This is relevant to educational domain as many students use contextual cues like parts-of-speech to help guide reading comprehension answers. While our first two model iterations were guided by the general direction of high performing models in NLP research, our final model was guided by applicability to the education domain. Through our three iterations, we approach this project with a focus on understanding how these models work on a technical level as well as how the models are applicable to help understand human learning.

3 Related Work

Attention[4] is an important contribution to the field of machine learning and since its introduction many models on the official SQUaD leader board utilize such mechanisms. The BiDAF model improves on such mechanisms by introducing a multi-stage hierarchical process that represents context using bidirectional attention flow to obtain a query-aware context representation without early summarizing [2]. Since then with the introduction of Transformers, models such as QANet [3] has again improved performance of QA by removing the RNN and replacing them with self-attention and convolutions. It does this by utilizing an Encoder Block which uses stacked convolutional sub-layers. The performance benefits including 3X+ times for training and 4X+ faster on inference with improved F1 score for the time. Such speed allows us to be able to experiment with bigger data set or augment existing data with similar training times. One such augmentation is the introduction of additional input features which is shown to improve F1 score in end-to-end systems such as DrQA [5]. Finally, utilizing pre-training and deep bidirectional representations, BERT [6] again out performs previous models but requires pre-trained weights which is restricted for pedological reasons in this project but a meaningful contribution to the machine learning community.

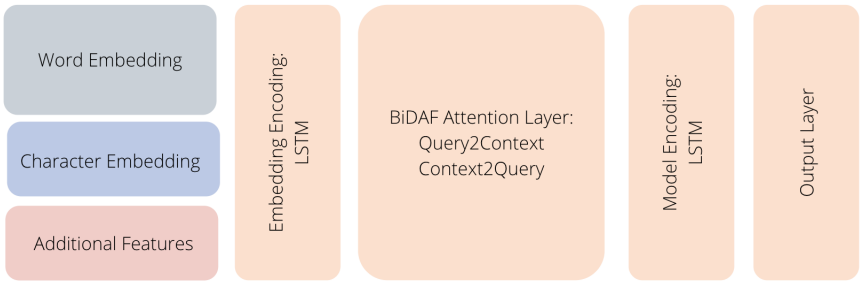
Additionally, there seems to be a gap between the domains of machine learning and learning sciences. QA has practical applications in the field of education, whether as a study tool[7] or to aid student work assessment[8]. However, while machine learning models now impact consumers through education technology, something is lost in the nuance of interpretation in applied settings as most machine learning studies do not directly work with students when assessing metrics or simulating results. On the other hand, principles from cognition and learning sciences could prove valuable to the advancement of QA systems. Domain knowledge in how people learn and strategically process information has the potential inform experiments in neural architecture. This can be a key area for further collaboration.

4 Approach

We implement and analyze the performance of three techniques to improve the baseline SQuAD system. We augment word vector representation in the embedding layer of the given BiDAF model with character-level embeddings added to context and question words, as well as additional token features added to context words (whether a word can be matched to a question word, part-of-speech tag, and entity type). We also implement the transformer-based QANet architecture with convolution and self attention encoding blocks.

4.1 BiDAF++

The project baseline is based on a BiDAF model, though missing a character-level embedding layer. We begin by extending the baseline model to match the original BiDAF model, adding character-embedding to the context and question words.



4.1.1 Character-level Embeddings

This layer maps each word to high-dimensional vector space using Convolutional Neural Networks (CNN) [2]. The characters are embedded into vectors, which are 1D inputs to the CNN [9]. The outputs are max-pooled over the entire width to obtain a fixed-size vector for each word. This approach allows us to condition on morphology and better handle out-of-vocabulary words.

4.1.2 Additional Input Features

We collect additional input features by using the *spaCy* library to tag linguistic features of each context word during pre-processing. These 22 binary feature vectors include the following attributes and are concatenated to the word and character-level vector representations in the embedding layer.

Exact Match Three features represent whether a context word can be exactly matched to a word in the question text, either in its original, lowercase or lemma form. These simple features have been shown to be boost performance significantly[5].

