REVERSING THE TWENTY QUESTIONS GAME

CSC 791 - NATURAL LANGUAGE PROCESSING

Parth Parikh
Department of Computer Science
North Carolina State University
Raleigh, NC 27606
pmparikh@ncsu.edu

Anisha Gupta
Department of Computer Science
North Carolina State University
Raleigh, NC 27606
agupta44@ncsu.edu

November 30, 2021

ABSTRACT

Twenty questions is a widely popular verbal game. In recent years, many computerized versions of this game have been developed in which a user thinks of an entity and a computer attempts to guess this entity by asking a series of boolean-type (yes/no) questions. In this research, we aim to reverse this game by making the computer choose an entity at random. The human aims to guess this entity by quizzing the computer with natural language queries which the computer will then attempt to parse using a boolean question answering model. The game ends when the human is successfully able to guess the entity of the computer's choice.

Keywords Twenty questions game · Query Reformulation · Passage Retrieval · Boolean Question-Answering Model · Natural Language Inference

1 Introduction

For our course project, we aim to reverse the roles of the computer and human, such that *the computer will act as an answerer and a human as a questioner*. In the past, no such study has been conducted as this problem presented sophisticated challenges of Natural Language Inference and Textual Entailment. However, with the advent of transformer-based machine learning techniques such as BERT [1], RoBERTa [2], GPT-2 [3], and datasets such as BoolQ [4], such a model can be constructed.

As this problem has not been formally defined, our goal is to formalize it and present preliminary results regarding the same. Furthermore, while there are several pre-trained question-answering models that select the start and end points of a corpus containing an answer, a simple yes/no answering task is surprisingly challenging and complex. A model for such a task would have to examine entailment as well as investigate if the corpus makes a positive answer to the question unlikely, even if it doesn't directly state a negative answer [4]. Our reverse Akinator model could be used for any sort of factual checker to examine whether a statement is true or not, given a knowledge corpus.

2 Methodology

2.1 History

Historically, Twenty Questions has been a popular multi-player parlour game wherein some participants would act as the *questioners* and the others would be the *answerers*. The answerers would come up with a random entity which the questioners would then try and deduce by asking a series of yes/no questions. A 19^{th} century rule-book [5] details the format of the game and introduces the concept of umpires (who resolve any dispute) and captains (an official spokesperson). Interestingly, though the rule book never constrained the *subject* (guess), every Sunday, it was mandatory for the participants to pick an object, person, or thing mentioned in the Bible.

Constrained versions of the game soon became popular and a variant known as the *animal, vegetable, minerals* was widely played in parlours. As constraints produced tractability, one of the earliest computerized implementations of this game solely used *Animals* as its subject [6]. This game was part of the *101 BASIC Computer Games* (1973). Around 1988, *20Q* created by Robin Burgener emerged. This version used an artificial neural network to answer questions based on a human's interpretation of that question. Today, popular internet-based variants such as *Akinator* deals with a wide category of entities and includes *Probably*, *Probably not* and *Don't know* as potential answers for a human.

2.2 Entity Formulation and Pronoun Resolution

Our proposed model starts by selecting a random Wikipedia page of a named entity. This entity acts as our model's main entity - *the guess*. These random Wikipedia pages can be extracted by passing SPARQL queries to Wikidata [7]. The model then accepts natural language queries from a user. As the first step, each of these queries undergoes a basic pronoun resolution wherein a pronoun gets replaced with the model's main entity. For example, the model is likely to predict better results if we formulate the query in the following manner -

Is **it** an animated character? $\rightarrow$ Is **Mickey Mouse** an animated character?

This step ensures that our model does not easily get confused when it sees another entity with a similar context.

2.3 Paragraph/Sentence Retrieval

To obtain a relevant passage from the entity's Wikipedia text, we require a passage retrieval phase. Here, *relevance* can be defined as a passage from the main entity's text-body which unambiguously answers a boolean-type query. For example -

Is *Mickey Mouse* a comic book character?

> "Beginning in 1930, **Mickey has also been featured extensively in comic strips and comic books**. The Mickey Mouse comic strip, drawn primarily by Floyd Gottfredson, ran for 45 years. Mickey has also appeared in comic books such as Mickey Mouse, Disney Italy's Topolino and MM – Mickey Mouse Mystery Magazine, and Wizards of Mickey."
> *- From the Wikipedia page of Mickey Mouse (paragraph 3)*

As mentioned in [8], a trivial solution to this problem would be to perform sentence segmentation on the entire Wikipedia page and pass all the sentences to the question answering model. However, this can significantly affect the computational complexity as certain phases in BERT such as the *multi-headed attention layer* requires $n^2 \cdot d + n \cdot d^2$ operations (here n is the sequence length and d is the depth) [9].

A sophisticated variant would be to rank the passages based on the query and retrieve the first N passages. We can use a ranking function such as Okapi BM25 [10] for such a task. However, as [10] uses a bag-of-words-based approach, its rankings can be too literal and devoid of any implicit context. To resolve this, we introduce a hybrid approach wherein a large subset of $N_1 \subseteq P$ passages is retrieved using BM25 and a much smaller subset $N_2 \subseteq N_1$ is then obtained using *Siamese BERT-Networks* [11]. Here, sentences/paragraphs are mapped to a dense vector representation using transformer networks such as BERT, which can then be compared using cosine similarity. We plan on embedding the query Q and comparing it against the embeddings of each $n \in N_2$, keeping a track of the top N passages. A Python library - *Sentence Transformers* [12] provides pre-trained models for this task.

The above mentioned model uses a *sparse-first search* mechanism wherein we retrieve the N_1 documents using a statistical approach which is followed by a neural model. The drawback of this is that we may propagate errors from the document retrieval phase. That is, if we retrieve the wrong documents then it might affect the performance of the Transformer models. To mitigate this, Facebook Research developed *Dense Passage Retrieval* [13] which uses the concept of indexing phrases using a dual-encoder framework. Here, they enumerate a document for all phrases in that document and use a phrase encoder to embed each phrase in vector space. The queries are mapped to the same vector space and Nearest Neighbour Search is used to obtain the most relevant answers.

2.4 Boolean Question Answering Model

To guess the boolean-type response, we propose a transformer-based model which takes as its input a query and N_2 relevant paragraphs. We plan on experimenting with a BERT model pre-trained on entailment tasks and fine-tuned using the BoolQ dataset [4]. [4] showed that the highest accuracy is obtained when we pre-train models on entailment tasks that have large datasets (such as MultiNLI [14] and SNLI [15]) and fine-tuning them on BoolQ's dataset.

