Quotation Detection on RiQuA Dataset

Team Members:

- Alex Feng; Email: ahfeng@seas.upenn.edu
- Renu Reddy Kasala; Email: renu1@seas.upenn.edu
- Anisha Singrodia; Email: singroa@seas.upenn.edu

home pod: gargantuan-tortoise

Abstract

Quotation detection is an important processing step in a variety of applications in machine learning, including generative applications like audiobook or script generation and classification applications such as automated fact checking. Quotation detection can be challenging since most words are not significantly more likely to be inside or outside of a quotation, so a quotation detection model needs to have some understanding of the context surrounding words. In this project, we experiment with various models for quotation detection, focusing on methods that do not require excessive feature engineering. We evaluate a simple most common tag baseline, a number of supervised models trained on BERT [3] embeddings, and a fine-tuned BERT model, and we find that BERT is able to embed the context around each token to significantly improve performance over the baseline.

1 Motivation

Identifying quotations and attributing them to speakers has various applications, including assigning appropriate voices to characters in audiobook generation and creating scripts based on novels [2]. Quotations and the context surrounding them can provide pertinent information about actors and their interactions, and direct quotations, especially when paired with contextual information, can be a valuable source of training data for expressive speech synthesis.

2 Related Work

In their paper "A Neural-Network-Based Approach to Identifying Speakers in Novels" [2], Chen et al. formulate speaker identification as a scoring task and build a network based on BERT [3] to score candidates. They define an instance as a sequence of sentences, consisting of a quote to be attributed and context before and after the quote. Their approach also takes a list of names and aliases of characters and assumes that at least one alias of the true speaker appears in the context. The model first finds the nearest mention of an alias of each candidate to the quote. Then, for each such candidate, the sentences containing the mention and the quote, and all sentences in between, are used as input to BERT to obtain representations of the candidate, context, and quote sentence. These representations are then scored by a Multi-Layer Perceptron for each candidate. The authors also propose an algorithm to detect and revise two-party conversations, which the model struggles with due to the overlap in context among densely packed quotes. Using this approach, they achieve state-of-the-art performance, significantly improving over the baseline using manual features.

In their paper "Quotation Detection and Classification with a Corpus-Agnostic Model" [4], Papay and Padó implement a corpus-agnostic neural model for quotation detection and evaluate it on three corpora

(PARC3, STOP, RWG) that vary in language, text genre, and structural assumptions. The model shows reasonable performance while operating on corpora substantially differing in form of text genre, annotation scheme, and theoretical assumptions. They have deployed a neural architecture Neural Quotation Detection (NQD), with the goal of modeling the quotations in all three corpora: The Penn Attribution Relation(PARC) Corpus, Speech, Thought, and Writing Presentation corpus (STOP), and The Redewiedergabe ('reported speech') corpus (RWG). NQD frames quotation prediction as token classification problem, classifying each token as either beginning a quotation (BEGIN), ending a quotation (END), or neither (NEITHER). The model architecture comprises a 2-layer bi-LSTM network, with the outputs of the second bi-LSTM feeding into a 3-class softmax classifier. Thus, the model takes token sequences as input and produces a sequence of token labels. As the corpora does not contain test sets, they used 10-fold cross validation to evaluate our model, using 8 folds for training, 1 for development, as 1 for testing. To compare model with state-of-the-art, precision, recall, and F1 in this setting for PARC3, STOP and RWG had been reported. The results on the three corpora could not out-perform the state-of-the-art, but approximates it closely despite the lack of corpus-specific tuning.

Papay and Padó went on to release their own dataset "RiQuA: A Corpus of Rich Quotation Annotation for English Literary Text" [5], which consists of excerpts from 11 19th century English literary works, annotated with information on speaker, addressee, and cue word, if present. The texts are annotated using Brat Rapid Annotation Tool to mark spans that contain a quotation, speaker, addressee, or cue word, along with relation information to link quotations with the corresponding context. The texts were annotated independently by two separate annotators and then merged by a third, more senior annotator. The corpus contains a total of 5963 quotations, most of which are direct. The vast majority of quotations have marked spearkers.

