

A Hypothesis-Driven Framework for the Analysis of Self-Rationalising Models

Marc Braun

Linköping University

University of Stuttgart

Fraunhofer IPA

marc.braun@ipa.fraunhofer.de

Jenny Kunz

Linköping University

jenny.kunz@liu.se

Abstract

The self-rationalising capabilities of LLMs are appealing because the generated explanations can give insights into the plausibility of the predictions. However, how faithful the explanations are to the predictions is questionable, raising the need to explore the patterns behind them further. To this end, we propose a hypothesis-driven statistical framework. We use a Bayesian network to implement a hypothesis about how a task (in our example, natural language inference) is solved, and its internal states are translated into natural language with templates. Those explanations are then compared to LLM-generated free-text explanations using automatic and human evaluations. This allows us to judge how similar the LLM's and the Bayesian network's decision processes are. We demonstrate the usage of our framework with an example hypothesis and two realisations in Bayesian networks. The resulting models do not exhibit a strong similarity to GPT-3.5. We discuss the implications of this as well as the framework's potential to approximate LLM decisions better in future work.

1 Introduction

With the increasing capabilities of large language models (LLMs), more and more tasks that were traditionally solved using human experts and statistical models are now aided by LLMs. Understanding how a model produces its output is an essential factor in the human acceptance of machine learning systems (Shin, 2021). However, understanding the connection between input and output in LLMs is not easily possible (Adadi and Berrada, 2018).

Recent advances in LLMs generating longer coherent text have popularised self-rationalising models, which produce a natural language explanation (NLE) alongside their output (Hase et al., 2020; Marasovic et al., 2021). NLEs have numerous benefits over other, non-textual explanations: NLEs are valued more highly by human users (Forrest

et al., 2018), they can be applied to a broad range of problems and they can combine external knowledge with the model input. However, even though the NLEs can give insights into how plausible the predictions made by LLMs are, the *faithfulness* of the explanations to the prediction process remains at best uncertain (Wiegrefe et al., 2021; Atanasova et al., 2023; Turpin et al., 2023).

In this work, we propose exploring the patterns behind generated NLEs using a hypothesis-driven framework, with the ultimate goal of deriving a surrogate model. Our framework is centred around a *hypothetical global explanation* (HGE): A hypothesis about how the LLM solves a specific task on a global, structural level. While we start off with an obviously oversimplified hypothesis to introduce and test the framework, we envision that it can be incrementally adapted to more refined hypotheses in the future. The patterns captured by each refinement step can then serve to measure their coverage, or e-recall (Goldberg, 2023), in the LLM.

The core component of our framework is a statistical surrogate model (SSM) that reflects the HGE. We propose using a Bayesian Network (Pearl, 1988) with a manually designed structure as a framework for the SSM, as the Bayesian Network allows us to visualise the independencies among the random variables used in the SSM via a directed acyclic graph (DAG). This allows us to define the variables in the SSM and the relationships among them such that they reflect the HGE. Furthermore, since the structure of the SSM is based on the HGE, each variable of the Bayesian Network is assigned a specific, semantically interpretable meaning. This allows us to generate local NLEs for individual inputs based on the internal state of the Bayesian Network. In the last step, we compare both the predicted labels and the NLEs produced by the SSM to those produced by the LLM in order to gain insights about the faithfulness of the HGE.

We demonstrate the usage of this framework

with an exemplary HGE for how the behaviour of the LLM GPT-3.5 (Brown et al., 2020) can be explained when performing English-language natural language inference (NLI). We discuss the challenges when designing, implementing and training the Bayesian Networks based on the HGE and outline the next steps on the way to a surrogate that models LLM predictions more accurately.

2 Related Work

Self-rationalising models have received increasing attention as the generation abilities of NLP models have improved in recent years. Human-annotated datasets such as e-SNLI (Camburu et al., 2018) for NLI, CoS-E (Rajani et al., 2019) and ECQA (Aggarwal et al., 2021) for commonsense question answering and ComVE (Wang et al., 2020) for commonsense validation are the basis for much NLE work (Wiegrefe and Marasovic, 2021). However, the role of LLMs in the annotation process itself is likely increasing (Wiegrefe et al., 2022), as it reduces the significant cost of human annotation (Belinkov and Glass, 2019).

A concern, however, is that the generation of the NLEs is as opaque as the prediction process. To address this, some recent work explores the faithfulness of NLEs: Wiegrefe et al. (2021) show that prediction and explanation exhibit correlated responses to input noise. Atanasova et al. (2023) propose analysing NLEs after counterfactual interventions on the input that alter the prediction, and testing the sufficiency of the reasons provided in the NLE. Turpin et al. (2023) show that biased features in the input lead to obviously unfaithful NLEs as those features affect the predictions heavily but are never mentioned in the generated NLEs.

Similar concerns have been raised for model-agnostic surrogates such as LIME (Ribeiro et al., 2016) and SHAP (Lundberg and Lee, 2017), which are widely used for highlighting input features as a form of explanation. They approximate a complex model by training a simpler, interpretable model on the original model’s predictions. Whether such surrogates are helpful for understanding complex models is subject to discussion. Rudin (2019) makes the case against their usage in high-stake decisions and argues for the deployment of models with an interpretable decision process. However, from a more practical perspective, Jacovi and Goldberg (2020) point out that faithfulness is not a binary feature but should be seen as a scale. They argue that

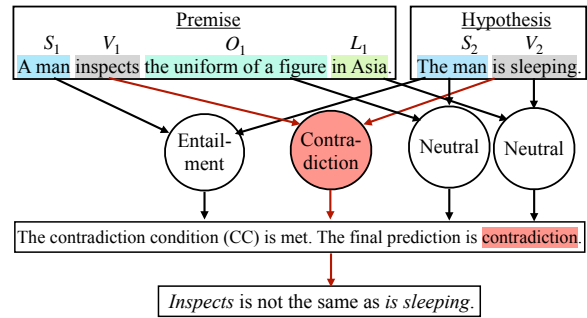


Figure 1: An illustrative (simplified) example for the small SSM. The input X consists of the subphrases of the premise and hypothesis. The circles are the hidden variables Z , followed by the final prediction Y (here, *contradiction*) and a template-based NLE (lowest box).

depending on the use case, it can be more important to have *plausibility* than *faithfulness*.

3 Proposed Framework

In this section, we describe our framework¹: How we design the SSM based on our hypothesis (§3.1-§3.3), how we determine and learn the parameters (§3.4) and how we generate NLEs based on the SSM’s parameters (§3.5). We demonstrate that we can successfully control for the assumptions made in our hypothesis, with a model that is intuitive to construct and understand. For a simplified demonstration of how the framework processes an example input, we refer to figure 1.