While playing games with Akinator, we observed that a certain class of questions can be answered using knowledge repositories such as Wikidata and DBpedia [16]. These questions involve highly distinguishing characteristics of the entity such as its gender, species, hypernyms, and significant others.

3 Experiments

As mentioned in Report 1's evaluation section, we verified our model's performance by playing it against the pre-existing Akinator using the Python library *akinator.py* [17]. This library acts as the original Akinator, posing questions to our model and trying to guess which entity our model has in mind. The number of questions asked by the Akinator is not constrained in our experiments. We only stop the game once the Akinator guesses an entity with a probability greater than 80%.

3.1 Akinator API

The Akinator API [17] allows us to access the Akinator's top guesses at a particular time, with a guess probability and a rank. The first guess is used to evaluate if the Akinator won (that is, if the Akinator was able to guess the answer correctly). The API also allows us to go back to a previous question and change our answers. Furthermore, we are able to select a nature of the entities that we want to guess. This comprises of language options (such as English, Chinese, German), and entity types (like animals, characters, and objects).

3.2 Baseline model

Our initial baseline model answers the Akinator's questions at random with *Yes*, *Probably*, *I don't know*, *Probably not* and *No*. However, when we performed our experiments, we observed that too many *i don't know* or *probably yes/no* responses would make the Akinator guess something along the lines of *a guy who plays randomly* (this statement is one of Akinator's named entity which it assigns to anyone who guesses randomly). So we allocated these responses a much lower probability of 0.05 each, and distributed the remaining probabilities uniformly among the rest of the answer options, such that the baseline model could make a probabilistic random choice.

An entity only shows up when it is within the top few guesses of the Akinator. From our experiments on our initial baseline model, we hardly ever see the desired item show up in the list of top few guesses of the Akinator.

From the results shown in Figure 1, we see that the Akinator's guess converges to a *Sharktopus* with a final probability $> 80\%$. However, the guess is incorrect, as is expected, since it's a random model. The desired animal (*Cheetah*) never features in the Akinator's guess list. In this model, the correct answer can only show up in the list of top guesses by chance, and this happens very rarely.

We performed some preliminary analysis using anaphora resolution on the questions asked by the Akinator. However, in some cases (ex. Table2), the extracted answer excerpts are more unrelated to the question after applying anaphora resolution to the question. As part of our preliminary analysis, we also explored the BERT Question Answering model. However, based on manual inspection of the results, the excerpts extracted using the BERT Question Answering model are less relevant to the question than that extracted using our pipeline. This could be supported by Reimers et al.'s work [11], where they show that averaging the `[CLS]` tokens for the BERT embeddings "...yields rather bad sentence embeddings, often worse than averaging GloVe embeddings".

3.3 Improved Model

For our improved model, we implemented the *Okapi-BM25/SBERT* pipeline proposed in Section 2.3. We fixed N_1 to 100 and N_2 to 5. For our current experiments, our pipeline outputs these top N_2 *most similar* excerpts that answers the Akinator's question at each step, and lets the human developer answer a *Yes*/*No* based on these top five excerpts.

An example output of the same is shown in Figure 2.

3.3.1 Constraining the domain

Our initial experiments using the aforementioned pipeline did not produce good results for general entities, including movie characters such as *Harry Potter*, as can be observed from the example in Figure3. This is often because the complexity of information is more for such characters, with both real life and reel life data, as well as information about a lot of other characters/persons documented in the Wikipedia articles. Given the lack of access to knowledge graphs, trivia questions are more difficult for our model to answer. We thus constrain our domain to English animal names.

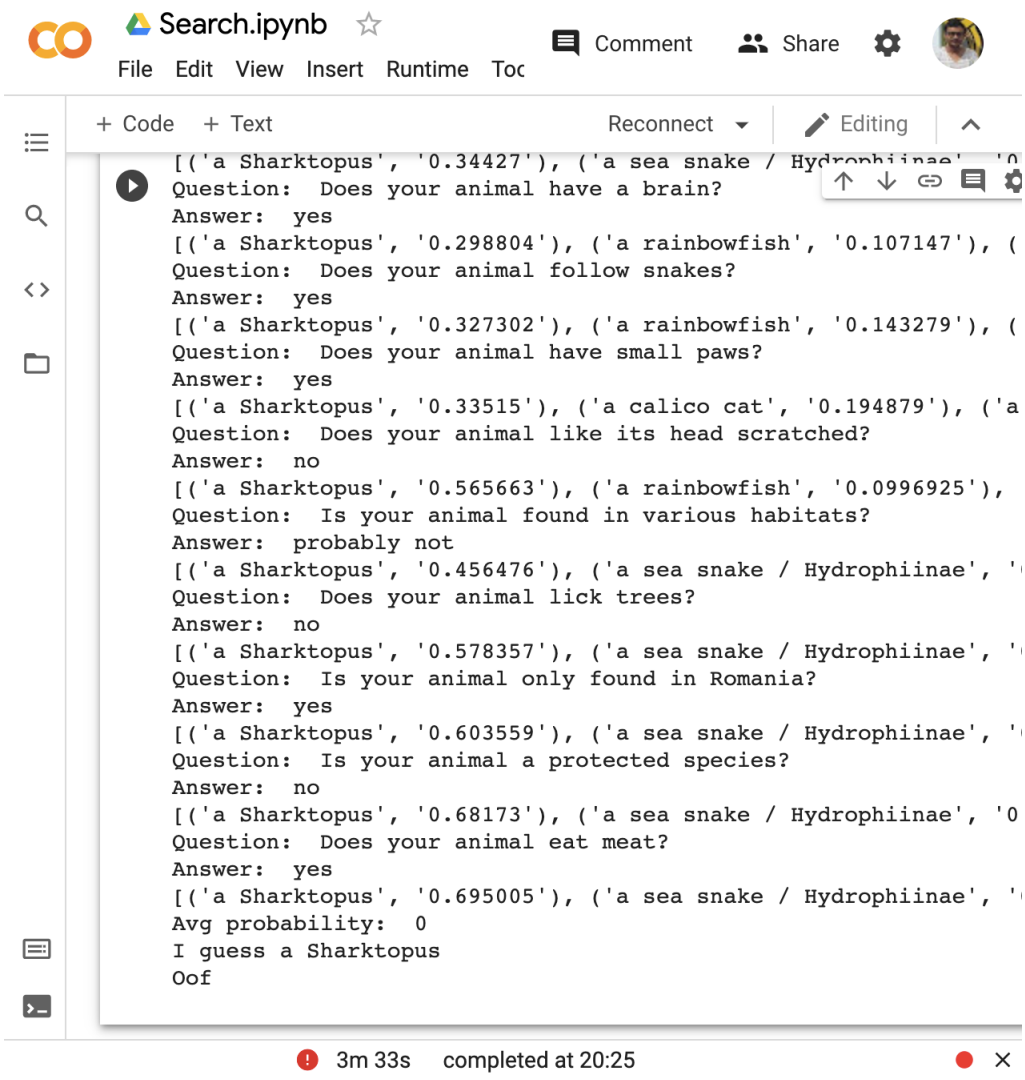

Figure 1: Random baseline model results

The Wikipedia articles corresponding to a certain animal usually only talks about this animal and does not have a lot of content on other animals or information that requires a knowledge base for answering questions, thus making our problem more tractable for our purposes.

