TU Wien at SemEval-2024 Task 6: Unifying Model-Agnostic and Model-Aware Techniques for Hallucination Detection

Varvara Arzt, Mohammad Mahdi Azarbeik, Ilya Lasy, Tilman Kerl, and Gábor Recski

Faculty of Informatics, TU Wien
{varvara.arzt, mohammad.azarbeik, ilya.lasy, tilman.kerl, gabor.recski}@tuwien.ac.at

Abstract

This paper discusses challenges in Natural Language Generation (NLG), specifically addressing neural networks producing output that is fluent but incorrect, leading to "hallucinations". The SHROOM shared task involves Large Language Models in various tasks, and our methodology employs both model-agnostic and model-aware approaches for hallucination detection. The limited availability of labeled training data is addressed through automatic label generation strategies. Model-agnostic methods include word alignment and fine-tuning a BERT-based pretrained model, while model-aware methods leverage separate classifiers trained on LLMs' internal data (layer activations and attention values). Ensemble methods combine outputs through various techniques such as regression metamodels, voting, and probability fusion. Our best-performing systems achieved an accuracy of 80.6% on the model-aware track and 81.7% on the model-agnostic track, ranking 3rd and 8th among all systems, respectively.[1]

1 Introduction

In Natural Language Generation (NLG), the trade-off of prioritising fluency over accuracy results in neural networks generating "hallucinations" – outputs fluent but factually inaccurate. The automatic identification of such errors represents a substantial challenge (Huang et al., 2023; Ji et al., 2022). The SHROOM shared task on hallucination detection (Mickus et al., 2024) highlights concerns about the practical utility of fluently generated yet inconsistent outputs. In the SHROOM shared task, Large Language Models' (LLMs) outputs for definition modeling (DM), machine translation (MT), and paraphrase generation (PG) tasks are presented with input source text and corresponding 'gold' reference text. Notably, for PG, the input source text serves as the reference 'gold' text. While the training dataset lacks labels, an issue which is addressed in Section 3, the validation dataset includes binary labels of Hallucination or Not Hallucination and hallucination probability of 0 to 1, corresponding to Hallucination and Not Hallucination, respectively, for the LLM's output. These assessments, based on five annotators' evaluations, rely on determining if the model's output is supported by the 'gold' reference, from either the 'gold' target text, source text, or both, depending on the task (DM, MT, or PG).

The SHROOM dataset is categorised into two tracks: model-agnostic and model-aware. The model-aware track, in contrast to the model-agnostic, includes the specific LLM responsible for the provided output. This paper introduces methods tailored to both tracks and since the model-agnostic techniques can be applied to both, evaluations for these methods are conducted on both test datasets to provide a comprehensive analysis. The model-agnostic methods entail employing word alignment to establish semantic similarity between the model output and the 'gold' reference as well as fine-tuning a BERT-based model for hallucination detection. On the other hand, model-aware approaches delve into the analysis of hidden states and attention flow within the model architecture. Ultimately, a diverse set of ensemble techniques, comprising logistic regression with binary labels, linear regression with raw probabilities, voting, and probability fusion, are introduced to amalgamate the proposed methods.

2 Related Work

Since the tendency of LLMs to produce incorrect output poses a serious challenge in their application, the task of hallucination detection has recently attracted a variety of research work. Model-agnostic approaches include the training of dedicated machine learn-

[1] Our code is available at `https://github.com/kleines-gespenst/shroom-hackathon`

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1183–1196
June 20-21, 2024 ©2024 Association for Computational Linguistics

ing models, such as a token-level classifier for detecting hallucination in machine translation (Zhou et al., 2021) or the BERT-based Vectara hallucination_evaluation_model[2], the latter which we also use in our model-agnostic experiments (see Section 3.2). Recent datasets for training and evaluating such models include task-agnostic corpora such as (Li et al., 2023) and HaDeS (Liu et al., 2022) as well as datasets focusing on specific generation tasks such as text summarisation (Laban et al., 2022), fact verification (Thorne et al., 2018), question answering (Pang et al., 2022; Longpre et al., 2021), or paraphrase generation (Zhang et al., 2019; Shen et al., 2022).

Another set of approaches involves comparing a model's output to some reference using any of a variety of unsupervised similarity metrics, including standard ngram-based measures such as BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) but also distributional similarity metrics such as BERTScore (Zhang et al., 2020), BARTScore (Yuan et al., 2021), or DiscoScore (Zhao et al., 2023). A large-scale analysis of the performance of these metrics on hallucination detection have been performed in the recent TRUE survey (Honovich et al., 2022). Further model-agnostic approaches to hallucination detection include comparing multiple LLM responses to a single query (Manakul et al., 2023), prompting LLMs to evaluate the likelihood of their own output being correct (Kadavath et al., 2022) as well as the use of external knowledge bases to assess the faithfulness of model outputs (Thorne et al., 2018; Guo et al., 2022; Peng et al., 2023)

Model-aware methods for hallucination detection include classifying an LLM's hidden layer activations to determine whether the question is answerable (Slobodkin et al., 2023) or whether its output is true (Azaria and Mitchell, 2023). The latter approach is the basis of the SAPLMA system, which we have also used in our experiments for the model-aware track of the shared task (see Section 3.3).

3 Methodology

Within the scope of hallucination detection, we employed both model-agnostic and model-aware methods. Our model-agnostic approaches encompassed rule-based techniques, featuring the application of string metrics and word embeddings, alongside the fine-tuning of pretrained language models. Model-aware methods leveraged the internal data of LLMs.