3.1 Constructing the SSM

The SSM aims to reflect a hypothesis about a decision-making process that is assumed to explain the behaviour of GPT-3.5 (Brown et al., 2020)² when performing NLI. A hypothetical global explanation (HGE) is defined that aims at providing a hypothesis on how GPT-3.5 produces its labels when performing NLI. The hypothesis is based on an intuitive yet oversimplified assumption about how a fully connected model such as the Transformer network (Vaswani et al., 2017) behind GPT-3.5, may solve the NLI task. The HGE is as follows:

¹The implementation used for our experiments is available at https://github.com/Marbr987/Hypothesis_Driven_Analysis_of_Self_Rationalising_Models.

²We use the *text-davinci-003* version that deviates from the cited paper by being fine-tuned using RLHF (Ouyang et al., 2022), but possibly even in other aspects.

HGE: “When performing NLI, GPT-3.5 compares pairs of subphrases from the premise and hypothesis to each other and classifies each pair into contradiction, entailment, or neutral. Based on this classification, the final prediction is made using deterministic rules.”

3.2 Extracting Subphrases from Premise and Hypothesis

The aim of NLI is to classify the relation of a *premise* and a *hypothesis* into *contradiction*, *entailment*, and *neutral* relation. We use the e-SNLI dataset (Camburu et al., 2018) that besides pairs of premises and hypotheses and the according NLI label also contains human-authored NLEs. An example of a premise and a hypothesis is given in figure 1, where the hypothesis “The man is sleeping” contradicts the premise “A man inspects the uniform of a figure in Asia”.

To implement the structure of the HGE, we define the subphrases to be the subject, verb, and object of the sentences as well as location and clothing of the subjects mentioned in the sentences.³ We extract subphrases using the syntactic dependency trees of the sentences as defined by the SpaCy dependency parser (Honnibal and Johnson, 2015). Sentences that contain multiple subjects, verbs, or objects were discarded from the dataset. The remaining data contains 30.8% of the data in the original dataset. The individual words in each subphrase are transformed into 300-dimensional vectors using Spacy’s pre-trained word embeddings (Honnibal and Montani, 2017). The embedding vectors of the individual words are then added up. If multiple location or clothing subphrases are extracted, the embedding vectors are also added up.

3.3 Defining the Structure of the SSM

The random variables (RVs) used in the SSM and the structure of the SSM are deduced from the HGE and defined using a Bayesian Network. A Bayesian Network is a graphical statistical tool to visually represent independencies among RVs. The RVs in a Bayesian Network are represented as nodes in a directed acyclic graph (DAG), where an edge from node A to node B is interpreted as A causes B .

³While we chose the categories after a manual inspection of the training data, they are obviously incomplete. More categories would increase the complexity of the Bayesian Network, but ensure better coverage.

Defining the input variables X . We represent each subphrase in the premise and hypothesis as a 300-dimensional vector as described in 3.2. Let S_i, V_i, O_i, L_i and C_i be RVs representing the *subject*, *verb*, *object*, *location*, and *clothing* subphrase of sentence i , where sentence 1 is the premise and sentence 2 is the hypothesis. For notation purposes, we introduce the following random vectors:

$$\begin{aligned} X_1 &:= (x_{1,1}, x_{1,2}, x_{1,3}, x_{1,4}, x_{1,5})^T \\ &:= (S_1, V_1, O_1, L_1, C_1)^T \\ X_2 &:= (x_{2,1}, x_{2,2}, x_{2,3}, x_{2,4}, x_{2,5})^T \\ &:= (S_2, V_2, O_2, L_2, C_2)^T, \\ X &:= (X_1, X_2)^T \end{aligned}$$

We also define the sets of random vectors:

$$\begin{aligned} \mathcal{X}_1 &:= \{S_1, V_1, O_1, L_1, C_1\}, \\ \mathcal{X}_2 &:= \{S_2, V_2, O_2, L_2, C_2\}, \\ \mathcal{X} &:= \mathcal{X}_1 \cup \mathcal{X}_2 \end{aligned}$$

Introducing hidden variables Z In order to model the assumption in the HGE that the pairs of subphrases are classified independently, a set of unobserved discrete RVs \mathcal{Z} is introduced. As shown in Figure 2, we define that each element $z_{k,l} \in \mathcal{Z}$ is caused by $x_{1,k}$ and $x_{2,l}$ where $x_{1,k}$ is the k -th element in X_1 and $x_{2,l}$ is the l -th element in X_2 . Let Z be the random vectors with entries equal to the elements in \mathcal{Z} . According to the HGE, each $z_{k,l}$ is a discrete RV with possible realisations *contradiction*, *entailment*, or *neutral*. In other words, each hidden RV $z_{k,l}$ models the relation of subphrase $x_{1,k}$ from the premise to the subphrase $x_{2,l}$ from the hypothesis.

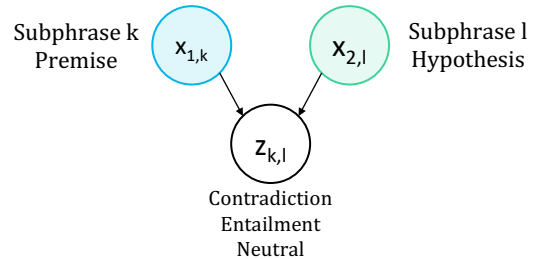


Figure 2: Relationship of any $z_{k,l} \in \mathcal{Z}$ to its parents

Depending on which combinations of k and l are included in \mathcal{Z} , the structure of the Bayesian Network changes and different interpretations of

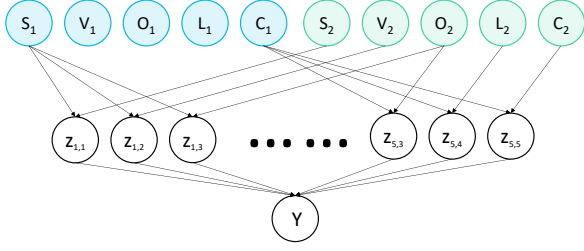


Figure 3: Structure of the SSM_{large} expressed as a Bayesian Network

the HGE can be modelled. We compare two different structures: A large SSM containing all possible connections, and a small SSM that only contains the connections that we hypothesise are relevant for the final prediction.