3.3.2 Simple Wikipedia pipeline

In a lot of cases, our model was unable to distinguish between excerpts that referred to the actual animal and cultural references to that animal. For instance, when asked the question '*Does your animal [cheetah] still exist?*', the following text excerpt is extracted from our cheetah wikipedia corpus with a very high confidence score:

> *The Bill Thomas Cheetah American racing car, a Chevrolet-based coupe first designed and driven in 1963, was an attempt to challenge Carroll Shelby's Shelby Cobra in American sports car competition of the 1960s era. Because only two dozen or fewer chassis were built, with only a dozen complete cars, the Cheetah was never homologated for competition beyond prototype status; its production ended in 1966.*

Based on this excerpt, the yes/no model answers '*No*', indicating that cheetahs are extinct, which immediately throws off the Akinator and it starts thinking of types of dinosaurs. However, we do not wish to completely disregard cultural references - these excerpts are helpful when questions such as '*Is there a car named after your animal?*' are posed by

```
scores = fetch_top_n_nq("Cheetah", "Is it fast?")
pprint([(sc[0].item(), sent) for sc, sent in sorted(scores, key=lambda x: x[0].item(),
```

```
[(0.2990071773529053,
  'The researchers suggested that a hunt consists of two phases—an initial '
  'fast acceleration phase when the cheetah tries to catch up with the prey, '
  'followed by slowing down as it closes in on it, the deceleration varying by '
  'the prey in question.'),
 (0.28799039125442505,
  'Its light, streamlined body makes it well-suited to short, explosive bursts '
  'of speed, rapid acceleration, and an ability to execute extreme changes in '
  'direction while moving at high speed.'),
 (0.10450960695743561,
  '== Interaction with humans ==\n'
  '\n'
  '\n'
  '=== Taming ===\n'
  '\n'
  'The cheetah shows little aggression toward humans, and can be tamed easily, '
  'as it has been since antiquity.'),
 (0.08533799648284912,
  "Hussein, An Entertainment, a novel by Patrick O'Brian set in the British "
  'Raj period in India, illustrates the practice of royalty keeping and '
  'training cheetahs to hunt antelopes.'),
 (0.06664007902145386,
  'Opponents stated the plan was "not a case of intentional movement of an '
  'organism into a part of its native range".')]
```

Figure 2: Sample results using improved pipeline

the Akinator. To avoid confusing our pipeline with such cultural references that do not directly relate to the animal in general, we ask the same question to the Simple Wikipedia corpus for our animal, and append the answer we get from here to the answer excerpt we get from the original Wikipedia article. If the average text confidence scores for the Simple Wikipedia and original Wikipedia articles is less than one standard deviation of the average negative sample scores on the same question, the pipeline outputs 'idk' as a response. Otherwise, we output yes/no based on our boolean answer model prediction on a combination of text answers from Simple Wikipedia and original Wikipedia.

3.3.3 Detecting comparisons

For certain questions such as '*Is your animal smaller than a human?*' or '*Is your animal bigger than your hand?*', the model requires real world knowledge to provide accurate answers - *How tall is a regular human?* and *How big is an average human hand?*. Handling such cases is challenging and beyond the scope of this project. However, to mitigate the consequences of answering these questions incorrectly, we inspect the question for '*comparison*' words included in NLTK's `comparative_sentences` dictionary, such as '*smaller*', '*shorter*', etc. If the question contains such comparison words, the pipeline outputs an 'idk' response. If a correct answer to this question was expected to boost the probability of our animal in the Akinator's guess list, it might reduce the probability by a bit, but not as dramatically as an incorrect answer would lower the probability.

3.3.4 Converting answer excerpts to Yes/No

Multilayer Perceptron Classifier For Report 1, we designed a baseline model for this classification task. We trained a Multilayer Perceptron Classifier model on the BoolQ dataset [4] to predict a Yes/No answer, given a question, and an answer excerpt from a passage. Each question and answer excerpt was first converted to an embedding vector by computing the GloVe embeddings of each token and averaging these over all the tokens. NLTK's TweetTokenizer [18] was used for word tokenization. The average of the question and excerpt embedding was then performed to obtain a semantic embedding representing the QnA phase, which was passed as an input to our classifier. The results of this model are shown in Figure 4.

DistilBERT From our results we see that the model has a low F1 score for prediction of *No*. For Report 2, we improved upon the Multilayer Perceptron Classifier model by architecturing an entailment model and fine-tuning it on the BoolQ dataset. The authors of the BoolQ paper observed their best performance by using the pretrained BERT-large

