Question Answering with Contextual Embeddings, Data Augmentation, and Bi-directional Attentional Flow on SQuAD 2.0 Dataset

Dhayarkar, Rishikesh
rbd291@nyu.edu

Kasireddy, Durga Prasad Reddy
dpk290@nyu.edu

Abstract

This project aims to implement a robust approach in solving the task of Question Answering(QA) on SQuAD 2.0 dataset. We utilize various techniques such a Pre-trained Contextual Embeddings(PCE), Data Augmentation, and, merging PCE models with classical QA models such as BiDAF to achieve this goal. First we augmented the training data by randomly substituting words with their synonyms. Then we use GPT-2 to generate new sentences to augment the context paragraphs in a coherent fashion. We experimented with BiDAF model by replacing GLOVE embeddings with BERT embeddings. Our best model with data augmentation on a fine tuned RoBERTa model gave a Dev EM/F1 of **80.46/82.61**. By further ensembling multiple models and architectures such as RoBERTa, BERT, ALBERT, DistilBERT, and BiDAF models, we were able to achive a top EM/F1 of **81.13/84.06**.
Github Link - `https://github.com/RishikeshDhayarkar/CS-GY-9223-Deep-Learning`

1 Introduction

Question Answering (QA) is one of the most important natural language processing tasks which has attracted numerous work and challenges on building effective and robust QA systems. QA systems can be used in many practical applications such as virtual assistants, improving search engines, and automated customer service. Question Answering (QA) is a challenging task for machines, as it requires both understanding of natural language and knowledge about the world.

QA has proved difficult for a variety of reasons. First, QA algorithms are often reliant on having corpora that contain verbatim of the actual answer. Because of this a critical component of any QA model is selecting documents which might contain the answer. One such dataset that provides questions with their answers is the SQuAD 2.0 dataset[16].

This dataset consists of a paragraph(context), and a question about that paragraph. The training data is a dictionary consisting of 442 ”titles” for example ”Beyonce” is one of the442 ”titles” and there are 66 ”context paragraphs” under ”Beyonce” title. For each ”context paragraph” there are about 2 - 10 question and answers. There are 29,035 contexts and130,000 such question and answer pairs and about half of these questions do not have answers in the respective ”context paragraph”. For such questions the answers will be ”no answer”. For the questions which have answers, the model just needs to select the span of text in the paragraph that answers the question.

It need not generate the text on its own. Think ofthis as a highlighter we use in a textbook.

We aim to implement a robust approach in solving the task of Question Answering(QA) on SQuAD 2.0 dataset. We present different methods that can be used to boost the performance of current SOTA QA models. We utilize various techniques such a Pre-trained Contextual Embeddings(PCE), Data Augmentation, and, merging PCE models with classical QA models such as BiDAF to achieve this goal. The final model that we present is an ensemble for multiple models which beats the individual performance of each of it components.

2 Literature survey

In the original SQuAD 1.0 paper [1], the baseline model used logistic regression on a large set of 180 million lexicalized features and dependency tree path features. Its EM/F1 score of **40.0/51.0** was significantly below the human baseline of **80.3/90.5**. In the subsequent SQuAD 2.0 paper [2], the authors employed a stronger baseline "DocQA No-Answer with ELMo" [3], which mostly closed the Stanford CS224N Natural Language Processing with Deep Learning human-machine gap with its score of **78.6/85.8**. However, on the more challenging 2.0 dataset, the DocQA only achieved a EM/F1 of **63.4/66.3**, well below the human baseline of **86.9/89.5**.

Before PCE took over the world of NLP, the task of question answering was dominated by attention flow based models such as BiDAF[4]. on SQuAD 1.0 BiDAF managed to achieve an EM/F1 of **73.74/81.52**. However on SQuAD 2.0 BiDAF exhibited a weak performance achieving an EM/F1 of only **57.5/61.0**.

In late 2018, BERT [7] was introduced and achieved SOTA result on SQuAD 2.0 among 11 other downstream NLP tasks. The key idea was pre-training the contextual embeddings on two tasks—Masked Language Model (MLM) and Next Sentence Prediction (NSP)—using a deep Transformer architecture and a massive corpus (Wikipedia 2.5B + BookCorpus), before fine-tuning the final layer to compute the start and end token probabilities for SQuAD. BERT is designed to pre-train deep bidirectional representations by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with minimal architectural change and achieve state-of-the-art performance.