Defining Z for the large SSM For the first structure (visualised in figure 3), Z contains all possible combinations of k and l , following the assumption that any subphrase from the premise can contradict, entail, or be neutral towards every subphrase in the hypothesis. Let this structure be called *large SSM*. In mathematical terms, we define

$$\mathcal{Z}_{large} := \{z_{k,l} | k, l \in \{1, 2, 3, 4, 5\}\}$$

Defining Z for the small SSM For the second structure, we remove all random variables in Z that we assume to be in the neutral state most of the time and therefore do not contain much relevant information for the prediction. These uninformative z are defined to be the ones that model the *inter*-relationships between the subject, verb, and object subphrases. For example, the subject of the premise rarely contradicts or entails the object of the hypothesis. In mathematical terms, we define

$$\mathcal{Z}_{small} := \mathcal{Z}_{large} \setminus \{z_{k,l} | k, l \in \{1, 2, 3\} \wedge k \neq l\}$$

Defining the output Y According to the HGE, we derive the final prediction from the subphrase classification using deterministic rules. The results from the subphrase classification are the values of Z . Consequently, the final prediction Y is made by defining directed edges from all variables in Z to Y . Y is the discrete RV representing the overall class of the NLI task and therefore has possible states *contradiction*, *entailment*, or *neutral*.

3.4 Determining the Parameters of the SSM

The aim of the SSM is to make predictions on $Y|X$, i.e. the final class given the subphrases, and to generate NLEs for why the prediction was made. The probability of $Y|X$ can be calculated as follows.

$$P(Y|X) = \sum_Z P(Y, Z|X) \quad (1a)$$

$$= \sum_Z P(Y|Z, X) \cdot P(Z|X) \quad (1b)$$

$$= \sum_Z P(Y|Z) \cdot P(Z|X) \quad (1c)$$

The summation is performed over all possible states of the vector Z and the equality in 1c holds because of the independence of Y and X given Z which can be verified using the Bayesian Network structure. Given Equation 1, we model the distributions of $Y|Z$ and $Z|X$ to make inference on $Y|X$.

3.4.1 Defining the Deterministic Distribution of $Y|Z$

Following the HGE, the distribution of $Y|Z$ can be modelled with a set of rules we define using propositional logic. Those rules are:

Contradiction Condition The RV Z models the relationship between the subphrases X . If any of the variables in Z has the value *contradiction*, i.e. if any subphrase in the premise contradicts any subphrase in the hypothesis, we define the final prediction Y to be *contradiction*.

$$\exists z \in Z : (z = \text{“contradiction”}) \quad (CC)$$

We call this formula *contradiction condition* (CC). If it evaluates to `True`, then Y is defined to be *contradiction*.

Entailment Condition If the final prediction Y is *entailment*, every subphrase in the hypothesis (i.e. every element in \mathcal{X}_2) has to be entailed by at least one subphrase in the premise (i.e. by any element in \mathcal{X}_1). This condition is expressed as follows:

$$\forall l \in \{1, 2, 3, 4, 5\} \exists k \in \{1, 2, 3, 4, 5\} : (z_{k,l} = \text{“entailment”}) \quad (EC)$$

We call this formula *entailment condition* or (EC). If it evaluates to `True`, then Y is defined to be *entailment*.

Neutral Condition If the (CC) and (EC) are both evaluated as `False`, then we define the final class to be *neutral*. The logical formula for this condition can be expressed as

$$\neg(\text{CC}) \wedge \neg(\text{EC}) \quad (\text{NC})$$

and is called *neutral condition* or (NC).

Distribution of $Y|Z$ Given these three conditions, the distribution of $Y|Z$ can be described as

$$\begin{aligned} P(Y = \text{“contradiction”}|Z) &= 1 \text{ if } (\text{CC}) \\ P(Y = \text{“entailment”}|Z) &= 1 \text{ if } (\text{EC}) \\ P(Y = \text{“neutral”}|Z) &= 1 \text{ if } (\text{NC}) \end{aligned} \quad (2)$$

3.4.2 Learning the Parameters θ of the Stochastic Distribution of $Z|X, \theta$

To be able to evaluate Equation 1, we must determine the distribution of $Z|X$. Let the parameters of this distribution be θ , i.e. the goal is to find θ that models the distribution of $Z|X, \theta$. Since all random vectors in \mathcal{Z} are independent of each other given X , it suffices to model the distribution of each individual random vector $z_{k,l} \in \mathcal{Z}$. In this paper, the distribution of each $z_{k,l}|X, \theta$ is modelled by a feed-forward neural network. From the structure of the Bayesian Network we know that each $z_{k,l}$ only has parents $x_{1,k}$ and $x_{2,l}$ which means that to model $z_{k,l}|X, \theta$, each neural network only needs to take $x_{1,k}$ and $x_{2,l}$ as input.

To optimise the parameters θ , we use the EM algorithm (Dempster et al., 1977). In Appendix A.1 we show that the EM algorithm for the problem at hand amounts to taking samples \tilde{Z} from the distribution of $Z|X, Y, \theta^{(t)}$ in the E-step and maximising the log-likelihood of the parameter θ with respect to the sampled values \tilde{Z} in the M-step. Since each distribution $z_{k,l}|X, \theta_{k,l}$ is modelled using a neural network, the M-step amounts to training each network on the sampled values $\tilde{z}_{k,l}$ as output and the respective observed values of X as input using the cross-entropy loss function (de Boer et al., 2005).

3.5 Generating NLEs Using the SSM

In the last step, NLEs are produced using the SSM. By comparing these NLEs with the NLEs produced by GPT-3.5 in 4.1, we gain insights about the extent to which the SSM reflects a similar reasoning path as the NLEs produced with GPT-3.5.

By defining the distribution of $Y|Z$ based on a set of rules, we can deduce NLEs from the random vectors in \mathcal{Z} . For example, the (CC) formula states that if any of the random vectors in \mathcal{Z} is classified as a contradiction, then Y is of class contradiction

as well. This means that if, for example, the random vector $z_{k,l} \in \mathcal{Z}$ is of class contradiction (and assuming all other elements in \mathcal{Z} are of class other than contradiction), it is reasonable to state that $z_{k,l}$ is responsible for the final prediction. This example illustrates that, by defining the relationship between Z and Y in a way that is semantically interpretable, explanations for the final prediction can be formulated that follow the predefined semantics of the hidden variables.

Based on the rules used to define the distribution $Y|Z$, the templates that provide reasons for the final prediction Y are defined. The complete templates can be found in Appendix A.2.

4 Comparison to the LLM

Now that we have constructed the SSM, we need to compare it to GPT-3.5. To that end, we first generate labels and explanations with GPT-3.5 (§4.1), and compare them to the SSM in human (§4.2.1) and automatic (§4.2.2) evaluations.

4.1 Generating Labels and Explanations with GPT-3.5

To compare the outputs of GPT-3.5 to the outputs of SSM, we prompt the GPT-3.5 model to generate entailment labels as well as NLEs that provide reasoning for why the respective label was chosen.

Performing NLI with GPT-3.5 using few-shot learning has shown to have a performance close to fine-tuned GPT-3.5 (Marasovic et al., 2021). Inspired by the prompts proposed by Marasovic et al. (2021), we use the following structure.

