Fishing for biases:
An experimental and theoretical inquiry with Semantic Match

MSc Thesis *(Afstudeerscriptie)*

written by

Ludovico Deponte

under the supervision of **Dr. Giovanni Cinà**, and submitted to the Examinations Board in partial fulfillment of the requirements for the degree of

MSc in Logic

at the *Universiteit van Amsterdam*.

Date of the public defense:	**Members of the Thesis Committee:**
June 25, 2024	Dr. Balder ten Cate (Chair)
	Dr. Giovanni Cinà (Supervisor)
	Dr. Tim van Erven
	Jaap Jumelet
	Dr. Sandro Pezzelle

Institute for Logic, Language and Computation

Abstract

In recent years, breakthroughs in model architecture and training, large availability of data, and increased computing power, conjunctively allowed AI models not only to improve performance on previous research tasks, but also to be useful in everyday life: from translation to image and text generation, the new models are now used in the daily workflow of million of users. With this widespread adoption, it is paramount to understand the model's workings and behaviour.

While in the eXplainable Artificial Intelligence field there are already tools to tackle this problem, some of the most often used ones rely on local explanations, evaluating the dependence of the model's outputs on one or more features of the input, data point by data point. Similar methods are especially problematic whenever explanations of single features are grouped together and used to make analogies with human concepts, in order to infer the general behaviour of the model. In turn, similar generalizations can lead to erroneous conclusions about model's behaviour and to confirmation bias.

The semantic match framework attempts to address the issue of generalization of local explanation methods by constructing global hypotheses on model's behaviour and verifying that the evidence provided by local explanations is consistent with the considered hypothesis. If that is the case, the hypothesis matches the behaviour of the model, and it can be used for an account of the model workings.

In this thesis, on one hand, we conduct an experiment to verify if it is possible to use semantic match to discover a known bias of a model; on the other, we assess the metrics of the framework, and compare them to alternatives inspired by previous work.
More specifically, we start by training a biased and unbiased model on the SQuAD dataset, and check different hypotheses on their functioning against their actual behaviour. We measure the match between a hypothesis and model behaviour with two metrics, median distance and area under the curve.
After reviewing the experiment results, we examine general trends of the used metrics, propose further evaluations based on them, and also apply alternative metrics. We close by discussing the advantages and disadvantages of each measure.

Acknowledgements

This work would not exist without the help and inspiration of many. Firstly, I would like to thank Giovanni, my supervisor; we met somewhat randomly after a conference at UvA, as I was questioning participants on their studies and careers. At the time, I was thorn between very different interests, and I was on the hunt for any clue that could aid me choosing a path, if not for my career, at least for my Master studies. After a chat he suggested me reading few papers and so we started a project which, as months passed and work progressed, became this very thesis.
During that time, I not only had the possibility to collaborate with other researchers on the semantic match framework, exchange opinions, and get precious feedback, which coalesced here; but, more importantly, I saw how active research is conducted, how ideas can grow, change, and interlock with apparently contradictory suppositions, and how the gamut of insights, beliefs, objectives, and instincts of different minds can converge in a coherent and structured framework.
For this, and for the immense patience, the inspiring comments and the continuous *labor limae* on the contents and the form of what follows, I am extremely grateful.

I would also like to thank each member of the committee, both for taking the time to go through this thesis, and for asking foundational questions regarding the many choices that one takes while using semantic match. As sometimes happens when working with tools, one glosses over the assumptions on which their functioning relies, just trusting them to work properly. Answering the committee's questions allowed me to see the whole project from a different perspective, to think about some of the most problematic postulates, and overall deepened my understanding of the framework.

I am also grateful for all the knowledgeable and kind people I met at the MoL, for all the interesting discussions and insights on logic, philosophy, AI, physics, life, and everything in between.

I thank all my friends, who supported me while I was far and busy, who called, visited and welcomed me back when I returned, and who I know will always be there for me. For all the brave and the mad adventures, visions and dreams, and for all the small glimpses of truths fished at sunrise.

To my family, thank you for always encouraging me, for supporting me through every choice, for teaching me to love and to care, and for continuously feeding my curiosity with Art, Literature and Science. Finally, I am grateful to Elettra, for coming up with great plans, awesome music, wild ideas and an untamed critical thinking. For sharing the good and the bad times, for bringing spring in Amsterdam every time, and for walking through beside me.

May each of us, with promptness and *metriotes*, catch their own Kairos.

Contents

Introduction

In recent years, three key factors aligned to greatly boost the progress and success of Machine Learning and Deep Learning: firstly, the availability of captioned images and videos, audio feeds and human conversations on widely used social media platforms allowed to build large datasets for a wide variety of tasks[1] (*Big Data*); secondly, technological advancement in chip manufacturing increased compute power and efficiency of modern computers, boosting them to boundaries unimaginable just few decades ago (*Big Compute*). Finally, the gradual development of new AI architectures based on more complex modules (such as convolutions and transformers), and the almost exponential increases in the number of parameters of AI models led to relevant improvements in their performance and capabilities (*Big Models*).[2]
Together, these three factors brought a new spring in the whole AI field, stimulating further research. The development of generative models and the great potential of their applications catalysed the attention of the public and of multiple investors, generating a new multi-billion industry in just few months [3], and resulting in a blossoming of tools to help with the most disparate daily chores. Previous discussions over applicability and usefulness of ML and DL approaches,[3] now closed by the evident practical advantages of using such models, shifted focus, centring on the problem of improving model's performance and tackling more and more tasks. While those developments were happening, Big Data and Big Compute were the magical answers to any issue, and the motto 'give me a big enough dataset and a powerful enough computer and I shall lift the world' (to echo the old Archimedean adage) well summarizes the feverish excitement, still common across AI enthusiasts, about the prosperous promises of the new blooming technologies.

Nowadays, AI models are used in low-stake environments, to enhance image capturing and processing, make better recommendations and improve user experience,[4] as well as in high-stake tasks to aid decision-makers in healthcare [48], military [32], and finance [61].[5] With all these advances, authors began to speak about big leaps in model accuracy and an exponential growth in investments, in what can be properly called a new Golden Age of AI [26].

Yet, in this modern land of milk and honey, few perils still persist, and major questions are left unanswered: crucial for the high-stake environment and important in the low-stake applications as well, are concerns over the risks and safety of AI tools, together with inquiries on privacy and transparency of data and model use. These problems contributed to foster the development of new techniques to assess and control the model's behaviour, leading to the growth of eXplainable Artificial Intelligence (XAI).

One of the main task of XAI is to build tools to understand the inner workings of *black-box* models, usually with the objective of verifying that the model is following a specific behaviour, is robust and reliable, or has no

[1] We use *task* as a technical term, close to the definition provided in [51], by which we mean a problem with an initial state, some desired (goal) and undesired (failure) states and an agent (model), which by interacting with the initial state can reach one or more goal or failure states.

[2] For transformers, see [54]. Convolutions were introduced by [12], but raised to prominence only later with [29]. For the increase in the number of parameters, see for instance Figure 1 of [55].

[3] Firstly prompted by the Perceptron [44], and partially contained by [37], the development of the AI field has seen cyclical spikes in attention and expectations.

[4] Think of your smartphone: it includes a smart assistant, AI models for taking pictures, and one or more services to suggest and recommend products, filter advertisements and rank contents; chatbots recently came to prominence thanks to the improvements in their performance, and are now integrated in the workflow of many. To mention only one example, the active user base of chatGPT grew to around 100 million in just two months after its release [24].

[5] Transformers have been recently applied successfully even to predict life events [46].

bias.
Lately, *feature attribution methods*, such as SHAP values, [35], emerged as a popular approach to solve this problem. The idea is to see how small changes in the inputs determine changes in the outputs, and to represent the relevance of each feature of the input for the model's prediction with a score. Features with highly positive or negative scores will be more important for the model, and influence in a more relevant way its output, thus giving insights into what the model considers more significant, see Section 1.2. Such methods are very intuitive, as they associate to each feature of the input a positive (negative) value if the feature increased (decreased) the probability of that prediction. With SHAP values, troubleshooting prediction errors is facilitated, as they enable users to look into the features considered important for the prediction, and highlight if the model is attending too much to the wrong information.

These methods, however, suffer from a major problem: when they are used to ascertain that the model is following a specific and wanted heuristic, they can very easily lead to *confirmation bias*. This happens when, while analysing model's behaviour, we fail to collect and properly analyse counterexamples, while assigning high importance to any evidence aligned with our pre-exiting expectations. As Cinà et al. [10] argue, this error is less common on tabular datasets, since each low-level feature is by default endowed with a meaning.[6] In images or text, where *low-level* features (as pixels or syllables) are fed to the model, but only *high-level* features (as objects or sentences) are meaningful for the user, confirmation bias is more dangerous, and common, as experimentally shown by Wan, Belo, and Zejnilović [56] and Bauer, Von Zahn, and Hinz [8].
The semantic match framework, proposed in [10] and [11], was developed to contain exactly that problem: instead of manually inspecting single instances of input-output pairs, together with their explanations, in this framework one can formulate a hypothesis on how the model is generally behaving, and then measure how well the hypothesis matches the explanations. Checking one hypothesis after the other, it becomes possible to verify if the model is following specific patterns, in an exchange that resembles a series of yes or no questions ('Are you behaving in *this* way?', 'Are you behaving in *that other* way?', ...).
With this machinery, we can go *fishing for biases*, progressively testing, and hopefully discarding, the possible prejudices that a model might suffer from. The goal of the thesis is to probe the fishing pole: we will purposefully train two model so that the first should be biased, while the second should not. Then we will formulate different hypotheses, to see if we can catch the bias of the first model, and to verify that the second model is not suffering from it.
While the goal of semantic match is to be applied in scenarios where no a priori knowledge of the dataset or model is assumed, as the framework is still under development, the experiment here presented is a needed step towards that objective. Continuing the analogy, we will conduct our test in a small lake, where we already put a big tuna, before our fishing trip in the oceans. Nonetheless, as we are using BERT, a state of the art black-box model, there might be unforeseen difficulties: we might incur into bigger or smaller fishes, and we cannot be sure that our tuna is still there when we will throw our hook in the lake.

Our main research question is therefore: is it possible to spot and describe a specific bias that a model is suffering from, using hypotheses of the semantic match framework? And does that machinery allow to also check if another model is not suffering from that same bias?
Collaterally, we will also probe the powers of semantic match, by using its metrics with different goals in mind, and by showing how to interpret their results.

Structure This work is divided in three: the introductory part, composed by Chapter 1 and Section 2.1 provides the reader with the general background and concepts needed to understand the goals and workings of the semantic match framework.
The experimental part starts in Section 2.2, where we describe the setup of the experiment, showing some preliminary results which confirm our expectations; in Chapter 3 we define the hypotheses of model's behaviour

[6] Note that high-level features can also be defined for tabular datasets, and might lead to confirmation biases.

that we will test, and present the results, discussing each of their implications on the way in which our models predict. Here we will also introduce additional metrics from Zhou, Ribeiro, and Shah [59], and comment the obtained results.
Finally, some more theoretical aspects are discussed in Chapter 4, where we will describe and interpret the typical trends of the semantic match metrics, we will re-evaluate them showing how changing the sets on which they depend can provide further evidence to support the results, and compare our metrics to those of Zhou, Ribeiro, and Shah [59].
In the Conclusion we are going to sum up the results and contributions of this work, briefly exploring some future directions of research on semantic match.

Related work While in recent years a plethora of studies highlighted limitations of feature attribution methods, to our knowledge, not a lot of work has been focused on the specific dangers of confirmation bias that they might rise. Zhou, Ribeiro, and Shah [59] introduce a general framework to explain model's behaviours with human-readable hypotheses, but their approach is limited in two ways: firstly, their tools were developed for text modality only, and are not straightforward to apply to other types of datasets; secondly, their hypotheses are concerned only with low-level features, which, according to Cinà et al. [10], are less prone to cause confirmation bias, as they are usually equipped with an interpretable semantic (in [59], features are single words, while in [10] the columns of a tabular dataset are used as an example). While the authors of Zhou, Ribeiro, and Shah [59] introduce compositional rules to capture some high-level features emerging from input's features, those are unable to capture other, very natural features, such as sentences of a text, or sections of an image containing a specific object.

Cinà et al. [10] and [11] propose a similar framework, applicable to any modality and agnostic to the choice of model and feature attribution method. As we shall see in Chapter 2, semantic match in principle allows the definition of features with an arbitrary level of abstraction, enabling to test the consistency of model's behaviours on any group of data points and on any feature of interest.

Zooming out from our specific research questions, this work is linked with major trends in XAI by multiple threads: the first is the search and documentation of dataset's and model's biases, which has sprung a lot of literature; for the design and setting of our experiment, we relied on the results shown by Ko et al. [27], which proved how it is possible to bias a BERT-based model by training it on a purposefully manipulated dataset. Other work on biases in visual datasets has been conducted, among others, by Tommasi et al. [52] and Fabbrizzi et al. [16]. Garrido-Muñoz et al. [17] survey biases in NLP, while frameworks to individuate biases in textual datasets were introduced in Raza, Reji, and Ding [41] and Raza et al. [42]. Gender biases are very frequent in such datasets and more work on this matter is referenced in Doughman and Khreich [15].

Development of feature attribution methods and inquiries in their limitations are also blooming in the late years, and linked to our research. While the current work focuses on SHAP values, introduced in Lundberg and Lee [35], the same experiment could be reproduced for any other feature attribution method. For a detailed overview of the available methods in computer vision we refer to Abhishek and Kamath [1], while an overview of tools for healthcare can be found in Singh, Sengupta, and Lakshminarayanan [50]; Ghassemi, Oakden-Rayner, and Beam [19] is a survey on some problems of current explanatory methods.
SHAP values rose to prominence in the last years due to their innate intuitiveness and some of their theoretical properties, together with a ready-to-use implementation in a well documented package, see Section 1.2; nonetheless, they were criticized for multiple reasons, one being their complexity: based on an iterative evaluation over all possible combinations of input features, the complexity of their evaluation can be exponential in the number of features, and thus it scales badly for datasets based on images, texts, videos and the like: SHAP values have also been proven to be #P-complete for simple AI models such as logistic regressions [9].[7] While in some cases exact computations can be conducted (as proved in [5, 6, 7]), they are often only estimated. Huang and Marques-Silva

[7] This complexity class is composed by counting problems associated with decision problems of NP; in simple terms, problems in #P are at least as difficult as those in NP, [53].

[25] conducted an experiment to investigate how good such estimates are, by comparing the exact values and the estimations, finding not only that the latter do not match the former, but also that the relative order of relevance of the features can differ.

Finally, the study of human biases in the interpretation of model's predictions and explanations has risen to prominence in the last years. Insights from philosophy and psychology inspired a theoretical framework by Wang et al. [57], made to improve explanations techniques and avoid common cognitive biases. The interaction between human decision-makers, the predictions of a model and the SHAP values related to it has also proven problematic, and Wan, Belo, and Zejnilović [56] empirically showed that, when humans are given both the prediction of a model and the corresponding SHAP values, users will increase their confidence in the choices they make, whenever the explanations align with their own motivations for taking those decisions. Other experiments by Bauer, Von Zahn, and Hinz [8] showed that feature attribution methods can influence and modify both the user's processing of the information received to make the decision, and their underlying mental model, with risks of manipulation and spillover to similar decision tasks, even for expert decision-makers. Importantly, they find asymmetries in the way the mental models are adjusted, leading to confirmation bias: if the explanations corroborate user's beliefs, those ideas are strongly reinforced, while if the explanations are not in line with the mental model, the latter is only slightly modified.

Contribution This work contributes to the development of the semantic match framework by applying it in a controlled experiment. It is the first experiment employing that framework in the textual modality with the goal of capturing the behaviour of two transformer-based models, proving that one is biased while the other is not.

The framework was firstly proposed in [10], and fully formalized as we present it in Section 2.1 in [11]. Previous experiments on images are also included in Cinà et al. [11], and the way in which the context was split into sentences in our experiment is reminiscent of the bounding boxes used for image experiments. The design of the experiment here discussed was a collaboration between the author and the collaborators in Cinà et al. [11]. The MD and AUC results presented in this work have been obtained by the author and are featured in [11]; the discussion of their meaning is influenced by [11]. The comparison with the metrics of Zhou, Ribeiro, and Shah [59] is original and unpublished at the time of writing.

In Chapter 3 we apply the metrics of [59] to our model, showing that the rules introduced in that paper are translatable into the semantic match framework, while the converse is not always possible.

For training, we followed the strategy of [27]: while that work focuses on how to mitigate dataset's biases with specific techniques of dataset manipulation, in our work the training was used to generate the two models whose behaviour is studied here.

Chapter 1

Background

The goal of this introductory chapter is to equip the reader with the basics needed to understand the rest of the work. We are firstly going to define very generally what a *model* is, and then we will delve deeper into the inner workings of BERT, which we later use for the experiment.
We are also going to define *feature attribution methods*, focusing on the intuitions and theory behind SHAP values. Finally, we will discuss how biases can arise when using such tools, presenting *semantic match* as a possible solution.
Notational conventions followed in the work close the chapter.

1.1 What is a model?

The first concept we need to get acquainted with is that of *artificial intelligence model*. That is typically an algorithm, implemented as a software program, which can learn how to match patterns and correlation in a given dataset.
With *patterns* and *correlations* we informally mean any relation or connection that incurs between parts of the input given to the model, while a *dataset*, $\mathcal{D}$ is defined as a set of *data points*, each being an input-label pair $\langle \boldsymbol{x}, y \rangle$, where $\boldsymbol{x}$ is the *input* (for instance an image, some text or a vector of numbers), and y is the *label*, which is the correct answer the model should predict. An input will generally be a vector of different *features*, which are parts or properties of the input; for example, a text input will have words or syllables as feature, an image the RGB values of its pixels, and a tabular input such as an inventory will have columns as features. Formally, an input $\boldsymbol{x}$ is defined as $\boldsymbol{x} = (x_1, x_2, \dots x_n)$ where $x_1, x_2, \dots x_n$ are its features.

Definition 1.1 (Model)**.** Given a dataset $\mathcal{D}$ with data points $d = \langle \boldsymbol{x}, y \rangle$, let $\mathcal{X}$ and $\mathcal{Y}$ be the set of inputs and labels respectively, then an *(AI) model* $\mathfrak{f}$ is a function $\mathcal{X} \to \mathcal{Y}$ with parameters $w_1, \dots, w_M$ that takes an input $\boldsymbol{x} \in \mathcal{X}$ and returns a prediction $p \in \mathcal{Y}$.
If the prediction p is equal to the label y of the data point $\langle \boldsymbol{x}, y \rangle$ we say that the model's prediction is *correct*. [1]

To better understand the inner workings of a model we have to know what architecture it features and how it is trained. The *architecture* of a model refers to the underlying structure of the model, the types of modules it is composed of, how they are connected, how many parameters they have and which are trainable, and finally, how the prediction is made. Hence, it specifies the algorithm used to obtain the prediction from the input and the model parameters. *Training*, on the other hand, refers to the algorithm which optimizes model's parameters usually starting from a random initialization; algorithmically, it is a process that starts with an architecture, its parameters and a dataset, and returns new values for the trainable parameters, optimized on the dataset.
As anticipated, we are going to detail the workings of BERT in the next section.

[1] Notice that *generative* models can be captured by the above definition as well, if we expand $\mathcal{Y}$ to be the set of all possible predictions the model can make. In non-generative models, the set of labels and the set of possible predictions do coincide, while in generative models the former is a subset of the latter.

BERT

Introduced by Devlin et al. [14] in 2019, BERT was developed to solve a plethora of tasks related to understanding, generation and manipulation of text. It had an enormous impact on the Natural Language Processing (NLP) field due to its performance: with a two-step training algorithm the proposed architecture was able to achieve top performance across multiple tasks, often beating models with ad hoc architecture optimized for one specific task.[2]

BERT's architecture is based on transformers, by Vaswani et al. [54], which are modules tailored to sequences of inputs (as text or time-series), composed by an encoder and a decoder, conjunctively creating a representation of the input sequence that is used by the model to make predictions.

Both submodules of a transformer are constituted by multiple *attention heads*; as tokens of the sequence are fed to the transformer, they are encoded keeping information on their relative position; those embeddings are then modified by attention heads to incorporate information from previous inputs. Each attention head has its own trainable parameters in order to attend to different types of contextual information (like grammatical, semantic, emotional, locational, ...), so, by stacking multiple attention heads, transformers are able to compute representations of the input sequence without loosing information of the positions and context in the sequence.[3] The resulting representations are then concatenated, and passed onto the next submodules of the transformer, which will use the encoded sequence and its contextual information to predict the next token.

Example 1.2 (Attention-head workings)**.** To make things clearer, we provide a visualization of the workings of one attention head for the sequence `'red lobate leaf'` in Figure 1.1. The image is not derived from an actual run of a transformer, and the 3D space where embedding are visualized is purely illustratory. Intuitively, information from the context is represented as a vector in the space of word embeddings, which is added to the original embedding to obtain an updated vector enriched with context. In the visualization, the head matches adjectives to their nouns, so the corrections from adjectives referring to `leaf` have higher magnitude than those coming from adjectives not related to the word, or from other grammatical constructs.

BERT is composed by 12 transformer's encoders stacked one on top of the other, each with a hidden dimensionality of 768 and 12 attention heads for a total of 110 million parameters, see Figure 1.2. The original model from [14] had two versions, we will always refer to the base version. The textual input $\boldsymbol{x}$ is firstly passed through a tokenizer, which transforms the word-sequence into a token-sequence according to a predetermined vocabulary, and also embeds each token in a 768 dimensional vector representing it.[4] Supplementary information on the relative and absolute position of the token in the sequence is added at this stage. Afterwards, the embedded sequence is passed progressively throughout all the layers of transformers, where contextual information is added, and finally it reaches the output layer, which is chosen based on the specific task BERT is applied to.

What makes BERT so flexible, is the fact that it can be trained for a specific task by only changing the parameters of its last output layer. In fact, BERT's training is a two-step process: during the first phase, *pre-training*, the model is trained on two tasks: *Masked LM (MLM)*, i.e. predicting a masked word in a sequence, and *Next Sentence Prediction (NSP)*, a binary classification task which involves predicting if two sentences are one after the other or not.[5] Throughout pre-training, the parameters of the tokenizer and of the transformer's encoders are adjusted; according to Devlin et al. [14] the model develops a general understanding of syntactical concepts as well as semantic relationships between words from the first task, while the second adds understanding

[2] At the time, that was particularly surprising, as a common assumption in the AI field was that specific tasks required specialized models.

[3] Notably, transformers achieve this without needing backward or recurrent connections, which were assumed to be fundamental for storing information on sequential inputs. Avoiding similar connection makes computing the error of the model's predictions and the training mathematically easier and computationally faster.

[4] The model is able to inspect the input sequence from left-to-right, as a human English reader and as most other NLP models, and also from right-to-left.

[5] Note that large datasets for the two tasks can be created cheaply and automatically, as labels are extracted directly from the inputs.

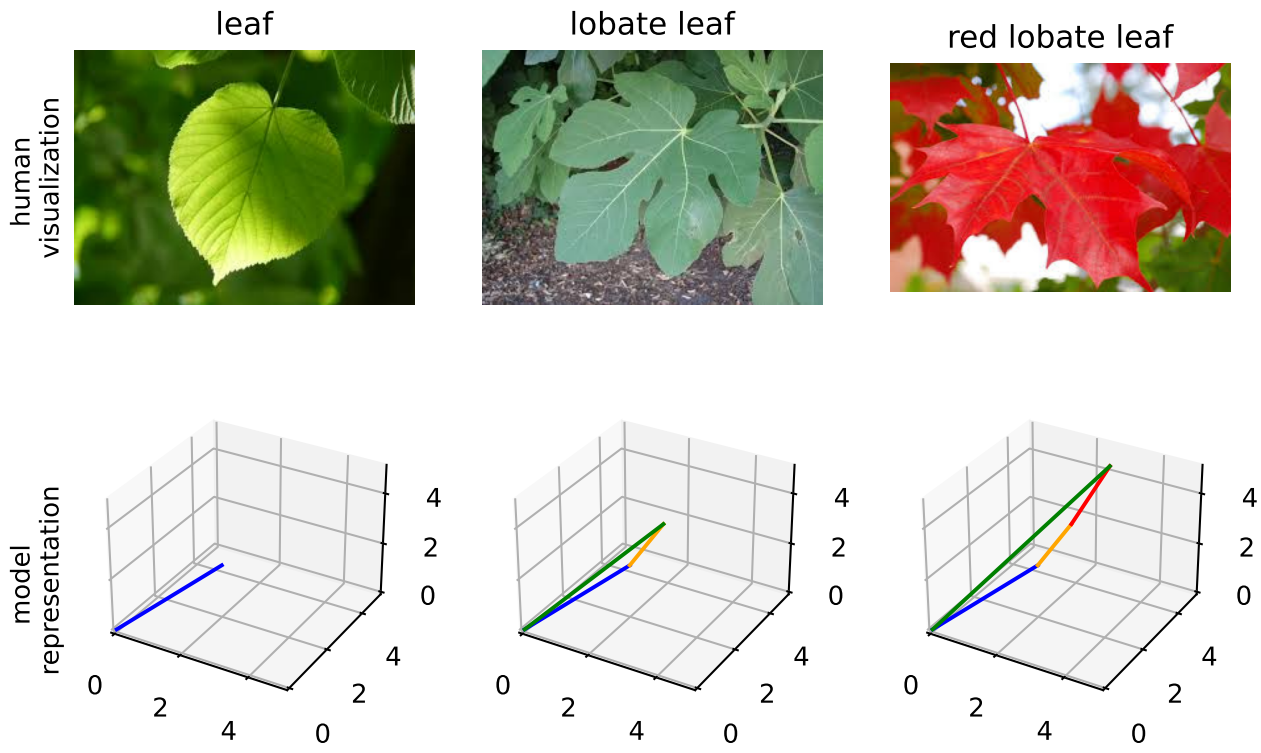

Figure 1.1: Visualization of the action of one attention head. In the first row, photos to visualize how a human would ideally internalize and represent the words; on the second row, an idealized visualization of the multidimensional space of the model's embeddings, as progressively modified by the attention head. The blue vector is the initial embedding, which is firstly projected into a subspace, and then adjusted by the attention head through the context. The first orange adjustment represents the addition of the information contained in `lobate`, while the red vector represents the addition of `red`. At each step, the green vector is the final representation formed by the attention head.
The representations of each head are concatenated and expanded back in the initial dimensionality.

of relationships between sentences and different contexts, which cannot be directly assimilated with MLM. The authors also note that including NSP improves the performance of the model on multiple tasks based on sentence to sentence relations, like question answering (which will be address with more detail in Chapter 2) and Natural Language Inference (where the model has to indicate if a sentence follows from another).
Later, begins *fine-tuning*, an additional training phase tailored at the specific task the model should solve (*downstream task*). The dataset of that task is now used, and the input and output layers are adjusted accordingly. In this phase, only the parameters of the output layer are modified, with an extremely low computational cost. Intuitively, during fine-tuning the model should calibrate the general representations learned in pre-training to the downstream task, similarly to a human agent, which, to become an expert, firstly learns general knowledge about the world, and then specializes in a subfield.

In Section 2.2 we are going to provide further details on how we fine-tuned BERT for our experiment, which greatly showcases both the importance of a good dataset for this phase, and the overall flexibility of the training method.

1.2 Why did the model predict that?

Given the recent growth in tools and technologies based on AI models, and the almost exponential increase in parameters and complexity, understanding how and why a model generates one specific output given an input has become more and more challenging. While with simple models such as perceptrons [44] or shallow neural networks with few parameters [4], it is in principle possible to understand why the inference was drawn by

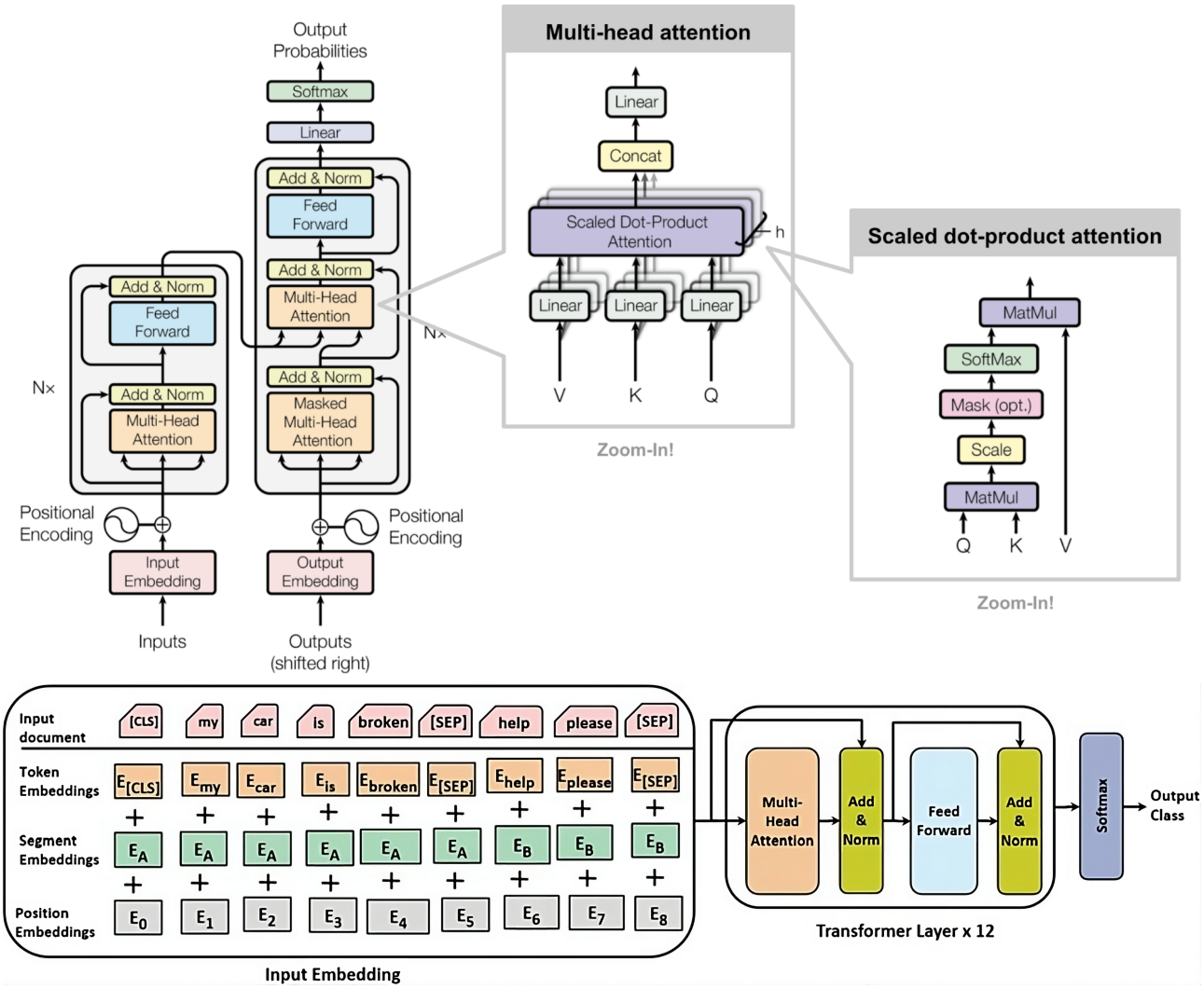

Figure 1.2: Above, the scheme of a transformer from [54] and [58]. The encoder (top, left) is a neural network made of N layers of sub-layers (highlighted in light-blue) constituted by a multi-head self-attention layer and a fully-connected layer (feed forward). The input sequence is firstly embedded, then it is passed through the first attention layer. Before the output is passed to the next sub-layer, the original input of the sub-layer is added (residual connection) and the result is normalized (with layer normalization). The decoder has a similar structure, with an added multi-head self-attention layer before the fully-connected one, which takes as input the output signal of the encoder.
Every multi-head attention is a stack of h scaled dot-product attention modules, each with its own parameters, attending to different kinds of contextual information.
Below, the architecture of BERT base model, from [14]. The input passes in an embedding layer, where words are translated into tokens, and additional information on their position is added (segment and position embeddings). Then, the embedded input is passed through 12 transformer's encoders, one on top of the other, each with 12 attention heads in the multi-head attention module (hence, $N = 12$ and $h = 12$ in the first image). The final output layer, trained in the fine-tuning step, is related to the downstream task the model should solve: for a classification problem it is a softmax, as in the picture, while for question answering there are two parallel output layers, one for the starting and one for the ending index of the answer, each composed by a linear layer and a softmax.

directly analysing the magnitude and sign of the model's parameters, this is impossible once the number of parameters reaches the thousands, not to mention the billions of current state-of-the-art models; thus, there seems to be a still unresolved trade-off between accuracy and interpretability.[6]
With application comes regulation, and multiple entities, from industry [2], academia [18] and legislative bodies [20], have called for new regulations, to ensure that AI models and the resulting tools are tested for privacy, fairness, robustness and interpretability. This has sprung new effort in the field of explainable artificial intelligence (XAI), which we briefly comment in this section.

The current directions of research in XAI can be divided into two branches: the first has grown from the idea of solving the accuracy-interpretability trade-off by developing new architectures that make decisions following algorithms which are understandable and explicable by design [45]. These models are called *white-boxes*, as they are purposefully designed so that developers and users can inspect how the model predicts. In contrast, most state-of-the-art models are called *black-boxes*, as they are composed of millions or billions of parameters, each only partially responsible for the final output of the model; thus, by design, their complexity prevents grasping the reasons behind their predictions.[7]

On contrary, the intuition behind the second branch is that the current black-box models, having high accuracy, should not just be thrown away because they are difficult to understand and troubleshoot. Instead, new techniques must be developed on top of them, to help us analyse their behaviours.
Among those, *feature attribution methods* were designed to highlight the relative importance of input features for the prediction of the model. In the Introduction, we anticipated how intuitive and helpful those explanations can be, and we are going to discuss some of their shortcomings and risks one might run into in the next paragraph. Since in what follows we are going to test a framework developed to solve one major problem of these methods, we define explanations as they are generated by feature attribution methods.

Definition 1.3 (Explanation)**.** Given a model $\mathfrak{f}$ and a dataset $\mathcal{D}$ with data points $d = \langle \boldsymbol{x}, y \rangle$, we define the (feature-attribution) *explanation* $\boldsymbol{e}$ of d to be

$$\boldsymbol{e} = \mathfrak{e}(\mathfrak{f}, d) = \mathfrak{e}(\mathfrak{f}, \boldsymbol{x}, y),$$

with $\mathfrak{e}$ a feature attribution method.
We say that such explanation is *local*, as it is a function of one data point only.

While multiple feature attribution methods were proposed [43, 13, 49], SHAP values have emerged as one of the most widely used solution, and have been employed to test robustness and fairness of the most disparate model architectures.

SHAP values

Inspired by the prolific game theoretic concept of Shapley values [47], Lundberg and Lee [35] defined a local feature attribution method with similar foundational intuitions and axiomatic motivations, SHAP values.
The original game theoretic formulation was created to solve the problem of how to calculate the importance of each player in coalition games from an axiomatic perspective: Shapley defined some axioms that a fair evaluation of player's contribution should respect, and found that there was only one equation that satisfied them all conjunctively, which is the one used to define Shapley values.
Similarly, authors of [35] noticed that most local feature attribution methods satisfied the *additivity axiom*. After formulating three other axioms which should reasonably hold for an explanatory method that seeks to find how much each input feature contributed to the final prediction of the model, similarly to Shapley, they

[6] Proposers of white-box models have argued that this trade-off, far from being actual, is only perceived [45]; note however, that while white-box models nowadays match the accuracy of state-of-the-art models for structured dataset, this is not the case for unstructured datasets such as images or text.
[7] For a survey on white- and black-box models see [31].

found a unique solution that respected all the axioms, and studied its computational complexity as well as its explanatory power, proving that it performed better than previous solutions.
We list the axioms and an intuitive explanation of what they assure next.

Definition 1.4 (SHAP axioms)**.** Given a model $\mathfrak{f}$, and datapoint with input $\boldsymbol{x}$, with a total of n features, we indicate with $\boldsymbol{x}'$ the simplified version of $\boldsymbol{x}$, i.e. $\boldsymbol{x}' \approx \boldsymbol{x}$, with a mapping function $h_{\boldsymbol{x}} : \{0,1\}^n \to \mathcal{X}$, such that $\boldsymbol{x} = h_{\boldsymbol{x}}(\boldsymbol{x}')$ and $\boldsymbol{x}' \in \{0,1\}^n$. We also use $\boldsymbol{v}'$ to refer to any vector in $\{0,1\}^n$ whose 1 entries are a subset of the 1 entries of $\boldsymbol{x}'$, whereas $\boldsymbol{v}' \setminus j$ is the vector equal to $\boldsymbol{v}'$ in all entries, except the jth, which is set to 0.
Then, according to [35], the explanation of the jth feature, ϕ_j should satisfy the following axioms:

Additivity The sum of the contribution values of all the feature generated by the explanation method approximates the output of the explained model; this amounts to require that SHAP values are linear functions of binary variables, which are obtained by mapping the original features of the input, $\boldsymbol{x}$, to a simplified input, $\boldsymbol{x}'$.[8]

$$\mathfrak{e}(\mathfrak{f}, \boldsymbol{x}') = \sum_{j=0}^{n} \phi_j \boldsymbol{x}'_j$$

Local Acc The prediction of the explained model on input $\boldsymbol{x}$ are approximated by the explanation model on its simplified version $\boldsymbol{x}'$ ($\boldsymbol{x} = h_{\boldsymbol{x}}(\boldsymbol{x}')$).

$$\mathfrak{f}(\boldsymbol{x}) = \mathfrak{e}(\mathfrak{f}, \boldsymbol{x}')$$

Missingness Features that are not present in the original input have no impact; therefore their contribution value equals to 0.

$$\boldsymbol{x}_j = 0 \Rightarrow \phi_j(\mathfrak{f}, \boldsymbol{x}) = 0$$

Consistency Comparing two explained model, $\mathfrak{f}$ and $\mathfrak{f}'$, if the contribution of a specific feature to the prediction of the $\mathfrak{f}$ is bigger or equal then the contribution of the same feature for $\mathfrak{f}'$, then the contribution value of that feature for $\mathfrak{f}$ should be bigger or equal to the contribution value of the same feature for $\mathfrak{f}'$.

$$\text{if } \mathfrak{f}_{\boldsymbol{x}}(\boldsymbol{v}') - \mathfrak{f}_{\boldsymbol{x}}(\boldsymbol{v}' \setminus j) \geq \mathfrak{f}'_{\boldsymbol{x}}(\boldsymbol{v}') - \mathfrak{f}'_{\boldsymbol{x}}(\boldsymbol{v}' \setminus j), \text{ for all } \boldsymbol{v}' \in \{0,1\}^n, \text{ then } \phi_j(\mathfrak{f}', \boldsymbol{x}) \geq \phi_j(\mathfrak{f}, \boldsymbol{x})$$

Definition 1.5 (SHAP values)**.** Given a model $\mathfrak{f}$, an input $\boldsymbol{x}$ with its simplified version, $\boldsymbol{x}'$, as before, the SHAP value of feature j, ϕ_j, is obtained as:

$$\phi_j(\mathfrak{f}, \boldsymbol{x}) = \sum_{\boldsymbol{v}' \subseteq \boldsymbol{x}'} \frac{|\boldsymbol{v}'|!(n - |\boldsymbol{v}'| - 1)!}{n!} \left[\mathfrak{f}_{\boldsymbol{x}}(\boldsymbol{v}') - \mathfrak{f}_{\boldsymbol{x}}(\boldsymbol{v}' \setminus j)\right]$$

where n is the total number of features of $\boldsymbol{x}'$, $\boldsymbol{v}' \in \{0,1\}^n$, $|\boldsymbol{v}'|$ is the number of 1s in $\boldsymbol{v}'$, and we use $\boldsymbol{v}' \subseteq \boldsymbol{x}'$ to indicate the set of all $\boldsymbol{v}'$ vectors for which all the entries with 1 are a subset of the 1-entries of $\boldsymbol{x}'$.
Intuitively, $\boldsymbol{v}'$ indicates which features are masked (0) and which aren't (1). The sum ranges over all possible combinations of masks, while the fraction before the square brackets is a weighting factor. The term in square brackets is the contribution of the jth feature for a specific masking $\boldsymbol{v}'$, calculated as the difference between the model output when j is present, $\mathfrak{f}_{\boldsymbol{x}}(\boldsymbol{v}')$, and when it is not, $\mathfrak{f}_{\boldsymbol{x}}(\boldsymbol{v}' \setminus j)$. The SHAP value of jth feature is thus calculated as the weighted average contribution of j for the model output across all possible masks.

As anticipated in the Introduction, the definition of SHAP values has quite bad computational complexity: the exact calculation requires the analysis and comparison of all possible subsets of input features (analogously to the original Shapley values). Nonetheless, [35] provides various alternative algorithms that are more efficient, although only approximating the correct values.[9]

[8] This axiom formalizes the summation aspect, while Local Accuracy expresses the approximation part.

[9] Similar to other local feature attribution methods [43, 13, 49], for the approximation authors assume feature independence, which is reasonable for (some) tabular domains, and at least debatable for text or image based tasks.

Example 1.6 (SHAP values)**.** In Figure 1.3, we showcase SHAP values of a linear regression model of a tabular dataset for predicting housing prices. The visualization is very intuitive and provides a clear overview of the features that mostly contributed to the model's prediction.

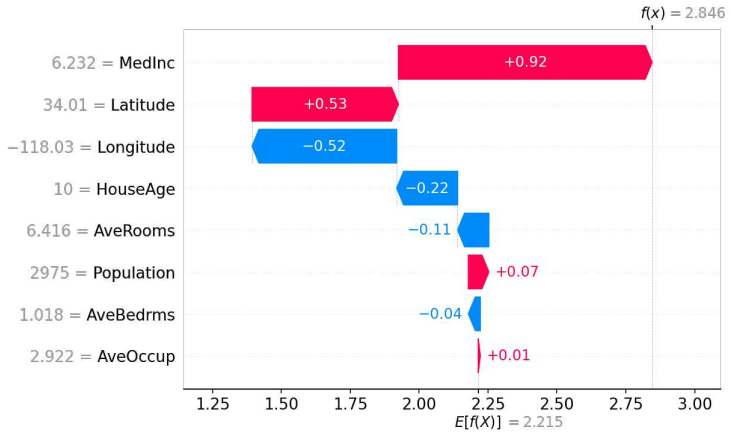

Figure 1.3: An example of SHAP values of a linear regression model for predicting housing prices. Features that contributed positively are plotted in red, while those that have a negative contribution are coloured in blue. Image is taken from [33].