With the release of BERT a series of PCE models have been implemented and published.Models such as ALBERT, RoBERTa, XLNet etc. The internal tasks used to train the model varied across all the models. A deep dive of a few models is done in section 3.

3 Approach

The first step of our approach was to fine tune all the top models that are good at question answering. From our literature survey we found that BERT[7], ALBERT[6], RoBERTa[8], and DistilBERT[9] performed well on the question answering tasks, specifically on SQuAD 2.0. We experimented with 'base' version of all these models and found that RoBERTa has the highest EM/F1 of **79.79/82.03**. ALBERT achieved an EM/F1 of **78.63/81.79**, BERT achieved **70.37/73.81**, DistillBERT achieved **65.08/68.18**. Among these PCE models we decided to use BERT, ALBERT, and, RoBERTa for further experimentation. We also tested the baseline Bidirectional Attention Flow model provided by Stanford[17] on SQuAD 2.0. This gave an EM/F1 of **56.39/59.76**. In this project we try to combine the classical way(BiDAF) of handling the QA task with modern approaches that are based on PCE.

Then, using RoBERTa as a baseline model, we proceed with our own extensions to build a robust model. We first augmented SQuAD 2.0 dataset by randomly substituting synonyms. Second, we used GPT-2 to generate new context paragraphs that are coherent with the existing paragraphs. Third, we replaced the GLOVE embedding layer of the BiDAF model with BERT embeddings. Finally, we created different ensemble configurations using the `nbest-predictions` file to achieve the best possible model.

Only 'base' PCE models were used in this project. All the approaches presented here can be directly translated to 'xlarge' and 'xxlarge' PCE models.

3.1 BiDAF: Bidirectional Attention Flow Model

The core idea behind this model is that attention should flow both ways(context to question and question to context). Each word is mapped to a vector space using a pre-trained word embedding model (GLoVE) to get fixed length vector. These two embeddings are passed to a Highway layer which gives 2 matrices, one for Context, one for queries. In the Phrase embed layer, we use a LSTM on top of the embeddings provided by the previous layers to model the temporal interactions between words. We place an LSTM in both directions, and concatenate the outputs of the two LSTMs. Now, we have our features converted to matrices/vectors. In attention flow layer, we combine the query and context vectors and produces a set of query-aware feature vectors for each word in the context. In next layers,we build an RNN using LSTMs and an output layer to provide answers.

The output is answer text and the phrase is derived by predicting the start and the end indices of the phrase in the paragraph. We obtain the probability distribution of the start, end index over the entire paragraph by using a softmax layer, which takes the query-aware representation and contextual embedding matrix. And use log likelihood on these two probability distributions

3.2 Experimenting with pre-trained contextual embeddings

3.2.1 BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

We use BERT [7], a current state of the art language representation system to generate word embeddings. We take the output from the hidden states to generate new embeddings for each text input. These embeddings are then fed into the remaining parts of our modified BiDAF model. BERT Base model consists of 12 encoder layers, where the output of each encoder layer along each token's path can be used as a feature representing that token. These transformers are pre-trained on masking and next sentence prediction objectives, enabling them to learn and encode bidirectional meaning in ways previous state of the art approaches could not.

3.2.2 ALBERT: A Lite BERT for Self-supervised Learning of Language Representations

ALBERT[6] used two parameter reduction techniques to achieve better results than BERT while having fewer parameters. First, "factorized embedding parameterization" decomposed the large vocabulary embedding matrix into two small matrices, which makes it easier to grow the hidden size without increasing the parameter size of the vocabulary embeddings. Second, "cross-layer parameter sharing" prevented the parameter from growing with the depth of the network. The default decision for ALBERT is to share all parameters across layers. It also replaced BERT's NSP task with a new auxiliary task "sentence-order prediction" (SOP). By using the positive examples in the same way as BERT, but swapping the orders of the negative examples, the SOP loss avoided the easier task of topic prediction and instead focused on modeling inter-sentence coherence.

