Conversational Question Answering System

Charmi Chokshi
University of Montreal (DIRO)
charmi.chokshi@umontreal.ca

Hena Ghonia
University of Montreal (DIRO)
hena.ghonia@umontreal.ca

Sandeep Kumar
University of Montreal (DIRO)
sandeep.kumar.1@umontreal.ca

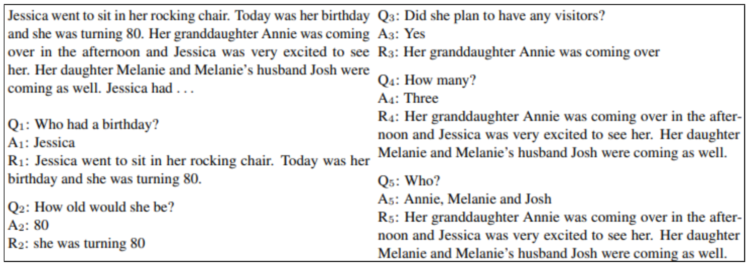
Vamsikrishna Chemudupati
University of Montreal (DIRO)
vamsikrishna.chemudupati@umontreal.ca

Abstract

Building a conversation system for human-like communication is a challenging yet interesting problem in the domain of Natural Language Processing. This field has faced several challenges since its inception but the biggest issue is the lack of appropriate data. We expect these systems to learn the underlying meaning and build reasoning by providing inadequate task-specific data. The research in this field started with building seq2seq based models such as LSTM/GRU but due to long-term dependency issues, in recent days, language models such as Transformers are being considered as one of the possible solutions. In this project, we hope to build a question answering system based on the CoQA dataset. The main aim of CoQA dataset challenge is to build a system that can understand a passage and be able to reply to interconnected queries on it. Our project aims to tackle the context problem faced in this challenge using FlowQA, Graphs and Transformer based methods.

1 Introduction

The main aim of asking a question in a conversation is to either seek or test information the other person has. In a conversation once the question is answered by someone then it is usually followed by another question whose answer depends on previously asked queries and their answers. This is what makes human conversation so clear & precise and has been a long standing challenge in modern NLP systems. Its due to this reason even modern virtual assistants don't feel like a competent conversation partner.CoQA dataset challenge aims to tackle this problem and make machines capable of understanding the long term dependencies required for this. It measures the ability of the machines to participate in human like question-answering conversation.



Jessica went to sit in her rocking chair. Today was her birthday and she was turning 80. Her granddaughter Annie was coming over in the afternoon and Jessica was very excited to see her. Her daughter Melanie and Melanie's husband Josh were coming as well. Jessica had . . .

Q_1: Who had a birthday?
A_1: Jessica
R_1: Jessica went to sit in her rocking chair. Today was her birthday and she was turning 80.

Q_2: How old would she be?
A_2: 80
R_2: she was turning 80

Q_3: Did she plan to have any visitors?
A_3: Yes
R_3: Her granddaughter Annie was coming over

Q_4: How many?
A_4: Three
R_4: Her granddaughter Annie was coming over in the afternoon and Jessica was very excited to see her. Her daughter Melanie and Melanie's husband Josh were coming as well.

Q_5: Who?
A_5: Annie, Melanie and Josh
R_5: Her granddaughter Annie was coming over in the afternoon and Jessica was very excited to see her. Her daughter Melanie and Melanie's husband Josh were coming as well.

Figure 1: A conversation in CoQA dataset. Each turn has a question (Q_i), an answer (A_i) and a rationale (R_i) that supports the answer

Preprint. Under review.

As we can see from the example in Figure 1 the conversation is being built up in the form of sequences of continuous questions and answers. It shows how concise the conversation can be in Q5 where the question is only one word "Who". Current systems are not capable of understanding and keeping track what has already been said in the conversation. To answer these sort of question a system has to learn how to handle a typical human conversation. Another goal of CoQA is to build QA systems that perform robustly across different domains. Usually these type of tasks deal with only one domain. Problem with that is the generalization performance of the models can't be tested properly. So to deal with this CoQA dataset comprises of seven different domains — children's stories, literature, middle and high school English exams, news, Wikipedia, Reddit and science.

2 Literature Survey

To tackle the challenges of CoQA dataset we have selected FlowQA, Graph Neural Networks and Transformers (BERT) based approach.

