BlueToad at SemEval-2025 Task 3: Using Question-Answering-Based Language Models to Extract Hallucinations from Machine-Generated Text

Michiel Pronk and Katja Kamyshanova and Thijmen Adam and Maxim van der Maesen de Sombreff
Faculty of Arts, University of Groningen, Groningen, NL
{m.t.pronk, e.a.kamyshanova, t.w.adam, m.a.x.van.der.maesen.de.sombreff}@student.rug.nl

Abstract

Hallucination in machine-generated text poses big risks in various domains, such as finance, medicine, and engineering. Task 3 of SemEval-2025, Mu-SHROOM, challenges participants to detect hallucinated spans in such text. Our approach uses pre-trained language models and fine-tuning strategies to enhance hallucination span detection, focusing on the English track. Firstly, we applied GPT-4o mini to generate synthetic data by labeling unlabeled data. Then, we employed encoder-only pre-trained language models with a question-answering architecture for hallucination span detection, ultimately choosing XLM-RoBERTa for fine-tuning on multilingual data. This model performed best, ranking 18th in IoU (0.469) and 22nd in Correlation (0.441) on the English track. It achieved promising results across multiple languages, surpassing baseline methods in 11 out of 13 languages, with Hindi having the highest scores of 0.645 intersection-over-union and 0.684 correlation coefficient. Our findings highlight the potential of a QA approach and using synthetic and multilingual data for hallucination span detection.

1 Introduction

Hallucinations can lead to dangerous and misleading information like mathematical inaccuracies in finance, programming errors in autonomous vehicles, and misunderstandings in medical diagnoses (Williamson and Prybutok, 2024). They pose a challenge in the development of AI models. In task 3 of SemEval-2025, called Mu-SHROOM, participants were challenged to create a model that can automatically extract hallucination spans in machine-generated text (Vázquez et al., 2025). This paper contains an overview of our approach for task 3 of SemEval-2025.

The organizers of this shared task define hallucinations as follows: *"Content that contains or describes facts that are not supported by the provided reference"*. (Vázquez et al., 2025)

In other words, hallucinations are cases where the machine-generated text is more specific than it should be or factually incorrect, given the information available in the provided context.

In the task from last year (Mickus et al., 2024), the seemingly best approach was to use pre-trained language models (PLMs) and fine-tuning. Most teams also used unlabeled training data, resulting in promising solutions. For this reason, we have incorporated these ideas. Firstly, we utilized the unlabeled data to fine-tune an open-source PLM to create a model that can effectively detect the spans of hallucinations in a text. We used GPT-4o mini (OpenAI, 2024) with prompt engineering to label the unannotated training data. Then, for our span detection system, we implemented a pre-trained question-answering (QA) architecture, in which we compared RoBERTa (Liu et al., 2019) and DeBERTa (He et al., 2021). In our final system, a multilingual version of RoBERTa, namely XLM-RoBERTa (Conneau et al., 2020), was fine-tuned on the training and validation data. This landed us the 18th and 22nd positions on the English track on the metrics of intersection-over-union and correlation, respectively.

The code is available on our GitHub[1] and the model on Huggingface[2].

2 Background

2.1 Task

This year's task was set up in a multilingual context. It contained fourteen languages: Arabic (Modern standard), Basque, Catalan, Chinese (Mandarin), Czech, English, Farsi, Finnish, French, German,

[1] https://github.com/MichielPronk/bluetoad-semeval-2025-Mu-SHROOM

[2] Tuned model on Huggingface: https://huggingface.co/MichielPronk/xlm-roberta-mushroom-qa

Proceedings of the The 19th International Workshop on Semantic Evaluation (SemEval-2025), pages 684–694

Hindi, Italian, Spanish, and Swedish. Participants in this task had to predict whether a character in a text generated by a large language model (LLM) is hallucinated. As our team is fluent in English, we mainly experimented with the provided English data in our approach. We also ran our model on the other languages, excluding Catalan, as this data was not properly formatted. The datasets provided by the organizers this year included a manually labeled validation set, an unlabeled training set, and an unlabeled test set, see Table A.1. For English, there were 809 training, 50 validation, and 154 test instances. The labeled data had two categories of labels: soft and hard labels. Both categories labeled text as hallucinations on the character level, with the difference between the soft and hard labels being that soft labels were the probabilities of a character being a hallucination as assigned by human annotators (an example of the data entry is provided in Appendix A.4).