3.1 Automatic Data Annotation

Beyond the primary task of hallucination detection, a significant challenge arose due to the limited availability of labeled data, with only the SHROOM validation set being provided with labels. Faced with the absence of labeled training data, we explored diverse strategies for automatic label generation. These approaches encompassed zero-shot prompting with GPT-3.5 Turbo and the utilisation of BERTScore (Zhang et al., 2020) as a quantifiable metric, capable of serving as a probability indicator for hallucination within the generated content. BERTScore, employed in our approach for automatic data labeling, entails the aggregation of pairwise cosine similarity scores between the BERT contextual embeddings of tokens in candidate and reference sentences.

3.2 Model-agnostic Methods

Word alignment for semantic similarity Based on the understanding that hallucination is defined as model output that is semantically inconsistent with the reference output, we reduce the task of hallucination detection to that of measuring semantic similarity between pairs of sentences. The probability that a hypothesis sentence generated by a model contains hallucination should then be inversely proportional to its semantic similarity to the reference. In an effort to provide measures of semantic similarity that are more explainable than modern distributional metrics such as BERTScore (see Section 3.1) we developed a set of simple methods based on word alignment and word similarity. Given a word similarity metric, we simply align each word of one sentence to the most similar word in the other and define sentence similarity as the average of these similarities. Formally, for any word similarity metric S_w that maps any pair of words w_1, w_2 to the $[0, 1]$ range we define the similarity of sentences s_1 and s_2 as

$$S = \sum_{u \in s_1} \max_{v \in s_2} \frac{S_w(u, v)}{|s_1|} \tag{1}$$

We defined several word similarity metrics and for each determined a custom threshold t for which a hypothesis sentence s_{hyp} is considered a hallucination w.r.t. the reference s_{ref} if and only if

[2] https://huggingface.co/vectara/hallucination_evaluation_model

$1 - S(s_{hyp}, s_{ref}) \geq t$ (values of t were determined empirically on the validation dataset of the model-agnostic track). For each similarity type we experimented with alternative methods for calculating overall similarity from word similarities, including the harmonic mean and the minimum of the best similarities for each word, but the plain average (Eq. 1) yielded the best performance. We also ran all experiments with stopword removal using the `nltk` library (Bird et al., 2009) but found it to cause a slight decrease in performance. In our submissions, we included outputs based on two word similarity metrics. The `levenshtein` system that uses Levenshtein distance of word pairs as the word similarity measure for Equation 1, a string similarity metric defined as the number of character-level edit operations required to transform one word into another (Levenshtein, 1966). The similarity metric S_L was defined as $1/(1 + L)$ to obtain a value between 0 and 1 that is inversely proportional to the distance measure L. The `paragram` system combines the word alignment method with distributional similarity, here word similarity is defined as the cosine similarity of two words in the static English word embedding `paragram_300_SL999` (Wieting et al., 2015), which has been fine-tuned on the task of measuring word similarity on the SimLex-999 dataset (Hill et al., 2014).

Finetuning a BERT-based Hallucination Detection Model Another approach for the model-agnostic track encompassed finetuning a pretrained hallucination detection model based on BERT (Devlin et al., 2019). An open-source model developed by Vectara was chosen for that purpose as it achieved high accuracy on a range of hallucination detection benchmarks including e.g. accuracy of 76% on the `SummaC` dataset (Laban et al., 2022). Built upon the deberta-v3-base (He et al., 2021), Vectara undergoes initial training on Natural Language Inference (NLI) data, followed by subsequent fine-tuning on summarisation datasets. The model is trained utilising a cross-encoder architecture.[3] Given the scarcity of labeled data of high quality necessary for the initial finetuning of a language model, a departure from the approach taken by Zhou et al. (2021) was considered. In their work, they utilised XLM-R (Conneau et al., 2020) for hallucination detection within the scope of a machine translation task, and RoBERTa (Zhuang et al., 2021) for hallucination detection within the scope of summarisation task. However, due to the insufficient availability of labeled data, this method was not deemed applicable in the current study.

3.3 Model-aware Methods

Hidden States Model-aware techniques are based on analysing internal data of LLM during inference. One of the possible approaches is the analysis of the outputs of the hidden layers of the transformer. Using vector values of hidden layers for hallucination detection was proposed in a method called Statement Accuracy Prediction, based on Language Model Activations (SAPLMA) (Azaria and Mitchell, 2023). SAPLMA is a probing technique that utilises a feedforward neural network trained on activation values of the hidden layers of LLM.

Attention Flow We follow the attention-based token-level importance metric proposed by DeRose et al. (2020) for sequence classification and adopt it to sequence-to-sequence. We extract and analyse the attention weights from the model predictions and trace how the model shifts its focus from the output back to the input. This is done by summing and averaging the weights in the decoder, and then mapping these influences back through cross-attention layers to the encoder. Thereby, highlighting which parts in the input text are influential for the output. As an addition, we apply exponential decay to the influence scores to account for a diminishing impact of distant tokens across layers.

Consequently, we derive an influence matrix that quantifies the influence scores for each layer and token. Under the assumption that there exists a meaningful correlation between the input and its corresponding output, and thereby, also in cases where the output is characterised by hallucinations, these identified features can be leveraged to build a classifier.

4 Experiments

4.1 Automatic Data Annotation

As described in Section 3.1, two approaches were utilised for automatic annotation of the SHROOM training set, specifically zero-shot prompting with GPT-3.5[4] and the BERTScore metrics. To generate

[3] Notably, in contrast to the SHROOM dataset, Vectara produces a probability scale from 0 to 1, where 0 represents hallucination and 1 denotes factual consistency. Implementing a threshold of 0.5 enables predictions to assess the alignment of a document with its source.