3.2.3 RoBERTa: A Robustly Optimized BERT Pretraining Approach.

RoBERTa[8] found that BERT was significantly under-trained, and the same MLM objective can achieve significantly better result with longer training (500K steps vs. BERT's 100K steps) and on a larger dataset (160GB vs. BERT's 16GB). It also offered enhancement in dynamic masking as RoBERTa will generate the masking pattern every time the sequence was fed to the model, while masking was done only once during data preprocessing in BERT. Similar to ALBERT, RoBERTa also removed the NSP task and found that it improved downstream task performance. RoBERTa also optimized some of the hyper-parameters, notably increasing the batch size from BERT's 2K to 8K as this improved the LM's perplexity while making the training easier to parallelize. Lastly, for SQuAD 2.0, RoBERTa utilized an additional binary classifier to predict whether the question is answerable, which was trained jointly by summing the classification and span loss terms.

3.3 Original Work

3.3.1 Data Augmentation with Synonym replacement

State of the Art NLP models produce really high scores on most of the datasets. Thus, it is really important to provide even more diversified dataset for the model to learn the features and improve it's performance. Through data augmentation, we try to increase the number of words used in the dataset and also teach the model word similarities. We particularly focused on word replacement in a context paragraph. By replacing words, we not only change the words in the context, we also change the answer span for the questions corresponding to a particular context. Questions are left unchanged.

After trying many hyperparameters, we decided to use the following logic. For each word in the context which is neither a stop word nor a noun, with probability 0.2, we replace it with one of it's synonym (if exists). Else, we leave it unchanged. This introduces new words to the model and yet preserve most of the original meaning of the context. Examples of such simulation are provided in the [Appendix]. We perform this task using NLTK, Wordnet libraries. One simulation of this algorithm on the whole dataset results in a new dataset of the same size. We augment this new dataset to the original dataset creating double the number of samples. We define "replication-factor" which signifies how many datasets we are augmenting (1 in this example). Due to economic constraints, we could not use replication-factor more than 2 (more than thrice the size), but we can see promising results in using such approach.

By using this technique of data augmentation we were able to increase the EM/F1 score by **+0.29/0.35** on the RoBERTa baseline scores.

3.3.2 Data Augmentation with context extensions using GPT-2

Natural Language generation(NLG) is often using in NLP applications and projects to enhance the quality and size of the datasets. With SOTA NLG models from the GPT series, NLG has become is a commonplace in NLP literature. We noticed that the length of the context paragraphs in SQuAD

2.0 were about 135, which is fairly short. To increase the length of these context paragraphs we decided to use GPT-2 to generate new sentences based on the current context and append them to the current context sentences. To achieve this we used the 'Simple transformers library'[11], a package that operates on top of the Huggingface library[10]. We experimented with augmentations of various lengths and found that an augmentation of 256 words gave the best results. We were able to increase the score by **+0.38/+0.30** over the RoBERTa baseline EM/F1. Examples of context extension are provided in the Appendix section.

3.3.3 Experimenting with BERT on BiDAF

The landscape of NLP has been shifted radically since the advent of PCE models such as BERT and ALBERT. The embeddings generated by these models are known to capture robust meanings and contexts from the sentences. Before PCE models were introduced to the world of NLP, attention flow models such as BiDAF were the best models for QA task. An obvious way to augment the BiDAF model is to replace the GLOVE embedding in the initial part of the network with an embedding generated from a PCE model like BERT.

BERT could also be used as a word-level embedding method with bidirectional contextual information. The input to the original BiDAF model was a matrix of GLOVE embeddings with dimensions of [max sequnce length, 300]. This was replaced with with a matrix of size [max sequnce length, 768], where 768 is BERT's transformer's hidden layer size. A major problem that we faced during the implementation of this portion of the project was performing suitable tokenization prepossessing. GLOVE uses a word based tokenizer to convert words to numbers whereas BERT uses a WordPiece tokenizer. A word word piece tokenizer assigns a token to the different snippets of a word. For example the word 'calligraphy' gets split into three tokens, 'call', '##ig', '##raphy'. A snippet with '##' symbol get attached to the snippet before it. Because of these word pieces the length of the BERT embeddings was longer than GLOVE embeddings. We explore two ways of exploring this problem. One is to use these extended embeddings as it is or to compress them to the size of GLOVE embeddings by averaging the embedding values of the snippets of a word. We found that the later approach gave better results. Also, during our training process BERT's weight were not altered.

Once the final BERT embeddings were obtained they were passed through a highway network, same as the original BiDAF. We trained the baseline BiDAF model and obtained an EM/F1 score of **56.39/59.76**. We were able to infer from our literature survey that replacing GLOVE by BERT increased EM/F1 by approximately 4 to 5 points. Our enchanced BiDAF model with BERT embed-dings obtained an EM/F1 score of **60.1/63.8**.

