Natural Language Processing 2022: Final Project

Group name: The Pythons

André Oskar Andersen
wpr684@alumni.ku.dk

Sebastian Hammer Eliassen
fsp585@alumni.ku.dk

Abstract

In this paper we implement multiple question answering models. This includes models that can classify whether a text contains the answer to a question, and also models that can extract said answer. This is done by implementing various logistic regression models, neural network and by finetuning state-of-the-art language models. Additionally, we analyse the dataset via these models to interpret how it works. All of the models ended up succeeding in both the classification and token classification tasks.

1 Introduction

The following paper is our final project in the course *Natural Language Processing* taught at the University of Copenhagen in 2022. The aim of the project is to create a multilingual question answering system. More specifically, the project will be focusing on English, Finnish and Japanese. The dataset used will be the TyDi QA dataset (Clark et al., 2020)

2 Introduction to NLP

2.1 Preprocessing and Dataset Analysis

Before we can analyse the data, we need to perform some preprocessing of the data, as well as become more familiar with the data.

2.1.1 Preprocessing of the Data

In order to tokenize the texts, we first make use of the `huggingface_hub` to download the TyDi QA dataset. We then filter the dataset, such that we only have entries which have English, Japanese and Finnish as their language. We then use `spacy` (Honnibal and Montani, 2017) to tokenize the data at word level, keeping in mind their corresponding language. We decided not to perform further preprocessing of the data, as this could essentially remove important information, that could have an impact later on.

2.1.2 Overview of Common Words

To further familiarise ourselves with the data, we have, in Table 1, given an overview of the most common tokens at the beginning and end of the questions (excluding the "?"-token), for each language, as well as the frequency of these tokens.

As we can see on table 1, English and Finnish has a tendency to put "question words" like "what" and "when" at the start of the sentence, while Japanese tend to put these at the end.

Code	Pos.	Token	Translation	Freq.
en	First	When	When	2242
		What	What	2103
		How	How	1296
	Last	Born	Born	342
		Founded	Founded	204
		Die	Die	122
fi	First	Milloin	When	3519
		Mikä	How	2328
		Missä	Where	1646
	Last	Syntyi	Born in	1072
		On	Is	723
		kuoli	Died	720
ja	First	日本	Japan	392
		『	"	306
		アメリカ	America	106
	Last	た	Did you	1507
		か	Is it	1299
		は	What	894

Table 1: Overview of first and last tokens. Here "Code" indicates the language code, and "Pos." indicates the position. "Freq." indicates the frequency of the token at the given position.

2.2 Binary Question Classification

Our first machine learning task will be to use features based on the question and context to classify

Language	Training Accuracy	Validation Accuracy
English	.748	.713
Finnish	.749	.714
Japanese	.732	.676

Table 2: Training and validation accuracy of the feature-based classifier.

whether or not the answer to the question is given in the context.

2.3 Choice of Setup

We first extract the tf-idf vector for each tokenized question and tokenized context for all articles. Due to memory-limits, we limit this to the 300 most frequent terms for questions and contexts, respectively. We use tf-idf since it uses a combination of how often each term appears in the instance, but also how specific that term is, which means that we weigh terms that are less frequent in the entire corpus higher. This is not the same as stopword-removal since we still keep some information from the common terms. This should yield a good representation of the instance. Lastly, we concatenate the two vectors of length 300 for each article to a single vector of length 600. To perform the classification-task we use logistic regression, as this is a linear method that still yields decent results, which thus should work as a decent baseline.

2.4 Results

Table 2 illustrates the training and validation accuracy of the three developed models. We decided to only report the binary accuracy and not metrics like F1, recall or precision, as the dataset is exactly balanced. Here we can see, that the model performs similarly on English and Finnish, and a bit worse on Japanese. This could be because, the tokenizer does not perform well on Japanese text. We can further see, that the validation accuracy is significantly above 50% for all three languages, suggesting that there is a pattern in the data to be found.

3 Representation Learning

To further improve the results of our binary-classifiers we have developed various models for each language, which use features based on continuous vector representations of words, in addition to the linguistic/lexical features from section 2.