[4] gpt-3.5-turbo-1106

probabilities with GPT-3.5 in a zero-shot manner, OpenAI API was used. The prompt was crafted specifically to incorporate multiple dataset samples in one pass in order to speed up the labeling process. Specifically, there were 32 samples in the prompt. Although the latest models support larger input context length, our experiments showed that passing more samples per request results in inconsistent and poor-quality labels. The input for the model was structured as pairs, consisting of a context ('tgt' for MT and DM, and 'src' for PG) and a sentence ('hyp'). Prompt engineering was inspired by the SHROOM baseline kit combined with instructions to return structured output in a JSON format. The explicit prompt passed as an instruction to GPT-3.5 is referenced in Appendix A. The total cost associated with utilizing the GPT-3.5 API amounted to ∼3$. Performance evaluation of GPT-3.5 on the validation dataset, comprising both its model-agnostic and model-aware parts, yielded metrics of **0.68** and **0.49** for accuracy and Spearman's correlation coefficient, respectively.

Simultaneously, BERTScore calculations were conducted on identical pairs of text instances used for probability generation with GPT-3.5. The candidate sentence ('hyp') and reference sentence ('tgt' for MT and DM, and 'src' for PG) served as inputs for BERTScore computation. The resulting BERTScore values, ranging between 0 and 1, were utilised in their raw form, wherein outputs of a specific LLM featuring BERTScores closer to 1 indicate a higher probability of being non-hallucinations. We utilised BERTScore version 0.3.13 with the RoBERTa Large model (Zhuang et al., 2021). Performance evaluation of BERTScore values on the validation dataset, comprising both its model-agnostic and model-aware parts, yielded metrics of **0.67** and **0.41** for accuracy and Spearman's correlation coefficient, respectively. For these calculations, the transformation of BERTScore values into labels utilized a threshold of 0.5.

4.2 Model-agnostic Methods

Word alignment The submissions `levenshtein` and `paragram` are based on the word alignment method described in Section 3.2. Threshold values for binary classification were determined empirically using the validation set of the model-agnostic dataset and set to 0.35 for `levenshtein` and 0.3 for `paragram`. Identical parameters were used to generate the submitted outputs for both model-agnostic and model-aware tracks of the shared task.

Finetuning a BERT-based Hallucination Detection Model This section presents the finetuning of Vectara pretrained hallucination detection cross-encoder model. We conducted finetuning on 4 different data combinations, including the validation set exclusively (`vectara-val`), the training set with probabilities generated by either GPT-3.5 (`vectara-gpt`) or BERTScore (`vectara-bertscore`), and a combination of both the validation set and the training set with GPT-3.5 probabilities (`vectara-gpt-val`). Given that Vectara predicts hallucination probability independently of a specific LLM, we concatenated both model-aware and model-agnostic datasets for additional finetuning to address limitations arising from their relatively small sizes. In addition to the mentioned data combinations, we performed separate finetuning on subsets of the validation set (`vectara-val-subset`), representing model-aware and model-agnostic validation sets. These finetuned models' predictions were later used in ensemble methods (Section 4.4). Despite acknowledging potential bias associated with finetuning using automatically generated probabilities, this approach was pursued due to the necessity of labeled data. To mitigate bias from GPT-3.5 and BERTScore probabilities, approaches (`vectara-val`, `vectara-gpt-val`, `vectara-val-subset`) utilised the validation set for finetuning. The finetuning process involved creating input instances for each training pair, comprising the LLM's generated text ('hyp'), the 'gold' reference text, and the probability of hallucination obtained from annotators ('p(Hallucination)'). The 'gold' reference text corresponds to the intended reference 'gold' text ('tgt') for MT and DM tasks, while for the PG task, where 'tgt' was mostly not provided in the SHROOM dataset, the model input ('src') served as the 'gold' reference text. Consistent hyperparameters were employed for finetuning vectara across all data combinations: the model was trained for 5 epochs with a batch size of 16 and a warmup of 0.1.

4.3 Model-aware Methods

Hidden States During our experiments with SAPLMA method (see Section 3.3), we used a feedforward neural network as an activations classifier. It features three hidden layers with decreasing numbers of hidden units (256, 128, 64), all utilising ReLU activations. We discovered that this approach requires more data than the validation set could provide. Therefore, the classifier was trained fully on the GPT-3.5 labeled training dataset. For each task, the dataset was fed into the original task model to get outputs of each hidden layer of the decoder for a final decoded token (EOS). Hallucination classifier was trained with binary cross entropy objective where inputs were original model layer activations and outputs were GPT labeled hallucination probabilities. The exact layer number is considered a hyperparameter in this case which was selected by grid search. Further experiments were focused on selecting the exact hidden layer of the original model that may contain most of the information regarding the uncertainty of the model. Based on evaluation on a validation set the best results are different for each task: layer #10 (out of 12) for MT, layer #1 (out of 12) for DM, layer #5 (out of 16) for PG.

Attention Flow For `att-flow`, we conducted our experiments using the `scikit-learn` library (Pedregosa et al., 2011). The feature matrices that we obtained, have dimensions $L \times T$ where L denotes the number of layers and T the number of tokens, can be quite large. To address this, we employed Principal Component Analysis for dimensionality reduction, achieving six components, chosen after a structural evaluation of component ranges across tasks. This dimensionality reduction significantly enhanced the classifier's performance. A Support Vector Machine with a Radial Basis Function kernel and Platt scaling was utilised for deriving probabilities and predictions. Out of alternative kernels, none yielded comparable results. Notably, the SVM was trained independently for each task, recognising that tasks may exhibit distinct attention flows, particularly on model-aware data.

4.4 Ensemble Methods

Finally, we also created simple ensemble models by combining the outputs of the individual systems presented in previous sections. We experimented with a variety of methods including simple voting, regression metamodels, and fusion of predicted probabilities.