2.2 Related work

This year's task is comparable to Task 6 of SemEval-2024, SHROOM, where participants were challenged to detect hallucinations on the document level instead of the character level (Mickus et al., 2024). An essential part of both of these tasks is handling the data made available by the shared task organizers. Last year, the data consisted of an unlabeled training set and a labeled validation set.

One of the challenges encountered last year was converting the unlabeled data into a useful dataset for experiments. Some of the entries from last year used LLMs to create training data (Das and Srihari, 2024; Bahad et al., 2024; Chen et al., 2024). Das and Srihari (2024) used the Claude 2.1 LLM. However, they found that this model would not always give reliable labels and added a confidence-based measure. Bahad et al. (2024) used Mixtral 8x7B to label the training data and obtained more consistent labels in comparison to Claude 2.1.

What is noticeable in the entries from last year is that some teams used the small validation set and the unlabeled data to create a larger labeled training set. Rösener et al. (2024) did not apply the unlabeled data in their research and focused on the provided labeled validation set. Instead, they used vectors as encoder input to solve the limited data, generating additional contextual information about the features, which helps the algorithm train on the little data more efficiently. In the results from last year (Mickus et al., 2024), it is visible that the teams that used the unlabeled training dataset obtained better results in comparison to the teams that did not, Chen et al. (2024) had the second best-performing model, indicating that good use of the provided unlabeled training data can be very effective towards increasing performance.

The use of ensemble models to classify documents was one of the most popular approaches last year (Das and Srihari, 2024; Chen et al., 2024; Rykov et al., 2024). This worked well in last year's task, but due to the differences in the current task and our time constraints, we have decided not to create an ensemble model. The teams with the highest scores from last year mostly used closed-source LLMs, which were often fine-tuned (Mickus et al., 2024). Other teams that also scored high used open-source LLMs and fine-tuning. Mickus et al. (2024) mentioned that a closed-source model like GPT-3.5 or GPT-4 is not requisite to build a good-working model, as is showcased in the paper by Chen et al. (2024), who used different open-source LLMs in combination with fine-tuning to obtain the second-best results.

Question-answering architecture allows a language model to return the start and end positions of an answer to a question in the given context. In an article by Sadat et al. (2023), QA is used to see whether an answer is grounded, and if it is not grounded, it is predicted to be hallucinated. For this, they use similarity-based testing because they want to detect whether the sentence contains a hallucination. The model obtained an F1-score of 71.1%.

3 System Overview

Our approach to the task consisted of annotating the supplied unlabeled training data and, in turn, using the data to fine-tune a pre-trained large language model for question answering. In both parts, we performed several experiments to find the optimal setup.

3.1 Synthetic Data Generation

A relatively small number of annotated hallucination entries in the English validation data may not provide the system with the necessary insight into the concept of hallucination. Therefore, we automatically converted a sample of provided unlabeled English data into a labeled one. We considered a decoder-only LLM to generate the data due to its strong in-context learning capabilities. This LLM

allows us to control the output by explaining the task to the system with a few-shot prompt. The decoder takes the prompt to generate synthetic data that can contribute to further system training.

3.2 Fine-tuning pre-trained language model

A difficult aspect of the task is to detect the characters that a hallucination consists of. We have boiled this down to extracting hallucination spans, as hallucinations almost always occur on a token level. This is an open exercise without predetermined answer options and number of hallucinations in each output, which poses a great challenge to the whole task.

Model architecture. We propose a pre-trained model with an extractive question answering architecture to tackle the task. This architecture allows a PLM to return an answer's start and end positions to a question in a given context. This approach is similar to the shared task, which aims to extract the hallucination spans from the model output text given an input question. However, a key difference is that in regular question-answering, the model tries to find the best answer to the posed question. In contrast, in our task, the model tries to find information in a context that cannot be inferred from the posed question. We used the `AutoModelForQuestionAnswering` from the transformers library by Huggingface.[3]

Span generation. One challenge encountered with this approach is that extractive question-answering systems are optimized for identifying the single most relevant answer, whereas the task at hand stressed the identification of all potential hallucination spans. The model predicts the probability of each token being the start of an answer span and the probability of it being the end. The start and end positions with the highest logit scores are combined with the answer span. To ensure our algorithm finds multiple spans, it takes the 20 best start and end positions and creates all possible span combinations. It then filters out combinations where the end occurs before the start and which exceed a set length of 30 characters. From the remaining combinations, the start and end logits are added, and this list is sorted descending on logit score. The highest is taken, and a threshold is set at 0.8 of the highest logit score. Every span that exceeds the threshold is seen as a possible hallucination. The maximum character length and logit threshold were determined through experimenting with different values on the validation set.