First, the instruction “Classify into entailment, neutral, and contradiction and justify the decision.” is given to the model, followed by six examples⁴, each of which has the following structure:

“Premise: *premise*
Hypothesis: *hypothesis*
Label: *label*
Explanation: *explanation*”

The examples are balanced among the classes, i.e. two examples are chosen at random from the training set for each class. The examples are followed by the premise and hypothesis which shall be classified. Based on this input, GPT-3.5 produces a label and an NLE.

⁴While the original work by Marasovic et al. (2021) uses 50 examples, we found six to be sufficient for the newer, instruction-tuned GPT models.

4.2 Evaluating the Similarity Between the SSM and GPT-3.5 for NLI

In comparison to surrogate models for input feature highlighting surrogates such as LIME, the proximity of our surrogates to the original model cannot be measured by performance only. Therefore, we introduce an evaluation setup consisting of a human evaluation and a set of automatic metrics that compare the predictions and NLEs produced by GPT-3.5 to the predictions and NLEs produced by the SSM. This setup allows us to assess the extent to which the HGE is a valid explanation for how GPT-3.5 solves the NLI task.

4.2.1 Human Evaluation

For the human evaluation, three raters were instructed to answer the following questions for a random sample of 100 explanations produced for the development set. The full instructions for the raters can be found in Appendix A.3.

The raters were asked to report which of the subphrases defined in Section 3.2 are mentioned in each NLE produced by GPT-3.5. The reported subphrases can then be compared to the subphrases used in the explanation produced by the SSM. Furthermore, annotators were instructed to indicate whether the GPT-3.5 explanation relates any subphrases from the premise or hypothesis to each other, i.e. if it has a structure similar to what is stated in the HGE.

Additionally, the raters were asked to assess whether the GPT-3.5 NLE supports the predicted label and whether the explanation is factually correct. Lastly, the factual correctness of the NLEs produced by the SSM was assessed by the raters. To assess the inter-rater agreement among the human raters, we report Fleiss' Kappa (Fleiss, 1971).

4.2.2 Automatic Evaluation

In addition to the human evaluation, we use a range of methods to automatically assess the alignment of the SSMs with GPT-3.5.

Similarity Between the Predicted Labels To compare the similarity between the predictions made by GPT-3.5 and by the SSMs, we report the accuracy and F1-Score for all predictions, Cohen's kappa (Cohen, 1960) as well as the precision and recall for each class.

Similarity Between the Explanations To quantitatively analyse how similar the explanations produced by GPT-3.5 and the SSM are, the cosine

similarity between the NLEs, the Jaccard similarity and the BERTScore (Zhang et al., 2019) are calculated. We report the mean similarity values of all three similarity measures between the NLEs produced by GPT-3.5 and the NLEs produced by the large and small SSM, respectively. To provide additional context for the similarity measures, the same similarity measures are also calculated between the gold standard NLEs and the NLEs produced by GPT-3.5.

5 Results

In this section, we report the experimental results using the evaluation procedures and measures introduced in Section 4.2, assessing how similar our surrogate models are to GPT-3.5. For illustrative purposes, we will provide example explanations generated by the SSMs in §5.1, followed by the results of the human (§5.2) and the automatic evaluations (§5.3).

5.1 Example Outputs of the SSMs

The NLEs produced by the SSM do indeed have a structure that follows the HGE. For the premise "A young woman sits crosslegged beside her purse on the grass among a crowd of dogs." and the hypothesis "The woman is on the couch with the dogs.", both SSMs generate "Grass is not the same as couch." The semantics of this example NLE capture the reason why there is a contradiction, namely that the location of the woman is different (and thereby contradictory) in the two statements.

Overly complex explanations are a challenge, particularly for the entailment relation. For the premise "A man tries to get himself into shape on a treadmill." and the hypothesis "A man exercising.", the small SSM's explanation is "Man is the same as a man and get is the same as exercising and if the location of sentence 1 is treadmill, then the verb of sentence 2 has to be exercising.", which is not only partially incorrect but also points to trivial information.

In Appendix A.4 we provide more examples for how the output of the SSM looked like.

5.2 Human Evaluation

The human evaluation revealed that the average Jaccard similarity between the subphrases used in the NLEs by GPT-3.5 and the SSM is 0.384 for the large SSM and 0.423 for the small SSM. The average Fleiss' Kappa for detecting the subphrases

used in the NLEs by GPT-3.5 is 0.640. In the remainder of this section, we report Fleiss’ Kappa for the three raters in parentheses following the average rating. The raters found that 84.2% (0.316) of explanations of GPT-3.5 follow a structure that, as stated in the HGE, relates different subphrases from premise and hypothesis to each other and that 96.8% (0.858) of NLEs support the predicted label. The raters marked 86.3% (0.287) of GPT-3.5’s NLEs, 27.4% (0.553) of the NLEs produced using the small, and 21.1% (0.522) of the NLEs produced by the large SSM as factually correct.

Overall, the results from the human evaluation indicate comparatively low correctness and low similarity of the SSM explanations, although a large part of the explanations follows a structure that would be possible to model with a Bayesian Network. The small SSM’s NLEs are more similar to GPT-3.5’s NLEs than the large SSM’s.

5.3 Automatic Evaluation

This section presents the results of the automatic evaluation, where different measures were calculated to assess the similarity between the outputs produced by GPT-3.5 and the SSM.

5.3.1 Similarity Between Predicted Labels

All analysed metrics shown in Table 1 reveal that the small SSM tends to predict the label of GPT-3.5 with a higher precision and recall (except for precision of class contradiction and recall of class neutral). The table also shows that the values are much lower than the values of the metrics between the gold standard label and the predictions made by GPT-3.5.

Metric	<i>Gold</i>	SSM _{large}	SSM _{small}
Contr. Precision	0.949	0.763	0.713
Contr. Recall	0.899	0.286	0.576
Entail. Precision	0.968	0.619	0.667
Entail. Recall	0.704	0.549	0.573
Neutr. Precision	0.485	0.242	0.299
Neutr. Recall	0.887	0.630	0.528
Accuracy	0.807	0.468	0.556
Avg. F1-Score	0.788	0.449	0.545
Cohen’s Kappa	0.709	0.219	0.339

Table 1: Metrics relating the SSMs’ to GPT-3.5’s predictions. The interval for Cohen’s Kappa is between -1 and 1 (with 0 being random and 1 perfect agreement) and for all other metrics from 0 to 1.

5.3.2 Similarity Between NLEs

The similarity scores between the NLEs produced by GPT-3.5 and the NLEs produced using the SSM can be seen in Table 2.

Model	Cosine S.	Jaccard S.	BERTScore
<i>Gold</i>	0.808	0.277	0.604
SSM _{large}	0.771	0.182	0.455
SSM _{small}	0.779	0.196	0.463

Table 2: Similarity Scores in relation to GPT-3.5 NLEs. The interval for all metrics is from 0 to 1.