3 Data Set

RiQuA, Rich Quotation Annotations is a corpus that provides quotations for English literary text. RiQuA dataset consists of 15 excerpts from 11 works of 19th century English novels by 6 authors annotated with quotation, entity, and cue word spans as well as corresponding relationships. Some of the works include Jane Austen's "Emma", Anton Chekhov's "The Lady with the Dog" and Mark Twain's "The Adventures of Tom Sawyer".

We use the RiQuA [5] dataset, which consists of:

- Percentage of quotation labelled words in the entire text: 33.94 %
- Percentage of entity labelled words in the entire text: 1.67 %
- Percentage of cue labelled words in the entire text: 0.89 %

We can see that it is a very sparse dataset where 63.5% of text is not annotated as either Quotation, Cue or Entity.

To tokenize using BERT tokenizer we need to convert our text in BERT supported input - CONLL format :

- Assigned each token a tag denoting whether it is the beginning of a span, inside a span, or outside to get the supported format. Like shown in Figure 1, we formatted our data such that the beginning and intermediate quotation words are marked with prefix B- and I- respectively. Similarly the cue and entity words are also converted. The unlabelled words are annotated as O.

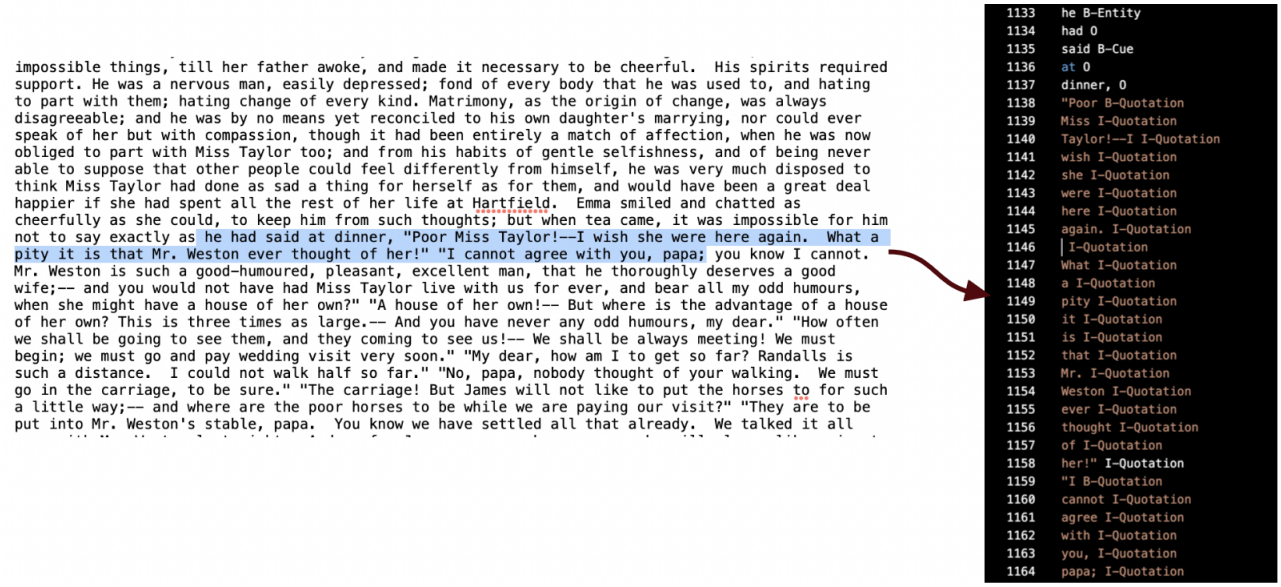

Figure 1: Excerpt from Jane Austen's book titled "Emma" on the left and corresponding Quotation, Cue, Entity detection on the right

- Text is passed into a pretrained BERT model sentence by sentence, and the hidden states of the final layers are summed and saved as a contextualized embedding for each token.
- These embeddings, paired with the labels, are then used for supervised learning methods such as K Nearest Neighbors and Logistic Regression, and Naive Bayes. We also fine-tuned the pre-trained BERT model designed for token classification on our data directly to utilize the model for classification.
- One example of Annotation can be seen below:

[" Especially when one of those two is such a fanciful, troublesome creature!"] QUOTATION [said] CUE [Emma] ENTITY playfully.