In Figure 2.2 we visualize SHAP values of BERT-based models solving question answering task. As the number of features in NLP tasks is very high, it is hard to track and understand why certain words or punctuation signs have higher values then others.

Despite their intuitiveness and applicability, multiple concerns were raised about the correctness of the approximation and of the underlying assumptions, as well as on the methods and practices where SHAP values are employed. While discussing each problem specific to SHAP values is outside the scope of this work, we detail in the next paragraph some of the most relevant critiques of local feature attribution methods as explanatory methods.

Issues of local feature attribution methods

A first attack to feature attribution methods in general is posed from the perspective of *consistency*: different explanatory methods, if they do not agree on the relative importance of each feature, should at least agree on the importance based ordering.
Yet, Neely et al. [39], while investigating the correlations between feature attribution methods (SHAP and LIME) and other attention-based explanatory methods in NLP, found not only that across multiple models and tasks the former correlate poorly with the latter, but also that different feature attribution methods generate explanations which are very weakly correlated.[10] Further experiments by Krishna et al. [28] and by Zhou et al. [60] corroborated these results, showing that also in tabular and image modalities such inconsistencies persist.

Another concern is that of *fidelity*: while feature attribution methods are defined so that they closely match the predictions of the model that is being explained (additivity and local accuracy axioms of Definition 1.4), at times they do not [45]. As such, explanations are just approximations of the inference process of the model, and thus, they can be a further source of errors when interpreting the model's predictions [19].

Nonetheless, we argue that the above critiques, while being applicable generally to feature attribution methods, fundamentally depend on the choices of definition and implementation of singular methods. To be clearer, while the feature attribution methods developed so far suffer, at times, of infidelity to the model we want to explain, and, at times, of inconsistencies with other explanations, this does not rule out that, in future, better

[10]The results are particularly relevant for transformers, and for specific tasks solved by LSTMs.

explanatory methods will be less effected by those problems. Furthermore, even if current methods are only approximations of the decision process of the explained model, they can still be useful to gain insights about what parts of the input are more relevant for the model, if approximation errors in explanations are limited to small subsets of the dataset.
However, a major difficulty remains, and it is congenital to the basic assumptions on which feature attribution methods rely. Suppose we have a model that solves a binary classification task; the explanations obtained for the model consist of a list of positive or negative numbers that quantify the relevance of each feature for the model's prediction. Inspecting those values we can only infer whether the model saw that feature as contributing positively or negatively towards the output. However, we have no information on how that feature was used in the decision process, nor we can understand if for those features the model has developed inner representations that correspond to concepts we would use to make the decision.
Even assuming that the explanations are faithful to the model, there is still an *interpretability gap* between the explanations and the concepts we would use to solve the task [19, 45]. In other words, even if the model and the explanation are correct, and the most relevant features align with our intuitions about how the task should be solved, we have no guarantee that the model is assigning relevance to those features for the same reasons we do: we solved the task relying on a conceptual framework that we have no justification to project on the model, even if the obtained explanations and our intuition align on some datapoints.
What is worse is that, as feature attribution methods are very intuitive and easy to visualize across all modalities, we tend to unconsciously fill the gap by assuming that the model is behaving as we would. Be it by cherry-picking examples that confirm our expectations, or by positively and optimistically interpreting few explanations as corroborating evidence for our desiderata, the risk of confirmation bias is high and documented [45, 57, 56, 8]. And because this problem is tightly rooted in the foundations behind the definition of feature attribution methods, it seems a better reason to argue against their use. Yet, as we will see in the next section, something can still be done to rescue them.

1.3 How to explain better?

While Cinà et al. [10] agree that the gap between explanations generated by feature attribution methods is the most problematic cause of errors when they are used to understand model's behaviour, they argue that those risks can be mitigated by a more careful study of the congruence between the explanations and our beliefs on the workings of the model and on the ways in which one should solve the downstream task.

The main contention of the authors is that there is a substantial difference between *structured* and *unstructured* datasets, and that the interpretability gap is present most often in the latter.

Example 1.7 (Structured and unstructured datasets)**.** Structured datasets are, for instance, tabular datasets: here, each input feature comes already endowed with an understandable semantic, which can be interpreted independently of all the others. Hence, in a tabular dataset containing results of blood exams, very high levels of sugar can be interpreted by doctors as hyperglycaemia, independently of other features (as white blood cells, red blood cells, ...). Sugar levels preserve their specific meaning when considered together with the other features for the final diagnosis.
On the other hand, typical examples of unstructured datasets are those of images and texts. In the latter, tokens consists of single worlds or syllables, which taken alone might offer multiple interpretative alternatives, while the actual meaning of each excerpt emerges only when considering all the features together. Similarly, in image datasets the input usually consists of RGB values of the image's pixels, each being a feature. Here too, focusing on just one individual pixel would be meaningless and pointless, and it is possible to interpret the features only by aggregating pixels into bigger portions of the image.

In structured datasets, the sub-symbolic representation of the input that is fed to the model is equal to the human representation: the level of sugar in blood is both an input feature for the model and something that we

can interpret. On the other hand, in unstructured datasets, the sub-symbolic representations given to the model (pixels of images, tokens corresponding to words or syllables) are different from the representations we build at higher levels of abstraction (parts of an image, words and general meaning of a text).
While for the first type of dataset there is no actual interpretability gap for low-level features, and thus, the previously discussed risks are mitigated, it is usually for the second type of dataset, that confirmation biases and projections of human beliefs can, and do, happen, as we understand the input and solve the task through high-level features which the model might not recognize.
Yet the presence of that gap is not sufficient to throw completely away feature attribution methods when the dataset is unstructured. On contrary, the idea is to fill that gap with a proper translation, to convert sub-symbolic features whose meaning is hard to grasp into human representations that are understandable. Whenever such translation exists, there is *semantic match* between the two, and we avoid the risk of projecting our knowledge onto the model, as our higher-level representations have a correspondence to the representations of the sub-symbolic features that the model receives and uses. In those cases risks of confirmation bias are also reduced: it is no longer possible to cherry-pick the data points and explanations that better suit our prior expectations, as either the translation of the obtained explanations align with our expectations, or they do not.

But how can we build a similar translation? And how can we test it to be sure that it holds? In [10, 11] authors propose a framework to formally define semantic match, and a procedure to test it in practice, agnostic to the choice of model and of feature attribution method. In the next chapter we are going to present the framework in full detail, while here we provide the basic intuitions behind it.
In a nutshell, we want to structure the high-level representations we understand and the intuitions about how the task should be solved so that it becomes possible to compare them with the obtained explanations: firstly, we make a hypothesis in natural language, and then we define a process to map low-level features into high-level ones; as we shall see, this enables us to measure how well the hypothesis aligns with the explanations of the model's behaviour.

Still, in order to properly fill the interpretability gap, we have to ensure that the overall translation from hypotheses to explanations is correct. To this end, two fundamental questions have to be tested, taking as reference a data point where the high-level feature of the hypothesis is present and the explanation $\boldsymbol{e}$ matches the behaviour specified in the hypothesis:

- Is it true that if a data point satisfies the hypothesis then it also has an explanation similar to $\boldsymbol{e}$?
- Is it true that the data points with explanations similar to $\boldsymbol{e}$ also have the high-level feature of interest and follow the defined behaviour?

Note that, to answer these questions, we also need a notion of similarity between explanations, which follows directly, once we define a distance.[11] Intuitively, the two queries investigate respectively how necessary and how sufficient the hypothesis is to assure explanations' similarity. A positive answer to both is required for semantic match, as, otherwise, the interpretability gap is not filled: a negative response to the second question would be detrimental, because it leads to confirmation bias and knowledge projection, while a negative answer to the first is equally dangerous, as it implies that the model is not coherent in the way in which it considers the high-level feature of interest, possibly being unreliable.

Example 1.8. To see this, consider a binary image classification with classes $\{\texttt{cow}, \texttt{bicycle}\}$. We would assume that a good high-level feature to solve the task is the presence of wheels, i.e. one or more circularly shaped areas in the image. Inspecting the explanations we find that when there are wheels the model is considering them important.

The first question would have a negative answer, when, for instance, the wheels' area in actual images of bicycles has at times positive, and at times negative contribution; in turn, this means that, while the model is

[11] We provide formal definitions of explanation, hypothesis and distance in Section 2.1.1 and 2.1.2; the procedure to test semantic match is detailed in Section 2.1.3.

focusing on that high-level feature as expected, sometimes it interprets it as positive evidence for the `bicycle` class, sometimes as a negative signal.

On the other hand, if we observe similar circular patterns on cow images, then our interpretation of those patterns as wheels is possibly wrong. Here the major risk for confirmation bias arises, as if we do not discover or test that the circular areas appear also on objects that are not wheels, we might conclude that the model is correctly recognizing wheels, whereas, in fact, it is not.

Another way to visualize semantic match is to think of explanations and human representations as sets of states (data points): if the two sets coincide, we have semantic match, otherwise, if the former is a subset of the latter, the first answer is negative, while if the latter is a subset of the former, the second answer is negative. In practice, the two sets will almost never fully coincide, as other high- or low-level features that also affect the final prediction might increase or decrease the relative importance of the hypothesis' feature. Even if semantic match is not exact, it is still important to test, as its lack could increase awareness of possible confirmation biases, and its presence could add insights in the model's behaviour and evidence of the correctness of its interpretation.

The translation between sub-symbolic features and hypotheses will rely on a *mapping* and a *composition*: the first aggregates all the sub-symbolic features of the input in the high-level features considered in the hypothesis; for instance, in Example 1.8 the mapping will segment the image and, if a wheel is present, associate to it all the pixels it is made of.
Next, with the second, we compute a statistic of the explanations of the sub-symbolic features returned by the mapping, based on the way we defined the hypothesis. So in Example 1.8, as the area where a wheel is present should have positive contribution towards predicting `bicycle` class, and negative contribution for `cow`, we could simply use the sum of the explanations of all the pixels composing it, and check if the result is positive or negative when bicycles or cows appear. Instead, if we are interested in knowing if the model attends to wheels at all, we could take the absolute sum, as a pixel with high contribution, be it positive or negative, is important for the model's prediction.

Concatenating the mapping and the composition, we have a way to obtain explanations for high-level, human understandable features from the low-level sub-symbolic input features.

1.4 Notational conventions

We will use *italics* to denote data points and numbers such as thresholds or constants, ***bold*** for vectors, and latex's `mathfrak` font for functions. We use latex's `mathcal` font to denote sets, `mathsf` for complexity classes and `texttt` for names of variables in the codebase and code snippets.
We use the subscript i to indicate the ith data point d_i of a dataset with N elements, and j to denote the jth feature of a data point input vector with n entries. The two exceptions to our notation used for functions are median distance and area under the curve, which, as metrics, are indicated with MD and AUC in formulas, and abbreviated as MD and AUC in the text.

Chapter 2

Methodology

This chapter is divided into two parts: in the first, we provide a detailed account of the semantic match framework, define its key concepts and metrics, and expand the previous discussion on the intuitions and objectives that motivated the framework. We additionally include a procedure to apply it to any case-study.
In the second part, we present the setup of the experiment, with details about the dataset and the task which the model should solve, the training procedure and the most important aspects of the implementation of the experiment.

2.1 Semantic match

Here we provide a more detailed account of the semantic match framework, its key-concepts and metrics, and close by outlining the steps needed to apply the framework on any model we want to study.

2.1.1 The framework

As seen in Section 1.3, the goal of semantic match is to avoid confirmation bias and to develop a more reliable and robust understanding of model's behaviour by checking how well explanations of the model obtained from feature attribution methods comply with human-friendly hypotheses.

The key components of the semantic match framework are the model, the explanation method, the hypotheses and distances used, and finally the evaluation metrics. We already defined a model (Definition 1.1) and an explanation (Definition 1.3) in Chapter 1, and in this section we lay out the remaining definitions and showcase how they integrate together.
As previously, we indicate with $\mathcal{D}$ the dataset, with $\langle \boldsymbol{x}, y \rangle$ any of its data points, as an input-label pair, with $x_{i,1}, \ldots, x_{i,n}$ the n features of the input $\boldsymbol{x}_i$ of data point $d_i \in \mathcal{D}$, with $\mathfrak{f}$ the model, with $p_i = \mathfrak{f}(\boldsymbol{x}_i)$ the model prediction for $\boldsymbol{x}_i$, and with $\boldsymbol{e}_i$ the explanation obtained from the explanation method $\mathfrak{e}$ when given $\mathfrak{f}$ and d_i, i.e. $\boldsymbol{e}_i = \mathfrak{e}(\mathfrak{f}, d_i) = \mathfrak{e}(\mathfrak{f}, \boldsymbol{x}_i, y_i)$. Finally, the explanation for the jth feature of the ith data point is written as $\boldsymbol{e}_{i,j}$.
As the hypotheses for semantic match will concern the model, the input and the label, we define tuples $\boldsymbol{u}_i = \langle \mathfrak{f}, \boldsymbol{x}_i, y_i \rangle$, and sometimes use those tuples as inputs for the explanation method, i.e. $\boldsymbol{e}_i = \mathfrak{e}(\boldsymbol{u}_i)$.[1] The set of all $\boldsymbol{u}$ tuples is $\mathcal{U} = \{\boldsymbol{u} = \langle \mathfrak{f}, \boldsymbol{x}, y \rangle \mid (\boldsymbol{x}, y) \in \mathcal{D}\}$, while the set of all explanations is $\mathcal{E} = \{\boldsymbol{e} = \mathfrak{e}(\boldsymbol{u}) \mid \boldsymbol{u} \in \mathcal{U}\}$, and by definition $|\mathcal{D}| = |\mathcal{U}| = N \geq |\mathcal{E}|$. As we use feature attribution methods, each explanation is a vector of n entries. When we are talking about any data point in general and when is otherwise clear, we omit the i subscript.

[1] Note that explanations generally do not require the label y_i, and we include it in the explanation method input both because we allow the hypotheses to refer to the label, and for ease of notation.

Definition 2.1 (Hypothesis)**.** A *hypothesis*, θ, is a logical statement structured as

$$\text{if } A \text{ then } B,$$

with A being a constraint on the data points, the model or its prediction, i.e. on $\boldsymbol{u}$, and B expressing a behaviour of the model in terms of an explanation, i.e. a condition on $\mathfrak{e}(\boldsymbol{u})$.

Example 2.2 (Housing prices)**.** Suppose we are studying a model that predicts housing prices.[2] We might suppose that the number of rooms or the total area of the dwelling correlate positively with the price, but we might have fewer clues about how exactly and how deeply the pupil-teacher ratio or the nitric oxides concentration influence the price, or if they are relevant at all for the model's predictions. Furthermore, we might assume that the first two features will have a stronger influence on the price, and should thus have a more profound influence on the prediction. To evaluate this supposition, we could formulate the hypothesis: $\theta_1 =$'for the prediction the number of rooms is very important', and also $\theta_2 =$'the influence of the number of rooms on the prediction is greater than that of the pupil-teacher ratio and that of nitric oxides together'.
Formalizing the two hypotheses, as per Definition 2.1, can be done as follows: firstly, we should *identify* a specific set of data points, and then *quantify* some difference in the explanation that relates to that behaviour: for both θ_1 and θ_2 the behaviour we want to ascertain is the model's attribution of importance to features. As such, the behaviour is not influenced under any specific condition, so for both hypotheses, the antecedent A will be an empty requirement (i.e. the always true condition, $\top$). Assuming that for each feature the explanation method always gives us a positive contribution value in $[0, 1]$, which reflects how much influence the feature had in the prediction, relatively to the importance of the other features,[3] the behaviour we want to quantify will be that the contribution of the first feature is bigger than the sum of the contributions of the other two for θ_2, and that the first feature has high contribution when compared to the others for θ_1.
Formally then:

$$\theta_1 = \top \Rightarrow \boldsymbol{e}_{i,1} \geq z$$
$$\theta_1 = \top \Rightarrow \boldsymbol{e}_{i,1} \geq \boldsymbol{e}_{i,3} + \boldsymbol{e}_{i,4}$$

with $\boldsymbol{x}_{i,1}$ being the number of rooms, $\boldsymbol{x}_{i,3}$ and $\boldsymbol{x}_{i,4}$ the pupil-teacher ratio and the nitric oxides of data point d_i, and z being a threshold in $[0, 1]$ for which different values can be tested.

Intuitively, A puts some constraint on the data points we want the hypothesis to apply to. If we want θ to apply to the whole dataset, we don't put any constraint: instead, when we want to check the behaviour of the model only on specific inputs, we define a related constraint which A enforces. For instance, in the above example, if we think that high levels of nitric oxides (associated with acid rains) should strongly decrease the predicted price, we could check if whenever the levels are high, the model considers very important the corresponding feature. The constraint on high levels of nitric oxides could be added by defining A as the condition $\boldsymbol{x}_{i,4} \geq 3$.[4]
Then we could check if in those cases the explanation of that feature, $\boldsymbol{e}_{i,4}$ is higher than the other features.
On the other hand, the consequent B has the role of quantifying the behaviour we want to test, and does so using the statistics provided by the explanation.
In contrast to Zhou, Ribeiro, and Shah [59], where hypotheses are formulated on low-level features, our framework allows for hypotheses on arbitrary sets of features, that is, high-level features. Thus, not only we can check which are the low-level features that the model deems more significant, but we can also evaluate model's behaviour on agglomerates of low-level features, from shapes, colours and parts of an image (at different compositional levels,

[2] For example a simple linear regression model trained on the Boston Housing Dataset of [21].
[3] While SHAP values can be negative, we will consider their magnitude, as we are interested in the overall influence of each feature on the prediction, be it positive or negative.
[4] According to a document by the New Jersey Department of Health, [22], the level at which it is possible to smell the gas is between 3 and 10 parts per 10 million.

i.e. from the single parts to the object as a whole), to entities' names, locutions, sentences, and even entire sections of a text.

Definition 2.3 (Hypothesis' satisfaction)**.** Given a tuple $\boldsymbol{u} = \langle \mathfrak{f}, \boldsymbol{x}, y \rangle$ with $\mathfrak{f}$ the model and a data point $d = \langle \boldsymbol{x}, y \rangle$, and given a hypothesis θ, if $\boldsymbol{u}$ satisfies the constraint imposed by A and $\mathfrak{e}(\boldsymbol{u})$ satisfies the condition of B, we say that $\boldsymbol{u}$ *satisfies* (complies with) θ, and we write

$$\boldsymbol{u} \vDash \theta.$$

We also use $\boldsymbol{u} \vDash_A \theta$ and $\boldsymbol{e} \vDash_B \theta$ with $\boldsymbol{e} = \mathfrak{e}(\boldsymbol{u})$ to indicate that the model and data point satisfy the constraint A or the behaviour B of θ. Finally, with $\mathcal{C}_A$ and $\mathcal{C}_B$ we indicate respectively the set of $\boldsymbol{u}$ tuples that satisfies the A- or the B-part of θ, and with $\mathcal{C}_\theta$ the set of tuples that satisfies the hypothesis. Then, $\mathcal{C}_\theta = \mathcal{C}_A \cap \mathcal{C}_B$.

We close this section by noting that the framework is based on the assumption that the explanations of a model faithfully capture the model's behaviour.[5] We chose SHAP values for their widespread use, but in principle any feature attribution method could be chosen instead.

2.1.2 The metrics

In order to evaluate how well a defined hypothesis matches the behaviour of the model, we use two metrics: median distance (MD) and area under the curve (AUC).
Intuitively, the former is a measure of how similar or dissimilar the obtained explanations are: the more similar they get, the more coherent the behaviour of the model is. The latter, on the other hand, is a way to assess how well data points that satisfy the hypothesis can be distinguished from those which do not satisfy it; as such, it can be used to check whether the model has learned to behave in a specific way on a specific group of data point.

Both metrics presuppose a notion of similarity of behaviours, which, in turn, hinges on a definition of similarity between explanations, as we assume that the explanations obtained from the explanation method, $\mathfrak{e}$, faithfully reflect the way in which the model makes predictions. This introduces an important relation between the definition of that distance, the hypothesis, and the final assessment of semantic match.[6] Once the hypothesis is defined, one must choose a distance accordingly: if θ specifies a behaviour over the explanation of one feature only, $\boldsymbol{e}_j$, any distance that depends on explanations of any other feature will scatter explanations with similar j contributions, when they have very different values on the other features considered by the distance. Conversely, explanations might get clustered together if the definition of distance does not include one of the features of the hypothesis. Both cases are undesired, as the AUC obtained will be lower, or higher than the one obtained with a correct definition of distance, in turn leading to wrong conclusions about the experiment.

Definition 2.4 (Distance)**.** Given two explanations $\boldsymbol{e}_i = \mathfrak{e}(\boldsymbol{u}_i)$ and $\boldsymbol{e}_{i'} = \mathfrak{e}(\boldsymbol{u}_{i'})$, we define the *distance* between them as a function $\mathfrak{d} : \mathcal{E} \times \mathcal{E} \to \mathbb{R}$ s.t. $\mathfrak{d}(\boldsymbol{e}_i, \boldsymbol{e}_{i'}) = 0$ iff $\boldsymbol{e}_i = \boldsymbol{e}_{i'}$, and $\mathfrak{d}$ satisfies positivity, symmetry and the triangle inequality.
Usually $\mathfrak{d}$ is normalized, thus restricting the codomain to $[0, 1]$.

Median distance

Median distance is used to assess the *coherence* of the model's behaviour, and relies on the definition of a distance $\mathfrak{d}$ between explanations.[7] It also depends on the specification of two sets: the reference set, which we fix to be the set of $\boldsymbol{u}$ tuples that satisfy the hypothesis, $\mathcal{C}_\theta$, and the sample set, the set of tuples selected to evaluate the median distance from. For the latter, we pick the set of all $\boldsymbol{u}$ that satisfy that A-part of the hypothesis, $\mathcal{C}_A$.

[5] Indeed, this should be the main goal of any explanation method; yet, the discussion on the reliability and fidelity of explanation is still open. We cited some pointers from that debate in the Related Work paragraph of Chapter 1.

[6] We will further discuss the interaction between the hypothesis' definition and the results of the metrics in Appendix F.

[7] In the following, we use coherent and consistent as interchangeable terms, to indicate that a model behaves in a peculiar and common matter on a given subset of data points.

Then, MD will give an indication of how consistently the model behaves according to B across all the tuples characterized by A. Ideally, the lower the median distance, the better, as this indicates that the behaviour of the model across the tuples that satisfy A is consistent. In Section 4.2 and Appendix E we will additionally evaluate MD using as sample set the set of all $\boldsymbol{u}$ tuples, $\mathcal{U}$, and choosing as reference $\mathcal{C}_\theta$ and then $\mathcal{C}_A$. The first will clarify how often the model follows the hypothesis' behaviour across all $\mathcal{U}$. In this case, if that median distance is low, the model behaves according to B regardless of the satisfaction of constraint A, while high MD signals that the model has very different behaviours across the dataset. The second evaluation will provide a measure of how separable $\mathcal{C}_A$ is from the set of tuples that do not satisfy A.

Definition 2.5 (Median distance)**.** Given a hypothesis θ, a fixed explanation $\boldsymbol{e}_r$, called *reference point*, and a set of $\boldsymbol{u}$ tuples $\mathcal{S}$, called the *sample set*, the *median distance of* $\mathcal{S}$ *from* $\boldsymbol{e}_r$ is defined as

$$MD(\theta, \boldsymbol{e}_r) = \text{median}\{\mathfrak{d}\big(\mathfrak{e}(\boldsymbol{u}), \boldsymbol{e}_r\big) | \boldsymbol{u} \in \mathcal{S}\},$$

with $\mathfrak{d}$ the distance between explanations and $\boldsymbol{u} = \langle \mathfrak{f}, \boldsymbol{x}, y \rangle$ tuples as previously defined. The reference explanation, $\boldsymbol{e}_r = \mathfrak{e}(\boldsymbol{u}_r)$, is s.t. $\boldsymbol{u}_r \vDash \theta$.

Since this metric depends on the choice of reference point, we are going to evaluate it on all the possible choices of reference points, and name that group *reference set*, $\mathcal{R}$. As anticipated, in the next chapter we present results for $\mathcal{S} = \mathcal{C}_A$ and $\mathcal{R} = \mathcal{C}_\theta$.

We use MD to check for specific patterns in explanations: for example, if the hypothesis' behaviour B is parameterized, we could check how median distance evolves with changes in the parameters of B. This analysis will be conducted for our experiment in Chapter 3, while in Chapter 4 we will discuss in more detail how to interpret similar results.
As a coherence measure, MD is useful, but it does not give a complete picture: ideally we would like to test also how *discriminative* the hypothesis is, i.e. how distinguishable are, in explanation space, the tuples that satisfy θ from those which do not.

Area under the curve

Contrary to MD, the goal of AUC is to measure how distinct is the behaviour of the model when presented with a data point which satisfies θ, compared to data points which do not satisfy θ.
This metric, together with the receiver operating characteristic curve (ROC), is usually chosen to evaluate the performance of a classifier. In our setting, this is relevant as the question on whether a tuple satisfies θ or not can be viewed as a binary classification problem, with positive labels for all $\boldsymbol{u}$ s.t. $\boldsymbol{u} \vDash \theta$, and negative labels otherwise. Then, given a $\boldsymbol{u}_i$, the distance of its explanation from the explanation of a θ-compliant tuples is interpreted as a score: the lower, the more probable it is that $\boldsymbol{u}_i \vDash \theta$. We use AUC rather than ROC because, unlike the second, AUC is cumulative and does not depend on the choice of classification threshold.
Notice that, as we are using distances instead of probabilities, increasing the classification threshold has the effect of increasing the true and false positives, while, normally, it decreases them. Thus, low classification thresholds will have low true positive rate and low false positive rates, while high thresholds will have high true positive rates and high false positive rate.

Definition 2.6 (AUC)**.** The *area under the curve* metric is defined as the area under the ROC curve.
Alternatively, for our case, given a set $\mathcal{S}$, it can be defined as the probability that the distance between the explanation of a tuple in $\mathcal{S}$ that satisfies θ and the reference explanation is smaller than the distance between the explanation of a tuple in $\mathcal{S}$ that does not satisfy θ and the reference explanation, i.e. $\mathfrak{d}(\boldsymbol{e}_r, \boldsymbol{e}_i) < \mathfrak{d}(\boldsymbol{e}_r, \boldsymbol{e}_{i'})$, with $\boldsymbol{e}_i$ the explanation of the tuple which satisfies θ, $\boldsymbol{e}_i = \mathfrak{e}(\boldsymbol{u}_i)$ and $\boldsymbol{u}_i \vDash \theta$, and $\boldsymbol{e}_{i'}$ the explanation of the tuple which does not, $\boldsymbol{e}_{i'} = \mathfrak{e}(\boldsymbol{u}_{i'})$ and $\boldsymbol{u}_{i'} \nvDash \theta$. Thus,

$$AUC = P\Big(\mathfrak{d}(\boldsymbol{e}_r, \boldsymbol{e}_i) < \mathfrak{d}(\boldsymbol{e}_r, \boldsymbol{e}_{i'}) \Big| \boldsymbol{u}_i \in \mathcal{C}_\theta, \boldsymbol{u}_{i'} \in \mathcal{S} \setminus \mathcal{C}_\theta\Big).$$

As usual, the more the ROC curve is distant from the diagonal, the better the model, and the bigger the area under it. For us, high AUC indicates that the distance between explanations is a good indicator of whether a tuple satisfies θ.[8] As for the median distance, we are firstly going to evaluate AUC for the tuples in $\mathcal{C}_A$, using as reference the explanations of $\boldsymbol{u} \in \mathcal{C}_\theta$. This will highlight if the model is following a specific behaviour on some A-compliant tuples, or if it is not.
In Section 4.2 we will also evaluate AUC on all $\mathcal{U}$; this will tell us if the explanations of the tuples that comply with the hypothesis are well distinguished from those that do not. A high AUC will then mean that the hypothesis characterizes a precise behaviour that the model follows when predicting all the θ-satisfying data points.

2.1.3 Step-by-step application procedure

Here we provide a step-by-step application procedure, showing how to apply semantic match, and outlining the key passages of the experiment in the next section.

Setup Firstly, one must have a (black-box) model $\mathfrak{f}$ which is already trained on some dataset $\mathcal{D}$ for solving a task.
Then, among feature-attribution methods, one explanation method $\mathfrak{e}$ needs to be chosen, based on the model architecture and the task for which the model was trained.
Once model and explanation method are fixed, the setup is complete. The explanations for each data point in the dataset are computed, and it is possible to start developing hypotheses and evaluating them against the explanations.

Hypotheses and distances The next step is to define some hypotheses, and an associated distance to check them against the obtained explanations. As we exemplify in the next section, both the hypothesis and the distance can be specified not only on the raw explanations obtained from $\mathfrak{e}$, but also on arbitrary functions of them, which enables us to define hypothesis on higher-level features. Indeed, in our experiment we will process the SHAP values of each token of the context to obtain a number proportional to the amount of contribution that a whole sentence has for the model's prediction.

Evaluation Then the set of $\boldsymbol{u}$ tuples satisfying the hypothesis must be individuated, to evaluate the match between the hypothesis and the explanations obtained from the model. As seen, the idea for median distance is to check how close to the explanation of a reference point in $\mathcal{C}_\theta$ are the explanations of tuples in $\mathcal{C}_A$. For AUC the intuition is to get a measure of how distinct the explanations of the data points that comply with θ are from those which do comply with the A-part, but do not satisfy B. Since the choice of reference can influence the results, we evaluate both metrics on each data point that satisfies the hypothesis. The result we obtain are thus box plots, which showcase how the score of each metric changes with the choice of reference point. The same process is iterated over all hypothesis and distances defined in the previous step.

Results Finally, comes the interpretation of the metrics and the assessment of the match between the hypothesis and the model's behaviour, as represented by the explanations.
The interpretation of MD and AUC scores can be quite different based on the way the hypothesis, the distance and the explanations are defined, and we address the interpretation of the metrics in Section 4.1.

2.2 Experimental setup

In this section we will provide an introduction to the experiment of the thesis, as well as some technical details on its setup and implementation.

[8] AUC has been criticized for multiple reasons, for an overview of some of its problems see [30] and for a discussion on the problems of AUC implementation and the relation between ROC and AUC see [38].

The main objective of the experiment is to verify if it is possible to apply the semantic match framework to successfully distinguish between a model that might be suffering from a specific bias, $\mathfrak{f}^b$, and a model which should not, $\mathfrak{f}^u$. After a first, general hypothesis, we formulate θ_2, to describe the bias we expect, and evaluate how well it aligns with the behaviours of $\mathfrak{f}^b$ and $\mathfrak{f}^u$. We hope to see a good semantic match for $\mathfrak{f}^b$, while $\mathfrak{f}^u$ would not match that hypothesis. We will also formulate θ_4, describing how we think a good model should behave and see if any of the models matches the idealized behaviour.
In order to do this, we are going to choose a dataset, and we are going to fine-tune two models on two different subsets of the original dataset, one without any apparent bias, the other with a bias. Then, we are going to formulate our hypotheses, evaluate how well each matches the behaviours of the models, and, finally, see if we can actually distinguish between the two models based on how well they match different hypotheses.

We opted for dataset with known and documented biases, hence, we operate in an 'idealized' environment where the biased and unbiased behaviours are manufactured and known a priori. This is due, in part, to the fact that the framework we employ is still in development, in part because the experiment is one of the firsts of this kind, and thus we preferred a controlled scenario where we already have clear expectations.
While previous experiments in Cinà et al. [11], tested semantic match on a tabular dataset and on different image datasets, here we focus on the text modality.
We rely on the results for the SQuAD dataset from Ko et al. [27], where the authors describe how it is possible to fine-tune different multipurpose NLP models to exhibit a bias. While in that article the main question is if it is possible to compensate those biases, and which techniques can achieve said goal, here we will use a biased model to see if it is possible to find hypotheses that describe and match its biased behaviour. This is as important as tools for debiasing: indeed, one needs firstly to individuate and know that there is a bias, to then act against it.

In the following sections we specify the task, detailing how the datasets were generated and how the training was conducted. We also describe how we processed SHAP values, and briefly comment on the implementation and complexity of explanation's distance, $\mathfrak{d}$.

2.2.1 Original dataset and Task

The SQuAD dataset, presented in [40], is centred around the question answering task (which is one of the main tasks of NLP). In particular, the task of SQuAD is *extractive*, as the challenge of the dataset is to answer a given question by correctly indicating the position of the answer in the context paragraph. The dataset is split in a training and a validation set, with around 88k and 10k entries respectively.
Each data point is constituted by an `id`, a `title`, a `context`, a `question` and the `answers`. The `id` and `title` fields are used to identify the data point and the general topic of the question. The `context` field contains the reference text, where the answer is located, while the `question` stores the open question that the model should answer. The label of each data point, contained in the `answers` field, is a dictionary consisting of a `text` key, with the text of the answer, and of a `answer_start` key, which is the index of the first character of the answer as it appears in the context. Extracts in the `context` field are passages of Wikipedia articles, while questions and labels were generated by humans reading them. The same context was reused to generate multiple questions.
Data points in the validation set have multiple correct answers, the entries at each of the dictionary's keys being substituted by lists.

Example 2.7. We display here a data point from the validation set of SQuAD.

`id`: 56be4db0acb8001400a502f0

`title`: Super_Bowl_50

`context`: Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24–10 to earn their third Super

Bowl title. The game was played on February 7, 2016, at Levi's Stadium in the San Francisco Bay Area at Santa Clara, California. As this was the 50th Super Bowl, the league emphasized the "golden anniversary" with various gold-themed initiatives, as well as temporarily suspending the tradition of naming each Super Bowl game with Roman numerals (under which the game would have been known as "Super Bowl L"), so that the logo could prominently feature the Arabic numerals 50.

`question`: What colour was used to emphasize the 50th anniversary of the Super Bowl?

`answers`: {`text`: ['gold', 'gold', 'gold'], `answer_start`: [488, 488, 521]}

The input of the model $\boldsymbol{x}$ is composed of the question and the context, which are passed through a tokenizer separated by the `[SEP]` token. The model bases its prediction on the list of tokens thus obtained.
The standard metrics of the dataset are f1 score and exact match. The latter is the percentage of predictions that is exactly equal to the (or one of the) provided answer(s). The former "measures the average overlap between the prediction and ground truth answer",[9] both taken as bags of tokens (i.e. treated as sets of tokens). When multiple answers are present, the maximum is taken, and then the average is computed across the dataset.

Since our objective is to train a biased and an unbiased model, instead of using the whole training set we generated two subsets to fine-tune our models on.

Dataset generation

Ko et al. [27] already provide a division of the original SQuAD training dataset into five groups based on answer's position: the first four sets have data points containing respectively only answers in the first, in the second, in the third or in the fourth sentence, while the fifth set has data points which have answers in the fifth sentence or later. These subsets were generated automatically by splitting the context into sentences with spaCy Sentencizer, [23].[10] The biggest set is the one with data points having answers in the first sentence of the context, containing approximately one third of the original SQuAD training set (precisely 28,263 data points).[11] Therefore, we use the first group for training the biased model, and will refer to it as $\mathcal{D}^b$.
In order to train a comparable unbiased model, we opted to generate a new dataset, of size similar to $\mathcal{D}^b$, but without its bias. Hence, we randomly sampled one data point from each of the five groups repeatedly, obtaining a dataset, $\mathcal{D}^u$, with a total of 28,260 data points, with an equal number of data points with answer in each sentence. We decided not to use the original SQuAD training set as, employing a dataset three times bigger as $\mathcal{D}^b$, might have given an unfair advantage to the resulting model.

2.2.2 Training

We fine-tune two BERT base uncased models on $\mathcal{D}^b$ and $\mathcal{D}^u$ respectively, in order to imitate the results from [27], and we fix the parameters of both models as follows: the maximum input length is 384 tokens and the stride is 128. The question and context joined by the `[SEP]` token are passed as a single string through the tokenizer of the model before the prediction begins. Then, the output of the tokenizer is split in the tokenized question and the tokenized context, which are fed to the model and result in an input with n features (each token is a feature). Since the question is the first part of the input, we allow truncation only in the second part (the context), as arbitrarily truncating the question might confuse the model, in turn leading to an incorrect prediction due to an incomplete question. Thus, if the context is too long, it is split into multiple inputs, each consisting of the full question and a part of the context.
The training is repeated for two epochs and the best model saved. On each epoch, the whole training dataset is

[9] See [40], p. 7.
[10] For further details, see [27], Section 2.1.
[11] The same bias is reflected in the validation set, which is also composed for one third of data points with answers in the first sentence.

consumed in batches of size 16, and we employ Adam optimizer with a learning rate of 0.00003.
We will refer to the model trained on $\mathcal{D}^b$ as *biased model*, $\mathfrak{f}^b$, and to the model trained on $\mathcal{D}^u$ as *unbiased model*, $\mathfrak{f}^u$.

Once the training is finished, we evaluate both models on the biased training dataset $\mathcal{D}^b$ and on the original SQuAD validation set, $\mathcal{D}^v$, consisting of 10,570 data points. The results are plotted in Figure 2.1. As expected from our setup, the biased model exhibits signs of a bias: on the training set, $\mathcal{D}^b$, the performance of $\mathfrak{f}^b$ is high, with exact match at 82.1 and f1 score of 91.5, while the performance of $\mathfrak{f}^u$ is slightly lower, loosing 3 points in both metrics. The high performance of $\mathfrak{f}^b$ is expected, as $\mathcal{D}^b$ was the dataset on which the model was trained. On the other hand, the good performance of $\mathfrak{f}^u$ suggests that it has learned to correctly focus on the first sentence if the answer is there.

The right plot shows that the biased model performs very poorly on the SQuAD validation set, $\mathcal{D}^v$. Exact

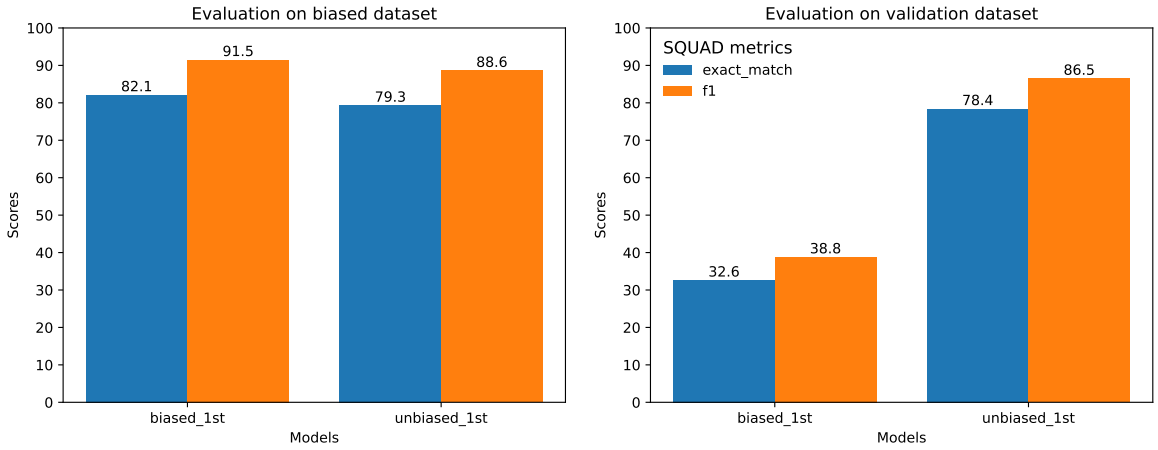

Figure 2.1: Evaluation results of the two models. On the right plot, exact match and f1 score on $\mathcal{D}^b$, in blue and orange respectively, for the biased model on the left and for the unbiased model on the right. On the left plot, exact match and f1 score on $\mathcal{D}^v$, on the left for $\mathfrak{f}^b$ and on the right for $\mathfrak{f}^u$.

match is now 32.6, while f1 score is just 38.8. On contrary, the performance of $\mathfrak{f}^u$ is comparable to the left plot, with exact match of 78.4 and f1 score of 86.5 (only one point less than on $\mathcal{D}^b$), which again is a good sign for the unbiased model, since, due to its high performance, we are prone to think that it has correctly learned to focus on the answer' sentence for its prediction. Oppositely, the poor performance of the biased model on the validation set, in conjunction with its good performance on the biased dataset, leads us to conclude that the model has not learned to generalize, and might always rely mostly on the first sentence for its predictions.
At this stage, the above conclusions are pure conjectures, and constitute a first intuition on how the two models are behaving. In the next chapter we will formalize these ideas in proper hypotheses and test them with the semantic match metrics.

2.2.3 SHAP values and sentence contribution

For the hypotheses of our experiment, rather than using the raw SHAP values, we will rely on *sentence contribution*, which we derive from SHAP values.
This is motivated, on one side, by the fact that the SHAP values library calculates the SHAP values not only for the actual prediction the model made, but also for all other possible predictions it could have made. This is out of the scope of our experiment, as at the current stage we are only trying to assess and evaluate the model's behaviour based on the prediction it produces.
On the other side, and more importantly, we are interested in hypotheses that talk about the relevance of each sentence of the context for the predictions of the two models, while SHAP values refer to each of the question's and context's tokens for every possible prediction.

In this section we detail how contribution is extrapolated from SHAP values. Intuitively, a high-level feature such as sentence contribution is worked out from explanations of single low-level feature by a mapping, which retrieves all the low-level features composing the high-level one, and some function which takes the mapped features as input and outputs a statistic based on those.

We calculate the SHAP values following the standard procedure reported in the documentation [34]. Accordingly, we pass each data point of $\mathcal{D}^v$ to the `shap.Explainer` that we initialize by inputting the tokenizer of the model and a function that specifies if to generate the *starting* or *ending* SHAP values. This difference is due to the fact that extractive question answering tasks require the model to evaluate both the starting and ending position of the answer, thus we obtain SHAP values for each of the two independent output distributions of the BERT model.
For each model we calculate both the starting and ending SHAP values. Since the number of tokens in each context is high, the evaluation of the SHAP values of a single data point is relatively slow; therefore, to save time and speed up the process, rather than generating the starting and ending SHAP values of both models for all the data points in $\mathcal{D}^v$ directly, we form batches of 500 data points and run the calculation of starting and ending SHAP values of each model on each batch in parallel.
We find a bug in the SHAP library, which throws a length mismatch error for some of the data points that contain a proper name in the question. Further investigation leads us to believe that it is caused by the masking function used to calculate SHAP values. As the causes of the problem are unclear, and the number of data points affected is minor, we decided to eliminate those points from the dataset. The full list of data points ids can be found in Appendix B. In the following, we will use $\mathcal{D}^v$ to refer to the set of data points from the SQuAD validation for which we obtained SHAP values correctly.