Logistic regression The submission `mm-logreg` involved hallucination prediction with a logistic regression model trained on a small set of binary features that correspond to the labels predicted by individual systems. For the model-agnostic track we included labels from 5 systems, `levenshtein`, `paragram`, `vectara-gpt`, `vectara-bertscore`, and `vectara-val-subset`.[5] For the model-aware track, we also included the labels predicted by the SAPLMA method (Section 4.3). For each track, the model was trained on the respective validation dataset, using default settings of the `LogisticRegression` model in the `scikit-learn` library (Pedregosa et al., 2011).

Linear Regression with Raw Probabilities Submission `mm-linreg-probs` followed a similar methodology to `mm-logreg` with the distinction being utilisation of raw probabilities predicted by individual systems instead of labels and employment of linear regression instead of logistic regression. For both the model-agnostic and model-aware tracks we included probabilities predicted by the same systems utilised in `mm-logreg`. Like `mm-logreg`, for each track the model was trained on the respective validation dataset, using default settings of the `LinearRegression` model in the `scikit-learn` library (Pedregosa et al., 2011).

Voting Simple `voting` was implemented as an additional ensemble method, using the same set of systems as for the `mm-logreg` method. For each model output, we counted the number of systems that predicted the `Hallucination` label, the threshold for the number of votes required to make a positive prediction was a parameter of the system that we optimised on the validation sets. For both tracks, the optimal strategy was to require at least 2 votes (2 out of 5 for the model-agnostic track and 2 out of 6 for the model-aware track).

Probability Fusion The `prob-fusion` method is proposed as a weighted average fusion approach for combining predictions from multiple models in hallucination detection. Confidence scores for each model are determined as the squared absolute difference between the model's predicted probability and its neutral point, serving as weights in the fusion process. The final fused probability is obtained as the weighted sum of individual model probabilities:

[5] The metamodel considered probabilities from a Vectara model fine-tuned on either the model-aware or model-agnostic validation set, depending on the track.

Table 1: Accuracy (Acc) and Spearman's rank correlation (Corr) results for each of the proposed models on detecting hallucination on the test datasets.

Model	**Agnostic**		**Aware**	
	Acc	**Corr**	**Acc**	**Corr**
baseline	0.697	0.403	0.745	0.488
levenshtein	0.663	0.362	0.711	0.418
paragram	0.643	0.355	0.685	0.379
vectara-val	0.809	0.723	***0.806***	0.707
SAPLMA	-	-	0.593	0.137
att-flow	-	-	0.61	0.245
mm-logreg	0.801	0.665	0.801	0.636
mm-linreg	***0.817***	***0.737***	0.801	***0.712***
prob-fusion	0.793	0.673	0.783	0.654
voting	0.735	0.597	0.756	0.587

$$P_H = \sum_{i=1}^{N} \left(\frac{|P_i - NP_i|^2}{\sum_{j=1}^{N} |P_j - NP_j|^2} \times P_i \right), \quad (2)$$

where P_H denotes the fused probability of hallucination, and P_i and NP_i denote predicted hallucination probability and neutral point of model i, respectively.

5 Results

Table 1 provides insights into the accuracy and correlation metrics for each method concerning hallucination detection, spanning both model-aware and model-agnostic tracks compared to the `baseline` results introduced by the SHROOM organisers. Compared to the organisers' baseline, which employs the Mistral model in a zero-shot manner for hallucination detection, our best-performing systems are based either on finetuning methods (`vectara-val`) or ensemble approaches (`mm-linreg`). As detailed in Table 1, our leading model in the model-agnostic track, the `mm-linreg`, achieves an accuracy of 81.7% and a Spearman's rank correlation coefficient (ρ) of 0.737 and our leading model in the model-aware track, the `vectara-val`, achieves an accuracy of 80.6% and a Spearman's rank correlation coefficient (ρ) of 0.707 for predicting labels and estimating the probability of hallucination, respectively. According to the official ranking provided by the Helsinki NLP group[6], our team's results, TU Wien team, are ranked 3rd in the model-aware track among 46 participants and 8th in the model-agnostic track among 49 participants. As reported in Table 2 in the Appendix B, the finetuning of Vectara on the validation set improves its accuracy and correlation by 5.06% and 6.48% in the model-agnostic track and 1.5% and 2.3% in the model-aware track, respectively.

6 Analysis

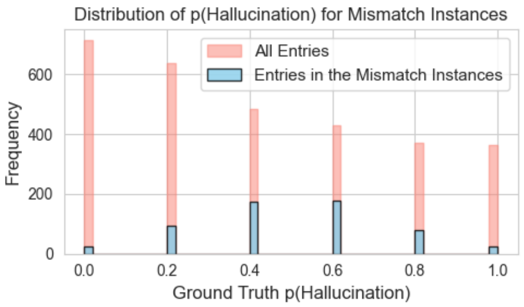

Figure 1: Distribution of hallucination probability for instances with disagreements between our top-performing `mm-linreg` predictions and the ground truth.

Figure 1 presents the distribution of ground truth p(Hallucination) entries in the entire test set encompassing both model-aware and model-agnostic test sets concatenated with those instances where the `mm-linreg` predictions mismatch the ground truth labels. The histogram highlights that the majority of p(Hallucination) entries across the entire ground truth test dataset are concentrated around 0. However, in instances where the metamodel predictions differ from the ground truth labels, there is a notable concentration around 0.4 and 0.6. This observation suggests that the model excels in predicting the correct label for non-controversial cases but encounters challenges in subjective scenarios, where the decision-making process becomes more intricate. Subsequent results, obtained by selectively filtering data based on p(Hallucination) values, reveal nuanced performance metrics: for instances where p(Hallucination) equals only 0 or 1, accuracy of 95.17% and Spearman Correlation (ρ) of 0.796 is achieved; when including p(Hallucination) values of 0, 0.2, 0.8, and 1, the accuracy remains high at 89.21% with a Spearman Correlation (ρ) of 0.731. In contrast, the analysis without filtering, yielding an accuracy of 80.87% and Spearman Correlation (ρ) of 0.724, underscores the impact of data filtering on the model's predictive perfor-

[6] https://helsinki-nlp.github.io/shroom/

mance. This inherent subjectivity in hallucination detection, especially evident with a concentration around the controversial interval, underscores the complex nature of such judgments.