Figure 2: Annotation example: 1

In this example, Quotation, Cue and Entity are annotated as shown in the figure and "playfully" is being annotated as '0' as it doesn't belong to either class. It is evident that most of the cue words are adjacent to the quotation, either cue comes just before the quotation or just after it but it's not the case always.

With a little reserve of manner, [Emma] ENTITY [continued:] CUE ["You mean to return a favourable answer, I collect."] QUOTATION

Figure 3: Annotation example: 2

Also, sometimes, cue words comes between two quotations. For example in Figure 4, Annotation example 3, an excerpt from Jane Austen's book Emma is correctly identified as quotation, cue and entity irrespective of the position of cue with respect to the quote.

["I do not know what your opinion may be, Mrs.Weston,"] QUOTATION
[said] CUE [Mr. Knightley,] ENTITY ["of this great intimacy between Emma and Harriet Smith, but I think it a bad thing."] QUOTATION

Figure 4: Annotation example: 3

4 Exploratory Data Analysis (EDA)

In our data analysis, we do the following:

- Word Cloud for whole text to see what all words are appearing and what are some most frequent words

Figure 5: Word cloud for whole text

- Word Cloud for Quotations tokens to see what all words are appearing as quotations and how frequently they appears.

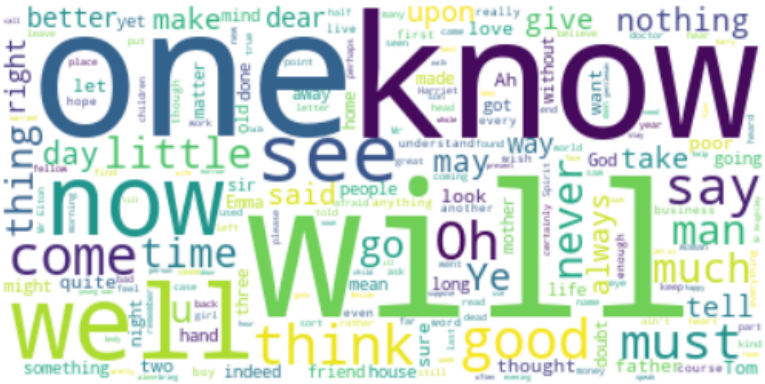

Figure 6: Quotation word cloud

- Word Cloud for Cue tokens to see what all words are appearing as Cues and how frequent. We saw that some of the most common cues used are said, asked, replied.

Figure 7: Cue word cloud

Some of the Cues are "said", "said,", "saying" which are similar to each other but appears as different cues in annotation data. To study further about their similarities we plotted their word2Vec vectors.