2.1 GraphFlow-GNN based model for conversational Machine comprehension

Many approaches for conversational machine comprehension do not capture conversation history and have trouble handling questions involving co-references. To overcome this challenge authors of GraphFlow[4] have proposed a graph structure learning technique to dynamically construct a question and conversation history context graph at each turn that consists of each word as a node. GraphFlow method consists of Encoding layer, Reasoning Layer and Prediction Layer; where Encoding layer encodes conversation history and context regarding question information using GloVe, BERT embedding and attention mechanism is applied on question embeddings. Since a fully connected context graph is computationally expensive, k-NN style graph sparsification is applied to select the most important edges. Reasoning Layer performs reasoning over context as a 'graph' of words that captures semantic relationships among words and applies Recurrent Graph neural network(RGNN) to process a sequence of context graphs. RGNN combines the advantages of sequential learning(RNN) and GNN which are good at relational reasoning. The prediction layer predicts the answer based on the matching score of question embedding and the context graph embedding. GraphFlow outperforms FlowQA and BiDAF++ on CoQA, DoQA and QuAC data set by 7.2% where F1 score is used as a metric.

2.2 FLOWQA

Many previously proposed systems train the machine comprehension system based on augmentation of the prior questions and context, to answer the current question. But these don't take into account the hidden representations generated during the process of answering previous questions related to the context. The hidden representations of the history provide clues and facts collected before, understanding the context of the present question. The architecture follows as word embeddings are constructed using glove for both questions and the context provided. Further, the attention-based mechanism is used for each word in the question to generate question-based context vectors using the glove embeddings. The intermediate representations for each question help in constructing a flow of data which helps in answering the present question. Along with this fully aware self-attention mechanism is applied to the complete context to extract the important sequences from them. The representations obtained are used collectively to predict the answer to the questions. Also, the case of unanswerable questions is considered where if the start word and end word probability values obtained are too low then the context available is not enough to answer the question. A larger gain in the performance is observed on CoQA dataset which contains longer dialog chains suggesting it can capture the long-range conversation history effectively.

2.3 BERT - Bidirectional Encoder Representations from Transformers

One of the important techniques developed in the domain of pre-trained language models is BERT or Bidirectional Encoder Representations from Transformers. It is based on transformer architecture and trained bidirectionally using the encoder module 2. It takes input in the form of token, segment, and positional embeddings and utilizes MLM(Masked Language Model) and NSP(Next sentence prediction) strategies to train. The main idea behind MLM is to train the model to predict a randomly masked out word which is based on the Cloze task. The NSP is the technique that allows BERT to perform well on Question Answering tasks. The model takes two input sequences and predicts whether the second sequence is the next sequence in the original text or not. The main strength of BERT comes from the pretraining it does beforehand on a large corpus. This knowledge is then transferred and fine-tuned for downstream tasks which is computationally much more efficient than training the complete model from scratch. The model when it was introduced was able to achieve state of the art in eleven NLP tasks and had some significant jumps in the case of Question Answering domain datasets like SQuAD v1.1 and SQuAD v2.0.