Once the SHAP values of all data points are generated, for each model, we obtain two dictionaries, one for the distribution of the starting index, one for the ending index. Both dictionaries contain `id`, `values`, `base_values` and `data` keys. The first provides the id of the data point as present in the SQuAD dataset, while the last is a list of $n+1$ words corresponding to the tokens of the question and the context, separated by a `[SEP]` token, and translated according to the model's tokenizer.
`base_values`, contains a NumPy array with $n+1$ entries, one for each token plus the separator between question and context. Each entry contains the SHAP value that would be assigned to each token in the input, assuming the model did not know any token of $\boldsymbol{x}$, see [35] and [34]. Intuitively, these values are a baseline to explain model predictions: the tokens which have greater base values (both highly positive or negative), will be those that the model attends the most on average and a priori. In other words, `base_values` is a prior on model outputs, taken as single tokens before assimilating the information in the question and context.[12]
`values` is a $(n+1)\times(n+1)$ NumPy array, containing the list of SHAP values for each token in the input. The jth list contains at the j'th entry the contribution that the j'th token of the input has on the output if the model were to output the jth token as the starting (ending) token.
Summing all the entries of the jth list in `values`, (that is $\sum_{j'=0}^{n}$ `values`$_{j,j'}$) one obtains the total SHAP values of the context, assuming the jth token as the starting (ending) token of the model's prediction.

As anticipated, in our calculation of the contribution we are only going to consider the SHAP values of the tokens that are part of the prediction. To better illustrate the procedure, we consider the data point of Example 2.7, where both $\mathfrak{f}^b$ and $\mathfrak{f}^u$ give a single token answer. The prediction of the former is incorrect, while the latter is correct. We plot in Figure 2.2 the unprocessed SHAP values, as obtained from the library.
We can already see that the highest SHAP values for $\mathfrak{f}^b$ come from tokens in the question and first sentence,

[12]We could also have studied if the two models consistently have high base values on specific tokens: in our case, it is possible that the tokens which most often appear in the first sentence will have higher base values for the biased model, compared to the unbiased one.
We decided not to inquiry in this direction as we suppose that this effect should be minimal, if the SQuAD dataset has no bias in terms of specific words always appearing in the first positions of the context. Such assumption is corroborated by the fact that contexts are typically long and usually contain multiple times, and in different positions, words important for the answer, or also contained in the question.

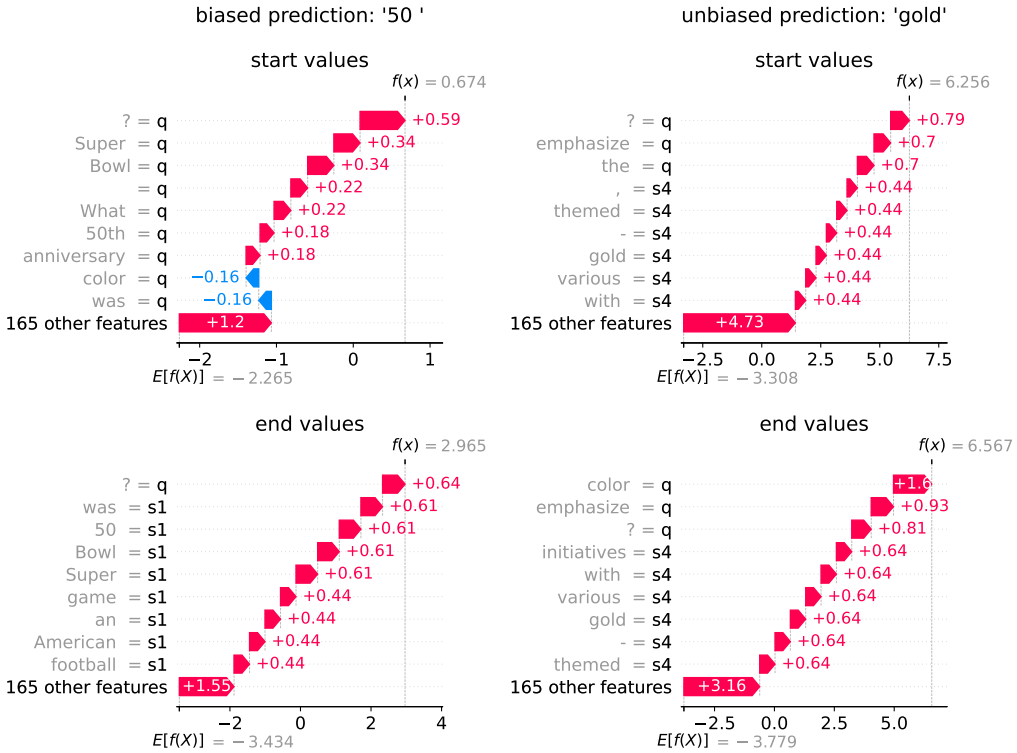

Figure 2.2: Visualization of starting (top row) and ending (bottom row) SHAP values for a single token prediction of the biased (left column) and unbiased (right column) models. The biased model predicts '50' while the unbiased model correctly predicts 'gold'. On the x-axes SHAP values are displayed in red if they are positive and in blue otherwise. On the y-axis the tokens (as translated by each model's tokenizer) are displayed in light gray, the black label indicates whether the token belongs to the question (q) or the ith sentence (si). The base value, $E[f(X)]$ and final value, $f(x)$ are also displayed on the x-axis.

while for $\mathfrak{f}^u$ the tokens are in the question and in the fourth sentence. We assume that for both models the question is going to be important for the prediction, and therefore we do not include it in the calculation of the contribution.[13] Furthermore, we assume that a positive as well as a negative SHAP value indicates that the model is attributing importance to that token, and, consequently, we consider the absolute value in our calculations.

Intuitively, sentence contribution is the sum of the absolute SHAP values of all the tokens in that sentence with respect to the answer. We thus need a mapping from tokens to sentences, which we obtain by splitting the context with spaCy Sentencizer, then joining the question and the obtained list of sentences with a `[SEP]`, and finally tokenizing the obtained string, with the model's tokenizer. Then, with the tokenized string we generate the mapping, which is a list of integers, with 0 for tokens in the question, l for tokens in the lth sentence and -1 for `[SEP]`. After, we remove all the added `[SEP]` except the one between question and context and the one after the last token.[14] Notice that a naive approach that checks, with a recursion on the sentences, if any of the words in the context occurs, and labels it accordingly would not work for two reasons: firstly, the same word might appear in different sentences, secondly, as not all words are present in the tokenizer dictionary, the number of tokens will not match the length of the mapping so produced. To define sentence contribution formally, we introduce a mapping function $\mathfrak{m}(l)$ s.t. $\mathbb{N} \to \{0, 1\}$ that, for each token in the input $\boldsymbol{x}$ outputs 1 if that token is in the lth sentence and 0 otherwise.[15]

[13] We tested the first hypothesis by also including the question's tokens in the calculation of the contribution and found similar results, which we report in Appendix D.

[14] The separator between question and context and the one at the end are considered by default by the SHAP library, which removes automatically the latter.

[15] We very generally took $\mathbb{N}$ as our domain, however NLP models have usually a finite vocabulary to translate words into tokens, using only positive integers for the latter. For instance, BERT's dictionary has 30,522 entries.

Definition 2.8 (sentence contribution)**.** Given a mapping from tokens to sentences $\mathfrak{m}(l)$, the lth *sentence contribution* to the single token prediction of model $\mathfrak{f}$ for data point d with explanation $\boldsymbol{e} = \mathfrak{e}(\boldsymbol{u})$, and $\boldsymbol{u} = \langle \mathfrak{f}, \boldsymbol{x}, y \rangle$ is defined as:

$$\mathfrak{c}_l(\boldsymbol{e}) = \Big(\frac{\sum_k \mathfrak{m}(l)|e_k^{start}|}{\sum_k |e_k^{start}|} + \frac{\sum_k \mathfrak{m}(l)|e_k^{end}|}{\sum_k |e_k^{end}|} \Big) \Big/ 2$$

if the prediction only has one token, with k ranging over $\{1, \ldots, K\}$, K being the total number of tokens in the tokenized context, and with $\boldsymbol{e}^{start}$ and $\boldsymbol{e}^{end}$ the lists of SHAP values with respect to the answer's token.[16] The formula is an average of two terms, representing respectively the starting and ending normalized contribution of the lth sentence. In both terms the norm is at the denominator, and is expressed as a sum of the absolute (starting or ending) SHAP values of all tokens in the context. The numerators are also sums ranging over all the tokens, but each $|e_k|$ term is multiplied by $\mathfrak{m}(l)$ which is 1 if the token k is in the lth sentence, and 0 otherwise, representing the total contribution of the lth sentence.
If the prediction has multiple tokens:

$$\mathfrak{c}_l(\boldsymbol{e}) = \underset{a \in \mathcal{A}}{avg} \left\{ \Big(\frac{\sum_k \mathfrak{m}(l)|e_{a,k}^{start}|}{\sum_k |e_{a,k}^{start}|} + \frac{\sum_k \mathfrak{m}(l)|e_{a,k}^{end}|}{\sum_k |e_{a,k}^{end}|} \Big) \Big/ 2 \right\}$$

with avg the average function across $\mathcal{A}$, which is the list of all tokens of the prediction, and with $\boldsymbol{e}_a^{start}$ and $\boldsymbol{e}_a^{end}$ the lists of SHAP values with respect to the ath answer's token.[17]

The lth sentence contribution is by definition in $[0, 1]$. To see this, notice that the two summands are normalized each by its own divisor, and are then averaged. The edge cases $\mathfrak{c}_l(\boldsymbol{e}) = 0$ and $\mathfrak{c}_l(\boldsymbol{e}) = 1$ are reached, respectively, when the model ignores the whole lth sentence for its prediction, and when the model only relies on the lth sentence for its prediction, while ignoring the rest of the context.
The two summands correspond to the percentage of contribution that the lth sentence has with respect to the starting and ending positions of the prediction, while $\mathfrak{c}_l(\boldsymbol{e})$ is the average of the two, thus giving a quantitative estimation of how important that sentence is for the model's prediction.[18] We close this section off by adding a plot similar to Figure 2.2, this time containing the contribution for each sentence of the context for that same data point.

2.2.4 Distance definition and implementation

We choose as distance for our hypotheses the absolute difference of sentence contribution between two points.

Definition 2.9 (Distance)**.** Given a sentence position l and two explanations $\boldsymbol{e}_i$ and $\boldsymbol{e}_{i'}$, the distance between the explanations (thought of as contribution values) is

$$\mathfrak{d}(\boldsymbol{e}_i, \boldsymbol{e}_{i'}) = |\mathfrak{c}_l(\boldsymbol{e}_i) - \mathfrak{c}_l(\boldsymbol{e}_{i'})|.$$

It is straightforward to check that this distance satisfies all the properties in Definition 2.4.

The above is motivated by the fact that we want to test the behaviour of the two models on sentences, which are high-level features. Since the number of tokens in different inputs can change, defining a distance that uses as inputs the raw SHAP values becomes not only troublesome, but also incorrect, as the differences between SHAP values vectors might introduce confounding factors, and score as more distant two explanations for which a model is placing similar contribution on the same sentence.

[16] We exclude the tokens of the question as both model consider it important, see Appendix D.
[17] Here, $\mathcal{A}$ is an ordered list rather than a set, as the same token might appear multiple times in the answer. Generally, the SHAP values of two different occurrences of the same token are different.
[18] In the codebase we include also another way to evaluate sentence contribution, namely, by finding the maximum of the two summands in the first equation. If the answer has multiple tokens, we then take the average, similar to the second equation of Definition 2.8.

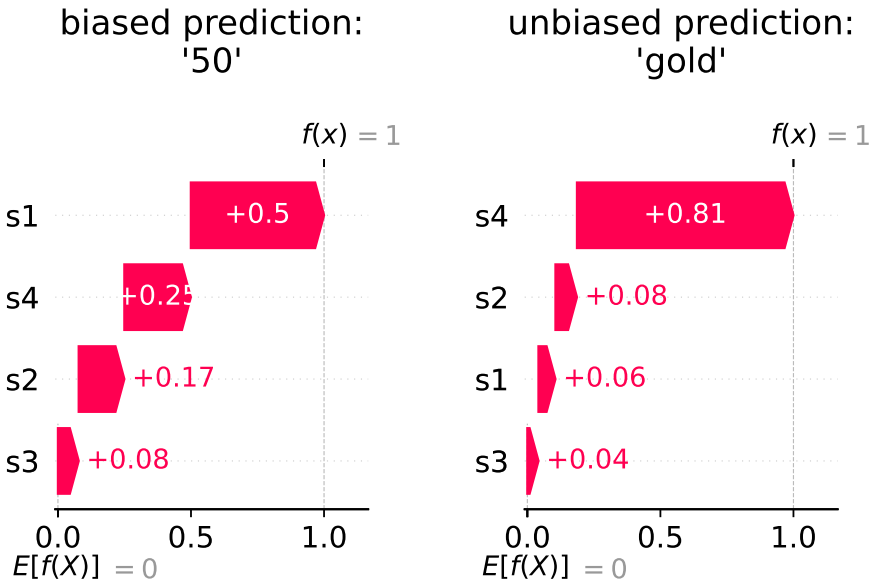

Figure 2.3: Contribution of each sentence towards the prediction, the biased model is on the right, while the unbiased on the left. Sentences are on the y-axis, while on the x-axis we plot the contribution of each sentence to the final prediction.
For the biased model the first sentence has half of the total contribution, while the same sentence for the unbiased model contributes less than 10%, and the fourth sentence, in which the answer is located has around 80% contribution.
The distance between the explanations of the biased (e_b) and unbiased (e_u) model, which we formalize in Definition 2.9, is for the first sentence $\mathfrak{d}(\mathfrak{c}_1(e_b), \mathfrak{c}_1(e_u)) = |0.5 - 0.06| = 0.44$.

As noted in Section 2.1.3, we need to evaluate $\mathfrak{d}$ by averaging over all possible reference points, that is, across all the data points which satisfy the hypothesis.
This means that computing all the distances needed for the evaluation of semantic match of any hypothesis, once each sentence contribution is calculated, is at least in $O(|\mathcal{S}| \cdot |\mathcal{R}|)$ and, at worse, $O(|\mathcal{D}|^2)$, with $|\mathcal{S}|$ the number of points in the sample set, $|\mathcal{R}|$ the number of points in the reference set and $|\mathcal{D}|$ the number of points in the dataset. When large datasets are used, or computational resources are scarce, one could avoid this problem by randomly sampling a subset of $\mathcal{R}$ and only averaging across it.
However, when using SHAP values with BERT model, this is less concerning, as the main computational bottleneck is the evaluation of the explanations themselves: although Lundberg et al. [36] show that there is an algorithm to efficiently compute exact SHAP values for models based on decision trees, in general, the exact computation of SHAP values is #P-complete, [9]. The algorithm implemented in the SHAP library avoids this simply by sampling only some subsets of features and skipping others, thus producing an approximation of the true value. Even so, due to the high number of features in NLP tasks, we estimated that calculating starting and ending SHAP values for both models on all data points of $\mathcal{D}^v$ would have taken several days in practice, if not ran in parallel; on the other hand, the computational overhead of evaluating the distances over all reference points amounts to just over an hour.

Chapter 3

Results

In this chapter we are going to illustrate and interpret the results of the experiment. In the first section we report the hypotheses we formulated and explain how to formalize them according to Definition 2.1, while in the second section we report median distance and AUC scores of each hypothesis and interpret the results.
Finally, we are going to introduce three further metrics, taken from Zhou, Ribeiro, and Shah [59], commenting on their evaluations and the additional information they provide.

3.1 Hypotheses definition

We set up the two datasets, $\mathcal{D}^b$ and $\mathcal{D}^u$, and fine-tuned the models $\mathfrak{f}^b$ and $\mathfrak{f}^u$ so that the first should exhibit a bias on the first sentence, while the second should not. After the training phase, we obtained a first confirmation of our expectations by evaluating the two models on the validation set of SQuAD, $\mathcal{D}^v$, see Figure 2.1. The generation of SHAP values for the predictions of both models on $\mathcal{D}^v$ and the inspection of those values for a specific data point, where $\mathfrak{f}^b$ is wrong while $\mathfrak{f}^u$ is correct, in Figure 2.2, provided further evidence of the suspected bias. However, this evidence is insufficient in two ways: firstly, it is based on the SHAP values of the tokens, which are low-level features, while the bias we are investigating concerns sentences, which are high-level features. This is problematic as it is possible that the observed bias concerns the specific tokens rather than their position in the context or their belonging to one sentence or another.
Secondly, and more importantly, our examination was restricted only to one data point, and indeed a very specific one, as it was chosen purposefully so that the predictions of both model were a single token, the unbiased model was correct and the biased model was incorrect. Thus, this analysis, though encouraging, is still incomplete.

The semantic match framework provides a fix to both limitations, allowing the description of behaviours concerning high-level features, and the possibility to test how common such behaviours are for a model. The goal of this section is to formulate different hypotheses that will help us distinguish $\mathfrak{f}^b$ from $\mathfrak{f}^u$: with the first two hypotheses we will try to capture the behaviour of the biased model, while to test our idea that a model should consider the answer' sentence important we define two additional hypotheses. Therefore, we expect $\mathfrak{f}^b$ to match the first two, and $\mathfrak{f}^u$ to only match the last two.

As noted, we expect $\mathfrak{f}^b$ to often rely a lot on the first sentence for its prediction, while we suppose $\mathfrak{f}^u$ not to do that. In general, we presume that an ideal model would rely more on the sentence where the correct answer is located and less on other sentences, rather than the opposite.
To make things clearer, we start with the following very general hypothesis:

$$\theta_1 = \text{'The contribution of the first sentence is} \geq z\% \text{ of the total contribution.'}$$

For a threshold z that could be tested at different values, and with *total contribution* intended as the sum of the contribution of each sentence of the context. We believe that this hypothesis will be matched by both

models for low z, while the higher z becomes, the less we expect $\mathfrak{f}^u$ to match it. However, θ_1 might introduce a confounding factor: data points which actually have the answer in the first sentence of the context will have high contributions on the first sentence for both models, and, indeed, we would hope that a model has high contribution there in similar cases.
Therefore, we formulate two additional hypotheses:

$$\theta_2 = \text{'If the answer is not in the first sentence, and the model is incorrect, then the contribution of the first sentence is } \geq z\% \text{ of the total contribution.'}$$

$$\theta_3 = \text{'If the answer is in the first sentence, and the model is correct, then the contribution of the first sentence is } \geq z\% \text{ of the total contribution.'}$$

If the biased model matches θ_2 for high z, then we could explain $\mathfrak{f}^b$ scarce performance on $\mathcal{D}^v$ with the fact that, when wrong, this model focuses a lot on the first sentence, instead of basing its prediction on the sentence where the correct answer is located.[1]
The unbiased model might also be wrong for the same fundamental reason, attributing too much importance to the wrong sentence, but we expect that it mistakes the first sentence for the answer's sentence less frequently.

On the other hand, θ_3 should be matched by both models for high z on all tuples as the A-part requires that the model is correct and the answer is in the first sentence. While helping us to understand if those tuples were confounding the results of θ_1, this hypothesis also provides further evidence to our initial assumption, namely, that an ideal model, when correct, should focus on the sentence of the answer.

Finally, to formalize our belief that reliable models should consider the answer's sentence very important for their prediction, we introduce our last hypothesis:

$$\theta_4 = \text{'If the model is correct, the contribution of answer' sentence is } \geq z\% \text{ of the total contribution.'}$$

For this hypothesis, we will redefine the distance as the absolute difference between the contribution of the answer' sentence of the $\boldsymbol{u}$ tuple considered, and the reference point's contribution of the answer' sentence.

To formalize the above hypotheses, we need to distinguish between the constraints on the data point, model and prediction (i.e. those on $\boldsymbol{u}$), from the description of the behaviour based on the explanations (i.e. the conditions on $\boldsymbol{e}$). The first will be formalized in the antecedent of the hypothesis, A, while the second will form the consequent B. We list in the next definition all the hypotheses discussed above, together with their formalization.

Definition 3.1 (Experiment's hypotheses)**.** Let $\mathfrak{f}$ be either the biased or the unbiased model, and $d = \langle \boldsymbol{x}, y \rangle$ a data point from $\mathcal{D}^v$, with $p = \mathfrak{f}(\boldsymbol{x})$ the prediction of the model, y the correct answer,[2] and $\boldsymbol{u} = \langle \mathfrak{f}, \boldsymbol{x}, y \rangle \in \mathcal{U}$; let $\boldsymbol{e} = \mathfrak{e}(\boldsymbol{u})$ be the SHAP values obtained for p of $\mathfrak{f}$ on d, and $\mathfrak{c}_l(\boldsymbol{e})$ the contribution of the lth sentence to the model's prediction, as per Definition 2.8. We will use $y \in s_l$ to denote the fact that the answer y is located in the lth sentence of the context, and $p = y$ to say that the model's prediction is correct. We list the hypotheses introduced above here, both in natural language and logically formalized.

$$\theta_1 \coloneqq \text{'The contribution of the first sentence is } \geq z\% \text{ of the total contribution.'}$$

$$\theta_1 \coloneqq \top \Rightarrow \mathfrak{c}_1(\boldsymbol{e}) \geq z$$

[1] This might also indicate that often the model predicts that the answer is part of the first sentence.
[2] The SQuAD validation has multiple correct answers, but in our algorithm we always refer to the first of the list. Thus, y refers here to the first answer, and not to the vector of accepted answers.

θ_2 :='If the answer is not in the first sentence, and the model is incorrect, then, the contribution of first the sentence is $\geq z\%$ of the total contribution.'

$$\theta_2 := \big(y \notin s_1 \wedge p \neq y\big) \Rightarrow \mathfrak{c}_1(\boldsymbol{e}) \geq z$$

θ_3 :='If the answer is in the first sentence, and the model is correct, then the contribution of first the sentence is $\geq z\%$ of the total contribution.'

$$\theta_3 := \big(y \in s_1 \wedge p = y\big) \Rightarrow \mathfrak{c}_1(\boldsymbol{e}) \geq z$$

θ_4 :='If the model is correct, the contribution of answer' sentence is $\geq z\%$ of the total contribution.'

$$\theta_4 := \big(p = y\big) \Rightarrow \mathfrak{c}_{s^*}(\boldsymbol{e}) \geq z$$

with s^* the sentence where the answer is found.

In synthesis, we expect the biased model to match θ_1, θ_3 for those $\boldsymbol{u}$ tuples which have the answer in the first sentence, θ_2 for most of the other tuples and to not match θ_4. On the other hand, we hope that the unbiased model matches θ_4 at high thresholds, and we expect also that θ_3 will exhibit a good match. We think that $\mathfrak{f}^u$ will not match θ_2, and that θ_1 might be matched for tuples which have the answer in the first sentence; if $\mathfrak{f}^u$ has learned to generalize, we also expect high MD for θ_1, reflecting the fact that sometimes the first sentence is very important, sometimes it is barely considered.

3.2 Hypotheses' results

In this section we analyse the box plots of MD and AUC for each of the hypotheses defined above.

3.2.1 Hypothesis 1

In Figure 3.1 we plot the median distance results of the two models for θ_1. Overall, both exhibit an increasing trend, which, is expected from a normal distribution of the first sentence contribution: the higher the threshold for contribution, the more specific the behaviour interval, $[z, 1]$, becomes; on one hand, this decreases the number of explanations that comply with it, and, on the other, it increases the distance between those and the ones which do not comply with θ_1.
However, notice two facts: (a) the biased model has the lowest median for $z = 50\%$; before that point the trend is actually slightly decreasing, and, only after, it turns upwards; (b) the unbiased model exhibits large variance except for the last two thresholds, which also have a very high distance.

The first fact suggests that the first sentence has always high contribution for the biased model. Indeed, while there are few outliers, more than half of the distances calculated when $z = 50\%$ are between 0.1 and 0.2, which means that for most explanations $\mathfrak{c}_1(\boldsymbol{e}) \in [0.3, 0.7]$, and often in $[0.4, 0.6]$. The highest extreme, at $z = 100\%$, further confirms this: the explanations with maximum contribution are, on average, just 0.4 away from the others, implying that a lot of $\mathfrak{f}^b$'s predictions receive around 60% of contribution from the first sentence. Furthermore, on low thresholds (i.e. between 10% and 50%), the distance between two explanations is also very low (frequently being between 0.1 and 0.2), suggesting that explanations of $\mathfrak{f}^b$ are clustered.

In addition, (b) suggests that the amount of contribution that the first sentence has for the unbiased model's prediction changes a lot, and ranges from low to high levels: at low thresholds, MD is low (though still higher than that of $\mathfrak{f}^b$), while at high thresholds (above 80%) MD is also high, meaning that most tuples have lower

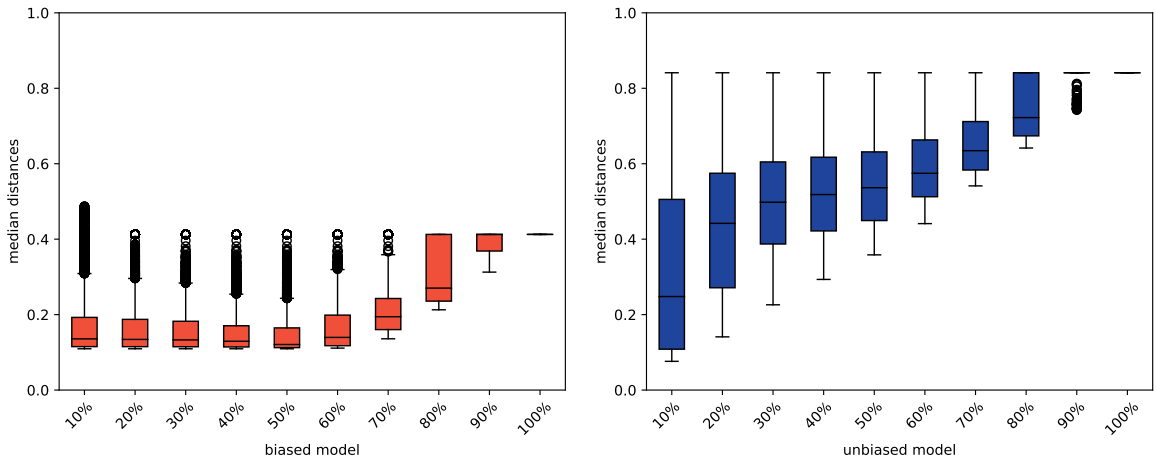

Figure 3.1: Median distance for θ_1. Results for the biased model are displayed in red on the left, while those for the unbiased one are on the right in blue.

contribution.

The unbiased model shows an upward trend, with the minimum MD reached at the lowest threshold. The average distance between two explanations is here around one fourth of the total contribution. Furthermore, the variance is very high, meaning that among the θ_1-compliant points, some explanations have contribution close to 10% while others reach contributions near to 100%. The upper whiskers are due to the latter group, and stay at the same level, as they satisfy B for all z. The lower whiskers show a linear upward trend, signalling that at each progression of z, some data points that were previously satisfying θ_1 are now not complying with it, thus supporting the idea that $\mathfrak{f}^u$ is giving varying amount of importance to the first sentence. A similar trend is also present in $\mathfrak{f}^b$'s plot, albeit starting much later.

In the next plot, we present the results for the AUC of both models. We expect very high AUC across all

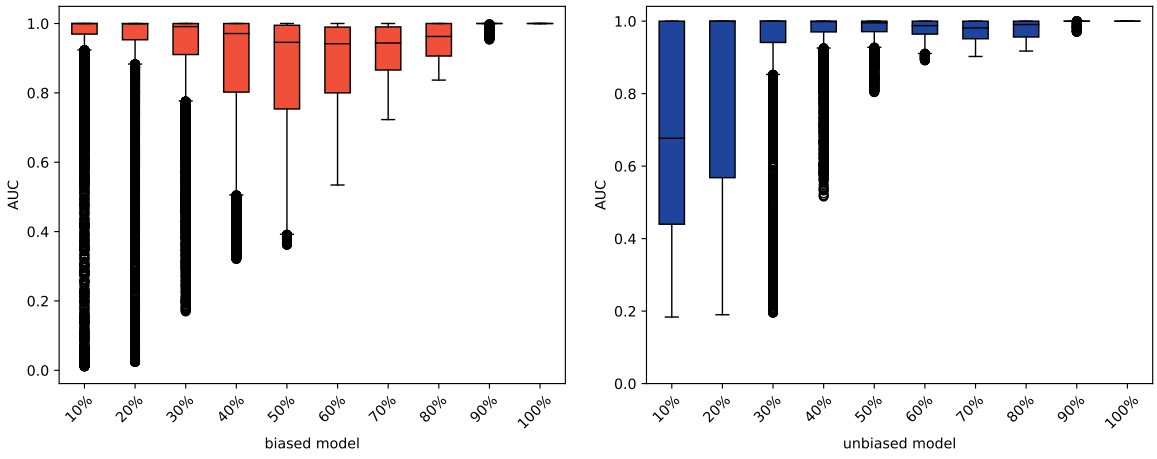

Figure 3.2: Area under the ROC curve for θ_1. Results for the biased model are displayed in red on the left, while those for $\mathfrak{f}^u$ are on the right, in blue.

choices of threshold: as hypothesis compliance and distance only depend on the amount of contribution the first sentence has, we expect the latter to be a good indicator of the former. Indeed, this is what we observe for both models.

However, there are important differences between $\mathfrak{f}^b$ and $\mathfrak{f}^u$: the variance of the unbiased model is very high, in particular for the lower thresholds. This is explained by the previous plot: since $\mathfrak{f}^u$ attributes at times a lot of contribution to the first sentence, and at times lower contribution, when an explanation with low contribution is chosen as reference, there will be explanations which have even lower contribution that are very

close to it, without satisfying θ_1; at the same time, there will be explanations with very high contribution on the first sentence, thus satisfying the hypothesis, that also have high distance from the reference. For that same threshold, when the reference point has a very high explanation value, there will be data points with much lower contribution from the first sentence at similar distances, some of which will comply with θ_1, while some others will not. This confounds the distance-based prediction, and lowers AUC.

On the contrary, $\mathfrak{f}^b$ reaches minimum AUC at $z = 60\%$; this is explained by the fact that most explanations carry approximately that amount of contribution (as concluded from MD), and therefore, an explanation with contribution just above 60% and one with contribution just below 60% will have very low distance, lowering AUC.

For both models and metrics, high z have low variance. This indicates that few tuples satisfy the hypothesis at those thresholds, as box plots exhibits how the change in reference explanation influences MD and AUC.[3]

In conclusion, θ_1 highlights a substantial difference between the two models: the contribution of the first sentence is often high for $\mathfrak{f}^b$, while, for $\mathfrak{f}^u$, it is sometimes very high and sometimes very low. Ideally, we hope that an optimal model would focus the most on the sentence where the answer is located, and therefore, the results above provide evidence in favour of our expectations: the unbiased model resembles, in its behaviour on the first sentence, the results we would hope from an optimal model, while $\mathfrak{f}^b$ seems to always rely a lot on the first sentence for its prediction, showcasing a bias. Of course, θ_1 alone does not prove that $\mathfrak{f}^u$ focuses on the first sentence exactly when the answer is there, and we will ascertain this only with the third hypothesis. For now, we focus on the bias and with the next hypothesis we are going to check if $\mathfrak{f}^b$ focuses on the first sentence even when it does not contain the answer.

3.2.2 Hypothesis 2

Median distance for θ_2 is shown in Figure 3.3. The plots are similar to those of the first hypothesis (Figure 3.1), however, there is no data point that satisfies θ_2 with thresholds above 80%. Thus, the bars for higher thresholds in the previous hypothesis originated from $\boldsymbol{u}$ tuples that either contained the answer in the first sentence or where the prediction was correct.[4] With respect to the previous hypothesis, both models have lower medians

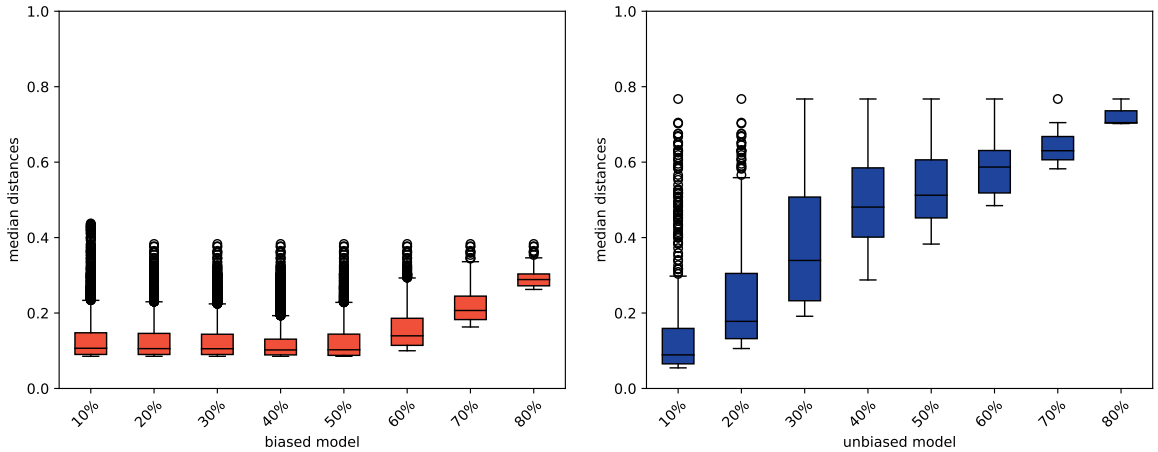

Figure 3.3: MD for θ_2. As previously, results for the biased model are displayed in red on the left, while those for the unbiased are on the right in blue.

and tighter quartiles, the latter indicating that $\mathcal{C}_\theta$ is smaller.

These results suggest that the unbiased model could match θ_2 at most at low thresholds (contrary to the previous hypothesis, which had higher median and very high variance on those zs), implying that if the answer is

[3]We conjecture that those data points are the ones where the answer is in the first sentence, and where the model is correct. Indeed, the second hypothesis will not showcase them, providing evidence to this intuition.

[4]While this does not imply that both disjuncts are true, we conjecture that it is indeed the case.

not in the first sentence and if the model is incorrect, then the importance that $\mathfrak{f}^u$ attributes to the first sentence is low. Thus, for the tuples that satisfy the hypothesis, the unbiased model is incorrect, but not because it is focusing systematically and too much on the first sentence.

On the other hand, the biased model has median distances slightly lower compared to θ_1, and shows the same trend, confirming our worries: often, if the answer is not in the first sentence and if the model is incorrect, $\mathfrak{f}^b$ focuses a lot on the first sentence, thus showcasing a biased behaviour. The rise in MD for higher thresholds (60% onwards) is very low compared to the increase of MD for the unbiased model, further corroborating our conclusions.

The results for AUC of the second hypothesis are shown in Figure 3.4. Both plots have similar scores then those of the first hypothesis.

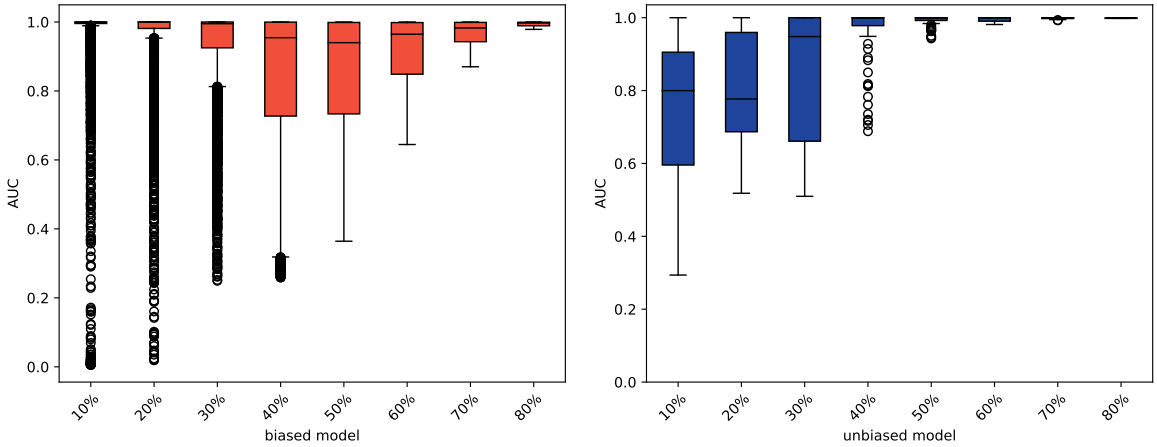

Figure 3.4: AUC for θ_2. As previously, results for the biased model are displayed in red on the left, while those for the unbiased are on the right in blue.

The unbiased model has median AUC always above 0.7, though with high variance for the lower thresholds. This can be explained by the fact that median distance is always pretty low for low zs, which in turn means that a hypothesis compliant data point might be very close to one which does not satisfy θ_2, while also being further from one that is compliant, thus lowering AUC.
For all thresholds above 30%, $\mathfrak{f}^u$ has almost perfect AUC, showing that, among the tuples that satisfy the A-part of θ_2, that is, tuples where the answer is not in the first sentence and the prediction is incorrect, contribution on the first sentence is a good predictor of θ_2-compliance. Then, the θ_2-compliant tuples stand out from those that only satisfy the A-part of the hypothesis, thus signalling that $\mathfrak{f}^u$ only rarely focuses a lot on the first sentence.[5]

On contrary, the biased model has high variance for $z = 40\%$ and $z = 50\%$, with high AUC for very low and very high thresholds (similarly to Figure 3.2). If for θ_1 this showed that the first sentence contribution to $\mathfrak{f}^b$'s prediction was often around half of the total contribution, the same result for θ_2 highlights that the biased model follows that behaviour even on tuples where it should focus on some other part of the context.

In conclusion, θ_2 is matched by the biased model, optimally for $z \in \{60\%, 70\%, 80\%\}$, as, for those thresholds we have both high AUC and reasonably low median distance. Conversely, the unbiased model does not match θ_2, as whenever the AUC is high, we also obtain high median distance, while at thresholds with low median distance the AUC score is also low.
The results confirm that the biased model is indeed following a biased behaviour, focusing too much and too often on the first sentence, and, in turn, making incorrect predictions.

[5]This will be confirmed from the perspective of other metrics in Section 3.3.2.

3.2.3 Hypothesis 3

The objective of the third hypothesis is to verify that a model attributes high contribution to the first sentence when the answer is located there and the prediction is correct. Semantic match with θ_3 (or lack thereof) will provide evidence for (or against) our idea that whenever the model is predicting correctly, it should focus primarily on the sentence that contains the answer.

Figure 3.5 illustrates the median distance for θ_3. The biased model exhibits the same trend of the previous

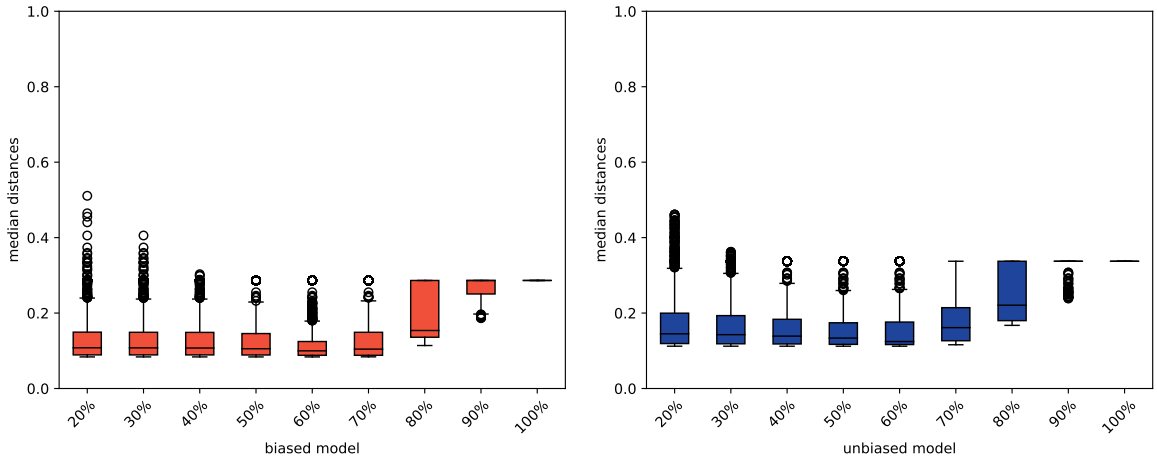

Figure 3.5: Median distance for θ_3

two hypotheses, albeit, now, the increase in median distance starts only later, at $z = 80\%$. The unbiased model also exhibits this behaviour, in contrast with the previous plots. This means that both models, when correct, attribute a lot of importance to the first sentence if the answer is there.
While we expected this from the biased model (due to the results for θ_1), the fact that also $\mathfrak{f}^b$ shows the same trend confirms our intuitions: when correct, a model should receive very high contribution from the sentence of the answer.

AUC, shown in Figure 3.6, is for both models similar, and resembles the plot of θ_1 for $\mathfrak{f}^b$ (Figure 3.2). Starting at maximum, the score reaches its minimum at $z = 70\%$ for both models, after which it recovers, again hitting 1.0 for $z \in \{90\%, 100\%\}$. The dip in AUC indicates that for both models the first sentence often has between

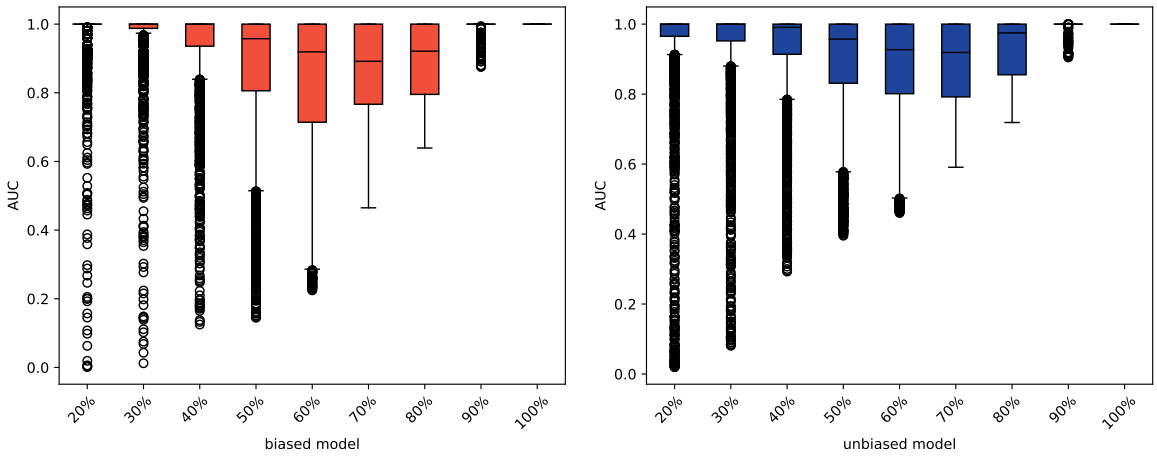

Figure 3.6: AUC for θ_3.