6.1 Qualitative error analysis

Building on the observations drawn from Figure 1, we conducted a more in-depth qualitative analysis of misclassifications within the entire test set. A subsequent in-depth manual analysis of misclassifications was carried out for each task independently. 46% of the misclassifications of the `mm-linreg` belong to the DM task, 33% of them belong to the MT, and 21% of them belong to the PG. For the MT task, we analysed 94 misclassified instances, covering all samples present in the test set for the Ru-En language pair[7], while for the DM and PG tasks, we analysed 20 randomly selected instances. While examining mislabeled samples, a crucial question arises about distinguishing between hallucinations and incorrect answers, such as inaccuracies in the DM task. In MT, ambiguity persists over categorising word-for-word translations, common among non-native speakers, as hallucinations or simply a translation that lacks accuracy. For instance, the model output *Her legs hurt* (MT instance with the id #826), as opposed to the 'gold' reference *Her feet ached*, was labeled by the annotators as a hallucination with a probability of 0.8. The Russian source text, У неё болели ноги, can be translated into English as both *Her feet ached* or *Her legs were hurting*, since the Russian ноги can refer to both *feet* and *legs*. Therefore, the only discrepancy in the model hypothesis lies in the incorrect grammatical form of the verb *hurt*. Analysing the aforementioned example and additional instances available in Appendix C suggests that, despite instructions for annotators to utilise either 'tgt' or 'src' as the 'gold' reference, annotators predominantly labeled the dataset using 'tgt' and 'hyp'. This discrepancy may pose a challenge. Additionally, the process of generating 'tgt' is only clarified for the DM task, with no explanation provided for the MT or PG tasks. The shared task documentation lacks explicit guidance on annotator instructions, stating only that annotators should verify if all information in the hypothesis is supported by the 'gold' reference. This formulation may lead to diverse interpretations of what constitutes a hallucination. Instances such as *If you persecute heretics or <define> discrepants </define> , they unite themselves as to a common defence [...]*[8] underscore the necessity for annotators to possess language proficiency, a requirement challenging to meet in a crowdsourced setting.

The error analysis of disagreements with 'gold' annotations highlights the subjective nature of hallucination detection, presenting a challenge for both machines and human annotators. The absence of a precise definition for hallucination further complicates the task. In NLG tasks like PG or DM, annotators require significant language proficiency or even a linguistic background. Instances with majority-based gold labels and low inter-annotator agreement (probabilities of 0.4 or 0.6) anticipate challenges for models, as these instances are ambiguous even for humans. Furthermore, qualitative analysis of the misclassifications suggests a tendency for annotators to mislabel longer texts. For instance, the model hypothesis for the MT instance with the phrase *The Beer of His Words Back*[9] corresponding to the gold reference *I stand corrected* was labeled by the annotators as a non-hallucination with a probability of 0.2. This may be attributed to the challenge of maintaining concentration when reading longer texts. This observation implies a heightened difficulty in detecting hallucinations in lengthier passages.[10] A comprehensive list of manually verified examples for all three tasks, accompanied by corresponding explanations, is extensively documented in Appendix C.

6.2 Complications with Model-aware Track

Our top-performing systems for both model-aware and model-agnostic test sets are based on model-agnostic approaches. However, the final results for our model-aware methods proved to be less promising, achieving about 60% accuracy on the test set (see Table 1). Possible reasons for the sub-optimal performance of the SAPLMA (4.3) method include the lack of reproducibility instructions for the model-aware track, poor quality of GPT-generated training labels, and the method's original design for decoder-only models, whereas all task models are encoder-decoder models. Results for the Attention Flow method (4.3) could also be enhanced through various techniques such as ad-

[7] The Ru-En language pair was selected because two authors of the paper are native Russian speakers.

[8] DM instance with the id #885 labeled by the annotators as a hallucination with a probability of 0.8

[9] Instance with the id #2251

[10] The average length of 'src' input in the SHROOM test set is 95 characters, or 17 tokens.

justing decay rates, incorporating decoder scores, and introducing feature weights.

During our experiments on model-aware datasets (train & validation set), we encountered challenges reproducing model outputs for two tasks: MT and DM. Using the provided Huggingface models with default parameters yielded different results than those shown in the dataset. Despite experimenting with numerous inference parameters from the Huggingface library, we could not obtain the same input-output pairs ('src'-'hyp') as in the dataset. This discrepancy significantly impacted label alignment, rendering samples labeled as hallucinations no longer hallucinations with newly generated results. This issue is crucial for the model-aware track, as SAPLMA and Attention Flow methods utilise internal data from forward pass for each sample but rely on labels from the dataset. We contend that this problem might substantially reduce the quality of these methods.

7 Conclusion

This paper outlines our systems for the SemEval shared task on LLM hallucination detection, covering both model-aware and model-agnostic subtasks. Our finetuned BERT-based model demonstrated strong performance, securing the 3rd rank in the model-aware track and underscoring the efficacy of our approach. Notably, our leading system in the model-agnostic track employs a metamodel that integrates predictions from diverse systems, including a finetuned BERT-based hallucination detection model, as well as rule-based methodologies and those relying on LLM hidden states.