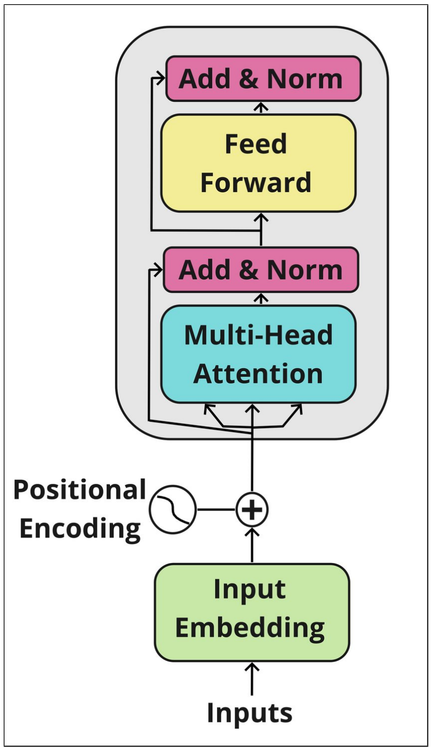


Figure 2: Transformer Encoder

3 Data Analysis

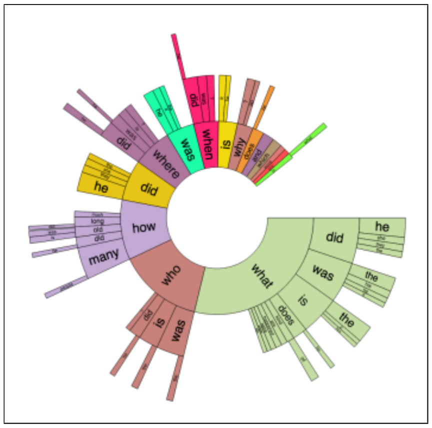
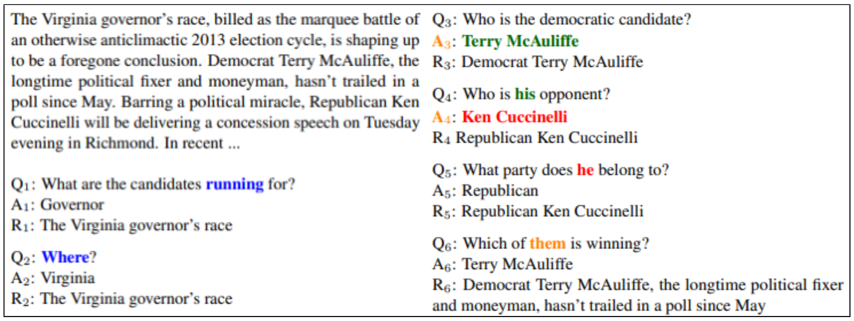


Figure 3: Distribution of Trigram Prefixes

Table 1: Domain Distribution CoQA

Domain	#Passages	Q/A pairs	Passage Length	Turns per passage
Part				
		In-Domain		
Children's Sto.	750	10.5k	211	14.0
Literature	1815	25.5k	284	15.6
Mid/High Sch.	1911	28.6k	306	15.0
News	1902	28.7k	268	15.1
Wikipedia	1821	28.0k	245	15.4
		Out-of-Domain		
Reddit	100	1.7k	361	16.6
Science	100	1.5k	251	15.3
Total	8399	127k	271	15.2

As the Figure 3 suggests the distribution of CoQA is spread across multiple question types. We see a wide variety of sectors indicated by prefixes like what, who, when, where etc. Almost every sector in CoQA has co-referencing indicating that it has a very high conversational aspect.



The Virginia governor's race, billed as the marquee battle of an otherwise anticlimactic 2013 election cycle, is shaping up to be a foregone conclusion. Democrat Terry McAuliffe, the longtime political fixer and moneyman, hasn't trailed in a poll since May. Barring a political miracle, Republican Ken Cuccinelli will be delivering a concession speech on Tuesday evening in Richmond. In recent ...

Q_1: What are the candidates **running** for?
A_1: Governor
R_1: The Virginia governor's race

Q_2: **Where**?
A_2: Virginia
R_2: The Virginia governor's race

Q_3: Who is the democratic candidate?
A_3: **Terry McAuliffe**
R_3: Democrat Terry McAuliffe

Q_4: Who is **his** opponent?
A_4: **Ken Cuccinelli**
R_4 Republican Ken Cuccinelli

Q_5: What party does **he** belong to?
A_5: Republican
R_5: Republican Ken Cuccinelli

Q_6: Which of **them** is winning?
A_6: Terry McAuliffe
R_6: Democrat Terry McAuliffe, the longtime political fixer and moneyman, hasn't trailed in a poll since May

Figure 4: Co-referencing (shown in color) for a conversation example

As we can see from the Figure 4 the questions are presented in a co-referenced form allowing us to simulate a human conversation. This free form in questioning and answering leads to long term dependencies in the conversation history which in turn leads to precise and concise answers. As we can see most of the answers are just one or two words (more in case of names). The dataset comprises of seven different domains — children's stories, literature, middle and high school English exams, news, Wikipedia, Reddit and science. There are always some limitations involved in big datasets. In this case its due to the fact that not all passages of each domain distributions are good enough for generating conversations like humans. The overall distribution of domains is explicitly mentioned in the Table 1 below. The In-Domain questions and answer conversations comes from the children's stories, literature, middle and high school English exams, news, Wikipedia. The remaining two categories of Reddit and science are used as Out of Domain part.