60% and 80% of contribution for the prediction, when the answer is in that sentence and the model is correct.

Even if the AUC dips, for all zs and for both models it always remains above 0.8. Since median distance is also low, we can conclude that θ_3 matches the behaviour of both models at high thresholds.

3.2.4 Hypothesis 4

Our final hypothesis is a generalization of θ_3: it checks whether the model focuses on the answer' sentence when its prediction is correct. As previously discussed, we expect this strategy to be the behaviour a good model should exhibit, and assessing semantic match with θ_4 is a way to test this assumption empirically for our two models.

For θ_4 we redefined distance as the absolute difference between the contribution of the answer' sentence (rather than between the first sentence contribution).

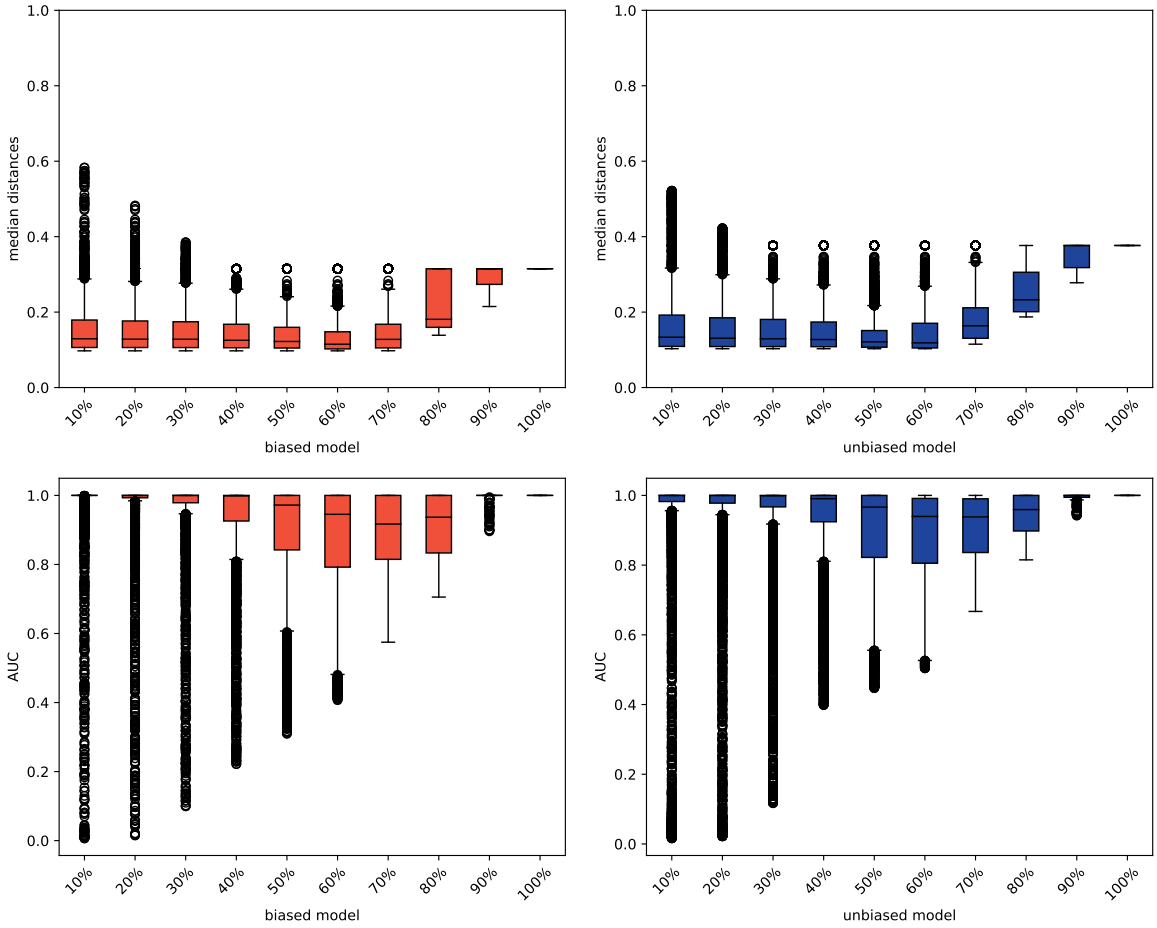

Figure 3.7: Median distance (first row) and AUC (second row) for θ_4.

The median distance and AUC plots, in Figure 3.7, are almost indistinguishable from the ones for θ_3 (Figure 3.6).

For the biased model, such result might at first look counterintuitive. However, the result is probably caused by θ_4 requiring the model prediction to be correct: in $\mathfrak{f}^b$'s case this almost always happens when the answer is in the first sentence. On those tuples the contribution of the first sentence is also very high (as we saw with θ_3), making distances and AUC higher.[6]

The fact that the unbiased model also shows the resemblance is more relevant: as $\mathfrak{f}^u$ is often correct when the answer is in other sentences,[7] low MD and high AUC provide confirmation to our intuitions about the way in which a good model should focus.

Both models have an almost flat median distance trend, slowly decreasing until the minimum, at $z = 60\%$; in turn, indicating that, when models are correct, the contribution of the answer's sentence is usually 60% or more. The highest threshold has distance 0.4, further confirming this. The plots for AUC reach minimum at 70%,

[6]The biased model is correct on 3,095 out of the 9,379 tuples in $\mathcal{U}$. Of those, 2,536 have answer in the first sentence, or 81.9%.
[7]The unbiased model is correct on 7,361 out of the 9,379 tuples in $\mathcal{U}$. Of those, 2,662 have answer in the first sentence, or 36.1%.

signalling that there are many tuples with contribution from the answer' sentence just between 60% and 70%.

To sum up, while for the biased model the fourth hypothesis is almost identical to the third, due to the fact that $\mathfrak{f}^b$ is almost never correct if the answer is not in the first sentence, the results for the unbiased model, together with its good performance, prove that it is behaving following the strategy we expected, i.e. focusing mostly on the sentence where the answer located in order to make the correct prediction.
As median distances are low and AUCs are high for both models, we can say that there is semantic match at high thresholds between the fourth hypothesis and the models. However, our metrics on θ_4 cannot alone distinguish the two models, because they do not provide insight on how many tuples actually satisfy the hypothesis, which is here key to make the distinction between biased and unbiased. In Section 3.3 we will introduce other three metrics, that will fill this gap.

3.2.5 Contribution histogram

Before closing the chapter, we provide the histogram of the first sentence contribution for the two models in Figure 3.8.
In accordance with the conclusions of our hypotheses, we find that the contribution of the first sentence for the

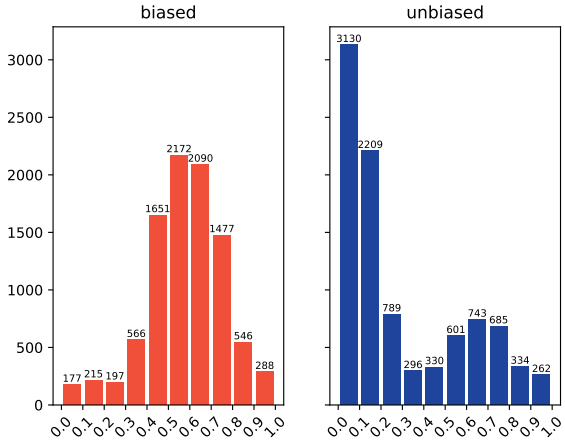

Figure 3.8: Frequency histogram for the first sentence contribution of biased (left, red) and unbiased (right, blue) models. Each column represents the number of data points in which the contribution of the first sentence is between z and the next threshold.

biased model is often between 40% and 80%, peaking between 50% and 60%. On contrary, for the unbiased model, the contribution of the same sentence is usually between 0% and 20%, with a smaller peak between 50% and 80%, which should correspond to the tuples that actually have the answer in the first sentence (the number of data points with answer in the first sentence is around 3,000, while the number of data points having at least 50% of contribution on that same sentence for $\mathfrak{f}^u$ is 2,625).

3.3 Other metrics: Coverage, Validity and Sharpness

In Zhou, Ribeiro, and Shah [59], authors measure the congruence of rules (hypotheses) with model's behaviours using three metrics: coverage, validity and sharpness. The way we defined hypotheses allows us to also apply those measures to our experiment directly.
In fact, we can translate their rules into hypotheses of the semantic match framework, but it is not always possible to translate our hypotheses into rules, as the following theorem demonstrates.

Proposition 1. Given a rule ρ as defined in [59], we can construct a hypothesis θ in our framework with the same semantics. The converse is not true.

Proof. The first direction follows directly by Definition 2.1 and by the definition of ρ: indeed, we use their applicability function as our constraint A, and their behaviour function as the B-part of θ. Rule composed of other rules are also recovered straightforwardly.
Translating in the other direction is not possible as rules of [59] can be defined only on low-level features, while we allow arbitrary functions to be applied to the explanations of low-level features to obtain explanations for high-level features. For a concrete counterexample, any of the hypotheses in Section 3.1 is not definable in the framework of [59]: sentences, which we are able to retrieve, are impossible to define only looking at their low-level constituents one by one. □

After defining the metrics from [59], we are going to comment the results obtained for our hypotheses. Later, in Section 4.3, we will compare them to MD and AUC of the semantic match framework highlighting the advantages and insights that each metric provides.

3.3.1 The metrics

Here we define the three metrics of Zhou, Ribeiro, and Shah [59]. As previously, we use $\mathcal{C}_A$ to indicate the set of tuples $\boldsymbol{u} = \langle \mathfrak{f}, \boldsymbol{x}, y \rangle$ that satisfy the constraint A of a hypothesis θ, $\mathcal{C}_B$ for the set of tuples that have explanations $\boldsymbol{e} = \mathfrak{e}(\boldsymbol{u})$ that comply with the behaviour B described by θ and $\mathcal{C}_\theta$ for the set of tuples that comply with the hypothesis, see Definition 2.3.

Definition 3.2 (Coverage)**.** Given a hypothesis θ, a dataset $\mathcal{D}$ and a model $\mathfrak{f}$, *coverage* is defined as the portion of $\boldsymbol{u} = \langle \mathfrak{f}, \boldsymbol{x}, y \rangle$ tuples that satisfies the A-part of θ. Formally:

$$cov = \frac{|\mathcal{C}_A|}{|\mathcal{D}|} = P(\boldsymbol{u} \vDash_A \theta)$$

with $|\mathcal{D}|$ the total number of data points of $\mathcal{D}$, and $\mathcal{C}_A$ the set of tuples satisfying A. This also means that we can think of coverage as the probability that a data point $(\boldsymbol{x}, y) \in \mathcal{D}$ will form with $\mathfrak{f}$ a tuple $\boldsymbol{u} = \langle \mathfrak{f}, \boldsymbol{x}, y \rangle$ s.t. $\boldsymbol{u}$ satisfies the A-part of θ.

Coverage measures the amount of data points that are characterized by the constraint A. In [59], it is used to establish how many data points the rule should explain. In our case it can be interpreted as a measure of how big the set of data points that we are characterizing with the antecedent of our hypothesis is.

Validity measures the fraction of A-compliant tuples where the model behaves as specified by B, therefore quantifying how often, on the tuples characterized by A, the model behaves according to the hypothesis.

Definition 3.3 (Validity)**.** Given a hypothesis θ, a dataset $\mathcal{D}$, a model $\mathfrak{f}$, an explanation method $\mathfrak{e}$, *validity* is defined as the probability that the explanation of a tuple $\boldsymbol{u} = \langle \mathfrak{f}, \boldsymbol{x}, y \rangle$ satisfies the B-part of θ, given that $\boldsymbol{u}$ satisfies the A-part,

$$val = \frac{|\mathcal{C}_\theta|}{|\mathcal{C}_A|} = P\big(\mathfrak{e}(\boldsymbol{u}) \vDash_B \theta \mid \boldsymbol{u} \vDash_A \theta\big)$$

with $\mathcal{C}_A$ and $\mathcal{C}_\theta$, respectively, the set of tuples that satisfy the A-part of θ and the set of tuples that satisfy the hypothesis.

A low validity indicates that the behaviour specified by B is not common across the set of A-compliant tuples, which in turn implies that either the model is behaving coherently, but in some other way, or that on $\mathcal{C}_A$ the model does not behave coherently at all. On the other hand, high validity implies that on the A-compliant tuples the model follows consistently the behaviour described in the hypothesis.

Similarly to validity, sharpness depends on the threshold chosen, and is a way to measure how necessary A-compliance is for B-compliance; in other terms, it tells us how often it is the case that when a tuple $\boldsymbol{u}$ has explanation that comply with B, then $\boldsymbol{u}$ also satisfies the constraint A.

Definition 3.4 (Sharpness)**.** Given a hypothesis θ, a dataset $\mathcal{D}$ and a model $\mathfrak{f}$, *sharpness* is defined as the probability that $\boldsymbol{u}$ satisfies the constraint imposed by A of θ, given that the explanation $\boldsymbol{e} = \mathfrak{e}(\boldsymbol{u})$ satisfies B.

$$sharp = \frac{|\mathcal{C}_\theta|}{|\mathcal{C}_B|} = P(\boldsymbol{u} \vDash_A \theta \mid \mathfrak{e}(\boldsymbol{u}) \vDash_B \theta)$$

3.3.2 Evaluation and insights

For each hypothesis, the results on the coverage metric are reported in Table 3.1, while in Figure 3.9, we include the results for validity and sharpness at different thresholds. As previously, we use $\mathcal{D}^v$ as dataset, restricting it by discarding all the data points for which we did not obtain valid SHAP values.

Hypothesis	Coverage	
	Biased	Unbiased
θ_1	1.0	1.0
θ_2	0.58	0.14
θ_3	0.27	0.28
θ_4	0.33	0.78

Table 3.1: Coverage of each hypothesis for biased and unbiased model.

As coverage depends only on the A-part of the hypothesis, it is not influenced by the choice of threshold. It has values in $[0, 1]$ reaching its maximum when every tuple in $\mathcal{U}$ satisfies A. In our case, this happens for θ_1 as it imposes no constraint on $\boldsymbol{u}$.

For the second hypothesis, the coverage of the biased model is almost 60%, suggesting that on more than half of the data points of $\mathcal{D}^v$, the answer is not in the first sentence, and the model is incorrect. As about 36% of data points in $\mathcal{D}^v$ have answers in the first sentence,[8] coverage can reach at most 0.64. Thus, $\mathfrak{f}^b$ is almost always incorrect when the answer is not in the first sentence. On contrary, the unbiased model has coverage of 14%, indicating that on that same portion of data points, it is less often wrong.

The third hypothesis has very similar coverage for both model. As A imposes that the data point has the answer in the first sentence, it cannot be higher than 36%, for the same reasons we gave for θ_2. Since A of θ_3 also requires the model to be correct (the prediction must match the answer exactly), the fact that coverage is very close to that threshold shows that both models are usually correct if the answer is in the first sentence.

Finally, θ_4 highlights the difference between the two models in terms of correctness: while $\mathfrak{f}^b$ prediction is correct about 1/3 of the times, the unbiased model has a much better performance, predicting the correct answer almost 80% of the times. For the fourth hypothesis, coverage provides the same information as the evaluation plots of Figure 2.1.

Validity and coverage, as explained in Zhou, Ribeiro, and Shah [59], should have an opposite trend: as z increases, the range of behaviours that satisfy the B-part of the hypothesis decreases; then, while the set of $\boldsymbol{u}$ tuples satisfying A, $\mathcal{C}_A$, stays fixed, $\mathcal{C}_B$, the set of tuples that has explanations that comply with B, decreases. Hence, validity, i.e. $|\mathcal{C}_A \cap \mathcal{C}_B| / |\mathcal{C}_A|$, is diminished, and sharpness, i.e. $|\mathcal{C}_A \cap \mathcal{C}_B| / |\mathcal{C}_B|$, increases.[9]

Hypothesis 1 For both models sharpness has a flat trend, which is always at its maximum, 1.0. As A of θ_1 does not impose any constraint, every $\boldsymbol{u}$ satisfies it, and therefore, $\mathcal{C}_A = \mathcal{D}^v$, $\mathcal{C}_A \cap \mathcal{C}_B = \mathcal{D}^v \cap \mathcal{C}_B = \mathcal{C}_B$, as $\mathcal{C}_B \subseteq \mathcal{D}^v$, hence $sharp = 1.0$.

Validity on the other hand, decreases as z increases, as expected. The biased model has a steep decrease for z between 40% and 70%, meaning that for most tuples, the first sentence has 40% to 70% of contribution to $\mathfrak{f}^b$'s

[8] Precisely, of the 9,379 data points for which we obtained SHAP values correctly, 3,359 (35.8%) have the answer in the first sentence of the context.

[9] If the hypothesis has a $\leq$ instead of a $\geq$, increasing z also increases the range of behaviours, leading to opposite trends: as the threshold increases, we expect validity to increase and sharpness to decrease, see Appendix F.

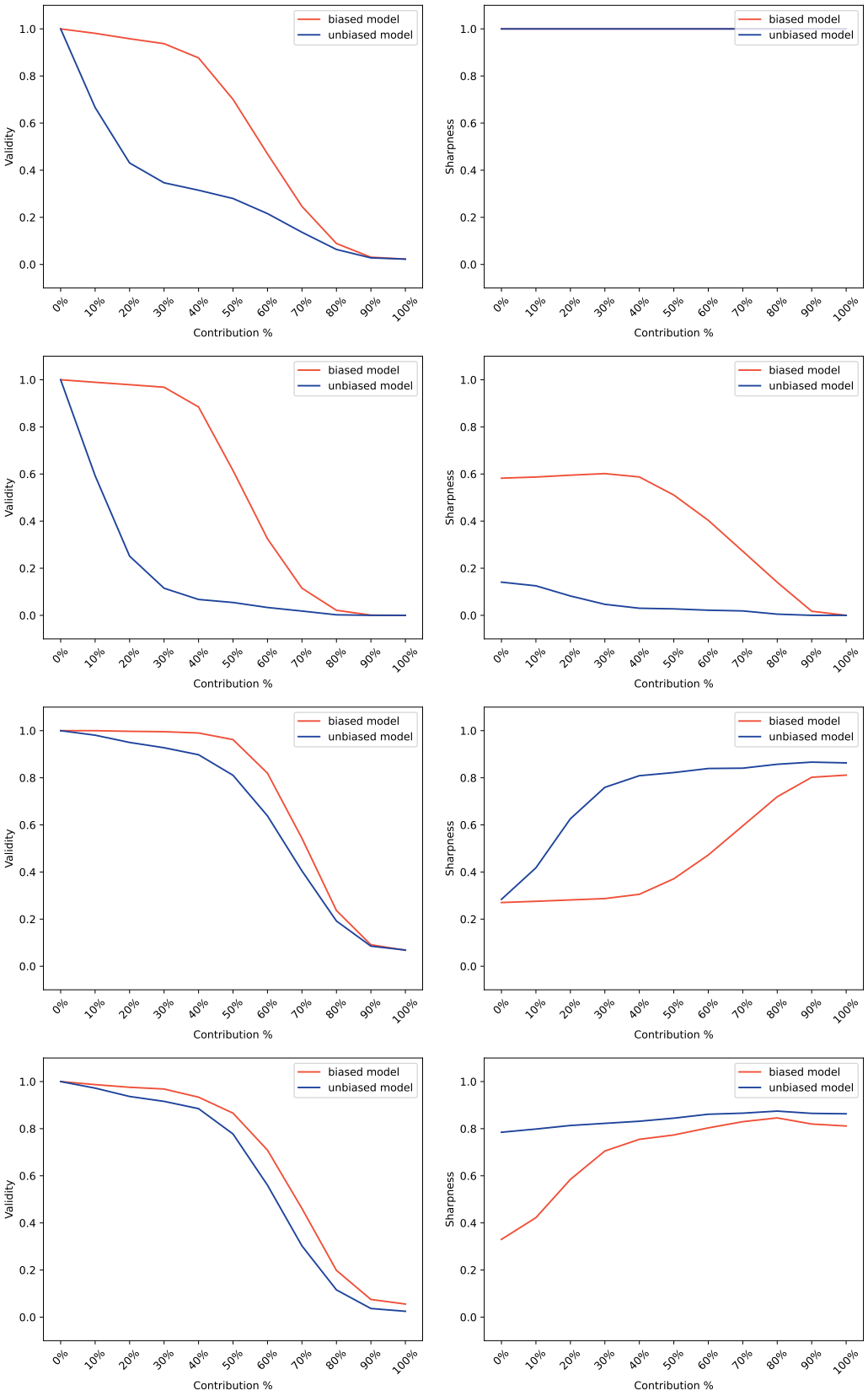

Figure 3.9: Validity (left column) and sharpness (right column) of the biased and unbiased model for θ_1, θ_2, θ_3 and θ_4, each in its respective row. We report thresholds for the B-part of each hypothesis on the x-axis and scores on the y-axis.

prediction.

The unbiased model has a strong decrease in validity between 0% and 30%, followed by a second gradual decline

between 50% and 80%. For $\mathfrak{f}^u$ then, most tuples have contribution of the first sentence between 0% and 30%, while a small group has first sentence contribution between 50% and 80%.
These results align with those of MD and AUC (see Section 3.2.1); however, for $\mathfrak{f}^u$ the second decline was less evident in the MD plot, and was signalled only by a small dip in the AUC between 50% and 80%.

Additionally, validity, gives us the percentage of data points that have first sentence contribution between two thresholds, by calculating the difference between validity at the first threshold and validity at the second threshold: for example, for the unbiased model, $val_{z=0.5} - val_{z=0.8} \approx 0.3$, meaning that for around 30% of the tuples the first sentence contributes between 50% and 80% to the prediction of $\mathfrak{f}^u$.

Hypothesis 2 The plots for θ_2 exhibit trends for validity similar to those of θ_1, while sharpness decreases with increase in z, contrary to the expected trade-off. This time, the biased model has an even stronger decrease between 40% and 70%, while, contrary to θ_1, the unbiased model has only one, sharper decrease between 0% and 20%. The first difference implies that if we only consider data points where the model is incorrect and the answer is not in the first sentence, for $\mathfrak{f}^b$, more tuples have first sentence contribution between 40% and 70%, rather than having it above 70%. The second difference, on the other hand, implies that, when the answer is not in the first sentence, and the model is wrong, $\mathfrak{f}^u$ considers the first sentence less important for its predictions.

Sharpness decreases for both models: while the biased model stays at 0.6 with a slightly increasing trend until 40%, where it begins to linearly decrease with z, the unbiased model starts at almost 0.2 decreasing to almost 0 before $z = 30\%$. Since θ_2 requires the model to be wrong (according to the exact match metric), sharpness at $z = 0\%$ is determined by the performance of each model on $\mathcal{D}^v$: from Figure 2.1 we know that the biased model is correct 32.6% of the times, while the unbiased model is correct 78.4% of the times, thus, even if B holds for all the tuples' explanations, $\mathcal{C}_B = \mathcal{D}^v$, we will have that sharpness is equal at most to $\frac{|\mathcal{C}_A \cap \mathcal{C}_B|}{|\mathcal{C}_B|} = \frac{|\mathcal{C}_A \cap \mathcal{D}^v|}{|\mathcal{D}^v|} = \frac{|\mathcal{C}_A|}{|\mathcal{D}^v|} \approx 1 - em(\mathfrak{f})$ with $em(\mathfrak{f})$ the model's exact match score on $\mathcal{D}^v$. The last equation is an approximation as $1 - em(\mathfrak{f})$ is bigger than sharpness, since the model might be wrong also on data points with answer in the first sentence, which are excluded by the A part of θ_2.[10]
The fact that sharpness decreases strongly for the biased model between 40% and 90% indicates that, if the contribution of the first sentence is very high (above 80%), it is rarely the case that the model is wrong and the answer is not in that sentence. Even so, the fact that the decline happens after 40% implies that a relevant portion of tuples where the model is incorrect has high first sentence contribution even if the answer is not there, in accordance with validity.
For $\mathfrak{f}^u$, the trend is also decreasing, starting from just below 0.2 and approaching 0.0 as z increases, showing that, when wrong, the unbiased model focuses less on the first sentence when answer is not there.
This analysis reinforces the conclusions of Section 3.2.2, adding details on the ratios of tuples that satisfy A and B at different thresholds.

Hypothesis 3 For the third hypothesis, the validity and sharpness exemplify the trade-off discussed in [59]: by increasing z, we make the behaviour more specific (as the contribution range $[z, 100]$, becomes smaller), decreasing validity and increasing sharpness. Since the A condition of θ_3 requires the answer to be in the first sentence and the model to be correct, this hypothesis is a control, and it should be matched by both models, as found in Section 3.2.3.

Both the biased and the unbiased models have similar trends for validity, starting high and relatively flat for low z, then falling steeply between 50% and 90%. This means that for both model the first sentence has often high contribution values if the answer is there and the model is correct.

The sharpness plot, however, highlights the difference between the two models: when it is high for a specific threshold z, it means that there are very few tuples that have $\geq z$ amount of first sentence contribution and

[10]Indeed, we find that $\mathfrak{f}^b$ is wrong on 823 data points, and $\mathfrak{f}^u$ is wrong on 697 data points out of the 3,359 data points of $\mathcal{D}^v$ having the answer in the first sentence.
The approximation is also due to the fact that sharpness is calculated among the tuples for which we obtained the SHAP values, while the model performance is checked on all $\mathcal{D}^v$.

that do not satisfy A. For $\mathfrak{f}^u$, the fact that the increase in sharpness is very sharp at low zs can be seen as further proof that, when either $\mathfrak{f}^u$ is incorrect or the answer is not in the first sentence, there are few tuples that receive very high (i.e. $\geq 40\%$) contribution from the first sentence. In other words, the unbiased model gives a lot of importance to the first sentence only for tuples that have the answer there. However, the fact that sharpness doesn't reach 1.0 also implies that there are few tuples that satisfy B at high thresholds, thus having high contribution on the first sentence, without satisfying A, or, in other words, that $\mathfrak{f}^u$ might be wrong because, sometimes, it looks too much at the first sentence, similarly, though less frequently, to $\mathfrak{f}^b$.
In contrast, the increase in sharpness for the biased model happens at much higher thresholds, confirming once again that it attributes a lot of importance to the first sentence regardless of the answer position or the correctness of its predictions.

Hypothesis 4 The results for validity are very similar to θ_3, even when generalizing the position of the answer. The sharpness plot for $\mathfrak{f}^b$ starts low, just above 0.3, and increases until 40% to over 0.7; then, it slowly increases until its maximum at 80%, after which it decreases again. For the unbiased model it is almost flat, starting just below 0.8, and slowly increasing to hit the maximum at $z = 80\%$.
The validity plot suggests that both model focus on the answer' sentence when they are correct. Sharpness for the unbiased model indicates that there are only few tuples where the model is incorrect and the sentence of the answer has high contribution. On contrary, for $\mathfrak{f}^b$, the low scores at low thresholds are evidence that there are a lot of tuples where the model is incorrect and the contribution of the answer sentence is low ($\leq 30\%$).

Chapter 4

Discussion

In this chapter we are going to discuss in more detail how to interpret the metrics introduced in Section 2.1.2 and see what they imply in some limit cases. Then, we are going to report the additional results of MD and AUC when evaluated on specific subsets of $\mathcal{D}^v$. Finally, we are going to compare the coverage, validity and sharpness metrics of Zhou, Ribeiro, and Shah [59] introduced in the previous chapter with ours.

4.1 Metrics interpretation

In general, from MD and AUC, we would like to draw insights on how the explanations are distributed, what is the relative frequency of explanations complying with B, and what kind of explanations are generated for tuples satisfying A. However, both metrics condense all that information in only one number per threshold and per reference point. Thus, we interpret here some common box plots trends they can form, and discuss their implications on the distribution of explanations.

To ease the task of understanding the metrics, our analysis will be conducted as a thought experiment, by assuming that the distribution of the explanations is uniform, and seeing how this reflects onto MD and AUC measures. For simplicity, we also assume that each explanation consists of a single number.[1]
The results obtained with this analysis can be compared with the trends of the experiment, to gain insights in the actual distributions of the experiment's explanations by highlighting similarities and differences between scores. We are also going to exemplify the discussed trends pointing at the results of the previous chapter.

Before we start, note that the expected plots will vary also based on how the threshold changes the behaviour range specified by θ: in our hypotheses we followed the schema '$\texttt{contrib} \geq z$' with `contrib` the relative contribution of some sentence (usually the first) and z the threshold; this scheme implicitly fixes the behaviour range at $[z, 1]$, and increasing z has the effect of restricting the range, making the behaviour more specific.[2]
Many more schemes could be envisioned, such as '$max(0, c - z) \leq \texttt{contrib} \leq min(1, c + z)$', with $c, z \in [0, 1]$, defining a progressively larger range of behaviours centred on c,[3] but they fall outside the scope of this work.

4.1.1 Median distance trends

Single box plot As we are evaluating the distances of tuples that comply with A from those that satisfy θ, at any fixed threshold, higher distances will generally indicate that the behaviour of the model is not consistent among the sample set, $\mathcal{C}_A$. This means either that the θ-compliant points are clustered and far away from the points which only satisfy the A-part of the hypothesis, or that the $\mathcal{C}_\theta$ points are far away from each other. The

[1] The restriction to a single number does not prevent us to investigate complex behaviours or biases, and indeed, in the experiment we captured the relative contribution of the first sentence with just one number.

[2] Defining hypothesis based on the scheme '$\texttt{contrib} \leq z$', with opposite sign, will have the opposite effect, increasing the range and making the behaviour less specific as z increases. We exemplify this in Appendix F.

[3] In the scheme, max and min are there to enforce the range being a subset of $[0, 1]$.

two cases can be distinguished by looking at the heights of the box plots: the former will have low variance, as $\mathcal{C}_\theta$ is clustered, and is exemplified in Figure 3.3, right, by the unbiased plot for $z = 70\%$ or $z = 80\%$ of θ_2; the latter will have wider quartiles and whiskers, and can be observed for $z = 10\%$ and $z = 20\%$ in the unbiased plot of θ_1 in Figure 3.1, right.

On contrary, if MD is low, points in $\mathcal{C}_A$ are all near each other, regardless of their B-compliance. The biased model exemplifies this in all the hypotheses and for most thresholds, as reported in Chapter 3. Low MD can be both positive and negative for assessing semantic match. On the negative side, having all the points in $\mathcal{C}_A$ clustered together might indicate that the hypothesis' behaviour is not distinctive enough, and refining B further might increase the distances. For instance, increasing z above 50% for the biased model in the first hypothesis, Figure 3.1 left, has the effect of restricting the range of behaviours and increases MD, leading to a clearer distinction between θ-compliant points and A-compliant-only points. But it is also possible that the distance chosen is not correct for the hypothesis we are testing.
On the positive side, a clustered $\mathcal{C}_A$ set is a good indicator of a possible semantic match, as it means that the explanations of the tuples in $\mathcal{C}_A$ are similar, and so, that the model is behaving consistently on that set. Points in $\mathcal{C}_A$ could then be well distinguished from tuples which do not comply with A, and this is indeed what is needed. If one wants to link specific features of the input data point (described by A) to peculiar behaviours of the model, one could make the behaviour more general, making $\mathcal{C}_A$ and $\mathcal{C}_\theta$ coincide.[4]

Trends As increases in z restrict the behaviour, explanations satisfying θ at high zs will be further away from those that satisfy the hypothesis only at lower thresholds. Hence, we expect MD to increase with z.
More precisely, assume that the distribution of explanations is uniform and has median m at 50%. Then, at threshold $z = 0\%$, all the tuples that comply with A will have explanations that satisfy B, and will be in the reference set. If we order the explanations in ascending order, m will be in the middle by definition, and the distances of $\mathcal{C}_A$ from $\mathcal{C}_\theta$ will be symmetric:

- tuples with lower explanations will have very few explanations close by, and thus few very low distances, a lot of explanations further away with higher distances (those around m), and finally some even higher distances, due to the reference points with higher explanations;
- tuples with higher explanations will generate distances similarly;
- tuples around m will have mostly lower distances, with higher ones due to the tuples with very low or very high explanations.

This is visualized in Figure 4.1. The medians will start high and decrease while approaching m to then grow again afterwards (keeping the ascending order of before). If we reorder the medians of distances in ascending order, as in Figure 4.2, increases in the threshold have the effect of eliminating the reference points with low explanations, in turn removing some higher medians of distances, at the right of the median of medians. This moves the median of medians leftwards, either decreasing it or keeping it equal.
Once z becomes bigger than m, more than half of the initial reference points are excluded; the remaining will have higher explanations values and higher distances from points in $\mathcal{C}_A$, so the median of medians will increase, reaching its maximum for the highest z.
Normal distributions also cause the same symmetry, making flat or slightly decreasing trends a good indicator of either normal or uniform distributions.

Hence, generally, MD can exhibit the following trends:

- Linear increases of MD after a specific z signal that the median of the explanation's distribution was located just before that threshold. For example, in the case of the first and second hypothesis, (Figure

[4] A clustered $\mathcal{C}_A$ is particularly useful when still in the process of generating a hypothesis and formalizing a behaviour. We are going to discuss and expand the comparison of $\mathcal{C}_A$ and the set of tuples which do not satisfy A in Section 4.2, by resampling MD and AUC using different $\mathcal{S}$ and $\mathcal{R}$.

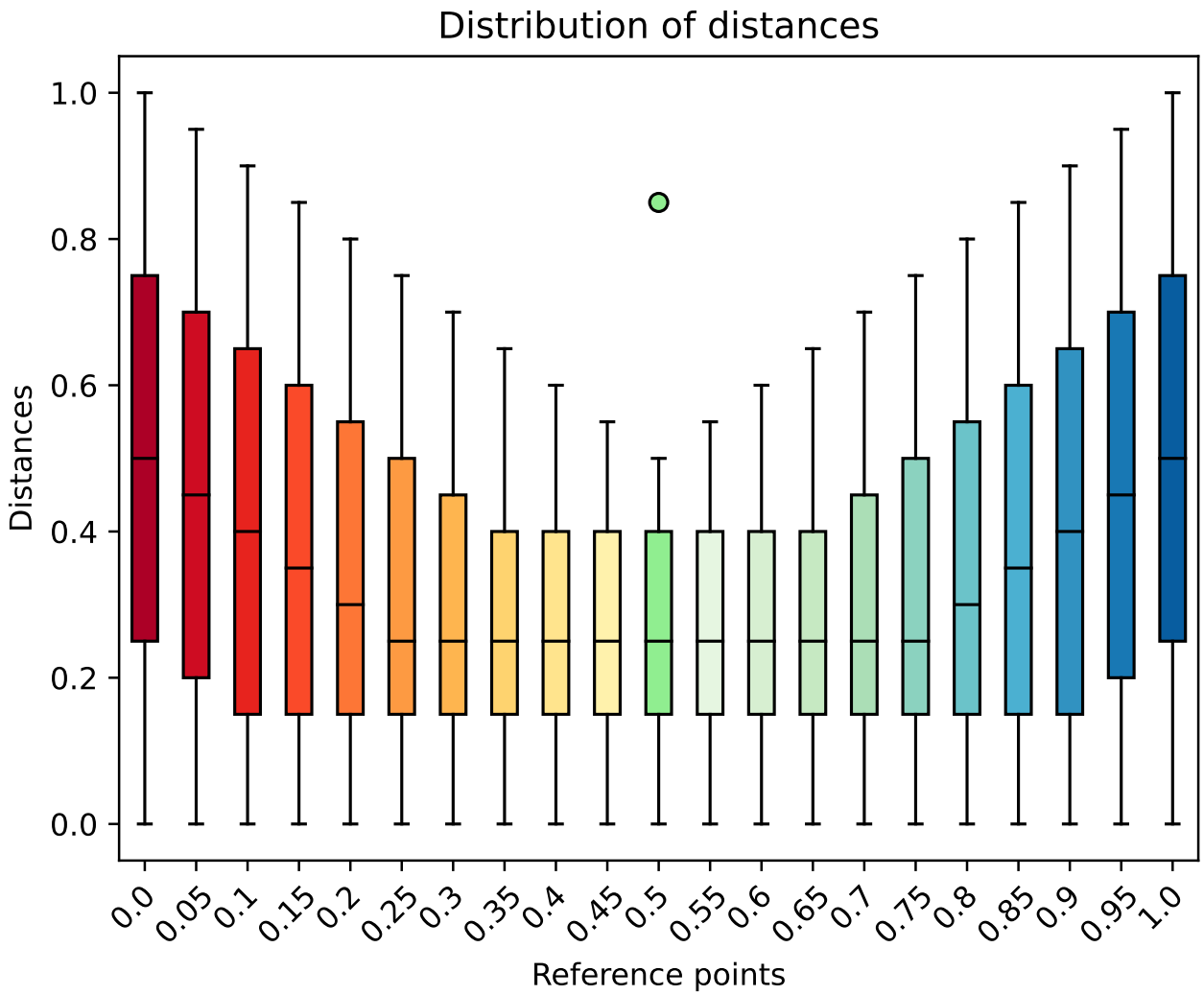

Figure 4.1: Box plots describing the distances of all the tuples in $\mathcal{S} = \mathcal{C}_A$ (y-axis) from each of the tuples in $\mathcal{R} = \mathcal{C}_\theta$ (x-axis). For $z = 0$, $\mathcal{S} = \mathcal{R}$; the box plots are symmetric and centred around the median, highlighted in light green and marked with a light green circle on top of the box plot. Gradients of colours at the left and right of the median showcase the order in which reference points in $\mathcal{C}_\theta$ will be excluded from the set as z increases: the leftmost, in red, will be the first to go while the rightmost, in blue, will remain in $\mathcal{C}_\theta$ even for $z = 1$.

3.1 and 3.3, left) the unbiased model showcases a linear trend across all thresholds, suggesting that the median is very low.

- Flat or slightly decreasing trends, indicate that the median of the explanations' distribution is not yet reached and indicate that the explanations' distribution is either uniform or normal, peaking at some higher threshold (the latter can be observed for $\mathfrak{f}^b$ in θ_1 and θ_2, Figure 3.1 and 3.3).

- Decreasing trends are not observable with our hypotheses. In general, the more $\mathcal{S}$ and $\mathcal{R}$ coincide, the more MD decreases. The scheme we used, however, restricts the behaviour as z increases, so the bigger z is, the less $\mathcal{R} = \mathcal{C}_\theta$ overlaps with $\mathcal{S} = \mathcal{C}_A$; in Appendix F we include an additional scheme, obtaining decreasing trends, which can be interpreted in a similar but mirrored manner as the first case above.

Whiskers and outliers can also provide useful insights: the reference point which is the furthest away will mark the upper whisker or the highest outlier, and, in our experiment, it is the one satisfying θ at the highest threshold. On contrary, the lowest outlier or the lower whisker comes from the reference point which is closest to the tuples of $\mathcal{C}_A$.
With a uniform distribution of explanations we expect the upper whiskers to be constant with respect to the

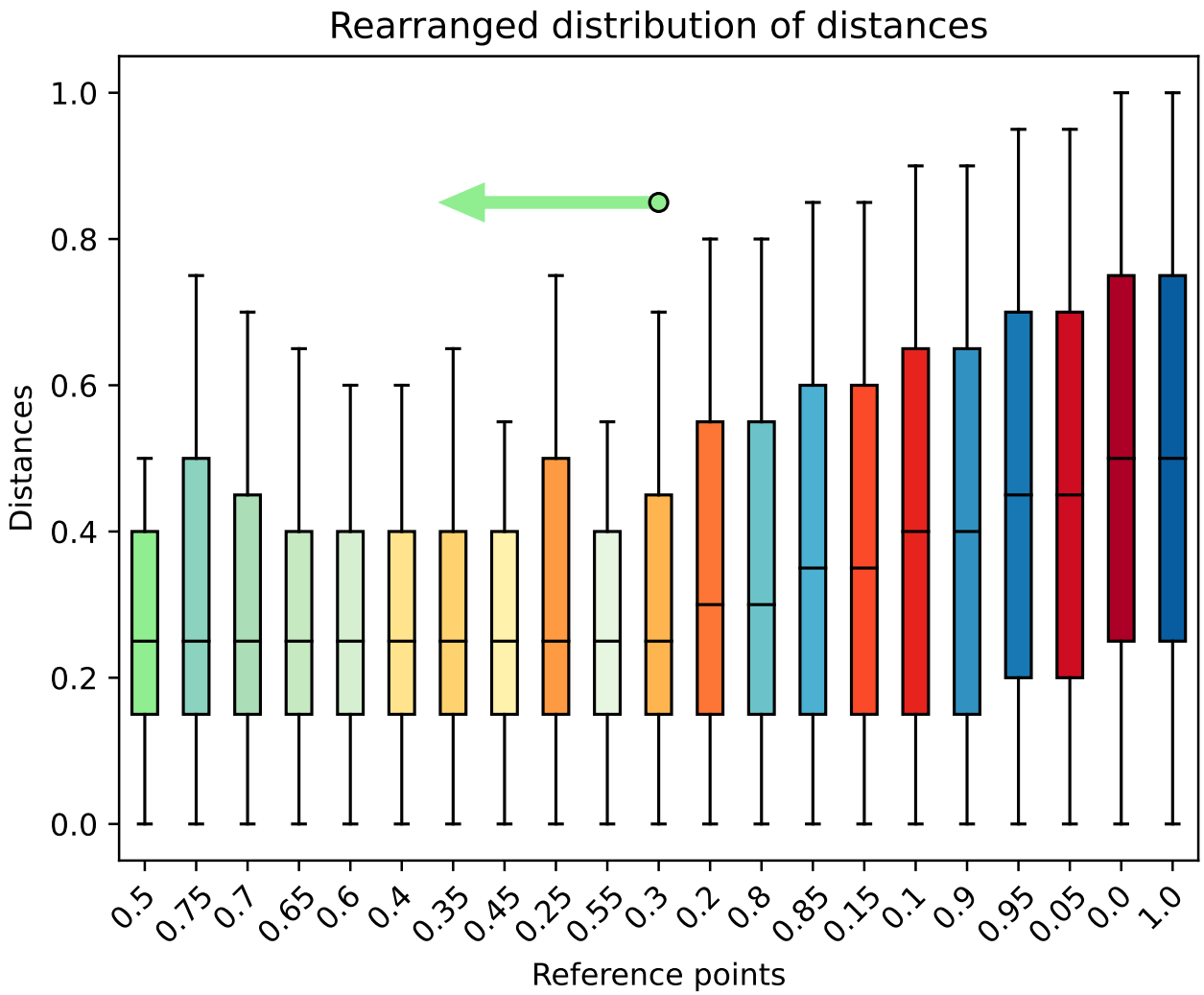

Figure 4.2: To find the median of medians, (marked as before with a light green circle), we rearranged the distances of Figure 4.1 in ascending order. As z increases, reference points in red will be removed. Because the median of distances is at the left of those points, once they are removed, it will move leftwards (light green arrow), forming a plateau or a further decreasing trend in the box plots for MD and AUC.

threshold, and the lower whiskers to increase linearly.[5]

A linear upward trend in the lower whiskers indicates that there are tuples that satisfy the B-part of θ at each specific threshold: this trend is exhibited by the unbiased model in Figure 3.1, right, implying that the model receives varying amounts of contribution from the first sentence. The biased model, instead, has a flat trend for the lower whiskers: increasing the threshold does not exclude points from $\mathcal{C}_\theta$, signalling that tuples in $\mathcal{C}_\theta$ always have high contribution from the first sentence.