Fine-tuning. A traditional question-answering architecture takes a question and a context from which to extract the answer as input. In fine-tuning, the answer is given as a single span with the start and end positions corresponding to the context. Our model input is the prompt as question and model output as context. In the dataset, each instance is a model input, output, and a list of hallucination spans. This list of spans did not work well with the chosen architecture. Therefore, we preprocessed the data to create separate instances for each hallucination span, paired with the corresponding model input and output. These were then fed to the model during fine-tuning.

4 Experimental Setup

Synthetic data generation. We selected GPT-4o mini (OpenAI, 2024) as the automatic hallucination detector for its easy-to-access online interface for prompt experimentation, strong instruction-following capabilities, and API availability. Using the various combinations of the entries from the English validation set, we constructed a few-shot prompt for the system (see Appendix A.2). The included examples correspond with the case types the GPT model found the most challenging to predict, mainly involving annotation span nuances. The designed prompt generated a sample of 500 entries with automatically annotated hallucinated text segments (see an example output entry in Appendix A.3). To prevent the system from incorrectly counting the hallucination spans and hallucinating on the numerical probabilities, we instructed the system to provide hallucinations in textual form only, which were then converted to corresponding hard labels using a Python script.

Pre-trained language model experiments. We fine-tuned a RoBERTa (Liu et al., 2019) and DeBERTa (He et al., 2021) model using the same hyperparameters. They were fine-tuned on the synthetically generated data and evaluated on the validation data in English. Our hyperparameter tuning focused on the learning rate, batch size, weight decay, number of epochs, and the logit threshold when selecting hallucination spans. The model was then fine-tuned on the synthetic training and vali-

[3] https://huggingface.co/transformers/v3.0.2/model_doc/auto.html#transformers.AutoModelForQuestionAnswering

dation English data of 550 instances in total. Furthermore, a multilingual model, XLM-RoBERTa (Conneau et al., 2020), was fine-tuned on the synthetic training English data and all validation data (see Table 4), which combined to 1000 instances. This model was also tuned, focusing on the number of epochs and learning rate and using only the training and English validation data. The specific hyperparameters of the final model can be found in Appendix A.5.

Packages. The models were loaded with `huggingface` (Wolf et al., 2020) and fine-tuned using the `transformers` and `datasets` Python libraries.

Metrics. We evaluated the models using the metrics set by the organizers, which are the intersection-over-union (IoU) of the characters marked as hallucinations in the predicted hallucinations and the true hallucinations and correlation (Cor) between the predicted probability of a character being hallucinations and the probability assigned by human annotators. As it seemed more relevant and less ambiguous to find the hallucinations than assigning probabilities to them, we focused on achieving the highest intersection-over-union.

There were three baselines introduced by the organizers of the task: the *mark all*, which predicted every character as hallucination, the *mark none*, which predicted no characters as hallucination, and the *neutral*, which estimates hallucinations based on probability distributions. The *mark all* baseline was in every language the highest scoring baseline.

Model	IoU	Cor
RoBERTa	0.368	0.355
DeBERTa	0.369	0.351

Table 1: Comparison of the RoBERTa and DeBERTa model on the validation set

5 Results and Analysis

5.1 Preliminary Results

We fine-tuned and compared a DeBERTa and RoBERTa model to see their performance using the same hyperparameters on the validation set. During experimentation, we ran each model once. The results can be found in table 1. Since we found that the results were quite close together, we decided to focus on one model only, namely RoBERTa. We experimented with the hyperparameters, including the learning rate, batch size, and weight decay, but found no improvement. We then opted to add the validation data to the training data when fine-tuning. This gave us better scores on the validation but not the test set. The better performance on the validation set could be attributed to overfitting. The score decline on the test set could be due to training and validation data sharing the same inputs, which could also have led to overfitting because more of the same data was added.

Finally, we combined the training and validation data for all languages and fine-tuned an XLM-RoBERTa model. This gave us better scores on the English validation data and higher scores on the English test set. A possible reason is that the multilingual data is more varied and of higher quality, resulting in a better overall performance, topping the performance of the English-specific model. Table 2 shows the results for our iterations of the RoBERTa models on the test set.