The whole CoQA dataset can be summarized as :

- It consists of 127k conversation turns collected from 8k conversations over text passages. The average conversation length is 15 turns, and each turn consists of a question and an answer.
- It contains free-form answers and each answer has a span-based rationale highlighted in the passage.
- Its text passages are collected from seven diverse domains: five are used for in-domain evaluation and two are used for out-of-domain evaluation.

4 Methodology

4.1 GraphFlow

GraphFlow architechture consists of 3 layers: Encoding layer, Reasoning Layer and Prediction Layer. The GraphFlow model can accurately depict the flow of a discussion and offer better interpretability.

4.1.1 Encoding Layer

This layer encodes the question and the context and interaction between them.
Linguistic features: For every context word, Part of speech tagging and Named entity recognition and exact matching whether context word appears in question or not is concatenated.
Pretrained word embeddings:300-dim GloVe embeddings and 1024-dim BERT embeddings. Using BERT embeddings improves the performance accuracy.
Attention mechanism: Attention mechanism is applied between context words and question words
Conversation history: A feature vector is concatenated with previous N embedding which we denote as History length. Here we use history length as 2. Higher the value of history length better accuracy is gives but number of parameters also increases which results is more computation time.

4.1.2 Reasoning Layer

This layer treats context as graph of work and find relationship among sequence of words.
kNN-style graph: to select most most important edges from fully connected graph which results in less sparse graph.
BiLSTM: captures local dependency followed by a Gated graph neural network (GGNN- RNN style structure) which provides relational reasoning.
Multihop message passing in GGNN to capture long range dependency.

4.1.3 Prediction Layer

Predicts the answer based on the matching score of question embedding and the context graph.
Answer type classifier- to handle unanswerable questions. (e.g., “unknown”, “yes” or “no”.)

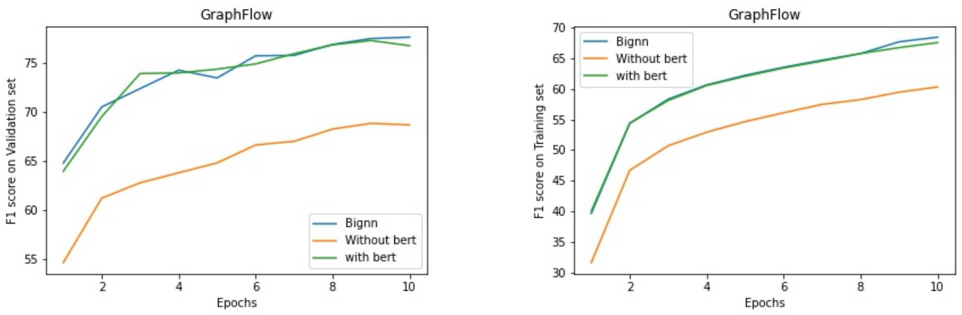
4.1.4 Model variants

- Trained model with BERT and GLoVe embedding(with Bert in 5). (number of parameters = 2,96,66,554)

- Trained model only with GloVe embedding(without Bert in 5) (number of parameters= 2,69,00,706)

- With Bidirectional - GNN(Bignn in 5) (performs better but computationally expensive, number of parameters = 3,03,87,154)

As seen in below Figure 5, Bidirectional - GNN performs slightly better than with normal GNN(with bert in 5) but at cost of more parameters than normal GNN. Concatenating BERT embedding with GLove significantly improves F1 score(large difference observed between with Bert and without bert in 5).



(a) Training F1 score

(b) Validation F1 score

Figure 5: Plot of training and validation F1 score for 3 model variants

4.2 FlowQA

Conversation flow is crucial in a Question Answering system to answer the upcoming questions .As the conversation progresses,the topic being discussed changes over time.The model integrates both previous question/answer pairs and FLOW, the intermediate context representation from conversation history.
In this section we describe the approach taken for FLOW based models during our experiments.GRU model has been used for dialogue context and contextual embeddings experiments due to their ability to train faster.

- **RNN types**:
  We implemented the model using both LSTM and GRU techniques.We see a minor difference in the f-score achieved between the two models and LSTM performs better as shown in figure 6.But its evident by the number of parameters they use as GRU uses less parameters compared to LSTM and can be used in case of light-weight models.