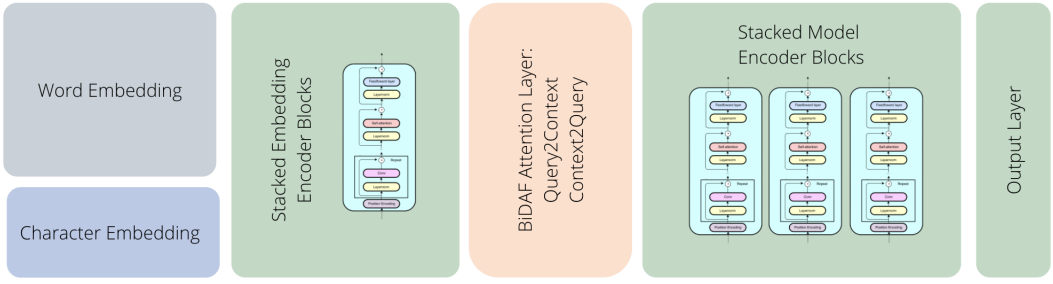
Part of Speech Seventeen features represent the part-of-speech property of a context word. Each feature is binary and corresponds with one part-of-speech.

Other Attributes Two features represent whether the context word is alphabetical (as opposed to numeric) and whether the context word is a stop word.

4.2 QANet

The QANet model incorporates transformer-architecture and replaces RNNs with self-attention and convolution[3]. The Encoder Block is the primary component of this model and consists of positional stacked convolutional sublayers using depthwise separable convolutions, a self-attention sublayer, and a feed-forward sublayer. Positional encoding[4], consisting of sin and cos functions at varying wavelengths, is applied to the input at the start of each encoder block. The encoder block uses layer normalization and residual connection between each layer.

We re-use the word and character-level embedding layer from the BiDAF model. The encoding layer consists of an encoder block each for the question and context, using 4 convolutional layers within each block and kernel size 5. Though the original QANet model uses DCN attention, we keep the BiDAF attention layer instead. Finally, three model encoders, each consisting of 7 blocks, feed into the task-specific output layer.



5 Experiments

5.1 Data

We use the official SQuAD 2.0 dataset with the predetermined splits. The data consists of (context, question, answer) triples, with the specific goal to predict an answer span given the input question and context. We analyze the characteristics of the train dev data splits and provide descriptive statistics.

- The train dataset has 43498 out of 130319 unanswerable questions
- The dev data set has 3168 out of 8277 unanswerable questions

	n	mean	sd	min	max	median
q_char_num	130319.0	58.5	73.8	1.0	25651.0	55.0
q_word_num	130319.0	10.1	3.5	1.0	40.0	10.0
context_word_num	130319.0	123.2	50.9	20.0	718.0	113.0
context_char_num	130319.0	756.1	308.2	151.0	3749.0	693.0
answer_text_word_num	130319.0	2.2	3.2	0.0	43.0	1.0

Table 1: Descriptive Statics of Training Data

	n	mean	sd	min	max	median
q_char_num	8277.0	59.5	21.8	13.0	182.0	56.0
q_word_num	8277.0	10.2	3.7	3.0	32.0	10.0
context_word_num	8277.0	138.0	68.8	26.0	636.0	120.0
context_char_num	8277.0	856.7	423.5	170.0	4065.0	742.0
answer_text_word_num	8277.0	2.4	3.6	0.0	30.0	1.0

Table 2: Descriptive Statics of Dev Data

Bigram	Count	Percentage
What is	716	8.65
What was	433	5.23
How many	309	3.73
What did	303	3.66
When did	275	3.32
In what	199	2.40
When was	176	2.13
What are	168	2.03
What does	166	2.01
Who was	154	1.86

Unigram	Count	Percentage
What	3811	46.04
Who	752	9.09
How	743	8.98
When	561	6.78
Where	338	4.08
In	328	3.96
Which	237	2.86
The	197	2.38
Why	160	1.93
By	44	0.53

Table 3: Top 10 Bigrams and Unigram of Dev-v2.0 Questions

5.2 Evaluation method

We use EM (Exact Match if system output catches ground truth) and F1 (harmonic mean of precision and recall) official SQuAD 2.0 evaluation metrics, as well as the AvNA (Answer vs. No Answer) metric visualized by TensorBoard.

5.3 Experimental details

Model	Epochs	Dropout	Lrn Rate	Batch	Hidden Size
Baseline	30	0.2	0.2	64	100
Char-embed	30	0.2	0.2	64	100
BiDAF + Inputs + Char-embed	30	0.2	0.2	64	100
BiDAF + Inputs	30	0.2	0.2	64	100
QANet	30	0.2	0.2	16	128

Due to memory limitations of the Azure VM, we trained the QANet model using a reduced batch size of 16.