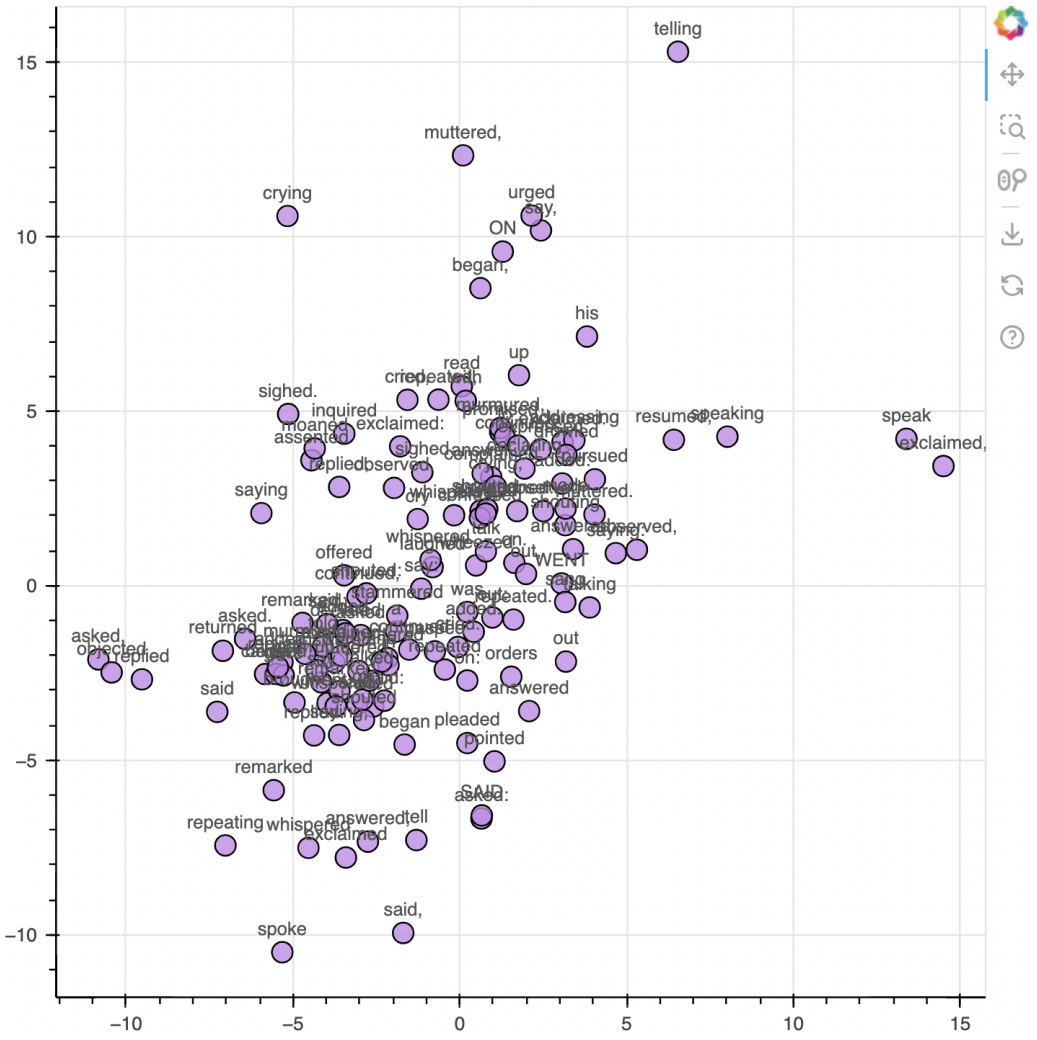

Figure 8: Cue words Embedding Vectors

We can see that similar cue words like: said, asked, saying, answered can be seen very close to each other.

- Word Cloud for Entity tokens to see what all words are appearing as Entities. One thing to be noted is that some of the entities includes "she", "he", "them" etc as annotated in the dataset.

Figure 9: Entity word cloud

5 Problem Formulation

We formulate the problem as a token classification problem, where each token is assigned a tag denoting whether it is the beginning of a span, inside of a span, or outside of a span. Under this formulation, we can take in some amount of text as an input, tokenize the text, and then predict a classification for each token. We choose this as a simple baseline that illustrates how single words do not contain much signal as to whether or not they are likely to be inside of a quotation.

6 Methods

6.1 Most Common Tag method

Our first baseline involves tokenizing the data and classifying each token in the unseen validation and test sets with the most common tag for that token in the training set, with previously unseen tokens classified as untagged. We utilize huggingface's transformers [6] to implement this.

6.2 BERT Embedding Methods

We utilized a pre-trained BERT [3] model to extract context-aware embeddings to use as features for supervised learning. These embeddings are extracted from the final layers of the BERT model when passed a sentence as input, and they include some representation of each token's meaning and role in the sentence.

6.2.1 BERT Embedding and KNN

We used K nearest neighbors over our BERT embedding and solved it as a multi-classification problem using sci-kit learn package: sklearn.neighbors.KNeighborsClassifier. We chose KNN as our second baseline since it does not require any optimization and is simple to set up.

6.2.2 BERT Embedding and Naive Bayes

We used Gaussian Naive Bayes method over our BERT embedding because similar to KNN, it does not require optimization and only needs to fit the data to be able to predict.

6.2.3 BERT Embedding and Logistic Regression

We used logistic regression method over our BERT embedding and solved it as a multi-classification problem using sci-kit learn package: sklearn.linear_model.LogisticRegression. We chose logistic regression as a more robust model to take advantage of the BERT embeddings. Since utilizing a logistic regression model on features extracted from the hidden layers of a BERT model is similar to adding another layer to a BERT model with frozen weights, we thought that Logistic Regression would be able to approximate the performance of a deep neural network with fewer resources needed to train.