3.1 Implementation of the Classifiers

We implement two different models: (1) a logistic regression, and (2) a neural network. The neural network consists of two dense layers, where the first layer is followed by a *ReLU* activation-function. The first layer has a dimensionality of H_{in}, where H_{in} is the size of the input length, the second layer has a dimensionality of 150, which outputs a single scalar.

The neural network is optimized by minimizing the `BCEWithLogitsLoss` from PyTorch (PyTorch), via the Adam Optimizer with a learning-rate of $5 \cdot 10^{-5}$. The network fitted for $10,000$ epochs and the best-performing model is picked for evaluation.

3.2 Encoding the Data

For the fusion based vector representations, we simply concatenate the tf-idf vectors from section 2.3 with the continuous vector representations that we just created, such that we have $H_{in} = 1200$.

We make the continuous vector representation by using *BPEmb* (Heinzerling and Strube, 2018) with a vocabulary size of 25.000 on the tokenized data from section 2. The encoding is done by considering the document text and question text separately. Each word is then encoded to a vector of 300 dimensions. We use mean pooling to encode each instance into one vector. For words that are not in the vocabulary of *BPEmb* we use the mean vector of the vocabulary of *BPEmb*. We concatenate the document text and question text vectors to form the $H_{in} = 600$ dimensional vector-representation of the document-question-pair.

3.3 Results

The training and validation accuracies of the 12 models have been summarized in Table 3. Looking at the table we can see, that the neural network always outperforms the logistic regression. This is most likely due to the fact, that the neural network has more parameters than the logistic regression, as well as the ability of the neural network to approximate non-linear functions.

We can also see that the fusion models outperform both the continuous vector based models, as well as the tf-idf models. This suggests

Fusion	Training			Validation		
	English	**Finnish**	**Japanese**	**English**	**Finnish**	**Japanese**
Logistic Regression	.767	.762	.778	.724	.721	.711
Neural Network	.982	.852	.100	.743	.735	.772
Continuous	**Training**			**Validation**		
	English	**Finnish**	**Japanese**	**English**	**Finnish**	**Japanese**
Logistic Regression	.707	.702	.723	.668	.657	.695
Neural Network	.915	.804	.946	.720	.699	0.762

Table 3: Accuracy of the developed classifiers based on continuous vector representations of words. "Continuous" refers to the models that only use continuous word representations. "Fusion" refers to models that use both continuous word representations and the linguistic/lexical features.

that our models can make use of both lexical and continuous vector representations of words. Something to note is that tf-idf also contains information about the other articles, since it weighs specific words higher, which *BPEmb* encodings do not. This added information could also help improve the results.

4 Language Modelling

We further extend our classifiers by extracting sentence representations from monolingual neural language models. All of the neural language models are based on GPT2, as GPT2 tends to yield stat-of-the-art results (Radford et al., 2019) (nlp waseda) (Finnish-NLP).

4.1 Finetuning of Language Models

We finetune the three language models, as this should give better results. The finetuning of the three language models is done by using Huggingface's Trainer API (Huggingface). The models are fitted for 5 epochs, by using the default optimizer and loss-function, a learning rate of $2 \cdot 10^5$ and a batch size of 8.

4.2 Sampling from the Language Models

When sampling from the three language models we get the results illustrated in Table 4. Looking at these sampled sentences we can see, that the English sentences semantically makes the most sense. However, this could be due to incorrect translations of the Finnish and Japanese samples, rather than the model making errors. We can further see, that sentences only contains few grammatical errors. However, this could be because Google translate corrects some of the errors that are in the sampled output, such that the English translation contains fewer errors, than the sampled outputs.

4.3 Evaluation of Language Models

Table 5 illustrates the perplexity of the three finetuned language models, computed from the validation set. We can clearly see, that the perplexities are in the range $26 - 36$, which is about the same as the range reported in the GPT2 paper (Radford et al., 2019), hinting towards us successfully finetuning the models.

4.4 Implementation of Classifiers

We further extend our classifiers by using the sentence representation generated by the language models as input for the classifiers. We follow the same procedure and setup as in Section 3, however, unlike the encodings from Section 3 the encodings from the language models are not of dimension 300, but instead of dimension 768, resulting in us having $H_{in} = 2 \cdot 768$.

4.5 Results

The training and validation accuracies of the six models are illustrated in Table 6. Comparing these results with those from the previous sections we can see, that by using the language models encodings for the data, we get results that outperforms all previous methods, which is probably due to GPT2's more sophisticated way of encoding the data.