The absence of a universally agreed-upon definition for hallucination complicates both human and machine evaluations, as evident in the error analysis of Section 6.1. Although attempts have been made to systematically define hallucination, such as by (Huang et al., 2023) and (Ji et al., 2022), the NLP community's understanding remains broad. This encompasses scenarios where a model outputs entirely false information or information close to the desired output but incomplete. Human annotators bring their world knowledge and views, influencing annotations and subsequently affecting model performance. Hallucination detection is challenging for both machines and humans, with achieving high inter-annotator agreement proving particularly difficult when the task definition is overly broad. Especially tasks like paraphrase generation or definition modeling, where numerous correct outputs are possible, are inherently subjective and tied to the annotator's real-world knowledge and beliefs (Heidegger, 2001; Honovich et al., 2022). A clearer definition of the annotation task, specifically detailing what constitutes hallucination for that task, could potentially enhance inter-annotator agreement and subsequently improve model performance. Detecting hallucinations across various NLP tasks poses a significant challenge. The SHROOM dataset encompasses three distinct tasks, whereas existing hallucination detection benchmarks often address only a single task.

References

Amos Azaria and Tom Mitchell. 2023. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976, Singapore. Association for Computational Linguistics.

S. Bird, E. Klein, and E. Loper. 2009. *Natural language processing with Python*. O'Reilly Media.

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. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Joseph F DeRose, Jiayao Wang, and Matthew Berger. 2020. Attention flows: Analyzing and comparing attention mechanisms in language models. *IEEE Transactions on Visualization and Computer Graphics*, 27:1160–1170.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

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

Martin Heidegger. 2001. On the Essence of Truth. In *The Nature of Truth: Classic and Contemporary Perspectives*. The MIT Press. _eprint: https://direct.mit.edu/book/chapter-pdf/2300694/9780262278690_car.pdf.

Felix Hill, Roi Reichart, and Anna Korhonen. 2014. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. TRUE: Re-evaluating factual consistency evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3905–3920, Seattle, United States. Association for Computational Linguistics.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Wenliang Dai, Andrea Madotto, and Pascale Fung. 2022. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55:1 – 38.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language models (mostly) know what they know.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics – doklady*, 10(8):707–710.

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6449–6464, Singapore. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Tianyu Liu, Yizhe Zhang, Chris Brockett, Yi Mao, Zhifang Sui, Weizhu Chen, and Bill Dolan. 2022. A token-level reference-free hallucination detection benchmark for free-form text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6723–6737, Dublin, Ireland. Association for Computational Linguistics.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. Entity-based knowledge conflicts in question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7052–7063, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.

Timothee Mickus, Elaine Zosa, Raúl Vázquez, Teemu Vahtola, Jörg Tiedemann, Vincent Segonne, Alessandro Raganato, and Marianna Apidianaki. 2024. Semeval-2024 shared 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 1980–1994, Mexico City, Mexico. Association for Computational Linguistics.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and Samuel Bowman. 2022. QuALITY: Question answering with long input texts, yes! In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5336–5358, Seattle, United States. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in

Python. *Journal of Machine Learning Research*, 12:2825–2830.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, and Jianfeng Gao. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback.

Lingfeng Shen, Lemao Liu, Haiyun Jiang, and Shuming Shi. 2022. On the evaluation metrics for paraphrase generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3178–3190, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Aviv Slobodkin, Omer Goldman, Avi Caciularu, Ido Dagan, and Shauli Ravfogel. 2023. The curious case of hallucinatory (un)answerability: Finding truths in the hidden states of over-confident large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3607–3625, Singapore. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. From paraphrase database to compositional paraphrase model and back. *Transactions of the Association for Computational Linguistics*, 3:345–358.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems*, volume 34, pages 27263–27277. Curran Associates, Inc.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

Wei Zhao, Michael Strube, and Steffen Eger. 2023. DiscoScore: Evaluating text generation with BERT and discourse coherence. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3865–3883, Dubrovnik, Croatia. Association for Computational Linguistics.

Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Francisco Guzmán, Luke Zettlemoyer, and Marjan Ghazvininejad. 2021. Detecting hallucinated content in conditional neural sequence generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1393–1404, Online. Association for Computational Linguistics.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

A Prompt for GPT-3.5

```
You will be provided with a Sentence
and your task is to rate the
consistency of that sentence to
that of the provided Context. Your
answer must be only a number between
0.0 and 1.0 rounded to the nearest two
decimal places where 0.0 represents no
consistency and 1.0 represents perfect
consistency and similarity.
Reply with a valid JSON in following
format:
{"answers":{"<pair_id>": <float>}}.
Example:
{"answers":{"0":0.7,"12":0.33}}.
Array of answers should contain reply
for each Context/Sentence pair.
```

Listing 1: Dataset labeling prompt

B Vectara performance

The performance of the Vectara finetuned on different datasets is shown in Table 2.

Table 2: The performance (Accuracy (Acc) and Spearman's rank correlation (Corr) of the Vectara finetuned on different datasets.

Model	Agnostic		Aware	
	Acc	**Corr**	**Acc**	**Corr**
Standard vectara	0.770	0.679	0.794	0.691
vectara-gpt-val	0.745	0.69	0.776	0.695
vectara-bertscore	0.661	0.421	0.705	0.455
vectara-val	***0.809***	***0.723***	***0.806***	***0.707***
vectara-gpt	0.697	0.706	0.772	0.699

C Disagreement in Annotations

During the examination of misclassifications in MT, it was observed that approximately 15% (13 out of 94) of instances pertaining to the Ru-En language pair could be subject to different labeling by the authors of this paper. Furthermore, a detailed analysis of 20 randomly selected misclassifications from the PG and DM segments of the test set, revealed notable discrepancies. Specifically, 20% for the PG task (4 out of 20) and approximately 35% (7 out of 20) in the DM task would receive distinct labels according to the authors' assessment. Tables highlighting these disagreements for each of the three tasks are provided below for reference.