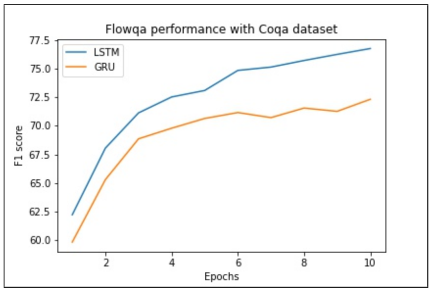


Figure 6: RNN types

- **Pretrained word embeddings**:
  The pre-trained embeddings in the first layer help the model to capture the semantic and syntactic meaning of a word.We used Glove,Word2vec as the pre-trained embeddings and evaluated its performance.Glove works slightly better than Word2vec.As per the literature sources their performance is dependent on dataset and the downstream task they are being used for.Hence for this task Glove works the best as shown in figure 7 .
- **Number of previous answers as features(dialogue context)**:
  The dialogue context parameter helps the model in replicating human answering procedure which tries to understand the context of previous answers and also add up as the features being used in predicting the next answer.We start with training a model keeping the dialogue context parameter to zero and gradually increase it to 4. As per the results when the dialogue context parameter is set above 2 we do not see any significant improvement in the performance as shown in figure 8.Therefore this gives us an analysis on representations which the model is able to extract from the previous answer statements.

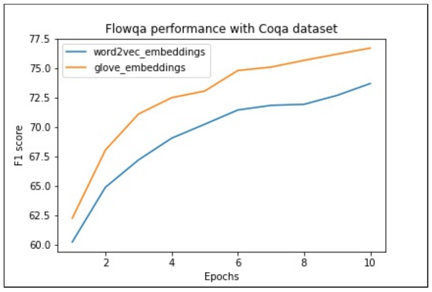


Figure 7: Pre-trained word embeddings

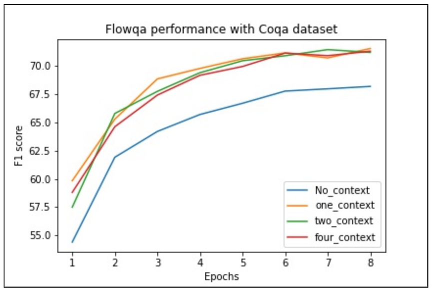


Figure 8: Dialogue context

- **Contextual and Non-contextual embeddings**:
  The experiment was performed considering two configurations one being Glove + Elmo and other only Glove. Glove + Elmo performed better than other configuration as shown in figure 9 which explains us the need for contextual embeddings in capturing the semantic information of the words and sentences.

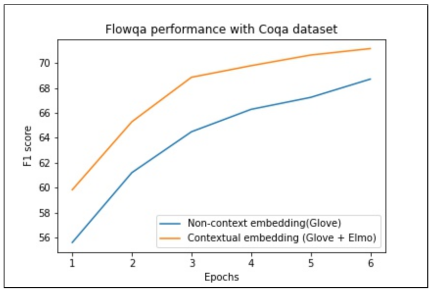


Figure 9: Contextual vs Non-contextual

4.3 BERT

Bidirectional Encoder Representations from Transformers or BERT for short are transformer based deep learning models. BERT models follow Self Supervised Learning approach where they are pre-trained on large corpus and then fine-tuned on a downstream task. In this section we describe the approach taken for BERT based models during our experiments.
We used the two different pretrained variants available on huggingface namely bert-base and bert-large. Model architecture details are as follows :

- Bert Base : 12-layer, 768-hidden, 12-heads, 110M parameters
- Bert Large : 24-layer, 1024-hidden, 16-heads, 340M parameters

These models are pretrained on lower-cased English wiki text and then fine tuned on CoQA dataset. We also experimented on Bert models that were previously fine-tuned on SQUAD dataset which

actually turned out to be the best performing.
To incorporate the human question answering style conversation we incorporated history aspect to the model. It took the previous questions and answers upto the length specified by user and embedded that into the query. This allowed the model to look at the previous questions that have been asked and the corresponding answers. This in turn helped the model to learn the expected behaviour which is required for CoQA dataset.
Given a passage C and the lets say the previous history of conversation had questions and answers as $Q_1, A_1, Q_2, A_2...Q_{k-1}, A_{k-1}$. Then the current question with the history parameter H is reformulated as $\hat{Q_k} = \{Q_{k-H}; A_{k-H}; ...; Q_{k-1}; A_{k-1}; Q_k\}$. This reformulation keeps track of previous H question and answers and helps the model to understand the new question better.