5 Error Analysis and Interpretability

We pick two different models to compare for our error analysis. The first model is described in section 2.3 and the second model is described in section 4. We denote the first model as `m1` and the second model as `m2`. Assuming an order of English, Finnish and Japanese, we get the same validation accuracy as seen on table 2 for `m1` and a validation accuracy of .80, .81 and .79 for `m2`. This might differ from that of table 6, due to the random weight

Sampled output	English translation
I was meaning to ask them about the issue but I was meaning to learn the story in my head I was meaning to get on a bus, but	
Tarkoitin tietenkin, että kun on aika, Tarkoitin, että koko maa saisi yhden äänen Tarkoitin kyllä sitä, että jos on pakko	I meant, of course, that when the time comes, I meant that the whole country would have one vote I did mean that if you have to
私はするつもりだった。おそらく 私はするつもりだったが、彼ら 私はするつもりだった。そこで	I was going to probably I was going to but they I was going to Therefore

Table 4: Results of sampling from the language models. For English we start the sampling with the sequence "I was meaning to". For Finnish we start the sampling with the word "Tarkoitin", which translates to "I meant". For Japanese we start the sampling with the sequence "私はするつもりだった", which translates to "I was going to". The translations are done by Google Translate.

English	Finnish	Japanese
28.83	35.63	26.68

Table 5: Perplexity of the three finetuned language models, based on the validation set.

initialization of neural networks.

5.1 Error analysis

To analyse the errors of our models, we split each of the samples into 4 classes: samples that both models get correct (`both`), samples that no models gets correct (`none`), the ones that only `m1` gets correct (`m1`) and the ones that only `m2` gets correct (`m2`). The results of this can be seen on table 7. As predicted, the `m2` class is significantly larger than the `m1` class. We should note that all of the splits have an equal amount of negative and positive ground truths, additionally both `m1` and `m2` have a mean prediction between .49 and .52. This means that both models are similar in the sense that they, do capture the balance of the dataset. For this reason we only report accuracy, as explained in subsection 2.4.

We also analysed whether or not the starting position of the answer in the text had an effect of how well the model predicted said answer. Here we can see on figure 1 that the `none` class is more prevalent when the answer is in the outer parts of the text, while the opposite applies to the `both` class. We can also see that the `m1` class struggles with texts in the middle, while the `m2` does not do this as much.

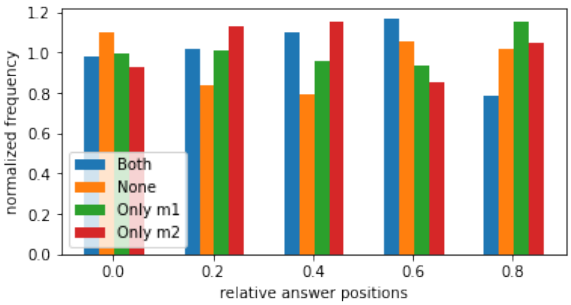

Figure 1: Normalized frequency of the different classes as a function of where the answer starts in the text. Here normalized means that each of the bins, according to the x-value has an average value of 1 and each of the classes also has the same total integral.

5.2 Explainability of best model

To explain what makes our model make correct and incorrect predictions, we try to find the so called "easy" and "hard" tokens. We first collect the articles into those that `m2` got correct, and those that it got incorrect. We then remove common stopwords from both collections. After that we look at the 100 most frequent tokens of both collections. Lastly we then look at the difference between frequent tokens of the easy articles and the frequent tokens of the hard articles, to find the easy token. We do the opposite to find the hard tokens.

By looking at the easy tokens, we can see words like "*longest*," and "*largest*," hinting to the fact that it is easy for the model to classify questions or answers that include superlatives. For hard tokens we can see words like "*difference*"

	Training			Validation		
	English	**Finnish**	**Japanese**	**English**	**Finnish**	**Japanese**
Logistic Regression	.865	.832	.848	.783	.756	.727
Neural Network	.994	.982	.100	.800	.812	.811

Table 6: Accuracy of the developed classifiers based on sentence representations from neural language models.