Model	Data	IoU	Cor
RoBERTa	Train	0.348	0.334
RoBERTa tuned	Train	0.38	0.347
RoBERTa	Train + EN Val	0.371	0.353
XLM-RoBERTa	-	0.125	0.04
XLM-RoBERTa	Train + all Val	**0.469**	**0.441**

Table 2: Model performance on English test data and what data we used. Also including non fine-tuned XLM-RoBERTa scores. Best results in **bold**.

5.2 Final results

We let the model predict for all the other languages as it is pre-trained on the multilingual data. The results and positions in the competition can be found in table 3. Here, the ranking is based on the IoU scores. We were quite surprised by the performance in the other languages. Our model competed in 13 languages and outperformed the baseline IoU scores in 11 languages, only falling behind in French and Chinese. On Cor we beat every baseline.

In the English task, our model ranked 18th in IoU and 22nd in Cor out of 41 teams and three baselines. Surprisingly, it performed best in Hindi, possibly due to dataset-specific hallucination traits. However, an examination of language represen-

tation within the XLM-RoBERTa model reveals no clear correlation. All languages from the task are in the training corpus of the model, but the languages there seem to have no connection to the performance or the number of tokens in the XLM-RoBERTa training data. Although Swedish and Chinese have lower representation in the data, the model performed well in Swedish and poorly in Chinese. Despite English having the highest data representation, it was not the best-performing language. The varying performance could be attributed to the characteristics of the language, the difference in hallucinations and/or annotations between languages, or the model that produced each output containing hallucinations.

Language	IoU position	Cor position	IoU	Cor
AR	8/29	14/29	0.547	0.506
CS	14/23	14/23	0.351	0.3628
DE	12/28	11/28	0.544	0.5243
EN	*18/41*	*22/41*	*0.469*	*0.441*
ES	19/32	15/32	0.279	0.4267
EU	12/23	13/23	0.506	0.457
FA	10/23	8/23	0.571	0.579
FI	11/27	16/27	0.569	0.491
FR	19/30	20/30	0.439	0.38
HI	8/24	8/24	**0.645**	**0.684**
IT	13/28	10/28	0.639	0.668
SV	7/27	11/27	0.585	0.427
ZH	20/26	20/26	0.278	0.226

Table 3: The final scores and positions for the IoU and Cor out of the total participants plus three baselines per language. English scores marked in *italics* and the highest IoU and Cor scores in **bold**

5.3 Error Analysis

We conducted a qualitative analysis of the system's best prediction attempt for the English test set, comparing them with the provided gold standard. One key observation was the variation in spans selected by the system. Since the original QA architecture is designed to identify a single answer, it uses logits to determine the most probable start and end positions. While this approach works for a single span, logits are not intended to link multiple spans or indicate precise hallucination boundaries.

The system provides multiple start and end positions, so we manually combined them into different span variations and assessed their probability of being hallucinations based on their logits. This solution ensured that the picked spans were considered as part of hallucination by the system. Still, it resulted in a lot of noise in the output, consisting of overlapping spans, which negatively reflected on the evaluation scores. Examples of such QA output can be found in Appendix A.6.

These examples also illustrate the system's tendency to pick positions based on the syntactic dependencies within the sentence. This behavior can be linked to the QA system's pre-training to process text at the token level, which ensures accurate span selection. While the current task can benefit from such an approach for a similar reason, the token-based span selection limits the system's ability to detect character-level hallucinations. Since the gold standard hallucination spans were annotated by humans at the character level, some spans appear abrupt and do not always include complete words. As the examples in Appendix A.7 illustrates, our system could not identify these hallucinations with the current fine-tuning and pre-training.

6 Conclusion

In this paper, we presented several contributions to the task of hallucination detection in machine-generated output. We used GPT-4o mini to generate synthetic hallucination span annotations, adapted QA architecture for hallucination span extraction and finetuned an XLM-RoBERTa model to generalize across 13 languages, outperforming the baseline in 11 languages. This approach resulted in the 18th and 22nd position in the English subtask with an intersection-over-union of 0.469 and a correlation of 0.441 respectively.