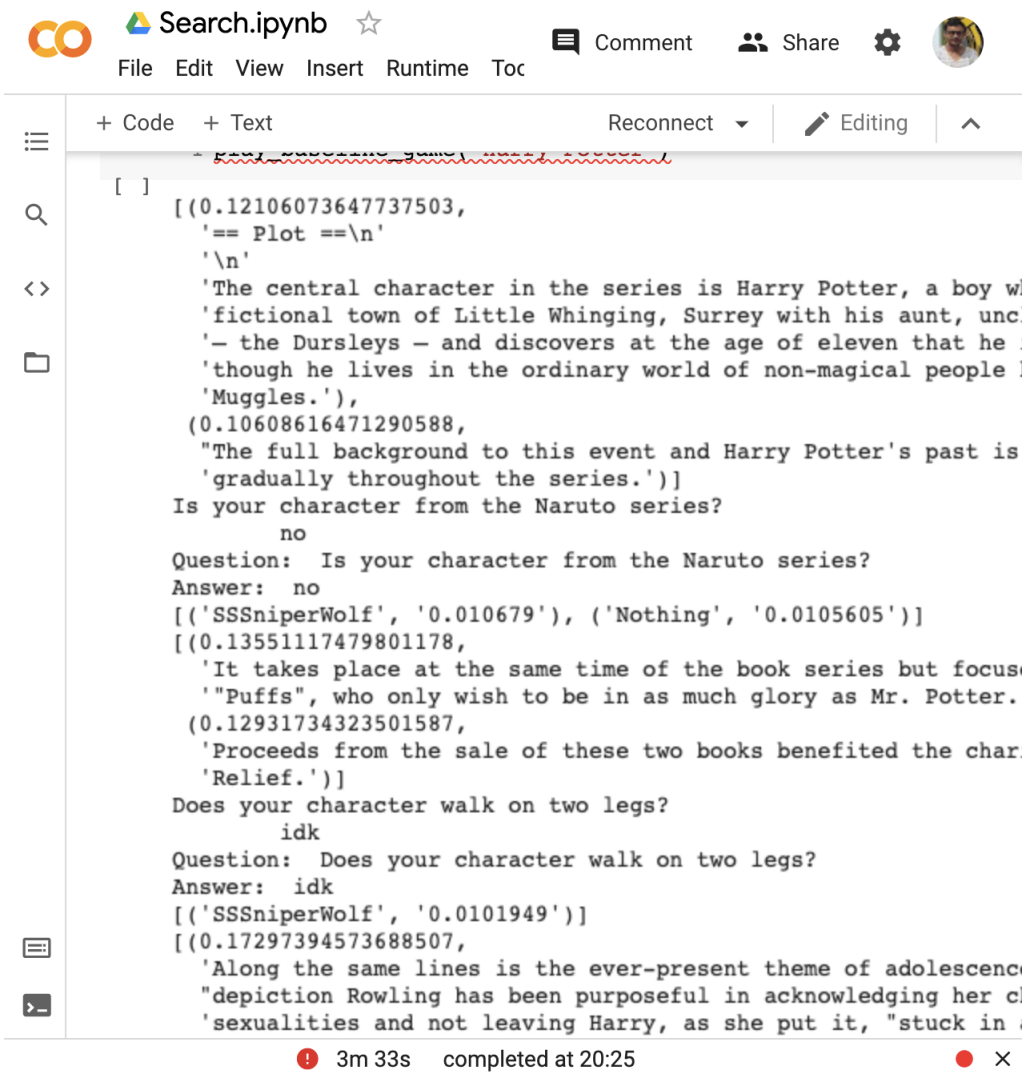

Figure 3: Answer excerpts extracted for fictional character Harry Potter

	precision	recall	f1-score	support
0	0.49	0.20	0.28	1237
1	0.64	0.88	0.74	2033
accuracy			0.62	3270
macro avg	0.57	0.54	0.51	3270
weighted avg	0.59	0.62	0.57	3270

Figure 4: Baseline results obtained after converting answer excerpts to Yes/No labels for the BoolQ dataset

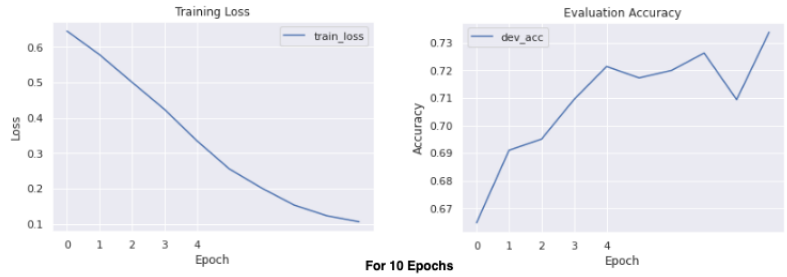

Figure 5: Training loss and Dev accuracy after fine-tuning on DistilBERT for 10 epochs

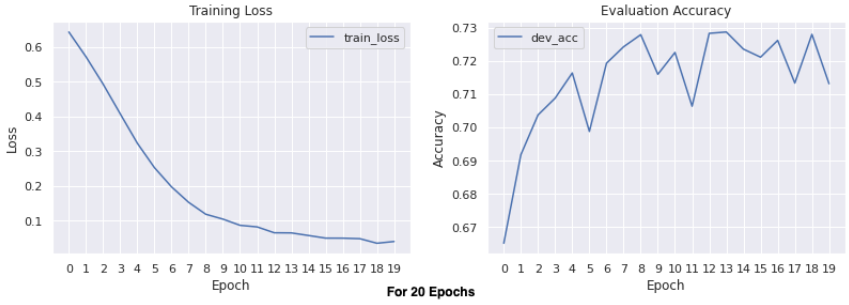

Figure 6: Training loss and Dev accuracy after fine-tuning on DistilBERT for 20 epochs

transformer model and fine-tuning it on their dataset. For our model, we experimented with DistilBERT - a lighter version of BERT with 97% of its language understanding capabilities and 60% faster. To train this model, we utilized its SequenceClassification model with batch size of 32, learning rate of 10^{-5} and Adam optimization for stochastic gradient descent with gradient clipping. This model was fine-tuned on the BoolQ dataset. We trained it for 3 different epochs - 5 (35 minutes), 10 (110 minutes), 20 (230 minutes), and observed that 5 epochs severely overfitted on "Yes" response. However, 10 epochs reduced the overfitting, decreased the training loss to nearly 10% and provided a dev accuracy of 73.3%. Moreover, with 20 epochs, we experienced a severe overfitting on BoolQ with the model having difficulty converging due to a high learning rate. The figures detailing the same are figures 5 and 6.

RoBeRTa-base For our model, we also experimented with RoBeRTa-base transformer which is an improvement over the BERT-large transformer, as it performs dynamic masking with 500 thousand optimization on batch sizes of 8000 (for comparison, BERT has batch size of 256), and is pretrained on 160 GB of data. It removes the next-sentence prediction as seen in BERT, and trains each batch over longer sequences of data. We kept the learning rate and Adam Optimization same as our DistilBERT implementation. After fine-tuning the RoBeRTa-base transformer on BoolQ for 20 epochs (with batch size of 32), we noticed a training loss of 4% and development-set accuracy of 80.7%. This is a significant improvement from the DistilBERT implementation which consisted of a development-set accuracy of 73.3%. Figures 7 and 8 display the training loss and development-set accuracy for 5 epochs.