It is clear that for all three metrics, the predictions made by GPT-3.5 are closer to the gold standard NLEs than to the NLEs of the SSM. As in the human evaluation, the self-rationalised NLEs by GPT-3.5 are closer to the NLEs by the small SSM compared to the NLEs by the large SSM.

6 Discussion

This paper set out with the aim of constructing and evaluating a hypothesis-driven surrogate for how GPT-3.5 performs NLI. As we have seen in section 3, the Bayesian network allows us to incorporate our hypothesis in an intuitive and testable way, making (in-)dependencies between phrases explicit. While the framework is convenient and functional, the results in section 5 indicate a low similarity to GPT-3.5. This can have two reasons: First, that the HGE does not resemble the way GPT-3.5 solves the NLI task, or second, that implementation details of the SSM have shortcomings that limit the performance of the model. In any case, several challenges need to be addressed before an SSM can potentially pass as a surrogate model. We discuss the shortcomings of our models as well as paths forward in §6.1 and 6.2, and promising paths for other future research based on our insights from this paper in 6.3.

6.1 Results and Similarity

Human raters found the factual correctness of the NLEs produced by the SSMs to be very low compared to GPT-3.5 (§5.2). The low alignment between the outputs of SSM and GPT-3.5 as observed in §5.2 and §5.3 suggests that the faithfulness of the HGE is relatively low.

As we kept the setup for this paper straightforward, many simplifying assumptions and tradeoffs have been made, limiting the expressiveness on several ends: The hypothesis is kept simple, the

coverage of the preprocessing and the template-based NLE generation are limited, and the models used are not optimised. We expect that work on these fronts can substantially improve the model, especially given that our human raters found that a large share of GPT-3.5’s explanations follows a structure that relates subphrases from premise and hypothesis to each other (§4.2.1), as reflected in the assumption behind the HGE. Another indication that development time investments may pay off is a recent work by [Stacey et al. \(2023\)](#) who propose an interpretable model based on logical rules similar to what is stated to the HGE defined in this paper. The high performance on the NLI task this model achieves suggests that an adapted HGE that reflects their model could have a considerably higher overlap with GPT-3.5 predictions than our current SSMs.

6.2 Uniqueness of the surrogate model

We found that there was no unique surrogate model for our HGE. The room for interpretation given the HGE formulated in natural language allows us to deduce a large number of structures, two of which we implemented. We showed that our different interpretations of the HGE (our small and large SSM) lead to different implementations of the surrogate models. However, to fully account for the ambiguity of the HGE when estimating the faithfulness, all viable SSMs would need to be constructed and compared to the LLM outputs.

The small model that incorporates more inductive biases performed better and produced NLEs more similar to GPT-3’s. This may indicate that the large model relied on incorrect cues introduced by less relevant connections. If this applies, it indicates that even the training regime of the SSM is of high importance in order for it to correctly represent the HGE. As we did not employ regularisation strategies in the feed-forward networks that we train in the Bayesian Network, this could be a path forward for potential improvements.

6.3 Future Work

The SSMs tested in this paper do not show sufficient similarity with the original model yet. Apart from altering the model or hypothesis as discussed in §6.1 and §6.2, we see other directions for future research building on this paper:

Estimating the uncertainty of the faithfulness

As previously mentioned, it is typically not possible

to deduce one unique SSM from an HGE formulated in natural language. Consequently, there is uncertainty in the faithfulness estimation that is gained by comparing the outputs of the LLM with one single SSM. Future research could investigate how this uncertainty might be estimated. For example, the ambiguity of the HGE would need to be taken into account when estimating this uncertainty. Research on investigating vagueness and ambiguity of written text samples is already an active field of research ([Freitas et al., 2015](#); [Wang and Agichtein, 2010](#); [Bernardy et al., 2018](#)). How these metrics can be used as a statistical measure of the uncertainty of an HGE remains an open question.

Automatically deriving surrogate models We conducted a case study for how an SSM can be constructed for an example HGE when performing English-language NLI. However, our manual design has the consequence that in many cases, different SSMs need to be constructed for new tasks, languages and HGEs.⁵ Therefore, in contrast to methods like LIME ([Ribeiro et al., 2016](#)) that can be applied directly to any classifier, defining a generalised procedure for how to automatically construct suitable surrogate models is challenging and requires further research.

7 Conclusion

This paper suggests a framework for testing a hypothesis about the decision-making process of LLMs using Bayesian networks. We demonstrate how to construct a Bayesian network based on a hypothetical global explanation and how to evaluate the alignment of the LLM with this network. The framework intuitively implements the hypothesis as the Bayesian Network that it is based on can model our assumptions natively, and its random variables can be translated into natural language.

Our surrogates were however not similar to GPT-3.5, which can have two reasons: That the hypothesis itself does not accurately describe GPT-3.5’s behaviour and needs to be replaced or complemented by other hypotheses, or that simplifications we made in our implementation affected the performance and thereby also the similarity to GPT-3.5. We assume that we are dealing with a combination of the two reasons, and suggest that working on the identified issues can lead to more accurate surrogate models that can help us understand the behaviour of LLMs better.

⁵For a detailed discussion, we refer to §8.

8 Limitations

In this section, we address further limitations of our work and outline potential paths to overcome them in future work.

Task and Data Natural language inference with the e-SNLI dataset is a task that naturally fits into the framework of a Bayesian Network, as most explanations are heavily built on input phrases and implicitly follow template-like structures (Camburu et al., 2018). Many tasks, particularly such that include facts and commonsense reasoning not explicitly stated in the input, are substantially harder to model with such intuitive and simple structures.

SNLI (Bowman et al., 2015), the base dataset for e-SNLI, has been shown to include various annotation artefacts that models can rely on (Gururangan et al., 2018). While our focus is on the evaluation of the explanations and not on the performance on the prediction task and the Bayesian Network by design cannot pick up the cues stated in the paper, it cannot be excluded that such artefacts have had an influence on the results of our automatic evaluations. For future work, it may be worthwhile to also consider alternatives, such as the more diverse MNLI dataset (Williams et al., 2018), and include explicit tests for lexical and syntactic cues, e.g. with the HANS dataset (McCoy et al., 2019).

Language English-language systems cannot always be trivially adapted to other languages, particularly where orthographic system, syntax and morphology differ substantially (Munro and Manning, 2010). As a result, our structure may be a worse fit for other languages with different features. In any case, our rule-based preprocessing would need to be adapted. While MNLI (Williams et al., 2018) has been translated into a relatively diverse set of languages (Conneau et al., 2018), there are currently no human-annotated NLEs for this data set. As we work with few-shot prompts for generating the LLM’s explanations, this may however be overcome with relatively little work.