3.3.4 Ensemble models

Ensemble is a proven method for improving the robustness and accuracy of most predictive models. In particular, we saw in Section 3.1 that ALBERT / BERT / RoBERTa have different LM objectives (e.g. masking vs.permutation), auxiliary tasks (e.g. SOP), varying hyper-parameters and training data, and different approach to handling unanswerable questions in SQuAD 2.0. Thus, we proposed to combine their `nbest-predictions` via majority voting.

We created two ensemble groups. The first one included only the PCE models, ALBERT, BERT and RoBERTa. The second group included these three PCE models along with one BiDAF model. We hypothesised that including a non-PCE model like BiDAF could somehow augment the learning and predictions of PCE models. Detailed results are available in the experiments section.

4 Experiments

4.1 Data

For our experiments we used the official SQuAD 2.0 dataset and the default training and dev set splits. Both of the data augmentation techniques were applied on this dataset.

4.2 Evaluation method

We use the two standard metrics of Exact Match (EM) and F1 Score (F1) for SQuAD. EM measures the percentage of predictions that match any one of the ground truths exactly. F1 measures the average overlap between the prediction and the ground truth, using a harmonic mean of precision and recall, where each prediction/answer pair is treated as a bag of tokens. The maximum F1 score is taken for questions with multiple answers and the final score is the average F1 across all questions.

4.3 Experimental details

4.3.1 Code and Infrastructure

We used Pytorch as our main Deep Learning library. All BiDAF related work was based the Pytorch implementation of word level BiDAF provided by Stanford. Work related to PCE models was performed using the Huggingface transformers library[10]. The code for ensemble models also utilized some snippets from the Huggingface transformers library. Synonym replacement was done using NLTK, Wordnet packages. Data augmentation using GPT-2 was performed using the Simple Transformers library.

Experiments were performed on our local machines. Google Colab Pro was used for all experiments. The GPUs included in this infrastructure are T4 and P100 with 16GB of memory. The CPUs come with 25GB of RAM and 250GB of disk space. We used AWS(p3.8xlarge) to generate the results for deliverable 3, which enabled us to use 'xxlarge' versions of the models. Due to the high number of experimental trials involved in our project AWS was not an economical option. We had to resort to cheaper alternatives like Colab Pro.

4.3.2 Hyper-Parameter Tuning

We conducted tests on various combinations of hyper parameters and found the below configuration for each of the models to be optimal. The table below summarises the optimal configurations for PCE models. We were not able to experiment a lot with the batch size parameter because of memory restrictions on our infrastructure. Bigger batch sizes might provide convergence of a better quality. Doc stride of 128 was more suitable when the models were running on datasets that were not augmented with GPT-2. Using GPT-2 increases the length of the contexts by two folds. To enable the model learn from longer contexts we found a Doc stride of 300 to be more suitable. For BiDAF models we used the default parameters provided by Stanford in their baseline BiDAF model.

Hyper parameter tuning for PCE models(Table 1)				
Models	BERT	ALBERT	RoBERTa	DistilBERT
Batch size	12	12-4	12-14	12-14
Learning rate	4e-5	3e-5	2.5e-5	3e-5
Epochs	3	2	2	3
Max Sequence	384	512	512	384
Doc stride	128/300	128/300	128/300	128/300

4.4 Results

The type of ensembling technique we used was majority voting. We created two ensembles, one with BiDAF and one without BiDAF. The model without BiDAF achieved a higher EM/F1 scores. We experimented with two types of ensembles. The first one

4.4.1 Overall comparison

Comparison of various models on Dev scores(Table 2)			
Models	Remarks	Dev EM	Dev F1
Human	-	86.9	89.5
BiDAF	-	56.39	59.76
BiDAF + BERT	base version of BERT	60.1	63.8
DistilBERT	base model	65.08	68.18
BERT	base model	70.37	73.81
ALBERT	base model	78.63	81.11
RoBERTa	baseline model	79.79	81.89
RoBERTa	with synonym replacement	80.08	82.24
RoBERTa	with GPT-2	80.17	82.19
RoBERTa	synonym replacement & GPT-2	80.46	82.61
Ensemble	ALBERT + BERT + RoBERTa + DistilBERT + BiDAF	80.35	82.93
Best Ensemble	ALBERT + BERT + RoBERTa + DistilBERT	**81.13**	**84.06**

4.4.2 Bert vs Albert vs Roberta vs Our Model

We further breakdown the EM/F1 scores of the different models by their performance on answerable and unanswerable questions in Table 3. While it is clear that 'Best Ensemble' has the best performance across No ans EM, No ans F1, Best EM, and, Best F1 categories. RoBERTa performed better than this ensemble Has ans EM and Has ans F1 by almost 2 points.