	`none`	`both`	`m1`	`m2`
Finnish	168 (.524)	1053 (.503)	150 (.540)	315 (.457)
English	120 (.400)	629 (52.5)	77 (.455)	164 (.500)
Japanese	110 (.536)	594 (.498)	106 (.519)	226 (.478)

Table 7: Instance frequency and label mean (parenthesized), as function of error class and language.

and "*example*," hinting to the fact that questions like *"what's an example of...?"* and *"what's the difference between...and ...?"* are rather hard for the model to pinpoint. Which might make sense, since they are not as specific.

5.3 Adversarial instances

We make use of what we just discovered to create some adversarial samples for our models. Here we have 3 easy question and 3 hard questions. To create the answers, we just look them up on Google, and copy the texts. We copy the texts such that they have approximately similar length across both classes to make the test somewhat fair. We also make sure that we have some texts that do not contain the answer, to see if our model can also predict negative labels. The easy questions that we use are *"When did John F. Kennedy die?"*, *"When did John F. Kennedy die?"*, *"What is the largest planet?"* and *What is the longest airplane?* while the hard questions that we use are *"How do I become a cricket coach?"*, *Whats the difference between psychopathy and sociopathy?* and *What is an example of something Denmark is known for?*. The contexts to these questions, in the same order, can be seen in appendix A. We can see the result of running these adversarial examples on the English GPT2 table 8. Here we can see that the easy questions are predicted correct more often than the hard questions, hinting to the fact that our adversarial questions worked.

Difficulty	**Ground Truth**	**Prediction**
Easy	(1, 1, 0)	(1, 1, 0)
Hard	(1, 0, 1)	(1, 1, 0)

Table 8: Ground truth and predictions of the hard and easy questions

6 Sequence Labelling

To perform the actual question answering we make use of a sequence labeller, which we can use to extract the answer to the question from the context. Since the job of the sequence labeller is to find tokens related to the answer of the question, and not to classify whether or not the question is answerable, like the task of the previously developed models, we will only be considering the instances where we know that the context contains the answer to the question.

6.1 Creating the Ground Truth labels

Before we can develop our sequence labeller, we need to make the ground truth labeling of each token in the context. For this we transform the context into its IOB-format. The TyDi QA dataset (Clark et al., 2020) has already tagged every character in the context as being part of the answer or not, so we can use this to tag each token as being part of the answer or not. However, this does raise one problem, illustrated in Table 9: in cases where a tokenizer does not work optimally and two tokens that are not part of the same IOB-class are tokenized as being one token, what should the corresponding resulting token be labelled as? We decided to label the incorrectly tokenized token as being part of the answer, since this token could potentially contain import information, but one could argue, that it should be labelled as not being part of the answer, as it then could contain information that is not part of the answer to the proposed question.

6.2 The Design of the Sequence Labeller

The architecture of the sequence labeller is based on an pretrained version of XLM-RoBERTa (deepset). We decided to use this model as it is

	Ground truth tokenization	Incorrect tokenization
Context	[A, dog, barks, and, it, has, four, legs]	[A, dog, barks, andit, has, four, legs]
IOB-labels	[O, O, O, O, B, I, I, I]	[O, O, O, O/B, B/I, I, I]

Table 9: A ground truth and an incorrect white-space tokenization of the sentence "A dog barks and it has four legs" and its effects on the resulting IOB-labels. Depending on one's approach, one may label the token *andit* as either *O* or *B*, as well as the token *has* having either token *B* or *I*.

very fitting for the task, as it has already been trained on the relevant languages as well as on question-answering data, and generally it delivers state-of-the art results (Conneau, 2020). RoBERTa returns two 1-dimensional vectors of the same length as the context. Each entry in the first vector represents the likelihood of the answer starting at that entry, whereas each entry in the second vector represents the likelihood of the answer ending at that entry.

Our goal is however not just to find the start and end of the answer, but instead to perform token classification. For that reason we stack these two vectors to form a 256×2 matrix and input this to an 1-layered bidirectional LSTM with 256 features in its hidden state, which in turn returns its output to a 256×3 fully connected linear layer. Lastly, row-wise softmax is applied. We decided to use a bidirectional LSTM, as it was suggested by (danielhers).