Human Evaluation Our human evaluation is conducted by a small set of annotators with a machine learning-related background and similar demographics. This is a common practice in NLP research, but it introduces sampling bias (van der Lee et al., 2019). While we do not explicitly ask for personal preferences, this may still affect their judgements and thereby the results of our evalu-

ation. A larger set of more diverse annotators, if feasible, is preferable.

Reproducibility We use GPT-3.5, a closed-source model by OpenAI that we only have API access to. Unfortunately, this limits our experiments’ reproducibility, as OpenAI may remove or restrict access to it in the future. At the time of writing, the GPT-3.5 model produced better-quality output for our purposes than its more open competitors. However, in recent months, an increasing number of high-quality LLMs is released to the public, such as various LLaMA (Touvron et al., 2023)-based models such as Alpaca (Taori et al., 2023) or the Pynthia (Biderman et al., 2023)-based Dolly model (Conover et al., 2023), paving the way for more reproducible LLM research.

Ethics Statement

As LLMs are trained on human data that can be biased, toxic, and immoral, they frequently produce unethical outputs (Liu et al., 2022; Abid et al., 2021; Bender et al., 2021). However, we use LLMs solely as an object of examination. This study aims at increasing the transparency and accountability of GPT-3.5, which can be a step in the direction of preventing LLMs from producing unethical outputs. That said, explainability techniques for models as large as current LLMs are only an approximation. We do not endorse any usage of LLMs for high-stake applications without humans in the loop, even as explainability research progresses.

Acknowledgements

We thank the anonymous reviewers for their constructive feedback that helped to refine this work.

References

- Abubakar Abid, Maheen Farooqi, and James Zou. 2021. [Persistent anti-muslim bias in large language models](#). In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, AIES ’21, page 298–306, New York, NY, USA. Association for Computing Machinery.
- Amina Adadi and Mohammed Berrada. 2018. [Peeking inside the black-box: A survey on explainable artificial intelligence \(xai\)](#). *IEEE Access*, 6:52138–52160.
- Shourya Aggarwal, Divyanshu Mandowara, Vishwa-jeet Agrawal, Dinesh Khandelwal, Parag Singla, and Dinesh Garg. 2021. [Explanations for CommonsenseQA: New Dataset and Models](#). In *Proceedings of the 59th Annual Meeting of the Association for*

- Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3050–3065, Online. Association for Computational Linguistics.
- Pepa Atanasova, Oana-Maria Camburu, Christina Lioma, Thomas Lukasiewicz, Jakob Grue Simonsen, and Isabelle Augenstein. 2023. [Faithfulness tests for natural language explanations](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 283–294, Toronto, Canada. Association for Computational Linguistics.
- Yonatan Belinkov and James Glass. 2019. [Analysis methods in neural language processing: A survey](#). *Transactions of the Association for Computational Linguistics*, 7:49–72.
- Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. [On the dangers of stochastic parrots: Can language models be too big?](#) In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.
- Jean-Philippe Bernardy, Rasmus Blanck, Stergios Chatzikyriakidis, and Shalom Lappin. 2018. [A compositional Bayesian semantics for natural language](#). In *Proceedings of the First International Workshop on Language Cognition and Computational Models*, pages 1–10, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O’Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. [Pythia: A suite for analyzing large language models across training and scaling](#). In *International Conference on Machine Learning*, pages 2397–2430. PMLR.
- Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. [A large annotated corpus for learning natural language inference](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#).
- Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. [e-snli: Natural language inference with natural language explanations](#). *CoRR*, abs/1812.01193.
- J. Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37.
- Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. [XNLI: Evaluating cross-lingual sentence representations](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.
- Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. [Free dolly: Introducing the world’s first truly open instruction-tuned llm](#).
- Pieter-Tjerk de Boer, Dirk P. Kroese, Shie Mannor, and Reuven Y. Rubinstein. 2005. [A tutorial on the cross-entropy method](#). *Annals of Operations Research*, 134(1):19–67.
- A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. [Maximum likelihood from incomplete data via the em algorithm](#). *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.
- Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.
- James Forrest, Somayajulu Sripada, Wei Pang, and George Coghill. 2018. [Towards making NLG a voice for interpretable machine learning](#). In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 177–182, Tilburg University, The Netherlands. Association for Computational Linguistics.
- André Freitas, Juliano Efon Sales, Siegfried Handschuh, and Edward Curry. 2015. [How hard is this query? measuring the semantic complexity of schema-agnostic queries](#). In *Proceedings of the 11th International Conference on Computational Semantics*, pages 294–304, London, UK. Association for Computational Linguistics.
- Yoav Goldberg. 2023. [Two kinds of recall](#). *arXiv preprint arXiv:2303.10527*.
- Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. [Annotation artifacts in natural language inference data](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

- Peter Hase, Shiyue Zhang, Harry Xie, and Mohit Bansal. 2020. [Leakage-adjusted simulatability: Can models generate non-trivial explanations of their behavior in natural language?](#) In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4351–4367, Online. Association for Computational Linguistics.
- Matthew Honnibal and Mark Johnson. 2015. [An improved non-monotonic transition system for dependency parsing.](#) In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1373–1378, Lisbon, Portugal. Association for Computational Linguistics.
- Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.
- Alon Jacovi and Yoav Goldberg. 2020. [Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness?](#) In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205, Online. Association for Computational Linguistics.
- Ruibo Liu, Ge Zhang, Xinyu Feng, and Soroush Vosoughi. 2022. [Aligning generative language models with human values.](#) In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 241–252, Seattle, United States. Association for Computational Linguistics.
- Scott M Lundberg and Su-In Lee. 2017. [A unified approach to interpreting model predictions.](#) In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Ana Marasovic, Iz Beltagy, Doug Downey, and Matthew E. Peters. 2021. [Few-shot self-rationalization with natural language prompts.](#) *CoRR*, abs/2111.08284.
- Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. [Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference.](#) In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.
- Robert Munro and Christopher D. Manning. 2010. [Subword variation in text message classification.](#) In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 510–518, Los Angeles, California. Association for Computational Linguistics.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. [Training language models to follow instructions with human feedback.](#) In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.
- Judea Pearl. 1988. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Representation and Reasoning Series. Morgan Kaufmann.
- Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. [Explain yourself! leveraging language models for commonsense reasoning.](#) In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942, Florence, Italy. Association for Computational Linguistics.
- Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. ["why should I trust you?": Explaining the predictions of any classifier.](#) *CoRR*, abs/1602.04938.
- Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215.
- Donghee Shin. 2021. [The effects of explainability and causability on perception, trust, and acceptance: Implications for explainable ai.](#) *International Journal of Human-Computer Studies*, 146:102551.
- Joe Stacey, Pasquale Minervini, Haim Dubossarsky, Oana-Maria Camburu, and Marek Rei. 2023. [Logical reasoning for natural language inference using generated facts as atoms.](#)
- Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- Miles Turpin, Julian Michael, Ethan Perez, and Sam Bowman. 2023. [Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting.](#) *ArXiv*, abs/2305.04388.
- Chris van der Lee, Albert Gatt, Emiel van Miltenburg, Sander Wubben, and Emiel Kraemer. 2019. [Best practices for the human evaluation of automatically generated text.](#) In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 355–368, Tokyo, Japan. Association for Computational Linguistics.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need.](#) In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Cunxiang Wang, Shuailong Liang, Yili Jin, Yilong Wang, Xiaodan Zhu, and Yue Zhang. 2020. [SemEval-2020 task 4: Commonsense validation and explanation](#). In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 307–321, Barcelona (online). International Committee for Computational Linguistics.