The MT example below was labeled by the annotators as a non-hallucination with a probability of 0.2 (instance with the id #2251 in the model-agnostic test set)[11]:

src: Беру свои слова обратно.
tgt: I stand corrected.
hyp: The Beer of His Words Back.

Our top-performing mm-linreg labeled this instance as a hallucination with a probability of 0.54.

As previously deliberated, the categorisation of a translation lacking natural fluency as a hallucination is subject to scrutiny. This contention is exemplified by the following MT example (instance with the id #826 in the model-agnostic test set):

src: У неё болели ноги.
tgt: Her feet ached.
hyp: Her legs hurt.

This instance was labeled by the annotators as a hallucination with a probability of 0.8, while our mm-linreg labeled it as a non-hallucination with a probability of 0.2. In Russian, ноги stands for both *feet* and *legs*. The only discrepancy in the 'hyp' lies in the incorrect grammatical form of the verb *hurt*. An English native speaker would likely formulate a sentence such as *her legs were hurting*.

Upon scrutinizing misclassified instances in machine translation, discernible patterns emerged where our model encounters challenges. Specifically, it grapples with correctly identifying the presence of a suitable pronoun in the English translation (instance with the id #1791 in the model-agnostic test set):

src: Кажется, он сломан.
tgt: I think it's broken.
hyp: Looks like he's broken.

The form сломан represents a short form of the past participle of the perfective aspect of the verb сломать. This verb form is exclusively applicable when describing an inanimate object that has been broken. Notably, due to the gender agreement of Russian pronouns with the associated noun, the sentence includes the pronoun он, corresponding to the English pronoun *he*. To express the idea of an animate object being broken, as in the English phrase *he is broken*, the Russian equivalent would be он сломлен, where сломлен represents a short form of the past participle of the perfective aspect of the verb сломить. This sample was labeled by the annotators as a hallucination with a probability of 0.8. Our best-performing model labeled it as a non-hallucination with a probability of 0.2. It is pertinent to note that the annotation of such instances lacked consistency (instance with the id #69 in the model-agnostic test set):

[11] The entire sample can be found in Table 4

src: На сколько врач тебе дал освобождение?
tgt: How long did the doctor sign you off school for?
hyp: How much did the doctor give you your release?

The annotators assigned a non-hallucination label to this instance with a probability of 0.4. Conversely, our best-performing model categorised it as a hallucination with a probability of 0.8. The English 'hyp' sentence appears somewhat unconventional, and alternatives such as *How long did the doctor grant you a release?* or *For how long did the doctor excuse you?* would convey a more natural phrasing. Notably, it is essential to acknowledge that the target ('tgt') reference, referred to as the 'gold' text, does not provide a fully accurate translation of the source Russian sentence. In the original Russian sentence, освобождение (translated as *release*) does not exclusively refer to a school release authorized by a doctor. Furthermore, it is pertinent to inquire about the methodology employed by the creators of the SHROOM dataset in generating the 'gold' text for the MT and PG sections, as this information is elucidated solely for the DM part of the dataset, indicating that the gold definition is sourced from Wiktionary.

It is imperative to acknowledge that our models were exclusively trained utilising 'tgt' and 'hyp' for both MT and DM, i.e. disregarding 'src'. Consequently, this means that our models cannot possess the capability to comprehend certain grammatical nuances of the Russian language, as the models were not trained on the Russian text. The decision to employ 'tgt' as a reference was motivated by the lack of statistical data regarding the languages encompassed in an MT part of the dataset since we could not use 'src' without specifying the language for a range of our approaches including finetuning of a BERT-based hallucination detection model. Considering the potential impact on system performance, an alternative approach could involve incorporating 'src' and 'hyp', or even 'src', 'tgt', and hyp' for the automatic generation of probabilities using GPT-3.5 for at least an MT part of the dataset. However, the quality of GPT-3.5 for low-resource languages cannot guarantee promising results for the plethora of languages encapsulated in the SHROOM dataset. Despite this caveat, such an approach may empower the model trained on such sort of automatically generated probabilities to better 'learn' the linguistic idiosyncrasies of a given language, which cannot be adequately captured solely through reliance on 'tgt'.

Below you can find an example for a PG task (PG instance with the id #2907 in the model-agnostic test set):

src: Does POTUS know?
tgt: Is the president aware of this?
hyp: Does POTUS know what you're doin' with?
gold_label: Not Hallucination
gold_prob: 0.2
label_metamodel: Hallucination
prob_metamodel: 0.78

A range of instances demonstrates that the SHROOM dataset annotation requires language proficiency from the annotators that cannot be expected in a crowdsourced setting[12] (DM instance with the id #885 in the model-agnostic test set):

src: If you persecute heretics or <define> discrepants </define> , they unite themselves as to a common defence [...]
tgt: (archaic) A dissident.
hyp: (obsolete) One who disagrees; a dissenter.
gold_label: Hallucination
gold_prob: 0.8
label_metamodel: Not Hallucination
prob_metamodel: 0.35

It is noteworthy to highlight that across all three tasks encompassed in the dataset, there exists a subset of instances that were correctly labeled but received a probability very proximate to 0.5. This scenario poses a potential challenge for models trained on such data (instance with the id #2998 in the model-aware test set):

src: The war with Serbia must be the last of Europe's civil wars.
tgt: -
hyp: Europe must end with the war with Serbia.
gold_label: Hallucination
gold_prob: 0.6
label_metamodel: Non-Hallucination
prob_metamodel: 0.4

[12] Organisers of the SHROOM shared task reported that the SHROOM validation dataset was built through crowdsourcing with 200 annotators involved annotating 20 instances each. Details about the test set annotations were not provided.