6.3 Training the Sequence Labeller

Training the sequence labeller happens in two steps. First, just XLM-RoBERTa is fully finetuned on our dataset. This is then followed by freezing XLM-RoBERTa and then fully training only the following layers on our dataset. The sequence labeller is fitted on the three languages seperately, resulting in three monolingual sequence labellers.

The finetuning of XLM-RoBERTa is done by making use of Huggingface's `Trainer`-API (Huggingface). The fitting of XLM-RoBERTa is done with a batch-size of 8, a learning-rate of $2e-5$, and is finetuned for a total of 5 epochs by using the default optimizer and the loss-function.

The training of the final layers uses 3 epochs, a learning rate of $1e-3$ and a batch-size of 1. This is done by using the AdamW-optimizer and the negative log likelihood loss. Since the distribution of the three labels is very imbalanced, we use a class-weight of $25, 1, 40$ for I, O, B, respectively.

6.4 Results

Table 10 presents the performance of the three developed sequence labellers, measured using the F1-score with various settings of the "average"-parameter from the sklearn library (scikit learn). As opposed to the data that we have been testing for previously, we now have a very imbalanced dataset on our IOB tokens. In fact we have a distribution of around 2-3%, 97% and 0.4% for the IOB tokens, respectively. For this reason we cannot just use accuracy to accurately measure the performance of our models, since we could achieve 97% accuracy by just predicting O. Therefore we primarily look at the macro to gauge the performance of our models. Macro gives equal importance to each class, as opposed to micro which gives equal importance to each token. Which means that we also need a good F1 score for the I- and B tokens, to get a good score. By looking at the table, we can see that we get a good score on all of the measures, which means that all of our tokens have been correctly predicted in some regard.

6.5 Beam Search

Table 11 gives an overview of using beam search to select the optimal sequence of labels for various settings of the amount of "beams" to use. For the amount of "beams" to use, we only tried the minimal possible value, the maximum possible value, as well as a value in between, as trying more settings would take too much time. By comparing the results we can see, that only using one "beam" yields the best results. Thus, using beam search does not yield better results than by just using the argmax to pick the label for each token, like we did in subsection 6.2. There are multiple reasons that can explain why beam search does not improve the results. First off, it could simply be because our models could yield so good results, that it is too difficult for beam search to improve the results. Secondly, it could be because our model does not use an autoregressive decoder, meaning that each generated token is not conditioned on previously

	Training			Validation		
	English	**Finnish**	**Japanese**	**English**	**Finnish**	**Japanese**
`Macro`	.952	.958	.948	.755	.835	.820
`Micro`	.991	.993	.991	.960	.981	.972

Table 10: F1-score of the three developed sequence labellers with difference setting of the parameter "average" provided by the `f1_score` from the sklearn-library (scikit learn)

k	**English**	**Finnish**	**Japanese**
1	.960 / .755	.979 / .833	.972 / 820
5	.958 / .703	.977/.751	.970/.795
256	.957/.693	.976/.749	.968/.782

Table 11: Results of using Beam Search for various settings of the amount of "beams" to use (k). The reported numbers are F1-scores on the validation data with the average-parameter set as "micro"/"macro". Note that this is not division.

generated tokens.

6.6 Qualitative Investigation

Table 12 gives an overview of the question, ground truth answer to the question, and the predicted answer to the question. From the table we can see, that our model generally predicts an answer that does in fact include the ground truth answer and sometimes even predict the answer perfectly. All in all the predicted answer span does seem to make sense for the task.

7 Multilingual QA

Lastly, we will implement a multilingual question-answering system, based on multilingual text representations.

7.1 Multilingual Encoder

For the multilingual encoder we use the same pretrained version of XLM-RoBERTa (deepset) as we did in subsection 6.2. However, instead of finetuning the encoder on the three languages separately, we instead finetune the encoder on the three languages collectively, making just one multilingual encoder that only works for these three languages. The finetuning of the encoder is following the same setup as we did when we finetuned the three encoders in subsection 6.3.

7.2 Zero-Shot Cross-Lingual Evaluation

Table 13 illustrates the results of performing zero-shot cross-lingual evaluation. By comparing the results in Table 13 with the results in sections 4.5 we see, that the logistic regression actually becomes better, whereas the neural network becomes a bit worse. The small decrease in the performance of the neural network could be due to the randomness in the initialization of the network, whereas the increase of performance for the logistic regression could be either because of the increased data amount or because we now use RoBERTa instead of GPT2.