Further research could focus on trying different large language models. Also, using human-annotated data seemed to give big improvements, so trying to add more quality data could yield better results. Furthermore, we found that our method to synthesize hallucination spans from model predictions could be improved as it can detect the correct spans but assigns a low probability to them, resulting in too many characters being marked as hallucinations. Finally, we used GPT-4o mini, which has the drawbacks of being closed-source and paid. Attempting the automatic annotation process with an open-source model would be preferable.

Acknowledgments

We would like to thank Professor Malvina Nissim and Dr. Huiyuan Lai from the University of Groningen for supervising and assisting us during this project.

References

Sankalp Bahad, Yash Bhaskar, and Parameswari Krishnamurthy. 2024. NootNoot at SemEval-2024 task 6: Hallucinations and related observable overgeneration mistakes detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 964–968, Mexico City, Mexico. Association for Computational Linguistics.

Ze Chen, Chengcheng Wei, Songtan Fang, Jiarong He, and Max Gao. 2024. OPDAI at SemEval-2024 task 6: Small LLMs can accelerate hallucination detection with weakly supervised data. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 721–729, Mexico City, Mexico. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. *arXiv:1911.02116v2*.

Souvik Das and Rohini Srihari. 2024. Compos mentis at SemEval2024 task6: A multi-faceted role-based large language model ensemble to detect hallucination. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1449–1454, Mexico City, Mexico. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DEBERTA: Decoding-enhanced BERT with disentangled attention. In *International Conference on Learning Representations*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Timothee Mickus, Elaine Zosa, Raul Vazquez, Teemu Vahtola, Jörg Tiedemann, Vincent Segonne, Alessandro Raganato, and Marianna Apidianaki. 2024. SemEval-2024 task 6: SHROOM, a shared-task on hallucinations and related observable overgeneration mistakes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1979–1993, Mexico City, Mexico. Association for Computational Linguistics.

OpenAI. 2024. Gpt-4o mini. Large language model, July 18 version.

Béla Rösener, Hong-bo Wei, and Ilinca Vandici. 2024. Team bolaca at SemEval-2024 task 6: Sentence-transformers are all you need. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1663–1666, Mexico City, Mexico. Association for Computational Linguistics.

Elisei Rykov, Yana Shishkina, Ksenia Petrushina, Ksenia Titova, Sergey Petrakov, and Alexander Panchenko. 2024. SmurfCat at SemEval-2024 task 6: Leveraging synthetic data for hallucination detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 869–880, Mexico City, Mexico. Association for Computational Linguistics.

Mobashir Sadat, Zhengyu Zhou, Lukas Lange, Jun Araki, Arsalan Gundroo, Bingqing Wang, Rakesh R Menon, Md Rizwan Parvez, and Zhe Feng. 2023. Delucionqa: Detecting hallucinations in domain-specific question answering. *arXiv preprint arXiv:2312.05200*.

Raúl Vázquez, Timothee Mickus, Elaine Zosa, Teemu Vahtola, Jörg Tiedemann, Aman Sinha, Vincent Segonne, Fernando Sánchez-Vega, Alessandro Raganato, Jussi Karlgren, Shaoxiong Ji, Liane Guillou, Joseph Attieh, and Marianna Apidianaki. 2025. SemEval-2025 Task 3: Mu-SHROOM, the multilingual shared-task on hallucinations and related observable overgeneration mistakes.

Steven M Williamson and Victor Prybutok. 2024. The era of artificial intelligence deception: Unraveling the complexities of false realities and emerging threats of misinformation. *Information*, 15(6):299.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics.

A Appendix

A.1 Data distribution

	N entries (total)	N entries (EN)	Available for Languages
Train	3351	809	EN, FR, SP, ZH
Validation	500	50	AR, DE, EN, ES, FI, FR, HI, IT, SV, ZH
Test	1902	154	AR, CA, CS, DE, EN, ES, EU, FA, FI, FR, HI, IT, SV, ZH

Table 4: Statistics of the provided unlabeled train, validation, and test sets, showing the number of English entries and data availability across different languages.

A.2 Prompt

You are a model that detects hallucinations in a decoder-generated text. We define a hallucination as "content that contains or describes facts that are not supported by the provided reference". In other words: hallucinations are cases where the answer text is more specific than it should be or factually incorrect, given the information available in the provided context.