Table 3: PG Disagreement

id	Source	Target	Hypothesis	GT	GT p(H)	Pred	Pred p(H)
1150	I'll take the air.	I'm going out for some air.	I'm gonna take the air, mister.	Not Hallucination	0.40	Hallucination	0.58
2187	That was yesterday.	Well, it was a day ago now.	That was yesterday's news.	Not Hallucination	0.20	Hallucination	0.62
2907	Does POTUS know?	Is the president aware of this?	Does POTUS know what you're doin 'with?	Not Hallucination	0.20	Hallucination	0.79
674	Estimates of the number of displaced persons within Kosovo are now up to 850 000, according to NATO.	-	NATO says there are 850 000 displaced people in Kosovo.	Hallucination	0.60	Not Hallucination	0.14

Table 4: MT Disagreement

id	Source	Target	Hypothesis	GT	GT p(H)	Pred	Pred p(H)
55	Ты правда этого не знал?	Did you really not know that?	You don't know what?	Not Hallucination	0.40	Hallucination	0.58
69	На сколько врач тебе дал освобождение?	How long did the doctor sign you off school for?	How much did the doctor give you your release?	Not Hallucination	0.40	Hallucination	0.80
479	Никто не узнает.	No one will find out.	No one knows.	Not Hallucination	0.20	Hallucination	0.62
745	Я плохо спал вчера ночью, но хорошо спал ночью сегодня.	I didn't sleep well the night before last, but I slept well last night.	I slept badly last night, but slept well tonight.	Not Hallucination	0.40	Hallucination	0.83
836	Я лучше не буду отвечать на этот вопрос.	I'd rather not answer that question.	I better not answer that question.	Hallucination	0.60	Not Hallucination	0.17
846	Ты всё испортил!	You've ruined it!	You ruined everything!	Hallucination	1.00	Not Hallucination	0.47
1456	Тому пришла в голову блестящая идея.	Tom had a bright idea.	That's why a brilliant idea came to mind.	Not Hallucination	0.40	Hallucination	0.82
2136	Я жалею, что зря потратил на это своё время.	I regret wasting my time on that.	I regret the fact that I spent my time here.	Not Hallucination	0.20	Hallucination	0.67
2251	«Глупых вопросов не бывает». – «Как мог Леонардо Дикаприо изобрести Мону Лизу, если в XIX веке не было цвета?» – «Беру свои слова обратно».	There's no such thing as a stupid question. "How did Leonardo DiCaprio invent the Mona Lisa if there was no color in the 1800s?" "I stand corrected."	"There are no stupid questions." — "How could Leonardo Dicaprio discover Mona Lisa if there was no color in the 19th century?" — "The Beer of His Words Back."	Not Hallucination	0.20	Hallucination	0.54
2326	Она повторно вышла замуж, когда ей было за сорок.	She remarried when she was in her mid-forties.	She married again when she was 40.	Not Hallucination	0.40	Hallucination	0.72
2634	Как жизнь, Майк? - "Меня Том зовут".	How are you doing, Mike? "My name is Tom."	How's life, Mike? - "I'm Tom."	Hallucination	0.60	Not Hallucination	0.21

Continuation of Table 4

id	Source	Target	Hypothesis	GT	GT p(H)	Pred	Pred p(H)
2642	Как думаешь, ты не мог бы внести десять долларов на подарок Тому ко дню рождения?	Do you think you could pitch in $10 for Tom's birthday present?	How do you think you wouldn't be able to bring ten dollars for a gift because of that birthday?	Not Hallucination	0.40	Hallucination	0.81
2727	Стоматологи рекомендуют менять зубную щётку каждые три месяца, потому что со временем её щетина всё хуже удаляет зубной налёт, а также в ней скапливаются микробы.	Dentists recommend to change toothbrushes every three months, because over time their bristles become worse at getting rid of plague, as well as accumulate microbes.	Dentists recommend changing the toothbrush every three months, because over time its bristle increasingly removes plaque, as well as microbes accumulate in it.	Not Hallucination	0.40	Hallucination	0.65

Table 5: DM Disagreement

id	Source	Target	Hypothesis	GT	GT p(H)	Pred	Pred p(H)
61	A grand distinction is to be drawn, in this respect, between the <define> swell mob </define> and common thieves; the former being, for the most part, men of the world, of some education — not appearing at all flash (thief - like), but, on the contrary, acting the part of gentlemen in society.	(archaic, slang) Well-dressed thieves and swindlers, regarded collectively.	(slang, dated) A group of thieves.	Not Hallucination	0.40	Hallucination	0.62
257	This is so because, as Kant already taught, the nonconsensual transfer of goods is only compatible with freedom when […]	In an omnilateral fashion.	In an omnidirectional manner.	Hallucination	0.60	Not Hallucination	0.14
885	If you persecute heretics or <define> discrepants </define>, they unite themselves as to a common defence […]	(archaic) A dissident.	(obsolete) One who disagrees; a dissenter.	Hallucination	0.80	Not Hallucination	0.35
266	Whilst the viewshed quantifies visibility for a limited set of test locations...	The view from a particular vantage point.	The area of a building or other structure that provides a view.	Not Hallucination	0.20	Hallucination	0.71
1405	Some areas were deluged with a month's worth of rain in 24 hours.	To flood with water.	To flood; to overwhelm.	Hallucination	0.60	Not Hallucination	0.44
1685	Through it, through what takes place, the celebrants try to obtain a result, to influence the course of the hoped for or dreaded events that either depend on the current dispositions of a divinity or […]	A person who officiates at a religious ceremony, especially a marriage or the Eucharist.	One who holds a ceremony.	Not Hallucination	0.40	Hallucination	0.53