By comparing the results in Table 13 with the results in Table 10 we see, that the performance of the model does decrease a bit when we use the multilingual encoding. This can be because the model becomes unspecific and can now contain more noisy weights, or it might simply be because of the randomness in the initialization of the weights.

Lastly, by comparing the top and middle tables in Table 13 against the bottom table we see, that, in the cross-lingual cases where the classifiers do no perform well, the sequence labeller does actually perform well. And oppositely, in the cross-lingual cases where the sequence-labeller does not perform well, the binary classifiers perform their best, hinting towards the multilingual encoder performing equally well on all three languages.

8 Discussion

A mistake that we have realized in our approach across the entire project is the fact that we have not made use of a three-way split, containing a train set, validation set and a test set. This means that we risk introducing bias, when we pick the best model out of N epochs, and also when we adapt our training hyperparameters, to increase our validation accuracy. While this might not introduce as much bias as if, for example, we had made a grid search, we should still expect some bias in our validation results, and might therefore expect slightly worse results on unseen test data.

Question	Ground Truth Answer	Predicted Answer
What is a way to increase your wound healing speed?	cleaning and protection from reinjury or infection	cleaning and protection from reinjury or infection
Who founded the Burntisland Shipbuilding Company?	Brothers Amos and Wilfrid Ayre	Brothers Amos and Wilfrid Ayre
When did the case of R (Factortame Ltd) v Secretary of State for Transport take place?	December 1988	and Wales in December 1988
Minä vuonna Euroviisut järjestettiin ensimmäisen kerran? (In what year was the Eurovision Song Contest held for the first time?)	1956 (1956)	1956 (1956)
化学兵器禁止条約はどこで採択された？ (Where was the Chemical Weapons Convention adopted?)	日にパリ (paris by day)	日にパリ (paris by day)

Table 12: Question, ground truth answer to the question, and the predicted answer to the question given the question and the context. The sentences in parentheses are the English translation of the previous sentence, done by Google translate.

Logistic	English	Finnish	Japanese
English	.803	.715	.652
Finnish	.697	.804	.511
Japanese	.719	.580	.783
Neural	**English**	**Finnish**	**Japanese**
English	.798	.705	.675
Finnish	.647	.812	.528
Japanese	.725	.607	.778
RoBERTa	**English**	**Finnish**	**Japanese**
English	.946/.720	.965/.787	.974/.802
Finnish	.968/.765	.974/.823	.979/.820
Japanese	.945/.721	.963/.790	.960/.801

Table 13: Top/middle: Results of performing zero-shot cross-lingual evaluation for the binary classification for the logistic regression (top) and the neural network (middle). The reported values are binary accuracy. Bottom: Results of performing zero-shot cross-lingual evaluation for the sequence labeller. The reported values are F1-scores with the average-parameter set as "micro"/"macro". The first column indicates the language of the training data. The first row indicates the language of the evaluation data.

Secondly, when we finetuned the multilingual question answering encoder, we found ambiguity in whether or not we were allowed to finetune the encoder on the three languages or if we were supposed to keep the pretrained RoBERTa model as it came, such that it would be functioning on more than just our three languages. We ended up deciding to finetune it on our three language, as this would yield better results and the encoder still remaining multilingual. However, we were not completely sure whether this was the correct decision, as this was never mentioned in the task description, as well as the encoder not being very multilingual, as it potentially only works on three languages.

9 Conclusion

We have throughout this paper successfully implemented a multilingual question answering system. This includes multiple binary classifiers, as well as a sequence labeller. We have further performed an error analysis to see what sort of data makes the model incorrectly predict on our QA data. Finally, the models have been evaluated and compared on both English, Finnish, and Japanese QA data.

References

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. Tydi qa: A benchmark for information-seeking question answering in ty-

pologically diverse languages. *Transactions of the Association for Computational Linguistics*.

A. Et al. Conneau. 2020. Unsupervised cross-lingual representation learning at scale.

danielhers. Sequence labelling.

deepset. Multilingual xlm-roberta large for qa on various languages.

Finnish-NLP. Gpt-2 for finnish.