You are given a source text represented as a question and an answer to that question. Detect whether the answer contains hallucinations and provide the spans in the answer text that are the source of hallucination. Work per sentence. In each sentence, firstly, detect the word phrases that represent hallucinations. Within each phrase, search for specific words that do not align with the context by introducing a hallucination. Include only these words in the final answer. Below are three examples of correct hallucination detection:

SOURCE TEXT: "What is the population of the Spanish region of Galicia?"
ANSWER: TEXT: "As of 2021, the estimated population in the region is around 1.5 million people."
HALLUCINATION SPANS: "2021", "1.5 million"

SOURCE TEXT: "Do all arthropods have antennae?"
ANSWER TEXT: "Yes, all arachnids have antennas. However, not all of them are visible to the naked eye."
HALLUCINATION SPANS: "Yes", "arachnids", "visible", "naked eye"

SOURCE TEXT: Which country is the World Chess Federation based in?
ANSWER TEXT: The World Chess Federation, also known as FIDE (Fédération Internationale des Échecs), is not based in any one specific country. It is an international organization with its headquarters currently located in Minsk, Belarus. However, it maintains offices in several countries and holds various events around the world.
HALLUCINATION SPANS: 'not based in any one specific country', 'Minsk, Belarus', 'maintains offices', 'several countries'

Follow the examples and provide the hallucination spans for the following text pair:
SOURCE TEXT: <input>
ANSWER TEXT: <input>

A.3 Synthetic Training Data Entry

{"model_input": "Is the Arts and Humanities Citation Index still maintained?", "model_output_text": "As of 2021, the A&HCI is no longer maintained by the U.S. government and is now maintained privately by JSTOR.", "hard_labels": [[6, 10], [53, 68], [104, 109]]}

A.4 Validation and Training Data

Validation data example:


```
{"id":"val-en-1","lang":"EN","model_input":"What did Petra van Staveren
win a gold medal for?","model_output_text":"Petra van Stoveren
won a silver medal in the 2008 Summer Olympics in Beijing,
China.","model_id":"tiiuaefalcon-7b-instruct","soft_labels":
[{"start":10,"prob":0.2,"end":12},{"start":12,"prob":0.3,"end":13}
,{"start":13,"prob":0.2,"end":18},{"start":25,"prob":0.9,"end":31}
,{"start":31,"prob":0.1,"end":37},{"start":45,"prob":1.0,"end":49}
,{"start":49,"prob":0.3,"end":65},{"start":65,"prob":0.2,"end":69}
,{"start":69,"prob":0.9,"end":83}],"hard_labels":[[25,31],[45,49],[69,83]],
"model_output_logits":[-5.5669536591,-11.90533638,-13.0743436813,-9.9514026642,
-8.8359375,-5.2216725349,-8.8481779099,-9.2853775024,-7.6449022293,-8.7612609863,-
9.1256427765,-5.7042989731,-5.7393956184,-8.409078598,-10.6083183289,-11.707988739,
-5.3747014999,-6.5602250099,-5.1362328529,-5.7765812874,-8.4669551849,-8.3430461884,
-8.7018699646],"model_output_tokens":["Pet","ra","Ġvan","Ġsto","ve","ren","Ġwon",
"Ġa","Ġsilver","Ġmedal","Ġin","Ġthe","Ġ","200","8","Ġsummer",
"Ġolympics","Ġin","Ġbeijing",",","Ġchina",".","<|endoftext|>"]}
```


Training data example:


```
{"lang":"EN","model_id":"togethercomputerPythia-Chat-Base-7B","model_input":"Do
all arthropods have antennae?","model_output_text":" Yes, all insects and
arachnids (including spiders, scorpions, and ticks) have at least one antenna.
","model_output_logits":[-2.57427001,5.1865358353,5.4173498154,2.3298389912,
6.6413722038,3.5858919621,10.0333786011,4.0347309113,2.5324678421,0.595354557,
2.0120403767,1.8963916302,6.8019108772,6.2508711815,4.6549735069,2.6137938499,
-3.9706561565,7.5047488213,4.2792539597,3.0500841141,-0.4772415757,3.1624026299,
13.7479448318,-1.7824890614,3.9432423115,4.9513559341,4.3375749588,14.8357839584],
"model_output_tokens":["Ġyes",",","Ġall","Ġinsects","Ġand","Ġar",
"ach","n","ids","Ġ(","including","Ġspiders",",","Ġsc","orp","ions",",",
"Ġand","Ġticks",")","Ġhave","Ġat","Ġleast","Ġone","Ġantenna",
".","Ċa"]}
```


A.5 XLM-RoBERTa Settings

The hyperparameters used for the best performing XLM-RoBERTa model:

```
max_length=384
stride=128
n_best=20
max_answer_length=30
learning_rate=2e-05
batch_size=16
epochs=4
weight_decay=0.1
```

Here `max_length` corresponds to the maximum input length in the tokenizer. `stride` is the number of overlapping tokens between two successive chunks. `n_best` is the top K of how many start and end span positions to return from the predictions. `max_answer_length` is the maximum span length allowed when creating possible hallucination spans.