In addition, asymmetries in quartiles and whiskers reveal that the explanations are amassed on a specific side of the median, and can be used to infer the shape of the explanations' distribution: if the first quartiles decreases while the second increases, as z increases, the explanations' distribution will be negatively skewed, with a quartile of the explanations concentrated just above the median, and the other quartile spread between a larger interval, with values lower than the median;[6] instead, if we observe the opposite, the distribution of explanations is positively skewed.[7]

[5] Changing the sign of the hypothesis' scheme inverts this, as shown in Appendix F.

[6] For instance, the biased model in Figure 3.1.

[7] This can be observed in the first few thresholds of $\mathfrak{f}^u$ in θ_1, Figure 3.1. Note that afterwards, at $z = 70\%$ and 80% the quartiles signal instead a negative skew, highlighting a second peak in the explanations' distribution, as can be further verified in the contribution histogram, Figure 3.8.

4.1.2 AUC trends

As described after Definition 2.6, AUC evaluates how distinguishable the set of tuples that comply with the hypothesis, $\mathcal{C}_\theta$, is from the set of tuples that comply only with the A-part of the hypothesis. Formally, we are interested in distinguishing $\mathcal{C}_\theta = \mathcal{C}_A \cap \mathcal{C}_B$ from $\mathcal{C}_A \cap \neg\mathcal{C}_B$.

Single box plot For a fixed z threshold, high AUC will mean that the distance between any given tuple $\boldsymbol{u}$ and a θ-compliant tuple is a good predictor of whether $\boldsymbol{u}$ is in $\mathcal{C}_\theta$ or not: if the distance is low, $\boldsymbol{u} \in \mathcal{C}_\theta$, otherwise $\boldsymbol{u} \notin \mathcal{C}_\theta$. Thus, $\mathcal{C}_\theta$ is clustered, and the distance between tuples in $\mathcal{C}_\theta$ is low, while the distances between tuples in $\mathcal{C}_A \cap \neg\mathcal{C}_B$ and in $\mathcal{C}_\theta$ is high. This is showcased by the AUC plot of the unbiased model for the fourth hypothesis at threshold $z = 80\%$, in Figure 3.7, bottom right.
On contrary, low AUC implies that the two sets are not well distinguishable: the explanations of tuples that comply with θ are similar to those of tuples that only satisfy the A-part of the hypothesis, and the distance between the two is small. Hence, predicting θ-compliance based on the distance alone will fail; the unbiased model exemplifies this for $z = 10\%$ in θ_1 (Figure 3.2, right), as most tuples contribute between 0% and 20%.

For a good semantic match we require high AUC, as this indicates that the θ-compliant tuples have similar explanations. Together with low MD, high AUC implies that across $\mathcal{C}_A$, only few tuples do not satisfy B, so $\mathcal{C}_A$ and $\mathcal{C}_\theta$ almost fully coincide.

Ideally, the smaller are quartiles and whiskers, the better, as we have a further indication that the explanations (and thus the model's behaviour) are consistent. Oppositely, if quartiles and whiskers are high, variance is high; if this happens, or if there are a lot of outliers (like in the biased model's plot of Figure 3.2 and 3.4) the behaviour described in B might be under specified and $\mathcal{C}_\theta$ might not be well clustered, implying that the model has an inconsistent behaviour. In similar cases, restricting B should reduce the number of θ-compliant tuples and return a $\mathcal{C}_\theta$ which is more clustered around a specific behaviour.

Trends In general, increases in z have the effect of shrinking $\mathcal{C}_\theta$, reducing the variance and the number of outliers. At higher thresholds, the distances between tuples in $\mathcal{C}_\theta$ and those not in $\mathcal{C}_\theta$ is higher, while the smaller range of behaviours allowed by B entails that the distances between data points in the reference set are smaller. As for MD, assuming a uniform distribution of explanations with median m at 50%, we have $\mathcal{C}_\theta = \mathcal{C}_A$ if $z = 0\%$, and AUC trivially at 1.0. Instead, for $z = 10\%$ AUC has great variance:

- for tuples in $\mathcal{C}_\theta$ with very high contribution, AUC is very high: having high distances from the tuples that only satisfy the constraint A, which have contribution lower than 10%, distance becomes a good predictor of membership in $\mathcal{C}_\theta$;
- tuples with very low contribution, but still in $\mathcal{C}_\theta$ will have low AUC, as they are very close to tuples that do not satisfy B, and very far from some tuples in $\mathcal{C}_\theta$;
- finally, tuples with contribution near to m will have AUC higher than 0.5, but lower than the tuples of the first case, as, by the symmetry of the distribution, they will have similar distances both from tuples in $\mathcal{C}_\theta$ with very high contribution, and from tuples not in $\mathcal{C}_\theta$ with very low contribution.

As z increases, tuples with very low contribution are excluded from $\mathcal{C}_\theta$. On one hand, this reduces the tuples falling in the second case above, and tightens lower whisker and quartile, decreasing the number of lower outliers too. On the other, while this improves AUC for tuples in $\mathcal{C}_\theta$ with very high contribution, it decreases AUC of those with lower contribution: with respect to each of those reference points, a bigger number of tuples with lower contribution, that now fall outside $\mathcal{C}_\theta$, will have similar distances to tuples with higher contribution still in $\mathcal{C}_\theta$.

Once $z = m$, tuples with contribution near m have the lowest AUC score, around 0.5. This is because, when calculating distances from those tuples, the number of tuples not in $\mathcal{C}_\theta$ at a fixed distance from the reference point is almost equal to the number of tuples in $\mathcal{C}_\theta$ at the same distance from that same reference point, by

the symmetry of the explanations' distribution. Thus, predicting based on the distance from those tuples will perform as well as randomly classifying tuples. The effect described in the previous paragraph is here maximum, and the median of AUC is at its lowest.
Further increases in z recover AUC, as the tuples with the highest contribution and almost perfect AUC are the majority of the reference points left in $\mathcal{C}_\theta$.

The above holds also for normal distribution and other symmetric distributions, so in such cases we can find the median by inspecting the median of AUC, together with its lower outliers/whiskers: if the median dips, while the whisker linearly increases with z, and for the deepest point in the dip the lower whisker or the lowest outlier is close to 0.5, then we have a strong indication that the median of the distribution of explanations is around that z. Note that the above is valid even for multimodal distributions, as shown by the small dip of AUC for $\mathfrak{f}^u$ in θ_1 for thresholds between 60% and 80% in Figure 3.2, right.

In summary, AUC can form the following trends relatively to increases in the threshold:

- typically, AUC increases with z. Indeed, since our hypotheses force higher and higher levels of sentence contribution as the threshold increases, $\mathcal{C}_\theta$ will become smaller and smaller, with tuples in it being further away from those only in $\mathcal{C}_A$. In turn, this eases the task of distinguishing θ-compliant tuples from those that only comply with A. Variance should also decrease, on one side because there are fewer tuples in $\mathcal{C}_\theta$, on the other, because the behaviour range is smaller, meaning that the reference explanations are clustered more closely.

- Dips signal that the distribution of the explanations peaks around the threshold at the lowest point of the dip: intuitively, when the explanation's distribution peaks around a threshold, there are a lot of tuples with explanations very close to tuples in $\mathcal{C}_\theta$, yet with contribution not high enough to satisfy B; then, predictions of θ-compliance based on the distance will be confounded, resulting in decreased AUC. As z increases, the peak is passed, and almost all tuples will fall outside $\mathcal{C}_\theta$; the remaining will have higher distances from the peak, making predictions of compliance easier, increasing AUC again, and drawing the end of the dip. This can be observed in Figure 3.6, left, for the AUC of $\mathfrak{f}^b$ in θ_3 between $z = 50\%$ and 80%.

- As with MD, linear decreases in AUC are harder to observe, but, contrary to the previous metric, this does not happen because of the way in which the hypothesis is specified: AUC is high both when the behaviour is very general or very restricted, albeit with different variance; hence a linearly decreasing AUC indicates that there might be some confounding factors in the chosen distance.

4.1.3 Limit cases

We discuss here the possible limit cases that can rise from our measures.

High MD, low AUC This is the worst scenario, as it implies that the distance is a very poor indicator of θ-compliance and that the tuples in $\mathcal{C}_\theta$ are mixed together with tuples that only satisfy A. It also means that in terms of the defined distance and hypothesis, the model is not following the described behaviour. In similar cases, revising the definition of distance, or strengthening the constraints of A and B might improve the results.

High MD, high AUC Here explanations from $\mathcal{C}_\theta$ are well clustered and, in explanation space, located quite far from the tuples that only satisfy A. High AUC indicates that the model is behaving in a peculiar and uniform way on the tuples in $\mathcal{C}_\theta$, but the fact that MD is high also signals that the model is not consistently behaving as specified by B on all the tuples that satisfy A. If the desideratum is that the model follows a specific behaviour on a specific set of tuples, high MD signals that this is not happening. One could then try to characterize A more specifically, in order to exclude the tuples of $\mathcal{C}_A$ that did not comply with B.
As we shall see in Section 4.2, if we change the sample set to include all the tuples, and use as reference $\mathcal{C}_A$, this scenario indicates a good separability between $\mathcal{C}_A$ and all the tuples which do not satisfy the A-part of θ. The

model is thus sensitive to $\mathcal{C}_A$-membership, showcasing a different behaviour if a tuple complies with A or not. This situation can be leveraged by fine-tuning the B-part of the hypothesis to achieve a full semantic match between the model's behaviour and a user-interpretable hypothesis.

Low MD, low AUC Cases of low scores for both metrics signal that the $\mathcal{C}_A$ set is clustered, but predicting only based on the distance is challenging. This might be due to the fact that some explanations that comply with B are very close to explanations which do not. In turn, this is possibly caused by a too restrictive hypothesis' behaviour, but it might as well be that the chosen distance is not sensitive enough to differences in the explanations, or that it includes confounding factors.

Low MD, high AUC These results are exactly what we require for a semantic match between θ and the behaviour of the model: the former guarantees that all the explanations of tuples in $\mathcal{C}_A$ are similar (i.e. that $\mathcal{C}_A$ is clustered), while under this condition the latter assures that $\mathcal{C}_A$ and $\mathcal{C}_\theta$ overlap. Then, it is possible to conclude that the model is behaving uniformly as specified by the hypothesis on the set of tuples that the constraint A characterizes.

4.2 Same metrics, different sets

As anticipated in Section 2.1.2 we are going to provide further evaluations of MD and AUC. Recall from Definition 2.5 and 2.6, that both metrics are calculated by working out the distance of all the tuples in a set $\mathcal{S}$ with respect to a reference point chosen from the reference set $\mathcal{R}$. Previously, we set $\mathcal{S} = \mathcal{C}_A$ and $\mathcal{R} = \mathcal{C}_\theta$ to check how consistent was the behaviour expressed by a hypothesis θ across the set of tuples characterized by the constraint A. Instead, we are now going to evaluate MD and AUC by fixing $\mathcal{S} = \mathcal{U}$ and $\mathcal{R} = \mathcal{C}_A$. As we shall see, this will provide a behaviour-independent measure of the separability of the A-compliant tuples from those which do not satisfy A. This can be useful as a preliminary study of the model's workings, before defining a specific B.
Furthermore, evaluating the metrics with $\mathcal{S} = \mathcal{U}$ and $\mathcal{R} = \mathcal{C}_\theta$, should give us more information about how the model behaves on tuples that do not satisfy A, and if there is a distinction between the behaviour applied to tuples in $\mathcal{C}_\theta$ and the one applied to tuples that do not comply with θ. The results we obtain for each of the for hypotheses of Definition 3.1 are added in Appendix E, as they confirm and substantially overlap with those presented in Chapter 3.

Note that, whenever the constraint A is empty, as in the first hypothesis, recalculating the metrics with $\mathcal{S} = \mathcal{U}$ and $\mathcal{R} = \mathcal{C}_\theta$ does not change the results, as $\mathcal{U} = \mathcal{C}_A$, while, for the same equivalence, evaluating AUC and MD with $\mathcal{S} = \mathcal{U}$ and $\mathcal{R} = \mathcal{C}_A$ is not informative.

$\mathcal{U}$ from $\mathcal{C}_A$

So far, we applied semantic match in a controlled setting, using it to distinguish between two models that were purposefully trained, one to incorporate a specific bias, and the other, hopefully, to not showcase it. This confined environment was chosen first and foremost to prove that the framework could be successfully applied on a state-of-the-art model trained on a real world dataset.
However, in less controlled settings, with less information about possible biases and problems of the model, coming up with a specification of the hypothesis and the right distance to test it might be significantly more challenging. In such cases, knowing if a group of tuples with some common characteristics gives rise to similar explanations according to the defined distance is very useful.

We can inquire this by employing the semantic match metrics and evaluating them with a different reference set. We will use the A-part of the hypothesis to properly characterize the set of tuples we are interested in, and then, we check with MD and AUC how well it is possible to distinguish that group from all the other tuples (as usual based on the relative distance of the obtained explanations). If all tuples in $\mathcal{C}_A$ are close to each other

and far from the tuples which do not satisfy A, then we have evidence that the model is recognizing $\mathcal{C}_A$ and is behaving on it in a specific way. Instead, if we obtain similar explanations from both sets, we have grounds to believe that either the distance is not defined correctly, or the model is not distinguishing the $\mathcal{C}_A$ set from the other tuples.

Example 4.1 (Weapon discriminator)**.** For instance, suppose we have a model to predict if one luggage contains forbidden articles, based on the x-ray images of an airport scanner. Among all prohibited items, we are particularly concerned that the model might not recognize weapons correctly. Then, we can define the A-part of our hypothesis as 'the luggage contains a weapon', leaving the specification of the exact behaviour of the model empty (i.e. $B \coloneqq \top$).[8] Finally, we need to define a distance, which could be a function measuring the amount of contribution that the pixels of the weapon have (if there is no weapon it will be 0). Once this is set up, we can evaluate MD and AUC using as sample set all the tuples and as reference set $\mathcal{C}_A$.
If the model actually treats the images containing weapons in a specific way, then the distance between those and the other images should be larger, leading both to higher MD and AUC. Furthermore, if the model behaves coherently on $\mathcal{C}_A$, we should also observe low variance, with thigh quartiles and whiskers.

In our setting, we suspect that one model has a bias on the first sentence while the other does not. Under this conjecture, we could use the simple distance of Definition 2.9 to see if it is possible to distinguish the tuples which have the answer in the first sentence from those which do not, based on the difference in contribution that said sentence provides. For an unbiased model, those tuples should be distinguishable, as the model should consider the first sentence very important for its predictions. A model biased on the first sentence, instead, will not be able to distinguish $\mathcal{C}_A$, as it would always focus on that sentence. To this end, we define a fifth hypothesis:

$$\theta_5 \coloneqq \text{'The answer is in the first sentence.'}$$

$$\theta_5 \coloneqq y \in s_1 \Rightarrow \top$$

Where, as previously $y \in s_1$ denotes the answer y being in the first sentence, s_1.
In Figure 4.3 we provide the plots of MD and AUC, with the addition of a distance density plot in Figure 4.4. The median distance of the biased model is low, with few outliers, tight quartiles and median below 0.2, implying that this model is treating all the tuples similarly, regardless of their membership to $\mathcal{C}_A$. Additionally, AUC is low when compared to the unbiased model, further confirming that it is hard to classify A-compliance based on the distance of the explanations obtained for $\mathfrak{f}^b$; the fact that AUC has a skewed distribution and very wide lower quartile and whisker further highlights that some tuples which do not have the answer in the first sentence receive the same amount of contribution from that sentence as some reference points.
Oppositely, the unbiased model has quite high median distance, with median of medians around 0.5 and symmetric quartiles and whiskers, indicating that the behaviour of $\mathfrak{f}^u$ on $\mathcal{U}$ varies. The AUC is very high, with median of medians almost at 1.0, and a negative skew, with a long tail of outliers. This implies that $\mathcal{C}_A$ is easily distinguishable if we base our predictions on the amount of contribution that the first sentence provides.
In figure 4.4 we add the distance density plots, which showcase the measured distances on the x-axis and the frequency on the y-axis. Tuples which satisfy A are labelled in blue, while tuples that do not are labelled in orange. The difference is very clear: while the two sets are almost fully overlapped for the biased model, the unbiased model has two distinct peaks, with far less overlap between the two groups. Notice also that the tuples not satisfying A have usually larger distances from tuples in $\mathcal{C}_A$. This suggests that if the answer is in the first sentence, the model is attributing a lot of importance to that sentence, while if the answer is not there $\mathfrak{f}^u$ is attending some other sentence.

To conclude, re-evaluating the semantic match metrics with $\mathcal{S} = \mathcal{U}$ and $\mathcal{R} = \mathcal{C}_A$, we can check if the model is behaving on tuples of $\mathcal{C}_A$ in a peculiar manner. Such analysis requires no specific definition of B, and thus, two

[8] Note that in this case $\mathcal{C}_A = \mathcal{C}_\theta$, hence $\mathcal{R} = \mathcal{C}_A = \mathcal{C}_\theta$.

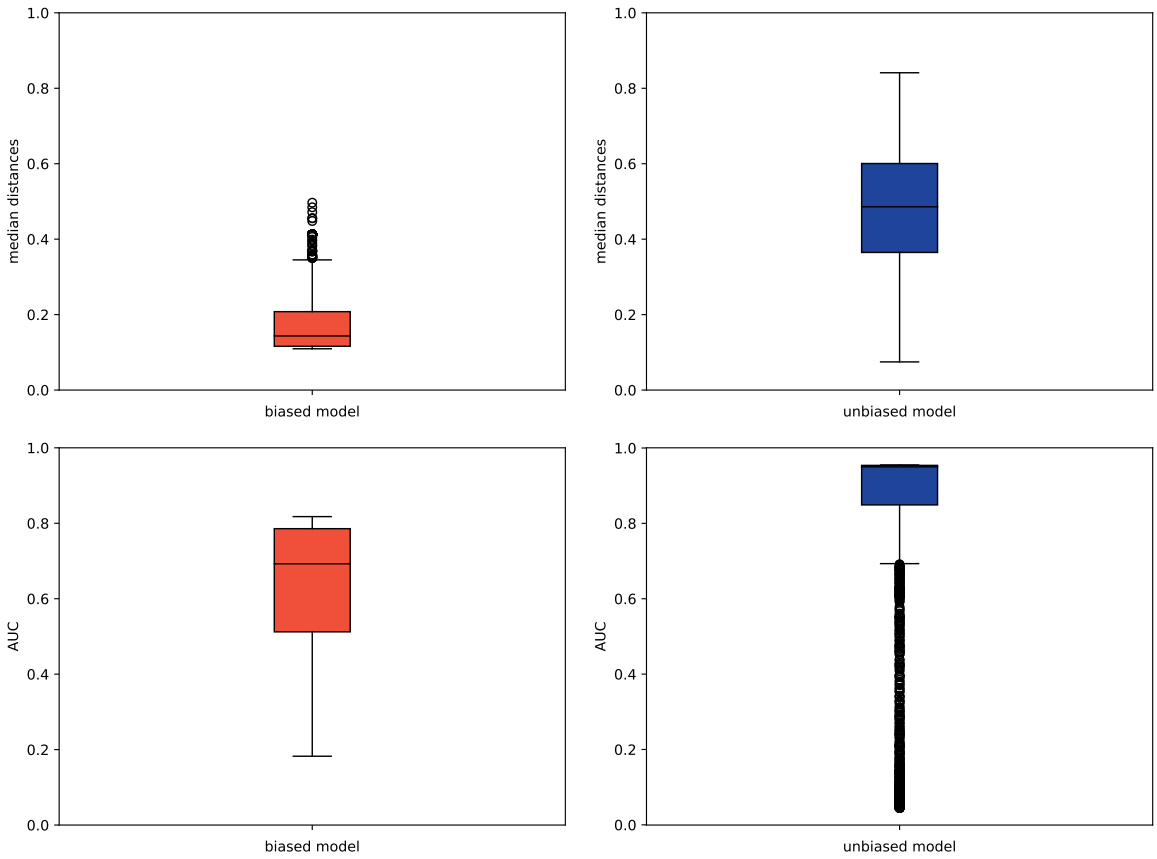

Figure 4.3: MD (first row) and AUC (second row) plots for θ_5, for the biased (left column, red) and unbiased (right column, blue) models.

very different distributions of explanations could lead to the same MD and AUC scores. This happens because for the separability of the A-compliant and not A-compliant tuples what matters is only that the explanations of the former set are different from those of the latter. Nonetheless, this can prove very useful as an a priori investigation to check if the model recognizes a particular set of the inputs and treats it differently.
In general, having MD around 0.5, with wide whiskers and quartiles, and AUC close to 1.0, with low variance are promising signs of a well separable $\mathcal{C}_A$ set.

4.3 Comparing MD and AUC to Coverage, Validity and Sharpness

As we discussed in Section 3.3, the three metrics of Zhou, Ribeiro, and Shah [59] complement the picture drawn by MD and AUC.

Coverage provides further insights on how many $\boldsymbol{u}$ tuples satisfy the constraint A of a hypothesis. Hence, it is very useful as MD and AUC do not provide a way to quantify the portion of tuples that satisfy the antecedent of θ. In addition, some distances and hypothesis allow for multiple interpretations of the MD and AUC results; coverage, in conjunction with other statistics of the dataset (as, for example, proportions of data points that have answer in the lth sentence, correctness of models and frequency of errors), can in those cases provide further and useful elements to cross out some possibilities and corroborate some available options.

Validity is precious as it gives a quantitative estimation of the percentage of $\boldsymbol{u}$ tuples that satisfy A and have contributions between two given thresholds (as exemplified in the paragraph for θ_1). Thus, validity can be thought of as a measure of how sufficient it is for a tuple to satisfy A, in order for the model to exhibit the behaviour B. Additionally, it is also a very intuitive metric and can be used, in parallel to MD and AUC, to validate the conclusions about the model predictions.

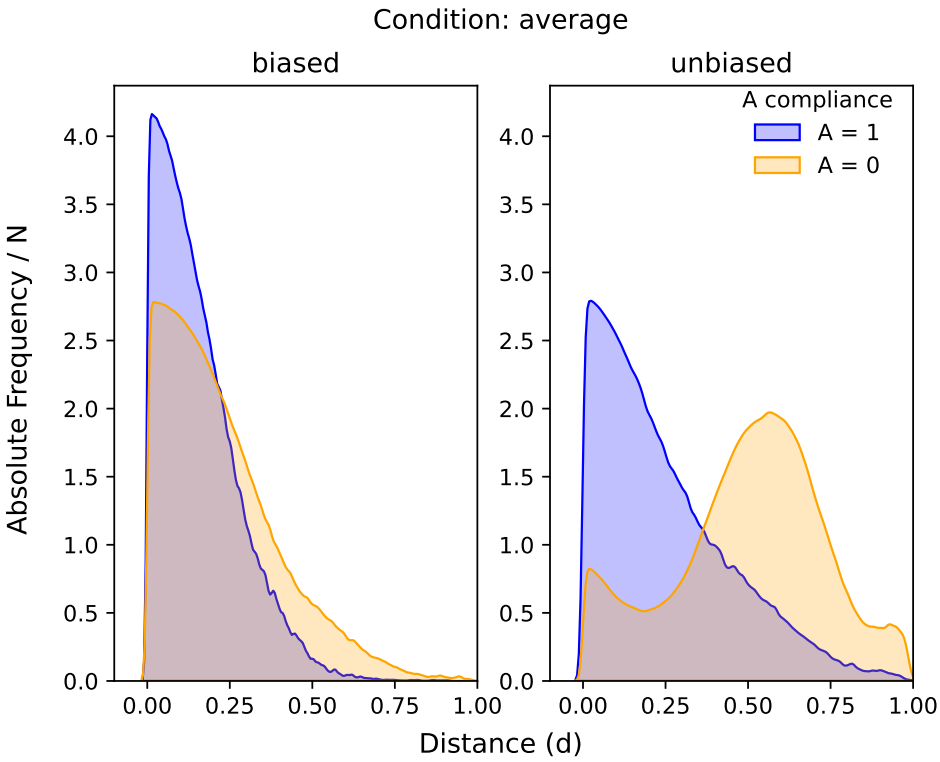

Figure 4.4: Distance density plots for θ_5, for the biased and unbiased models. In blue, tuples that satisfy A, in orange tuples which do not satisfy A. Results for the biased model are on the left, while results for the unbiased model are on the right. The y-axis is for the frequency at which the value of the distance occurs, while on the x-axis we plot the value.

Sharpness, similarly to validity gives a more quantitative overview on hypothesis compliance than MD and AUC. It also allows us to calculate the percentage of tuples that, while satisfying B, are not complying with A. Intuitively, this can be used to ascertain how necessary complying with A is to have the model behave as described by B.

To sum up, the main advantages of coverage, validity and sharpness, are two. Firstly, they do not require defining a distance or choosing a reference point. Secondly, they provide further insights on the percentages of tuples that comply with the A part and the fraction of tuples that has contribution between two thresholds. Similar quantitative information is lost when using median distance and AUC: both hide the information about the specific amounts of tuples that satisfy θ at specific thresholds in one single number, as the first measure is a median, while the second is cumulative.
However, the main shortcoming of validity and sharpness is that they do not provide information about the relative distance or the distribution of explanations: in explanation space, regardless of whether tuples in $\mathcal{C}_\theta$ are very close to those in $\mathcal{C}_A$, or are very far from them, scores for validity and sharpness will be unchanged, as they are not sensitive of such differences.

If we want a measure of how well a hypothesis is matched by the behaviour of a model, the metrics of Zhou, Ribeiro, and Shah [59] are insufficient: firstly, we need a way to compare the behaviours of the model on different data points, and to quantify how similar is one behaviour to another; exactly for that purpose we introduced a distance in Definition 2.4 and 2.9. After having a way to compare behaviours, we want to know if a given behaviour, B, is specific of a particular group of data points (characterized by A), or if it is not. We also want to know if the model behaves coherently across that group, or if it follows that behaviour only occasionally. To the latter end, we must have a cumulative measure of the similarity of the behaviours of the model across the group of data points selected with A, which is exactly what MD provides; for the former goal we want a measure of how distinguishable are data points of the group of interest from the others, and that is indeed what AUC gives us.

In Table 4.1, we provide a summary showcasing the dependencies and interpretations of all the metrics discussed in this chapter. In the *Dep* columns one can check the dependencies of each metric, i.e. if it depends on the definition of distance (D) or threshold (T), while in the *Interpretation* columns the meaning of each metric is summarized, together with its general implications when high or low, and what we would ideally want for a good semantic match; 'Nan' is used to indicate that, based on the hypothesis and the objectives one might desire a high or low score in different scenarios.

Metric	Dep		Interpretation			
	D	**T**	**Meaning**	**High**	**Low**	**Want**
AUC	Y	Y	Separability of θ-compliant tuples from A-compliant tuples	Distance is a good predictor of θ-compliance	Distance is a bad predictor of θ-compliance	High
MD	Y	Y	Distance between A-compliant from θ-compliant	Distance between $\mathcal{C}_\theta$ and $\mathcal{C}_A$ is high	Distance between $\mathcal{C}_\theta$ and $\mathcal{C}_A$ is low	Low
cov	N	N	Proportion of A-compliant tuples	A-compliance is common	A-compliance is rare	Nan
val	N	Y	Proportion of B-compliant tuples among A-compliant	A-compliance is sufficient for B-compliance	A-compliance is not sufficient for B-compliance	High
$sharp$	N	Y	Proportion of A-compliant tuples among B-compliant	A-compliance is necessary for B-compliance	A-compliance is not necessary for B-compliance	High
$AUC^{\mathcal{U}}_{\mathcal{C}_\theta}$	Y	Y	Separability of θ-compliant tuples from all tuples	Distance is a good predictor of θ-compliance	Distance is a bad predictor of θ-compliance	High
$MD^{\mathcal{U}}_{\mathcal{C}_\theta}$	Y	Y	Distance between all tuples from θ-compliant	Distance between $\mathcal{C}_\theta$ and $\mathcal{U}$ is high	Distance between $\mathcal{C}_\theta$ and $\mathcal{U}$ is low	Nan
$AUC^{\mathcal{U}}_{\mathcal{C}_A}$	Y	N	Separability of A-compliant tuples from not A-compliant tuples	Distance is a good predictor of A-compliance	Distance is a bad predictor of A-compliance	High
$MD^{\mathcal{U}}_{\mathcal{C}_A}$	Y	N	Distance between all tuples from A-compliant	Distance between $\mathcal{C}_A$ and $\mathcal{U}$ is high	Distance between $\mathcal{C}_A$ and $\mathcal{U}$ is low	High

Table 4.1: Summary of the main characteristics of the metrics discussed in this chapter.

Conclusion

The objective of this thesis was to verify that the semantic match framework can be fruitfully applied to find and characterize biases in trained models. Having defined the main concepts of the framework, as well as the tools it relies on in Chapter 1 and 2, we set up an experiment to test if it is possible to formalize hypotheses to differentiate between two models, based on their behaviours.

The experiment was conducted in a controlled setting, adopting a specific training policy, inspired by Ko et al. [27], in order to obtain a model which was possibly biased on the first sentence of the context, $\mathfrak{f}^b$, and a second model which should not have that bias, $\mathfrak{f}^u$. An initial evaluation of the two corroborated those beliefs, displaying a relevant difference in the accuracy of the models, Figure 2.1.
Afterwards, we defined four hypotheses in order to match the bias of the first model, and to further check that the second model had indeed learned to generalize properly. Employing MD and AUC we were able to measure the match between the hypotheses and behaviours of the models, highlighting a substantial difference between the two: while $\mathfrak{f}^b$ always relied on the first sentence to make its predictions, regardless of the answer's location, $\mathfrak{f}^u$ was rarely considering that sentence significant, and was doing so often only when the answer was there. With θ_4 we tested that the behaviour of $\mathfrak{f}^u$ was general: when the prediction is correct, the model is considering the answer's sentence important. This also confirmed our a priori intuition on where a model should focus to solve the question answering task correctly, thus verifying that our expectations of what a good model *ought to do* were indeed what $\mathfrak{f}^u$ *was doing*.

Finally, we analysed the metrics used, explaining the significance and implications of common trends encountered in the experimental results. We commented on the major limit cases for the metrics' scores and discussed how to assess semantic match, providing suggestions on how to redefine distance and hypotheses to improve the scores.
As both metrics depend on the choice of a sample set and a reference set, we showed how changing those sets and re-evaluating the metrics can give further insight on the coherence of the behaviour described by θ, and also on how well the model can distinguish data points with specific characteristics (defined in A) from all the others. To broaden our discussion we included alternative metrics, introduced in Zhou, Ribeiro, and Shah [59], that immediately apply to the semantic match framework, and showed how they can help to confirm the results of MD and AUC from a more quantitative perspective. We considered advantages and disadvantages of each metric and summarized them in Table 4.1.

The main contribution of this work is thus twofold: on one side, we successfully applied semantic match with an experiment in the text modality, on a real-world dataset, albeit in a controlled setting. This provided evidence of the utility and applicability of the framework to complex models solving non-trivial real-world tasks. With the experiment we also showed that it is indeed possible to characterize and match both undesired biased behaviours and desired strategies. While the former is useful to check for fairness and robustness of trained models, the latter is valuable, as it can inform users about the inner workings of black-box or gray-box models, providing a statistically-sound way to verify if human expectations on how the model should behave align with the actual explanations drawn from the model in action.

On the other side, we discussed how to interpret MD and AUC comparing them to other metrics; we highlighted advantages and disadvantages of each, showing also how evaluating MD and AUC on different sets

can be helpful not only to further corroborate previous results, but also to explore the space of possibilities for hypothesis-definition, in a scenario where no a priori knowledge of the model and possible biases of the dataset is given.

As the semantic match framework is yet under development, there is still much work to do, in the experimental and in the theoretical directions alike.

In the experimental branch, while there is already work on images in Cinà et al. [11], applying semantic match to tabular and medical datasets could really showcase its power, and possibly help in the discovery of unbalances in datasets and of confirmation biases in the analysis of trained model. Besides adding modalities, an experiment in less-controlled scenario, where there is no previous knowledge of the dataset and of the trained model is primary, as this would further prove the effectiveness of the framework.
Employing a white-box model and testing different hypotheses, comparing the ones for which there is semantic match with the knowledge we have of its inner working would also be important, as it is yet another way to test the effectiveness of the framework.
Trials with other feature attribution methods, besides highlighting the applicability of semantic match, could be interesting under another perspective: do different explanatory methods bring to the same conclusions? That is, are the results regarding a hypothesis and a model invariant under changes of explanatory method? And if they are not, is this generally true? If so, the framework itself might not be reliable enough; conversely, if this happens only for some feature attribution methods, while most of the other agree, we might have reasons to rely less on the explanatory methods that have divergent results.
There are also many improvements which could be implemented in the codebase: a low-hanging fruit in that direction is the refactoring of the loops for testing hypotheses at different thresholds; currently, operations to calculate distances are superfluously repeated, and re-implementing some list comprehensions with `numpy` should further speed up the process. While the repetitions were noted only later in development, list comprehensions were chosen to make code more readable and understandable, which was deemed more valuable at this stage.
Furthermore, hypotheses generation and automatic hypotheses testing are also important upgrades; they would not only provide a semi-automatic framework to make and test hypotheses on model's behaviours, but might also further reduce the risk of confirmation bias, as human intervention in the explanatory process is reduced.

In the theoretical realm, one of the main future objectives is the formalization of hypotheses in logical language, which should provide structure to the space of hypotheses, in turn enabling automated generation and search. This should also allow an analysis of the dependence of the results to the specification of hypothesis: for instance, how does adding logical connectives or changing signs of the mathematical expressions reflect on the results of the metrics? Are there some patterns and relations in such changes?
In addition, fine-tuning the metrics and studying in more detail how their scores link back to the distribution of explanations, albeit not an easy task, could make these cumulative measures even more informative, providing a better picture of what the model is doing.
A broader inquiry on the relation between hypotheses and distances is also needed, as both MD and AUC rely on the definition of the latter. While for simple hypothesis like θ_1 finding the *right* distance can be straightforward, for more complex conjectures multiple options seem natural, and choosing between them is not always immediate.[9] After defining a logical language to formalize the hypotheses, one could also create algorithms to automatically specify the distance.
Finally, one could define counterfactual hypotheses by adding new data points that differ in specific features, compute their predictions and explanations, and re-evaluate the hypotheses on them. This is a way to investigate some specific virtual scenarios related to the actual use cases of the model. For example, one could generate new data points near the outliers (by regenerating some features with a re-sampling from the dataset distribution),

[9]For instance, in θ_4 we used the absolute difference between the contribution of the answer's sentence, but we could have used also the absolute difference between the contribution of the sentence with the highest contribution, or, even, the absolute difference between, on one side, the absolute difference between the contributions of sentence with the highest contribution and the answer' sentence of the examined tuple, and on the other, the same statistic for the reference point.

and compare the explanations obtained for the new points with the ones of the outliers. In principle, if the median distance is small, it will imply that the behaviour of the model over similar outliers will at least be coherent, and therefore we expect the model to behave as observed during training for similar and unseen critical points.

Bibliography

[1] Kumar Abhishek and Deeksha Kamath. *Attribution-Based XAI Methods in Computer Vision: A Review*. Nov. 2022. arXiv: 2211.14736 [cs]. (Visited on 04/10/2024).

[2] Google AI. *Responsible AI Practices: Interpretability*. 2019. URL: https://ai.google/responsibility/responsible-ai-practices/ (visited on 11/04/2024).

[3] "AI Investment Forecast to Approach $200 Billion Globally by 2025". In: *Goldman Sachs* (Aug. 2023). URL: https://www.goldmansachs.com/intelligence/pages/ai-investment-forecast-to-approach-200-billion-globally-by-2025.html (visited on 04/12/2024).

[4] Shunichi Amari. "A Theory of Adaptive Pattern Classifiers". In: *IEEE Transactions on Electronic Computers* EC-16.3 (June 1967), pp. 299–307. DOI: 10.1109/PGEC.1967.264666. (Visited on 05/27/2024).

[5] Marcelo Arenas, Pablo Barceló Leopoldo Bertossi, and Mikaël Monet. *The Tractability of SHAP-Score-Based Explanations over Deterministic and Decomposable Boolean Circuits*. Apr. 2021. arXiv: 2007.14045 [cs]. (Visited on 04/12/2024).

[6] Marcelo Arenas et al. *On Computing Probabilistic Explanations for Decision Trees*. June 2022. arXiv: 2207.12213 [cs]. (Visited on 04/12/2024).

[7] Marcelo Arenas et al. *On the Complexity of SHAP-Score-Based Explanations: Tractability via Knowledge Compilation and Non-Approximability Results*. Mar. 2023. arXiv: 2104.08015 [cs]. (Visited on 02/21/2024).

[8] Kevin Bauer, Moritz Von Zahn, and Oliver Hinz. "Expl(AI)Ned: The Impact of Explainable Artificial Intelligence on Users' Information Processing". In: *Information Systems Research* 34.4 (Dec. 2023), pp. 1582–1602. DOI: 10.1287/isre.2023.1199. (Visited on 04/10/2024).

[9] Guy Van den Broeck et al. *On the Tractability of SHAP Explanations*. Jan. 2021. arXiv: 2009.08634 [cs]. (Visited on 04/12/2024).

[10] Giovanni Cinà et al. *Semantic Match: Debugging Feature Attribution Methods in XAI for Healthcare*. Feb. 2023. arXiv: 2301.02080 [cs]. (Visited on 02/21/2024).

[11] Giovanni Cinà et al. *Fixing Confirmation Bias in Feature Attribution Methods via Semantic Match*. Feb. 2024. arXiv: 2307.00897 [cs]. (Visited on 04/04/2024).

[12] Yann Le Cun, L. Bottou, and Y. Bengio. "Reading Checks with Multilayer Graph Transformer Networks". In: *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 1. 1997, 151–154 vol.1. DOI: 10.1109/ICASSP.1997.599580. (Visited on 05/27/2024).

[13] Anupam Datta, Shayak Sen, and Yair Zick. "Algorithmic Transparency via Quantitative Input Influence: Theory and Experiments with Learning Systems". In: *2016 IEEE Symposium on Security and Privacy (SP)*. May 2016, pp. 598–617. DOI: 10.1109/SP.2016.42. (Visited on 05/27/2024).

[14] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. May 2019. arXiv: 1810.04805 [cs]. (Visited on 02/21/2024).

[15] Jad Doughman and Wael Khreich. *Gender Bias in Text: Labeled Datasets and Lexicons*. Feb. 2023. arXiv: 2201.08675 [cs]. (Visited on 04/12/2024).

[16] Simone Fabbrizzi et al. "A Survey on Bias in Visual Datasets". In: *Computer Vision and Image Understanding* 223 (2022), p. 103552. DOI: 10.1016/j.cviu.2022.103552. (Visited on 05/27/2024).

[17] Ismael Garrido-Muñoz et al. "A Survey on Bias in Deep NLP". In: *Applied Sciences* 11.7 (Apr. 2021), p. 3184. DOI: 10.3390/app11073184. (Visited on 04/12/2024).

[18] Julie Gerlings, Arisa Shollo, and Ioanna Constantiou. *Reviewing the Need for Explainable Artificial Intelligence (xAI)*. 2021. arXiv: 2012.01007 [cs.HC]. (Visited on 05/27/2024).

[19] Marzyeh Ghassemi, Luke Oakden-Rayner, and Andrew L Beam. "The False Hope of Current Approaches to Explainable Artificial Intelligence in Health Care". In: *The Lancet Digital Health* 3.11 (Nov. 2021), e745–e750. DOI: 10.1016/S2589-7500(21)00208-9. (Visited on 04/15/2024).

[20] Bryce Goodman and Seth Flaxman. "European Union Regulations on Algorithmic Decision-Making and a "Right to Explanation"". In: *AI Magazine* 38.3 (Sept. 2017), pp. 50–57. DOI: 10.1609/aimag.v38i3.2741. (Visited on 04/11/2024).

[21] David Harrison and Daniel L Rubinfeld. "Hedonic Housing Prices and the Demand for Clean Air". In: *Journal of Environmental Economics and Management* 5.1 (1978), pp. 81–102. DOI: 10.1016/0095-0696(78)90006-2. (Visited on 05/27/2024).

[22] *Hazardous Substance Fact Sheet*. Aug. 2009. URL: https://www.nj.gov/health/eoh/rtkweb/documents/fs/1357.pdf.

[23] Matthew Honnibal and Ines Montani. "spaCy 2: Natural Language Understanding with Bloom Embeddings, Convolutional Neural Networks and Incremental Parsing". 2017. URL: https://spacy.io/ (visited on 05/27/2024).

[24] Krystal Hu. "ChatGPT Sets Record for Fastest-Growing User Base - Analyst Note". In: *Reuters* (Feb. 2023). URL: https://www.reuters.com/technology/chatgpt-sets-record-fastest-growing-user-base-analyst-note-2023-02-01/ (visited on 04/12/2024).