Breakdown of performance various models on Dev scores(Table 3)					
Metrics	BERT	DistilBERT	ALBERT	RoBERTa	Best Ensemble
Has ans EM	68.03	64.18	75.13	**77.39**	75.44
Has ans F1	74.91	70.38	81.47	**81.78**	80.83
No ans EM	72.71	65.98	80.11	81.18	**87.95**
No ans F1	72.71	65.98	80.11	81.18	**87.95**
Best EM	70.37	65.08	78.63	79.79	**81.13**
Best F1	73.81	68.18	81.11	81.89	**84.06**

5 Conclusion

In this project we were able to implement a robust QA system that achieved a top EM/F1 of **81.13/84.06**, which is an increment of **+1.34/2.17** on EM/F1. To achieve this boost in performance we first performed data augmentation via synonym replacement and context extension using GPT-2. Then, we explored the BiDAF architecture and improved its baseline performance by an EM/F1 of **+3.51/4.04**. This was done by replaceing GLOVE embeddings in the baseline BiDAF by BERT embeddings. We expected a classical QA model like BiDAF to augment the performance of a PCE models in the form of a robust ensemble, but this was not the case. The ensemble without BiDAF performed better than the one with BiDAF. The ensemble without BiDAF beats the ensemble with BiDAF by an EM/F1 of **0.78/0.32**.

Our project can be improved in many ways. All the experimentation done in this project was done on 'base' versions of the PCE models. By replacing the 'base' versions by 'xlarge' and 'xxlarge' versions of the models we expect a significant boost in the EM and F1 scores. For example, from our literature survey we found out that the base version of ALBERT gives an EM/F1 of **78.63/81.11** whereas, the 'xxlarge version gives an EM/F1 of **86.3/88.9**. By implementing the methods from this project on 'xlarge' and 'xxlarge' models, the final scores of the ensembles can definitely beat human performance on SQuAD 2.0.

Other ways to increase the performance would be using additional datasets such as TriviaQA and NewsQA. Including models such as XLNet, ELECTRA, XLM, and SpanBERT could boost the performance by a good amount. We actually tried to incorporate XLNet in our project but because of the ongoing issues and rapid changes in the code base of Huggingface transformers, we were not able to include it in our ensemble. Apart from this usage of sophisticated ensembling techniques such as stacking ensemble could improve the scores by a reasonable margin.

Appendix

Examples of Synonym Replacement

The words striken off are in the original context paragraph and the words highlighted in blue are the words the algorithm has picked to replace with original words. Here, we show an example of successful augmentation and unsuccessful yet useful augmentation. We are accustomed to using few words and phrases and any deviation of that wouldn't seem like a coherent sentence. The unsuccessful augmentation is an example of that situation.

Successful Augmentation Example

Beyoncé Giselle Knowles-Carter (/bijnse/ bee-YON-say) (born sept 4, 1981) is an American singer, songwriter, record producer and actress. Born and ~~raised~~ brought up in Houston, Texas, she performed in various singing and dancing competitor as a ~~child~~ tyke, and ~~rose~~ upsurge to fame in the late 1990s as ~~lead~~ chair singer of RB girl-group Destiny's ~~Child~~ Nipper. ~~Managed~~ Handled by her father, Mathew Knowles, the group ~~became~~ turned into one of the world's best-selling girl groups of all time.

Unsuccessful Augmentation Example

Beyoncé Giselle Knowles-Carter (/bijnse/ bee-YON-say) (born September 4, 1981) is an American singer, ~~songwriter~~ songster, record producer and actress. Born and raised in Houston, Texas, she ~~performed~~ do in various ~~singing~~ tattle and dancing ~~competitions~~ rival as a ~~child~~ youngster, and rose to celebrity in the recently 1990s as lead singer of RB girl-group ~~Destiny~~ lot's Child. Managed by her father, Mathew Knowles, the group became ~~one~~ one and only of the ~~world's~~ earth's best-selling girl groups of all time.