A.6 Examples of the Noisy Output by the QA System

The examples below were taken from the English test set. The ROBERTA spans are generated by the best XLM-RoBERTa QA system, while the TRUE spans are the gold standard provided by the organizers. The examples illustrate the large amount of noise the QA system indicates as hallucinations.

Example 1

SOURCE TEXT: "Did Health ever compose videogame soundtracks?"
ANSWER TEXT: " Yes, in 2015, Health composed the soundtrack for "Inside" by Tame Impala."
ROBERTA SPANS: ['2015, Health composed the soundtrack for "Inside', '2015,', '2015, Health composed the soundtrack for "Inside" by Tame Impala', 'Yes, in 2015, Health composed the soundtrack for "Inside', 'Yes, in 2015,', 'Yes, in 2015, Health composed the soundtrack for "Inside" by Tame Impala', 'Yes', 'Inside', 'Inside" by Tame Impala', 'Tame Impala']
TRUE SPANS: ['2015', 'Inside" by Tame Impala']
MATCHES: ['Inside" by Tame Impala']
MISSED SPANS: ['2015']
NOISE: ['2015, Health composed the soundtrack for "Inside', 'Inside', 'Yes, in 2015,', 'Yes', '2015,', 'Tame Impala', '2015, Health composed the soundtrack for "Inside" by Tame Impala', 'Yes, in 2015, Health composed the soundtrack for "Inside', 'Yes, in 2015, Health composed the soundtrack for "Inside" by Tame Impala']

Example 2

SOURCE TEXT: "Who developed International Klein Blue?"
ANSWER TEXT: " The dye was developed by a chemist named Ewald Klein in the 1930s."
ROBERTA SPANS: ['1930s', 'Ewald Klein', 'Ewald Klein in the 1930s', 'chemist named Ewald Klein', 'chemist named Ewald Klein in the 1930s']
TRUE SPANS: ['chemist', 'Ewald', '1930s']
MATCHES: ['1930s']
MISSED SPANS: ['chemist', 'Ewald']
NOISE: ['Ewald Klein', 'chemist named Ewald Klein', 'chemist named Ewald Klein in the 1930s', 'Ewald Klein in the 1930s']

Example 3

SOURCE TEXT: "Is there any year when the BAFTA Award for Best Actor in a Supporting Role hasn't been awarded?"
ANSWER TEXT: " No, there is not."
ROBERTA SPANS: ['No, there is not', 'No', 'there is not']
TRUE SPANS: ['No, there is not']
MATCHES: ['No, there is not']
MISSED SPANS: []
NOISE: ['No', 'there is not']

A.7 Examples of the Challenging Output

The examples below were taken from the English test set. The ROBERTA spans were generated by the best XLM-RoBERTa QA system, while the TRUE spans are the gold standard provided by the organizers. The examples illustrate the challenging cases of gold standard spans the QA system cannot comprehend.

Example 1

```
SOURCE TEXT: "How many martellos are there on Formentera?"
ANSWER TEXT: " There are a total of 4 martello towers on the island of Forments."
ROBERTA SPANS: ['4 martello towers on the island of Forments', '4', 'Forments']
TRUE SPANS: ['s']
MATCHES: []
MISSED SPANS: ['s']
NOISE: ['Forments', '4 martello towers on the island of Forments', '4']
```

Example 2

```
SOURCE TEXT: "is there an airfield in Beauvechain?"
ANSWER TEXT: " no, there is no airport in beauvechains"
ROBERTA SPANS: ['no, there is no airport in beauvechains', 'no, there is no airport',
'no']
TRUE SPANS: ['no', 'e is no airport']
MATCHES: ['no']
MISSED SPANS: ['e is no airport']
NOISE: ['no, there is no airport', 'no, there is no airport in beauvechains']
```