6.3 Finetuned BERT

We also fine-tune a BERT model on our data. This model is initialized using weights from a pre-trained model, and is further trained for five epochs on our data. This allows the model to make adjustments to its weights to better specialize for our task and data. For this task, we process our data into CoNLL format and modify an existing notebook[1] for BERT classification of CoNLL data.

7 Experiments and Results

We find that, as expected, the most common tag has some success in identifying cue words but fails spectacularly at identifying quotations, and the BERT-based models that are able to capture more context perform better. The fine-tuned model performs the best, with an F1 score of .75 both on quotations and overall, falling a bit short of the state-of-the-art of around .85. KNN and Naive Bayes both significantly outperformed the most common tag baseline but still fell far short of the finetuned model, achieving F1 scores of .20 and .23 on quotations in the test set, respectively. The Logistic Regression model, however, was able to approach the performance of the finetuned model, reaching .65 F1 on quotations in the test set. Our F1 scores for each model are listed in Table 1, with more detailed results available in Appendix A.

Table 1: F1 Scores

Model	Val F1					Test F1				
	Cue	Entity	Quotation	O	Avg	Cue	Entity	Quotation	O	Avg
Most Common Tag	0.72	0.13	0.01	0.37	0.20	0.72	0.11	0.01	0.33	0.24
BERT KNN	0.85	0.45	0.31	0.51	0.43	0.82	0.44	0.20	0.42	0.33
BERT Logistic Regression	0.88	0.62	0.65	0.74	0.74	0.88	0.62	0.65	0.74	0.70
BERT Naive Bayes	0.44	0.17	0.22	5103	0.46	0.52	0.16	0.23	0.55	0.55
BERT Finetune	**0.89**	**0.70**	**0.85**	-	**0.81**	**0.88**	**0.67**	**0.75**	-	**0.75**

8 Conclusion and Discussion

Overall, our results are in line with expectations, with the baseline performing the worst by far, and the finetuned model performing the best. The vast gap in performance between the baseline and even simple models like KNN illustrates the need for context in a task like quotation detection. Logistic Regression is able to perform fairly well, which makes sense since it is similar to adding a fully-connected layer on top of a pretrained BERT model with frozen weights. Still, this is an encouraging result, showing that it is possible to achieve markedly increased performance over the baselines without changing any weights in the BERT model, requiring fewer computational resources than a full finetune.

8.1 Error Analysis

Mistakes that the finetuned model makes include:

- Prematurely starting or ending a quotation at a sentence separation

[1] https://github.com/chnsh/BERT-NER-CoNLL

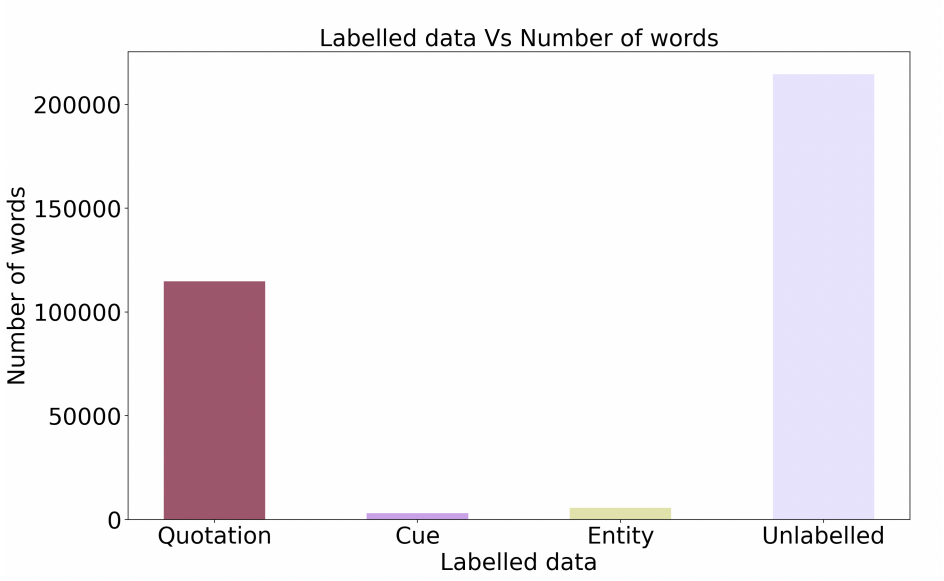

Figure 10: Labelled data Vs Number of words

- Classifying things surrounded by quotation marks as quotations even when they are not