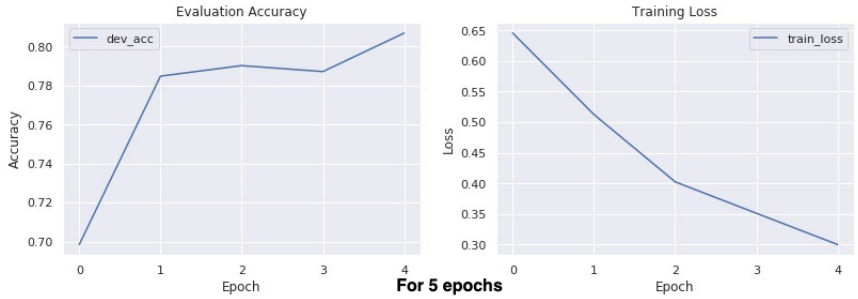

Figure 7: Training loss and Dev accuracy after fine-tuning on RoBeRTa for 5 epochs

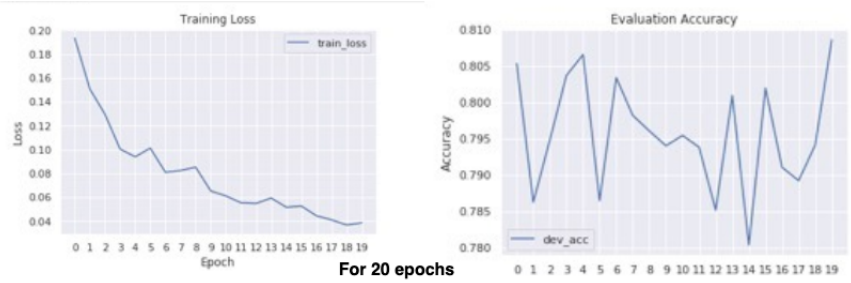

Figure 8: Training loss and Dev accuracy after fine-tuning on RoBeRTa for 20 epochs

3.3.5 Negative Sampling

Usually, if an animal does not possess a certain characteristic, it is not mentioned in the Wikipedia article for that animal. In such cases, the results obtained by BM25 and cosine similarity might be misleading. Despite computing similarity scores for the relevant answers, the threshold determining which answer is appropriate for the question could be difficult to determine. For instance, if the entity is a *cheetah* and we want to find out if *it is an animal that can be used in shows*, the most relevant answer from the Wikipedia article for *cheetah* is *The cheetah has been widely portrayed in a variety of artistic works.* However, this does not answer the original question in the sense in which it was asked. To tackle this challenge, if we do not get *Yes* as an answer to our question on the correct animal, we propose a negative sampling technique where we design a taxonomy of animals and select one entity at random from each broad category and treat these as negative samples to our model. The taxonomy uses a sample of well-known animals from ten broad categories - amphibians, birds, carnivores, domestic, fish, herbivores, invertibrates, mammals, primates and reptiles. We ask the same question with respect to all these negative samples and select the top-most ranking answer excerpts for each animal. We then compare the scores of these top answers with our current animal.

If the score of a negative sample is more than one standard deviation of that of our top answer (for the correct animal), it reflects low probability of finding the answer in the Wikipedia file for our correct animal. In this case, we check if the score of our top answer is within one standard deviation of the mean score of all the negative samples considered - if not, it will indicate that the score for our top answer is really low and there is no mention of the answer in the Wikipedia file, which would mean that the model doesn't know the answer and should output *idk*. Otherwise, if the top answer score is not within one standard deviation of the best negative sampling score but more than the mean score, we output *probably yes* if the BoolQ yes/no answering model outputs *yes* or *probably no* if the yes/no answering model outputs *no*. An example of how this works is shown in Table 1. An example game excerpt incorporating negative sampling with the BoolQ outputs is shown in Figure 9.

3.3.6 Training improved Yes/No model using negative samples

To further leverage answers extracted from the randomly selected negative samples, we hand-annotated 250 questions asked by the animal, for a list of 15 animals. We recorded the yes/no answers generated by our automated question answering pipeline, as well as the text score statistics (average, best and standard deviation) of the negative samples on the same question. We tried to train a model that aims to identify situations where the initial yes/no answer must be modified if the negative sample scores hint that the answer may not be present in the corpus for our animal. Given the limited number of hand-annotated samples, we used simple models like MLP, SVC and decision trees. However, most of the yes/no answers (>78%) matched with the human annotations and did not need any correction, resulting in the model overfitting on the initial yes/no answer and not utilizing the negative sample score statistics to make an improved prediction. We believe that an increased number of hand-annotated samples will improve the predictive performance of such models and can be incorporated as an improvement step after obtaining the initial yes/no answer from the model.