5.4 Results

The official results are from non-PCE leaderboard, evaluated on the test split dataset. The EM,F1 scores are from dev split dataset leaderboards and the AvNA are reported from training results.

Model	EM	F1	AvNA	Official (F1, EM)
Baseline	58.461	61.696	67.57	
BiDAF + Char-embed	60.797	64.459	70.95	
BiDAF + Additional Inputs Features	59.57	62.788	70.21	62.173, 58.648
BiDAF + Char-embed + Additional Inputs Features	62.174	65.275	70.41	61.254, 58.022
QANet	**62.998**	**66.547**	**72.64**	**64.098, 60.49**

Despite the restricted batch size limitation, QANet is the best performing model and the results affirm the strength of its transformer-inspired architecture. It should be noted, however, that the QANet model is heavily memory-intensive. Though the original paper credits its training speed, this is only achievable with powerful computational resources.

The addition of context word input features improved both the baseline and character-embedding BiDAF models. Compared to the 400 dimensions of word and character-level embeddings, the 22 additional input feature dimensions had a disproportionately significant impact on performance.

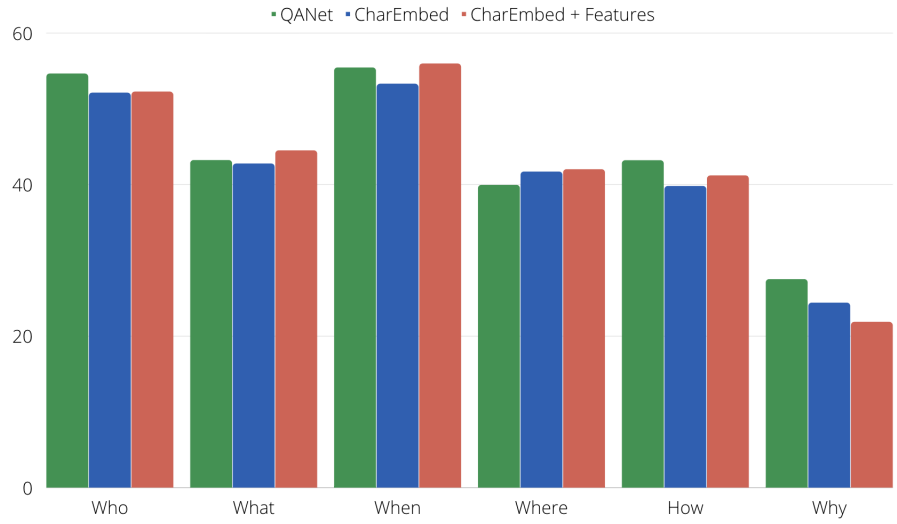
5.4.1 Fine Tuning

We experimented briefly with replacing the Adadelta optimizer with the Adam optimizer to take advantage of the momentum features. However, after 17 epochs our metrics did not improve and we abandoned the experiment due to time constraints. We also tried increasing the dropout rate to 0.5 to prevent over-fitting, but this experiment did not produce significant effects toward the results.

As fine tuning experiments were done in under time and budget constraints, the hyper-parameter search is limited. Given additional time, we would further explore robust fine tuning experiments upon our final model iteration.

6 Analysis

We analyze the results of the three best performing models by assessing the types of questions answered correctly and incorrectly. Building upon the "who-what-when-where-why" question-word unigrams of the question analysis explored in the data section, we calculate the percentage of correct answers that each model achieved on each type of question.



All models perform better on "who" and "when" questions and perform poorly on "why" questions. This result suggests that the models have greater accuracy for questions assessing declarative knowledge, but are ill-equipped to handle questions requiring more complex reasoning. Though QANet performs better overall on most question types, the contribution of the additional input features compared to only using word and character-level embeddings is interesting to note. Perhaps due to the features representing whether a word is alphabetical or numeric, this model has an advantage in answering "when" questions. The inclusion of part-of-speech features additional identifies proper nouns, which could improve the performance of "who" and "where" questions.

However, the additional input features that improve the model for "when-who-where" declarative knowledge assessments further decrease performance for "why" questions. The added focus on numbers and proper nouns, combined with distracting features such as the identification of stop words and determinants, could contribute to this finding. The chosen features further segregate the types of QA tasks into those with short factoid answers and those with longer explanatory answers.