Benjamin Heinzerling and Michael Strube. 2018. BPEmb: Tokenization-free Pre-trained Subword Embeddings in 275 Languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Huggingface. Trainer.

nlp waseda. nlp-waseda/gpt2-small-japanese.

PyTorch. Bcewithlogitsloss.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

scikit learn. sklearn.metrics.f1_score.

A Adversarial answers

A.1 Easy context 1

On November 22, 1963, when he was hardly past his first thousand days in office, John Fitzgerald Kennedy was killed by an assassin's bullets as his motorcade wound through Dallas, Texas. Kennedy was the youngest man elected President; he was the youngest to die. Of Irish descent, he was born in Brookline, Massachusetts, on May 29, 1917. Graduating from Harvard in 1940, he entered the Navy. In 1943, when his PT boat was rammed and sunk by a Japanese destroyer, Kennedy, despite grave injuries, led the survivors through perilous waters to safety. Back from the war, he became a Democratic Congressman from the Boston area, advancing in 1953 to the Senate. He married Jacqueline Bouvier on September 12, 1953. In 1955, while recuperating from a back operation, he wrote Profiles in Courage, which won the Pulitzer Prize in history. In 1956 Kennedy almost gained the Democratic nomination for Vice President, and four years later was a first-ballot nominee for President. Millions watched his television debates with the Republican candidate, Richard M. Nixon. Winning by a narrow margin in the popular vote, Kennedy became the first Roman Catholic President. His Inaugural Address offered the memorable injunction: "Ask not what your country can do for you–ask what you can do for your country." As President, he set out to redeem his campaign pledge to get America moving again. His economic programs launched the country on its longest sustained expansion since World War II; before his death, he laid plans for a massive assault on persisting pockets of privation and poverty. Responding to ever more urgent demands, he took vigorous action in the cause of equal rights, calling for new civil rights legislation. His vision of America extended to the quality of the national culture and the central role of the arts in a vital society.

A.2 Easy context 2

Jupiter is primarily composed of hydrogen, but helium constitutes one-quarter of its mass and one-tenth of its volume. It probably has a rocky core of heavier elements, but, like the other giant planets in the Solar System, it lacks a well-defined solid surface. The ongoing contraction of Jupiter's interior generates more heat than it receives from the Sun. Because of its rapid rotation, the planet's shape is an oblate spheroid: it has a slight but noticeable bulge around the equator. The outer atmosphere is divided into a series of latitudinal bands, with turbulence and storms along their interacting boundaries. A prominent result of this is the Great Red Spot, a giant storm which has been observed since at least 1831. Jupiter is the fifth planet from the Sun and the largest in the Solar System. It is a gas giant with a mass more than two and a half times that of all the other planets in the Solar System combined, but slightly less than one-thousandth the mass of the Sun. Jupiter is the third brightest natural object in the Earth's night sky after the Moon and Venus, and it has been observed since prehistoric times. It was named after the Roman god Jupiter, the king of the gods. Jupiter is surrounded by a faint planetary ring system and a powerful magnetosphere. Jupiter's magnetic tail is nearly 800 million km (5.3 AU; 500 million mi) long, covering nearly the entire distance to Saturn's orbit. Jupiter has 80 known moons and possibly many more, including the four large moons discovered by Galileo Galilei in 1610: Io, Europa, Ganymede, and Callisto. Io

and Europa are about the size of Earth's Moon; Callisto is almost the size of the planet Mercury, and Ganymede is larger.