3.3.7 Detecting and fixing a detour

Based on our experiments, we noticed that the Akinator's guess list is extremely volatile and sensitive to all answers. Even if the correct animal shows up in the guess list with the highest probability, the answer to the immediate next question can reduce its probability drastically, to the point of it getting eliminated entirely from the guess list. Fortunately, the Akinator has a weak long term memory, giving more importance to recent answers. This helps the Akinator return to animals similar to the correct animal after taking a long detour, and the answer often converges to the correct animal after the Akinator recovers from the detour. However, this might take a long time, and we might hit the maximum

```
Std dev:  0.06904296783987053
Question:  Does your animal have paws?
Answer:  yes
[('a Berger Picard ', '0.0663456'), ('a cat', '0.0595841'), ('a Siberian Husky', '0.0365824'), ('a dog', '0.0333202'), ('a Labrador Retriever', '0.0294264'), ('a Germa
owl', '0.017844'), ('a Golden Retriever ', '0.0164499'), ('a calico cat', '0.0151491'), ('a fox', '0.013894'), ('a Ragdoll cat', '0.0138902'), ('a wolf', '0.0129893'),
'0.0122679'), ('an orange tabby cat', '0.0121557'), ('a koala', '0.0109514'), ('a Lop rabbit', '0.0100982')]
Does cheetah bark?
text_score:  0.2076128125190735
Computer's reply:  ('yes', 0.98, 0.02)
Purring: Similar to purring in domestic cats but much louder, it is produced when the cheetah is content, and as a form of greeting or when licking one another.
Best score:  0.4324566125869751
Avg score:  0.2222099658101797
Std dev:  0.10362278423383109
Question:  Does your animal bark?
Answer:  probably not
[('a cat', '0.0680975'), ('a Siberian Husky', '0.0313777'), ('a Berger Picard ', '0.0292269'), ('a dog', '0.0285797'), ('a Labrador Retriever', '0.0252399'), ('a Germa
'0.0202276'), ('a wolf', '0.0176151'), ('a Ragdoll cat', '0.0158748'), ('a Golden Retriever ', '0.0141096'), ('a red panda', '0.0140357'), ('an orange tabby cat', '0.0
Pseudechis porphyriacus', '0.0133325'), ('a red fox', '0.0128323'), ('a koala', '0.0125161'), ('a Lop rabbit', '0.0115411'), ('a brown wood owl', '0.0115339'), ('a cal
'0.0105384'), ('a pug', '0.0105226'), ('a cheetah', '0.010375'), ('a Munchkin', '0.0102963')]
Is cheetah a pet?
text_score:  0.33219027519226074
Computer's reply:  ('no', 0.01, 0.99)
=== Diet and hunting ===

The cheetah is a carnivore that hunts small to medium-sized prey weighing 20 to 60 kg (44 to 132 lb), but mostly less than 40 kg (88 lb).
Best score:  0.42574453353881836
Avg score:  0.24759238668613964
Std dev:  0.09844582139192697
Question:  Is your animal a pet?
Answer:  no
[('a fox', '0.0545106'), ('a wolf', '0.0474702'), ('a red panda', '0.0378243'), ('a koala', '0.0337291'), ('a cheetah', '0.0279592'), ('a lion', '0.0265503'), ('a Berg
'0.016728'), ('a Tasmanian Devil', '0.0159528'), ('a thylacine / Tasmanian tiger', '0.0156951'), ('a raccoon', '0.0155667'), ('a Royal Bengal tiger', '0.015544'), ('a
'0.0131453'), ('a platypus', '0.0126553'), ('a jaguar', '0.0104103'), ('a fennec fox', '0.0102809'), ('a tiger', '0.0101235'), ('a capybara', '0.0101062')]
Does cheetah eat other animals?
text_score:  0.3681190311908722
Computer's reply:  ('yes', 0.99, 0.01)
Habitat loss is caused mainly by the introduction of commercial land use, such as agriculture and industry; it is further aggravated by ecological degradation, like bu
Best score:  0.5008896589279175
Avg score:  0.37292628318947907
Std dev:  0.10104970885576776
Question:  Does your animal eat other animals?
Answer:  yes
[('a fox', '0.093922'), ('a wolf', '0.0878034'), ('a cheetah', '0.0517148'), ('a lion', '0.0491088'), ('a Royal Bengal tiger', '0.0287511'), ('a hyena', '0.0284334'),
'0.0270428'), ('a Tasmanian Devil', '0.025312'), ('a Berger Picard ', '0.0218667'), ('a jaguar', '0.0192554'), ('a tiger', '0.018725'), ('a brown wood owl', '0.0166242
wolf', '0.0132442'), ('a honey badger / ratel', '0.0127071'), ('a black panther', '0.0124525'), ('a fennec fox', '0.0111686')]
Is cheetah dog-like?
text_score:  0.3509838283061981
Computer's reply:  ('no', 0.18, 0.82)
The cheetah appears to have evolved convergently with canids in morphology and behaviour; it has canine-like features such as a relatively long snout, long legs, a dee
retractable claws.
Best score:  0.3068915605545044
Avg score:  0.2380332483185662
Std dev:  0.04813022610876642
Question:  Is your animal dog-like?
Answer:  no
[('a cheetah', '0.0879142'), ('a lion', '0.0783842'), ('a Royal Bengal tiger', '0.04733'), ('a fox', '0.0417696'), ('a jaguar', '0.0327338'), ('a tiger', '0.0318322'),
Tasmanian Devil', '0.0180035'), ('a honey badger / ratel', '0.0170339'), ('a king cheetah', '0.0147351'), ('a white tiger', '0.0137576'), ('a snow leopard', '0.0130442
brown wood owl', '0.0125929'), ('a crocodile', '0.0112601'), ('an ocelot', '0.0107987'), ('a liger', '0.0106929'), ('a leopard', '0.010136')]
Does cheetah roar?
text_score:  0.22896137833595276
Computer's reply:  ('yes', 0.98, 0.02)
Its light, streamlined body makes it well-suited to short, explosive bursts of speed, rapid acceleration, and an ability to execute extreme changes in direction while
Best score:  0.3808732032775879
Avg score:  0.185155700167848
Std dev:  0.08093202036125588
Question:  Does your animal roar?
Answer:  probably not
[('a cheetah', '0.094175'), ('a lion', '0.062408'), ('a fox', '0.0443121'), ('a Tasmanian Devil', '0.026794'), ('a tiger', '0.0253442'), ('a king cheetah', '0.0223849'
```

Figure 9: Example game for animal cheetah

number of questions (80) after which the Akinator throws an error. We want to detect such detours early without allowing our pipeline to peek into the Akinator's guess list at any given time. This is challenging, given that we do not know where the Akinator's guesses are headed at any given time, and we are not aware of whether our past answers are correct or incorrect. We propose a technique to detect misleading answers using negative sampling results, and bring the Akinator back using positive samples - animals that are most similar to the correct animal.

Negative sampling to detect a detour To judge which animals are similar/dissimilar to the correct animal, we extract the word embeddings for each animal in the negative sampling list and our vocabulary of animals, and compare these with the word embedding for the correct animal. We expect the embeddings for animals such as 'dog' and 'cat' to be more similar to each other and different from 'crocodile' and 'giraffe'. We consider a fixed negative sampling list (sampled randomly) for the entire game. For each question that the Akinator asks, we answer yes/no for all the animals in our negative sampling list, as well the correct animal. We store W most recent yes/no answers for all animals in the negative sampling list and for the correct animal. After answering each question, we check to see if our last W yes/no answers have been too similar to an animal in the negative sampling list that is very dissimilar to the correct animal. If so, we report a detour.

Fixing a detour If we detect a detour, we inspect our animal vocabulary to identify N animals that are most similar to the correct animal. We call this our positive sampling list. We answer the next question with a majority yes/no vote from these positive samples. We do not answer every question this way because the Akinator is not likely to converge to the correct animal if we answer specific questions such as '*Does your animal have spots?*' incorrectly. Once we have fixed a detour, we empty the past W yes/no answers list for all animals in the negative samples - we do not want to apply this technique too early.

Entity name	Sentence	Probability	Positive Sample?
Cheetah	They have been widely depicted in art, literature, advertising, and animation.	0.17	Yes
Cheetah	An open area with some cover, such as diffused bushes, is probably ideal for the cheetah because it needs to stalk and pursue its prey over a distance.	0.10	Yes
Dog	In conformation shows, also referred to as breed shows, a judge familiar with the specific dog breed evaluates individual pure-bred dogs for conformity with their established breed type as described in the breed standard.	0.26	No
Dog	In 2015, a study found that pet owners were significantly more likely to get to know people in their neighborhood than non-pet owners.Using dogs and other animals as a part of therapy dates back to the late 18th century, when animals were introduced into mental institutions to help socialize patients with mental disorders.	0.17	No
Frog	It is typically used when the frog has been grabbed by a predator and may serve to distract or disorient the attacker so that it releases the frog.	0.19	No
Frog	Frogs are used for dissections in high school and university anatomy classes, often first being injected with coloured substances to enhance contrasts among the biological systems.	0.15	No
Penguin	Several species are found in the temperate zone, and one species, the Galápagos penguin, lives near the Equator.	0.11	No
Penguin	In the 60s Batman TV series, as played by Burgess Meredith, he was one of the most popular characters, and in Tim Burton's reimagining of the character in the 1992 film Batman Returns, he employed an actual army of penguins (mostly African penguins and king penguins).	0.09	No
Snail	Snails have considerable human relevance, including as food items, as pests, and as vectors of disease, and their shells are used as decorative objects and are incorporated into jewelry.	0.15	No
Snail	Land snails are known as an agricultural and garden pest but some species are an edible delicacy and occasionally household pets.	0.11	No