Feature engineering introduces significant tradeoffs, better preparing a model to perform well on certain types of tasks at the expense of end-to-end learning. When the QA task is specific and well-understood, feature engineering can be a powerful performance enhancement, but may not be well-suited for more general tasks. Due to time constraints we tested a very limited set of input features, but we find that careful and theoretically-rooted feature selection may be fruitful for future exploration.

6.1 QANet Analysis

In this section we perform a deeper analysis on the results of our best performing model. First we examine the top 5 bigram/unigram (table 4 5)[1] of the questions using exact match[2]. We used a normalization score [3] to analyze if the a certain type of question performs better on our model. For correctly answered questions if the normalization score is greater then 1 that indicates that we are answering those questions correctly at a higher proportion then expected, if it is less then 1 it is answering worse then expected. Similar for incorrectly answered questions if the norm. score is greater then 1 the model answered those questions incorrectly more frequently then expected, while less then 1 means less frequent incorrectness then expected.

[1]This is percentage of the correct answers: N/(Total Number of correct answers) similar analysis with incorrect answers.

[2]This is using R and matching the answer of the dev-v2.0 triple with the predicted outcome of the dev split. We used the dev-v2.0 for analysis here. For some reason despite having a 60+ EM on the EM using R is only around 45. We speculate that it has to do with the conversion of files and screwing up the strings, and pre-processing of the model

[3]Calculated based on ((Count of Correct Question type)/(Total number of question in correct))/(percent of question type in dev set) this only works for samples of substantial size

Bigram	Percentage	Norm_score
What is	8.43	0.97
What was	4.82	0.92
How many	4.77	1.28
When did	3.85	1.16
When was	3.06	1.44

Unigram	Percentage	Norm_score
What	44.63	0.97
Who	11.14	1.23
How	8.70	0.97
When	8.43	1.24
In	4.12	1.04

Table 4: Top 5 Bigrams and Unigrams of Correctly Answered Questions

Bigram	Percentage	Norm_score
What is	8.94	1.03
What was	5.55	1.06
What did	3.98	1.09
How many	3.01	0.81
When did	2.99	0.90

Unigram	Percentage	Norm_score
What	47.14	1.02
How	9.37	1.04
Who	7.52	0.83
When	5.59	0.83
Where	4.57	1.12

Table 5: Top 5 Bigrams and Unigrams of Incorrectly Answered Questions

The model performs better on "Who" and "When" questions then expected with bigram question type of "How many", "When was", and "When did" performing better then expected, which matches with the scores for incorrect questions. Interestingly, the model seemed to perform better then expected on certain types of questions but does not perform worse then expected on certain types of questions.

Based on answer produced, QANet performs better on unanswerable questions then any other question type. Despite the fact that unanswer questions do not make up majority of the questions in the dev set, the answers predicted correctly are majority unanswerable questions. There seems to be a direct relationship between the complexity of the answer and the accuracy of the model. The which would make sense based on the type of questions the model was accurately predicting as "When did", "When was", "Who", and "When" questions probably resulted in answers that were shorter.

Ans Wrd Cnt	N	Percentage
0	2090	56.64
1	612	16.59
2	426	11.54
3	244	6.61
4	142	3.85

Table 6: Correct Answers

Ans Wrd Cnt	N	percentage
0	1016	23.00
2	724	16.39
1	713	16.14
3	550	12.45
4	316	7.15

Table 7: Wrong Answers

6.2 Comparing Norm Scores

bigram	featembed_norm_score	char_embed_norm_score	QA_norm_score
When was	1.49	1.45	1.44
How many	1.22	1.21	1.28
In what	1.17	1.39	1.22
When did	1.12	1.15	1.16
Who was	1.16	1.17	1.14
unigram	**featembed_norm_score**	**char_embed_norm_score**	**QA_norm_score**
When	1.26	1.23	1.24
Who	1.18	1.21	1.23
The	1.10	1.07	1.13
In	1.01	1.12	1.04
By	0.97	0.95	1.02

Table 8: Norm Scores of Most Common Bigrams/Unigram from Dev set

6.2.1 Examples of Predictions

Below is a wrong prediction. First reader must understand multiple parts of sentence to identify the question. Second the answer is in a different part of the context then the actual sequence that gives the correct information for identifying the answer meaning the model would need utilize multiple parts of the context in order to generate an answer. This type of question is complicated for human readers, and you would expect some test takers to answer this problem incorrectly which would would make it reasonable for our model to also incorrectly answer such questions.