[25] Xuanxiang Huang and Joao Marques-Silva. *The Inadequacy of Shapley Values for Explainability*. Feb. 2023. arXiv: 2302.08160 [cs]. (Visited on 04/12/2024).

[26] Okyay Kaynak. "The Golden Age of Artificial Intelligence: Inaugural Editorial". In: *Discover Artificial Intelligence* 1.1 (Dec. 2021). DOI: 10.1007/s44163-021-00009-x. (Visited on 04/12/2024).

[27] Miyoung Ko et al. "Look at the First Sentence: Position Bias in Question Answering". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, 2020, pp. 1109–1121. DOI: 10.18653/v1/2020.emnlp-main.84. (Visited on 02/21/2024).

[28] Satyapriya Krishna et al. *The Disagreement Problem in Explainable Machine Learning: A Practitioner's Perspective*. 2022. arXiv: 2202.01602 [cs.LG]. (Visited on 05/27/2024).

[29] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Communications of the ACM* 60.6 (May 2017), pp. 84–90. DOI: 10.1145/3065386. (Visited on 04/12/2024).

[30] Jorge M. Lobo, Alberto Jiménez-Valverde, and Raimundo Real. "AUC: A Misleading Measure of the Performance of Predictive Distribution Models". In: *Global Ecology and Biogeography* 17.2 (Mar. 2008), pp. 145–151. DOI: 10.1111/j.1466-8238.2007.00358.x. (Visited on 02/21/2024).

[31] Octavio Loyola-Gonzalez. "Black-Box vs. White-Box: Understanding Their Advantages and Weaknesses From a Practical Point of View". In: *IEEE Access* 7 (2019), pp. 154096–154113. DOI: 10.1109/ACCESS.2019.2949286. (Visited on 04/16/2024).

[32] W Lu et al. "Deep Learning for HRRP-based Satellite Recognition". In: *Journal of Physics: Conference Series* 1267.1 (July 2019), p. 012002. DOI: 10.1088/1742-6596/1267/1/012002. (Visited on 04/10/2024).

[33] Scott Lundberg. *An Introduction to Explainable AI with Shapley Values*. 2018. URL: https://shap.readthedocs.io/en/latest/example_notebooks/overviews/An%20introduction%20to%20explainable%20AI%20with%20Shapley%20values.html (visited on 05/27/2024).

[34] Scott Lundberg. *Explaining a Question Answering Transformers Model*. 2018. URL: https://shap.readthedocs.io/en/latest/example_notebooks/text_examples/question_answering/Explaining%20a%20Question%20Answering%20Transformers%20Model.html (visited on 05/27/2024).

[35] Scott Lundberg and Su-In Lee. *A Unified Approach to Interpreting Model Predictions*. Nov. 2017. arXiv: 1705.07874 [cs, stat]. (Visited on 02/21/2024).

[36] Scott M. Lundberg et al. "From Local Explanations to Global Understanding with Explainable AI for Trees". In: *Nature Machine Intelligence* 2.1 (Jan. 2020), pp. 56–67. DOI: 10.1038/s42256-019-0138-9. (Visited on 02/21/2024).

[37] Marvin Minsky and Seymour A. Papert. *Perceptrons: An Introduction to Computational Geometry*. The MIT Press, 2017. DOI: 10.7551/mitpress/11301.001.0001. (Visited on 05/27/2024).

[38] John Muschelli. "ROC and AUC with a Binary Predictor: A Potentially Misleading Metric". In: *Journal of Classification* 37.3 (Dec. 2019), pp. 696–708. DOI: 10.1007/s00357-019-09345-1. (Visited on 05/27/2024).

[39] Michael Neely et al. *A Song of (Dis)Agreement: Evaluating the Evaluation of Explainable Artificial Intelligence in Natural Language Processing*. 2022. arXiv: 2205.04559 [cs.CL]. (Visited on 05/27/2024).

[40] Pranav Rajpurkar et al. *SQuAD: 100,000+ Questions for Machine Comprehension of Text*. Oct. 2016. arXiv: 1606.05250 [cs]. (Visited on 02/21/2024).

[41] Shaina Raza, Deepak John Reji, and Chen Ding. "Dbias: Detecting Biases and Ensuring Fairness in News Articles". In: *International Journal of Data Science and Analytics* (Sept. 2022), pp. 1–21. DOI: 10.1007/s41060-022-00359-4. (Visited on 05/27/2024).

[42] Shaina Raza et al. *NBIAS: A Natural Language Processing Framework for Bias Identification in Text*. Aug. 2023. arXiv: 2308.01681 [cs]. (Visited on 04/12/2024).

[43] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. *"Why Should I Trust You?": Explaining the Predictions of Any Classifier*. Aug. 2016. arXiv: 1602.04938 [cs, stat]. (Visited on 04/14/2024).

[44] F. Rosenblatt. "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain." In: *Psychological Review* 65.6 (1958), pp. 386–408. DOI: 10.1037/h0042519. (Visited on 04/12/2024).

[45] Cynthia Rudin. *Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead*. 2019. arXiv: 1811.10154 [stat.ML]. (Visited on 05/27/2024).

[46] Germans Savcisens et al. "Using Sequences of Life-events to Predict Human Lives". In: *Nature Computational Science* 4.1 (Dec. 2023), pp. 43–56. DOI: 10.1038/s43588-023-00573-5. (Visited on 04/11/2024).

[47] Lloyd S. Shapley. *Notes on the N-person Game - II: The Value of an N-person Game*. Santa Monica, CA: RAND Corporation, 1951. DOI: 10.7249/RM0670. (Visited on 05/27/2024).

[48] Benjamin Shickel et al. "Deep EHR: A Survey of Recent Advances in Deep Learning Techniques for Electronic Health Record (EHR) Analysis". In: *IEEE Journal of Biomedical and Health Informatics* 22.5 (Sept. 2018), pp. 1589–1604. DOI: 10.1109/JBHI.2017.2767063. (Visited on 04/10/2024).

[49] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. *Learning Important Features through Propagating Activation Differences*. 2019. arXiv: 1704.02685 [cs.CV]. (Visited on 05/27/2024).

[50] Amitojdeep Singh, Sourya Sengupta, and Vasudevan Lakshminarayanan. *Explainable Deep Learning Models in Medical Image Analysis*. May 2020. arXiv: 2005.13799 [cs, eess]. (Visited on 04/10/2024).

[51] Kristinn R. Thórisson et al. *Why Artificial Intelligence Needs a Task Theory — And What It Might Look Like*. May 2016. arXiv: 1604.04660 [cs]. (Visited on 04/10/2024).

[52] Tatiana Tommasi et al. *A Deeper Look at Dataset Bias*. 2015. arXiv: 1505.01257 [cs.CV]. (Visited on 05/27/2024).

[53] Leslie G. Valiant. "The Complexity of Enumeration and Reliability Problems". In: *SIAM Journal on Computing* 8.3 (Aug. 1979), pp. 410–421. DOI: 10.1137/0208032. (Visited on 04/12/2024).

[54] Ashish Vaswani et al. "Attention Is All You Need". In: *Advances in Neural Information Processing Systems*. 2017, pp. 5998–6008. URL: https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html (visited on 05/27/2024).

[55] Pablo Villalobos et al. *Machine Learning Model Sizes and the Parameter Gap*. July 2022. arXiv: 2207.02852 [cs]. (Visited on 04/12/2024).

[56] Charles Wan, Rodrigo Belo, and Leid Zejnilović. "Explainability's Gain Is Optimality's Loss? – How Explanations Bias Decision-making". In: *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*. July 2022, pp. 778–787. DOI: 10.1145/3514094.3534156. (Visited on 04/10/2024).

[57] Danding Wang et al. "Designing Theory-Driven User-Centric Explainable AI". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Glasgow Scotland Uk: ACM, May 2019, pp. 1–15. DOI: 10.1145/3290605.3300831. (Visited on 04/10/2024).

[58] Lilian Weng. *Attention? Attention!* June 2018. URL: https://lilianweng.github.io/posts/2018-06-24-attention/ (visited on 05/27/2024).

[59] Yilun Zhou, Marco Tulio Ribeiro, and Julie Shah. *ExSum: From Local Explanations to Model Understanding*. Apr. 2022. arXiv: 2205.00130 [cs]. (Visited on 02/21/2024).

[60] Yilun Zhou et al. "Do Feature Attribution Methods Correctly Attribute Features?" In: *Proceedings of the AAAI Conference on Artificial Intelligence* 36.9 (June 2022), pp. 9623–9633. DOI: 10.1609/aaai.v36i9.21196. (Visited on 04/12/2024).

[61] Jinan Zou et al. *Stock Market Prediction via Deep Learning Techniques: A Survey*. Feb. 2023. arXiv: 2212.12717 [q-fin]. (Visited on 04/10/2024).

Appendix A

Details on the implementation of hypotheses

In this appendix, we delve in the most important aspects and features of the implementation of hypotheses in the codebase.

Since hypotheses are logical statements, we modelled them as functions that return either `true` or `false`. We define an *ad hoc* class, with `name`, `description`, `condition`, `A` and `B` attributes by default, together with the additional `split_idx`, `theta` and `aux` attributes. The first two default attributes are used during the experiment for logging and feedback, the third acts as a switch, choosing if the average or max contribution should be used, while the `A` and `B` attributes are functions, implementing the constraint A and describing the behaviour B of the hypothesis. Their input is a dictionary with two keys for the explanation $\boldsymbol{e}_i = \mathfrak{c}_1(\mathfrak{e}(\boldsymbol{u}_i))$, `average`, which is a vector of the sentence contributions (Definition 2.8), and `max`, which is a vector of sentence contributions as defined in Footnote 18 of Chapter 2. Additionally, there are keys to specify characteristics of the $\boldsymbol{u}_i$ tuple: `pos`, `max_split`, `max_att`, `em_score` and `f1_score`. The last two are used to pass the scores of the model's prediction on that data point, while the first is used to pass the location of the sentence in which the correct answer is located. `max_split` and `max_att` contain respectively the location of the sentence which has maximum contribution and the amount of contribution of that sentence. With this information it is possible to add constraints on properties of $\boldsymbol{u}$, such as answer position or model correctness.
The additional attributes of the `Hypothesis` class are used as follows: `split_idx` chooses a specific sentence for the hypothesis (for example in θ_1, θ_2 and θ_3 this equals 0, as in those hypotheses we are interested in the contribution of the first sentence), `theta` sets a specific z threshold for the hypothesis, and `aux` is a dictionary providing any further information needed for the hypothesis or evaluation of the metrics (for instance labels or colours for the graphs).
To implement a hypothesis one must define the `A` and `B` functions, which, together with the inputs and additional attributes, should make coding a hypothesis simple and general enough to express very diverse statements about contributions.

After the hypothesis is defined, the `hypothesis_compliance` function applies it to a `Pandas-Dataframe` storing the processed contributions of each data point in $\mathcal{D}^v$. This returns a list with one entry per data point, 0 if the A-part hypothesis is not satisfied, 1 if the hypothesis is satisfied and 2 if the A-part is satisfied but the B-part is not. When the A-part returns `None`, signalling an error, -1 is added. A -2 is added if the B-part of the hypothesis returns `None`, while -3 is used if the value returned by A is not in $\{\texttt{true}, \texttt{false}, \texttt{None}\}$, and -4 is used in the same case for B. These signals should ease the troubleshooting when the hypothesis is producing unexpected outputs, and checking if errors occurred just amounts to search if -1, -2, -3 or -4 appear in the list. Furthermore, with a simple list comprehension it is possible to obtain the set of data points which satisfy the hypothesis, the set of those that only satisfy A, and the set of those that do not satisfy the

hypothesis (either because they do not comply with its A- or B-part).

Note that the mapping process, the post-processing of SHAP values, the hypothesis and distance are all independent of the way in which the input is divided into sentences. Any other splitting function could be employed, thus the number of possible hypotheses is further broadened: while it is reasonable for the dataset we used to split the context based on sentences, an alternative choice could have been that of splitting the context into groups of c-many tokens. This is motivated by the fact that, in principle, while the model could use punctuation to develop the idea of a *sentence*, dots are often used in English for abbreviations, and might confuse the model.[1] Therefore, one could also investigate if a hypothesis holds on subordinate clauses, or specific fractions of the context.

[1]While running the experiment, we actually observed that the spaCy sentencizer we employ is susceptible of such errors.

Appendix B

List of data points throwing errors for SHAP values

In the next pages we list all the ids of the original SQuAD validation set that threw an error while calculating SHAP values. For this reason they were not included in the calculations of MD and AUC.

The error is likely caused by the masking function of the SHAP library. This function is used to generate the subsets on which the contribution of each feature is calculated, as specified in Definition 1.5. Sometimes, when proper names are present in the question, an inconsistent tokenization is produced, which in turn throws a length mismatch error, preventing the calculation of SHAP values. As this does not happen every time a proper name is in the question, it is hard to find a working patch, and we opted to simply exclude the data points with this problem.
Out of the 10,570 data points of $\mathcal{D}^v$, a total of 1,191 threw the error, or around 11.3%.

Table B.1: List of all the ids of data points from the SQuAD validation set that threw errors while calculating SHAP values.

56be53b8acb8001400a50316	56d6edd00d65d21400198251	56be53b8acb8001400a50314	56beb4343aeaaa14008c925d
56bebad93aeaaa14008c92fa	56d70daa0d65d21400198336	56bf3fd53aeaaa14008c9591	56bf6b303aeaaa14008c960e
56d99da8dc89441400fdb5ff	56d9a7addc89441400fdb6a8	56d70daa0d65d21400198334	56bf3e803aeaaa14008c9588
56bf38383aeaaa14008c956f	56beb86b3aeaaa14008c92c1	56bebd713aeaaa14008c932f	56d9a7addc89441400fdb6ab
56beaf5e3aeaaa14008c91ff	56bf48cc3aeaaa14008c95ac	56be53b8acb8001400a50315	57340a094776f4190066177f
5737a9afc3c5551400e51f61	5730ec85e6313a140071cabb	5733d5704776f41900661310	573403394776f419006616df
573403394776f419006616e0	5730c52fb54a4f140068cc47	57373a9fc3c5551400e51e7f	5737a84dc3c5551400e51f5b
573735e8c3c5551400e51e75	5733fd66d058e614000b6735	573749741c45671900574460	5737a9afc3c5551400e51f62
57379a4b1c456719005744cd	573407d7d058e614000b6813	5734025d4776f419006616c4	573796edc3c5551400e51f35
573796edc3c5551400e51f33	57309a6c2461fd1900a9cf03	57340111d058e614000b677d	573796edc3c5551400e51f37
57378862c3c5551400e51f21	5733fe73d058e614000b673f	57377aac1c45671900574479	5730e936aca1c71400fe5b63
5737821cc3c5551400e51f19	5737a9afc3c5551400e51f64	5737a9afc3c5551400e51f63	5730aaa88ab72b1400f9c64f
57378862c3c5551400e51f22	5737432bc3c5551400e51e9c	573750f61c45671900574468	573403394776f419006616de
573735e8c3c5551400e51e73	5737a5931c456719005744e8	5737a5931c456719005744ea	5730ac6b8ab72b1400f9c671
5733fe73d058e614000b673d	5733f9fa4776f41900661620	5737958ac3c5551400e51f2a	5730ac6b8ab72b1400f9c672
573406d1d058e614000b6801	5733f5264776f419006615a5	57377aac1c4567190057447a	5730bf03069b5314008322ed
5733db8dd058e614000b642a	5730ac6b8ab72b1400f9c673	5730c6d3b54a4f140068cc4e	5737a9afc3c5551400e51f65
5730afed069b53140083225f	57378862c3c5551400e51f23	5737a0acc3c5551400e51f48	5737958b1c456719005744c4
573796edc3c5551400e51f34	573403394776f419006616dd	5733fb7bd058e614000b66ff	57379a4b1c456719005744cf
5730b9852461fd1900a9cffb	5730b5cc396df919000962d4	5730ac6b8ab72b1400f9c670	5730b07c8ab72b1400f9c696
5733e5a14776f4190066145c	56dfb6d17aa994140058e056	56de3ebc4396321400ee26e7	56e10179cd28a01900c67415
56de0ed14396321400ee257a	56de148dcffd8e1900b4b5bc	56df9ee138dc421700152108	56e0ccaa7aa994140058e71a
573368e54776f41900660a54	56dfb5777aa994140058e025	56de4a89cffd8e1900b4b7bd	56de0ed14396321400ee2579
56de15104396321400ee25b9	57337f6ad058e614000b5bcd	56e10179cd28a01900c67414	57337f6ad058e614000b5bce
56dfa2c54a1a83140091ebf3	56de4a474396321400ee2787	56de49564396321400ee277a	56de52614396321400ee27fd
573383d0d058e614000b5c39	56de148dcffd8e1900b4b5bd	56de3e414396321400ee26d9	56e74d1f00c9c71400d76f70
56e1e9dfe3433e14004231fd	56e1c2eee3433e1400423138	56e74d1f00c9c71400d76f71	56e11a73e3433e1400422bf3
56e10325cd28a01900c67438	56e1127bcd28a01900c6754a	56e74faf00c9c71400d76f98	56e1b62ecd28a01900c67aa5
56e11d8ecd28a01900c675f3	56e200e4cd28a01900c67c18	56e11d8ecd28a01900c675f5	56e12005cd28a01900c67617

Continued on next page

Table B.1 : List of data points throwing errors when calculating SHAP values. (Continued)

56e1b355e3433e14004230b2	56e7535037bdd419002c3e73	56e1e9dfe3433e14004231fe	56e75b8237bdd419002c3ed4
56e11f05e3433e1400422c31	56e127bccd28a01900c6765e	56e1b8f3e3433e14004230e9	56e1b169cd28a01900c67a75
56e10e73cd28a01900c674ec	56e748a200c9c71400d76f37	56e748a200c9c71400d76f3a	56e120a1e3433e1400422c38
56e1b355e3433e14004230b0	56e1b8f3e3433e14004230e7	56e1ee4de3433e1400423210	56e74d1f00c9c71400d76f6f
56e1aff7cd28a01900c67a6a	56e1b62ecd28a01900c67aa4	56e2042ecd28a01900c67c1f	56e74d1f00c9c71400d76f6e
56e1fc57e3433e140042322f	56e1c720e3433e140042316b	56e1febfe3433e1400423238	56e748a200c9c71400d76f38
56e1c2eee3433e1400423137	56e748a200c9c71400d76f39	56e1b62ecd28a01900c67aa3	56e111e5e3433e1400422b90
56e74af500c9c71400d76f65	56e10f14e3433e1400422b5e	56e111e5e3433e1400422b91	56e1ded7cd28a01900c67bd5
56e1dc62cd28a01900c67bcc	56e20a3ae3433e140042324a	56e10f14e3433e1400422b5f	56e1ec83cd28a01900c67c0a
56e1e9dfe3433e14004231fc	56f852fba6d7ea1400e1756e	56f8907faef2371900626 1b6	56f86d30a6d7ea1400e17606
56f826a7a6d7ea1400e1742a	56f81537aef2371900625db5	56f84b68aef2371900625fa9	56e77da237bdd419002c403d
56f88c37aef2371900626178	56f86d30a6d7ea1400e17607	56e77b8c00c9c71400d77196	56f8907faef23719006261b5
56f81537aef2371900625db4	56f86680a6d7ea1400e175d1	56f86b44a6d7ea1400e175f9	56f86966aef2371900626056
56e772bf37bdd419002c3fbe	56f8046faef2371900625d73	56f87392aef237190062609a	56e7788200c9c71400d77183
56f8575aaef2371900626001	56e769dc00c9c71400d770ea	56f81393aef2371900625dac	56f897059b226e1400dd0c5d
56f80143aef2371900625d68	56f84760aef2371900625f84	56f80604a6d7ea1400e17388	56f7fde8a6d7ea1400e1736b
56f86680a6d7ea1400e175ce	56f86966aef2371900626055	56f851b1a6d7ea1400e1755e	56f8907faef23719006261b3
56f84d33aef2371900625fb2	56f88025aef2371900626120	56f84760aef2371900625f81	56e772bf37bdd419002c3fbf
56f848e0a6d7ea1400e1752f	56f86680a6d7ea1400e175cf	56f81393aef2371900625da9	56f81537aef2371900625db7
56f80604a6d7ea1400e1738b	56f86d30a6d7ea1400e17609	56e772bf37bdd419002c3fbc	56f879bdaef23719006260e0
56f7e9caaef2371900625c57	56f855caaef2371900625ff7	56f82989aef2371900625e6d	56f84760aef2371900625f82
56f86d30a6d7ea1400e17608	56f867e3a6d7ea1400e175d7	56f87760aef23719006260d0	56e7673a37bdd419002c3f56
56e77e4a00c9c71400d771b3	56f8837aa6d7ea1400e176fe	56f7fde8a6d7ea1400e17367	56f855caaef2371900625ff5
56e763e800c9c71400d77087	56f7ef96aef2371900625c78	56f86966aef2371900626054	56f8907faef23719006261b2
56f867e3a6d7ea1400e175da	56f85bb8aef2371900626013	56f80143aef2371900625d6b	56f8094aa6d7ea1400e17392
56f88690a6d7ea1400e17724	56f7fde8a6d7ea1400e17368	56f7e9caaef2371900625c56	56f80604a6d7ea1400e17387
56f80604a6d7ea1400e17389	56f86d30a6d7ea1400e17605	56e772bf37bdd419002c3fbb	56f851b1a6d7ea1400e1755f
56f88025aef237190062611f	56f81537aef2371900625db3	56f7f15aa6d7ea1400e172ec	56f80604a6d7ea1400e1738a
56f8575aaef2371900625ffe	56f82549a6d7ea1400e17418	56f8c7029e9bad19000a04a4	56f8c7029e9bad19000a04a3

Continued on next page

Table B.1 : List of data points throwing errors when calculating SHAP values. (Continued)

5705fb7f52bb891400689750	56f8a4e99e9bad19000a0254	5706094b52bb8914006897de	56f8a2969e9bad19000a022b
57094a79efce8f15003a7dc6	570960cf200fba1400367f03	56f8a6969e9bad19000a025b	5705e33f52bb89140068964d
56f8b2499b226e1400dd0e3c	57097141200fba1400367ff9	57096c95200fba1400367fbc	5705ec1675f01819005e7770
570d35b7b3d812140066d54f	57096c95200fba1400367fbb	570d2556fed7b91900d45c4a	5705f09e75f01819005e77a4
56f8cc399e9bad19000a0518	56f89a959b226e1400dd0c9f	56f8b2499b226e1400dd0e3d	56f8c5909b226e1400dd0f80
56f8c0cd9b226e1400dd0f38	570d3468b3d812140066d547	56f8aa749b226e1400dd0da8	56f8a4e99e9bad19000a0251
56f8c0cd9b226e1400dd0f35	56f8aa749b226e1400dd0da7	5706143575f01819005e7951	570960cf200fba1400367f04
56f8cc399e9bad19000a0516	571a484210f8ca1400304fc0	5710e8c8a58dae1900cd6b29	57106644b654c5140001f8e5
57107a3ea58dae1900cd69e1	57115b8b50c2381900b54a8b	5710f114a58dae1900cd6b64	57111713a58dae1900cd6c02
57107d73b654c5140001f920	571077ecb654c5140001f90a	5710f114a58dae1900cd6b61	57115c7450c2381900b54aa0
57115ff82419e314009555c6	571094b7a58dae1900cd6a68	57108c95b654c5140001f97d	570d4c3bfed7b91900d45e33
571077ecb654c5140001f90c	57111ab8a58dae1900cd6c3e	571c3c47dd7acb1400e4c09f	57111b95a58dae1900cd6c51
571127a5a58dae1900cd6cc7	571114cfb654c5140001fb0a	571a4b0f10f8ca1400304fd7	5710f2e2a58dae1900cd6b75
571090abb654c5140001f998	57115f652419e314009555ba	571135b8a58dae1900cd6d12	571114cfb654c5140001fb0b
57111428b654c5140001fb00	571135b8a58dae1900cd6d11	5711163bb654c5140001fb14	5711163bb654c5140001fb15
5710eca0a58dae1900cd6b3c	5710eb6fb654c5140001fa1a	571117d4a58dae1900cd6c0a	5711119cb654c5140001fae6
57108198b654c5140001f93b	57106644b654c5140001f8e8	57115f0a50c2381900b54aa7	571077ecb654c5140001f909
57107e6ca58dae1900cd69f5	57107c24a58dae1900cd69eb	571163172419e314009555e7	57114e8d50c2381900b54a5e
57107e6ca58dae1900cd69f2	57111b95a58dae1900cd6c54	57108d69b654c5140001f987	57106185b654c5140001f8db
571126dfa58dae1900cd6cb6	5710f2e2a58dae1900cd6b76	571155ae2419e31400955591	57111428b654c5140001fb01
571117d4a58dae1900cd6c0e	571090abb654c5140001f995	570d4c3bfed7b91900d45e34	571135b8a58dae1900cd6d10
57107a3ea58dae1900cd69e0	57111713a58dae1900cd6c00	57113c6da58dae1900cd6d33	570d4c3bfed7b91900d45e32
571094b7a58dae1900cd6a67	57108073b654c5140001f929	57107e6ca58dae1900cd69f3	571094b7a58dae1900cd6a69
57106d2fb654c5140001f8ef	5710f4b8b654c5140001fa48	571093aba58dae1900cd6a5f	57115dbe2419e314009555a7
571153422419e3140095557e	57109180a58dae1900cd6a44	57111380a58dae1900cd6bd9	57107d73b654c5140001f91d
57111b95a58dae1900cd6c50	571099b2b654c5140001f9b5	5710eb6fb654c5140001fa1b	57109180a58dae1900cd6a42
570d4c3bfed7b91900d45e31	57107a3ea58dae1900cd69de	571090abb654c5140001f997	5710f2e2a58dae1900cd6b73
57112686b654c5140001fbd4	57109180a58dae1900cd6a43	571099b2b654c5140001f9b3	57115ff82419e314009555c5
57114b1a2419e31400955576	57108c95b654c5140001f97a	57108073b654c5140001f925	571c83f3dd7acb1400e4c0dc

Continued on next page

Table B.1 : List of data points throwing errors when calculating SHAP values. (Continued)

5725c604271a42140099d189	5725c604271a42140099d188	571a4ead10f8ca1400304fdd	57265526708984140094c2bd
571a4ead10f8ca1400304fde	5725c604271a42140099d186	571c96095efbb31900334dbd	571cbe35dd7acb1400e4c13e
571c7abfdd7acb1400e4c0bc	571ccc00dd7acb1400e4c15e	571ccfbadd7acb1400e4c165	571ccfbadd7acb1400e4c167
571cca4add7acb1400e4c150	571ce7f25efbb31900334e40	5725b5a689a1e219009abd29	571cd703dd7acb1400e4c184
57261dab38643c19005ad037	571c8198dd7acb1400e4c0cf	5725b76389a1e219009abd4c	57265a58dd62a815002e8271
571cd5b1dd7acb1400e4c179	571cbe35dd7acb1400e4c13d	5725c7f5271a42140099d1a2	5725c604271a42140099d187
571ce6655efbb31900334e36	571cd11add7acb1400e4c16e	571c7d55dd7acb1400e4c0c6	5725cda338643c19005acd40
571cc6f85efbb31900334de4	571a53d410f8ca1400304fe5	5726241189a1e219009ac2e0	571cbe35dd7acb1400e4c13c
571cbe35dd7acb1400e4c140	571cd88ddd7acb1400e4c18d	571a4d1a4faf5e1900b8a95a	5725c604271a42140099d185
571cbe35dd7acb1400e4c13f	571ccfbadd7acb1400e4c166	571caac55efbb31900334dc7	5725bb34271a42140099d0cb
5725c6dcec44d21400f3d532	5726975c708984140094cb21	5726975c708984140094cb20	5726b929f1498d1400e8e8ea
57269bb8708984140094cb95	5726975c708984140094cb1f	5726bc1add62a815002e8ea7	57263eaa38643c19005ad372
5725b9db38643c19005acbe1	5725be0f271a42140099d11a	57269e3bf1498d1400e8e519	57269e3bf1498d1400e8e517
5726a09f708984140094cc39	5729edd56aef051400155115	5726bc1add62a815002e8ea8	5729e6313f37b319004785a9
5725c0f289a1e219009abdf6	5726c5a9f1498d1400e8eac7	5729f3831d0469140077967f	57263eaa38643c19005ad371
57268e2bf1498d1400e8e3b1	5726926a5951b619008f7709	5725c0f289a1e219009abdf4	57265e455951b619008f70bc
5729eb34af94a219006aa6cb	5726bc1add62a815002e8ea6	5726b718dd62a815002e8dc1	5725c071271a42140099d127
5726b929f1498d1400e8e8eb	5726c3da708984140094d0db	5729ea263f37b319004785bf	5725c0f289a1e219009abdf2
5725b9db38643c19005acbe2	5726c3da708984140094d0d9	5726926a5951b619008f770a	572847dd4b864d19001648bf
572651f9f1498d1400e8dbf0	57263eaa38643c19005ad373	57269e3bf1498d1400e8e518	5726938af1498d1400e8e448
5726c5a9f1498d1400e8eac8	5726bc1add62a815002e8eaa	5726c3da708984140094d0da	572651f9f1498d1400e8dbf1
572847dd4b864d19001648bd	5725c0f289a1e219009abdf5	57265e455951b619008f70bd	572651f9f1498d1400e8dbef
57264e455951b619008f6f65	5726b718dd62a815002e8dc2	5726b718dd62a815002e8dbe	572651f9f1498d1400e8dbf2
5729e500af94a219006aa6b7	5726c002708984140094d074	5726938af1498d1400e8e449	5726c3da708984140094d0dc
572a020f6aef051400155199	572651f9f1498d1400e8dbee	5726b929f1498d1400e8e8e8	5726bc1add62a815002e8ea9
5726c5a9f1498d1400e8eac5	5726938af1498d1400e8e446	5726b929f1498d1400e8e8e9	5725c0f289a1e219009abdf3
5726c5a9f1498d1400e8eac4	5726b929f1498d1400e8e8ec	572a020f6aef05140015519a	5726b718dd62a815002e8dbf
5726ba2c708984140094cf59	5726938af1498d1400e8e447	57269e3bf1498d1400e8e51a	5726c5a9f1498d1400e8eac6
5725c95f38643c19005accf3	5726ba2c708984140094cf5b	57265e455951b619008f70bb	5726b718dd62a815002e8dc0

Continued on next page

Table B.1 : List of data points throwing errors when calculating SHAP values. (Continued)

5726a09f708984140094cc3a	57269f3ef1498d1400e8e537	5725cbb289a1e219009abed5	57269e3bf1498d1400e8e516
5725be0f271a42140099d118	572a020f6aef051400155198	5725b9db38643c19005acbe3	5726938af1498d1400e8e44a
57268f05dd62a815002e8994	572658daf1498d1400e8dcb0	57263b1638643c19005ad333	572683075951b619008f7517
57265d86f1498d1400e8dd54	572647d0708984140094c14e	57268a37f1498d1400e8e33f	57268c01dd62a815002e8913
572691bedd62a815002e89dd	5725bdbe38643c19005acc39	5726516a708984140094c226	57264cc6dd62a815002e80e7
5725fcbe271a42140099d3af	5725bae289a1e219009abd90	5725bae289a1e219009abd91	57264cc6dd62a815002e80e6
572658daf1498d1400e8dcae	57265746dd62a815002e8219	5726431d271a42140099d7f9	57265746dd62a815002e821c
5725cb33271a42140099d1de	57265746dd62a815002e821b	5726415bec44d21400f3dcd1	5725bdbe38643c19005acc3a
572686fc708984140094c8e6	5726887e708984140094c918	5725e1c4271a42140099d2db	57263eaa38643c19005ad374
57265c10f1498d1400e8dd37	5725bc0338643c19005acc11	5726887e708984140094c91a	572658daf1498d1400e8dcad
5725cb33271a42140099d1dc	5725d34aec44d21400f3d63d	5725c337271a42140099d163	5725c69738643c19005accbc
5725cb33271a42140099d1db	5726431d271a42140099d7f8	5725c69738643c19005accbb	57265746dd62a815002e8218
57268da7f1498d1400e8e39c	572691bedd62a815002e89dc	57264cc6dd62a815002e80e4	5725d662ec44d21400f3d689
5725cb33271a42140099d1df	5725d662ec44d21400f3d688	57263b1638643c19005ad335	57265aaf5951b619008f706f
572683075951b619008f7515	57264845f1498d1400e8db0d	572655e5f1498d1400e8dc60	57268a37f1498d1400e8e33e
572646655951b619008f6ebf	57268f05dd62a815002e8992	57265e97708984140094c3c6	57264e66dd62a815002e811b
57264228ec44d21400f3dcf5	5725ce4d38643c19005acd51	5725fabc89a1e219009ac12b	5726431d271a42140099d7f7
57264e66dd62a815002e811a	57264d58f1498d1400e8db7c	5725e1c4271a42140099d2da	57264a0ef1498d1400e8db43
5725c69738643c19005accb9	5726446a89cfff1900a8404e	5726398589a1e219009ac589	5726400589a1e219009ac5ee
57263eaa38643c19005ad375	5726431d271a42140099d7f5	572655e5f1498d1400e8dc61	5725d34089a1e219009abf52
57264cfa708984140094c1c6	5725f00938643c19005aced7	57265e97708984140094c3c4	57268a37f1498d1400e8e33d
572658daf1498d1400e8dcaf	57264b1ddd62a815002e80a1	5725c57a89a1e219009abe61	572655e5f1498d1400e8dc5f
57265e97708984140094c3c5	572634a789a1e219009ac56e	57264fe65951b619008f6fa1	57263b1638643c19005ad336
5726415bec44d21400f3dcd2	57265c10f1498d1400e8dd36	572658daf1498d1400e8dcac	57263b1638643c19005ad334
5726400589a1e219009ac5f2	572647d0708984140094c14b	5725d34aec44d21400f3d63b	57265d86f1498d1400e8dd50
57264a0ef1498d1400e8db40	572655e5f1498d1400e8dc62	5726398589a1e219009ac58a	5725db98ec44d21400f3d6c7
572647935951b619008f6ec9	572646655951b619008f6ec1	572691bedd62a815002e89de	57266193dd62a815002e832a
57265d08708984140094c39b	57268d1b708984140094c9cf	57265d08708984140094c398	57267383dd62a815002e8554
5726800add62a815002e8754	572671165951b619008f72b8	57267f695951b619008f74bf	57267de1f1498d1400e8e196

Continued on next page

Table B.1 : List of data points throwing errors when calculating SHAP values. (Continued)

57266ab3dd62a815002e8437	57269698dd62a815002e8a70	57266e72f1498d1400e8df8c	572673f5708984140094c69c
572673f5708984140094c69b	57265f605951b619008f70dc	57266e72f1498d1400e8df8e	5726a0205951b619008f781f
57266ec2dd62a815002e84a3	5726769c708984140094c715	572679c35951b619008f73df	57267947f1498d1400e8e0ec
572678c0dd62a815002e8641	57269120708984140094ca5c	572673f5708984140094c69f	57269fab5951b619008f7809
572657d9dd62a815002e8230	57268d1b708984140094c9ce	5726a9ff708984140094cd4d	57269fab5951b619008f7808
572671165951b619008f72ba	57269698dd62a815002e8a6f	5726769c708984140094c712	57269698dd62a815002e8a6e
57265f605951b619008f70df	5726847f708984140094c8ab	572686ac5951b619008f75ac	57266193dd62a815002e832e
57269e80f1498d1400e8e520	57267383dd62a815002e8553	57267947f1498d1400e8e0ed	572677e7708984140094c724
572677e7708984140094c723	57269698dd62a815002e8a6d	56d9c3a6dc89441400fdb7b5	5733638fd058e614000b59e8
57335fcad058e614000b5971	5733a2a9d058e614000b5f28	5733a2a9d058e614000b5f2a	5733a32bd058e614000b5f34
57332562d058e614000b5734	56bec8a13aeaaa14008c9437	573362b94776f41900660975	56bec9133aeaaa14008c9445
56bec9133aeaaa14008c9446	5733a5f54776f41900660f46	5733638fd058e614000b59ea	57332c1e4776f4190066073d
57339dd94776f41900660ecf	56d9c79edc89441400fdb805	56bec2013aeaaa14008c9372	56d7253b0d65d214001983d5
5733314e4776f4190066076c	56d9b7dcdc89441400fdb744	5733a32bd058e614000b5f36	57332442d058e614000b5722
57336755d058e614000b5a3f	57335c20d058e614000b58fa	57339c16d058e614000b5ec6	57332442d058e614000b5723
5733a2a9d058e614000b5f2b	57332562d058e614000b5731	56d71d150d65d2140019836f	57335c20d058e614000b58f9
57339dd94776f41900660ecd	57339a554776f41900660e77	56d9bdc1dc89441400fdb76a	57339ad74776f41900660e89
57332442d058e614000b5721	56becb823aeaaa14008c948d	5733a32bd058e614000b5f32	57339c16d058e614000b5ec9
57339902d058e614000b5e72	5733a32bd058e614000b5f33	57335c20d058e614000b58fd	56bec4343aeaaa14008c93ac
5726e06df1498d1400e8ee55	5726cc11dd62a815002e9089	5726f755708984140094d738	5726e4eedd62a815002e943a
57275273dd62a815002e9b19	5727387b5951b619008f86ea	5726e834dd62a815002e94a7	5726dba1dd62a815002e92e6
5726fc63dd62a815002e9707	57275e125951b619008f88d8	5726f90b708984140094d760	57273d19708984140094db41
5726e4eedd62a815002e943b	5726f90b708984140094d75d	5726cfa3708984140094d20a	572747dd5951b619008f87a9
57275573708984140094dc45	57273d19708984140094db3e	5726c9a4708984140094d172	5727387b5951b619008f86eb
5726dba1dd62a815002e92e5	5726d4a45951b619008f7f6a	572747dd5951b619008f87ac	5726c9a4708984140094d171
5726e834dd62a815002e94a6	572747dd5951b619008f87ab	5726e4eedd62a815002e9439	5726d4a45951b619008f7f69
57267b755951b619008f7434	5726e4eedd62a815002e9438	5726af765951b619008f7a52	5726ddf6f1498d1400e8ee06
572732f8f1498d1400e8f475	5726e313f1498d1400e8eeb4	5726ddf6f1498d1400e8ee05	5727580bf1498d1400e8f69b
5726f2375951b619008f8313	5726e3c4dd62a815002e9408	572739a75951b619008f86f9	57277632f1498d1400e8f8c7

Continued on next page

Table B.1 : List of data points throwing errors when calculating SHAP values. (Continued)

5726eb4b5951b619008f826d	572776e85951b619008f8a81	572a135daf94a219006aa7a0	5726e08e5951b619008f810f
5726ddf6f1498d1400e8ee04	572758c3dd62a815002e9b78	5726edecdd62a815002e957e	57275250708984140094dc27
5726dcbddd62a815002e9321	5726f2375951b619008f8311	5726ed6cf1498d1400e8f00f	5726e3c4dd62a815002e9404
5726f2375951b619008f830f	57269e8a5951b619008f77f5	5728eff82ca10214002daadf	57273887dd62a815002e99a3
5726ed6cf1498d1400e8f00c	5726f36cdd62a815002e9601	5726ea985951b619008f8265	5726f868dd62a815002e9685
5726c20fdd62a815002e8fa7	572756715951b619008f8879	5726eb4b5951b619008f826c	5726e313f1498d1400e8eeb6
572758c3dd62a815002e9b7b	5726ddf6f1498d1400e8ee08	572771a5f1498d1400e8f841	5726a21bf1498d1400e8e57b
5726d9935951b619008f7ff0	5726ed6cf1498d1400e8f010	5727580bf1498d1400e8f69d	5726c20fdd62a815002e8fa6
5727515af1498d1400e8f63c	5726a00d708984140094cc26	5726d9935951b619008f7fee	5726fa525951b619008f83f7
5726e985dd62a815002e94dc	5726ab47f1498d1400e8e6a4	5726ddf6f1498d1400e8ee07	5727515af1498d1400e8f63d
5726a00d708984140094cc29	5726f36cdd62a815002e9602	5726a21bf1498d1400e8e579	5726e179dd62a815002e93af
5726e313f1498d1400e8eeb2	5726eb4b5951b619008f826b	5726f2375951b619008f8310	5726e65e708984140094d540
5726db5add62a815002e92d5	5727525070898414009 4dc29	5726db5add62a815002e92d7	5726ef73f1498d1400e8f05e
5726a784708984140094cd01	5726ed6cf1498d1400e8f00e	5728d5793acd2414000dffb7	572908c13f37b31900477fbf
572908c13f37b31900477fbe	5728fb6a1d04691400778ef9	5729efab3f37b319004785cf	5728d6f02ca10214002da912
572a0b0b6aef0514001551f8	5728dddc2ca10214002da9d5	5729081d3f37b31900477faf	5729f39a6aef051400155150
57280f974b864d1900164374	5728fc2eaf94a219006a9ec7	5729081d3f37b31900477fab	5728f2e26aef051400154898
572a142e3f37b319004786bb	57274f67708984140094dbf8	572a0bf96aef051400155206	572a12386aef051400155236
572a03086aef0514001551a4	5728202c4b864d19001644ee	572a07fc6aef0514001551de	5729f8516aef05140015516d
5728dafe3acd2414000e0060	572905ce1d04691400778f84	5728d9403acd2414000e001d	57271f125951b619008f8639
57282dfb4b864d1900164669	5728d5793acd2414000dffb4	5728fc2eaf94a219006a9eca	572753335951b619008f8856
572900f73f37b31900477f6c	5728151b4b864d1900164428	5728d5793acd2414000dffb3	5729f9953f37b31900478620
572822233acd2414000df557	5728dafe3acd2414000e005e	57274e0d708984140094dbe8	572754fff1498d1400e8f661
5729f9953f37b31900478621	572812e74b864d19001643cf	57273f27dd62a815002e9a0a	5729f799af94a219006aa709
572a02483f37b3190047864b	5726eb8bf1498d1400e8efe3	572a07fc6aef0514001551dd	5728fb6a1d04691400778ef8
5728fd206aef05140015494e	5728fb6a1d04691400778ef7	5726eb8bf1498d1400e8efe6	5729fa40af94a219006aa70f
572a06af3f37b31900478668	572900f73f37b31900477f6b	5728fb6a1d04691400778ef5	5729efab3f37b319004785d0
572742bd5951b619008f8786	5729f06f1d04691400779675	5726eb8bf1498d1400e8efe5	572a0ce11d04691400779700
572a10cd6aef051400155223	5728ebcb3acd2414000e01db	5729f39a6aef05140015514f	572900f73f37b31900477f6d