Yu Wang and Eugene Agichtein. 2010. [Query ambiguity revisited: Clickthrough measures for distinguishing informational and ambiguous queries](#). In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 361–364, Los Angeles, California. Association for Computational Linguistics.

Sarah Wiegrefe, Jack Hessel, Swabha Swayamdipta, Mark Riedl, and Yejin Choi. 2022. [Reframing human-AI collaboration for generating free-text explanations](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 632–658, Seattle, United States. Association for Computational Linguistics.

Sarah Wiegrefe and Ana Marasovic. 2021. [Teach me to explain: A review of datasets for explainable natural language processing](#). In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1. Curran.

Sarah Wiegrefe, Ana Marasović, and Noah A. Smith. 2021. [Measuring association between labels and free-text rationales](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10266–10284, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. [A broad-coverage challenge corpus for sentence understanding through inference](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. [Bertscore: Evaluating text generation with BERT](#). *CoRR*, abs/1904.09675.

A Appendix

A.1 EM-Algorithm

E-Step

Given the observed variables X, Y , the hidden variable Z , and the parameters θ , the E-step of the EM-algorithm can be expressed in the following way:

$$\begin{aligned}
 Q(\theta|\theta^{(t)}) &= E_{Z|X,Y,\theta^{(t)}} [\log \mathcal{L}(X, Y, Z|\theta)] \\
 &= E_{Z|X,Y,\theta^{(t)}} \left[\sum_{i=1}^n \log P(x_{(i)}, y_{(i)}, Z|\theta) \right] \\
 &= \sum_{i=1}^n E_{Z|X,Y,\theta^{(t)}} [\log P(x_{(i)}, y_{(i)}, Z|\theta)] \quad (3) \\
 &= \sum_{i=1}^n \sum_Z P(Z|x_{(i)}, y_{(i)}, \theta^{(t)}) \\
 &\quad \cdot \log P(x_{(i)}, y_{(i)}, Z|\theta)
 \end{aligned}$$

In Equation 3, $x_{(i)}, y_{(i)}$ is the i -th observed realisation of the random vector X and Y respectively and n is the total number of observations. The summation over Z is performed over all possible states of the discrete random vector Z . As all valid probability distributions must sum up to one over their domain, we know that $\sum_Z P(Z|x_{(i)}, y_{(i)}, \theta^{(t)}) = 1$. This makes the expression $\sum_Z P(Z|x_{(i)}, y_{(i)}, \theta^{(t)}) \cdot \log P(x_{(i)}, y_{(i)}, Z|\theta)$ a weighted average for a given i where the weight is $P(Z|x_{(i)}, y_{(i)}, \theta^{(t)})$.

This means that we can approximate this weighted average by calculating the mean of $\log P(x_{(i)}, y_{(i)}, \tilde{Z}|\theta)$ where \tilde{Z} are Z samples from the distribution given by the weights (i.e. $P(Z|x_{(i)}, y_{(i)}, \theta^{(t)})$). Consequently we can approximate $Q(\theta|\theta^{(t)})$ as

$$Q(\theta|\theta^{(t)}) \approx \sum_{i=1}^n \frac{1}{s} \sum_{\tilde{Z}_{(i)}} \log P(x_{(i)}, y_{(i)}, \tilde{Z}_{(i)}|\theta) \quad (4)$$

where $\tilde{Z}_{(i)}$ are s sampled values from the distribution of $Z|x_{(i)}, y_{(i)}, \theta^{(t)}$. This distribution can be expressed as

$$\begin{aligned}
 P(Z|x_{(i)}, y_{(i)}, \theta^{(t)}) &= \frac{P(Z, y_{(i)}|x_{(i)}, \theta^{(t)})}{P(y_{(i)}|x_{(i)}, \theta^{(t)})} \\
 &\propto P(Z, y_{(i)}|x_{(i)}, \theta^{(t)}) \\
 &= P(y_{(i)}|Z, x_{(i)}, \theta^{(t)}) \\
 &\quad \cdot P(Z|x_{(i)}, \theta^{(t)}) \\
 &\text{Independence of } Y \text{ and } X, \theta \text{ given } Z \\
 &= P(y_{(i)}|Z) \cdot P(Z|x_{(i)}, \theta^{(t)}) \quad (5)
 \end{aligned}$$

The distribution of $Y|Z$ was defined in Equation 2. The distribution of $Z|X, \theta^{(t)}$ is known assuming the current parameter estimate $\theta^{(t)}$ is the true parameter of the distribution. Furthermore, we know that $P(Y|Z)$ is either of value 0 or 1. We can therefore produce a sample \tilde{z} from $Z|X, Y, \theta^{(t)}$ by sampling from $Z|X, \theta^{(t)}$ and rejecting the sample

if $P(Y|\tilde{z}) = 0$. If the sample is rejected, we repeat the sampling process until a sample is accepted.

Let s be the number of samples we generated for each i from $Z|x_{(i)}, y_{(i)}, \theta^{(t)}$, n the number of observations in the dataset of x and y , and let \tilde{Z} be the collection of all $n \cdot s$ samples. Let \tilde{z}_j be the j -th sample in \tilde{Z} . We then produce s duplicates of each $x_{(i)}, y_{(i)}$ and define $x_{(j)}, y_{(j)}$ to be those datapoints x, y that were used to produce the sample \tilde{z}_j . Based on that definition, we can write Equation 4 as