A.3 Easy context 3

An extremely long stretch limo, dubbed "The American Dream," has earned the coveted title of the world's longest car by Guinness World Records coming in at 100 feet, 1.5 inches long. The automobile has 26 wheels, a large waterbed, a swimming pool complete with a diving board, as well as a hot tub, a bathtub, a mini-golf course and a helipad, according to Guinness World Records. It was restored from the body of the previous record holder, taking multiple years of hard work and skill to complete. "When I [saw] the car, it was in very poor condition, on a scale of 1-10, it was minus 1," Mike Manning, director and president of Autoseum, the company responsible for the restoration, said in a statement. History of 'The American Dream,' world's longest car "The American Dream" was first recognized by Guinness World Records in 1986 after being built in Burbank, California, by famed car customizer Jay Ohrberg. It originally measured 60 feet, rolled on 26 wheels and had a pair of V8 engines at the front and rear. The unique automobile shot to fame and was often rented for cinematic appearances and featured in various movies, Guinness World Records said. But over time, obstacles arose such as where to park the long vehicle and dedication to its maintenance faded - leading to its demise. "Over time, it began to rust until parts of it were rendered unsalvageable," the company said. Manning, based in Nassau County, New York, saw "The American Dream" on eBay and made an offer, hoping it was his chance at owning the super stretched limo. "The corporation that had it listed did not want to sell it to me because they thought my offer was too low, so I made a deal to partner with them and bring it to New York," Manning said. But again, the iconic car faced hurdles amid funding and logistics challenges. Manning ended up listing it back on eBay - where it was spotted and purchased by Michael Dezer, owner of the Dezerland Park Car Museum and Tourist Attractions in Orlando, Florida, in 2019.

A.4 Hard context 1

The ICC Training and Education programme was launched in 2021 and is designed to provide educational resources and training opportunities through ICC-certified pathways to develop more coaches, umpires, scorers and pitch curators around the world. In partnership with Members, the ICC is committed to delivering a full range of courses to those interested in starting or furthering their cricket journey. Register and start learning Register and start learning CLICK HERE Current ICC courses ICC Foundation Certificate The first step for aspiring coaches is to undertake the ICC Coaching Foundation Certificate - a course which equips participants with the knowledge necessary to support qualified coaches deliver fun-first experiences for new and beginner participants. The course does not require any prior knowledge of the game, and introduces learners to the basic fundamentals of cricket, and what it takes to facilitate cricket sessions. The ICC Coaching Foundation Certificate is entirely self-paced, meaning it can be done as fast or as slow as the learner wants, and is delivered entirely online. Six modules make up the ICC Foundation Certificate: The Game, Safety & Inclusion, The Participants, The Coach, Effective Training Sessions Game Day. Assessments are available at the end of each module, and require the aspiring coach to hit the pass mark of 75% before progressing to the next module. Once all six modules and assessments have been completed, a certificate of completion will be issued Training & Education Training & Education.

A.5 Hard context 2

Society has conspired with Hollywood to put two seemingly-sexy psychology terms into our collective consciousness - psychopath and sociopath. Psychopath and sociopath are pop psychology terms for what psychiatry calls antisocial personality disorder. These two terms are not well-defined in the psychology research literature - hence the confusion about them. Nonetheless, there are some general similarities as well as differences between these two personality types. Both sociopaths and psychopaths have a pervasive pattern of disregard for the safety and rights of others. Deceit and manipulation are central features to both types of personality. Contrary to popular belief, a psychopath or sociopath is not necessarily violent. The common features of a psychopath and sociopath lie in their shared diagnosis: antisocial personality disorder. The DSM-5 defines antisocial personality as someone having three or more of the following traits. In both cases, some signs or symptoms are nearly always present before age 15. By the time

a person is an adult, they are well on their way to becoming a psychopath or sociopath.

A.6 Hard context 3

Denmark is famous for having the oldest flag still in use today. In 2019, the Danish flag turned 800 years old! The legend of the Dannebrog, the Danish nickname for their flag, is just as interesting. The story goes that in 1219, during a crusade in present-day Estonia, the Danish King Valdemar Sejr saw a red cloth with a white cross fall from heaven in front of him. It's rumored that the sight of this flag led Denmark to an unexpected victory. Well, King Valdemar would say that - he got the nickname Valdemar the Victorious after the battle. Naturally, there's no truth to this story, but I think you'll agree it's a good one. Danish parents still tell their children this legend - and Danes really do love their flag. Christmas trees, cakes, outside the house - you'll find Danish flags flying everywhere on your visit to Denmark. 2. Agricultural exports bacon pan Denmark is famous for exporting milk and pork to European countries. Both of these industries are a major source of income and combined, they make up a quarter of all of Denmark's agricultural exports. 90% of the pork from Denmark is exported, and the country has over 5,000 pig farms. There are actually more pigs in Denmark than people! The milk and dairy industry earns Denmark a massive 13 billion DKK ($2.1 billion) every year. The main dairy provider in Denmark is Arla, which is a Danish-Swedish cooperatively owned company.