Table 1: Negative Sampling

4 Model Evaluation

We use accuracy, recall, precision and F1 score on the BoolQ test set as the evaluation metric for the submodel used to convert extracted answers to *Yes*/*No*. We can evaluate the submodels on pre-existing benchmarks. GLUE [19] contains several tasks such as similarity, paraphrasing and inference tasks, and can be used to evaluate the quality of sentence embeddings used in our model. SuperGLUE [20] can be used to test our question answering model. QNLI [19] dataset can be used to determine whether our selected answer excerpt contains the answer to the question posed by the Akinator. WNLI [19] can be used to evaluate our model's anaphora resolution performance, if we include this as a component of our final model.

We hand-annotated answers to 250 questions and compared these answers to the yes/no outputs of our pipeline. The answers matched with an accuracy of 78.69% and F1 scores 81.67% (class no) and 74.61% (class yes). Since it requires a lot of manual effort to hand-annotate these answers and play long games with the Akinator, we devised an approximate answering technique that guesses the correct yes/no answer for each question. This technique can only be applied to answers that result in the correct animal appearing in the Akinator's guess list. If the probability of the correct animal in the Akinator's guess list increases after answering a question, we estimate that answer to be a correct answer (correct answer equals the pipeline's output). Otherwise, we mark the answer as incorrect (correct answer is the opposite of the pipeline's output). Using this estimated correct answer, we labeled another 264 question-answer pairs that were automatically generated in games with the Akinator. 62.88% questions were answered correctly, considering

the expected correct answer as the ground truth. However, there might be inconsistencies in this ground truth. For instance, there have been instances of cheetahs being tamed in human history, and a cheetah is technically not able to roar - but the Akinator reduces the probability of 'cheetah' when it asks these questions and the pipeline answers correctly based on the Wikipedia article. So a probability reduction in the guess list may not always be indicative of an incorrect answer. The Akinator, at the end of the game, asks for the actual answer if it fails to identify the entity that the user had in mind, suggesting that it updates its knowledge by some sort of crowdsourcing, which may result in these anomalous results.

For further evaluation, we designed a couple of metrics - number of questions it takes the akinator to reconsider the correct animal after being thrown off by an incorrect answer (*detour recovery time*) and the best probability of the animal in the Akinator's guess list over the entire game (*best guess probability*).

Detour recovery time We measure the time (measured by the number of questions) taken by the Akinator to recover from an incorrect answer that knocks off the correct animal from the guess list to the point where it is reintroduced in the Akinator's guess list. Evaluating on our automated game results, we observe an average span of approximately 8 questions before the Akinator is able to come back on track. This gives us an intuition of how fast the model is able to redirect the Akinator's focus - the lesser the detour recovery time, the better. A longer detour recovery time would indicate that the pipeline has answered incorrectly multiple times in succession, which might cause the Akinator to drift further away from the actual answer. The Akinator is able to come back on track eventually most of the time because it does not seem to have a strong long term memory and focuses more on recent answers.

Best guess probability We record the highest probability with which the correct animal features in the Akinator's guess list over the course of a single game. The average best guess probability for an experiment on 15 animals was 25.91%. This is a relatively high probability, given that most of the times when the Akinator considers an animal in the guess list, it starts off with a probability of less than 1%.

We propose an additional metric for future implementation to get a better understanding of our pipeline's performance - convergence rate. This metric could consider the initial probability (from the time that the correct animal shows up in the Akinator's guess list) and the final probability (highest probability achieved by the Akinator for the correct animal), and the rate of this increase over the number of questions asked between the initial and final probability timestamps. If the correct animal disappears from the Akinator's guess list, the convergence rate metric would be reset to zero. If an item does not converge, the convergence rate for that game would be zero.

5 Limitations

As we defined a new problem in NLP and provided preliminary results for the same, we observed some significant shortcomings in the problem-definition, current state of transformer models, our primary dataset BoolQ, using Wikipedia as our primary corpus, and limitations of word2vec models. While working with general entities, our baseline models failed to understand subtleties as it seemed to require a vast amount of global information to decisively answer 'no'. Hence, to make the problem tractable, we modified the problem definition to only include animal names as our 'guess' words. Furthermore, the transformer models we worked with - DistilBERT and RoBeRTa - showed difficulty in performing comparison and counting tasks. For example, our model would often fail when presented with questions such as 'Is it smaller than a monkey?' (comparative type) and 'Does it have 8 legs?' (counting type). While a human can visually comprehend such tasks, it becomes difficult to find such sentences in a corpus which can validate the presence of such sentences. Moreover, we believe as a future-scope in the Computer Vision domain, one can include a multimodal pipeline which combines ours with one that performs question-answering by observing an image.

Another limitation of the transformer model is that the negative results are hard to guess - as mentioned in the BoolQ paper [4], the subtlety of negation lies in understanding that 'a positive assertion in the text excludes, or makes unlikely, a positive assertion in the question'. As mentioned in RoBeRTa and DistilBERT sections, another limitation we observed was overfitting during our finetuning on the BoolQ dataset.

We observed that while BoolQ dataset is modelled to solve a yes/no problem, the subtleties between their and our problem definitions add up significantly. For instance, almost all of our questions start with the word 'is', however, more than 50% of our training data (5234 examples) consists of questions not starting with 'is'. Furthermore, as mentioned before, many 'animal' related questions required prior knowledge of other animals to answer correctly - however, the training corpus was largely devoid of questions from our problem domain. We also observed that both the Spacy and Gensim word2vec models had difficulty understanding the relationship between an animal and its parent class - for example, a 'tiger' had a higher correlation with a reptile, than with a carnivore or a mammal. This made it significantly difficult to perform positive sampling, requiring us to utilize UCI's zoo dataset [21] for obtaining the parent-child