Examples of context extension using GPT-2

The text in black is the original context information from the dataset. The sentences in blue are the extensions added by the GPT-2 model. It is evident from the examples that the model is doing a good job of understanding the context and appending new coherent sentences to it.

Example 1

Beyoncé Giselle Knowles-Carter (/bijnse/ bee-YON-say) (born September 4, 1981) is an American singer, songwriter, record producer and actress. Born and raised in Houston, Texas, she performed in various singing and dancing competitions as a child, and rose to fame in the late 1990s as lead singer of RB girl-group Destiny's Child. Managed by her father, Mathew Knowles, the group became one of the world's best-selling girl groups of all time. In 2010, Beyoncé released the "Beyoncé: Life in the Moment" single, and later in 2011, she released the "The Body I Am" single and the single "Crazy In Love". With her new hit "This Is Love" and "The Body I Am," Knowles is considered the most vocal member of the group. Prior to this release, the group was based at Little Mix.2012, she released her latest single, "Get Lucky", a remix of Beyoncés song "I Cant Wait to Go Home."

Example 2

Frédéric François Chopin (/opæn/; French pronunciation: ̌200b[fe.de.ik f.swa .p]; 22 February or 1 March 1810 – 17 October 1849), born Fryderyk Franciszek Chopin,[n 1] was a Polish and French (by citizenship and birth of father) composer and a virtuoso pianist of the Romantic era, who wrote primarily for the solo piano. He gained and has maintained renown worldwide as one of the leading musicians of his era, whose "poetic genius was based on a professional technique that was without equal in his generation." Chopin was born in what was then the Duchy of Warsaw, and grew up in Warsaw, which after 1815 became part of Congress Poland. Chopinś father was a physician, and his mother is a naturalist.[8]1855, he received the piano at the age of seven. At school, Chopin received the instrument, but was unable to write for the piano because his father had been unable to read his poems, and he was too young to begin composing them.

6 References

1. Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ Questions for Machine Comprehension of Text. arXiv:1606.05250, 2016.
`https://arxiv.org/abs/1606.05250`

2. Pranav Rajpurkar, Robin Jia, and Percy Liang. Know What You Don't Know: Unanswerable Questions for SQuAD. arXiv:1806.03822, 2018.
`https://arxiv.org/abs/1806.03822`

3. Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep Contextualized Word Representations. arXiv:1802.05365, 2018.
`https://arxiv.org/abs/1802.05365`

4. Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, Hannaneh Hajishirzi. Bidirectional Attention Flow for Machine Comprehension, arXiv:1611.01603, 2018.
`https://arxiv.org/abs/1611.01603`

5. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pretraining of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805v2, 2018.
`https://arxiv.org/abs/1810.04805`

6. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. arXiv:1909.11942 , 2019.
`https://arxiv.org/abs/1909.11942`

7. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805, 2018.
`https://arxiv.org/abs/1810.04805`

8. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692, 2019.
`https://arxiv.org/abs/1907.11692`

9. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv:1910.01108, 2019.
`https://arxiv.org/abs/1910.01108`

10. Huggingface transformers library.
`https://github.com/huggingface/transformers`

11. Simple transformers library.
`https://simpletransformers.ai/`

12. Avengers: Achieving Superhuman Performance for Question Answering on SQuAD 2.0 Using Multiple Data Augmentations, Randomized Mini-Batch Training and Architecture Ensembling.
`https://web.stanford.edu/class/cs224n/reports/default/report15.pdf`

13. SQuAD 2.0 Based on ALBERT and Ensemble.
`https://web.stanford.edu/class/cs224n//reports/default/report08.pdf`

14. Enhancing BiDAF with BERT Embeddings, and Exploring Real-World Data.
`https://web.stanford.edu/class/cs224n/reports/default/15791074.pdf`

15. Really Paying Attention: A BERT+BiDAF Ensemble Model for Question-Answering.
`https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1194/reports/default/15792214.pdf`

16. The Stanford Question Answering Dataset.
`https://rajpurkar.github.io/SQuAD-explorer/`

17. Word level implementation of BiDAF from Stanford University.
`https://github.com/chrischute/squad`