Continued on next page

Table B.1 : List of data points throwing errors when calculating SHAP values. (Continued)

5729efab3f37b319004785d2	572a142e3f37b319004786b9	57280f974b864d1900164373	572a0ce11d04691400779701
5728fc2eaf94a219006a9ec8	5728dc2d3acd2414000e007f	5728d6f02ca10214002da911	572908c13f37b31900477fc0
57274971708984140094dbbd	5729d609af94a219006aa665	5727ff083acd2414000df1ab	5727ee372ca10214002d99ee
57274f49f1498d1400e8f620	5727ec062ca10214002d99b8	5727cd0f4b864d1900163d72	572757bef1498d1400e8f690
5729e1101d04691400779645	572756fe708984140094dc74	5727c3b02ca10214002d95bb	5727b0892ca10214002d93e8
5727e8424b864d1900163fc1	5729f1283f37b319004785d7	572750df5951b619008f8830	572756fe708984140094dc73
5729e4291d04691400779654	57274a1edd62a815002e9a9a	5727ee372ca10214002d99ef	5727d0f73acd2414000ded13
5727c0402ca10214002d9565	5729e1e36aef0514001550be	572759665951b619008f8887	572a1fe16aef0514001552d0
57281f203acd2414000df4f6	5729f4e46aef051400155156	5727ffb5ff5b5019007d9a8e	572a11663f37b31900478697
572a20816aef0514001552e8	572a1046af94a219006aa78f	572a05eb3f37b31900478653	572a20816aef0514001552e7
572a13841d0469140077973e	572813b52ca10214002d9d68	572a0bafaf94a219006aa765	572a1046af94a219006aa78d
572a1a5c6aef051400155285	5727ffb5ff5b5019007d9a8d	572847ff3acd2414000df86d	57283e652ca10214002da16a
57284b904b864d19001648e3	57282036ff5b5019007d9d9f	572a0d21af94a219006aa783	572a1dbb3f37b319004786f8
572825714b864d1900164591	572a1ba46aef051400155290	572a13841d0469140077973d	57283e652ca10214002da167
5728710c3acd2414000df9f1	57287338ff5b5019007da236	5729046aaf94a219006a9f4d	5728705c2ca10214002da35b
572909406aef0514001549dd	572913626aef051400154a33	57284d484b864d1900164904	57287c142ca10214002da3d0
57286b003acd2414000df9c3	57287338ff5b5019007da232	572918bd3f37b31900478015	57286fa83acd2414000df9e9
572914441d04691400779025	57291b461d04691400779049	57287e512ca10214002da3f8	5728809f2ca10214002da40c
572867543acd2414000df9a1	5728f9342ca10214002dab52	5728759cff5b5019007da25e	57286c8cff5b5019007da21c
57287fec4b864d1900164a3c	57290b21af94a219006a9fd2	572867543acd2414000df9a5	572883153acd2414000dfa71
5728661e2ca10214002da2ea	57290d811d04691400778fd1	5728683b3acd2414000df9b1	572882242ca10214002da422
5728742cff5b5019007da246	5728742cff5b5019007da248	5728683b3acd2414000df9b0	57286ead2ca10214002da349
572863c72ca10214002da2d5	5728683b3acd2414000df9af	572867543acd2414000df9a4	572864542ca10214002da2e1
572883153acd2414000dfa72	57286ead2ca10214002da34a	57287ee3ff5b5019007da275	57287c142ca10214002da3d1
572881022ca10214002da419	5728855d3acd2414000dfa90	57286192ff5b5019007da1df	572867543acd2414000df9a2
57291beb1d04691400779054	57287e512ca10214002da3fb	572881022ca10214002da416	5729046aaf94a219006a9f4f
572909406aef0514001549dc	5728742cff5b5019007da249	572881d34b864d1900164a5e	57287e512ca10214002da3f9
57286d4f2ca10214002da329	57286951ff5b5019007da212	5728fea1af94a219006a9ef6	572871bd3acd2414000dfa04
572963221d04691400779386	572975073f37b31900478415	57296c5c3f37b31900478381	572956c86aef051400154d1d

Continued on next page

Table B.1 : List of data points throwing errors when calculating SHAP values. (Continued)

57293bc91d0469140077919f	57296c5c3f37b3190047837f	57296e43af94a219006aa3e8	5729645b3f37b31900478322
572957361d046914007792d1	5729582b1d046914007792e7	572926086aef051400154ac4	57296eee6aef051400154e91
57296fd71d04691400779440	57296f85af94a219006aa406	572973ccaf94a219006aa44a	5729645b3f37b31900478321
572957ad1d046914007792da	57297427af94a219006aa453	57296fd71d04691400779443	572975073f37b31900478418
57296de03f37b3190047839b	572963876aef051400154dd3	572970916aef051400154ebe	572968cf1d046914007793cd
572970916aef051400154ebc	57296f293f37b319004783a4	572975511d046914007794a8	572958cc6aef051400154d2c
5729582b1d046914007792e4	572957361d046914007792d0	572963876aef051400154dd5	572965e73f37b3190047832b
572927d06aef051400154adf	5729686d1d046914007793c5	572974923f37b3190047840e	57296eee6aef051400154e90
57297725af94a219006aa49b	5729582b1d046914007792e3	57296eee6aef051400154e92	572965566aef051400154e03
572966ebaf94a219006aa395	57296eb01d04691400779437	572968cf1d046914007793ce	57294baaaf94a219006aa26d
57296d1b1d0469140077940d	57296de03f37b3190047839d	5729703d3f37b319004783bb	572967e31d046914007793b3
572970916aef051400154eba	57296f85af94a219006aa403	572963221d04691400779385	572963221d04691400779389
57297427af94a219006aa456	57295b5b1d04691400779316	572976cfaf94a219006aa496	57296f85af94a219006aa404
57296f85af94a219006aa407	57296de03f37b3190047839e	572974923f37b3190047840f	572962953f37b319004782f8
572956c86aef051400154d1e	57292046af94a219006aa0bb	5729789b6aef051400154f6f	57296eb01d04691400779439
5729544c3f37b31900478259	5729779b6aef051400154f63	5729506d6aef051400154cad	572925491d046914007790c3
57296eb01d04691400779436	572970916aef051400154ebb	5729789b6aef051400154f6e	572957361d046914007792d3
572968cf1d046914007793cf	572978e66aef051400154f77	572975511d046914007794aa	5729686d1d046914007793c4
57293f353f37b3190047819c	572975511d046914007794a7	572963221d04691400779387	572956c86aef051400154d1c
572956c86aef051400154d1b	572958cc6aef051400154d2d	57296d8d1d04691400779421	572962953f37b319004782f7
5729723c6aef051400154eeb	57296d8d1d0469140077941e	572975511d046914007794a9	57297103af94a219006aa424
572967e31d046914007793b1	572924b53f37b31900478067	5729582b1d046914007792e6	572965566aef051400154e01
57296bf96aef051400154e52	57293bc91d0469140077919c	57294e6b1d04691400779275	572958cc6aef051400154d2b
572970916aef051400154ebd	572976791d046914007794b0	5729703d3f37b319004783bc	57293f8a6aef051400154bdf
57296de03f37b3190047839c	572963876aef051400154dd2	57296eee6aef051400154e8f	57296a65af94a219006aa3c4
57296d1b1d0469140077940f	5729735c3f37b319004783ff	5729686d1d046914007793c2	572963876aef051400154dd6
5729686d1d046914007793c1	572976cfaf94a219006aa493	5729703d3f37b319004783bf	572967e31d046914007793b5
572973ccaf94a219006aa44d	57296b151d046914007793f4	5729582b1d046914007792e5	57296eb01d04691400779435
572957ad1d046914007792dd	57297103af94a219006aa426	5729789b6aef051400154f70	572961f61d0469140077935b

Continued on next page

Table B.1 : List of data points throwing errors when calculating SHAP values. (Continued)

57294baaaf94a219006aa26b	57296d8d1d0469140077941f	57297427af94a219006aa454	57297991af94a219006aa4ba
5729784b1d046914007794c9	572962953f37b319004782f6	572928bf6aef051400154af2	57295a116aef051400154d48
5729729a1d0469140077948f	5729686d1d046914007793c3	572957361d046914007792cf	57296eee6aef051400154e8e
572965566aef051400154e02	57300a9a04bcaa1900d77067	57297bc9af94a219006aa4cb	57298ef11d0469140077952d
572996c73f37b319004784b4	572ff56304bcaa1900d76f2e	572f6ec7a23a5019007fc624	572fbfa504bcaa1900d76c74
57300580b2c2fd1400568750	572faf74b2c2fd1400568348	572989846aef051400154fc3	5730042804bcaa1900d77013
5729a03f1d04691400779595	57299ec43f37b3190047850e	57299c2c6aef051400155020	57297bc9af94a219006aa4c7
572996c73f37b319004784b6	57299c2c6aef051400155022	572fb059947a6a140053cb83	57299ec43f37b31900478510
572f6c85947a6a140053c943	572ff56304bcaa1900d76f30	57299d1c1d04691400779582	572faec7b2c2fd1400568335
57299d1c1d04691400779581	572f58d9a23a5019007fc580	57297d421d046914007794e5	572fadcbb2c2fd140056832b
57299021af94a219006aa50f	572ff430a23a5019007fcbaa	57299021af94a219006aa50b	572ff56304bcaa1900d76f31
57299326af94a219006aa518	57297ed93f37b31900478460	5730042804bcaa1900d77011	572fbb04a23a5019007fc8f8
572f7b33947a6a140053c9a4	572985011d04691400779502	5730069004bcaa1900d7702f	572f7588947a6a140053c988
572ffb02b2c2fd14005686b7	57300888b2c2fd1400568778	572faf74b2c2fd140056834b	5729a03f1d04691400779594
572ff4ca04bcaa1900d76f27	572facb0a23a5019007fc867	57299c2c6aef051400155021	572fad30a23a5019007fc86e
572985011d04691400779503	572fe288a23a5019007fcad9	572f6ec7a23a5019007fc623	572f59b4a23a5019007fc587
573003dd947a6a140053cf43	572fbfa504bcaa1900d76c77	572f58d9a23a5019007fc57f	572ff293947a6a140053ce56
57299c2c6aef051400155024	57300bf504bcaa1900d7708a	572ff935b2c2fd140056869c	572ff5fcb2c2fd140056865b
5730042804bcaa1900d77012	572ffb02b2c2fd14005686b8	57298ef11d0469140077952f	573009a004bcaa1900d77051
5729a03f1d04691400779596	572fff45947a6a140053cf2a	572fe92204bcaa1900d76e98	572faf74b2c2fd140056834a
5730042804bcaa1900d77014	572996c73f37b319004784b3	57300888b2c2fd1400568779	572ff56304bcaa1900d76f2f
572973f76aef051400154f0b	572faf74b2c2fd1400568347	57297a276aef051400154f89	572facb0a23a5019007fc865
57300911947a6a140053cfb9	57308f6b8ab72b1400f9c580	57309446396df919000961bc	5730909d8ab72b1400f9c58e
573092088ab72b1400f9c596	573088da069b53140083216e	57300137b2c2fd140056871a	57309ef18ab72b1400f9c603
573083dc2461fd1900a9ce6e	572fffb1b2c2fd14005686fd	572ff932a23a5019007fcbd6	573099ee8ab72b1400f9c5de
573027d6a23a5019007fcea1	573011de04bcaa1900d770fa	573010fab2c2fd14005687d7	572fdc34a23a5019007fca97
57309ef18ab72b1400f9c601	572ffabf04bcaa1900d76fa1	573088da069b53140083216b	572fda6fb2c2fd140056850f
572ff932a23a5019007fcbd7	572ff86004bcaa1900d76f67	57300200b2c2fd1400568729	5730005db2c2fd1400568706
573010fab2c2fd14005687db	573083dc2461fd1900a9ce71	57308ddc396df919000961a5	572ff626947a6a140053ce92

Continued on next page

Table B.1 : List of data points throwing errors when calculating SHAP values. (Continued)

572fd73e947a6a140053cd32	57309ef18ab72b1400f9c604	5730b255396df919000962b1	572ffd9e04bcaa1900d76fc7
57300200b2c2fd140056872a	5730b8ca8ab72b1400f9c705	573010fab2c2fd14005687d9	573083dc2461fd1900a9ce70
5730005db2c2fd1400568705	5730bb522461fd1900a9d011	5730131c947a6a140053d055	573020f7b2c2fd14005688f7
573085ea8ab72b1400f9c550	5730876a396df9190009617c	5730bb522461fd1900a9d012	5730b6592461fd1900a9cfd0
57309446396df919000961ba	572ffd9e04bcaa1900d76fc8	57309bfb8ab72b1400f9c5e8	5730b8ca8ab72b1400f9c708
5730088e947a6a140053cfac	572ff932a23a5019007fcbd5	573011de04bcaa1900d770fd	57308ddc396df919000961a7
572ff86004bcaa1900d76f68	572ff626947a6a140053ce8f	5730b6592461fd1900a9cfd2	573088da069b53140083216c
572ff626947a6a140053ce90	5730876a396df9190009617d	572fc5a1947a6a140053cc8b	

Appendix C

Distance densities

In the next pages, we showcase the plots of distance densities of each of the four hypotheses of Chapter 3, at different thresholds. The x-axis is for the distance value, while the y-axis is used to display the frequency of that value.

Results for the biased and the unbiased model are presented side by side; the tuples that satisfy the hypothesis are coloured in blue, while those that only satisfy the A-part of the hypothesis are coloured in orange. On the top of each couple of plots, the threshold is specified together with the condition, which is always `average`, as for sentence contribution we averaged between starting and ending values (see Definition 2.8 and Footnote 18 of Chapter 2).

These plots help us to visualize the difference between $\mathcal{C}_\theta$ and $\mathcal{C}_A$ in an intuitive way: if the two colours are well separated, AUC will be high, whereas, if the two have a lot of overlap, AUC will be low.

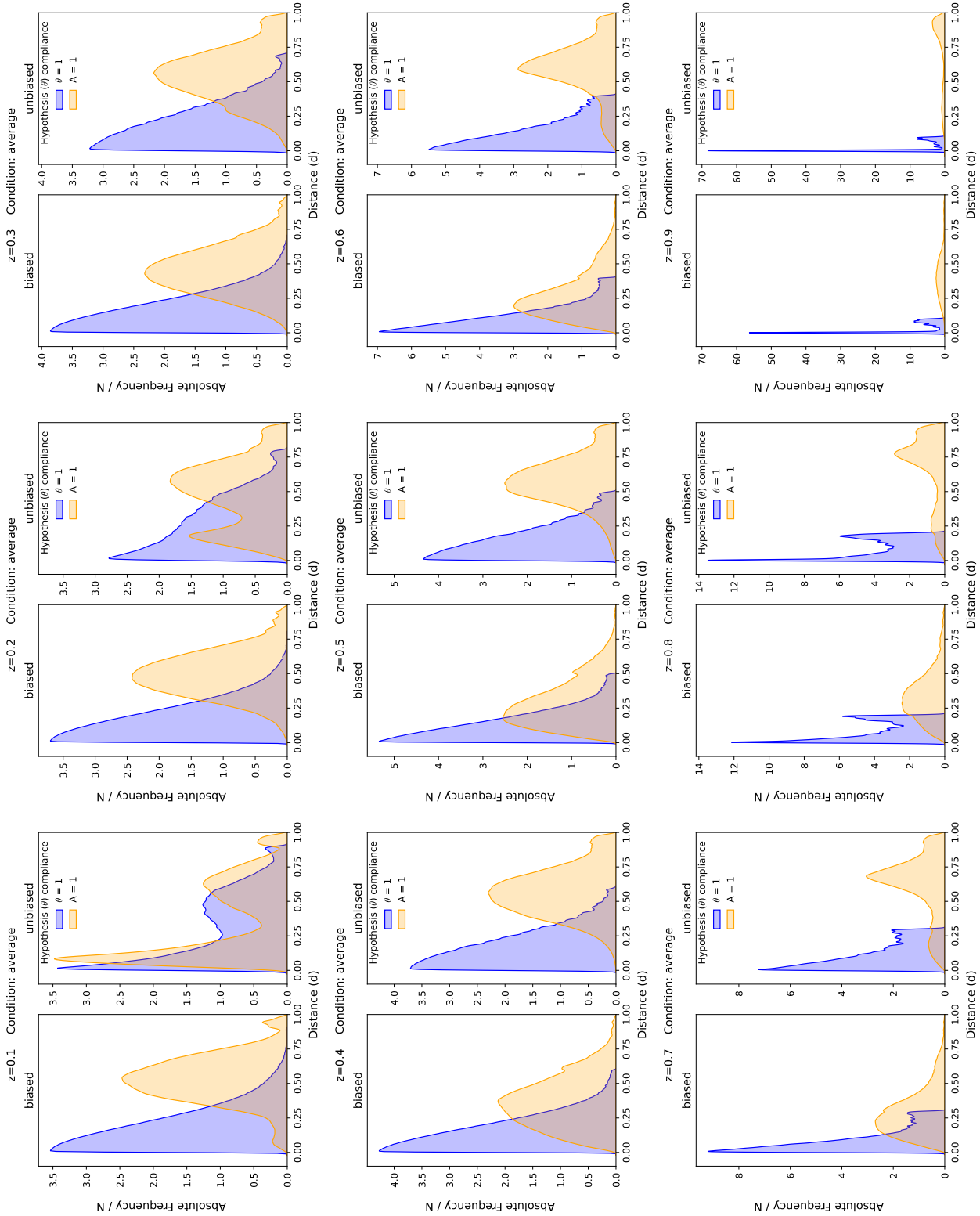

Figure C.1: Distance densities plots of biased and unbiased model for θ_1 at different thresholds.

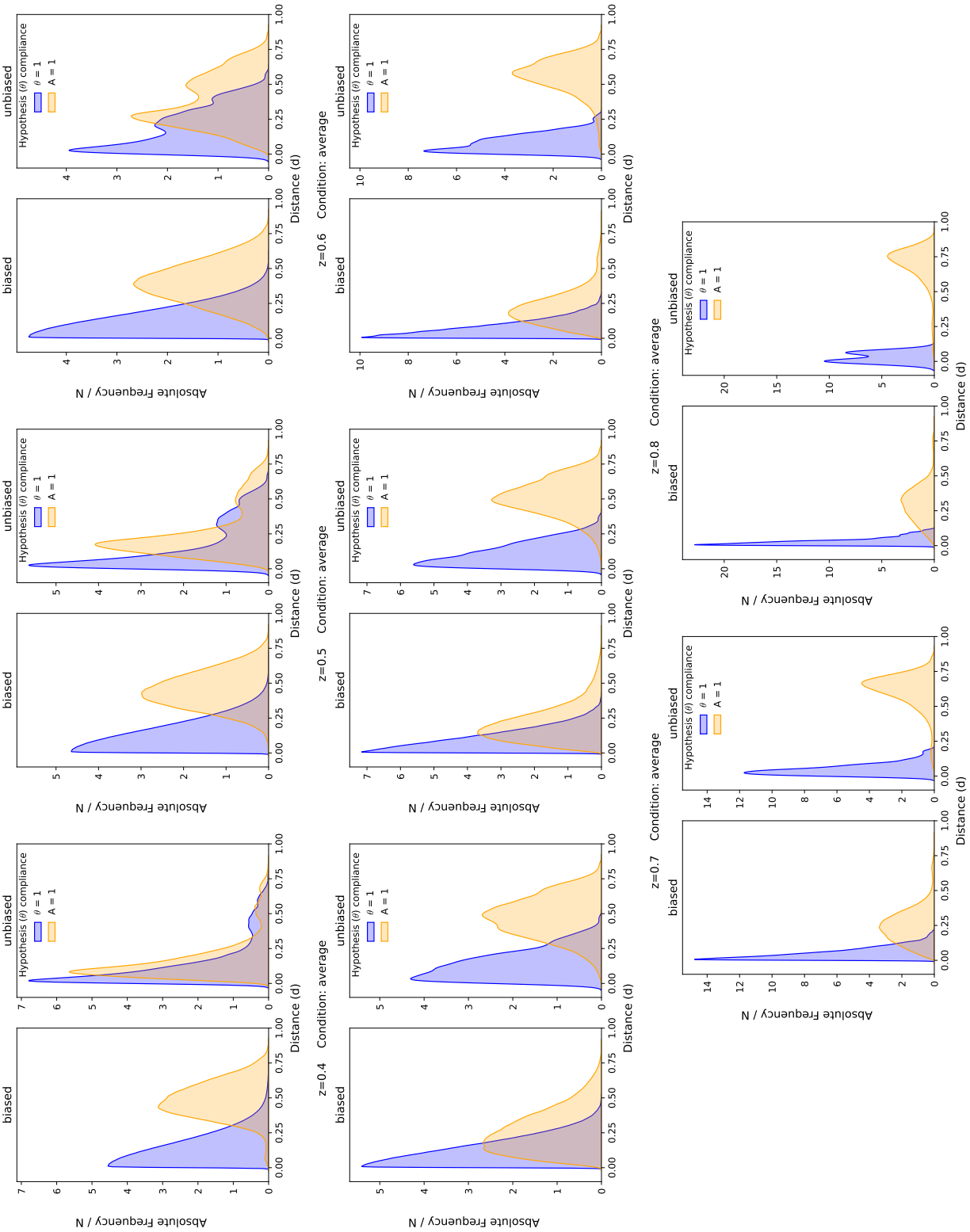

Figure C.2: Distance densities plots of biased and unbiased model for θ_2 at different thresholds.

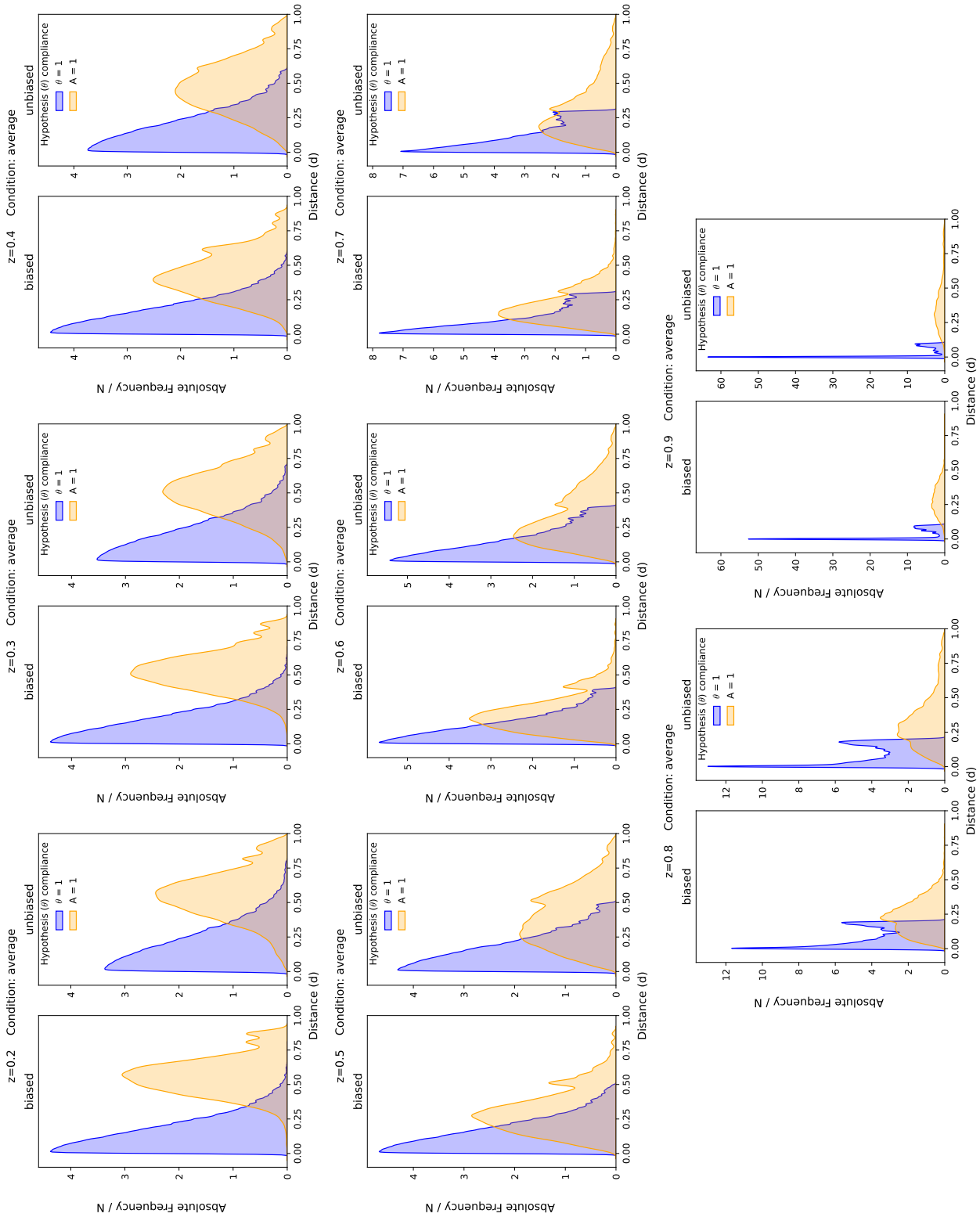

Figure C.3: Distance densities plots of biased and unbiased model for θ_3 at different thresholds.

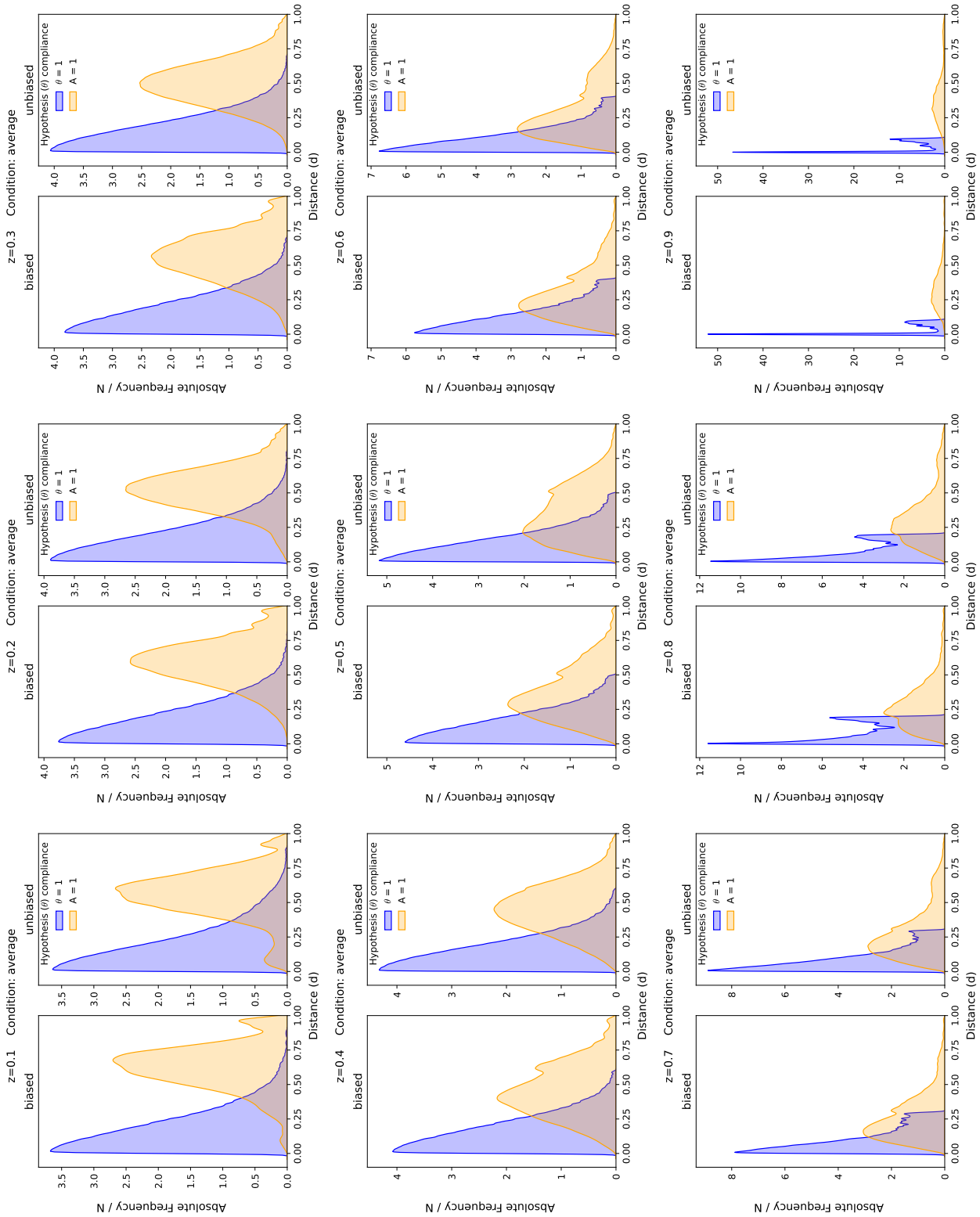

Figure C.4: Distance densities plots of biased and unbiased model for θ_4 at different thresholds.

Appendix D

Additional results considering the question

In the main text we had only short space to discuss the inclusion of the question in the processed contribution, so we provide more detail here.

Previously, we excluded the question from the calculation of sentence contribution as we assumed that both model should focus on it equally. Inspecting the raw SHAP values 2.2 we had evidence to support this assumption, as, for both models, the top three tokens by SHAP values were all part of the question.
To be sure, we reprocessed SHAP values following Definition 2.8, this time extending the mapping $\mathfrak{m}$ so that it includes the question as the first sentence. Evaluating once again θ_1 with the new contribution vectors we obtained the plots of Figure D.1.
Firstly, we observe that now thresholds higher than 70% are not present in the plot, so the first sentence contribution is at most $< 80\%$, which in turn means that the question always contributes, and for both models, at least 20% to the prediction, in line with our expectations.

Furthermore, MD and AUC of both models closely resemble the trends exposed in Section 3.2.1: MD of the biased model has a flat trend until 40%, and afterwards it increases linearly, signalling that often the model attributes around 40% of contribution to that sentence. AUC is in line with this, dipping at the same threshold because there are data points with close to 40% contribution, and that for $z = 40\%$ do not satisfy the B-part of θ_1. Considering that the question has contribution of at least 20%, the contribution that the first sentence receives relative to the other sentences is still very high.

The unbiased model also exhibits trends comparable to Figure 3.1 and 3.2: MD linearly increases with z, and AUC is high across all thresholds. The minimum MD is reached at 10%, where we also have a highly negatively skewed distribution for AUC, indicating that a lot of tuples have contribution close to 10%. The small variance at higher thresholds, as well as the linear increases in MD are evidence that at times $\mathfrak{f}^u$'s predictions receive a lot of contribution from the first sentence.

In summary, including the question does not change in any significant way the results and conclusions of Section 3.2.1. Furthermore, as the question contributes consistently (at least 20%) to the predictions of both models, it constitutes a confounding factor, if the goal is to analyse how the model's predictions are influenced by the context.

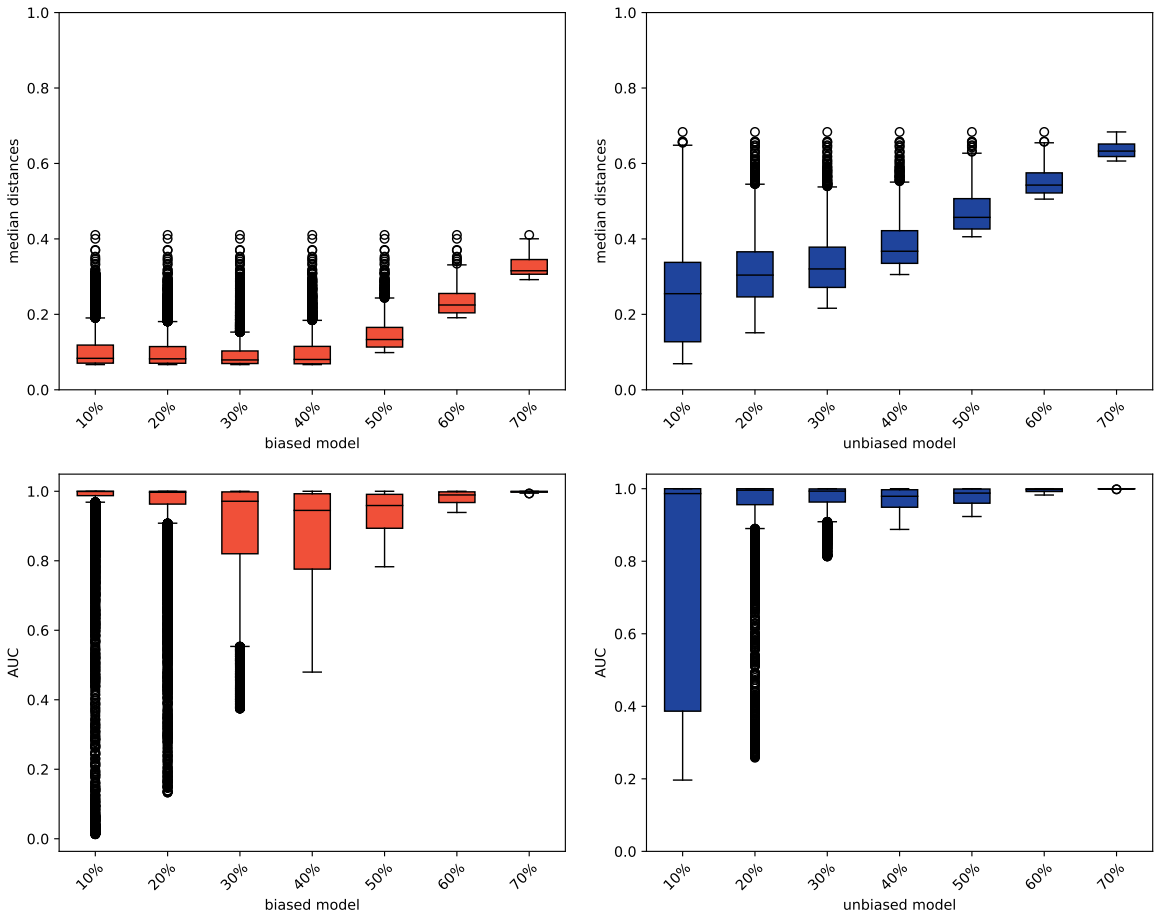

Figure D.1: MD and AUC for θ_1 with question. Results for the biased model are displayed in red on the left, while those for the unbiased are on the right in blue.

Appendix E

Further evaluations of MD and AUC

In this appendix we provide all the results obtained by choosing the sample and reference set for MD and AUC differently, as discussed at the beginning of Section 4.2. In the first section we fixed $\mathcal{S} = \mathcal{U}$ and $\mathcal{R} = \mathcal{C}_\theta$. In the second section we add the results for the hypothesis not covered in Section 4.2 for $\mathcal{S} = \mathcal{U}$ and $\mathcal{R} = \mathcal{C}_A$.

E.1 $\mathcal{U}$ from $\mathcal{C}_\theta$

For the first hypothesis, we obtain the same plots of Figure 3.1 and 3.2, as the constraint of θ_1 is $A \coloneqq \top$, and does not impose any condition on the tuples, we have $\mathcal{U} = \mathcal{C}_A$, so this evaluation coincides with the previous.

Hypothesis 2 We plot the obtained median distance and AUC in Figure E.1. In comparison to the previous images (Figure 3.3 and 3.4), MD plots look almost identical, with only a small, but relevant difference for $\mathfrak{f}^b$. At $z = 50\%$, adding to $\mathcal{S}$ the tuples where either the model is correct or the answer is in the first sentence further reduces the median and the quartiles, providing another perspective on the previous results: $\mathfrak{f}^b$ behaves coherently, attributing a lot of importance to the first sentence both on tuples that have the answer in that sentence and on those which do not; in other words, is not making any distinction based on the location of the answer.

AUC is here lower for both models: thus, using only the difference of contribution of the first sentence, it is harder to distinguish among all the tuples in $\mathcal{U}$, those where the models are incorrect and the answer is in the first sentence from the rest. Notice that at high thresholds, the AUC reaches nearly 1.0, which can be interpreted as evidence that, when contribution is really high, either the models are rarely wrong or they are mostly wrong. Inspecting the validity plot for this hypothesis (Figure 3.9, second row, on the left) confirms that the first alternative is the true one, for both models.
The fact that AUC increases steadily for the unbiased model is further evidence that the more $\mathfrak{f}^u$ focuses on a sentence, the less likely it is to be wrong, while we cannot say the same for $\mathfrak{f}^b$: the AUC trend is lower than the previous evaluation for all thresholds except $z \geq 80\%$, meaning that based on the amount of contribution that the first sentence receives, it is hard to predict on which tuples $\mathfrak{f}^b$ will answer correctly.

Hypothesis 3 Median distance and AUC for θ_3 are displayed in Figure E.2.

For the biased model, results are similar to the previous paragraph (Figure E.1). Flat AUC around 0.8 for all the thresholds before 70% and median distance also flat and around 0.2 until that same z, signal that only at very high thresholds first sentence contribution is a good predictor of $\mathcal{C}_\theta$; in turn, this means that also on tuples not in $\mathcal{C}_\theta$ the model is regarding the first sentence as very important (50% to 70%). Notice also that in comparison to Figure 3.5, the median distance is almost equal: adding *all* the tuples does not change the results, so the behaviour that the model displays on $\mathcal{C}_\theta$ and $\mathcal{C}_A$ is also common on the tuples which do not satisfy A.

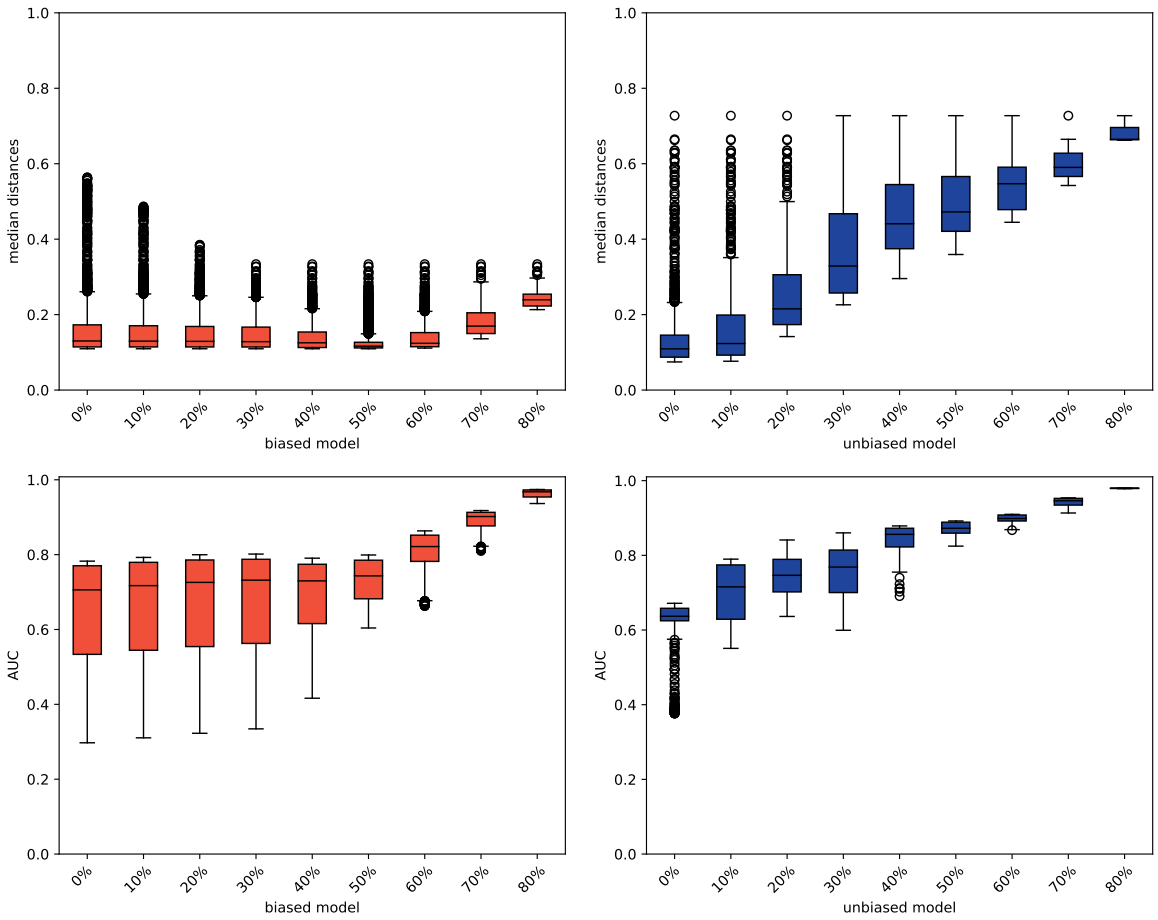

Figure E.1: Median distance (first row) and AUC (second row) for θ_2 calculated with $\mathcal{S} = \mathcal{U}$ and $\mathcal{R} = \mathcal{C}_\theta$. As usual, results for the biased model are displayed in red on the left, while those for the unbiased one are on the right in blue.

For $\mathfrak{f}^u$ median distance exhibit a trend similar to the one of $\mathfrak{f}^b$, though centred above 0.5. This means, on one side, that the unbiased model does not always attribute a lot of importance to the first sentence (MD is high), on the other, that the $\mathfrak{f}^b$ receives a lot of contribution from the first sentence when the answer is there and $\mathfrak{f}^u$ is correct (MD is flat for the first thresholds). The fact that AUC is high suggests that for $\mathfrak{f}^u$ the first sentence contribution is a good indicator if we want to predict whether a tuple is in $\mathcal{C}_\theta$ or not; in other words, if the contribution is high, the tuple will probably be in $\mathcal{C}_\theta$, whereas if the contribution is low, $\boldsymbol{u} \notin \mathcal{C}_\theta$. This confirms our expectations about $\mathfrak{f}^u$: it has learned to focus a lot on the first sentence only when the answer is there.

Hypothesis 4 In Figure E.3 we plot the semantic match metrics for the fourth hypothesis.