Animal	Coreference resolution	Excerpt extracted	Probability
Cheetah	No	They have been widely depicted in art, literature, advertising, and animation.	0.17
Cheetah	No	An open area with some cover, such as diffused bushes, is probably ideal for the cheetah because it needs to stalk and pursue its prey over a distance.	0.10
Cheetah	Yes	Generally, the female can not escape on her own; the males themselves leave after they lose interest in her.	0.41
Cheetah	Yes	== Interaction with humans ==\n\n\n=== Taming ===\n\n,The cheetah shows little aggression toward humans, and can be tamed easily, as it has been since antiquity.	0.41
Monkey	No	Some are kept as pets, others used as model organisms in laboratories or in space missions.	0.24
Monkey	No	They are used primarily because of their relative ease of handling, their fast reproductive cycle (compared to apes) and their psychological and physical similarity to humans.	0.16
Monkey	Yes	The most common monkey species found in animal research are the grivet, the rhesus macaque, and the crab-eating macaque, which are either wild-caught or purpose-bred.	0.49
Monkey	Yes	Some are kept as pets, others used as model organisms in laboratories or in space missions.	0.45
Elephant	No	In the past, they were used in war; today, they are often controversially put on display in zoos, or exploited for entertainment in circuses.	0.26
Elephant	No	It can be used for delicate tasks, such as wiping an eye and checking an orifice, and is capable of cracking a peanut shell without breaking the seed.	0.13
Elephant	Yes	=== Zoos and circuses ===\n\nElephants were historically kept for display in the menageries of Ancient Egypt, China, Greece, and Rome.	0.50
Elephant	Yes	In the past, they were used in war; today, they are often controversially put on display in zoos, or exploited for entertainment in circuses.	0.44

Table 2: An example showing the Coreference Resolution Dilemma

relationships for positive/negative sampling. Lastly, we would like to stress that in spite of the vast sea of resources in Wikipedia articles, we found many instances in which both the Simple Wikipedia and Full Wikipedia were unable to find a relevant sentence. For example, while tigers can swim well, their Wikipedia article has no such reference of it, which in turn confuses our model which is dependent upon a strong reference to base its answer on.

6 Applying in practice

The biggest prerequisites to apply this problem in practice would be to fine-tune the yes/no model on a transformer trained on a larger dataset such as GPT-2 (which has 1.5 billion parameters and was trained on a dataset of 8 million web pages) [22]. Another prerequisite would be to build a vast taxonomy to improve the performance of the positive/negative sampling stages of the pipeline. We also propose using a hybrid corpus consisting of answers from Wikipedia and domain-specific knowledge graphs. We observed that knowledge graphs such as DBPedia [16] heavily borrowed their content from Wikipedia, making it less effective for this task. Moreover, if the domain problem requires a broader category of entities, we highly suggest creating a custom dataset for your task, instead of overly relying upon BoolQ due to its limitations (as mentioned in the Limitations section). Lastly, if one expects the questions to include more than one pronoun, we encourage building a pronoun resolution model - starting with a baseline model (like the Hobbs' algorithm) [23] and eventually experimenting with Google's GAP dataset [24].

7 Future Work

It would be helpful if we could detect questions that require real world knowledge to answer. These questions are often in the form of comparisons to other objects/animals such as *Is your animal bigger than a human*? As future work, it would be interesting to identify questions that present a comparison-type query and answer these questions with an *idk* to avoid confusing the model with confident but incorrect answers. The original Akinator tends to guess *a guy who answers randomly* if the model answers *idk*, *probably* or *probably not* too many times. We could maintain a penalty for such answers that increases every time the model outputs an uncertain answer and decreases with every definite answer that the model outputs.

References

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

[2] Yinhan Liu et al. "Roberta: A robustly optimized bert pretraining approach". In: *arXiv preprint arXiv:1907.11692* (2019).

[3] Alec Radford et al. "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8 (2019), p. 9.

[4] Christopher Clark et al. "BoolQ: Exploring the surprising difficulty of natural yes/no questions". In: *arXiv preprint arXiv:1905.10044* (2019).

[5] Mansfield Tracy Walsorth. *Twenty Questions: A short treatise on the game to which are added a code of rules and specimen games for the use of beginners*. Holt, 1882.

[6] *The Animal Game*. URL: `https://www.animalgame.com/play/faq.php` (visited on 09/11/2021).

[7] Denny Vrandečić and Markus Krötzsch. "Wikidata: a free collaborative knowledgebase". In: *Communications of the ACM* 57.10 (2014), pp. 78–85.

[8] Daniel Jurafsky and James H Martin. "Speech and language processing (draft)". In: *preparation Available from: https://web. stanford. edu/~ jurafsky/slp3* (2021).

[9] Łukasz Kaiser. *Tensor2Tensor Transformers*. URL: `https://nlp.stanford.edu/seminar/details/lkaiser.pdf`.

[10] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. *Introduction to information retrieval*. Vol. 39. Cambridge University Press Cambridge, 2008.

[11] Nils Reimers and Iryna Gurevych. "Sentence-bert: Sentence embeddings using siamese bert-networks". In: *arXiv preprint arXiv:1908.10084* (2019).

[12] Nils Reimers and Iryna Gurevych. *Sentence Transformers: Multilingual Sentence, Paragraph, and Image Embeddings using BERT & Co.* URL: `https://github.com/UKPLab/sentence-transformers`.

[13] Vladimir Karpukhin et al. *Dense Passage Retrieval for Open-Domain Question Answering*. 2020. arXiv: `2004.04906 [cs.CL]`.

[14] Adina Williams, Nikita Nangia, and Samuel Bowman. "A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 1112–1122. DOI: `10.18653/v1/N18-1101`. URL: `https://aclanthology.org/N18-1101`.

[15] Samuel R. Bowman et al. "A large annotated corpus for learning natural language inference". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, Sept. 2015, pp. 632–642. DOI: `10.18653/v1/D15-1075`. URL: `https://aclanthology.org/D15-1075`.

[16] Sören Auer et al. "Dbpedia: A nucleus for a web of open data". In: *The semantic web*. Springer, 2007, pp. 722–735.

[17] *akinator.py*. URL: `https://github.com/NinjaSnail1080/akinator.py`.

[18] Edward Loper and Steven Bird. "Nltk: The natural language toolkit". In: *arXiv preprint cs/0205028* (2002).

[19] Alex Wang et al. "GLUE: A multi-task benchmark and analysis platform for natural language understanding". In: *arXiv preprint arXiv:1804.07461* (2018).

[20] Alex Wang et al. "Superglue: A stickier benchmark for general-purpose language understanding systems". In: *arXiv preprint arXiv:1905.00537* (2019).

[21] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2017. URL: `http://archive.ics.uci.edu/ml`.

[22] Alec Radford et al. “Language models are unsupervised multitask learners”. In: *OpenAI blog* 1.8 (2019), p. 9.

[23] Jerry R Hobbs. “Resolving pronoun references”. In: *Lingua* 44.4 (1978), pp. 311–338.

[24] Kellie Webster et al. “Mind the gap: A balanced corpus of gendered ambiguous pronouns”. In: *Transactions of the Association for Computational Linguistics* 6 (2018), pp. 605–617.