$$\begin{aligned}
Q(\theta|\theta^{(t)}) &\approx \frac{1}{s} \sum_{j=1}^{n \cdot s} \log P(x_{(j)}, y_{(j)}, \tilde{z}_{(j)}|\theta) \\
&= \frac{1}{s} \sum_{j=1}^{n \cdot s} \log [P(y_{(j)}, \tilde{z}_{(j)}|x_{(j)}, \theta) \cdot P(x_{(j)}|\theta)] \\
&\text{Independence of } X \text{ and } \theta \\
&= \frac{1}{s} \sum_{j=1}^{n \cdot s} \log [P(y_{(j)}, \tilde{z}_{(j)}|x_{(j)}, \theta) \cdot P(x_{(j)})] \\
&= \frac{1}{s} \sum_{j=1}^{n \cdot s} \log [P(y_{(j)}|\tilde{z}_{(j)}, x_{(j)}, \theta) \\
&\quad \cdot P(\tilde{z}_{(j)}|x_{(j)}, \theta) \cdot P(x_{(j)})] \\
&\text{Independence of } Y \text{ and } X, \theta \text{ given } Z \\
&= \frac{1}{s} \sum_{j=1}^{n \cdot s} \log [P(y_{(j)}|\tilde{z}_{(j)}) \\
&\quad \cdot P(\tilde{z}_{(j)}|x_{(j)}, \theta) \cdot P(x_{(j)})] \\
&P(y_{(j)}|\tilde{z}_{(j)}) = 1 \text{ for the sampled } \tilde{Z}, \text{ because} \\
&\text{otherwise the sample got rejected} \\
&= \frac{1}{s} \sum_{j=1}^{n \cdot s} \log [P(\tilde{z}_{(j)}|x_{(j)}, \theta) \cdot P(x_{(j)})]
\end{aligned} \tag{6}$$

M-Step

In the M-step, the estimation for θ is updated by setting $\theta^{(t+1)}$ to the value that maximises $Q(\theta|\theta^{(t)})$.

$$\begin{aligned}
\theta^{(t+1)} &= \arg \max_{\theta} Q(\theta|\theta^{(t)}) \\
&\text{With Equation 6} \\
&\approx \arg \max_{\theta} \frac{1}{s} \sum_{j=1}^{n \cdot s} \log [P(\tilde{z}_{(j)}|x_{(j)}, \theta) \cdot P(x_{(j)})] \\
&= \arg \max_{\theta} \sum_{j=1}^{n \cdot s} \log \left[\frac{P(\tilde{z}_{(j)}, x_{(j)}|\theta) \cdot P(x_{(j)})}{P(x_{(j)}|\theta)} \right] \\
&\text{Independence of } X \text{ and } \theta \\
&= \arg \max_{\theta} \sum_{j=1}^{n \cdot s} \log [P(\tilde{z}_{(j)}, x_{(j)}|\theta)] \\
&= \arg \max_{\theta} \log \mathcal{L}(\theta|\tilde{Z}, X)
\end{aligned} \tag{7}$$

From Equation 7 we can see that the M-step amounts to maximising the log-likelihood of the

parameter θ with respect to the sampled values \tilde{Z} and the given realisations of the random vector X . This in return means that, in the E-step, it suffices to produce the samples \tilde{Z} that are needed in the M-step.

Since all elements $z_{k,l} \in \mathcal{Z}$ are independent of each other given X , the parameters θ can be separated between each distribution $z_{k,l}|X$ and are called $\theta_{k,l}$. With Equation 7 it follows that

$$\theta_{k,l}^{(t+1)} = \arg \max_{\theta} \log \mathcal{L}(\theta|\tilde{z}_{k,l}, X) \tag{8}$$

A.2 SSM Explanation Templates

Conditions	Template
$Y = \text{contradiction}$ $z_{k,l} = \text{contradiction}$ $k = l$	" $x_{1,k}$ is not the same as $x_{2,l}$ "
$Y = \text{contradiction}$ $z_{k,l} = \text{contradiction}$ $k \neq l$	"If the subphrase _{k} of sentence 1 is $x_{1,k}$, then the subphrase _{l} of sentence 2 cannot be $x_{2,l}$ "
$Y = \text{entailment}$ $z_{k,l} = \text{entailment}$ $k = l$	" $x_{1,k}$ is the same as $x_{2,l}$ "
$Y = \text{entailment}$ $z_{k,l} = \text{entailment}$ $k \neq l$	"If the subphrase _{k} of sentence 1 is $x_{1,k}$, then the subphrase _{l} of sentence 2 has to be $x_{2,l}$ "
$Y = \text{neutral}$ $z_{k,l} \neq \text{entailment } \forall k$	"There is no indication that the subphrase _{l} of sentence 2 is $x_{2,l}$ "

Table 3: Templates for the explanations of the SSM predictions. subphrase _{i} is "Subject" for $i = 1$, "Verb" for $i = 2$, "Object" for $i = 3$, "Location" for $i = 4$, and "Clothing" for $i = 5$.

e

A.3 Instructions for Evaluating free-text Explanations

Background

1. Evaluation of GPT-3.5 Explanations

Please answer the following questions for each of the provided explanations produced by GPT-3.5. If not stated otherwise, mark the respective answer column in the evaluation template with 1 if the question is answered with yes.

- Subphrases:** If any, which subphrases are mentioned in the explanation? Possible subphrases are subject, verb, object, clothing, and location either of sentence one or sentence two. Mark the answer column for the respective subphrase with 1 if the subphrase is mentioned in the explanation.

2. **Structure:** Does the explanation describe the relationship between subphrases? This question should also be answered with yes if the explanation makes a statement about the relationship between subphrases that are not mentioned in question 1.
3. **Support:** Does the explanation justify the predicted label? i.e. given that the explanation is true, is the predicted label correct?
4. **Correctness:** Is the explanation factually correct? i.e. given your experience about our world and given the statements in sentence one and sentence two, is the explanation true?

2. Evaluation of SSM Explanations

The are explanations and predictions provided for two different versions of the statistical surrogate model (SSM). The first SSM's predicted labels and explanations are marked with subscript "large" and the predicted labels and explanations produced by the second SSM are marked with subscript "small". Please answer the following question for each of the provided explanations produced by the large and by the small SSM.

1. **Overall Correctness:** Is the explanation factually correct given the premise and hypothesis? i.e. given your experience about our world and given the statements in sentence one and sentence two, is the explanation true?
2. **Subphrase Correctness:** Is the explanation factually correct given the subphrases? i.e. given your experience about our world and given the subphrases extracted from sentence one and sentence two, is the explanation true?

A.4 Examples for SSM Output

1. **Premise:** A young woman sits crosslegged beside her purse on the grass among a crowd of dogs.
Hypothesis: The woman is on the couch with the dogs.
NLE Small SSM: Grass is not the same as couch.
NLE Large SSM: Grass is not the same as couch.
2. **Premise:** Two men are shopping for orange juice.
Hypothesis: Two men are getting breakfast
NLE Small SSM: There is no indication that

the verb of sentence 2 is getting.

NLE Large SSM: There is no indication that the object of sentence 2 is breakfast.

3. **Premise:** A man tries to get himself into shape on a treadmill.

Hypothesis: A man exercising.

NLE Small SSM: Man is the same as a man and get is the same as exercising and if the location of sentence 1 is treadmill, then the verb of sentence 2 has to be exercising.

NLE Large SSM: There is no indication that the subject of sentence 2 is a man and there is no indication that the verb of sentence 2 is exercising.