The median distance plots of θ_3 in the previous paragraph seem now inverted: the unbiased model has scores below 0.2 and an increase only after $z = 70\%$, while $\mathfrak{f}^b$ has a trend centred around 0.4. Then, $\mathfrak{f}^u$ is attributing a lot of contribution to the answer' sentence both when it is correct and when not. The biased model, on contrary, focuses a lot on the answer's sentence when it is correct, which, by all the previous results, only happens regularly when the answer is in the first sentence. It follows that when the unbiased model is wrong, it might not be due to the fact that it is looking for the answer in the wrong place, but because of other factors.

AUC of $\mathfrak{f}^b$ is high as the amount of contribution for the answer sentence is a good indicator of model correctness regardless of the threshold. This is because the biased model focuses consistently on the first sentence (50%), and thus, somewhat counter-intuitively, whenever the contribution of a sentence which is not the first is very high (70% or above), it is almost always the answer' sentence and the model is correct in its predictions. For the unbiased model, AUC has a linear trend with respect to z. The fact that AUC decreases for the lower

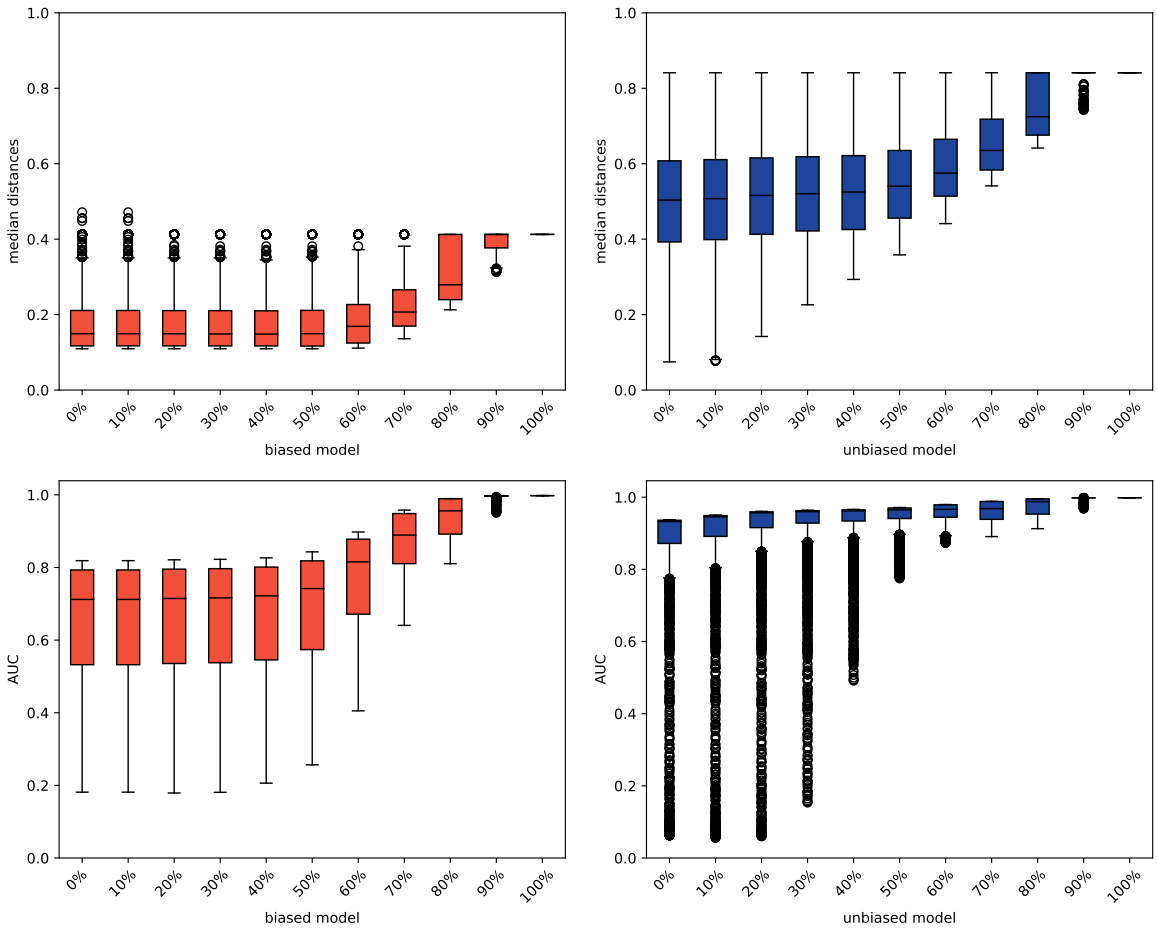

Figure E.2: Median distance and AUC for θ_3 calculated with $\mathcal{S} = \mathcal{U}$ and $\mathcal{R} = \mathcal{C}_\theta$.

thresholds when compared to the previous evaluation (Figure 3.7) suggests that when we add all the tuples of $\mathcal{U}$, distinguishing where $\mathfrak{f}^u$ is correct based on the contribution of the answer sentence becomes harder. Hence, at times, and as indicated by the high number of outliers for smaller z of AUC of $\mathfrak{f}^u$ in Figure 3.7, the unbiased model is making the correct predictions even if it is focusing less prominently on the answer' sentence. Therefore, while it is important that the model looks for the answer in the right sentence, it is not always necessary.

E.2 $\mathcal{U}$ from $\mathcal{C}_A$

We report here the evaluations of MD and AUC for the four hypothesis of Definition 3.1, with $\mathcal{C}_A$ as reference set and $\mathcal{U}$ as sample set. As for the previous section, θ_1 imposes no constraints in A, and thus re-evaluating MD and AUC using $\mathcal{C}_A$ is meaningless.

Hypothesis 2 Results for θ_2 are shown in Figure E.4. Interestingly both models have very low MD with tight quartiles and whiskers, and a lot of outliers; respectively, AUC is just below 0.8 for $\mathfrak{f}^b$, and just above 0.6 for $\mathfrak{f}^u$, in this case with very tight quartiles. Inspecting the distance density plots (Figure E.5) we see that the $\mathcal{C}_A$ set overlaps almost completely with the set of tuples that do not satisfy A. As the constraint of θ_2 imposed that the answer is not in the first sentence and the model's prediction is wrong, the above implies that it is impossible to distinguish those tuples only based on the amount of contribution of the first sentence (and the corresponding distance). While in light of the previous conclusions this might not be surprising for the biased model, it means that the unbiased model, when making wrong predictions, is not following a specific pattern based on the first sentence. On one hand, this is good, as it means that $\mathfrak{f}^u$ is not focusing on the first sentence when it should not,

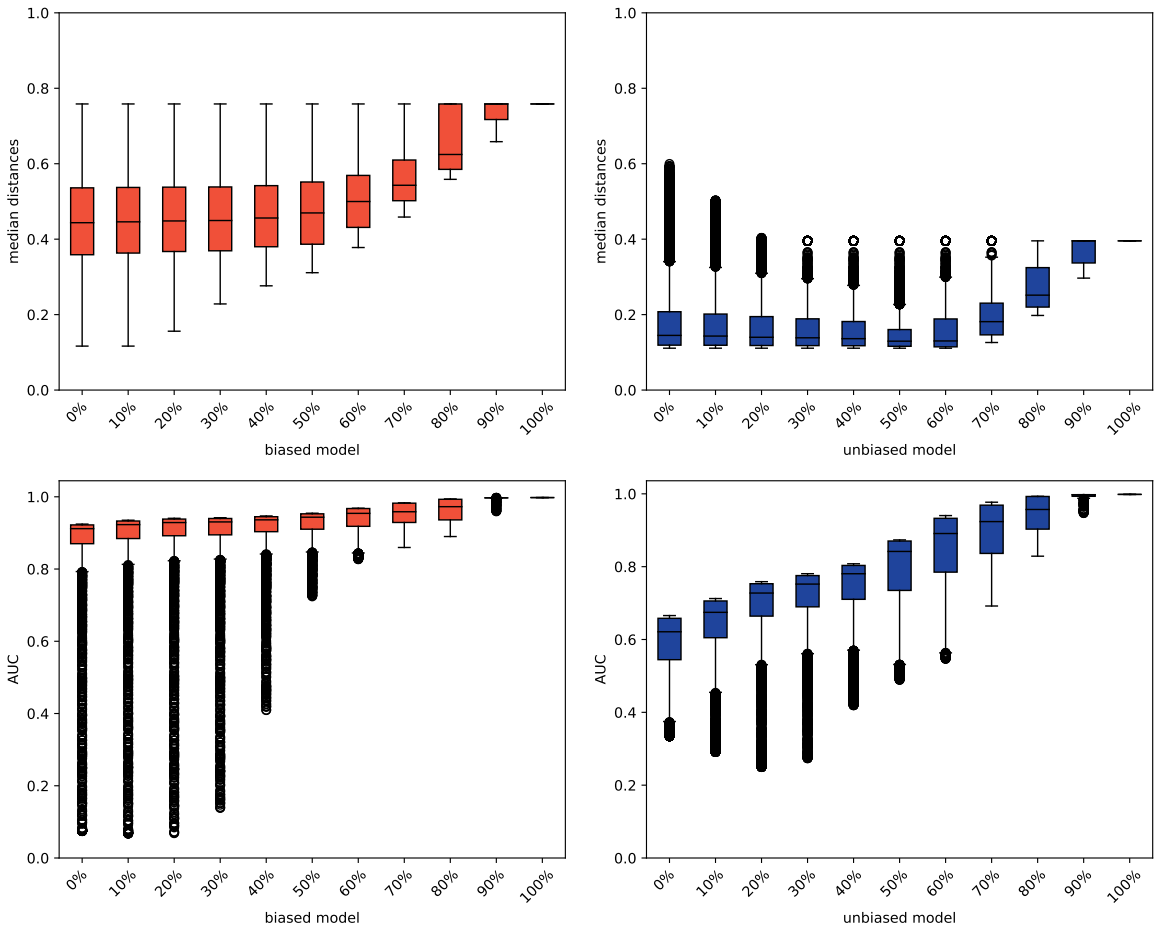

Figure E.3: Median distance and AUC for θ_4 calculated with $\mathcal{S} = \mathcal{U}$ and $\mathcal{R} = \mathcal{C}_\theta$.

on the other, this also implies that solving those errors or finding their cause might not be an easy task.

Hypothesis 3 The results for the third hypothesis, shown in Figure E.6 and E.7 are similar to those for θ_5, Figure 4.3 and 4.4, leading to similar conclusions.

Hypothesis 4 Finally, MD, AUC and distance density for θ_4 are presented in Figure E.8 and E.9. For this hypothesis the $\mathcal{C}_A$ set is separable for the biased model but not for the unbiased one. Indeed, MD of $\mathfrak{f}^b$ is around 0.5 with high variance, while AUC, albeit having a lot of outliers, is around 0.9, with very tight quartiles. On contrary, $\mathfrak{f}^u$ has MD under 0.2 and low AUC, just above 0.6, with wider quartiles when compared to $\mathfrak{f}^b$. Recall that in this case, the distance is defined as the difference between the contribution of the answer' sentence.

For the biased model, this can be explained by the fact that it often focuses on the first sentence, making the wrong prediction and attributing only low contribution on the answer' sentence. When the model is correct, it is frequently the case that the answer is in the first sentence, which also has high contribution, and thus, the tuples where the model is correct will have bigger distance from those where the model is not, giving good separability of $\mathcal{C}_A$. Note that, in general, this is not great: a good model should always receive a lot of contribution from the answer' sentence, while $\mathfrak{f}^b$ does that only sometimes.

The unbiased model instead does not separate well the tuples on which its predictions are correct from the others. This is both positive and negative: on one side, we expect a good model to focus on the correct sentence, and the low MD of $\mathfrak{f}^u$ confirms that this is happening. On the other, the fact that $\mathcal{C}_A$ is not separable by the distance of θ_4 suggests that the above behaviour is not sufficient to assure that the model's prediction will be correct.

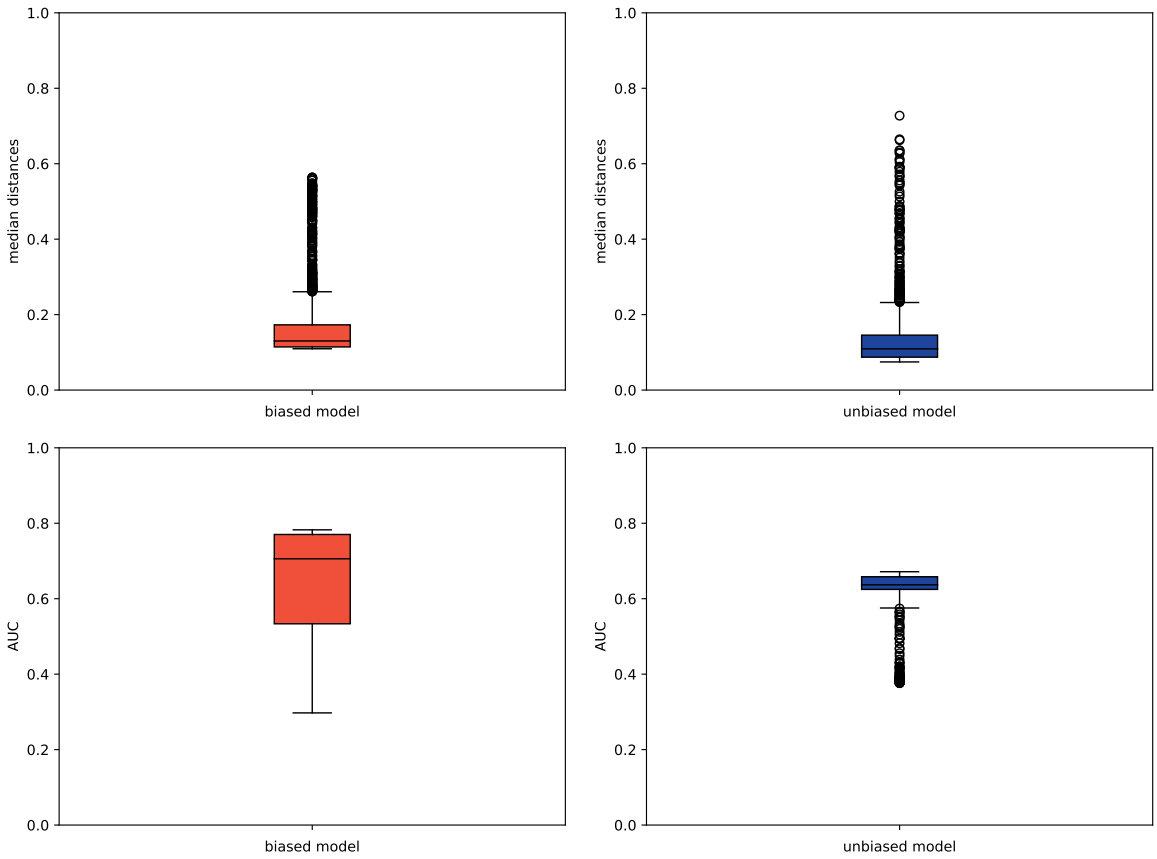

Figure E.4: MD (first row) and AUC (second row) plots for θ_2, for the biased (left column, red) and unbiased (right column, blue) models, calculated with $\mathcal{S} = \mathcal{U}$ and $\mathcal{R} = \mathcal{C}_A$.

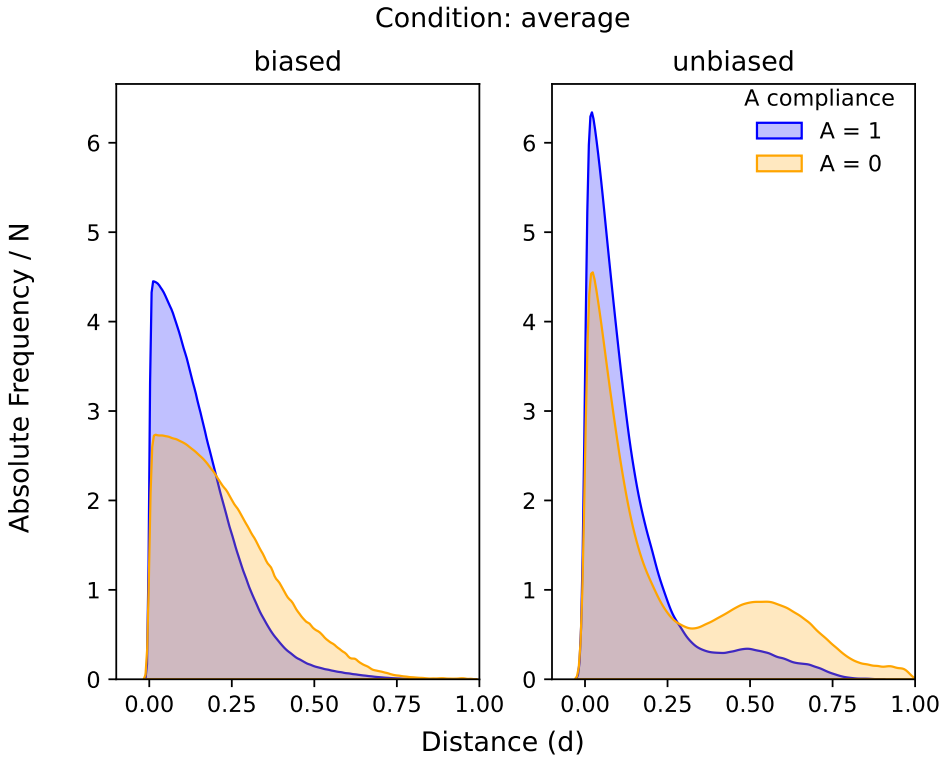

Figure E.5: Distance density plots for θ_2, for the biased and unbiased models.

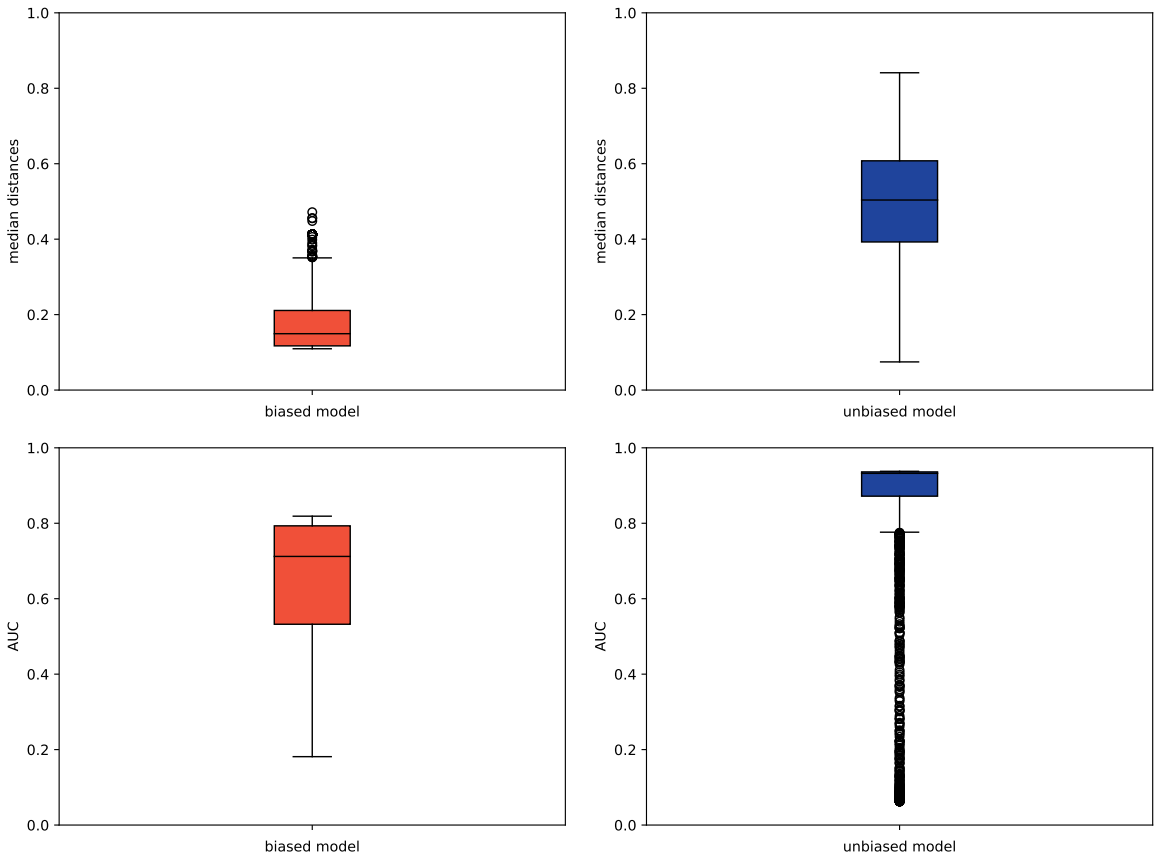

Figure E.6: MD (first row) and AUC (second row) plots for θ_3, for the biased (left column, red) and unbiased (right column, blue) models, calculated with $\mathcal{S} = \mathcal{U}$ and $\mathcal{R} = \mathcal{C}_A$.

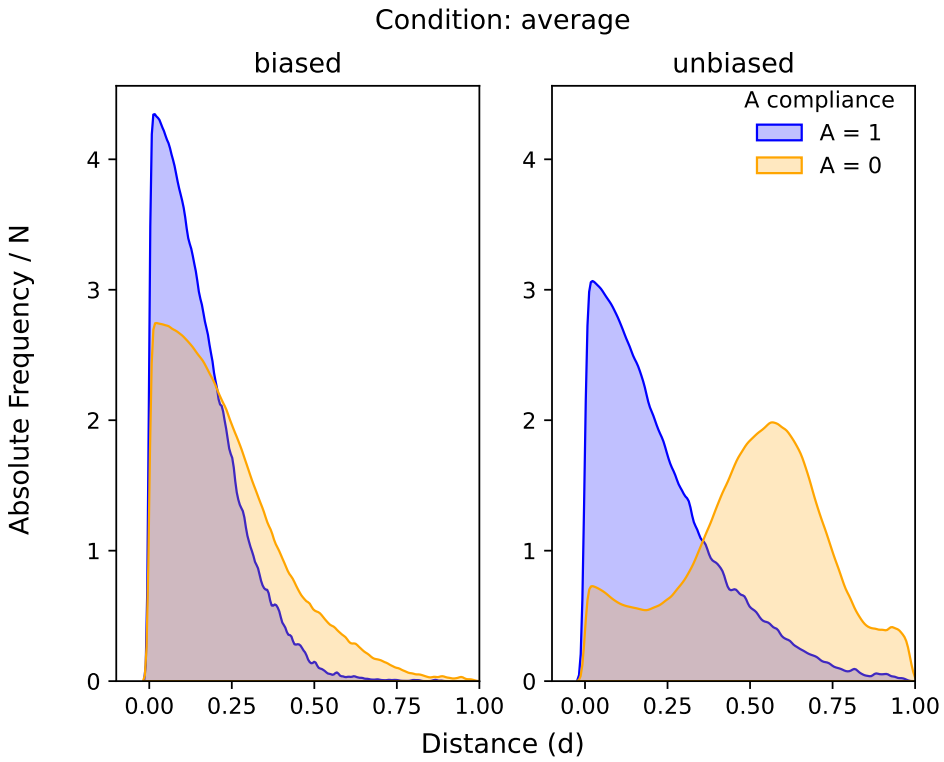

Figure E.7: Distance density plots for θ_3, for the biased and unbiased models.

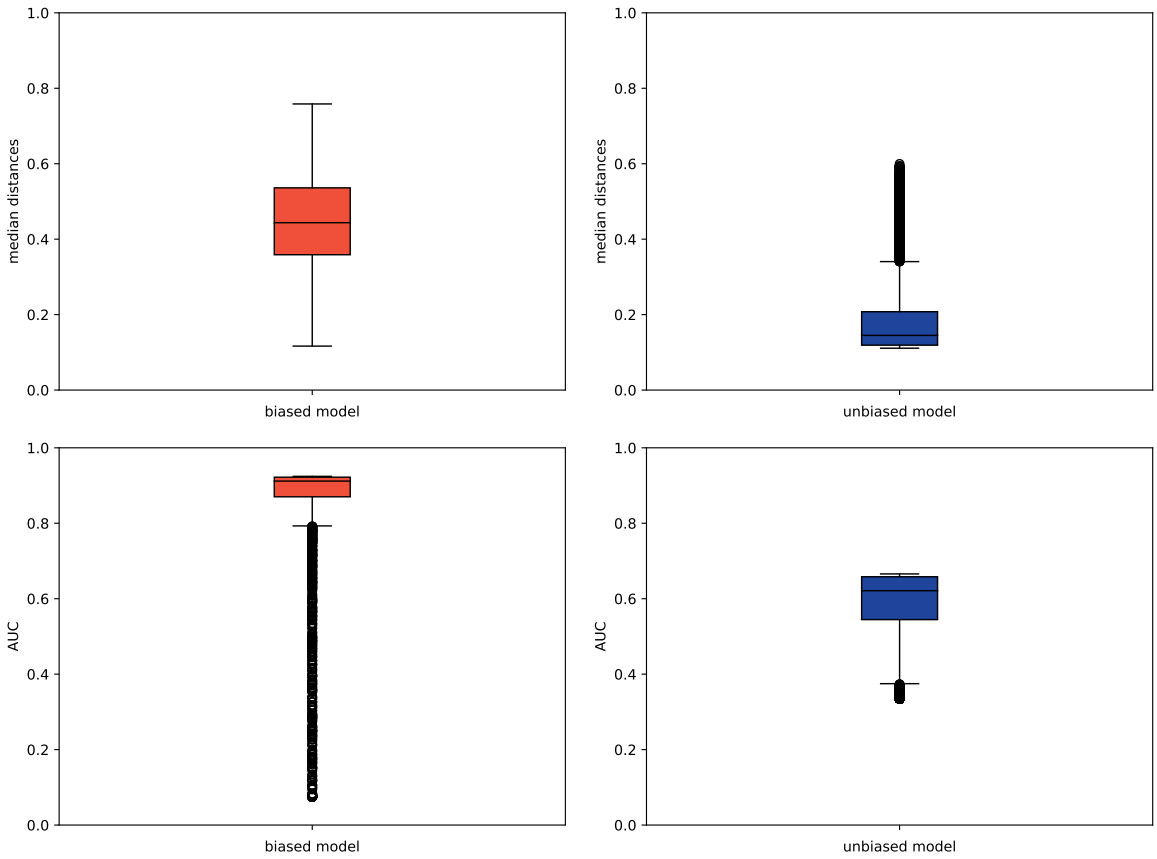

Figure E.8: MD (first row) and AUC (second row) plots for θ_4, for the biased (left column, red) and unbiased (right column, blue) models, calculated with $\mathcal{S} = \mathcal{U}$ and $\mathcal{R} = \mathcal{C}_A$.

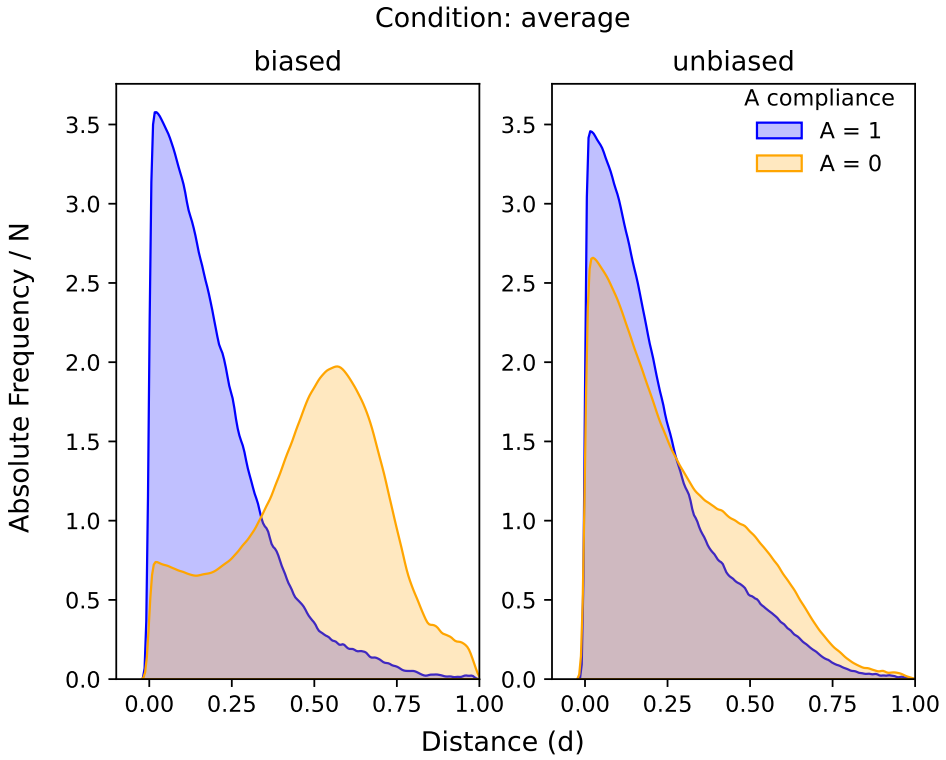

Figure E.9: Distance density plots for θ_4, for the biased and unbiased models.

Appendix F

Sign of hypothesis and metrics

In the main text we had little space to discuss how the definition of the behaviour B influences the results of the semantic match metrics. While this is still an open research question, and a systematic answer would benefit from a fully formalized syntax for the hypotheses, we add here a new version of θ_1, with switched sign, and compare the previous results with the scores for the new hypothesis, providing insights on the relation between the specification of θ and the results of the metrics.

The hypothesis we use here is thus:

$$\theta_6 \coloneqq \text{'The contribution of the first sentence is} \leq z\% \text{ of the total contribution.'}$$

$$\theta_6 \coloneqq \top \Rightarrow \mathfrak{c}_1(\boldsymbol{e}) \leq z$$

In Figure F.1 we compare MD scores for θ_1 and θ_6, while in Figure F.2 we relate AUC results. Both metrics are calculated by fixing $\mathcal{C}_A$ as sample set and $\mathcal{C}_\theta$ as reference set.

Interestingly, MD scores of the biased model seem to be a mirrored version of θ_1. As we saw, for $\mathfrak{f}^b$ the distribution of explanations is close to a normal distribution, with centre at 0.6. Switching the sign, the behaviour range will increase with z; the same reasoning used to explain the trend for θ_1 is still valid, but this time reversed: at low thresholds, the reference points will be fewer, and far from the median, leading to higher distances, while the more z increases, the bigger the reference set, and the closer the tuples in it are to the median, decreasing the distances. Surpassing 0.6, the median will remain low, with a flat trend, as the majority of the tuples falls near 0.6.

For the unbiased model, the rationale is the same, however, as its distribution is centred around 0.1, we observe low median distance across all thresholds. For the highest thresholds, the quartiles are wider, due to a secondary peak of the distribution around 0.7; those explanations fall further away from the main peak, having higher distances.

The AUC scores, for the biased model are also in line with our previous analysis, and resemble almost identically the plot for θ_1, albeit mirrored on z.
On the other hand, AUC of $\mathfrak{f}^u$ for the two hypotheses are quite different. Indeed, for θ_6 the median is always at 1.0, with no variance between $z = 30\%$ and $z = 70\%$, and a lot of outliers. This is due to the fact that θ_6 requires to have $\leq z$ contribution, and, as made clear from Figure 3.8, almost all tuples satisfy this constraint even at low thresholds (like 30%). At high thresholds, variance increases, and the distribution of AUC scores is skewed negatively, signalling that predicting θ_6-compliance with respect to some reference points is now harder. This happens because of the second spike in the distribution of explanations: as there are tuples with more than 0.7 contribution from the first sentence, the tuples with more than 0.8 contribution (which do not satisfy θ_6) now fall close to those that satisfy θ_6, making prediction harder. Similarly, the big number of outliers is also

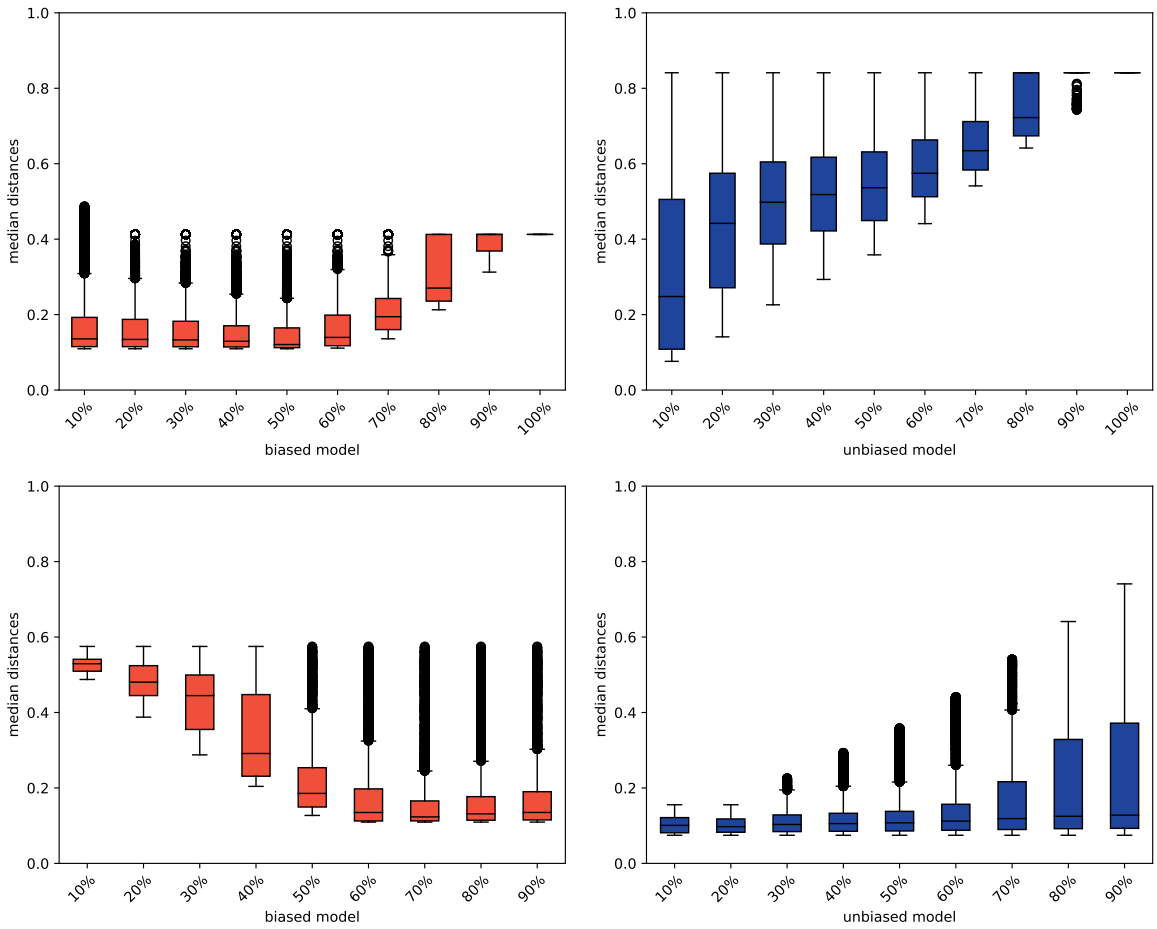

Figure F.1: In the first row, median distance of $\mathfrak{f}^b$ (red) and $\mathfrak{f}^u$ (blue) for θ_1. In the second row, median distance of the two models for θ_6.

due to tuples that actually have more contribution from the first sentence, which usually happens when that sentence contains the answer, as shown in Chapter 3.

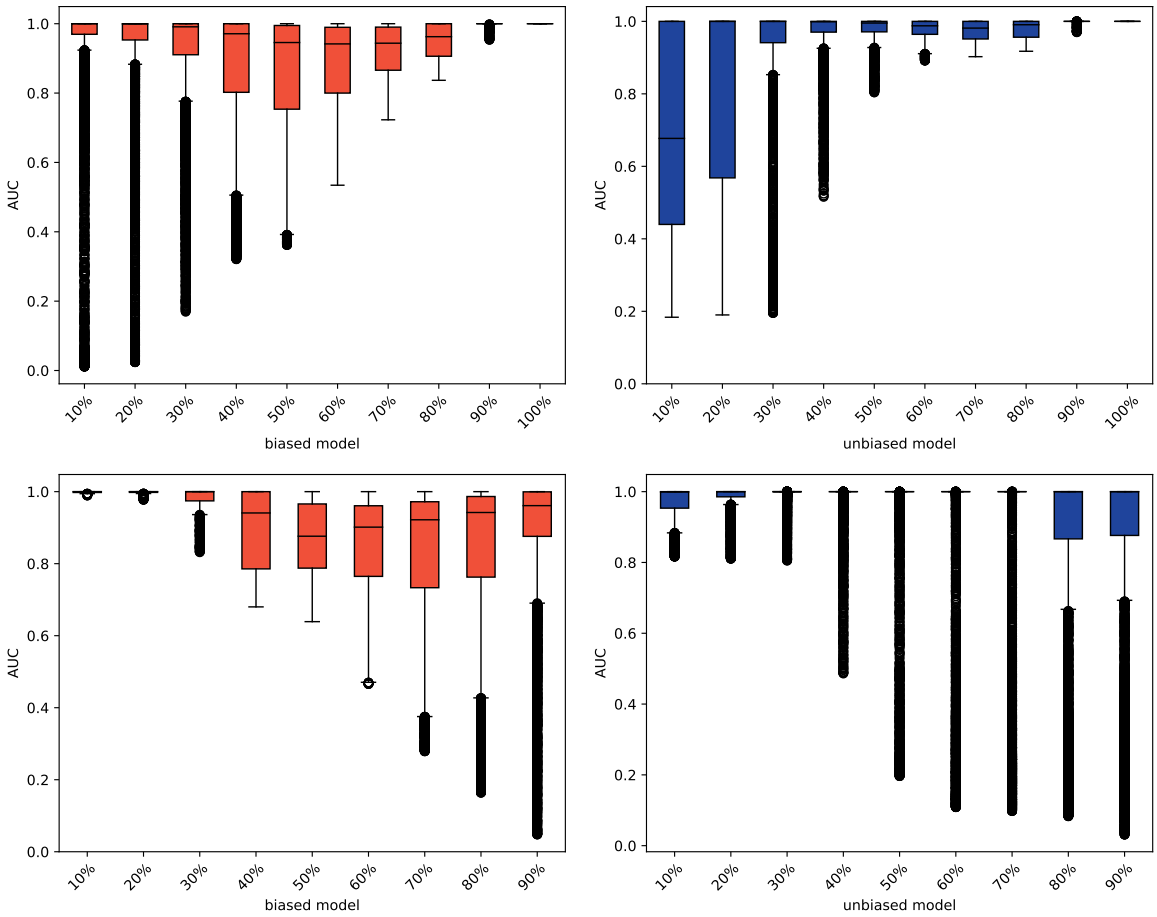

Figure F.2: In the first row, AUC of $\mathfrak{f}^b$ (red) and $\mathfrak{f}^u$ (blue) for θ_1. In the second row, AUC of the two models for θ_6.

Appendix G

Avoiding one more risk[1]

Zooming out and looking at the framework as a whole, one might rise concerns regarding the many choices that need to be made in order to measure semantic match: before applying AUC and MD, one has to choose an explanation method and to define a function to aggregate low-level features into high-level ones, a distance between the functions of explanations thus obtained, and, only after, a series of hypotheses to evaluate.[2]
Given that the ultimate goal of the framework is to properly explain the behaviour of a model without falling pray to confirmation bias, one might rise the concern that all these choices actually make the user vulnerable to the very same bias she is trying to avoid.

In the main text this issue was not discussed, as the objective of the thesis was to apply semantic match and to prove that it can be fruitfully employed on a state-of-the-art model to highlight a bias it may suffer from. As such, choosing the high-level feature of interest was a natural consequence of the induced bias, and so was choosing the distance. Nonetheless, in a more general environment, where less is known about the model, the risk of incurring in confirmation bias might still persist, if one applies the framework carelessly.
The root of the problem is this: because the framework is very *general*, it must be very *flexible*, and indeed, there are only few constraints on the feature aggregation function and on the distance function. However, this flexibility might be exploited to achieve desired results even against the evidence provided by the feature attribution methods.

Yet, the above risk can be neutralized in multiple ways.

The definition of the distance function, should, as a rule of thumb, only depend on features that are present in the considered hypothesis; intuitively, including more features increases the dimensionality, resulting in higher distance scores, even between very similar explanations. Including fewer features has the opposite effect, reducing the distance. Furthermore, there are standard candidates to score similarities between vectors, and having to define a new kind of distance just to get an acceptable match is a tip-off that we might already be suffering from confirmation bias.

The choice of high-level feature should also closely depend on the behaviour we want to match, with particular attention regarding the assumptions made about the model and its workings. To give an example, we used the relative contribution of the first sentence, calculated without including the question; this was motivated by the assumption that the question was important for both models, and never ignored. However, while it was a sensible assumption to make, it is something that needs to be checked, in order to avoid confirmation bias. Hence, we re-evaluated the hypotheses using the same distance function based on the relative contribution of the first sentence, now calculated including the question. And only because the results substantiated this hypothesis, we decided to go forward and drop the question from the definition of attention on a sentence[3]
Thus, in general, one should always try to make any assumptions underlying the definition of the high-level

[1] I owe inspiration for the following to the committee's questions and the resulting discussion during the public defense of this thesis.
[2] See Chapter 2, Section 2.1.3 in particular.
[3] See Appendix D.

feature explicit, and formulate specific tests to prove that they indeed hold.

In addition, it is possible to form a vocabulary of useful feature aggregators and distances across multiple experiments in different modalities: indeed, our mappings for sentences work in a very similar manner as the bounding boxes introduced to measure model's attention on part of an image in the experiments of [11]. A similar vocabulary could then be carried on across multiple modalities, to suggest the initial choices of the experiment or to build a baseline reference.

Finally, in crucial experiments one could add a phase of calibration of the framework to further limit the arbitrariness of the initial choices: instead of starting directly with the black-box model that needs explanation, we could begin by training a white-box model for the same dataset and task. Then, formulate hypotheses that reflect the knowledge we have about the inner workings of the white box model, and experiment with different choices of distances and high-level features, to see with which we get the best match between the hypotheses, the behaviour, and the knowledge we have of the model. Since the hypotheses encode that knowledge, they should display high semantic match, and if they do not, we have a clear sign that the distance or the high-level feature considered is somewhat ill-defined, and should not be used to evaluate the black-box model. On the other hand, if high semantic match is observed, we evidence that the distance and high-level feature are aligned with the phenomena the hypotheses describe, so they can be employed for evaluating the black-box model.