Of these errors, the first is the most common, and it is likely caused by the fact that BERT takes in input one sentence at a time, so the previous context is lost when moving to the next sentence. This limitation is far more prevalent in quotation detection than in other token classification tags such as Named Entity Recognition since quotation spans can be very long and often encompass multiple sentences. This could potentially be addressed by preserving some representation of previous sentences or by utilizing a large language model with a longer context window, such as a GPT [1] flavor. Another approach could be to have a two-stage prediction process, where a fine-tuned BERT model provides intermediate predictions which are used as features by a non-fixed-length sequence model such as a Hidden Markov Model or Recurrent Neural Network. The second error is likely an overfit on quotation marks, as they by far the most prevalent indicator of direct quotations. A possible direction for future work is to remove punctuation entirely and analyze the effects on performance.

A Additional Results

BERT KNN	Test Data				Validation Data			
	Precision	Recall	F1-score	Support	Precision	Recall	F1-score	Support
Cue	0.81	0.83	0.82	162	0.87	0.82	0.85	280
Entity	0.51	0.38	0.44	319	0.52	0.39	0.45	475
Quotation	0.15	0.29	0.20	3805	0.25	0.42	0.31	4860
O	0.48	0.37	0.42	3994	0.56	0.47	0.51	5103
Micro avg	0.27	0.34	0.30	8280	0.37	0.45	0.41	10718
Macro avg	0.49	0.47	0.47	8280	0.55	0.53	0.53	10718
Weighted avg	0.34	0.34	0.33	8280	0.43	0.45	0.43	10718

BERT Logistic Regression	Test Data				Validation Data			
	Precision	Recall	F1-score	Support	Precision	Recall	F1-score	Support
Cue	0.84	0.83	0.84	162	0.91	0.86	0.88	280
Entity	0.70	0.53	0.60	319	0.71	0.55	0.62	475
Quotation	0.52	0.64	0.57	3805	0.60	0.70	0.65	4860
O	0.75	0.67	0.71	3994	0.78	0.71	0.74	5103
Micro avg	0.62	0.65	0.64	8280	0.69	0.70	0.69	10718
Macro avg	0.70	0.67	0.68	8280	0.75	0.70	0.72	10718
Weighted avg	0.64	0.65	0.64	8280	0.70	0.70	0.70	10718

BERT Naive Bayes	Test Data				Validation Data			
	Precision	Recall	F1-score	Support	Precision	Recall	F1-score	Support
Cue	0.39	0.78	0.52	162	0.33	0.65	0.44	280
Entity	0.09	0.76	0.16	319	0.09	0.78	0.17	475
Quotation	0.19	0.29	0.23	3805	0.19	0.27	0.22	4860
O	0.54	0.55	0.55	3994	0.44	0.48	0.46	5103
Micro avg	0.29	0.44	0.35	8280	0.25	0.40	0.31	10718
Macro avg	0.30	0.59	0.37	8280	0.26	0.54	0.32	10718
Weighted avg	0.36	0.44	0.39	8280	0.31	0.40	0.34	10718

References

[1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[2] Yue Chen, Zhen-Hua Ling, and Qing-Feng Liu. A Neural-Network-Based Approach to Identifying Speakers in Novels. In *Interspeech 2021*, pages 4114–4118. ISCA, August 2021.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[4] Institute for Natural Language Processing, University of Stuttgart, Germany, Sean Papay, and Sebastian PadÃ³. $QuotationDetectionandClassificationwithaCorpus - AgnosticModel. InProceedings - NaturalLanguageProcessinginaDeepLearningWorld, pages888 - -894. IncomaLtd., Shoumen, Bulgaria, October2019.$

[5] Sean Papay and Sebastian PadÃ³. $RiQuA : ACorpusofRichQuotationAnnotationforEnglishLiteraryText.page$ 7.

[6] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, RÃ©mi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing, 2019.