5 Result

Model	**History Length**	**# Epochs**	**# Parameters**	**Training Time(hours)**	**F1 score (Validation)**
FlowQA: GRU (No dialogue context)	0	8	10.5M	6	68.188
FlowQA: GRU(Glove embedding only)	2	6	8.7M	6	68.708
GraphFlow (Glove embedding)	2	10	26M	10	68.86
BERT-base (wikitext data)	0	2	110M	2	70.3
FlowQA: GRU (Glove + Elmo)	2	10	10.5M	6	72
FlowQA: LSTM (Word2vec+Elmo)	2	10	12M	6	73.726
Bert-large (squad data)	6	4	340M	24	76.4
GraphFlow (Glove & Bert embeddings)	2	10	29M	11	76.75
FlowQA: LSTM (Glove + Elmo)	2	10	12M	6	77
GraphFlow: BiGNN model	2	10	30M	20	77.62
Bert-base (wikitext data)	2	2	110M	2	78
Bert-base (squad data)	2	2	66M	4	78.9
Bert-large (squad data)	**2**	**2**	**340M**	**8**	**82.1**

As observed in above table Bert-large(trained on Squad data for history length=2) and finetuned on CoQa dataset performs best compared to all other experiments. However, it has 340M parameters which is highest compared to all other experiments. Flow QA and GraphFlow also gives better performance with less number of parameters. Hence trade-off between number of parameters and performance is observed from above results. Please find source code of all methods at https://github.com/Hstellar/Conversational_Question_Anwering_System

6 Conclusion

We tried various graph, flow and transformer based approaches on the CoQA dataset and understood the domain of question answering and transfer learning. BERT based models have an immense number of parameters and hence, take a lot of time (1X Tesla V100 GPU) to be trained but have good F1 score. Graph based models have comparatively less no of params and training time (1 X

Tesla K80/P100 GPU) but have comparatively less F1 score than BERT. Flow based models have the least number of parameters and hence take the least training time (1X Tesla V100 GPU). In terms of performance it is almost equivalent to the Graph based models but is not at par with BERT. The models are sequential and hence to train them faster, requires major architectural change.

7 Statement of Contribution

All of the work was equally divided among the teammates. The main division would be as follows:
Bert Based Experiments - Charmi Chokshi and Sandeep Kumar
GraphFlow Experiment - Hena Ghonia
FlowQA Experiment - VamsiKrishna Chemudupati

Acknowledgments and Disclosure of Funding

Thanks to Google Developer Experts help for providing us with the credits of Google Cloud. With Cloud credits enabled us to train really large models like bert which turned out to be best among all the experiments we conducted.

References

[1] Reddy, Siva, Danqi Chen, and Christopher D. Manning. "Coqa: A conversational question answering challenge." Transactions of the Association for Computational Linguistics 7 (2019): 249-266.

[2] Chen, Yu, Lingfei Wu, and Mohammed J. Zaki. "Graphflow: Exploiting conversation flow with graph neural networks for conversational machine comprehension." arXiv preprint arXiv:1908.00059 (2019).

[3] Huang, Hsin-Yuan, Eunsol Choi, and Wen-tau Yih. "Flowqa: Grasping flow in history for conversational machine comprehension." arXiv preprint arXiv:1810.06683 (2018).

[4]Huang, Hsin-Yuan, et al. "Fusionnet: Fusing via fully-aware attention with application to machine comprehension." arXiv preprint arXiv:1711.07341 (2017).

[5] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

[6] Zhu, Chenguang, Michael Zeng, and Xuedong Huang. "Sdnet: Contextualized attention-based deep network for conversational question answering." arXiv preprint arXiv:1812.03593 (2018).

[7] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

[8] Zaib, Munazza, Quan Z. Sheng, and Wei Emma Zhang. "A short survey of pre-trained language models for conversational ai-a new age in nlp." Proceedings of the Australasian Computer Science Week Multiconference. 2020.

[9] Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).

[10] https://wikipedia2vec.github.io/wikipedia2vec/pretrained/

[11] https://nlp.stanford.edu/projects/glove/