Question: A forced trade agreement between two countries would be an example of what?
Context: The definition of imperialism has not been finalized for centuries and was confusedly seen to represent the policies of major powers, or simply, general-purpose aggressiveness. Further on, some writers[who?] used the term imperialism, in slightly more discriminating fashion, to mean all kinds of domination or control by a group of people over another. To clear out this confusion about the definition of imperialism one could speak of "formal" and "informal" imperialism, the first meaning physical control or "full-fledged colonial rule" while the second implied less direct rule though still containing perceivable kinds of dominance. Informal rule is generally less costly than taking over territories formally. This is because, with informal rule, the control is spread more subtly through technological superiority, enforcing land officials into large debts that cannot be repaid, ownership of private industries thus expanding the controlled area, or having countries agree to uneven trade agreements forcefully.
Answer:"informal" imperialism
Prediction:N/A

Below is an QA triple that is one of the most accurately predicted question types in our model. The question is straight forward in comprehension we are clearly looking for a number here. The answer and the contextual clue are right next to each other, with the contextual clue similar to the phrasing of the question. This type of question would be easy for both human readers and machines.

Question: How many combinatory and graph theoretical problems, formerly believed to be plagued by intractability, did Karp's paper address?
Context: In 1967, Manuel Blum developed an axiomatic complexity theory based on his axioms and proved an important result, the so-called, speed-up theorem. The field really began to flourish in 1971 when the US researcher Stephen Cook and, working independently, Leonid Levin in the USSR, proved that there exist practically relevant problems that are NP-complete. In 1972, Richard Karp took this idea a leap forward with his landmark paper, "Reducibility Among Combinatorial Problems", in which he showed that 21 diverse combinatorial and graph theoretical problems, each infamous for its computational intractability, are NP-complete.
Answer:21
Prediction:21

Finally, some of the performed human like for some questions. For the following the setup is similar to the above example however the question phrasing is complicated. The question requires understanding a dependency then understanding the question. It would be easily mistaken by many huamn test takers to produce the same answer as the model. This type of inaccuracy may be useful as it replicates a form a human learning and mistakes.

Question: Of Poland's inhabitants in 1901, what percentage was Catholic?
Context: Throughout its existence, Warsaw has been a multi-cultural city. According to the 1901 census, out of 711,988 inhabitants 56.2% were Catholics, 35.7% Jews, 5% Greek orthodox Christians and 2.8% Protestants. Eight years later, in 1909, there were 281,754 Jews (36.9%), 18,189 Protestants (2.4%) and 2,818 Mariavites (0.4%). This led to construction of hundreds of places of religious worship in all parts of the town. Most of them were destroyed in the aftermath of the Warsaw Uprising of 1944. After the war, the new communist authorities of Poland discouraged church construction and only a small number were rebuilt.
Answer:N/A
Prediction:56.2%

Examination of the examples show that the our model is able to represent some forms of human reading comprehension. Questions that require more cognitive resources for students resulting in more errors, resulted in the model producing similar types of errors. Applications of such systems could be experimenting with of various learning tasks to understand biases and issues in reading based assessments without the use of human subjects, or guide which experiments are fruitful endeavors dramatically reducing the cost/time of random controlled experiments.

7 Conclusion

Through this project, we learned several model architectures and gained practice with neural network concepts. Understanding the BiDAF model, learning to implement character-level embeddings, and understanding and implementing the QANet model provided valuable experience in replicating successful QA model techniques while exercising the knowledge we have learned throughout the quarter in CS224n. The QANet model was overall the best performing of those we implemented, but including additional input features to represent words more expressively was surprisingly effective as well. Through implementing the inclusion of additional input features, we started to see the potential for the application of learning science concepts for further improvements. In future work we could explore the effect of priming, prior knowledge, and process-of-elimination in machine QA systems.

References

[1] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[2] Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *CoRR*, abs/1611.01603, 2016.

[3] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. Qanet: Combining local convolution with global self-attention for reading comprehension. *CoRR*, abs/1804.09541, 2018.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[5] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions. *CoRR*, abs/1704.00051, 2017.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[7] Xinran Zhu, Bodong Chen, Rukmini Avadhanam, Hong Shui, and Raymond Zhang. Reading and connecting: Using social annotation in online classes, May 2020.

[8] Jill Burstein. Opportunities for natural language processing research in education. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, pages 6–27, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[9] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics.