UNIVERSIDADE DE LISBOA
INSTITUTO SUPERIOR TÉCNICO

Selective Sparsity for Explainability with Applications to Translation Quality Estimation

Marcos Vinícius Treviso

Supervisor: Doctor André Filipe Torres Martins

Thesis approved in public session to obtain the PhD Degree in
Electrical and Computer Engineering

Jury final classification: Pass with Distinction and Honour

2023

TÉCNICO LISBOA

UNIVERSIDADE DE LISBOA
INSTITUTO SUPERIOR TÉCNICO

Selective Sparsity for Explainability with Applications to Translation Quality Estimation

Marcos Vinícius Treviso

Supervisor: Doctor André Filipe Torres Martins

Thesis approved in public session to obtain the PhD Degree in Electrical and Computer Engineering

Jury final classification: Pass with Distinction and Honour

Jury

Chairperson: Doctor Mário Alexandre Teles de Figueiredo, Instituto Superior Técnico, Universidade de Lisboa

Members of the Committee:

Doctor João Miguel da Costa Magalhães, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa;
Doctor André Filipe Torres Martins, Insituto Superior Técnico, Universidade de Lisboa;
Doctor Bruno Emanuel da Graça Martins, Insituto Superior Técnico, Universidade de Lisboa;
Doctor Ana Marasovic, Kahlert School of Computing, University of Utah, EUA;
Doctor Marina Fomicheva, CONTEX-AI, UK

Funding Institutions - European Research Council and Instituto de Telecomunicações

2023

Abstract

This thesis addresses the challenge of improving the interpretability of neural networks in Natural Language Processing (NLP), particularly in the context of quality estimation (QE) for machine translation. Current NLP models are hindered by their dependence on over-parameterized black boxes, raising concerns about their reliability, confidence, and fairness. While several explainability approaches have been proposed for shedding light into neural networks' decisions, ranging from built-in (e.g., attention mechanisms) to post-hoc methods (e.g., gradient-based measures), their evaluation often sidesteps the crucial aspect of effectively **communicating** the underlying model behavior to humans. In this thesis, we propose the development of frameworks to automatically evaluate explainability methods in terms of forward and counterfactual **simulability**—the ability to use explanations for predicting model outputs on unseen examples. We also design an interpretable and more efficient complement to the multi-head attention mechanism found in transformers, the backbone of state-of-the-art QE models. Moreover, we provide empirical evaluations of the plausibility of various explainability methods for QE, and design new explainability methods for interpreting transformed-based QE models, employing **sparsity** as key interpretability driver. Our findings reveal that simulability is a valuable tool for evaluating explainability methods under a single perspective, as well as for designing more plausible and robust explainers, while sparsity is a useful feature for improving the interpretability of transformer-based models. In particular, our empirical evaluations reveal that attention-based methods often outperform other approaches for explaining QE models, and that sparsity can be effectively leveraged to identify relevant internal components, such as attention heads, and to determine influential input words. Furthermore, we demonstrate that sparse signals not only serve to guide the design of efficient attention mechanisms, but also offer valuable information for counterfactual generation. Our successful strategies led to winning submissions in two consecutive editions of the Explainable Quality Estimation Shared Task, in 2021 and 2022, further highlighting the relevance and effectiveness of our approaches. By improving the interpretability of neural networks along these dimensions, this thesis contributes to the development of more transparent, efficient, and understandable systems. Finally, to foster future research in this area, we have made our code open-source and publicly available.

Keywords: Machine learning, natural language processing, explainability, sparsity, translation quality estimation.

Resumo

Esta tese aborda o desafio de melhorar a interpretabilidade de redes neurais em Processamento de Linguagem Natural (NLP, do inglês *Natural Language Processing*), particularmente no contexto de estimativa de qualidade (QE, do inglês *Quality Estimation*) para tradução automática. Os modelos atuais de PNL são limitados por sua dependência de caixas pretas superparametrizadas, levantando preocupações sobre sua confiabilidade, confiança e imparcialidade. Embora várias abordagens de explicabilidade tenham sido propostas para esclarecer as decisões das redes neurais, variando de métodos embutidos (e.g., mecanismos de atenção) a métodos *post-hoc* (e.g., medidas baseadas em gradiente), sua avaliação muitas vezes evita o aspecto crucial de efetivamente **comunicar** o comportamento subjacente do modelo para humanos. Nesta tese, propomos o desenvolvimento de *frameworks* para avaliar automaticamente os métodos de explicabilidade em termos de **simulabilidade** direta e contrafactual —a capacidade de usar explicações para prever saídas de modelos em novos exemplos. Também projetamos um complemento interpretável e mais eficiente para o mecanismo de atenção multicabeças encontrado em transformadores, a espinha dorsal dos modelos QE de última geração. Além disso, fornecemos avaliações empíricas da plausibilidade de vários métodos de explicabilidade para QE e projetamos novos métodos de explicabilidade para interpretar modelos de QE baseados em transformação, empregando **esparsidade** como a principal guia de interpretabilidade. Nossas descobertas revelam que a simulabilidade é uma ferramenta valiosa para avaliar métodos de explicabilidade sob uma única perspectiva, bem como para projetar explicadores mais plausíveis e robustos, enquanto a esparsidade é um recurso útil para melhorar a interpretabilidade de modelos baseados em transformadores. Em particular, nossas avaliações empíricas revelam que os métodos baseados em atenção geralmente superam outras abordagens para explicar os modelos de QE e que a esparsidade pode ser efetivamente aproveitada para identificar componentes internos relevantes, como cabeças de atenção, e para determinar palavras de entrada influentes. Além disso, demonstramos que sinais esparsos não servem apenas para orientar o *design* de mecanismos de atenção eficientes, mas também oferecem informações valiosas para a geração de textos contrafactuais. Nossas estratégias bem-sucedidas levaram a submissões vencedoras em duas edições consecutivas da *Explainable QE Shared Task*, em 2021 e 2022, destacando ainda mais a relevância e a eficácia de nossas abordagens. Ao melhorar a interpretabilidade de redes neurais ao longo dessas dimensões, esta tese contribui para o desenvolvimento de sistemas mais transparentes, eficientes e compreensíveis. Por fim, para fomentar pesquisas futuras nessa área, tornamos nosso código aberto e disponível ao público.

Palavras-chave: Aprendizado de máquina, processamento de linguagem natural, explicabilidade, esparsidade, estimativa de qualidade de tradução.

Acknowledgments

I feel incredibly fortunate to have had the support of so many amazing individuals along the way.

First, I must express my gratitude towards my advisor, André, who is not only exceptionally brilliant but also able to contemplate deeply on various topics while maintaining unparalleled mathematical accuracy. André continuously encouraged me to delve deeper into ideas, explore new possibilities, and expand my horizons as a researcher. André's unwavering support and belief in my abilities propelled me to take risks, learn from mistakes, and celebrate my achievements. Without his guidance and support throughout this process, achieving this milestone would have been impossible. Thank you!

My lab mates, Gonçalo, Ben, Tsvety, Erick, Vlad, Patrick, Nuno, and *many* others, have been a constant source of knowledge and support. I also extend profound gratitude to my esteemed co-authors, notably Alexis Ross, Ricardo Rei, and António Gois. Patrick and Nuno, in particular, deserve special mention for their timely assistance in numerous projects throughout my PhD journey. Old friends from Alegrete, Fabio and Michelle, have not only provided the light to come to Lisbon, but also enriched my life in this city in so many ways. I cherish your friendship and am grateful for the perspectives you have shared. To my new friends in Lisbon, Marcelo, Adriana, Mariana, and Vinicius, thank you for welcoming me with open arms. I would also like to acknowledge Henrico and Thales, dear friends from my MSc days, who contributed to shape my personal traits to face the challenges of a PhD quest and offered invaluable support, even from afar.

This work is dedicated to my mother, Loeiri, and my father, Gilmar, who have nurtured me from the very beginning of my exploration into the realm of "informatics". Knowing that I come from such loving roots reminds me daily how fortunate I am, and no matter the obstacles ahead, I know they are there for me every step of the way. With affection and admiration in my native tongue: Eu amo vocês e sou grato por tudo que fizeram por mim! My sister, Thaís, has been an indispensable ally, and I am forever grateful for her care, patience, and those much-needed croissants that always arrived right on time.

I reserve my warmest thanks for my wife, Flávia, who has been my rock since the beginning of this adventure. From our initial encounter in São Carlos, until now, our union continues to grow stronger by day, serving as the backbone that supports my dreams. You are an indispensable partner, travel companion, co-chef, and above all, someone who fills my heart beyond measure. Together, we have navigated the twists and turns of life in Brazil and abroad, holding steadfast through the stress of COVID and the joy of discovering new countries. My dear Flávia, I am eternally grateful for your love and companionship, and I look forward to exploring the universe with you, hand in hand.

For small creatures such as we the vastness is bearable only through love.

Carl Sagan, Contact.

Contents

Notation

a, $\boldsymbol{a}$, $\boldsymbol{A}$, and $\mathcal{A}$	a scalar, a vector, a matrix, and a set, respectively;
v_i	the ith element of vector $\boldsymbol{v}$;
w_{ij}	the element on the ith row and jth column of $\boldsymbol{W}$;
$\triangle^K$	the canonical simplex, *i.e.*, $\{\boldsymbol{\xi} \in \mathbb{R}^K \ : \ \langle \mathbf{1}, \boldsymbol{\xi} \rangle = 1, \boldsymbol{\xi} \geq \mathbf{0}\}$;
$\mathsf{H}^{\mathsf{S}}(\boldsymbol{p})$	the Shannon's entropy of a distribution $p(z)$, *i.e.*, $-\sum_i p_i \log p_i$;
$\mathsf{KL}\left[p \| q\right]$	the Kullback-Leibler divergence of $p(z)$ from $q(z)$;
$\|\boldsymbol{z}\|_0 \coloneqq \|\{t : z_t \neq 0\}\|$	the number of non-zeros of a vector $\boldsymbol{z}$;
$\mathbb{E}_{p(z)}[f(z)]$	the expectation of a function $f : \mathcal{Z} \to \mathbb{R}$ under distribution $p(z)$, letting $z \in \mathcal{Z}$ be a random variable;

1

Introduction

Contents

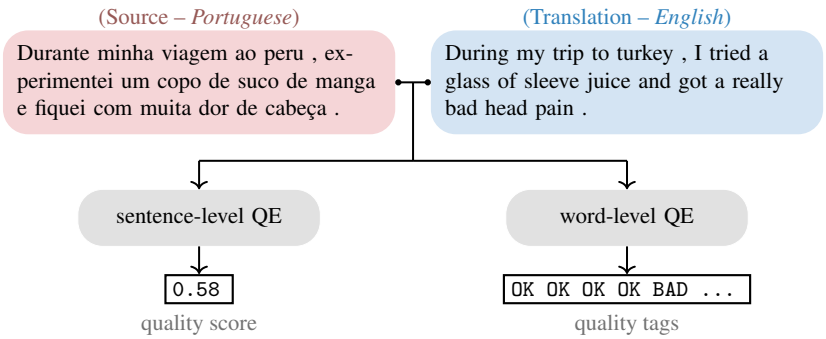


Figure 1.1: Overview of sentence and word-level QE tasks. The goal of sentence-level QE is to estimate the quality of the whole translated sentence, while word-level QE aims at tagging individual words with quality labels (e.g., OK or BAD).

The widespread accessibility of the Web has significantly amplified the volume of multilingual user-generated content, creating an increasing demand for high-quality translations to understand the world. For example, Google Translate, one of the most popular translation apps, is currently installed in more than 1 billion smartphones, supporting the translation of texts in over 100 languages.[1] As a result of this high demand, there has been an increased emphasis on assessing the quality of translations. The task of quality estimation (QE), illustrated in Figure 1.1, addresses this need by providing an estimate of how reliable is an automatically generated translation (Specia et al., 2015), possibly pinpointing words mistranslated in the process.

Following the trend of employing neural-based methods in machine translation (MT), QE systems based on neural networks have become more prominent, allowing automatic translation tools to be more helpful in several academic and industrial applications (Johnson et al., 2017; Graça, 2018; Specia et al., 2018). Underpinning the success of neural-based systems is the *attention* mechanism, which allows the model to focus on different parts of the input sentence when making a decision (Bahdanau et al., 2015). Current QE models are based on the transformer architecture (Vaswani et al., 2017; Junczys-Dowmunt et al., 2018; Ott et al., 2018; Kepler et al., 2019a; Ranasinghe et al., 2020), composed of a stack of attention layers responsible for contextualizing information within and across inputs dynamically. Due to their outstanding performance and parallelizable training regime, large transformers have become the backbone of top submission systems in both WMT-MT and WMT-QE shared tasks (Barrault et al., 2019; Specia et al., 2020).

Despite their effectiveness, neural-based models are considered black-boxes, meaning that they are not amenable to human interpretation. This characteristic has raised concerns about their reliability, confidence, and fairness (Doshi-Velez and Kim, 2017; Lipton, 2018; Rudin, 2019). To address this issue, several explainability approaches have been proposed for shedding light into neural networks decisions, ranging from built-in (e.g., attention mechanisms) to post-hoc methods (e.g., gradient-based measures). These approaches are usually assessed with human-likeness comparisons or faithfulness proxies when explaining shallow models trained on monolingual text classification datasets (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019; DeYoung et al., 2020). However, relying solely on these evaluation methods may not be sufficient, as they might not fully **communicate** the model's behavior to humans (Miller, 2019).

In this thesis, we start by investigating and comparing explainability approaches along new interpretability dimensions, such as forward and counterfactual simulability. Then, we shift our focus to the multilingual task

[1] https://blog.google/products/translate/one-billion-installs/

of QE and propose novel explainability approaches tailored for transformers. Our winning submissions for the Explainable QE shared tasks 2021 and 2022 employ these approaches (Fomicheva et al., 2021; Zerva et al., 2022), building on the findings from the earlier parts of our investigation.

At the start of this project, in 2019, many explainability methods have been proposed to justify neural networks' decisions in Natural Language Processing (NLP). Among them, a popular approach was to inspect the weights in the attention mechanism to interpret the model's decision (Cho et al., 2014b; Luong et al., 2015; Junczys-Dowmunt et al., 2018; Peters et al., 2018a). However, a series of impactful works have questioned the interpretability claims of attention mechanisms, including Jain and Wallace (2019) and Serrano and Smith (2019), who showed that attention weights do not correlate with traditional gradient-based and leave-one-out methods, and that it is easy to find adversarial attention explanations for the same model's decision, raising concerns that attention weights may not always be a fail-safe explainability measure. Wiegreffe and Pinter (2019) probe the previous works and advocate the notion of faithful explanations to determine explanations that capture the true reasoning process that leads to the model's decision, concluding that attention weights may provide a plausible but not always faithful explanation. However, defining and assessing faithfulness remain open questions, as definitions may require causal assumptions and evaluation outcomes are often task-dependent (Jacovi and Goldberg, 2020; Grimsley et al., 2020; Bastings et al., 2022).

The debate regarding attention explainability in NLP reveals the difficulty in evaluating explainability methods. On the one hand, it is unreasonable to expect that explainability methods should behave similarly to each other, as none of them provide ground-truth measures (Neely et al., 2021). On the other hand, while some works evaluate explanations with plausibility ratings (Lei et al., 2016; Camburu et al., 2018), or with faithfulness proxies such as sufficiency and comprehensiveness (DeYoung et al., 2020; Carton et al., 2020), these measures offer a limited view of explainability, sidestepping an important goal of explainability: the ability to **communicate** the underlying model behavior to humans (Treviso and Martins, 2020; Hase and Bansal, 2020; Pruthi et al., 2022). In this thesis, we take a first step to address this issue. Concretely, we first develop communication-based frameworks for assessing explanations on the dimensions of forward and counterfactual simulability (Doshi-Velez and Kim, 2017).[2] Later, we leverage these frameworks to also design new interpretability methods based on selective, sparse attention approaches. As we will see in future chapters, our findings show that sparse attention is not only more informative than gradient and erasure-based methods along both plausibility and simulability dimensions (§3), but can also be exploited for guiding the generation of counterfactuals (§4) and improving inference efficiency in transformers (§5).

Furthermore, at the time this project started, most works analyzed attention explanations on monolingual text classification using shallow neural networks, such as recurrent neural networks (RNNs) equipped with a single attention mechanism. In contrast, more structured tasks such as QE were unexplored, and deeper models based on pretrained transformers composed of multiple attention heads, such as BERT (Devlin et al., 2019), XLM (Conneau and Lample, 2019), and T5 (Raffel et al., 2020) were becoming popular and feasible to use in practice. In this thesis, we study and design explainability methods for interpreting QE models based on pretrained transformers, and as we will see, besides outperforming other methods in terms of plausibility (§6 and §7), attention-based approaches can be easily combined with sparse transformations to optimize simulability

[2]Simulability refers to the degree of informativeness in an explanation when it is presented to a human, who is then asked to predict the outcome of the classifier—"simulating" the underlying model behavior in the process.

and automatically identify relevant attention heads (§8).

Many works laid out the foundation of this thesis. In particular, Doshi-Velez and Kim (2017) provided the essential motivation to design simulability frameworks to automatically assess explainability methods. We also build upon a line of research that aims to interpret neural networks using sparsity, including the use of sparse normalized transformations (Martins and Astudillo, 2016; Niculae and Blondel, 2017; Peters et al., 2019) and their application to transformers (Correia et al., 2019), which we explore to design novel explainability methods and improve attention efficiency. Finally, our idea to interpret QE models is largely due to previous work that established human-annotated data as a reliable source of ground-truth for explainability (Fomicheva et al., 2020, 2022a), which we use to systematically compare different approaches and propose sparsity-based solutions.

1.1 Contributions and Thesis Statement

We now summarize the main contributions of this thesis.

- We create a framework to automatically evaluate explainability methods in terms of forward simulability, as defined by Doshi-Velez and Kim (2017) with human participants. We further exploit this framework to design a new approach that maximize simulability by leveraging sparse attention (§3).
- We incorporate learnable sparse signals to guide the generation of synthetic counterfactuals using masked language models, enabling the design of more robust explainers. Equipped with a counterfactual generator, we propose an automatic approach to evaluate explainability methods based on counterfactual simulability (§4).
- We design an efficient and interpretable complement to the multi-head attention mechanism found in transformers. Specifically, we train a small model (student) to predict, *a priori*, the sparse attention pattern of a large model (teacher) for a given input, effectively reducing computation time (§5).
- We provide an empirical evaluation of the plausibility of several explainability methods for QE, including gradient, erasure, and attention-based approaches (§6). Further on, we leverage previous findings and insights to design more plausible and practical explainers based on sparsity (§7).
- We investigate the extent to which an explainability method can learn to produce better explanations for QE. To achieve this, we design a differentiable attention explainer that maximizes forward simulability, allowing us to use sparsity to identify important attention heads in transformers (§8).

Thesis Statement. The main claim of this thesis is that simulability and sparsity do have the ability to improve the interpretability of neural networks: we find that sparsity can be a key ingredient for improving the explainability of neural models, and that simulability is not only helpful for evaluating explainability methods, but can also be exploited to design more plausible and robust explainers. Through rigorous empirical evaluations, we also find that attention solutions often outperform other approaches for explaining QE models, and that sparsity can act as a guide to identify relevant inner components of the model, such as attention heads.

1.2 Publications

During the course of this Ph.D., I have co-authored the following works, some of which will be covered in this thesis (marked with references to their respective chapters):

- **OpenKiwi: An open source framework for quality estimation** (Kepler et al., 2019b). Accepted at ACL 2019. *This paper won the best demo paper award.* Not included in this thesis.
- **Unbabel's Participation in the WMT19 Translation Quality Estimation Shared Task** (Kepler et al., 2019a). Accepted at WMT 2019, co-allocated with ACL 2019. *This paper won the Shared Task for word, sentence, and document-level*. Not included in this thesis.
- **The Explanation Game: Towards Prediction Explainability through Sparse Communication** (Treviso and Martins, 2020). Accepted at the BlackBoxNLP workshop, co-allocated with EMNLP 2020. Described in §3.
- **Sparse and Continuous Attention Mechanisms** (Martins et al., 2020). Accepted for *spotlight presentation* at NeurIPS 2020. Not included in this thesis.
- **IST-Unbabel 2021 Submission for the Explainable Quality Estimation Shared Task** (Treviso et al., 2021). Accepted at the Eval4NLP workshop, co-allocated with EMNLP 2021. *This paper won the unconstrained track of the Shared Task and the Best Explainability Approach Award*. Described in §6.
- **Predicting Attention Sparsity in Transformers** (Treviso et al., 2022). Accepted at the SPNLP workshop, co-allocated with ACL 2022. Described in §5.
- **CometKiwi: IST-Unbabel 2022 Submission for the Quality Estimation Shared Task** (Rei et al., 2022b). Accepted at WMT 2022, co-allocated with EMNLP 2022. *This paper won the QE and Explainable QE tracks of the Shared Task.* Only my part of the collaboration will be covered in this thesis. Described in §7.
- **Learning to Scaffold: Optimizing Model Explanations for Teaching** (Fernandes et al., 2022). Accepted at NeurIPS 2022. This paper was co-led with Patrick Fernandes, and only my part of the collaboration will be covered in this thesis. Described in §8.
- **Sparse Continuous Distributions and Fenchel-Young Losses** (Martins et al., 2022). Accepted at JMLR 2022. Not included in this thesis.
- **CREST: A Joint Framework for Rationalization and Counterfactual Text Generation** (Treviso et al., 2023b). Accepted at ACL 2023. Described in §4.
- **The Inside Story: Towards Better Understanding of Machine Translation Neural Evaluation Metrics** (Rei et al., 2023). Accepted at ACL 2023. Not included in this thesis.
- **Efficient Methods for Natural Language Processing: A Survey** (Treviso et al., 2023a). Accepted at TACL 2023. Not included in this thesis.

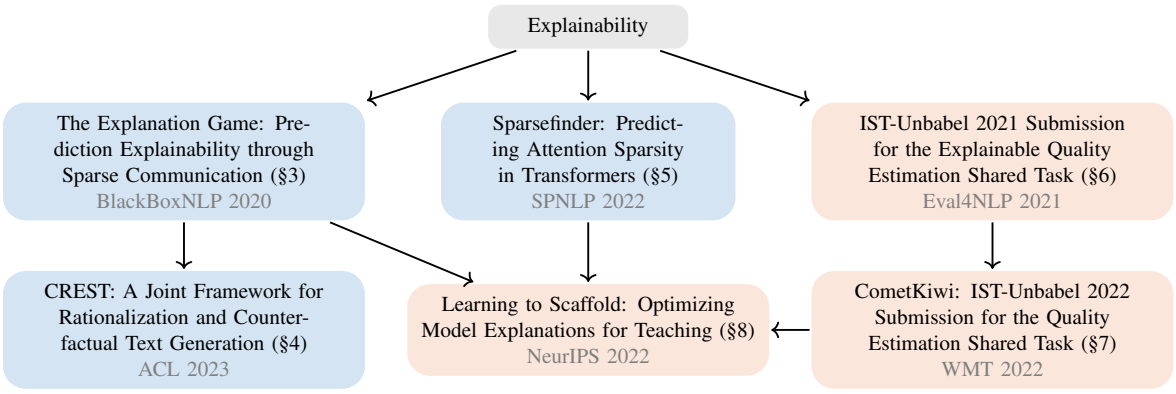


Figure 1.2: Schematic overview of the works covered in this thesis, covering explainability with selective sparse methods (in blue), and explainability applied to QE (in orange).

1.3 Thesis Roadmap

Herein, we show the outline of this thesis, which we divide into two parts: in the first part we explore selective sparsity for explainability in NLP, and in the second part we specifically change our focus to explainable QE. Figure 1.2 depicts the connections between the works covered in this thesis.

Chapter 2: Background. We provide a succinct overview of the theoretical foundation for this thesis, describing key concepts used in our neural models, such as the encoder-decoder architecture, sparse attention mechanisms, and transformers. We also provide high-level definitions for sentence and word-level QE, alongside key explainability notions employed throughout this thesis, such as popular explainability methods and evaluation metrics.

Chapter 3: The Explanation Game: Prediction Explainability through Sparse Communication. We propose a simulability framework that provides an unique perspective of explainability as a communication problem, which we use to compare several approaches and design a new post-hoc method that is trained to optimize the communication.

Chapter 4: CREST: A Joint Framework for Rationalization and Counterfactual Text Generation. We propose a joint framework for selective rationalization and counterfactual text generation, and further leverage its ability to synthesize counterfactuals to assess counterfactual simulability.

Chapter 5: Sparsefinder: Predicting Attention Sparsity in Transformers. We study model efficiency of transformer architectures via the tradeoff between the sparsity and recall of the predicted attention graph, and propose *Sparsefinder*, a model that preemptively identifies sparse attention patterns to reduce computational costs without abdicating interpretability.

Chapter 6: An Empirical Comparison of Explainability Methods for Quality Estimation. We present the joint contribution of IST and Unbabel to the Eval4NLP 2021 Explainable Quality Estimation Shared Task,

where we propose to use attention heads information to explain accurate QE models on two settings: constrained (without word-level supervision) and unconstrained (with word-level supervision).

Chapter 7: Sparse Bottleneck Layer for Explainable Quality Estimation. We present the joint contribution of IST and Unbabel to the Explainable track of the WMT 2022 Shared Task on Quality Estimation, where we further complement attention heads with gradient information, and propose a sparse selection mechanism to automatically identify well-performing attention heads.

Chapter 8: Learning to Scaffold: Optimizing Model Explanations for Quality Estimation. Given a trained transformer-based model finetuned on QE, we propose a parameterized sparse attention-based explainer that is trained to optimize simulability and identify relevant attention heads in a post-hoc manner.

Chapter 9: Conclusions. We summarize the main contributions, outline the existent limitations, and discuss promising future directions linked to this thesis.

2

Background

Contents

2.1 Sequence-to-Sequence Models

Considerable progress has been made in machine translation (MT) in recent times fostered on the breakthrough of deep learning. Modern approaches have switched from statistical-based machine translation (SMT) to neural-based machine translation (NMT), which rely on the rich parameterization enabled by deep neural networks to transform the source sentence into the target sentence in an end-to-end fashion (Sutskever et al., 2014; Cho et al., 2014b). In this section, we introduce the key concepts employed in modern NMT and QE models adopted in this thesis: the encoder-decoder architecture, the attention mechanism, and the transformer architecture (Vaswani et al., 2017).

2.1.1 Encoder-Decoder Architecture

The goal of a sequence-to-sequence (*seq2seq*) model is to generate a target sequence of symbols y from a source sequence x, which is usually modeled as a conditional distribution $p_\theta(y \mid x)$ parameterized by θ. Most sequence-to-sequence models are parameterized with an encoder-decoder architecture, which consists of the following blocks:

- The **encoder**, which receives a source sequence $x = \langle x_1, ..., x_n \rangle$ as input and produces hidden representations $\boldsymbol{H} = \langle \boldsymbol{h}_1, ..., \boldsymbol{h}_n \rangle$ for each input word x_i.
- The **decoder**, which uses the encoder's hidden representations as contextual information to generate a variable-length target sequence $y = \langle y_1, ..., y_m \rangle$.

In the remainder of this section, we describe in more detail the attention mechanism commonly employed in encoder-decoder architectures.

2.1.2 Attention Mechanisms

In early *seq2seq* models, the encoder summarizes the entire input into a fixed vector representation $\boldsymbol{c}$, called contextual vector, which, in turn, is used by the decoder to generate an output sequence. However, a fixed contextual vector creates a bottleneck since a fixed representation "does not have enough capacity to encode a long sentence with complicated structure and meaning" (Cho et al., 2014a). This problem is mitigated by attention mechanisms (Bahdanau et al., 2015), which automatically focus on important hidden states $\boldsymbol{h}_i$ to create a contextual vector $\boldsymbol{c}_j$ at each time-step j. More formally, at each time-step j, the decoder's hidden state $\boldsymbol{q}_j$ and the encoder's hidden states $\boldsymbol{h}_i$ are used to compute attention scores $z_i = f(\boldsymbol{q}_j, \boldsymbol{h}_i)$, where f is a vector-valued function that might be parameterized by learnable weights $\boldsymbol{W}$, such as $f(\boldsymbol{q}_j, \boldsymbol{h}_i) = \boldsymbol{q}_j^\top \boldsymbol{W} \boldsymbol{h}_i$. The vector $\boldsymbol{z} \in \mathbb{R}^n$ is then used to compute the contextual vector $\boldsymbol{c}_j$ via a weighted average of the encoder's hidden states:

$$\boldsymbol{c}_j = \sum_{i=1}^{n} \pi(\boldsymbol{z})_i \boldsymbol{h}_i, \tag{2.1}$$

where $\pi : \mathbb{R}^n \to \triangle^n$ is a function that maps scores to probabilities, and $\triangle^n := \{\boldsymbol{p} \in \mathbb{R}^n \mid \boldsymbol{p} \geq \boldsymbol{0},\ \boldsymbol{1}^\top \boldsymbol{p} = 1\}$ is the $(n-1)$-probability simplex. The most common choice for $\pi(\cdot)$ is the softmax transformation:

$$\mathsf{softmax}(\boldsymbol{z})_j = \frac{\exp(z_j)}{\sum_{j'} \exp(z_{j'})}. \tag{2.2}$$

Softmax is a dense transformation, i.e., it places some probability mass to every jth input, even if small. However, we can also consider $\pi(\cdot)$ transformations with other desirable properties, such as sparsity.

Sparse attention. A natural way to get a sparse attention distribution is by using the **sparsemax transformation** (Martins and Astudillo, 2016), which computes an Euclidean projection of the score vector onto the probability simplex $\triangle^n$, or, more generally, the **α-entmax transformation** (Peters et al., 2019):

$$\alpha\text{-entmax}(\boldsymbol{z}) := \underset{\boldsymbol{p}\in\triangle^n}{\operatorname{argmax}}\, \boldsymbol{p}^\top \boldsymbol{z} + \mathsf{H}^{\mathsf{T}}_\alpha(\boldsymbol{p}), \tag{2.3}$$

where $\mathsf{H}^{\mathsf{T}}_\alpha$ is a generalization of the Shannon and Gini entropies proposed by Tsallis (1988), parametrized by a scalar $\alpha \geq 1$:

$$\mathsf{H}^{\mathsf{T}}_\alpha(\boldsymbol{p}) := \begin{cases} \frac{1}{\alpha(\alpha-1)} \sum_j (p_j - p_j^\alpha), & \alpha \neq 1 \\ \mathsf{H}^{\mathsf{S}}(\boldsymbol{p}), & \alpha = 1, \end{cases} \tag{2.4}$$

where $\mathsf{H}^{\mathsf{S}}(\boldsymbol{p}) := -\sum_j p_j \log p_j$ is the Shannon entropy. To use the α-entmax transformation, we need to solve the maximization in Equation 2.3, i.e., obtain $\boldsymbol{p}^\star = \alpha\text{-entmax}(\boldsymbol{z})$ for a given vector of scores $\boldsymbol{z}$. The solution is given in the following form (Peters et al., 2019):

$$\alpha\text{-entmax}(\boldsymbol{z}) = [(\alpha-1)\boldsymbol{z} - \tau(\boldsymbol{z})\mathbf{1}]_+^{1/\alpha-1}, \tag{2.5}$$

where $[\cdot]_+$ is the positive part (ReLU) function, and $\tau : \mathbb{R}^n \to \mathbb{R}$ is a normalizing function that satisfies $\sum_j [(\alpha-1)z_j - \tau(\boldsymbol{z})]_+^{1/\alpha-1} = 1$ for any $\boldsymbol{z}$. That is, entries with score $z_j \leq {}^{\tau(\boldsymbol{z})}/_{\alpha-1}$ get exact zero probability. In the limit $\alpha \to 1$, α-entmax recovers the softmax function, while for any value of $\alpha > 1$ this transformation can return sparse probability vectors (as the value of α increases, the induced probability distribution becomes more sparse). In particular, letting $\alpha = 2$ we recover sparsemax.

In this thesis, we often use attention weights $\pi(\boldsymbol{z})_i$ as an explainability score for the ith input token. For example, we evaluate dense (softmax) and sparse (sparsemax and 1.5-entmax) attention methods with a simulability framework in Chapter 3.

2.1.3 Transformers

Traditionally, both the encoder and the decoder in *seq2seq* models were built using gated recurrent networks, such as long short-term memory units (LSTM, Hochreiter and Schmidhuber 1997) or gated recurrent units (GRU, Cho et al. 2014b). Recent NMT systems use transformers, an architecture that avoids recurrences by stacking self-attention layers to contextualize information within and across input sentences dynamically (Vaswani et al., 2017). In contrast to recurrent networks, self-attention layers only require matrix multiplications, allowing parallelizable computations across time steps on modern hardware during training.

The main component of transformers is the **multi-head attention** mechanism. Concretely, given as input a matrix $\boldsymbol{Q} \in \mathbb{R}^{m\times d}$ containing d-dimensional representations for m queries, and matrices $\boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{n\times d}$ for n keys and values, the *scaled dot-product attention* at a single head is computed as:

$$\mathsf{att}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \pi\underbrace{\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d}}\right)}_{\boldsymbol{Z}\in\mathbb{R}^{m\times n}}\boldsymbol{V} \in \mathbb{R}^{m\times d}. \tag{2.6}$$

The π transformation maps rows to distributions, with softmax being the most common choice, $\pi(\boldsymbol{Z})_{ij} = \mathsf{softmax}(\boldsymbol{z}_i)_j$. Multi-head attention is computed by evoking Eq. 2.6 in parallel for each head h:

$$\boldsymbol{h}_h = \mathsf{att}(\boldsymbol{Q}\boldsymbol{W}_h^Q, \boldsymbol{K}\boldsymbol{W}_h^K, \boldsymbol{V}\boldsymbol{W}_h^V), \tag{2.7}$$

where $\boldsymbol{W}_h^Q$, $\boldsymbol{W}_h^K$, $\boldsymbol{W}_h^V$ are learnable linear transformations. The output of the multi-head attention module at the ℓth layer is a set of hidden states $\boldsymbol{H}_\ell \in \mathbb{R}^{m\times d}$ formed via the concatenation of all $\boldsymbol{h}_{\ell,1}, ..., \boldsymbol{h}_{\ell,H}$ heads in that layer followed by a learnable linear transformation $\boldsymbol{W}^O$:

$$\boldsymbol{H}_\ell = \mathsf{concat}(\boldsymbol{h}_{\ell,1}, ..., \boldsymbol{h}_{\ell,H})\boldsymbol{W}^O. \tag{2.8}$$

These hidden states are further refined through position-wise feed-forward blocks and residual connections to obtain a final representation: $\tilde{\boldsymbol{H}}_\ell = \mathsf{FFN}(\boldsymbol{H}_\ell) + \boldsymbol{H}_\ell$. Thanks to the application of different learnable linear transformations, distinct attention heads have the capability of learning specialized phenomena.

Importantly, transformers have three types of multi-head attention mechanism: encoder self-attention (source-to-source), decoder self-attention (target-to-target), decoder cross-attention (target-to-source). While there are no restrictions to over which elements are attended in the encoder, elements in position $j > i$ in the decoder self-attention are masked at time-step i ("causal mask"). Transformers with only encoder-blocks, such as BERT (Devlin et al., 2019) and XLM (Conneau et al., 2020), have only the encoder self-attention, and thus $m = n$.

In several chapters of this thesis, we employ pretrained transformers as the backbone of our models and use attention weights $\boldsymbol{A} = \pi(\boldsymbol{Z}) \in \mathbb{R}^{m\times n}$ as interpretability indicators, where each ith row $\boldsymbol{A}_i \in \triangle^n$. In Chapter 5 we study attention heads trained with α-entmax. In Chapters 6 and 7 we investigate the explainability power of independent attention heads on the task of QE. In addition, in Chapter 8 we design a new explainability method based on a sparse parameterization of attention heads using the sparsemax transformation.

2.2 Quality Estimation

Quality estimation (QE) is the task of evaluating a translation system's quality without access to reference translations (Blatz et al., 2004; Specia et al., 2018). Among its potential usages are: informing an end user about the reliability of automatically translated content; deciding if a translation is ready for publishing or if it requires human post-editing; and highlighting the words that need to be post-edited. QE systems are usually framed according to the granularity in which predictions are made, such as sentence or word-level,[1] We illustrate the tasks of sentence and word-level QE in Figure 1.1. In this section, we describe the main concepts to formulate and evaluate sentence and word-level QE tasks, which we will explore later in the second part of this thesis (in Chapters 6, 7, and 8).

Sentence-level. The goal of sentence-level QE is to predict the quality of the whole translated sentence, either in terms of how many edit operations are required to fix it (*Human Translation Error Rate*, HTER, Snover et al. 2006), in terms of direct assessments (DA, Graham et al. 2013) obtained via human judgments, or more recently in terms of a finegrained annotation schema known as Multidimensional Quality Metrics (MQM, Lommel et al. 2014; Zerva et al. 2022), in which translation errors are annotated with severity (minor, major, critical) and type

[1] QE can be framed in a phrase or document-level format when assessing the translation quality of phrases or documents, respectively.

(omission, style, mistranslation, etc) markers. Sentence-level QE can be cast as a regression problem, where a source $\boldsymbol{s} = \langle s_1, ..., s_n \rangle$ and a translation sequence $\boldsymbol{t} = \langle t_1, ..., t_m \rangle$ are given as input to a model, which predicts a single score $\hat{y} \in \mathbb{R}$ that represents an estimate of the translation quality. Since sentence-level systems usually predict a continuous score, regression-based metrics such as Mean Absolute Error (MAE), Pearson's correlation, and Spearman's rank correlation are usually used for evaluation (Fonseca et al., 2019; Specia et al., 2020; Zerva et al., 2022).

Word-level. Word-level QE aims at assigning quality labels (e.g., OK or BAD) to individual words. More precisely, models should assign a label to each *machine-translated word*, indicating whether that word is a translation error or not. Additionally, current systems can also classify *source words*, to denote words in the source sequence that have been mistranslated or omitted in the target, and *machine-translated gaps*, to account for context words that need to be inserted. Word-level QE can be cast as a sequence labelling problem, where the translation sequence $\boldsymbol{t} = \langle t_1, ..., t_m \rangle$ is augmented with NULL tokens $\emptyset$ at each position $1 \leq i \leq m + 1$ to account for gap labels $\boldsymbol{t}_+ = \langle \emptyset, t_1, \emptyset, t_2, ..., \emptyset, t_m, \emptyset \rangle$. That is, models should predict a label $\hat{y}_i^{(t)} \in \mathcal{Y}$ for each ith token in $\boldsymbol{t}_+$, and a label $\hat{y}_i^{(s)} \in \mathcal{Y}$ for each ith token in $\boldsymbol{s} = \langle s_1, ..., s_n \rangle$, where $\mathcal{Y}$ represents the label set. To avoid favoring pessimistic (always predict BAD tags) and optimistic (always predict OK tags) predictions, word-level QE models are evaluated using F_1-MULT, which is calculated via the product of the F_1 score for the BAD class with the F_1 score for the OK class. Recent editions of the WMT QE shared task started to use Matthews correlation coefficient (MCC) as the primary metric to evaluate word-level QE systems since it is unaffected by class unbalance (Fonseca et al., 2019; Specia et al., 2020; Zerva et al., 2022).

2.3 Explainability for NLP

The widespread use of machine learning systems to assist humans in decision making brings the need for providing interpretations for models' predictions (Lipton, 2018; Doshi-Velez and Kim, 2017; Rudin, 2019; Miller, 2019). This poses a challenge in NLP, where current systems are based on deep neural networks that generally lack transparency (Goldberg and Hirst, 2017; Peters et al., 2018b; Devlin et al., 2019). The goal of explainability is to provide additional information that helps to unravel why a prediction was made in a certain way, depending on the application at hand and on the social attributions related to it (Miller, 2019; Jacovi and Goldberg, 2021). In this section, we cover the main explainability methods and evaluation setups that are commonly explored in the NLP literature and that are relevant for this thesis.

2.3.1 Methods

From the model's perspective, we can have different types of explanations depending on the task at hand and on the explanation method used. For instance, when recognizing objects from an image, it is common to select pixels or regions of the image as part of the explanation (Ribeiro et al., 2016; Montavon et al., 2018). In NLP, the notion of *rationales* that support the model's decision is commonly applied. For instance, Lei et al. (2016); Bastings et al. (2019) define a rationale, or *highlights*, as "a short yet sufficient part of the input text". Next, we describe existing approaches to generate highlights explanations.

Gradient-based methods. Given a differentiable model f that predicts $y = f(\boldsymbol{x}_{1:n}) \in \mathcal{Y}$ for a sequence of n input vectors, where $\boldsymbol{x}_i \in \mathbb{R}^d$ is the ith word vector and $\mathcal{Y}$ represents a label set, explanations extracted for target class c from gradients stored during backpropagation can be computed as **sensitivity scores**:[2]

$$\|\nabla_{\boldsymbol{x}_i} f_c(\boldsymbol{x}_{1:n})\|_2. \tag{2.9}$$

Alternatively, saliency scores can be obtained by taking the dot-product of the gradient with the input vector, an approach known as **Input** $\times$ **Gradient**:

$$\boldsymbol{x}_i \cdot \nabla_{\boldsymbol{x}_i} f_c(\boldsymbol{x}_{1:n}). \tag{2.10}$$

Note that for a linear model this gradient equals the total contribution of the ith feature to the target class c. To overcome the problem of having gradients close to zero, Sundararajan et al. (2016) propose **Integrated Gradients**, an approach that average gradients for m inputs linearly spaced between a baseline vector $\bar{\boldsymbol{x}}_{1:n}$ and the original input $\boldsymbol{x}_{1:n}$:

$$\frac{\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i}{m} \cdot \sum_{j=1}^{m} \frac{\partial f_c(\bar{\boldsymbol{x}}_{1:n} + \frac{j}{m}(\boldsymbol{x}_{1:n} - \bar{\boldsymbol{x}}_{1:n}))}{\partial \boldsymbol{x}_i}. \tag{2.11}$$

In NLP, the baseline vector is usually defined as a sequence of zero, `<unk>`, or `<mask>` vectors.

Pertubation-based methods. To analyze the impact of the model output with respect to perturbations to the input, perturbation-based methods compute the following expression for each ith word:

$$f_c(\boldsymbol{x}_{1:n}) - f_c(e_i(\boldsymbol{x}_{1:n})), \tag{2.12}$$

where e_i is a function that perturbs that ith input vector, such as erasing it or replacing it by a placeholder vector.

Attention-based methods. Attention mechanisms automatically learn a probability distribution $\boldsymbol{p} = \pi(\boldsymbol{z}) \in \triangle^n$ over the input vector, as defined in §2.1.2. This distribution is frequently used to explain the model's decision since it is a component that directly aids the model during the forward propagation. The flexibility of attention mechanisms allows the design of many attention-based explanations. For example, as discussed in §2.1.2, **sparse attention** explanations can be easily obtained by sparsity-inducing transformations, such as sparsemax or α-entmax transformations. Attention explanations can also be improved by considering other components of the network, such as the norm of value vectors—an approach known as **Attention** $\times$ **Norm** (Kobayashi et al., 2020)—or as we will see in Chapter 7, using gradient information.

Rationalizers. Models in which a hard attention mechanism is employed to dynamically highlight relevant tokens are known as rationalizers. More precisely, the traditional framework of rationalization involves training two components cooperatively: the *generator* and the *predictor*. The generator encodes the input and produces a "rationale" (e.g., highlights), while the predictor classifies the text given only the rationale as input (Lei et al., 2016). The full process can be summarized as follows:

$$\boldsymbol{z} = \mathsf{gen}(\boldsymbol{x}; \phi), \tag{2.13}$$

$$y = \mathsf{pred}(\boldsymbol{x} \odot \boldsymbol{z}; \theta). \tag{2.14}$$

[2]Gradient methods can also be applied in regression problems ($\mathcal{Y} \subseteq \mathbb{R}$).

where $\odot$ represents element-wise multiplication. To ensure that the explainer does not select all tokens (i.e., $z_i = 1, \forall i$), sparsity and contiguity penalties are often applied to $\boldsymbol{z}$ to encourage the selection of compact and contiguous rationales:

$$\Omega(\boldsymbol{z}) = \lambda_1 \underbrace{\sum_i |z_i|}_{\text{sparsity}} + \lambda_2 \underbrace{\sum_i |z_i - z_{i+1}|}_{\text{contiguity}}. \tag{2.15}$$

Contiguity is usually desired as there is some evidence that it improves readability (Jain et al., 2020). A rationalizer is considered "stochastic" when the rationale generator is modeled with stochastic random variables, such as Bernoulli (Lei et al., 2016) or HardKuma (Bastings et al., 2019), resulting in a sampling operation $\boldsymbol{z} \sim \mathsf{gen}(\boldsymbol{x})$. On the other hand, a rationalizer is "deterministic" when it avoids sampling and instead computes a deterministic mapping $\boldsymbol{z} = \mathsf{gen}(\boldsymbol{x})$, often relaxing latent selections $z_i \in [0, 1]$, such as with sparsemax (Treviso and Martins, 2020) or SparseMAP (Guerreiro and Martins, 2021). To train stochastic rationalizers, the expected cost can be minimized using REINFORCE or the reparameterization trick. In contrast, training deterministic rationalizers is simpler because the gradients can be calculated exactly. In this thesis, specifically in Chapter 4, we will employ deterministic rationalizers for masking the input text in order to create counterfactuals with a span-infilling model, and later leverage their differentiability properties to produce better rationales.

Counterfactuals. In NLP, counterfactuals refer to alternative texts that describe a different outcome than what is encoded in a given factual text. Prior works (Verma et al., 2020; Ross et al., 2021; Wu et al., 2021; Robeer et al., 2021) have focused on developing methods for generating counterfactuals that adhere to certain key properties, such as:

- **Validity**: the generated counterfactuals should encode a different label from the original text.
- **Closeness**: the changes made to the text should be small, not involving large-scale rewriting of the input.
- **Fluency**: the generated counterfactuals should be coherent and grammatically correct.
- **Diversity**: the method should generate a wide range of counterfactuals with diverse characteristics, rather than only a limited set of variations.

We note that although adversarial examples are also defined as inputs that change the model prediction, they are conceptually different from counterfactuals. More precisely, adversarial examples are inputs that change the model's prediction but are not necessarily realistic, as they may not necessarily adhere to closeness, fluency, or diversity properties (Wallace et al., 2019). In Chapter 4, we propose a counterfactual generator method that follow these properties.

2.3.2 Evaluation

Over the past few years, the process of evaluating explainability in NLP has evolved along multiple paths. Bringing explainability methods under a unified framework is challenging, as they are often designed for specific tasks and deliver explanations in a variety of formats. Nonetheless, a growing trend has emerged in using certain metrics to assess highlight-based explanations, such as:

- **Plausibility:** assesses the human-likeness of the explanations. It can be computed by performing human evaluations or by calculating the overlap with annotated snippets.

- **Readability:** assesses if the explanations are human-readable. It can be computed via human evaluation, via reference-based metrics (e.g., BLEU), or by using a proxy model (e.g., perplexity, BERT score).

- **Faithfulness:** gives the degree in which the explanation resembles the true reasoning process the led to the model's prediction. As pointed by Wiegreffe and Pinter (2019), evaluating faithfulness is very challenging since we might need to know causal dependencies a priori. However, some proxy metrics have been proposed to quantify this notion, such as sufficiency and comprehensiveness (DeYoung et al., 2020; Carton et al., 2020).

- **Simulability:** tells how informative an explanation with the process of presenting an explanation for a particular prediction (and possibly the input) to a human, who then tries to correctly guess the model's prediction (Doshi-Velez and Kim, 2017; Treviso and Martins, 2020; Hase and Bansal, 2020; Pruthi et al., 2022). In Chapter 3, we design an automatic framework for computing forward simulability, assessing several explainability methods with it. And next, in Chapter 4, we exploit an automatic counterfactual generator to evaluate explanations in terms of counterfactual simulation, where the goal is instead to identify parts of the input that must be changed in order to induce a different model's prediction.

Part I

Selective Sparsity for Explainability

3

The Explanation Game: Prediction Explainability through Sparse Communication

Contents

In this chapter, we first categorize explainability techniques according to a typology that links classical feature selection for model interpretability with dynamic selection for decision interpretability. In Chapter 1, we covered the debate around attention explanations and the difficulty in evaluating explainability in NLP. To address this issue, we then take inspiration from the simulability setup proposed by Doshi-Velez and Kim (2017) and cast explainability as a communication problem between an explainer and a layperson about a classifier's decision, allowing empirical evaluations under a unique perspective.

We use this framework to systematically compare various explainers applied on top of simple recurrent neural network (RNN) models—such as erasure, gradient methods, and attention mechanisms—on three tasks: text classification, natural language inference, and machine translation.[1]

With different configurations of explainers and laypeople (including both machines and humans), our experiments reveal an advantage of attention-based explainers over gradient and erasure methods. We also show that selective attention is a simpler alternative to stochastic rationale extractors. Furthermore, human experiments show strong results on text classification with post-hoc explainers trained to optimize communication success.

This chapter is based on Treviso and Martins (2020).

3.1 Motivation

The widespread use of machine learning to assist humans in decision making brings the need for explaining models' predictions (Doshi-Velez and Kim, 2017; Lipton, 2018; Rudin, 2019; Miller, 2019). This poses a challenge in NLP, where current neural systems are generally opaque (Goldberg and Hirst, 2017; Peters et al., 2018b; Devlin et al., 2019). Despite the large body of recent work (reviewed in §3.7), a unified perspective modeling the human-machine interaction—a *communication* process in its essence—is still missing.

Many methods have been proposed to generate explanations. Some neural network architectures are equipped with built-in components—attention mechanisms—which weigh the relevance of input features for triggering a decision (Bahdanau et al., 2015; Vaswani et al., 2017). Top-k attention weights provide plausible, but not always faithful, explanations (Jain and Wallace, 2019; Serrano and Smith, 2019; Wiegreffe and Pinter, 2019). Rationalizers with hard attention are arguably more faithful, but require stochastic networks, which are harder to train (Lei et al., 2016; Bastings et al., 2019). Other approaches seek local explanations by evaluating the gradient of the predicted label with respect to the input features (Li et al., 2016a; Arras et al., 2016), or in a post-hoc manner by training a sparse linear model on a vicinity of the input example (Ribeiro et al., 2016), or by repeatedly querying the classifier with leave-one-out strategies (Li et al., 2016a; Feng et al., 2018).

How should these different approaches be compared? Several diagnostic tests have been proposed: Jain and Wallace (2019) assessed the explanatory power of attention weights by measuring their correlation with input gradients; Wiegreffe and Pinter (2019) and DeYoung et al. (2020) developed more informative tests, including a combination of comprehensiveness and sufficiency metrics and the correlation with human rationales; Jacovi and Goldberg (2020) proposed a set of evaluation recommendations and a graded notion of faithfulness. Most proposed frameworks rely on correlations and proxy metrics, sidestepping the main practical goal of prediction explainability—the ability to *communicate* an explanation to a human user.

[1] While pretrained transformers were still in their infancy at the time of this work, RNNs with attention were considered the standard architecture for studying interpretability in NLP, particularly for text classification and natural language inference tasks.

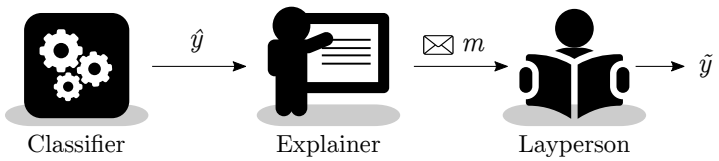


Figure 3.1: Our framework to model explainability as communication. Predictions $\hat{y}$ are made by a classifier C; an explainer E (either embedded in C or operating post-hoc) accesses these predictions and communicates an explanation (a message m) to the layperson L. Success of the communication is dictated by the ability of L and C to match their predictions: $\tilde{y} \stackrel{?}{=} \hat{y}$. Both the explainer and layperson can be humans or machines.

In this work, we fill the gap above by proposing a unified framework that regards explainability as a **communication problem**. Our framework is inspired by human-grounded evaluation through **forward simulation/prediction**, as proposed by Doshi-Velez and Kim (2017, §3.2), where humans are presented with an explanation and an input, and must correctly simulate the model's output (regardless of the true output). We model this process as shown in Figure 3.1, by considering the interaction between a *classifier* (the model whose predictions we want to explain), an *explainer* (which provides the explanations), and a *layperson* (which must recover the classifier's prediction). We show that different configurations of these components correspond to previously proposed explanation methods, and we experiment with explainers and laypeople being both humans and machines.

Our framework recovers as particular cases many previously proposed explainers, according to a typology that draws a connection between traditional feature selection (Guyon and Elisseeff, 2003) and modern explanation techniques, and it also inspires new ones: embedded explainers based on **selective attention** (Martins and Astudillo, 2016; Peters et al., 2019), and **trainable explainers** based on emergent communication (Foerster et al., 2016; Lazaridou et al., 2016).

Overall, our contributions are:

- We draw a link between recent techniques for explainability of neural networks and classic feature selection in linear models (§3.2). This leads to new embedded methods for explainability through selective, sparse attention (§3.3).
- We propose a new framework to assess explanatory power as the communication success rate between an explainer and a layperson (§3.4).
- We experiment with text classification, natural language inference, and machine translation, using different configurations of explainers and laypeople, both machines (§3.5) and humans (§3.6).

3.2 Revisiting Feature Selection

A common way of generating explanations is by highlighting *rationales* (Zaidan and Eisner, 2008). The principle of parsimony ("Occam's razor") advocates simple explanations over complex ones. This principle inspired a large body of work in traditional feature selection for linear models.

We start by drawing a link between explainability of neural networks and feature selection in linear models, making a bridge between the two worlds. This connection is tied to the distinction between model interpretability and prediction explanations made by Lipton (2018). Table 3.1 highlights the connections.

	Static selection (model interpretability)	**Dynamic selection** (prediction explainability)
Wrappers	Forward selection, backward elimination (Kohavi and John, 1997)	Input reduction (Feng et al., 2018), representation erasure (leave-one-out) (Li et al., 2016b; Serrano and Smith, 2019), LIME (Ribeiro et al., 2016)
Filters	Pointwise mutual information (Church and Hanks, 1989), recursive feature elimination (Guyon et al., 2002)	Input gradient (Li et al., 2016a), layerwise relevance propagation (Bach et al., 2015), top-k softmax attention
Embedded	ℓ_1-regularization (Tibshirani, 1996), elastic net (Zou and Hastie, 2005)	Stochastic attention (Xu et al., 2015; Lei et al., 2016; Bastings et al., 2019), sparse attention (**this work**, §3.3)

Table 3.1: Overview of static and dynamic feature selection techniques.

Traditional feature selection methods (Guyon and Elisseeff, 2003) are mostly concerned with **model interpretability**, i.e., understanding how models behave globally. Feature selection happens *statically* during model training, after which irrelevant features are permanently deleted from the model. This contrasts with **prediction explainability** in neural networks, where feature selection happens *dynamically* at run time: here explanations are input-dependent, hence a feature not relevant for a particular input can be relevant for another. Are these two worlds far away? Guyon and Elisseeff (2003, §4) proposed a typology for traditional feature selection with three classes of methods, distinguished by how they model the interaction between their main two components, the *feature selector* and the *learning algorithm*. We argue that this typology can also be used to characterize various explanation methods, if we replace these two components by the *explainer* E and the *classifier* C, respectively.

- **Wrapper methods**, in the wording of Guyon and Elisseeff (2003), "utilize the learning machine of interest as a black box to score subsets of variables according to their predictive power." This means greedily searching over subsets of features, training a model with each candidate subset. In the dynamic feature selection world, this is somewhat reminiscent of the leave-one-out method of Li et al. (2016b), the ablative approach of Serrano and Smith (2019), and LIME (Ribeiro et al., 2016), which repeatedly queries the classifier to label new examples.

- **Filter methods** decide to include/exclude a feature based on an importance metric (such as feature counts or pairwise mutual information). This can be done as a preprocessing step or by training the model once and thresholding the feature weights. In dynamic feature selection, this is done when we examine the gradient of the prediction with respect to each input feature, and then select the features whose gradients have large magnitude (Li et al., 2016a; Arras et al., 2016; Jain and Wallace, 2019),[2] and when thresholding softmax attention scores to select relevant input features, as analyzed by Jain and Wallace (2019) and Wiegreffe and Pinter (2019).

- **Embedded methods**, in traditional feature selection, embed feature selection within the learning algorithm by using a sparse regularizer such as the ℓ_1-norm (Tibshirani, 1996). Features that receive zero weight become irrelevant and can be removed from the model. In dynamic feature selection, this encompasses methods where the classifier produces rationales together with its decisions (Lei et al., 2016; Bastings et al., 2019). We propose in §3.3 an alternative approach via **sparse attention** (Martins and Astudillo,

[2] In linear models this gradient equals the feature's weight.

2016; Peters et al., 2019), where the selection of words for the rationale resembles ℓ_1-regularization.

In §3.4, we frame each of the cases above as a communication process, where the explainer E aims to communicate a short message with the relevant features that triggered the classifier C's decisions to a layperson L. The three cases above are distinguished by the way C and E interact.

3.3 Embedded Sparse Attention

The case where the explainer E is embedded in the classifier C naturally favors faithfulness, since the mechanism that explains the decision (the *why*) can also influence it (the *how*).

Attention mechanisms (Bahdanau et al., 2015) allow visualizing relevant input features that contributed to the model's decision. However, the traditional softmax-based attention is *dense*, i.e., it gives *some* probability mass to every feature, even if small. The typical approach is to select the top-k words with largest attention weights as the explanation. However, this is not a truly embedded method, but rather a filter, and as pointed out by Jain and Wallace (2019) and Wiegreffe and Pinter (2019), it may not lead to faithful explanations.

An alternative is to embed in the classifier an attention mechanism that is inherently **selective**, i.e., which can produce sparse attention distributions natively, where some input features receive exactly zero attention. An extreme example is hard attention, which, as argued by DeYoung et al. (2020), provides more faithful explanations "by construction" as they discretely extract snippets from the input to pass to the classifier. A problem with hard attention is its non-differentiability, which complicates training (Lei et al., 2016; Bastings et al., 2019). We consider in this work a different approach: using end-to-end differentiable sparse attention mechanisms, via the **sparsemax** (Martins and Astudillo, 2016) and the recently proposed **1.5-entmax** transformation (Peters et al., 2019), described in detail in §2.1.2. These sparse attention transformations have been applied successfully to machine translation and morphological inflection (Peters et al., 2019; Correia et al., 2019). Words that receive non-zero attention probability are *selected* to be part of the explanation. This is an embedded method akin of the use of ℓ_1-regularization in static feature selection. We experiment with these sparse attention mechanisms in §3.5.

3.4 Explainability as Communication

We now have the necessary ingredients to describe our unified framework for comparing and designing explanation strategies, illustrated in Figure 3.1.

Our fundamental assumption is that explainability is intimately linked to the ability of an explainer to **communicate** the rationale of a decision in terms that can be understood by a human; we use the success of this communication as a criterion for how informative the explanation is.

3.4.1 The Classifier-Explainer-Layperson setup

Our framework draws inspiration from Lewis' signaling games (Lewis, 2008) and the recent work on emergent communication (Foerster et al., 2016; Lazaridou et al., 2016; Havrylov and Titov, 2017). Our starting point is the classifier $C : \mathcal{X} \to \mathcal{Y}$ which, when given an input $x \in \mathcal{X}$, produces a prediction $\hat{y} \in \mathcal{Y}$. This is the prediction that we want to explain. An explanation is a **message** $m \in \mathcal{M}$, for a predefined message space $\mathcal{M}$ (for

example, a rationale). The goal of the explainer E is to compose and **successfully communicate** messages m to a layperson L. The success of the communication is dictated by the ability of L to reconstruct $\hat{y}$ from m with high accuracy. In this work, we experiment with E and L being either humans or machines. Our framework is inspired by human-grounded evaluation through forward simulation/prediction, as proposed by Doshi-Velez and Kim (2017, §3.2). More formally:

- The **classifier** C is the model whose predictions we want to explain. For given inputs x, C produces $\hat{y}$ that are hopefully close to the ground truth y. We are agnostic about the kind of model used as a classifier, but we assume that it computes certain internal representations h that can be exposed to the explainer.

- The **explainer** E produces explanations for C's decisions. It receives the input x, the classifier prediction $\hat{y} = C(x)$, and optionally the internal representations h exposed by C. It outputs a message $m \in \mathcal{M}$ regarded as a "rationale" for $\hat{y}$. The message $m = E(x, \hat{y}, h)$ should be simple and compact enough to be easily transmitted and understood by the layperson L. The message space $\mathcal{M}$ must be composed of representations that are readable to humans (Wiegreffe and Pinter, 2019). For reasons that will be clear later, an explainer might not use all of its inputs to produce explanations, characterizing explanations with limited information. In this work, we constrain messages to be bags-of-words (BoWs) extracted from the textual input x.[3]

- The **layperson** L is a simple model (e.g., a linear classifier)[4] that receives the message m as input, and predicts a final output $\tilde{y} = L(m)$. The communication is successful if $\tilde{y} = \hat{y}$. Given a test set $\{x_1, \ldots, x_N\}$, we evaluate the **communication success rate** (CSR) as the fraction of examples for which the communication is successful:

$$\text{CSR} = \frac{1}{N} \sum_{n=1}^{N} \left[\left[C(x_n) = L(E(x_n, C(x_n)))\right]\right], \tag{3.1}$$

  where $[[\cdot]]$ is the Iverson bracket notation.

Under this framework, we regard the communication success rate as a quantifiable measure of explainability: a high CSR means that the layperson L is able to replicate the classifier C's decisions a large fraction of the time when presented with the messages given by the explainer E; this assesses how informative E's messages are for the two agents to communicate successfully.

Our framework is flexible, allowing different configurations for C, E, and L. In Figure 3.2 we show examples of how sparse attention can be treated as explanation in the context of natural language inference and machine translation. Later, in §3.5, we carry experiments with different explainers and laypeople for text classification, natural language inference, and machine translation.

Relation to filters and wrappers. In the wrapper and filter approaches described in §3.2, the classifier C and the explainer E are separate components. In these approaches, E works as a *post-hoc explainer*, querying C with new examples or requesting gradient information.

[3] Note that our framework is flexible about the choice of this message space $\mathcal{M}$. For example, explanations could also be *prototypes*, i.e., small subsets of training examples.

[4] The reason why we assume the layperson is a simple model is to encourage the explainer to produce simple and explanatory messages, in the sense that a simple model can learn with them. A more powerful layperson could potentially do well even with bad explanations.

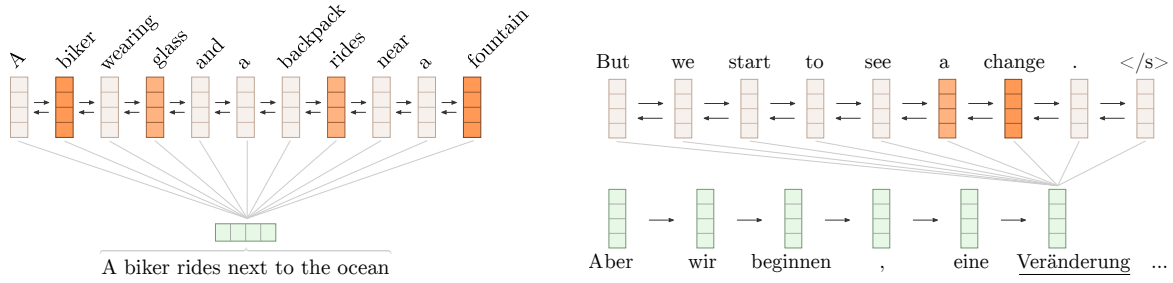


Figure 3.2: (left) Example of sparse attention for natural language inference. The selected premise words ("biker", "glass", "rides", and "fountain") form the message, together with the hypothesis in the bottom. (right) Example of sparse attention for machine translation. When the model is generating the word "Veränderung", the source words "a" and "change" are treated as explanation and sent as message.

Relation to embedded explanation. By contrast, in the embedded approaches of Lei et al. (2016) and the selective sparse attention introduced in §3.3, the explainer E is directly *embedded* as an internal component of the classifier C, returning the selected features as the message. This approach is arguably more faithful, as E is directly linked to the mechanism that produces C's decisions.

3.4.2 Joint training of explainer and layperson

So far we have assumed that E is given beforehand, chosen among existing explanation methods, and that L is trained to assess the explanatory ability of E. But can our framework be used to *create* new explainers by training E and L jointly? We will see how this can be done by letting E and L play a cooperative game (Lewis, 2008). The key idea is that they need to learn a communication protocol that ensures high CSR (Eq. 3.1). Special care needs to be taken to rule out "trivial" protocols and ensure plausible, potentially faithful, explanations. We propose a strategy to ensure this, which will be validated using human evaluation in §3.6.[5]

Let E_θ and layperson L_ϕ be **trained models** (with parameters θ and ϕ), learned together to optimize a multi-task objective with two terms:

- A **reconstruction term** that controls the information about the classifier's decision $\hat{y}$. We use a cross-entropy loss on the output of the layperson L, using $\hat{y}$ (and not the true label y) as the ground truth: $\mathcal{L}(\phi, \theta) = -\log p_\phi(\hat{y} \mid m)$, where m is the output of the explainer E_θ.
- A **faithfulness term** that encourages the explainer E to take into account the classifier's decision process when producing its explanation m. This is done by adding a squared loss term $\Omega(\theta) = \|\tilde{h}(E_\theta), h\|^2$ where $\tilde{h}$ is E's prediction of C's internal representation h.

The objective function is a combination of these two terms, $\mathcal{L}_\Omega(\phi, \theta) := \lambda\Omega(\theta) + \mathcal{L}(\phi, \theta)$. We used $\lambda = 1$ in our experiments. This objective is minimized in a training set that contains pairs $(x, \hat{y})$. Therefore, in this model the message m is latent and works as a "bottleneck" for the layperson L, which does not have access to the full input x, to guess the classifier's prediction $\hat{y}$ — related models have been devised in the context of emergent communication (Lazaridou et al., 2016; Foerster et al., 2016; Havrylov and Titov, 2017) and sparse autoencoders (Trifonov et al., 2018; Subramanian et al., 2018).

[5]Other approaches, as proposed by Lei et al. (2016) and Yu et al. (2019), develop rationalizers from cooperative or adversarial games between generators and encoders. However, those frameworks do not aim at explaining an external classifier.

We minimize the objective above with gradient backpropagation. To ensure end-to-end differentiability, during this joint training we use sparsemax attention (§3.3) to select the relevant words in the message. One important concern in this model is to prevent E and L from learning a trivial protocol to maximize CSR. To ensure this, we forbid E from including stopwords in its messages and during training we use a linear schedule for the probability of the explainer accessing the predictions of the classifier ($\hat{y}$), which are hidden otherwise, such that at the end of training, the explainer access $\hat{y}$ with probability β. In our experiments, we set β to 20% (chosen on the validation set as described in §A.3.2).

3.5 Experiments

We experimented with our framework in three NLP tasks: text classification, natural language inference (NLI), and machine translation.

3.5.1 Text classification and NLI

We picked the same datasets as Jain and Wallace (2019) and Wiegreffe and Pinter (2019), excluding the smallest ones. Concretely, we used 4 datasets (SST, IMDB, AgNews, Yelp) for text classification and one dataset (SNLI) for NLI, with statistics in Table 3.2. For SST, IMDB, and SNLI we used the standard splits, and for AgNews and Yelp we randomly split the dataset, leaving 85% for training and 15% for test.

Name	# Train	# Test	Avg. tokens	# Classes
SST	6920	1821	19	2
IMDB	25K	25K	280	2
AgNews	115K	20K	38	2
Yelp	5.6M	1M	130	5
SNLI	549K	9824	14 / 8	3
IWSLT	206K	2271	20 / 18	134,086

Table 3.2: Dataset statistics. The average number of tokens for SNLI is related to the premise and hypothesis, and for IWSLT to the source and target sentences.

Classifier C. For text classification, the input $x \in \mathcal{X}$ is a document and the output set $\mathcal{Y}$ is a set of labels (e.g. topics or sentiment labels). The message is a bag of words (BoW) extracted from the document. As in Jain and Wallace (2019) and Wiegreffe and Pinter (2019), our classifier C is an RNN with attention. For NLI, the input x is a pair of sentences (premise and hypothesis) and the labels in $\mathcal{Y}$ are entailment, contradiction, and neutral. We let messages be again BoWs, and we constrain them to be selected from the premise (and concatenated with the full hypothesis). We used a similar classifier as above, but with two independent BiLSTM layers, one for each sentence. We used the additive attention of Bahdanau et al. (2015) with the last hidden state of the hypothesis as the query and the premise vectors as keys.

We also experimented with RNN classifiers that replace softmax attention by 1.5-entmax (C_{ent}) and sparsemax (C_{sp}), and with the rationalizer models of Lei et al. (2016) (C_{bern}) and Bastings et al. (2019) (C_{hk}). Details about these classifiers and their hyperparameters are listed in §A.1. Table 3.3 reports the accuracy of all classifiers used in our experiments. The attention-based models all perform very similarly and generally better than

CLASSIFIER	SST	IMDB	AGNEWS	YELP	SNLI
BoW (L)	82.54	88.96	95.62	68.78	69.81
RNN softmax (C)	86.16	**91.79**	96.28	**75.80**	78.34
RNN 1.5-entmax (C_{ent})	86.11	91.72	96.30	75.72	79.20
RNN sparsemax (C_{sp})	**86.27**	91.52	96.37	75.72	78.78
Bernoulli (C_{bern})	81.99	87.65	95.68	70.12	79.24
HardKuma (C_{hk})	84.13	90.52	**96.38**	74.36	**85.49**

Table 3.3: Accuracies of the original classifiers on text classification and natural language inference.

the rationalizer models, except for SNLI, where the latter use a stronger model with decomposable attention. As expected, in general, all these classifiers outperform a bag-of-words model which is the model we use as the layperson.

Layperson L and explainer E. We used a simple linear BoW model as the layperson L. For NLI, the layperson sees the full hypothesis, encoding it with a BiLSTM. The BoW from the explainer is passed through a linear projection and summed with the last hidden state of the BiLSTM.

We evaluated the following explainers:

1. **Erasure**, a wrapper similar to the leave-one-out approaches of Jain and Wallace (2019) and Serrano and Smith (2019). We obtain the word with largest attention, zero out its input vector, and repass the whole input with the erased vector to the classifier C. We produce the message by repeating this procedure k times.

2. **Top-k gradients**, a filter approach that ranks words by their "input $\times$ gradient" product, $|\boldsymbol{x}_i \cdot \frac{\partial \hat{y}}{\partial \boldsymbol{x}_i}|$ (Ancona et al., 2018; Wiegreffe and Pinter, 2019). The top-k words are selected as the message.

3. **Top-k and selective attention:** We experimented both using attention as a *filter*, by selecting the top-k most attended words as the message, and *embedded* in the classifier C, by using the selective attentions described in §3.3 (1.5-entmax and sparsemax).

4. **The rationalizer models of Lei et al. (2016) and Bastings et al. (2019).** These models compose the message by stochastically sampling rationale words, respectively using Bernoulli and HardKuma distributions. For SNLI, since these models use decomposable attention instead of RNNs, we form the message by selecting all premise words that are linked with any hypothesis word via a selected Bernoulli variable.

We also report a **random** baseline, which randomly picks k words as the message. We show examples of messages for all explainers in §A.6.

Results. Table 3.4 reports results for the communication success rate (CSR, Eq. 3.1) and for the accuracy of the layperson (ACC_L). For each explainer, we indicate which classifier it is explaining; note that the CSR is only comparable across explainers that use the same classifier. The goal of this experiment is to answer the following questions:

- How do different explainers (wrappers, filters, embedded) compare to each other in CSR?

		SST		IMDB		AGNEWS		YELP		SNLI	
CLF.	EXPLAINER	CSR	ACC_L	CSR	ACC_L	CSR	ACC_L	CSR	ACC_L	CSR	ACC_L
C	Random	69.41	70.07	67.30	66.67	92.38	91.14	58.27	53.06	75.83	68.74
C	Erasure	80.12	81.22	92.17	88.72	97.31	95.41	78.72	68.90	77.88	70.04
C	Top-k gradient	79.35	79.24	86.30	83.93	96.49	94.86	70.54	62.86	76.74	69.40
C	Top-k softmax	84.18	82.43	93.06	89.46	**97.59**	95.61	81.00	70.18	78.66	71.00
C_{ent}	Top-k 1.5-entmax	**85.23**	**83.31**	93.32	89.60	97.29	**95.67**	82.20	70.78	80.23	73.39
C_{sp}	Top-k sparsemax	**85.23**	81.93	93.34	89.57	95.92	94.48	82.50	70.99	**82.89**	**74.76**
C_{ent}	Selec. 1.5-entmax	83.96	82.15	92.55	89.96	97.30	95.66	81.38	70.41	77.25	71.44
C_{sp}	Selec. sparsemax	**85.23**	81.93	93.24	89.66	95.92	94.48	83.55	71.60	82.04	73.46
C_{bern}	Bernoulli	82.37	78.42	91.66	86.13	96.91	94.43	84.93	66.89	76.81	69.65
C_{hk}	HardKuma	85.17	80.40	**94.72**	**90.16**	97.11	95.45	**87.39**	**71.64**	74.98	71.48

Table 3.4: CSR and layperson accuracy (ACC_L) for several explainers. For each explainer, we indicate the corresponding classifier from Table 3.3; in all cases the layperson is a BoW model. Only explainers of the same classifier can be compared in terms of CSR. Top rows report performance for random, wrapper and filter explainers, for fixed k-word messages (the values of k for the several datasets are {5, 10, 10, 10, 4}, respectively). Bottom rows correspond to embedded methods where k is given automatically via sparsity. The average k obtained by 1.5-entmax, sparsemax, Bernoulli and HardKuma are: SST: {4.65, 2.59, 6.10, 4.82}; IMDB: {28.23, 12.94, 39.40, 24.18}; AGNEWS {5.65, 4.14, 4.01, 9.68}; YELP: {60.61, 23.86, 9.15, 33.18}; SNLI: {12.96, 8.27, 15.04, 6.40}.

- Are selective sparse attention methods effective?
- Does a layperson guided by an explainer perform better than an unguided layperson that sees the entire document?
- How is the trade-off between message length and CSR?

The first thing to note is that, as expected, the random baseline is much worse than the other explainers, for all text classification datasets.[6] Among the non-trivial explainers, **the attention and erasure outperform gradient methods**: the erasure and top-k attention explainers have similar CSR, with a slight advantage for attention methods. Note that the attention explainers have the important advantage of requiring a single call to the classifier, whereas the erasure methods, being wrappers, require k calls. The worse performance of top-k gradient (less severe on AGNEWS) suggests that the words that locally cause bigger output changes are not necessarily the most informative ones.[7]

Regarding the different attention models (softmax, entmax, and sparsemax), we see that **sparse transformations tend to have slightly better ACC_L**, in addition to better ACC_C (see Table 3.3). The embedded sparse attention methods achieved communication scores on par with the top-k attention methods without a prescribed k, while producing, by construction, more faithful explanations. Both our proposed models (sparsemax and 1.5-entmax) seem generally more accurate than the Bernoulli model of Lei et al. (2016) and comparable to the HardKuma model of Bastings et al. (2019), with a much simpler training procedure, not requiring gradient estimation over stochastic computation graphs.

By comparing the accuracy of the classifiers in Table 3.3 with the ACC_L columns on Table 3.4, we see a consistent drop from the RNN classifiers to the layperson, regardless of the explainer. This is expected, since the

[6] This is less pronounced in SNLI, as the hypothesis alone already gives strong baselines (Gururangan et al., 2018).

[7] A potential reason is that attention directly influences C's decisions, being an inside component of the model. Gradients and erasure, however, are extracted after decisions are performed. The reason might be similar to filter methods being generally inferior to embedded methods in static feature selection, since they ignore feature interactions that may jointly play a role in model's decisions.

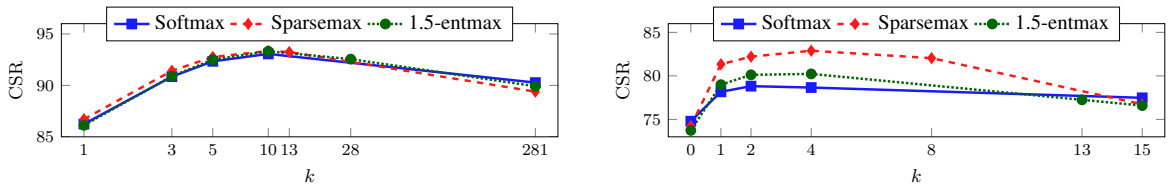


Figure 3.3: Message sparsity analysis for IMDB (left) and SNLI (right). For SNLI, $k = 0$ corresponds to a case where the layperson only sees the hypothesis. The rightmost entry represents an explainer that simply passes forward all words to the layperson. The average k for sparsemax and 1.5-entmax are, respectively: 13 and 28 for IMDB; 8 and 13 for SNLI.

layperson is a much weaker BoW classifier, and it only has access to a limited number of words in the document. However, we can also see that for several explainers the associated **laypersons are on par or outperform a BoW classifier that has access to the entire input**. This characteristic enables the direct usage of the layperson as a powerful, compact, and transparent model.[8]

Finally, Figure 3.3 shows the trade-off between the length of the message and the communication success rate for different values of k both for IMDB and SNLI. Interestingly, we observe that **CSR does not increase monotonically with k.** As k increases, CSR starts by increasing but then it starts dropping when k becomes too large. This matches our intuition: in the two extreme cases where $k = 0$ and where k is the document length (corresponding to a full bag-of-words classifier) the message has no information about how the classifier C behaves. By setting $k = 0$, meaning that the layperson L only looks at the hypothesis, the CSR is reasonably high ($\sim$74%), but as soon as we include a single word in the message this baseline is surpassed by 4 points or more. This is consistent with the finding by Gururangan et al. (2018), which suggests that we can achieve a high accuracy for SNLI by considering only the hypothesis as input.

3.5.2 Machine Translation

To compare explainers on a more challenging task with large $|\mathcal{Y}|$, we ran an experiment on neural machine translation (NMT), adapting the JoeyNMT framework (Kreutzer et al., 2019). We used the EN→DE IWSLT 2017 dataset (Cettolo et al., 2017), with the standard splits (Table 3.2).

We consider the decision taken by the NMT system when generating the t^{th} target word (y), given the source sentence x and the previously generated words $y_{1:t-1}$. Note that in this example $\mathcal{Y}$ is the entire target vocabulary. The message is the concatenation of k source words (ranked by importance, without any word order information) with the prefix $y_{1:t-1}$. The layperson must predict the target word given this limited information. Fig. 3.2 illustrates our setup.

We employed beam search decoding with beam size of 5, achieving a BLEU score of 20.49, 21.12 and 20.75 for softmax, 1.5-entmax and sparsemax, respectively. The layperson is a model that uses an unidirectional 256D LSTM to encode the translation prefix, and a feed-forward layer to encode the concatenation of k source word embeddings (the message) to a 256D vector. The two vectors are concatenated and passed to a linear output layer to predict the next word $\tilde{y} \in \mathcal{Y}$. Results comparing different filtering methods varying k are shown in Table 3.5.

We show the CSR as we varied $k \in \{0, 1, 3, 5\}$. There are two main findings. First, we see again that **top-k**

[8]Since the layperson is trained on the classifier's predictions and not on ground-truth labels, this corresponds to a scenario similar to knowledge distillation (Hinton et al., 2015).

EXPLAINER	$k=0$	$k=1$	$k=3$	$k=5$
Random	-	-	-	23.32
Top-k gradient	21.99	35.21	38.33	40.30
Top-k softmax	21.99	**62.58**	62.82	62.64
Top-k 1.5-entmax	**22.31**	62.53	**63.48**	**62.69**
Top-k sparsemax	22.14	62.21	61.94	61.92

Table 3.5: Results for IWSLT in terms of CSR.

attention outperforms top-k gradient, in this case with a wider margin. Second, we see that all methods perform better as we increase k, albeit we can see a performance degradation of attention-based explainers for $k = 5$. An interesting case is when $k = 0$, meaning that L has no access to the source sentence, behaving like an unconditioned language model. In this case the performance is much worse, indicating that both explainers are selecting relevant tokens when $k > 0$. This becomes clearer by looking at a random explainer, which yields a CSR of 23.32 for $k = 5$, very close to the CSRs obtained with $k = 0$. Moreover, as we found for IMDB and SNLI, increasing k does not necessarily leads to a higher CSR on IWSLT. Fig.3.4 depicts this finding.

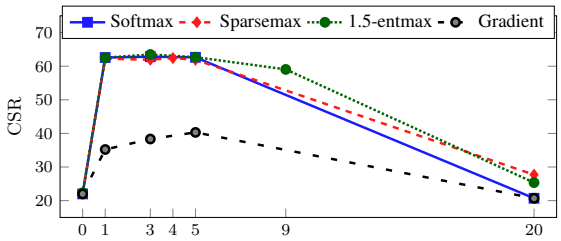


Figure 3.4: Message sparsity analysis for IWSLT. $k = 0$ corresponds to a case where the layperson only sees the translation prefix. The rightmost entry is the average length of the examples in the test set, and therefore it represents an explainer that simply pass forward all words to the layperson (i.e. a full bag-of-words). The average k for sparsemax and 1.5-entmax are, respectively: 4.5 and 9.4.

3.6 Human Evaluation

To fully assess the quality of the explanations in a more realistic forward simulation setting, we performed human evaluations, where the layperson L is a human instead of a machine.

Joint training of E and L. So far we compared several explainers, but what happens if we train E and L jointly to optimize CSR directly, as described in §3.4.2? We experiment with the IMDB and SNLI datasets, comparing with using humans for either the layperson, the explainer, or both.

Human layperson. We randomly selected 200 documents for IMDB and SNLI to be annotated by humans. The extracted explanations (i.e. the selected words) were shuffled and displayed as a cloud of words to two annotators, who were asked to predict the label of each document when seeing only these explanations. For SNLI, we show the entire hypothesis as raw text and the premise as a cloud of words. The agreement between annotators and other annotation details can be found in §A.5.

CLF.	EXPLAINER	IMDB					SNLI				
		k	CSR_H	CSR_L	ACC_H	ACC_L	k	CSR_H	CSR_L	ACC_H	ACC_L
C	Erasure	5.0	89.25	94.00	86.25	90.00	4.0	72.50	73.50	83.50	70.00
C	Top-k gradient	5.0	73.50	84.50	73.00	80.50	4.0	65.75	72.50	76.75	68.00
C	Top-k softmax	5.0	89.25	93.00	88.25	88.00	4.0	72.00	76.50	82.75	71.50
C_{ent}	Top-k 1.5-entmax	5.0	89.25	92.50	85.75	86.50	4.0	70.00	81.50	80.50	76.50
C_{sp}	Top-k sparsemax	5.0	89.00	89.50	87.50	88.00	4.0	68.25	88.00	80.25	77.00
C_{ent}	Selec. 1.5-entmax	27.2	86.50	92.50	84.00	89.50	12.9	75.25	77.00	87.00	77.00
C_{sp}	Selec. sparsemax	12.8	87.75	92.50	86.75	89.00	8.0	72.25	82.00	85.00	79.00
C_{bern}	Bernoulli	39.4	79.00	93.50	75.00	87.00	15.2	74.50	76.00	86.75	69.50
C_{hk}	HardKuma	24.3	83.75	93.50	80.75	89.00	6.4	79.25	71.50	**87.50**	68.50
C	Joint E and L	2.7	**96.75**	**98.50**	**89.25**	**91.50**	2.8	58.00	**93.50**	70.00	78.50
-	Human highlights	-	-	-	-	-	2.8	**83.25**	83.50	83.25	**83.50**

Table 3.6: Results of the human evaluation. Reported are average message length k, human layperson CSR_H/ACC_H, and machine layperson CSR_L/ACC_L. Only explainers of the same classifier can be compared in terms of CSR.

Human explainer. We also consider explanations generated by humans rather than machines. To this end, we used the e-SNLI corpus (Camburu et al., 2018), which extends the SNLI with human rationales. Since the e-SNLI corpus does not provide highlights over the premise for neutral pairs, we removed them from the test set.[9] We summarize our results in Table 3.6.

As in our previous experiments, better results were found both in terms of CSR and ACC for top-k attention methods in comparison to top-k gradient. The ACC of erasure, top-k attention models, and human highlights explainers are close, reinforcing again the good results for these explainers. Among the different attention explainers, we see that selective attention explainers (§3.3) got very high ACC_H, outperforming top-k explainers for SNLI.

By carefully optimizing the communication (§3.4.2), we see that the joint explainer outperformed all the other explainers in terms of ACC_L and CSR_L, and was able to achieve a very high human performance on IMDB, largely surpassing other systems in CSR_H and ACC_H. This shows the potential of our communication-based framework to also develop new post-hoc explainers with good forward simulation properties. However, for SNLI, the joint explainer had much lower CSR_H and ACC_H, suggesting that for this task more sophisticated explainers are required. Outputs for these explainers can be consulted in §A.6.

3.7 Related Work

There is a large body of work on analysis and interpretation of neural networks. Our work focuses on *prediction explainability*, different from transparency or model interpretability (Doshi-Velez and Kim, 2017; Lipton, 2018; Gilpin et al., 2018).

Rudin (2019) defines explainability as a plausible reconstruction of the decision-making process, and Riedl (2019) argues that it mimics what humans do when rationalizing past actions. This inspired our post-hoc explainers in §3.4.2 and their use of the faithfulness loss term.

Recent works questioned the interpretative ability of attention mechanisms (Jain and Wallace, 2019; Serrano

[9]Note that the human rationales from eSNLI are not explanations about C, since the humans are explaining the gold labels. Therefore, we have CSR=ACC always.

and Smith, 2019). Wiegreffe and Pinter (2019) distinguished between faithful and plausible explanations and introduced several diagnostic tools. Mullenbach et al. (2018) use human evaluation to show that attention mechanisms produce plausible explanations, consistent with our findings in §3.6. None of these works, however, considered the sparse selective attention mechanisms proposed in §3.3. Hard stochastic attention has been considered by Xu et al. (2015); Lei et al. (2016); Alvarez-Melis and Jaakkola (2017); Bastings et al. (2019), but a comparison with sparse attention and explanation strategies was still missing.

Besides attention-based methods, many other explainers have been proposed using gradients (Bach et al., 2015; Montavon et al., 2018; Ding et al., 2019), leave-one-out strategies (Feng et al., 2018; Serrano and Smith, 2019), or local perturbations (Ribeiro et al., 2016; Koh and Liang, 2017), but a link with filters and wrappers in the feature selection literature has never been made. We believe the connections revealed in §3.2 may be useful to develop new explainers in the future.

Our trained explainers from §3.4.2 draw inspiration from emergent communication (Lazaridou et al., 2016; Foerster et al., 2016; Havrylov and Titov, 2017). Some of our proposed ideas (e.g., using sparsemax for end-to-end differentiability) may also be relevant to that task. Our work is also related to sparse auto-encoders, which seek sparse overcomplete vector representations to improve model interpretability (Faruqui et al., 2015; Trifonov et al., 2018; Subramanian et al., 2018). In contrast to these works, we consider the non-zero attention probabilities as a form of explanation.

Some recent work (Yu et al., 2019; DeYoung et al., 2020) advocates *comprehensive* rationales. While comprehensiveness could be useful in our framework to prevent trivial communication protocols between the explainer and layperson, we argue that it is not always a desirable property, since it leads to longer explanations and an increase of human cognitive load. In fact, our analysis of CSR as a function of message length (Figure 3.3) suggests that shorter explanations might be preferable. This is aligned to the "explanation selection" principle articulated by Miller (2019, §4): *"Similar to causal connection, people do not typically provide all causes for an event as an explanation. Instead, they select what they believe are the most relevant causes."* Our sparse, selective attention mechanisms proposed in §3.3 are inspired by this principle.

3.8 Conclusions and Subsequent Works

We proposed a unified framework that regards explainability as a communication problem between an explainer and a layperson about a classifier's decision. In doing so, we organized existing approaches in a typology that makes a bridge between traditional feature selection and modern explanation techniques. Following this typology, we proposed new embedded methods based on selective attention, and post-hoc explainers trained to optimize communication success. In our experiments, we observed that attention mechanisms and erasure tend to outperform gradient methods on communication success rate, using both machines and humans as the layperson, and that selective attention is effective, while simpler to train than stochastic rationalizers.

After our publication (Treviso and Martins, 2020), our work served as a base for the simulability framework of Pruthi et al. (2022), who devised a novel strategy for incorporating explanations into the layperson' learning process. Their framework rebrands the classifier and layperson as teacher and student, respectively, and uses a scaffolding setup to guide the student's learning with the help of explanations provided by the teacher. Attention-based methods were shown to outperform other methods, which supports our findings. Notably, this design

effectively eliminates the impact of trivial protocols on simulability accuracy because the layperson does not have access to explanations at test time. In Chapter 8, we explore this framework further by designing a new explainer that uses sparsity to identify relevant heads in transformer-based quality estimation models. Hase et al. (2020) follow our aspiration of assessing explainability via simulability and propose Leakage-Adjusted Simulability (LAS), a metric for evaluating free-text explanations. Consistent with our findings, they discovered that optimizing simulability leads to better explanations.

Another connection to our work is the improved learnable sparse explainer with structured constraints proposed by Guerreiro and Martins (2021), called SPECTRA (Sparse Structured Text Rationalization), which uses LP-SparseMAP (Niculae and Martins, 2020) to induce contiguous spans and limit the number of selected tokens in the explanation. We leverage SPECTRA in the next Chapter to guide the generation of high quality counterfactuals, a necessary ingredient for computing counterfactual simulability in an automatic way.

Concurrent work. Notably, coinciding with the public release of our work, a concurrent work led by Hase and Bansal (2020) proposed a human-centric communication setup and also demonstrated the significance of leveraging simulability for evaluating explanations.

4

CREST: A Joint Framework for Rationalization and Counterfactual Text Generation

Contents

In this chapter, we present a framework called CREST (ContRastive Edits with Sparse raTionalization), which combines two effective and complementary methods for interpreting and training NLP models: selective rationalization and counterfactual text generation. We begin by leveraging our previous findings and employing an embedded method based on sparse attention to produce explanations. Specifically, we explore SPECTRA (Guerreiro and Martins, 2021), which replaces the α-entmax transformation with LP-SparseMAP (Niculae and Martins, 2020) in order to induce contiguity and limit the maximum number of selected tokens in the explanation. We then add a natural layer to our simulability setup to evaluate explanations in terms of counterfactual simulation, where the aim is to predict a contrastive outcome, rather than the classifier's prediction, given an explanation as input.

While selective rationales and counterfactual examples are effective methods for interpreting NLP models, prior work has not examined how they can be combined to leverage their complementary advantages. CREST addresses this limitation by introducing a joint framework for selective rationalization and counterfactual text generation, leading to improvements in counterfactual quality, model robustness, and interpretability. First, CREST generates valid counterfactuals that are more fluent, diverse, and plausible than those produced by previous methods. We show that these counterfactuals can be effectively used for data augmentation at scale, reducing the need for human-generated examples. Second, we introduce a new loss function that leverages CREST counterfactuals to regularize selective rationales and show that this regularization improves both model robustness and rationale quality, compared to methods that do not leverage CREST counterfactuals. Our results demonstrate that CREST successfully bridges the gap between selective rationales and counterfactual examples, addressing the limitations of existing methods and providing a more comprehensive view of a model's predictions.

This chapter is based on (Treviso et al., 2023b)—currently under review.

4.1 Motivation

As NLP models have become larger and less transparent, there has been a growing interest in developing methods for finer-grained interpretation and control of their predictions. One class of methods leverages **selective rationalization** (Lei et al., 2016; Bastings et al., 2019), which trains models to first select *rationales*, or subsets of relevant input tokens, and then make predictions based only on the selected rationales. These methods offer increased interpretability, as well as learning benefits, such as improved robustness to input perturbations (Jain et al., 2020; Chen et al., 2022). Another class of methods generates **counterfactual examples**, or modifications to input examples that change their labels. By providing localized views of decision boundaries, counterfactual examples can be used as explanations of model predictions, contrast datasets for fine-grained evaluation, or new training datapoints for learning more robust models (Ross et al., 2021; Gardner et al., 2020; Kaushik et al., 2020).

This work is motivated by the observation that selective rationales and counterfactual examples allow for interpreting and controlling model behavior through different means: selective rationalization improves model transparency by weaving interpretability into a model's internal decision-making process, while counterfactual examples provide external signal more closely aligned with human causal reasoning (Wu et al., 2021).

We propose to combine both methods to leverage their complementary advantages with **CREST**, a joint

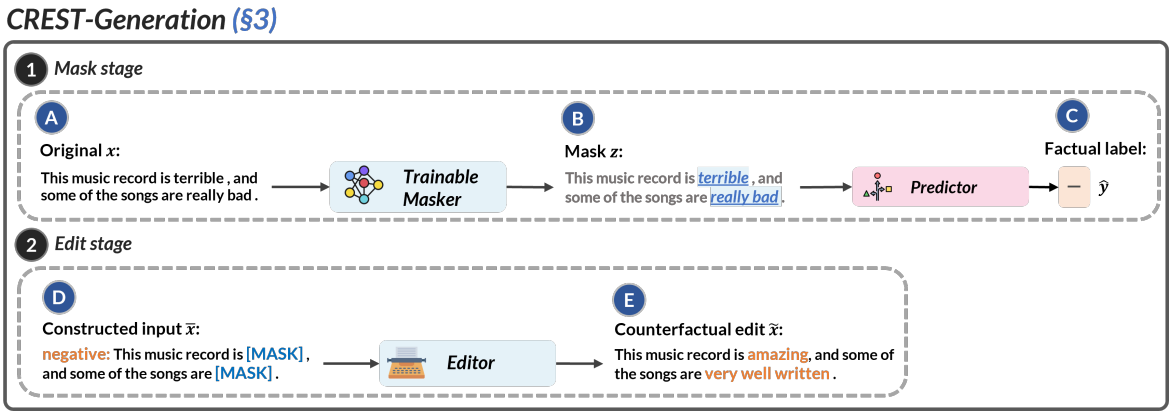


Figure 4.1: Our generation procedure consists of two stages: (i) a mask stage that highlights relevant tokens in the input through a learnable masker; and (ii) an edit stage, which receives a masked input and uses a masked language model to infill spans conditioned on a prepended label.

framework for rationalization and counterfactual text generation. CREST first generates high-quality counterfactuals (Figure 4.1), then leverages those counterfactuals to encourage consistency across "flows" for factual and counterfactual inputs (Figure 4.2). In doing so, CREST unifies two key important dimensions of interpretability introduced by Doshi-Velez and Kim (2017, §3.2), forward simulation and counterfactual simulation. Our main contributions are:[1]

- We present **CREST-Generation** (Figure 4.1), a novel approach to generating counterfactual examples by combining sparse rationalization with span-level masked language modeling (§4.3), which produces valid, fluent, and diverse counterfactuals (§4.4, Table 4.1).
- We introduce **CREST-Rationalization** (Figure 4.2), a novel approach to regularizing rationalizers. CREST-Rationalization decomposes a rationalizer into factual and counterfactual flows and encourages agreement between the rationales for both (§4.5).
- We show that CREST-generated counterfactuals can be effectively used to increase model robustness, leading to larger improvements on contrast and out-of-domain datasets than using manual counterfactuals (§4.6.2, Tables 4.2 and 4.3).
- We find that rationales trained with CREST-Rationalization not only are more plausible, but also achieve higher forward and counterfactual simulabilities (§4.6.3, Table 4.4).

Overall, our experiments show that CREST successfully combines the benefits of counterfactual examples and selective rationales to improve the quality of each, resulting in a more interpretable and robust learned model.

4.2 Background

4.2.1 Rationalizers

As detailed in §2.3.1, the traditional framework of rationalization involves training two components cooperatively: the *generator*—which consists of an encoder and an explainer—and the *predictor*. The encoder module

[1] Our code will be made publicly available (MIT license).

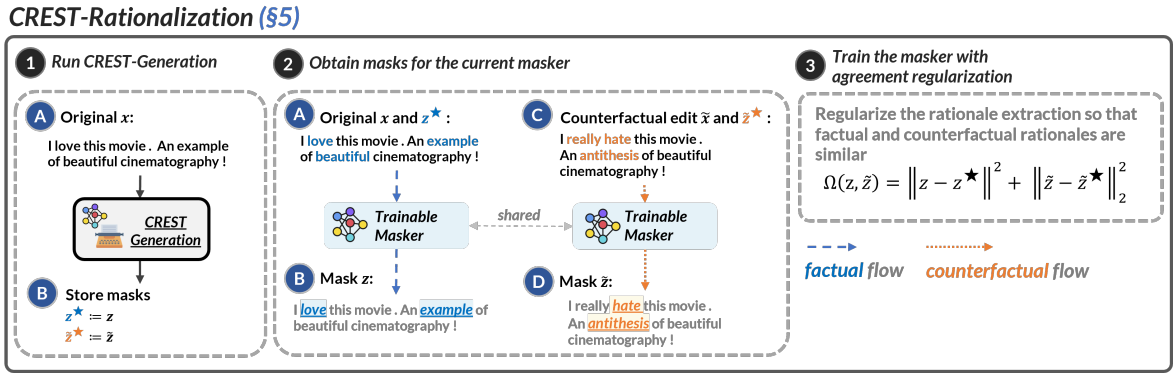


Figure 4.2: Overview of CREST-Rationalization. We start by passing an input $\boldsymbol{x}$ through CREST-Generation, which yields a counterfactual edit $\tilde{\boldsymbol{x}}$ along side two masks: $\boldsymbol{z}^\star$ for the original input, and $\tilde{\boldsymbol{z}}^\star$ for the counterfactual. Next, we train a new rationalizer (masker) decomposed into two flows: a **factual flow** that takes in $\boldsymbol{x}$ and produces a rationale $\boldsymbol{z}$, and a **counterfactual flow** that receives $\tilde{\boldsymbol{x}}$ and produces a rationale $\tilde{\boldsymbol{z}}$. Lastly, we employ a regularization term $\Omega(\boldsymbol{z}, \tilde{\boldsymbol{z}})$ to encourage agreement between rationales for original and counterfactual examples.

(enc) converts n input tokens into d-dimensional hidden state vectors $\boldsymbol{H} \in \mathbb{R}^{n \times d}$, which are passed to the explainer (expl) to generate a latent mask $\boldsymbol{z} \in \{0, 1\}^n$. The latent mask serves as the rationale since it is used to select a subset of the input $\boldsymbol{x} \odot \boldsymbol{z}$, which is then passed to the predictor module (pred) to produce a final prediction $\hat{y} \in \mathcal{Y}$, where $\mathcal{Y} = \{1, ..., k\}$ for k-class classification. The full process can be summarized as follows:

$$\boldsymbol{z} = \mathsf{expl}(\mathsf{enc}(\boldsymbol{x}; \phi); \gamma), \tag{4.1}$$

$$\hat{y} = \mathsf{pred}(\boldsymbol{x} \odot \boldsymbol{z}; \theta), \tag{4.2}$$

where $\Theta = \{\phi, \gamma, \theta\}$ represents trainable parameters. To ensure that the explainer does not select all tokens (i.e., $z_i = 1, \forall i$), sparsity is usually encouraged in the rationale extraction. Moreover, explainers can also be encouraged to select contiguous words, as there is some evidence that it improves readibility (Jain et al., 2020). These desired properties may be encouraged via regularization terms during training (Lei et al., 2016; Bastings et al., 2019), or via application of sparse mappings (Treviso and Martins, 2020; Guerreiro and Martins, 2021).

In this work, we will focus specifically on the SPECTRA rationalizer (Guerreiro and Martins, 2021): this model leverages an explainer that extracts a deterministic structured mask $\boldsymbol{z}$ by solving a constrained inference problem with SparseMAP (Niculae et al., 2018). SPECTRA has been shown to achieve comparable performance with other rationalization approaches, in terms of end-task performance, plausibility with human explanations, and robustness to input perturbation (Chen et al., 2022). Moreover, it is easier to train than other stochastic alternatives (Lei et al., 2016; Bastings et al., 2019), and, importantly, it allows for simple control over the properties of the rationales such as sparsity via its constrained inference formulation: by setting a budget B on the rationale extraction, SPECTRA ensures that the rationale size will not exceed $\lceil Bn \rceil$ tokens.

4.2.2 Counterfactuals

In NLP, counterfactuals refer to alternative texts that describe a different outcome than what is encoded in a given factual text. Prior works (Verma et al., 2020) have focused on developing methods for generating counterfactuals that adhere to several key properties, including:

- **Validity**: the generated counterfactuals should encode a different label from the original text.

- **Closeness**: the changes made to the text should be small, not involving large-scale rewriting of the input.
- **Fluency**: the generated counterfactuals should be coherent and grammatically correct.
- **Diversity**: the method should generate a wide range of counterfactuals with diverse characteristics, rather than only a limited set of variations.

While many methods for automatic counterfactual generation exist (Wu et al., 2021; Robeer et al., 2021; Dixit et al., 2022), our work is mostly closely related to MiCE (Ross et al., 2021), which generates counterfactuals in a two stage process that involves masking the top-k tokens with the highest ℓ_1 gradient attribution of a pretrained classifier, and infilling tokens for masked position with a T5-based model (Raffel et al., 2020). MiCE further refines the resultant counterfactual with a binary search procedure to seek strictly *minimal* edits. However, this process is computationally expensive and, as we show in §4.4.2, optimizing for closeness can lead to counterfactuals that are less valid, fluent, and diverse. Next, we present an alternative method that overcomes these limitations while still producing counterfactuals that are close to original inputs.

4.3 CREST-Generation

We now introduce CREST (ContRastive Edits with Sparse raTionalization), a framework that combines selective rationalization and counterfactual text generation. CREST has two key components: (i) **CREST-Generation** offers a controlled approach to generating counterfactuals, which we show are valid, fluent, and diverse (§4.4.2); and (ii) **CREST-Rationalization** leverages these counterfactuals through a novel regularization technique encouraging agreement between rationales for original and counterfactual examples. We demonstrate that combining these two components leads to models that are more robust (§4.6.2) and interpretable (§4.6.3). We describe CREST-Generation below and CREST-Rationalization in §4.5.

Formally, let $\boldsymbol{x} = \langle x_1, ..., x_n \rangle$ represent a factual input text with a label y_f. We define a counterfactual as an input $\tilde{\boldsymbol{x}} = \langle x_1, ..., x_m \rangle$ labeled with y_c such that $y_f \neq y_c$. A counterfactual generator is a mapping that transforms the original text $\boldsymbol{x}$ to a counterfactual $\tilde{\boldsymbol{x}}$. Like MiCE, our approach for generating counterfactuals consists of two stages, as depicted in Figure 4.1: the mask and the edit stages.

Mask stage. We aim to find a mask vector $\boldsymbol{z} \in \{0, 1\}^n$ such that tokens x_i associated with $z_i = 1$ are relevant for the factual prediction $\hat{y}_f$ of a particular classifier C. To this end, we employ a SPECTRA rationalizer as the **masker**. Concretely, we pretrain a SPECTRA rationalizer on the task at hand with a budget constraint B, and define the mask as the rationale vector $\boldsymbol{z} \in \{0, 1\}^n$ (see §4.2.1).

Edit stage. Here, we create edits by infilling the masked positions using an **editor** module G, such as a masked language model: $\tilde{\boldsymbol{x}} \sim G_{\text{LM}}(\boldsymbol{x} \odot \boldsymbol{z})$. In order to infill spans rather than single tokens, we follow MiCE and use a T5-based model to infill spans for masked positions. During training, we fine-tune the editor to infill spans by prepending the gold target label y_f to the input. In order to generate counterfactual edits at test time, we prepend a counterfactual label y_c instead, and sample counterfactuals using beam search.

Overall, our procedure differs from that of MiCE in the mask stage: instead of extracting a mask via gradient-based attributions and subsequent binary search, we leverage SPECTRA to find an optimal mask. Interestingly,

Method	IMDB					SNLI				
	val. ↑	fl. ↓	div. ↓	clo. ↓	#tks	val. ↑	fl. ↓	div. ↓	clo. ↓	#tks
Chance baseline	50.20	-	-	-	-	52.70	-	-	-	-
References	97.95	66.51	-	-	184.4	96.75	63.52	-	-	7.5
Manual edits	93.44	72.89	81.67	0.14	183.7	93.88	65.25	35.82	0.42	7.7
PWWS	28.07	101.91	74.56	**0.16**	179.0	17.97	160.11	31.81	0.36	6.8
CFGAN	-	-	-	-	-	34.46	155.84	68.94	**0.23**	7.0
PolyJuice	36.69	68.59	56.41	0.45	94.6	41.80	62.02	39.01	0.40	11.6
MiCE (bin. search)	72.13	76.72	73.76	0.20	171.3	76.17	63.94	42.18	0.35	7.9
MiCE (30% mask)	76.80	79.35	49.64	0.39	161.3	77.26	**59.71**	34.08	0.40	8.3
MiCE (50% mask)	83.20	89.92	**20.71**	0.65	115.7	**84.12**	68.32	**24.27**	0.52	7.6
CREST (30% mask)	75.82	67.29	57.58	0.33	180.9	75.45	62.00	41.36	0.29	7.4
CREST (50% mask)	**93.24**	**50.69**	23.08	0.66	193.9	81.23	61.96	30.53	0.41	7.4

Table 4.1: Intrinsic evaluation of counterfactuals generated by various methods. Validity is computed as the accuracy of an off-the-shelf RoBERTa-base classifier in relation to the gold counterfactual label (not available for PWWS and PolyJuice); fluency is determined by the perplexity score given by GPT-2 large; diversity is computed with self-BLEU; and closeness is reported by the (normalized) edit distance to the factual input. In addition, we report average number of tokens in the input.

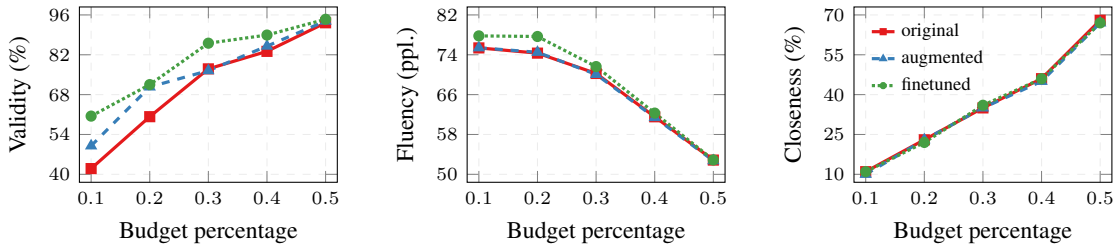


Figure 4.3: Sparsity analysis of CREST on IMDB with different budget percentages. The *original* curves show the performance of CREST without any changes, while the *augmented* and *finetuned* curves show the performance of CREST when using manually crafted counterfactuals for data augmentation or finetuning, respectively.

by doing so, we not only avoid the computationally expensive binary search procedure, but we also open up new opportunities: as our masking process is differentiable, we can optimize our masker to enhance the quality of both the counterfactuals (§4.4.2) and the selected rationales (§4.6.3). We will demonstrate the latter with our proposed CREST-Rationalization setup (§4.5). All implementation details for the masker and the editor can be found in §B.2.

4.4 Evaluating CREST Counterfactuals

This section presents an extensive comparison of counterfactuals generated by different methods.

4.4.1 Experimental Setting

Data and evaluation. We experiment with our counterfactual generation framework on two different tasks: sentiment classification using IMDB (Maas et al., 2011) and natural language inference (NLI) using SNLI (Bowman et al., 2015). In sentiment classification, we only have a single input to consider, while NLI inputs consist of a premise and a hypothesis, which we concatenate to form a single input. To assess the quality of our automatic counterfactuals, we compare them to manually crafted counterfactuals in the revised IMDB and SNLI datasets created by Kaushik et al. (2020). More dataset details can be found in §B.1.

Training. We employ a SPECTRA rationalizer with a T5-small architecture as the masker, and train it for 10 epochs on the full IMDB and SNLI datasets. We also use a T5-small architecture for the editor, and train it for 20 epochs with early stopping, following the same training recipe as MiCE. Full training details can be found in §B.2.3.

Generation. As illustrated in Figure 4.1, at test time we generate counterfactuals by prepending a contrastive label to the input and passing it to the editor. For sentiment classification, this means switching between positive and negative labels. For NLI, in alignment with Dixit et al. (2022), we adopt a refined approach by restricting the generation of counterfactuals to entailments and contradictions only, therefore ignoring neutral examples, which have a subtle semantic meaning. In contrast, our predictors were trained using neutral examples, and in cases where they predict the neutral class, we default to the second-most probable class.

Baselines. We compare our approach with four open-source baselines that generate counterfactuals: PWWS (Ren et al., 2019), PolyJuice (Wu et al., 2021), CounterfactualGAN (Robeer et al., 2021),[2] and MiCE (Ross et al., 2021). In particular, to ensure a fair comparison with MiCE, we apply three modifications to the original formulation: (i) we replace its RoBERTa classifier with a T5-based classifier (as used in SPECTRA); (ii) we disable its validity filtering;[3] (iii) we report results with and without the binary search procedure by fixing the percentage of masked tokens.

Metrics. To determine the general **validity** of counterfactuals, we report the accuracy of an off-the-shelf RoBERTa-base classifier available in the HuggingFace Hub.[4] Moreover, we measure **fluency** using perplexity scores from GPT-2 large (Radford et al., 2019) and **diversity** with self-BLEU (Zhu et al., 2018). Finally, we quantify the notion of **closeness** by computing the normalized edit distance to the factual input and the average number of tokens in the document.

4.4.2 Results

Results are presented in Table 4.1. As expected, manually crafted counterfactuals achieve high validity, significantly surpassing the chance baseline and establishing a reliable reference point. For IMDB, we find that CREST outperforms other methods by a wide margin in terms of validity and fluency. At the same time, CREST's validity is comparable to the manually crafted counterfactuals, while surprisingly deemed more fluent by GPT-2. Moreover, we note that our modification of disabling MiCE's minimality search leads to counterfactuals that are more valid and diverse but less fluent and less close to the original inputs.

For SNLI, this modification allows MiCE to achieve the best overall scores, closely followed by CREST. However, when controlling for closeness, we observe that CREST outperforms MiCE: at closeness of $\sim$0.30, CREST (30% mask) outperforms MiCE with binary search in terms of fluency and diversity. Similarly, at a closeness of $\sim$0.40, CREST (50% mask) surpasses MiCE (30% mask) across the board. As detailed in §B.3, CREST's counterfactuals are more valid than MiCE's for all closeness bins lower than 38%. We provide

[2] Despite many attempts, CounterfactualGAN did not converge on IMDB, possibly due to the long length of the inputs.
[3] MiCE with binary search uses implicit validity filtering throughout the search process to set the masking percentage.
[4] `mtreviso/roberta-base-imdb`, `mtreviso/roberta-base-snli`.

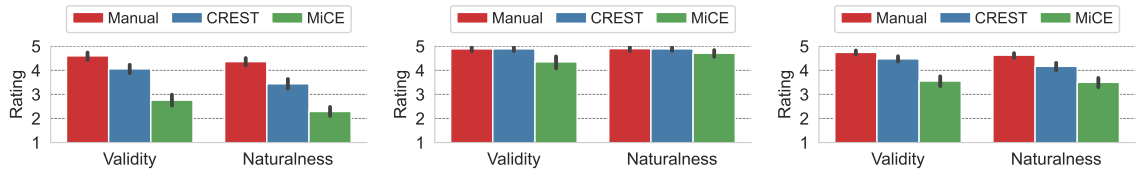


Figure 4.4: Human study results for counterfactuals produced manually and automatically by CREST and MiCE on IMDB (left), SNLI (middle), and overall (right).

examples of counterfactuals produced by CREST and MiCE in Appendix B.6. Finally, we note that CREST is highly affected by the masking budget, which we explore further next.

Sparsity analysis. We investigate how the number of edits affects counterfactual quality by training maskers with increasing budget constraints (as described in §4.2.1). The results in Figure 4.3 show that with increasing masking percentage, generated counterfactuals become less textually similar to original inputs (i.e., less close) but more valid and fluent. This inverse relationship demonstrates that strict minimality, optimized for in methods like MiCE, comes with tradeoffs in counterfactual quality, and that the sparsity budget in CREST can be used to modulate the trade-off between validity and closeness. In Figure 4.3 we also examine the benefit of manually crafted counterfactuals in two ways: (i) using these examples as additional training data; and (ii) upon having a trained editor, further fine-tuning it with these manual counterfactuals. The results suggest that at lower budget percentages, exploiting a few manually crafted counterfactuals to fine-tune CREST can improve the validity of counterfactuals without harming fluency.

Validity filtering. As previously demonstrated by Wu et al. (2021) and Ross et al. (2022b), it is possible to filter out potentially disfluent or invalid counterfactuals by passing all examples to a classifier and discarding the subset with incorrect predictions. In our case, we use the predictor associated with the masker as the classifier. We found find that applying this filtering increases the validity of IMDB counterfactuals from 75.82 to 86.36 with $B = 0.3$, and from 93.24 to 97.36 with $B = 0.5$. For SNLI, validity jumps from 75.45 to 96.39 with $B = 0.3$, and from 81.23 to 96.67 with $B = 0.5$. These results indicate that CREST can rely on its predictor to filter out invalid counterfactuals, a useful characteristic for doing data augmentation, as we will see in §4.6.2.

4.4.3 Human Study

We conduct a small-scale human study to evaluate the quality of counterfactuals produced by MiCE and CREST with 50% masking percentage. Annotators were tasked with rating counterfactuals' *validity* and *naturalness* (e.g., based on style, tone, and grammar), each using a 5-point Likert scale. Two fluent English annotators rated 50 examples from the IMDB dataset, and two others rated 50 examples from SNLI. We also evaluate manually created counterfactuals to establish a reliable baseline. More details can be found in §B.4.

The study results, depicted in Figure 4.4, show that humans find manual counterfactuals to be more valid and natural compared to automatically generated ones. Furthermore, CREST's counterfactuals receive higher ratings for validity and naturalness compared to MiCE, aligning with the results obtained from automatic metrics. Interestingly, counterfactuals for SNLI appear more valid and natural compared to those for IMDB, highlighting the challenge in generating counterfactuals for long movie reviews.

4.5 CREST-Rationalization

Now that we have a method that generates high-quality counterfactual examples, a natural step is to use these examples for data augmentation. However, vanilla data augmentation does not take advantage of the paired structure of original/contrastive examples and instead just treats them as individual datapoints. In this section, we present CREST's second component, CREST-Rationalization (illustrated in Figure 4.2), which leverages the relationships between factual and counterfactual inputs through a SPECTRA rationalizer with an **agreement regularization** strategy, described next.

4.5.1 Linking Counterfactuals and Rationales

We propose to incorporate counterfactuals into a model's functionality by taking advantage of the fully differentiable rationalization setup. Concretely, we decompose a rationalizer into two flows, as depicted in Figure 4.2: a **factual flow** that receives factual inputs $\boldsymbol{x}$ and outputs a factual prediction $\hat{y}$, and a **counterfactual flow** that receives counterfactual inputs $\tilde{\boldsymbol{x}}$ and should output a counterfactual prediction $\tilde{y} \neq \hat{y}$. As a by-product of using a rationalizer, we also obtain a factual rationale $\boldsymbol{z} \in \{0,1\}^n$ for $\boldsymbol{x}$ and a counterfactual rationale $\tilde{\boldsymbol{z}} \in \{0,1\}^m$ for $\tilde{\boldsymbol{x}}$, where $n = |\boldsymbol{x}|$ and $m = |\tilde{\boldsymbol{x}}|$.

Training. Let $\Theta = \{\phi, \gamma, \theta\}$ represent the trainable parameters of a rationalizer (defined in §4.2.1). We propose the following loss function:

$$\begin{aligned}\mathcal{L}(\Theta) = \mathcal{L}_f(y_f, \hat{y}(\Theta)) + \alpha\mathcal{L}_c(y_c, \tilde{y}(\Theta)) \\ + \lambda\Omega(\boldsymbol{z}(\Theta), \tilde{\boldsymbol{z}}(\Theta)),\end{aligned} \tag{4.3}$$

where $\mathcal{L}_f(\cdot)$ and $\mathcal{L}_c(\cdot)$ represent cross-entropy losses for the factual and counterfactual flows, respectively, and $\Omega(\cdot)$ is a novel penalty term to encourage factual and counterfactual rationales to focus on the same positions, as defined next. $\alpha \in \mathbb{R}$ and $\lambda \in \mathbb{R}$ are hyperparameters.

Agreement regularization. To produce paired rationales for both the factual and counterfactual flows, we incorporate regularization terms into the training of a rationalizer to encourage the factual explainer to produce rationales similar to those originally generated by the *masker* $\boldsymbol{z}^\star$, and the counterfactual explainer to produce rationales that focus on the tokens modified by the *editor* $\tilde{\boldsymbol{z}}^\star$. We derive the ground truth counterfactual rationale $\tilde{\boldsymbol{z}}^\star$ by aligning $\boldsymbol{x}$ to $\tilde{\boldsymbol{x}}$ and marking tokens that were inserted or substituted as 1, and others as 0. The regularization terms are defined as:

$$\Omega(\boldsymbol{z}, \tilde{\boldsymbol{z}}) = \|\boldsymbol{z}(\Theta) - \boldsymbol{z}^\star\|_2^2 + \|\tilde{\boldsymbol{z}}(\Theta) - \tilde{\boldsymbol{z}}^\star\|_2^2 . \tag{4.4}$$

To allow the counterfactual rationale $\tilde{\boldsymbol{z}}$ to focus on all important positions in the input, we adjust the budget for the counterfactual flow based on the length of the synthetic example produced by the counterfactual generator. Specifically, we multiply the budget by a factor of $\frac{\|\tilde{\boldsymbol{z}}^\star\|_0}{\|\boldsymbol{z}^\star\|_0}$.

Setup	IMDB	rIMDB	cIMDB	RotTom	SST-2	Amazon	Yelp
F	91.1 ± 0.3	91.4 ± 0.8	88.5 ± 0.9	76.5 ± 1.6	79.8 ± 1.6	86.0 ± 0.7	88.5 ± 0.7
With data augmentation:							
$F + C_H$	90.9 ± 0.5	**92.9** ± 0.9	**90.4** ± 1.6	76.6 ± 1.5	80.7 ± 1.3	86.3 ± 1.0	89.1 ± 1.2
$F + C_{S,V}$	91.0 ± 0.2	91.2 ± 1.0	89.3 ± 0.8	76.8 ± 0.9	79.3 ± 0.3	85.2 ± 0.9	88.0 ± 1.0
$F + C_S$	90.8 ± 0.2	91.6 ± 1.3	89.2 ± 0.4	76.7 ± 1.0	80.6 ± 0.6	86.4 ± 0.6	89.1 ± 0.5
With agreement regularization:							
F & $C_{S,V}$	90.7 ± 0.5	92.2 ± 0.7	88.9 ± 1.0	76.3 ± 1.4	80.2 ± 1.3	86.3 ± 0.7	88.9 ± 0.7
F & C_S	**91.2** ± 0.5	**92.9** ± 0.5	89.7 ± 1.1	**77.3** ± 2.3	**81.1** ± 2.4	**86.8** ± 0.8	**89.3** ± 0.7

Table 4.2: Accuracy of SPECTRA trained on IMDB and evaluated on in-domain, contrast, and out-of-domain datasets. We present mean and std. values across five random seeds. Values in **bold**: top results; underlined: second-best.

4.6 Exploiting Counterfactuals for Training

In this section, we evaluate the effects of incorporating CREST-generated counterfactuals into training by comparing a vanilla data augmentation approach with our CREST-Rationalization approach. We compare how each affects model robustness (§4.6.2) and interpretability (§4.6.3).

4.6.1 Experimental Setting

We use the IMDB and SNLI datasets to train SPECTRA rationalizers with and without counterfactual examples, and further evaluate on in-domain, contrast and out-of-domain (OOD) datasets. For IMDB, we evaluate on the revised IMDB, contrast IMDB, RottenTomatoes, SST-2, Amazon Polarity, and Yelp. For SNLI, we evaluate on the Hard SNLI, revised SNLI, break, MultiNLI, and Adversarial NLI. Dataset details can be found in §B.1. To produce CREST counterfactuals, which we refer to as "synthetic", we use a 30% masking budget as it provides a good balance between validity, fluency, and closeness (*cf.* Figure 4.3). We tune the counterfactual loss (α) and agreement regularization (λ) weights on the dev set. We report results with $\alpha = 0.01$ and $\lambda = 0.001$ for IMDB, and $\alpha = 0.01$ and $\lambda = 0.1$ for SNLI.

4.6.2 Robustness Results

Tables 4.2 and 4.3 show results for counterfactual data augmentation and agreement regularization for IMDB and SNLI, respectively. We compare a standard SPECTRA trained on factual examples (F) with other SPECTRA models trained on augmentated data from human-crafted counterfactuals ($F + C_H$) and synthetic counterfactuals generated by CREST ($F + C_S$), which we additionally post-process to drop invalid examples ($F + C_{S,V}$).

Discussion. As shown in Table 4.2, CREST-Rationalization (F & C_S) consistently outperforms vanilla counterfactual augmentation ($F + C_S$) on all sentiment classification datasets. It achieves the top results on the full IMDB and on all OOD datasets, while also leading to strong results on contrastive datasets—competitive with manual counterfactuals ($F + C_H$). When analyzing the performance of CREST-Rationalization trained on a subset of valid examples (F & $C_{S,V}$) versus the entire dataset (F & C_S), the models trained on the entire dataset maintain a higher level of performance across all datasets. However, when using counterfactuals for data augmentation, this trend is less pronounced, especially for in-domain and contrastive datasets. In §B.5,

Setup	SNLI	SNLI-h	rSNLI	break	MNLI-m	MNLI-mm	ANLI
F	86.6 ± 0.2	73.7 ± 0.2	71.1 ± 0.8	69.5 ± 1.5	**64.6** ± 1.1	65.9 ± 0.9	32.6 ± 0.7
With data augmentation:							
$F + C_H$	86.6 ± 0.3	74.9 ± 1.1	**72.4** ± 0.3	70.1 ± 1.9	64.2 ± 0.9	65.8 ± 0.9	31.8 ± 0.4
$F + C_{S,V}$	86.5 ± 0.3	**75.8** ± 1.2	71.8 ± 1.0	69.1 ± 2.0	64.4 ± 0.3	65.9 ± 0.4	32.2 ± 0.2
$F + C_S$	86.6 ± 0.3	74.7 ± 1.1	71.6 ± 0.8	**71.2** ± 1.4	64.5 ± 0.4	**66.4** ± 0.6	32.2 ± 1.0
With agreement regularization:							
F & $C_{S,V}$	**86.8** ± 0.1	75.3 ± 0.8	66.8 ± 0.7	68.2 ± 2.1	**64.6** ± 0.7	66.1 ± 0.6	**32.8** ± 0.6
F & C_S	86.6 ± 0.1	75.5 ± 1.3	67.0 ± 1.3	69.9 ± 1.7	64.2 ± 1.1	66.0 ± 0.7	32.5 ± 0.5

Table 4.3: Accuracy of SPECTRA trained on SNLI and evaluated on in-domain, contrast, and out-of-domain datasets. We present mean and std. values across five random seeds. Values in **bold**: top results; underlined: second-best.

we explore the impact of the number of augmented examples on results and find that, consistent with previous research (Huang et al., 2020; Joshi and He, 2022), augmenting the training set with a small portion of valid and diverse synthetic counterfactuals leads to more robust models, and can even outweigh the benefits of manual counterfactuals.

Examining the results for NLI in Table 4.3, we observe that both counterfactual augmentation and agreement regularization interchangeably yield top results across datasets. Remarkably, in contrast to sentiment classification, we achieve more substantial improvements with agreement regularization models when these are trained on valid counterfactuals, as opposed to the full set.

Overall, these observations imply that CREST-Rationalization is a viable alternative to data augmentation for improving model robustness, especially for learning contrastive behavior for sentiment classification. In the next section, we explore the advantages of CREST-Rationalization for improving model interpretability.

4.6.3 Interpretability Analysis

In our final experiments, we assess the benefits of our proposed regularization method on model interpretability. We evaluate effects on rationale quality along three dimensions: plausibility, forward simulability, and counterfactual simulability.

Plausibility. We use the MovieReviews (DeYoung et al., 2020) and the e-SNLI (Camburu et al., 2018) datasets to study the human-likeness of rationales by matching them with human-labeled explanations and measuring their AUC, which automatically accounts for multiple binarization levels.[5]

Forward simulability. Simulability measures how often a human agrees with a given classifier when presented with explanations, and many works propose different variants to compute simulability scores in an automatic way (Doshi-Velez and Kim, 2017; Treviso and Martins, 2020; Hase et al., 2020; Pruthi et al., 2022). Here, we adopt the framework proposed by Treviso and Martins (2020), which views explanations as a message between a classifier and a linear student model, and determines simulability as the fraction of examples for which the communication is successful. In our case, we cast a SPECTRA rationalizer as the classifier, use its rationales as explanations, and train a linear student on factual examples of the IMDB and SNLI datasets. High simulability scores indicate more understandable and informative explanations.

[5]We determine the explanation score for a single word by calculating the average of the scores of its word pieces.

	Sentiment Classification			Natural Language Inference		
Setup	**Plausibility**	**F. sim.**	**C. sim.**	**Plausibility**	**F. sim.**	**C. sim.**
F	0.6733 ± 0.02	91.70 ± 0.92	81.18 ± 2.79	0.7735 ± 0.00	59.26 ± 0.41	70.01 ± 0.44
With data augmentation:						
$F + C_H$	0.6718 ± 0.04	91.44 ± 1.46	80.53 ± 4.17	0.7736 ± 0.01	59.51 ± 0.86	69.90 ± 0.57
$F + C_S$	0.6758 ± 0.01	91.68 ± 0.59	84.54 ± 1.09	0.7779 ± 0.00	**59.54** ± 0.08	**70.76** ± 0.54
With agreement regularization:						
F & C_S	**0.6904** ± 0.02	**91.93** ± 0.83	**86.43** ± 1.56	**0.7808** ± 0.00	59.31 ± 0.20	70.69 ± 0.29

Table 4.4: Interpretability analysis of rationalizers trained with CREST-generated counterfactuals, either with data augmentation or agreement regularization. Plausibility represents matching with human rationales, whereas F. sim. and C. sim. represent forward and counterfactual simulability. **Bold**: top results; underlined: second-best.

Counterfactual simulability. Building on the manual simulability setup proposed by Doshi-Velez and Kim (2017), we introduce a new approach to automatically evaluate explanations that interact with counterfactuals. Formally, let C be a classifier that when given an input $\boldsymbol{x}$ produces a prediction $\hat{y}$ and a rationale $\boldsymbol{z}$. Moreover, let G be a pre-trained counterfactual editor, which receives $\boldsymbol{x}$ and $\boldsymbol{z}$ and produces a counterfactual $\tilde{\boldsymbol{x}}$ by infilling spans on positions masked according to $\boldsymbol{z}$ (e.g., via masking). We define *counterfactual simulability* as follows:

$$\frac{1}{N}\sum_{n=1}^{N}[[C(\boldsymbol{x}_n) \neq C(G(\boldsymbol{x}_n \odot \boldsymbol{z}_n))]], \tag{4.5}$$

where $[[\cdot]]$ is the Iverson bracket notation. Intuitively, counterfactual simulability measures the ability of a rationale to change the label predicted by the classifier when it receives a contrastive edit with infilled tokens by a counterfactual generator as input. Therefore, a high counterfactual simulability indicates that the rationale $\boldsymbol{z}$ focuses on the highly contrastive parts of the input.

Results. The results of our analysis are shown in Table 4.4. We observe that plausibility can substantially benefit from synthetic CREST-generated counterfactual examples, especially for a rationalizer trained with our agreement regularization, which outperforms other approaches by a large margin. Additionally, leveraging synthetic counterfactuals, either via data augmentation or agreement regularization, leads to a high forward simulability score, though by a smaller margin—within the standard deviation of other approaches. When looking at counterfactual simulability, we note that models that leverage CREST counterfactuals consistently lead to better rationales. In particular, agreement regularization leads to strong results on both tasks while also producing more plausible rationales, showing the efficacy of CREST-Rationalization in learning contrastive behavior.

4.7 Related Works

Generating counterfactuals. Existing approaches to generating counterfactuals for NLP use heuristics (Ren et al., 2019; Ribeiro et al., 2020), leverage plug-and-play approaches to controlled generation (Madaan et al., 2021), or, most relatedly, fine-tune language models to generate counterfactuals (Wu et al., 2021; Ross et al., 2021, 2022b; Robeer et al., 2021). For instance, PolyJuice (Wu et al., 2021) finetunes a GPT-2 model on human-crafted counterfactuals to generate counterfactuals following pre-defined control codes, while CounterfactualGAN (Robeer et al., 2021) adopts a GAN-like setup. We show that CREST-Generation outperforms both

methods in terms of counterfactual quality. Most closely related is MiCE (Ross et al., 2021), which also uses a two-stage approach based on a masker and an editor to generate counterfactuals. Unlike MiCE, we propose to relax the minimality constraint and generate masks using selective rationales rather than gradients, resulting not only in higher-quality counterfactuals, but also in a fully-differentiable set-up that allows for further optimization of the masker. Other recent work includes Tailor (Ross et al., 2022b), a semantically-controlled generation system that requires a human-in-the-loop to generate counterfactuals, as well as retrieval-based and prompting approaches such as RGF (Paranjape et al., 2022) and CORE (Dixit et al., 2022).

Training with counterfactuals. Existing approaches to training with counterfactuals predominantly leverage data augmentation. Priors works have explored how augmenting with both manual (Kaushik et al., 2020; Khashabi et al., 2020; Huang et al., 2020; Joshi and He, 2022) and automatically-generated (Wu et al., 2021; Ross et al., 2022b; Dixit et al., 2022) counterfactuals affects model robustness. Unlike these works, CREST-Rationalization introduces a new strategy for training with counterfactuals that leverages the paired structure of original and counterfactual examples, improving model robustness and interpretability compared to data augmentation. Also related is the training objective proposed by Gupta et al. (2021) to promote consistency across pairs of examples with shared substructures for neural module networks, and the loss term proposed by Teney et al. (2020) to model the factual-counterfactual paired structured via gradient supervision. In contrast, CREST can be used to *generate* paired examples, can be applied to non-modular tasks, and does not require second-order derivatives.

Rationalization. There have been many modifications to the rationalization setup to improve task accuracy and rationale quality. Some examples include conditioning the rationalization on pre-specified labels (Yu et al., 2019), using an information-bottleneck formulation to ensure informative rationales (Paranjape et al., 2020), training with human-created rationales (Lehman et al., 2019), and replacing stochastic variables with deterministic mappings (Guerreiro and Martins, 2021). We find that CREST-Rationalization, which is fully unsupervised, outperforms standard rationalizers in terms of model robustness and quality of rationales.

4.8 Conclusions and Future Works

We proposed CREST, a joint framework for selective rationalization and counterfactual text generation that is capable of producing valid, fluent, and diverse counterfactuals, while being flexible for controlling the amount of perturbations. We have shown that counterfactuals can be successfully incorporated into a rationalizer, either via counterfactual data augmentation or agreement regularization, to improve model robustness and rationale quality. Our results demonstrate that CREST successfully bridges the gap between selective rationales and counterfactual examples, addressing the limitations of existing methods and providing a more comprehensive view of a model's predictions.

One possibility to improve CREST is to replace its T5-based generator by more recent and larger language models, such as Flan-T5 (Chung et al., 2022; Longpre et al., 2023). Additionally, there are alternative methods to guide the generation of counterfactuals from highlight explanations, such as prompting language models with high-level or explicit information about which words should be perturbed.

5

Sparsefinder: Predicting Attention Sparsity in Transformers

Contents

A bottleneck in transformer architectures is their quadratic complexity with respect to the input sequence, which has motivated a body of work on efficient sparse *approximations* to softmax. An alternative path, used by α-entmax transformers, consists of having built-in *exact* sparse attention (Correia et al., 2019). As we also showed in Chapter 3, the selective nature of α-entmax brings interpretability benefits. However, computing sparse attention in entmax transformers still requires a quadratic cost.

In this chapter, we propose *Sparsefinder*, in which a simple student model is trained to identify, *a priori*, the sparsity pattern of entmax attention of a large teacher model—without actually computing it. Sparsefinder is designed to reduce computational cost while preserving the interpretable behavior of learned attention heads, which were shown to capture specialized linguistic knowledge (Voita et al., 2019; Correia et al., 2019; Raganato et al., 2020). This scheme resembles our previous simulability framework introduced in Chapter 3; however, here we teach Sparsefinder to simulate the attention patterns of the teacher, rather than its predictions.

We experiment with three variants of Sparsefinder, based on distances, quantization, and clustering, on two tasks: machine translation (attention in the decoder) and masked language modeling (encoder-only). Our work provides a new angle to study model efficiency by doing extensive analysis of the tradeoff between the sparsity and recall of the predicted attention graph. This allows for a detailed comparison between different models and may guide future benchmarks for sparse models.

This chapter is based on Treviso et al. (2022).

5.1 Motivation

Transformer-based architectures have achieved remarkable results in many NLP tasks (Vaswani et al., 2017; Devlin et al., 2019; Brown et al., 2020). However, they also bring important computational and environmental concerns, caused by their quadratic time and memory computation requirements with respect to the sequence length. This comes in addition to the difficulty of interpreting its inner workings, caused by their over-parameterization and large number of attention heads.

There is a large body of work developing ways to "sparsify" the computation in transformers, either by imposing local or fixed attention patterns (Child et al., 2019; Tay et al., 2020; Zaheer et al., 2020), by applying low-rank kernel approximations to softmax (Wang et al., 2020; Choromanski et al., 2021), or by learning which queries and keys should be grouped together (Kitaev et al., 2019; Roy et al., 2021). Most of the existing work seeks to *approximate* softmax-based attention by ignoring the (predicted) tails of the distribution, which can lead to performance degradation. An exception is transformers with **entmax-based sparse attention** (Correia et al., 2019), a content-based approach which is natively sparse; however, this approach still requires a quadratic computation to determine the sparsity pattern, failing to take computational advantage of attention sparsity.

In this work, we propose **Sparsefinder**, which fills the gap above by making entmax attention efficient (§5.4). Namely, we investigate three methods to predict the sparsity pattern of entmax without having to compute it: one based on metric learning, which is still quadratic but with a better constant (§5.4.3), one based on quantization (§5.4.4), and another one based on clustering (§5.4.5). In all cases, the predictors are trained offline on ground-truth sparse attention graphs from an entmax transformer, seeking high recall in their predicted edges without compromising the total amount of sparsity. Figure 5.1 illustrates our method.

To evaluate the effectiveness of our method across different scenarios, we perform experiments on two NLP

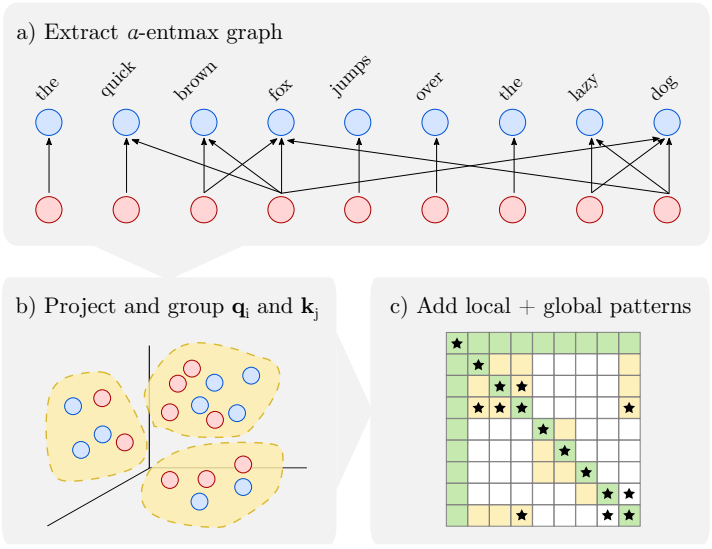


Figure 5.1: (a) Extract sparse attention graphs from a pretrained α-entmax transformer; (b) Project query and key vectors to a smaller and appropriated space such that similar points are likely to fall in the same vicinity; (c) Additionally, we can combine window and global patterns (green blocks) with the learned pattern (yellow blocks) to increase the recall in recovering ground-truth edges from the sparse graph at the top (starred blocks).

tasks, encompassing encoder-only and decoder-only configurations: machine translation (MT, §5.5) and masked language modeling (MLM, §5.6). We compare our method with four alternative solutions based on efficient transformers: Longformer (Beltagy et al., 2020), Bigbird (Zaheer et al., 2020), Reformer (Kitaev et al., 2020), and Routing Transformer (Roy et al., 2021), doing an extensive analysis of the trade-off between sparsity-recall and sparsity-accuracy. We complement these experiments by analyzing qualitatively what is selected by the different attention heads at the several layers and represented in different clusters/buckets.

Overall, our contributions are:[1]

- We propose a simple method that exploits learnable sparsity patterns to efficiently compute multi-head attention (§5.4).
- We do an extensive analysis of the trade-off between sparsity-recall and sparsity-accuracy in MT (§5.5) and MLM (§5.6), showing that there is a sweet spot that can be used to design efficient methods (§5.5,§5.6).
- We perform experiments on two tasks that span two disclosed scenarios of using transformers in NLP: MT (§5.5), and masked LM (§5.6).
- We analyze qualitatively what is selected by the different attention heads at the several layers and represented in different clusters/buckets.

5.2 Related Work

Interpreting multi-head attention. Several works analyze the functionalities learned by different attention heads, such as positional and local context patterns (Raganato and Tiedemann, 2018; Voita et al., 2019). Building upon prior work on sparse attention mechanisms (Peters et al., 2019), Correia et al. (2019) constrain the

[1]Our code is public available at: https://github.com/deep-spin/sparsefinder

attention heads to induce sparse selections individually for each head, bringing interpretability without post-hoc manipulation. Related approaches include the explicit sparse transformer (Zhao et al., 2019) and rectified linear attention (Zhang et al., 2021), which drops the normalization constraint. Raganato et al. (2020) show that it is possible to fix attention patterns based on previous known behavior (e.g. focusing on previous token) while improving translation quality. However, a procedure that exploits learnable sparsity patterns to accelerate multi-head attention is still missing.

Low-rank softmax approximations. Methods based on low-rank approximation to the softmax such as Linearized Attention (Katharopoulos et al., 2020), Linformer (Wang et al., 2020), and Performer (Choromanski et al., 2021) reduce both speed and memory complexity of the attention mechanism from quadratic to linear, but also hardens interpretability since its scores are not computed explicitly. On the other hand, methods that focus on defining or inducing sparse patterns provide interpretable alignments and also have performance gains in terms of speed and memory.

Fixed attention patterns. Among fixed pattern methods, Sparse Transformer (Child et al., 2019) and Longformer (Beltagy et al., 2020) attend to fixed positions by using strided/dilated sliding windows. BigBird uses random and two fixed patterns (global and window) to build a block sparse matrix representation (Zaheer et al., 2020), taking advantage of block matrix operations to accelerate computations in GPUs. In contrast, we replace the random pattern by a learned pattern that mimics pretrained α-entmax sparse attention graphs.

Learnable attention patterns. Learnable pattern methods usually have to deal with assignment decisions within the multi-head attention mechanism. Clustered Attention (Vyas et al., 2020) groups query tokens into clusters and compute dot-products only with centroids. Reformer (Kitaev et al., 2020) uses locality-sensitive hashing to efficiently group tokens in buckets. More similar to our work, Routing Transformer (Roy et al., 2021) clusters queries and keys with online k-means and compute dot-products over the top-k cluster points. Some queries and keys are discarded due to this filtering, which affects the overall recall of the method (as we show in §5.5 and §5.6).

5.3 Background

The main component of transformers is the multi-head attention mechanism (Vaswani et al., 2017), which responsible for contextualizing the information within and across input sentences (see §2.1.3 for a complete overview of the transformer architecture). However, in order to use the multi-head attention mechanism we need to compute the matrix multiplication $\boldsymbol{Q}\boldsymbol{K}^\top \in \mathbb{R}^{n\times m}$ used in the *scaled dot-product* operation in Eq. 2.6, which costs $\mathcal{O}(nmd)$ time and can be impractical when n and m are large. Many approaches, discussed in §5.2, approximate the attention matrix by ignoring entries far from the main diagonal or computing only some blocks of this matrix, with various heuristics. By doing so, the result will be an *approximation* of the softmax attention. This is because the original softmax-based attention is *dense*, i.e., it puts *some* probability mass on all tokens – not only a computational disadvantage, but also making interpretation harder, as it has been observed that only a small fraction of attention heads capture relevant information (Voita et al., 2019).

An alternative to softmax is the α-**entmax transformation** (Peters et al., 2019; Correia et al., 2019), which leads to sparse patterns directly, *without any approximation* (we describe α-entmax in detail in §2.1.2). In this work, we use $\alpha = 1.5$, which works well in practice and has a specialized fast algorithm (Peters et al., 2019). However, while sparse attention improves interpretability and head diversity when compared to dense alternatives (Correia et al., 2019), the learned sparsity patterns can not be trivially exploited to reduce the quadratic burden of self-attention, since we still need to compute dot-products between all queries and keys ($\boldsymbol{Q}\boldsymbol{K}^\top$) before applying the α-entmax transformation. In the next section (§5.4), we propose a simple method that learns to *identify* these sparsity patterns beforehand, avoiding the full matrix multiplication.

5.4 Sparsefinder

We now propose our method to extract sparse attention graphs and learn where to attend by exploiting a special property of α-entmax: *sparse consistency* (§5.4.1). We design three variants of Sparsefinder to that end, based on metric learning (§5.4.3), quantization (§5.4.4), and clustering (§5.4.5).

5.4.1 Attention graph and sparse consistency

For each attention head h, we define its **attention graph** as $\mathcal{G}_h = \{(\boldsymbol{q}_i, \boldsymbol{k}_j) \mid p_{i,j} > 0\}$, a bipartite graph connecting query and key pairs $\boldsymbol{q}_i, \boldsymbol{k}_j \in \mathbb{R}^d$ for which the α-entmax probability $p_{i,j}$ is nonzero. An example of attention graph is shown in Figure 5.1. We denote by $|\mathcal{G}_h|$ the total size of an attention graph, i.e., its number of edges. With α-entmax with $\alpha = 1.5$ we typically have $|\mathcal{G}_h| \ll nm$. In contrast, softmax attention always leads to a complete graph, $|\mathcal{G}_h| = nm$.

Problem statement. Our goal is to build a model – which we call *Sparsefinder* – that predicts $\hat{\mathcal{G}}_h \approx \mathcal{G}_h$ without having to perform all pairwise comparisons between queries and keys. This enables reducing the complexity of evaluating Eq. 2.6 from $\mathcal{O}(nmd)$ to $\mathcal{O}(|\hat{\mathcal{G}}_h|d)$, effectively taking advantage of the sparsity of α-entmax. In order to learn such model, we first extract a dataset of sparse attention graphs $\{\mathcal{G}_h\}$ from a pretrained entmax-based transformer (acting as a teacher). Then, the student learns where to pay attention based on this information. This procedure is motivated by the following **sparse-consistency** property of α-entmax:

Proposition 1 (Sparse-consistency property)**.** *Let $\boldsymbol{b}$ be a binary vector such that $b_j = 1$ if $p_j^\star > 0$, and $b_j = 0$ otherwise. For any binary mask vector $\boldsymbol{m}$ "dominated" by $\boldsymbol{b}$ (i.e. $\boldsymbol{m} \odot \boldsymbol{b} = \boldsymbol{b}$), we have*

$$\alpha\text{-entmax}(\boldsymbol{z}) = \alpha\text{-entmax}(\boldsymbol{z}|_{\boldsymbol{m}}), \tag{5.1}$$

where $z_j|_{\boldsymbol{m}} = z_j$ if $m_j = 1$ and $-\infty$ if $m_j = 0$.

Proof. From the definition of $\boldsymbol{z}|_{\boldsymbol{m}}$ and from Eq. 2.5, we have that

$$\begin{cases} z_j|_{\boldsymbol{m}} = z_j > \frac{\tau(\boldsymbol{z})}{\alpha-1} & \text{if } p_j^* > 0 \\ z_j|_{\boldsymbol{m}} \leq z_j \leq \frac{\tau(\boldsymbol{z})}{\alpha-1} & \text{if } p_j^* = 0. \end{cases} \tag{5.2}$$

We first prove that $\tau(\boldsymbol{z}|_{\boldsymbol{m}}) = \tau(\boldsymbol{z})$. From the definition of $\tau(\boldsymbol{z})$ we have that $\sum_j [(\alpha-1)z_j - \tau(\boldsymbol{z})]_+^{1/\alpha-1} = 1$. Plugging the (in)equalities from Eq. 5.2, we thus have

$$1 = \sum_j [(\alpha-1)z_j - \tau(\boldsymbol{z})]_+^{1/\alpha-1} = \sum_j [(\alpha-1)z_j|_{\boldsymbol{m}} - \tau(\boldsymbol{z})]_+^{1/\alpha-1}. \tag{5.3}$$

Since $\tau(\boldsymbol{z})$ satisfies the second equation – which is the condition that defines $\tau(\boldsymbol{z}|_{\boldsymbol{m}})$ – we thus conclude that $\tau(\boldsymbol{z}|_{\boldsymbol{m}}) = \tau(\boldsymbol{z})$. Combining the results in Eq. 5.2 and Eq. 5.3, we see that the supports of α-entmax$(\boldsymbol{z})$ and α-entmax$(\boldsymbol{z}|_{\boldsymbol{m}})$ are the same and so are the thresholds τ, and therefore from Eq. 2.5 we conclude that α-entmax$(\boldsymbol{z}|_{\boldsymbol{m}}) = \alpha$-entmax$(\boldsymbol{z})$. □

This property ensures that, if $\hat{\mathcal{G}}_h$ is such that $\mathcal{G}_h \subseteq \hat{\mathcal{G}}_h$, then we obtain *exactly* the same result as with the original entmax attention. Therefore, we are interested in having high recall,

$$\text{recall}(\hat{\mathcal{G}}_h; \mathcal{G}_h) = \frac{|\hat{\mathcal{G}}_h \cap \mathcal{G}_h|}{|\mathcal{G}_h|}, \tag{5.4}$$

meaning that our method is nearly exact, and high sparsity,

$$\text{sparsity}(\hat{\mathcal{G}}_h) = 1 - \frac{|\hat{\mathcal{G}}_h|}{nm}, \tag{5.5}$$

which indicates that computation can be made efficient.[2] Although a high sparsity may indicate that many computations can be ignored, converting this theoretical result into efficient computation is not necessarily trivial and potentially hardware-dependent. In this work, rather than proposing a practical computational efficient method, we focus on showing that such methods do exist and that they can be designed to outperform fixed and learned pattern methods while retaining a high amount of sparsity when compared to the ground-truth graph.

Our strategies. We learn the student model to predict $\hat{\mathcal{G}}_h \approx \mathcal{G}_h$ by taking inspiration from the Reformer model (Kitaev et al., 2020) and from the Routing Transformer (Roy et al., 2021). Formally, we define a set of B buckets, $\mathcal{B} = \{1, \ldots, B\}$, and learn functions $f_q, f_k : \mathbb{R}^d \to 2^{\mathcal{B}} \setminus \{\varnothing\}$, which assign a query or a key to one or more buckets. We will discuss in the sequel different design strategies for the functions f_q, f_k. Given these functions, the predicted graph is:

$$\hat{\mathcal{G}}_h = \{(\boldsymbol{q}_i, \boldsymbol{k}_j) \mid f_q(\boldsymbol{q}_i) \cap f_k(\boldsymbol{k}_j) \neq \varnothing\}, \tag{5.6}$$

that is, an edge is predicted between $\boldsymbol{q}_i$ and $\boldsymbol{k}_j$ iff they are together in some bucket.

We present three strategies, one based on distance-based pairing (§5.4.3), one based on quantization (§5.4.4) and another one on clustering (§5.4.5). All strategies require as a first step learning a metric that embeds the graph (projecting queries and keys) into a lower-dimensional space $\mathbb{R}^r$ with $r \ll d$, such that positive query-key pairs are close to each other, and negative pairs are far apart.

5.4.2 Learning projections

According to the α-entmax sparse-consistency property, in order to get a good approximation of $\mathcal{G}_h$, we would like that f_q and f_k produce a graph $\hat{\mathcal{G}}_h$ that maximizes recall, defined in Equation 5.4. However, maximizing recall in this setting is difficult since we do not have ground-truth bucket assignments. Instead, we recur to a contrastive learning approach by learning projections via negative sampling, which is a simpler and more scalable than constrained clustering approaches (Wagstaff et al., 2001; de Amorim, 2012).

For each head, we start by projecting the original query and key $\boldsymbol{q}, \boldsymbol{k} \in \mathbb{R}^d$ vectors into lower dimensional vectors $\boldsymbol{q}', \boldsymbol{k}' \in \mathbb{R}^r$ such that $r \ll d$. In practice, we use a simple head-wise linear projection for all queries and

[2] For the decoder self-attention the denominator in Equation 5.5 becomes $n(n+1)/2$ due to "causal" masking.

keys $g_\theta : \mathbb{R}^d \to \mathbb{R}^r$. To learn the parameters of the projection layer we minimize a hinge loss with margin ω for each head h:

$$\mathcal{L}_\theta(\mathcal{G}_h) = \left[\omega + \|\boldsymbol{q}' - \boldsymbol{k}'_{\mathrm{P}}\|_2^2 - \|\boldsymbol{q}' - \boldsymbol{k}'_{\mathrm{N}}\|_2^2\right]_+, \tag{5.7}$$

where $(\boldsymbol{q}', \boldsymbol{k}'_{\mathrm{P}}) \in \mathcal{G}_h$ is a positive pair, $(\boldsymbol{q}', \boldsymbol{k}'_{\mathrm{N}}) \notin \mathcal{G}_h$ is a negative pair sampled uniformly at random, and $[\cdot]_+$ is the positive part (ReLU) function. In words, we want the distance between a query vector to negative pairs to be larger than the distance to positive pairs by a margin ω. This approach can also be seen as a weakly-supervised learning problem, where the goal is to push dissimilar points away while keeping similar points close to each other (Xing et al., 2002; Weinberger and Saul, 2009; Bellet et al., 2015).

5.4.3 Distance-based pairing

To take advantage of the proximity of data points on the embedded space, we first propose a simple method to connect query and key pairs whose Euclidean distance is less than a threshold t, i.e. $\hat{\mathcal{G}}_h = \{(\boldsymbol{q}_i, \boldsymbol{k}_j) \mid \|\boldsymbol{q}'_i - \boldsymbol{k}'_j\|_2 \leq t\}$. Although this method also requires $O(n^2)$ computations, it is more efficient than a vanilla transformer since it reduces computations by a factor of d/r by using the learned projections. This method is also useful to probe the quality of the embedded space learned by the projections, since the performance of our other methods will be contingent on it.

5.4.4 Buckets through quantization

Our second strategy quantizes each dimension $1, \ldots, r$ of the lower-dimensional space into β bins, placing the queries and keys into the corresponding buckets ($B = r\beta$ buckets in total). This way, each $\boldsymbol{q}_i$ and $\boldsymbol{k}_j$ will be placed in exactly r buckets (one per dimension). If $\boldsymbol{q}_i$ and $\boldsymbol{k}_j$ are together in some bucket, Sparsefinder predicts that $(\boldsymbol{q}_i, \boldsymbol{k}_j) \in \hat{\mathcal{G}}_h$. Note that for this quantization strategy no learning is needed, only the hyperparameter β and the binning strategy need to be chosen. We propose a fixed-size binning strategy: divide each dimension into β bins such that all bins have exactly $\lceil n/\beta \rceil$ elements. In practice, we append padding symbols to the input to ensure that bins are balanced.

5.4.5 Buckets through clustering

The clustering strategy uses the low-dimensional projections and runs a clustering algorithm to assign $\boldsymbol{q}_i$ and $\boldsymbol{k}_j$ to one or more clusters. In this case, each cluster corresponds to a bucket. In our work, we employed k-means to learn B centroids $\{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_B\}$, where each $c_b \in \mathbb{R}^r$, over a small portion of the training set. This strategy is similar to the Routing Transformer's online k-means (Roy et al., 2021), but with two key differences: (a) our clustering step is applied offline; (b) we assign points to the top-k closest centroids rather than assigning the closest top-k closest points to each centroid, ensuring that all queries are assigned to a cluster.[3] At test time, we use the learned centroids to group queries and keys into k clusters each:

$$f_q(\boldsymbol{q}_i) = \underset{1 \leq b \leq B}{\arg\,\text{top-}k} -\|\boldsymbol{q}_i - \boldsymbol{c}_b\|_2^2, \tag{5.8}$$

$$f_k(\boldsymbol{k}_j) = \underset{1 \leq b \leq B}{\arg\,\text{top-}k} -\|\boldsymbol{k}_j - \boldsymbol{c}_b\|_2^2, \tag{5.9}$$

[3]The difference relies on the dimension on which the top-k operation is applied. Routing Transformer applies top-k on the input dimension, possibly leaving some queries unattended, whereas Sparsefinder applies on the centroids dimension, avoiding this problem.

where the arg top-k operator returns the indices of the k^{th} largest elements. As in the quantization-based approach, queries and keys will attend to each other, i.e., Sparsefinder predicts $(\boldsymbol{q}_i, \boldsymbol{k}_j) \in \hat{\mathcal{G}}_h$ if they share at least one cluster among the k closest ones. Smaller values of k will induce high sparsity graphs, whereas a larger k is likely to produce a more dense graph but with a higher recall.

5.4.6 Computational cost

Let L be the maximum number of elements in a bucket. The time and memory cost of bucketed attention computed through quantization or clustering is $\mathcal{O}(BL^2)$. With balanced buckets, we get a complexity of $\mathcal{O}(n\sqrt{n})$ (Kitaev et al., 2020). Although this cost is sub-quadratic, leveraging the sparse structure of $\hat{\mathcal{G}}_h$ in practice is challenging, since it might require specialized hardware or kernels. In general, we have $|\hat{\mathcal{G}}_h| = \sum_{b=1}^{B} n_b m_b \ll nm$, where n_b and m_b are the number of queries and keys in each bucket, since we have small complete bipartite graphs on each bucket. Instead of viewing quadratic methods only in light of their performance, we adopt an alternative view of assessing the tradeoff of these methods in terms of sparsity and recall of their approximation $\hat{\mathcal{G}}_h$. This offers a theoretical perspective to the potential performance of each approximation on downstream tasks, helping to find the best approximations for a desired level of sparsity.

5.4.7 Combining learned and fixed patterns

As pointed out in prior work (Voita et al., 2019), several attention heads rely strongly in local patterns or prefer to attend to a particular position, more promimently in initial layers. Therefore, we take inspiration from the Longformer (Beltagy et al., 2020) and BigBird (Zaheer et al., 2020) and combine learned sparse patterns with window and global patterns by adding connections in the predicted graph $\hat{\mathcal{G}}_h$ to improve the recall of all methods. Figure 5.1 illustrates how these patterns are combined in the last step. The inclusion of these patterns offer a complementary analysis to the pattern learned by bucketing-based methods.

5.5 Experiments: Machine Translation

Setup. We pretrain a *transformer-large* model (6 layers, 12 heads) on the Paracrawl dataset (Esplà et al., 2019). Next, we finetune it with α-entmax, fixing $\alpha = 1.5$ for all heads, on EN→DE and EN→FR language pairs from IWSLT17 (Cettolo et al., 2017). We use the 2011-2014 sets as validation data and the 2015 set as test data. We encode each word using byte pair encoding (BPE, Sennrich et al. 2016) with a joint segmentation of 32k merges. As Vaswani et al. (2017), we finetune our models using the Adam optimizer with an inverse square root learning rate scheduler, with an initial value of 5×10^{-4} and a linear warm-up in the first 4000 steps. We evaluate translation quality with sacreBLEU (Post, 2018). Training details, hyperparameters, and data statistics are described in §C.1.1.

Learning projections. To learn projections for queries and keys (§5.4.2), we randomly selected 10K long instances ($n > 20$ tokens) from the training set and extracted the α-entmax attention graphs $\mathcal{G}_h$ from the decoder self-attention for each head. This led to an average of 8M and 9M positive pairs $(\boldsymbol{q}_i, \boldsymbol{k}_j)$ per layer for EN→DE and EN→FR, respectively. We use 10% of this set as validation data. In practice, due to the small

number of parameters for each head (only 4,160), a single epoch with Adam was sufficient to optimize the loss in Eq. 5.7. Hyperparameters and training details are set in §C.1.2.

Qualitative analysis. Using this subset of 10K samples, we investigate the sparsity-recall tradeoff of the clustering and quantization variants as the number of buckets varies, comparing them with Longformer, BigBird, Reformer, Routing Transformer, and a simple window baseline, which connects query and key pairs within a sliding window. To better see what each layer and head captures, we show examples for selected layers and heads in Figure 5.2 varying the number of buckets $B \in \{2, 4, 6, 8, 10, 12\}$ for bucket-based methods, the threshold $t \in \{0.5, 1.0, 1.5, 2.0, 2.5\}$ for the distance-based method, and the window size within $\{0, 1, 3, 5, 7, 9, 11, 15, 19, 23, 27\}$ for the window baseline. The tradeoff curves for all heads and layers can be consulted in §C.1.2. We note that heads and layers exhibit specialized behavior, confirming the findings of

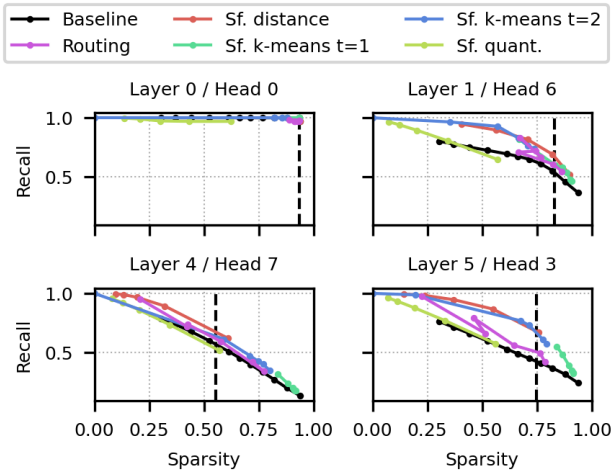


Figure 5.2: Examples of the sparsity-recall tradeoff on MT for different layers and heads. Shown are several variants of Sparsefinder, the Routing Transformer, and a window-level baseline. The vertical line indicates the gold sparsity obtained by the full entmax transformer.

Voita et al. (2019) and Correia et al. (2019). For instance, the first layer seems to focus mostly on local content, since they tend to have high sparsity and the window baseline performs well, whereas the attention of last layers are more spread. The overall findings from this analysis suggest that distance and cluster-based methods outperform quantization-based methods, and that among the cluster-based methods, Sparsefinder with top-$k \in \{1, 2\}$ lead to better sparsity-recall tradeoffs than Routing Transformer, with $k = 2$. Moreover, all methods achieve a recall higher than a simple window baseline, without compromising sparsity too much when the number of buckets is within $B \in \{4, 6, 8\}$, with 6 being the best choice overall. For this reason, we kept top-$k = 2$, $B = 6$, and $t = 2.0$ for the following experiments.

Sparsity-recall tradeoff. With a good choice of B and t on hand, we now turn to the full IWSLT dataset for both language pairs. We measure the performance gap as a function of the approximation to the ground-truth α-entmax attention graph $\mathcal{G}_h$ by replacing it by $\hat{\mathcal{G}}_h$ at test time. Moreover, we now add global and local patterns to all methods, varying the window size within $\{0, 1, 3, 5, 7, 9, 11, 15, 19, 23, 27\}$ to get different levels of sparsity/recall. We compare all variants of Sparsefinder (distance-based, quantization, k-means) with fixed

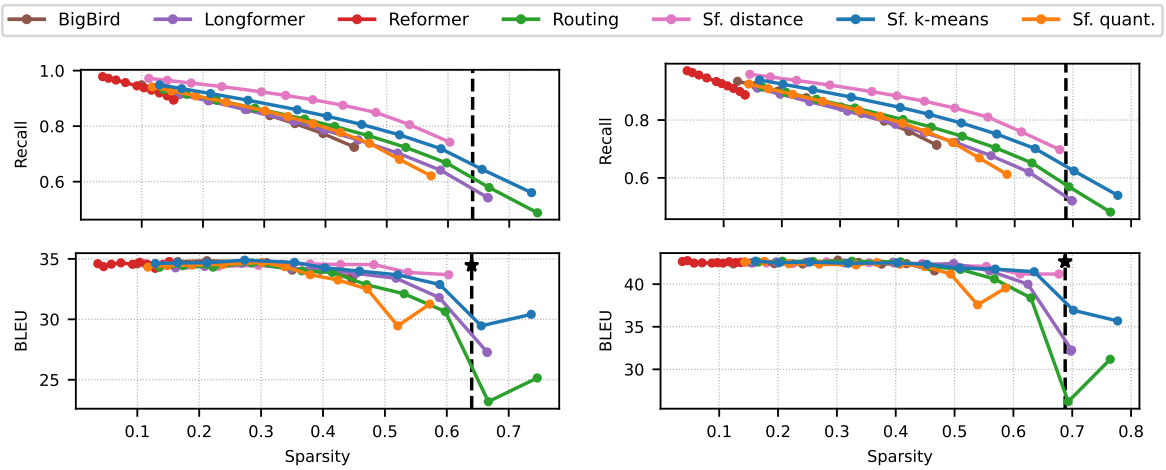


Figure 5.3: Sparsity-recall (top) and sparsity-BLEU (bottom) tradeoff averaged across all layers and heads on IWSLT EN→DE (left) and EN→FR (right). The vertical dashed line represents the gold sparsity obtained by the full α-entmax transformer, and the starred marks depict its BLEU score: 34.47 on EN→DE and 42.65 on EN→FR.

and learnable pattern methods: BigBird with 6 random blocks of size of 1; Longformer with 6 random global tokens; Reformer with $B = 6$ buckets; and Routing transformer with $B = 6$ clusters and top-k set to $\lceil n/B \rceil$ to have balanced clusters.

Plots for the sparsity-recall tradeoff by varying the window size are shown in the top of Figure 5.3 for both language pairs. Overall, both language pairs have similar trends for all methods. As the window size increases, we get higher recall but lower sparsity. At the far right, we can see that all methods have a moderate recall when no window is used, indicating that locality plays an important role in transformers. This is reinforced by the good performance of Longformer and Bigbird, which have a dominant window pattern. The distance-based method Pareto-dominates the other methods, followed by Sparsefinder k-means and Routing Transformer. Since the LSH attention in Reformer concatenates queries and keys before hashing, the resultant buckets are very similar to each other, therefore they tend to induce a graph with very high recall and very low sparsity.

Sparsity-accuracy tradeoff. We show the tradeoff between sparsity and BLEU in the bottom of Figure 5.3. For lower levels of sparsity, all methods perform well, close to the full entmax transformer. But as sparsity increases, indicating that only a few computations are necessary, we see that the distance-based and the k-means variants of Sparsefinder perform better than other methods, keeping a very high BLEU without abdicating sparsity. Moreover, the distance-method performs on par with the full entmax transformer even on the absence of a fixed window pattern, i.e., when the window size is zero — the far right point on the curve. The k-means variant lags behind the distance-based method for high sparsity scenarios, but as soon as we add a window with size of 3, it recovers a high BLEU. Overall, these plots show that methods with a high recall for higher levels of sparsity also tend to have a higher BLEU score.

Learned patterns. We select some heads and show in Figure 5.4 examples of the pattern learned by our k-means variant by the decoder self-attention on EN→FR. More examples can be found in §C.3. We note that the window pattern is useful to recover local connections. We can see that the k-means variant groups more query and key pairs than the actual number of ground-truth edges (left and middle plots). However, due to the

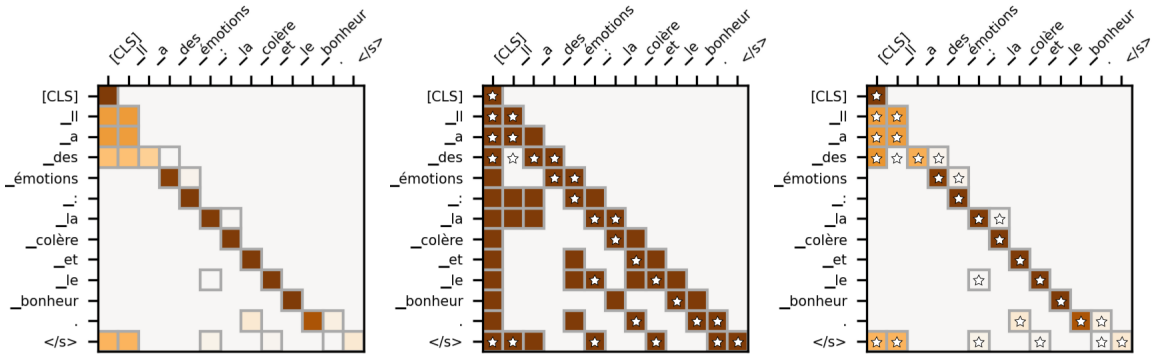


Figure 5.4: Ground-truth (left), learned patterns by Sparsefinder k-means (middle), and the subsequent attention weights (right). Starred blocks represent ground-truth edges.

sparse-consistency property (right plot), we can see that most of these predictions receive zero probability by α-entmax. Furthermore, in contrast to methods based on the approximation to the softmax such as Linformer and Performer (Wang et al., 2020; Choromanski et al., 2021), the clear pattern exhibited by our method allows the inspection of attention probabilities, enhacing interpretability.

5.6 Experiments: Masked LM

Setup. Following Beltagy et al. (2020), we initialize our model from a pretrained RoBERTa checkpoint. We use the `roberta-base` model from Huggingface's transformers library, with 12 layers and 12 heads.[4] We finetune on WikiText 103 (Merity et al., 2017), replacing softmax by α-entmax with $\alpha = 1.5$ for all heads. Training details, model hyperparameters, and data statistics are set in §C.2.

Learning projections. Like before, we learn to project keys and queries from the original 64 dimensions into $r = 4$ dimensions. For this we use 1K random samples from the training set, each with length of 512, keeping half for validation. Similarly to §5.5, we extract the α-entmax attention graphs $\mathcal{G}_h$ but from the encoder self-attention of each head, leading to an average of 3M positive pairs per layer. We set the number of buckets to 8 for all cluster and quatization-based methods, 8 random blocks/tokens for BigBird/Longformer, $t = 1.0$ for the distance-based method, and top-$k = 2$ for Sparsefinder k-means, by inspection on the validation set.

Results. Our full transformer trained with α-entmax achieved a perplexity score of 1.2529 with an overall sparsity of 0.9804 on WikiText 103. As in sentence-level MT experiments, we measure the sparsity-recall tradeoff and the performance gap via the change of $\mathcal{G}_h$ by $\hat{\mathcal{G}}_h$ at test time. To get different levels of sparsity we vary the window size within $\{31, 41, 51, 75, 101, 125, 151, 175, 201, 251\}$.

Results in terms of perplexity (or log-likelihood) as sparsity increases (window size changes) for each method are shown in Figure 5.5. The curves for the sparsity-recall tradeoff are similar to the ones found in MT experiments, with the distance-based method outperforming all methods, followed by the k-means variant of Sparsefinder. Moreover, we can still achieve very high recall and low perplexity with all methods by sacrificing some sparsity. We can see that our distance and clustering variants present the best Pareto curves in terms of

[4] https://huggingface.co/roberta-base

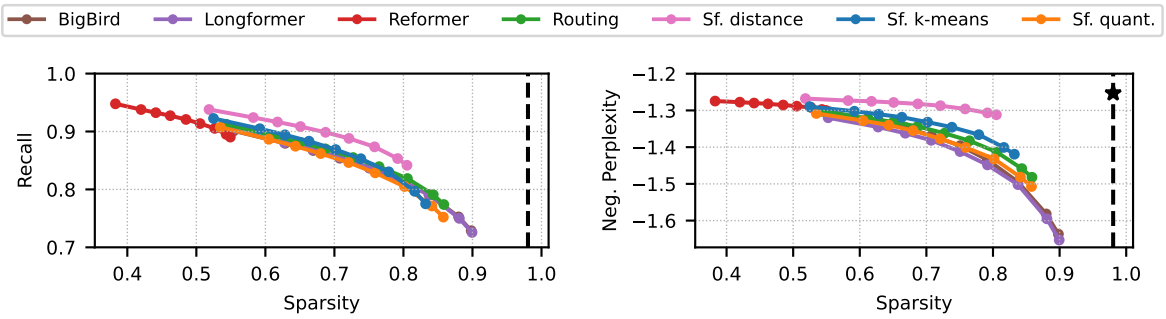


Figure 5.5: Sparsity-recall and sparsity-(neg-)perplexity tradeoff averaged across all layers and heads on WikiText 103. The vertical dashed line represents the gold sparsity obtained by the full α-entmax transformer.

perplexity, followed by Routing Transformer. The overall aspect of these plots suggest that local patterns have a great impact on performance, and that it is possible to design powerful sub-quadratic transformers without compromising sparsity too much. Moreover, although the distance-based method requires a quadratic number of computations, it reduces them by a factor of $d/r = 64/4 = 16$, as described in §5.4.3, and achieves better recall and accuracy than any other tested method, indicating an open area to design even more effective and efficient algorithms.

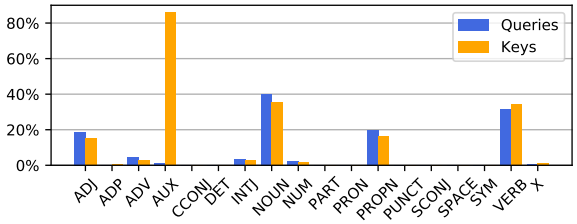


Figure 5.6: Tokens assigned to a given cluster, for the entire Wikitext 103 validation set. In the following sample we show which tokens are ***keys*** and *queries* for this cluster: *There **have been** been a large **number** of examples published where the requisite cation **is** arrived at by a **variety** of **rearrangements***

Analysis. To understand what is represented in each cluster, we run the following experiment: we obtain POS tags using spaCy, and calculate the distribution of each tag over clusters for all heads. We show an example in Figure 5.6 which focuses on auxiliary verbs. Here the cluster learned to group certain words (including verbs and nouns) which can attend to words of those same classes, and additionally attend to most auxiliary verbs.

Learned patterns. In Figure 5.7 we show Sparsefinder k-means' predicted attention graphs for a specific attention head that originally learned to focus on coreference tokens. We can see that the pattern induced by Sparsefinder keeps the behavior of attending to coreference tokens. In particular, this attention head achieves a high recall score ($\sim 80\%$) with a high sparsity rate ($\sim 75\%$).

5.6.1 Efficient Sparsefinder

We now turn to the question of making Sparsefinder efficient in practice. Before we proceed, we note that comparison between methods usually depends on the specific implementation used, which influences the measurements and can also require specialized hardware. This leaves BigBird and Routing Transformer as the

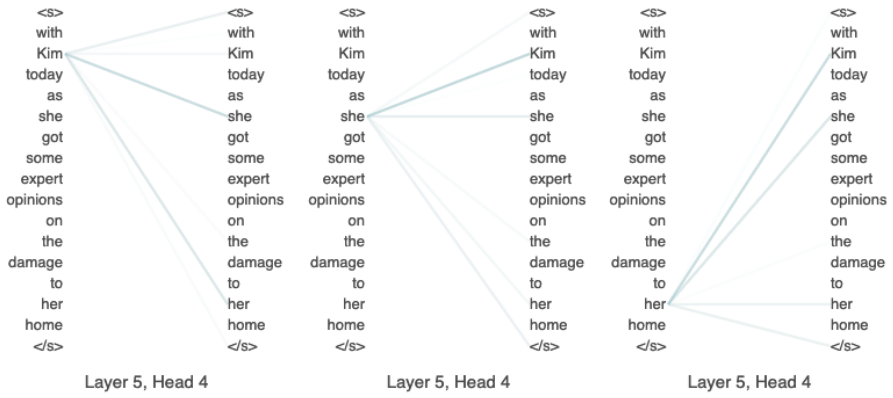


Figure 5.7: Learned pattern by Sparsefinder k-means of an attention head that focus on coreferences. Sparsefinder k-means achieves a recall of $\sim 80\%$ with a sparsity rate of $\sim 75\%$ on this attention head.

only models we can compare with in practice: Reformer includes other optimizations that are not part of the attention mechanism, and Longformer is based on CUDA kernels, specialized for fast computation. Lastly, the strategy used in Routing Transformer is incorporated in Sparsefinder (v2), where we use Sparsefinder's centroids with Routing Transformer top-k strategy. In order to make Sparsefinder more efficient, we adopt the key strategy of BigBird: work with contiguous chunks rather than single tokens, creating blocks in the attention matrix. More precisely, we learn projections over chunked tokens following Equation 5.7, where $(\boldsymbol{q}', \boldsymbol{k}'_P)$ is a positive pair if any token inside the chunk is part of a positive pair of the original α-entmax graph, and similarly, a pair $(\boldsymbol{q}', \boldsymbol{k}'_N)$ is negative if all tokens inside the chunk are negative. Thus, given a block/chunk size z, the size of the dense attention graph reduces from $|\mathcal{G}_h| = nm$ to $|\mathcal{G}_h| = \lceil nm/z^2 \rceil$ (with zero-padding).

Implementation. In order to be comparable to BigBird, we implement a routine that caps the maximum number of attended blocks in Sparsefinder, analogous to the number of random blocks used in BigBird. We propose two variants: **(v1)** computes dot-products between all chunked vector projections and then returns the top-k blocks, and **(v2)** selects the top-k blocks closest to the learned centroids and computes dot-products for these blocks. The first variant is more costly, yet it may lead to a more robust selection, whereas the second variant resembles Routing Transformer's top-k strategy.

Results. We measure the clock-time of the MLM model evaluated on 500 examples with a batch size of 8. We vary the number of attended blocks within $\{2, 3, 4, 8, 16, 22 \approx \sqrt{n}\}$, the block size in $\{2, 4, 8, 16\}$, and compute perplexity for values of B (number of clusters) within $\{2, 4, 8, 12, 16, 20\}$. We use a window size of 3 in all experiments to capture the controlled hyperparameters' impact better. Figure 5.8 shows plots by averaging runs with different block sizes and number of clusters. As expected, using a lower number of attended blocks leads to improvements in terms of running time, yet all models perform poorly on the MLM task. As we increase the number of blocks, we can see both a boost in terms of MLM performance and an increased running time. By comparing Sparsefinder and BigBird, we notice that BigBird is faster than Sparsefinder, but increasing the number of attended (random) blocks in BigBird does not lead to significant improvements on the real task. In contrast, both versions of Sparsefinder can improve the MLM performance while still being faster than a regular α-entmax transformer. In particular, by attending to only 2 blocks, Sparsefinder is able to achieve a better MLM

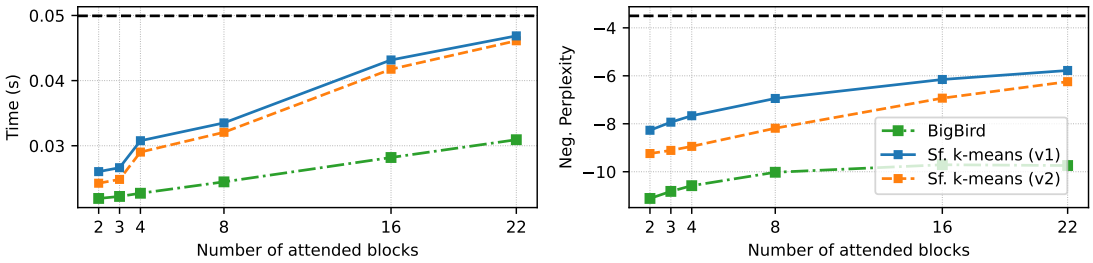


Figure 5.8: Comparison of Sparsefinder and BigBird in terms of running time and (negative) perplexity by varying the number of attended blocks. The black dashed line represents the results obtained by the full α-entmax transformer.

score than BigBird with 22 random blocks while still being faster than it. Plots for each block size can be found in §C.4.

5.7 Conclusions and Subsequent Works

We proposed Sparsefinder, a method to identify the sparsity pattern of entmax-based transformers while avoiding full computation of the score matrix. Our method learns a low-dimensional projection of queries and keys with a contrastive objective, and comes with three variants: distance, quantization, and clustering-based. We compared these variants against competing approaches on two tasks: machine translation and masked language modeling. We obtained favorable sparsity-recall and sparsity-accuracy tradeoff curves, and provided evidence that optimized attention heads are remained amenable to human interpretation.

Since our theoretical sparsity estimation provides a lower bound for how much computational sparsity can be achieved, we believe that our extensive analysis may guide future research on efficient transformers as hardware keeps evolving. Finally, Sparsefinder stands out from other approaches by seeking a balance between theoretical sparsity and approximation quality while preserving interpretability. As we will explore further in Chapters 7 and 8, we believe that our proposed approach to treat each attention head independently has potential applications to improve the quality of explanations extracted from them.

Part II

Explainable Machine Translation Quality Estimation

6

An Empirical Comparison of Explainability Methods for Quality Estimation

Contents

In the second part of this thesis, we investigate and propose new ways to interpret decisions made by quality estimation (QE) models. Our research started in 2019, in a collaboration with the Unbabel AI team to design accurate QE models for the WMT19 Shared Task on Quality Estimation (Kepler et al., 2019a), where we pioneered the use of pretrained transformers for QE, which led us to winning the shared task at the time. Our winning approaches have since been incorporated into the open-source framework OpenKiwi (Kepler et al., 2019b), which implements the best QE systems from recent WMT shared tasks.

Motivated by the work of Fomicheva et al. (2022a) on bridging explainability with word-level QE, we turned our focus towards extracting explanations from modern QE models based on pretrained transformers. Our first contributions to this end are detailed in this chapter, where we collaborated with the Unbabel AI team to participate in the Explainable Quality Estimation Shared Task (Fomicheva et al., 2021). The systems submitted to the shared task were divided into two tracks: constrained (without word-level supervision) and unconstrained (with word-level supervision). Initially, we explored multiple explainability methods, including gradient, erasure, and rationalization approaches (Lei et al., 2016; Bastings et al., 2019), as well as averaging attention from different layers, as suggested in (Fomicheva et al., 2022a), to extract the relevance of input tokens from sentence-level QE models built on top of multilingual pretrained transformers.

Subsequently, our finding from the previous chapter that distinct attention heads learn specialized linguistic phenomena motivated us to explore the role of single attention heads, which proved to be beneficial for QE. We further improve our explanations by scaling attention weights by the norm of values vectors, a strategy that was shown to improve plausibility at the time (Kobayashi et al., 2020). By ensembling explanation scores extracted from models trained with different pretrained transformers, we produced winning submissions for the constrained track for almost all language pairs, and achieved strong results for the unconstrained track without using synthetic data for word-level supervision.

This chapter is based on Treviso et al. (2021).

6.1 Motivation

Recent advances in QE have led to consistent improvements at predicting quality assessments such as *Direct Assessments* (DAs, Graham et al. 2013). Traditional QE systems had to predict *Human Translation Error Rate* (HTER, Snover et al. 2006), yet with the advent of neural machine translation, we observed a shift from fluency into adequacy errors (Martindale and Carpuat, 2018). For that reason, DAs started getting used as the ground-truth score for assessing the quality of translations (Specia et al., 2020). However, with DAs we lose the ability to generate word-level supervision, impacting the interpretability of sentence-level predictions in terms of lower granularity elements such as word-level translation errors.

At the same time, prominent QE systems such as OpenKiwi (Kepler et al., 2019b) and TransQuest (Ranasinghe et al., 2020) build on top of multilingual pretrained models such as BERT (Devlin et al., 2019) and XLM-RoBERTa (Conneau et al., 2020), which are largely responsible for the performance boost we have observed in recent editions of the WMT QE shared task (Fonseca et al., 2019; Specia et al., 2020). Due to the usage of such over-parameterized black-box models, this performance boost also comes at the cost of efficiency and interpretability.

Research in explainable NLP uncovered several strategies to interpret models' decisions, either in a post-

hoc manner by querying a trained model for extracting perturbation or gradient measures (Ribeiro et al., 2016; Arras et al., 2016), or by building models that are inherently interpretable (Lei et al., 2016; Chang et al., 2020). Recent works have also put transformers under the lens of explainability, aiming at unraveling interpretable patterns that clarify how decisions emerge from attention heads and across hidden states at each layer (De Cao et al., 2020; Abnar and Zuidema, 2020; Voita et al., 2021).

In this shared task, we experiment with several of these methods to extract the relevance of input tokens from sentence-level QE models built on top of multilingual pretrained transformers.[1] For the constrained track, where models are unaware of word-level supervision, our best results were derived from attention-based explanations. When we used word-level labels during training, the best results were obtained by using word-level predicted probabilities. Furthermore, we were able to push the performance further by ensembling explanations for both tracks.

6.2 Background

Quality Estimation. As noted in §2.2, QE systems are usually designed according to the granularity in which predictions are made. Sentence-level QE aims at predicting the quality of the whole translated sentence, either in terms of how many edit operations are required to fix it (HTER) or in terms of human judgments (DA). The goal of word-level QE is to assign quality labels (OK or BAD) to each *machine-translated word*, indicating whether that word is a translation error or not. Additionally, current systems also classify *source words* to denote words in the original sentence that have been mistranslated or omitted in the target.

Transformers. The multi-head attention mechanism is the bedrock on which transformers are built, being responsible for contextualizing information within the input dynamically (Vaswani et al., 2017). We describe the multi-head attention mechanism in detail in §2.1.3.

Explainability in NLP. There is a large body of work on the analysis and interpretation of models in NLP. Some of these models are built on top of attention mechanisms, which automatically learn a weighted representation of input features. Attention weights provide plausible, but not always faithful, explanations (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019). In contrast, rationalizers with hard attention are arguably more faithful but require stochastic networks (Lei et al., 2016; Bastings et al., 2019), with recent works avoiding stochasticity via sparse deterministic selections (Treviso and Martins, 2020; Guerreiro and Martins, 2021). Other approaches seek local explanations by considering gradient measures (Arras et al., 2016; Bastings and Filippova, 2020), or by perturbing the input and querying the classifier in a post-hoc manner (Ribeiro et al., 2016; Kim et al., 2020). Since transformers are composed of several layers and attention heads, many works analyze and improve the multi-head attention mechanism directly to produce better explanations (Kobayashi et al., 2020; Hao et al., 2021). More elaborated methods consider the entire flow of information coming from attention weights, hidden states, or gradients to interpret the model's decision (De Cao et al., 2020; Abnar and Zuidema, 2020; Voita et al., 2021).

[1] Our code can be found at: https://github.com/deep-spin/explainable_qe_shared_task/.

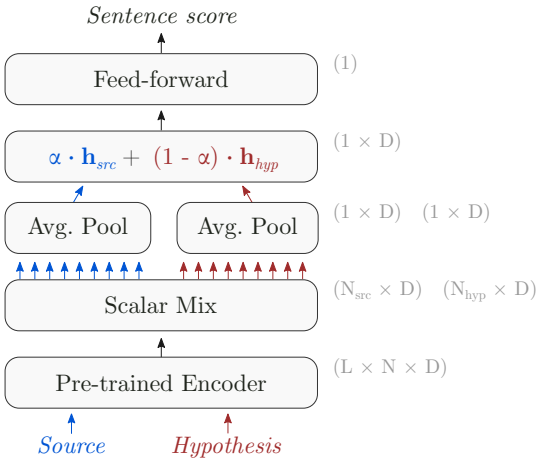


Figure 6.1: General architecture of our models for the constrained track. L represents the number of layers. N_{src} and N_{hyp} represent the number of words in the source and hypothesis sentences, respectively. $N = N_{\text{src}} + N_{\text{hyp}}$ is the number of words after concatenating the two sentences. D is the size of hidden vectors.

6.3 Constrained Track

The goal of the constrained track is to identify machine translation errors without explicit word-level annotation. More precisely, it aims at performing word-level quality estimation by casting the task as a prediction explainability problem. In the context of QE, explanations can be seen as highlights, representing the relevance of input words w.r.t. the model's prediction via continuous scores. We next describe the datasets, models, and explainability methods that we used for this track.

6.3.1 Datasets

Seeking to improve the performance of our models on the zero-shot language pairs (LPs), we used all language pairs from the MLQE-PE dataset (Fomicheva et al., 2022b) to train our models for both tracks. For RO-EN and ET-EN, we evaluated our models on the validation set of these LPs. For the two zero-shot LPs, DE-ZH and RU-DE, we used the 20 sentences made available by the shared task and the validation sets of EN-ZH and EN-DE to improve the robustness of the evaluation of explanations w.r.t. the target language. We used word-level labels to train word-level models for the unconstrained track only. For sentence-level models, we supervise our models using DA scores.

6.3.2 Sentence-level Models

Since QE is a fundamental tool in many MT pipelines, we focus our efforts on designing and explaining QE systems with high sentence-level performance. Therefore, we opted to follow the recent trend in this area (Kepler et al., 2019b; Ranasinghe et al., 2020) and employed two pretrained multilingual language models as the feature extractors for our models: XLM-RoBERTa and RemBERT.

The overall architecture of our models is shown in Figure 6.1. The tokenized source $\boldsymbol{s} = \langle s_1, ..., s_n \rangle$ and hypothesis $\boldsymbol{t} = \langle t_1, ..., t_m \rangle$ sentences are concatenated and passed as input to the encoder, which produces hidden state vectors $\boldsymbol{H}_0, ..., \boldsymbol{H}_L$ for each layer $0 \leq \ell \leq L$, where $\boldsymbol{H}_i \in \mathbb{R}^{(n+m)\times d}$. Next, all hidden states are fed to a scalar mix module (Peters et al., 2018b) that learns a weighted sum of the hidden states of each layer of the encoder, producing a new sequence of aggregated hidden states $\boldsymbol{H}_{L+1}$. We split $\boldsymbol{H}_{L+1}$ into source

ENCODER	RO-EN	ET-EN	DE-ZH	RU-DE
OpenKiwi	0.820	0.757	0.395	0.176
XLM-R	0.878	0.756	0.521	**0.563**
XLM-R-M	0.877	**0.780**	**0.797**	0.352
RemBERT	**0.883**	0.762	-0.002	0.505

Table 6.1: Pearson correlation of our sentence-level QE systems by varying the model used as the encoder layer.

$\boldsymbol{H}_{src} \in \mathbb{R}^{n \times d}$ and hypothesis hidden states $\boldsymbol{H}_{hyp} \in \mathbb{R}^{m \times d}$, which are independently passed to an average pooling layer to get their sentence representations $\boldsymbol{h}_{src}$ and $\boldsymbol{h}_{hyp}$. We merge both representations via a convex combination with $\alpha = 0.5$ to encourage the model to use both source and hypothesis contexts. Finally, we pass the combined vector to a 2-layered feed-forward module in order to get a sentence score prediction $\hat{y} \in \mathbb{R}$. Moreover, attention matrices $\boldsymbol{A}_1, ..., \boldsymbol{A}_L$ are also recovered as a by-product of the forward propagation, where $\boldsymbol{A}_i \in \mathbb{R}^{(n+m)\times(n+m)}$. The hyperparameters used for training can be found in §D.1.

XLM-RoBERTa as encoder. We set a XLM-RoBERTa Large (XLM-R, Conneau et al. 2020) as the encoder layer.[2] XLM-R is a cross-lingual transformer pretrained on massive amounts of multi-lingual data. It consists of 24 encoder blocks with 16 attention heads each. Following (Zerva et al., 2021) we train our complete model on DAs by using adapters for the XLM-R encoder (Houlsby et al., 2019; Pfeiffer et al., 2020) to adapt it to the domain specific data of the QE task with minimal training effort.

XLM-RoBERTa for zero-shot LPs. To improve the robustness of XLM-R on out-of-domain data, we used an XLM-RoBERTa Large model that was trained with DA's from the metrics shared task.[3] Next, we set it as the encoder layer, and adapted it for predicting DAs from the MLQE corpus as in (Zerva et al., 2021). Altogether, the data from the Metrics shared task encompasses 30 language pairs from the news domain—yet, the zero-shot LPs are not included in this set. The hyperparameters and the training regime of this model are the same as the previously described XLM-R. We denote this model as XLM-R-M from here on.

RemBERT as encoder. We replace the XLM-R by a RemBERT model as the encoder layer (Chung et al., 2021).[4] Multilingual BERT (Devlin et al., 2019) has been shown to provide complementary performance to XLM-based models for sentence-level and word-level QE (Kepler et al., 2019a). We opted to use RemBERT since it can be seen as a larger multilingual BERT with decoupled input and output embeddings, which helps to accelerate finetuning as output embeddings can be discarded. It consists of 32 encoder blocks with 18 attention heads each. Rather than aggregating layers with the scalar mix layer, we perform average pooling over the hidden states of the last layer of RemBERT. For training, we simply finetune the whole model with small learning rates.

Results. Table 6.1 summarizes the performance of our sentence-level models on the validation set in terms of Pearson correlation for each language pair evaluated in the shared task. For completeness, we show results for the 20 sentences made available by the shared task for DE-ZH and RU-DE. We also include OpenKiwi with a

[2] https://huggingface.co/xlm-roberta-large
[3] https://huggingface.co/Unbabel/xlm-roberta-wmt-metrics-da
[4] https://huggingface.co/google/rembert

XLM-R Large as the encoder for comparison. We note that results for DE-ZH and RU-DE are noisy due to the small amount of validation data available for these LPs.

6.3.3 Explainability Methods

Several explainability methods can be used to extract highlights from a trained model in a post-hoc fashion. It is also possible to design a model that is explainable by construction, such as rationalizers (Lei et al., 2016; Bastings et al., 2019). We investigate rationalizers, attention, gradient, and perturbation-based methods for this shared task.

Attention-based methods. Since the backbone of our models consists of pretrained multilingual transformers, we studied their main component—the multi-head attention mechanism—expecting to find interpretability patterns that assign higher scores to words associated with translation errors. We extracted the following explanations from the multi-head attention mechanism:

- **Attention weights:** average the attention matrix $\boldsymbol{A}$ row-wise for all heads in all layers, amounting to a total of $24 \times 16 = 384$ and $32 \times 18 = 576$ explanation vectors $\boldsymbol{a} \in \mathbb{R}^{n+m}$ for XLM-R and RemBERT-based models, respectively.
- **Cross-attention weights:** by manual inspection of attention weights, we noticed that some attention heads learn plausible connections from source-to-hypothesis and hypothesis-to-source. Therefore, instead of computing a row-wise average of the entire attention matrix, we average only cross-alignment rows.[5]
- **Attention $\times$ Norm:** following the findings of Kobayashi et al. (2020), we scale attention weights by the norm of value vectors $\|\boldsymbol{V}\boldsymbol{W}_h^V\|_2$.

Gradient-based methods. Explanations extracted by storing gradients computed during the backward propagation is a standard tool used to interpret NLP models. For this shared task, we investigate the following gradient-based methods:[6]

- **Gradient $\times$ Hidden States:** we compute gradients w.r.t. the hidden states of each layer, and multiply the resultant vectors by the hidden state vectors themselves: $\nabla_{\boldsymbol{H}_i} \times \boldsymbol{H}_i \in \mathbb{R}^{N+M}$, for $0 \leq i \leq L+1$.
- **Gradient $\times$ Attention:** the same as before, but we use the output of the multi-head attention module instead of the hidden states.
- **Integrated Gradients**: we extract integrated gradient explanations w.r.t. the hidden states of each layer. We use a zero-vector as the baseline. We map gradients to explainability scores by normalizing them by their L2 norm and summing the hidden dimensions: $\mathbf{1}^\top \nabla_{\boldsymbol{H}_i} / \|\nabla_{\boldsymbol{H}_i}\|_2$.

Perturbation-based methods. As baselines, we also extracted explanations using **LIME** (Ribeiro et al., 2016) and a **leave-one-out** strategy, where we replace the "erased" token by the `[mask]` token, which is used for the masked-language model training of XLM-R and RemBERT.

[5] Note that we can get cross-attentions from XLM-R and RemBERT by selecting only the words of the source that attend to the hypothesis and vice-versa.

[6] Our implementation is based on Captum: https://captum.ai/

	RO-EN		ET-EN		DE-ZH		RU-DE	
ENCODER	Src.	Tgt.	Src.	Tgt.	Src.	Tgt.	Src.	Tgt.
OpenKiwi	0.581	0.620	0.488	0.554	0.**271**	0.184	0.243	0.029
XLM-R	0.610	0.644	**0.503**	**0.559**	0.230	0.312	0.273	0.061
XLM-R-M	**0.636**	**0.667**	0.464	0.530	0.262	**0.336**	**0.343**	0.179
RemBERT	0.624	0.659	0.474	0.555	0.173	0.211	0.247	**0.201**

Table 6.2: Source and target MCC results of our word-level QE systems by varying the model used as the encoder layer. The values of λ for each model are: $10^3, 10^4, 10^4, 10^4$.

Rationalizers. We append a differentiable binary mask layer (Bastings et al., 2019) on top of the XLM-R model in order to select which tokens are passed on for an estimator for the prediction of a sentence-level score. For each instance, we take the model representations from the scalar-mix layer and pass it to an encoder module, in which we sample a binary mask $\boldsymbol{z} \in [0,1]^{n+m}$ from a relaxed Bernoulli distribution (Maddison et al., 2017; Jang et al., 2017), and pass $\boldsymbol{z} \odot [\boldsymbol{s}; \boldsymbol{t}]$ to an estimator module, which re-embeds the masked input and pass it to a linear output layer. Therefore, good explanations $\boldsymbol{z}$ will aid the estimator in producing good sentence-level scores. In training time, the parameters of the encoder and the estimator are jointly trained. In test time, we do not sample binary masks. Instead, we use the relaxed Bernoulli distribution probabilities as explanations.

6.4 Unconstrained Track

In this track, we opted to use word-level annotation by incorporating a word-level loss to our previous models. To do this, we apply a map from word pieces to tokens after the scalar mix layer and pass the hidden vectors of each token through a feed-forward layer with a sigmoid activation to predict scores $\hat{y}_i \in [0,1]$. We weight the word-level loss by λ and sum it with the sentence-level loss. As baseline, we train a XLM-R Large model using OpenKiwi with the default hyperparameters. For all word-level models, we train with $\lambda \in \{10^3, 10^4, 10^5\}$ and save the checkpoint with the best performance on the validation set.

Results. Table 6.2 shows the results of our word-level models on the validation set in terms of Matthews correlation coefficient (MCC) for each LP evaluated in the shared task. For completeness, we include the results for the 20 available sentences for DE-ZH and RU-DE.

6.5 Experimental Results

Although we can regard the extracted explanations as errors in the translation output, an analogous evaluation of word-level QE is not straightforward since the standard metrics require binary labels rather than continuous scores. Therefore, the explanations are evaluated against the ground-truth word-level labels in terms of the Area Under the Curve (AUC), Average Precision (AP), and Recall at Top-K (R@K) metrics only on the subset of translations that contain errors.

Furthermore, since all of our models use subword tokenization, to get explanations for an entire word, we tried aggregating the scores of its word pieces by taking the sum, mean, or max, and we found that taking the sum performs better overall.

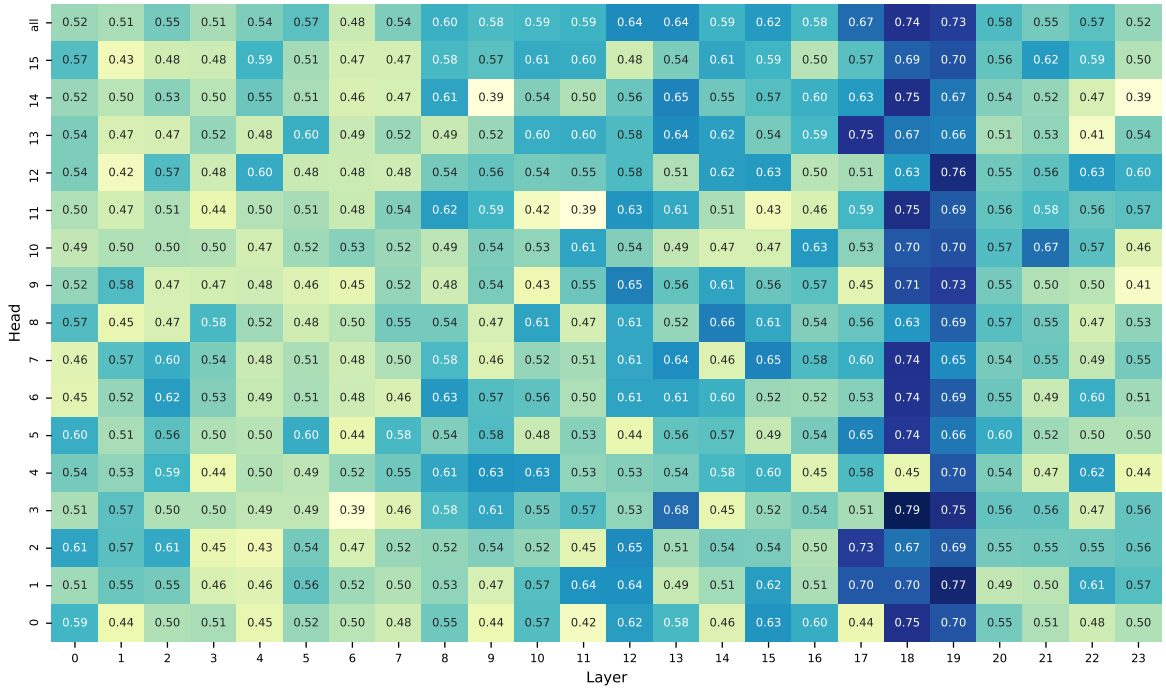


Figure 6.2: Target AUC of different attention heads at each layer of our XLM-R model for RO-EN. The last tick on the y-axis represents the average of all attention heads.

6.5.1 Constrained Track

Attention heads are better alone. We found that some attention heads (mostly at upper layers) learned to focus on words associated with BAD tags, achieving great performance in terms of AUC and AP on the validation set. We show in Figure 6.2 the target AUC of different attention heads per layer as a heatmap for RO-EN, with darker colors indicating higher results.[7] We can see that attention heads in layers 18 and 19 perform better than other layers in general, and that some attention heads solely outperform the average of all attention heads for all respective layers. For example, the attention head 3 at layer 18 achieves an AUC score of 0.79, while the average of all attention heads from layer 18 gets an AUC score of 0.74 (5 points difference). The findings are similar for source AUC, with the exception that attention heads at lower layers also seem to achieve comparable, yet not better, results. This behavior was also noted by Fomicheva et al. (2022a), with the difference that we analyzed attention heads independently rather than averaging them at each layer. Kobayashi et al. (2020) also arrive at similar findings but in terms of alignment error rate in a neural machine translation context.

Attention $\times$ Norm outperforms other explainers. By scaling attention probabilities by the L2 norm of value vectors, we improved the performance further. All of our best results consist of attention-based explainers, with the majority being the explanations that consider the norm of value vectors. We show the results of our best explainers on the validation set of RO-EN in Table 6.3 using XLM-R as encoder.[8] When using XLM-R-M or RemBERT as encoder the results are similar, except that the best explainer comes from different attention heads at different upper layers.

Overall, we observed that attention methods outperform gradient and perturbation methods by a considerable

[7] We got similar findings for ET-EN.
[8] Results for ET-EN follow the same trend (see §D.2).

	Source			Target		
EXPLAINER	AUC	AP	R@K	AUC	AP	R@K
Attention	0.7445	0.6353	0.5164	0.7894	0.7189	0.6054
Cross-attention	0.7514	0.6345	0.5170	0.8066	0.7378	0.6293
Attention × Norm	**0.7851**	**0.6875**	**0.5701**	**0.8136**	**0.7432**	**0.6342**
Gradient × Hidden States	0.6949	0.5629	0.4399	0.6780	0.5388	0.4044
Gradient × Attention	0.7104	0.5942	0.4913	0.7618	0.6747	0.5628
Integrated Gradients	0.6539	0.5251	0.4059	0.6560	0.5148	0.3853
LIME	0.6470	0.5160	0.3922	0.5892	0.4576	0.3300
Leave-one-out	0.6970	0.5673	0.4409	0.5921	0.4752	0.3567
Relaxed-Bernoulli Rationalizer	0.4803	0.3638	0.2483	0.5434	0.4043	0.2914

Table 6.3: Constrained track results for different explainability methods on the validation set of RO-EN using XLM-R as encoder.

margin, and gradients w.r.t. attention outputs yield better results than gradients w.r.t. hidden states, indicating that the information stored in attention heads is valuable. In Figure 6.3 we show the attention map of two attention heads that perform well in terms of source AUC and target AUC on the validation set of RO-EN. We noted qualitatively that attention-heads that perform well on source AUC usually focus on cross-sentence tokens,[9] whereas attention-heads that have good results in terms of target AUC usually focus on hypothesis tokens. Lastly, our strategy of appending a bottleneck layer acting as rationalizer did not work well, achieving worse results than perturbation-based methods.

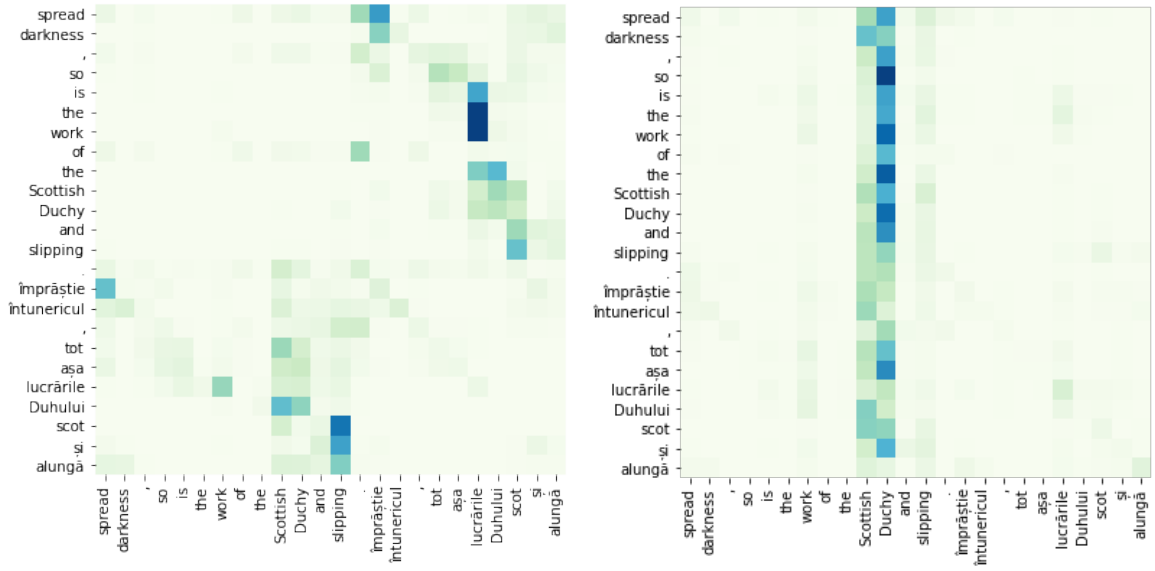


Figure 6.3: Example of two attention maps from particular heads that perform well on source AUC (left) and target AUC (right) for RO-EN.

Results for all LPs. We show the results on the validation set for all LPs in Table 6.4 (left) with the best Attention × Norm explanations for each tested encoder. We also report results of ensembled explanations, which are obtained by simply averaging selected Attention × Norm explanations from models with different encoders. Results for all tested explanation methods can be found in §D.2. When comparing single encoders for in-domain LPs, we see that explanations from our XLM-R-based model achieved the best results for source and

[9]Cross-sentence tokens are hypoehsis tokens attended by source tokens and also source tokens attended by hypoehsis tokens.

LP	Encoder	Source (constrained)			Target (constrained)			Source (unconstrained)			Target (unconstrained)		
		AUC	AP	R@K	AUC	AP	R@K	AUC	AP	R@K	AUC	AP	R@K
RO-EN	OpenKiwi	-	-	-	-	-	-	0.907	0.811	0.704	0.921	0.826	0.718
	XLM-R	0.785	0.687	0.570	0.814	0.743	0.634	0.914	0.825	0.722	0.928	0.851	0.764
	XLM-R-M	0.753	0.661	0.548	0.769	0.693	0.593	0.913	0.826	0.724	0.926	0.851	0.761
	RemBERT	0.784	0.699	0.590	0.790	0.686	0.572	0.918	0.831	0.731	0.934	0.862	0.769
	Ensemble	**0.807**	**0.720**	**0.607**	**0.842**	**0.772**	**0.662**	**0.927**	**0.844**	**0.744**	**0.942**	**0.874**	**0.786**
ET-EN	OpenKiwi	-	-	-	-	-	-	0.848	0.749	0.635	0.873	0.798	0.692
	XLM-R	0.733	0.618	0.486	0.740	0.648	0.530	0.858	0.768	0.656	0.881	0.814	0.711
	XLM-R-M	0.623	0.504	0.367	0.712	0.625	0.513	0.854	0.751	0.630	0.875	0.804	0.704
	RemBERT	**0.750**	**0.638**	**0.523**	0.708	0.595	0.476	0.851	0.747	0.631	0.881	0.806	0.703
	Ensemble	0.744	0.637	0.509	**0.764**	**0.680**	**0.569**	**0.870**	**0.778**	**0.668**	**0.896**	**0.832**	**0.735**
DE-ZH	OpenKiwi	-	-	-	-	-	-	0.721	0.616	0.545	0.648	0.483	0.356
	XLM-R	0.720	0.465	0.288	0.683	0.542	0.406	0.674	0.486	0.298	0.650	0.511	0.352
	XLM-R-M	0.773	**0.609**	**0.454**	0.697	0.545	0.427	0.711	0.574	0.463	**0.712**	**0.595**	**0.468**
	RemBERT	0.762	0.579	0.405	0.692	0.470	0.358	0.619	0.443	0.341	0.585	0.445	0.354
	Ensemble	**0.792**	0.581	0.440	**0.711**	**0.575**	**0.477**	**0.745**	**0.635**	**0.548**	0.705	0.575	0.418
RU-DE	OpenKiwi	-	-	-	-	-	-	0.727	0.620	0.559	0.620	0.409	0.359
	XLM-R	0.719	0.400	0.316	0.822	0.500	0.335	0.729	0.604	0.485	0.623	0.369	0.282
	XLM-R-M	0.743	0.529	0.425	0.838	0.532	0.369	0.740	0.645	0.545	0.640	0.470	**0.447**
	RemBERT	0.776	**0.646**	**0.550**	0.826	0.537	0.418	**0.802**	0.712	0.607	**0.721**	0.504	0.393
	Ensemble	**0.804**	0.604	0.459	**0.855**	**0.628**	**0.514**	0.799	**0.716**	**0.616**	0.719	**0.521**	0.439

Table 6.4: Constrained (left) and unconstrained (right) track results on the validation set for all LPs using the Attention × Norm explainer.

target metrics on RO-EN, with competitive results on ET-EN, for which explanations from a RemBERT-based model ranked first for source metrics. Despite being a simple strategy, we usually got ∼2 more points of AUC, AP, and R@K by averaging attention explanations. We note that explanations from XLM-R-M and RemBERT perform well on the 20 sentences made available by the shared task for zero-shot LPs. Between XLM-R and XLM-R-M, explanations from the latter lead to better results for both DE-ZH and RU-DE, suggesting that the additional data from the Metrics shared task might help to improve the robustness for zero-shot LPs. Ensembling explanations also leads to higher performance for zero-shot LPs. However, we note that results for DE-ZH and RU-DE are noisy due to the small amount of validation data.

6.5.2 Unconstrained Track

In this track, we used the predicted probabilities of BAD tags from supervised word-level QE models as explanation scores. The results are shown in Table 6.4 (right). As found in the constrained track, XLM-R and RemBERT-based models perform better for in-domain LPs, while XLM-R-M and RemBERT lead to better results for zero-shot LPs. Consistent with our findings in the constrained track, ensembling explanations also reflects in improvements in this track.

6.6 Official results

The official results of the shared task are shown in Table 6.5 for both tracks in terms of R@K.[10] Our final submissions consist of ensembled explanations since they proved to perform better for all LPs in both tracks. More specifically, we ensembled Attention × Norm explainers from the models shown in Table 6.4 (left) for the

[10]We report results in terms of R@K for consistency with the 2022 edition of the shared task, which we cover in the next chapter.

Team	Source				Target			
	RO-EN	ET-EN	DE-ZH	RU-DE	RO-EN	ET-EN	DE-ZH	RU-DE
Constrained track:								
Baseline: Random	0.15	0.19	0.17	0.24	0.19	0.25	0.17	0.22
Baseline: XMover+SHAP	0.15	0.23	0.16	0.26	0.30	0.34	0.22	0.23
Baseline: TransQuest+LIME	0.24	0.31	0.20	0.32	0.42	0.43	0.14	0.16
HeyTUDa (Eksi et al., 2021)	-	-	-	-	0.38	0.41	0.18	0.23
CLIP-UMD (Kabir and Carpuat, 2021)	0.37	0.45	0.25	0.37	0.49	0.53	0.27	0.36
Gringham (Leiter, 2021)	0.45	0.59	0.27	**0.56**	0.61	0.60	0.22	0.46
IST-Unbabel (*this work*)	**0.62**	**0.64**	**0.32**	0.52	**0.68**	**0.63**	**0.37**	**0.47**
Unconstrained track:								
CUNI Prague (Polák et al., 2021)	0.70	0.76	0.27	0.56	0.73	0.75	0.30	0.50
NICT Kyoto* (Rubino et al., 2021)	**0.75**	**0.77**	**0.51**	**0.71**	**0.78**	**0.76**	**0.57**	**0.74**
IST-Unbabel (*this work*)	0.71	**0.77**	0.35	0.59	0.75	**0.76**	0.34	0.52

Table 6.5: Official test set results in terms of R@K (Fomicheva et al., 2021). *As noted by the organizers, the NICT Kyoto team used additional synthetic data for word-level supervision, thus we categorize them in the unconstrained track.

constrained track; and we ensembled the predicted probabilities of BAD tags from the models shown in Table 6.4 (right) for the unconstrained track. We note that results for the unconstrained track are superior to those obtained in the constrained track. However, the opposite is true for DE-ZH on the target side, suggesting that extracting rationales from a sentence-level QE model is a promising weak-supervised strategy for identifying translation errors. In summary, our submissions demonstrate superior performance across nearly all language pairs in the constrained track when compared to other teams, highlighting the effectiveness and robustness of our approach in handling diverse language combinations. In the unconstrained track, the NICT Kyoto team obtained the best overall results by augmenting the training data with millions of synthetic examples (Rubino et al., 2021), while our approach achieved the second-best results by using only the official in-domain data.

6.7 Conclusions and Subsequent Works

We have shown that the multi-head mechanism—the bedrock on which transformers are built—is able to learn the importance of tokens associated with BAD tags. Furthermore, composing explanations in the form of attention probabilities scaled by the norm of value vectors leads to further improvements (Kobayashi et al., 2020). Ensembling these explanations yields the best results overall for all tested metrics on all LPs, including zero-shot ones.

Transformers are composed of many parameters across a vast amount of heads and layers. Strategies that explore how explanations are formed as we move to upper layers are promising, such as computing attention flows and differentiable binary masks per layer (Abnar and Zuidema, 2020; De Cao et al., 2020). This shared task focused only on the intersection between explainability and QE, yet for future work we plan to apply explainability methods to recent MT metrics such as COMET (Rei et al., 2020a,b; Glushkova et al., 2021) and BLEURT (Sellam et al., 2020a,b).

As we will detail in the next two chapters, this work paved the way for building better explainability methods for QE, such as exploring the combination of attention with gradient information and automatically identifying relevant attention heads (Rei et al., 2022b; Fernandes et al., 2022).

7

Sparse Bottleneck Layer for Explainable Quality Estimation

Contents

In the previous chapter, we demonstrated that attention-based methods outperformed other methods for explaining transformer-based QE models. Building on this finding, we continued to explore information contained in attention heads in a new collaboration with the Unbabel AI team to the WMT 2022 QE Shared Task, where we participated on all three subtasks: (i) Sentence and Word-level Quality Prediction; (ii) Explainable QE; and (iii) Critical Error Detection. However, in this chapter we cover only the part concerned with Explainable QE.

A significant weakness of our previous approach, detailed in Chapter 6, is that we had to manually search over all attention heads to find the best ones, as determined by evaluation on a held-out dataset. To address this issue, we improved our methods by designing a sparse bottleneck layer that aggregates hidden states from selected attention heads to produce a sentence-level prediction, which we call *Sparse Head Mix*. Drawing inspiration from the literature (Chrysostomou and Aletras, 2022), we further improve our explanations by combining attention with gradient information extracted at the attention head level. We show that the coefficients extracted from the Sparse Head Mix module can be leveraged to automatically identify relevant attention heads, thus alleviating the cost of exhaustive manual search. This innovation not only saves time and effort but also improves the interpretability of the model by providing insight into which attention heads are contributing most to the final prediction. Overall, we found that these two innovations were essential to our success in winning the shared task in that year for 7 out of 9 language pairs.

Chapter based on Rei et al. (2022b).

7.1 Motivation

In this work, we leverage the similarity between the tasks of MT evaluation and QE and bring together the strengths of two frameworks, COMET (Rei et al., 2020a), which has been originally developed for reference-based MT evaluation, and OPENKIWI (Kepler et al., 2019b), which has been developed for word-level and sentence-level QE. Namely, we implement some of the features of the latter, as well as other new features, into the COMET framework. The result is COMETKIWI, which links the predictor-estimator architecture with COMET training-style, and incorporates word-level sequence tagging.

Given that some language pairs (LPs) in the test set were not present in the training data, we aimed at developing QE systems that achieve good multilingual generalization and that are flexible enough to account for unseen languages through few-shot training. To do so, we start by pretraining our QE models on Direct Assessments (DAs) annotations from the previous edition's Metrics shared task as it was shown to be beneficial in our previous submission (Zerva et al., 2021; Treviso et al., 2021). Then we fine-tune our models with the data made available by the shared task.[1]

We experimented with different pretrained multilingual transformers as the backbones of COMETKIWI, and we developed new explainability methods to interpret them. We describe our systems and their training strategies in §7.3. Overall, our main contributions are:[2]

- We investigate explainability methods on top of COMETKIWI, a model that and achieved the best sentence and word-level results on the shared task for several language pairs (Zerva et al., 2022).

[1]For zero-shot LPs we use 500 training examples which means we turn it into a few-shot setting. The only exception is English→Yoruba which was kept zero-shot.

[2]Code available at: https://github.com/Unbabel/COMET

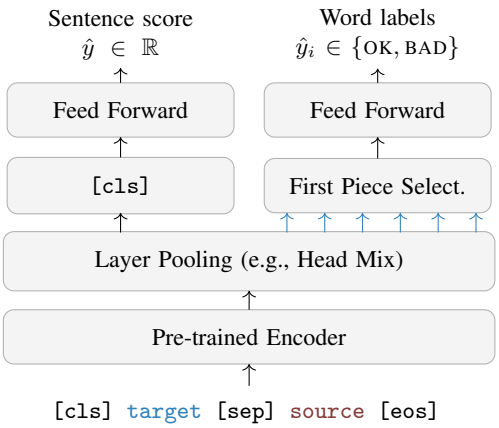


Figure 7.1: General architecture of COMETKIWI for sentence-level (left part) and word-level QE (right part).

- We propose a new interpretability method that uses attention and gradient information along with a head-level scalar mix module that further refines the relevance of attention heads.
- We demonstrate that our approach can be explored to perform word-level QE in an unsupervised fashion, and provide complementary information that can be used to further boost the performance of COMETKIWI.

Our submitted systems achieve the best multilingual results on all tracks by a considerable margin: for sentence-level DA our system achieved a 0.572 Spearman correlation (+7% than the second best system); for word-level our system achieved a 0.341 MCC score (+2.4% than the second best system); and for Explainable QE our system achieved 0.486 R@K score (+10% than the second best system). The official Explainable QE results for all LPs are presented in Table 7.4.

7.2 Background

Quality Estimation. QE systems are usually designed according to the granularity in which predictions are made, such as sentence and word-level. In sentence-level QE, the goal is to predict a single quality score $\hat{y} \in \mathbb{R}$ given the whole source and its translation as input. Word-level QE works in a lower granularity level, with the goal of predicting binary quality labels $\hat{y}_i \in \{\text{OK}, \text{BAD}\}$ for all $1 \leq i \leq n$ *machine-translated words*, indicating whether that word is a translation error or not. More details about QE can be found in §2.2.

Transformers. The multi-head attention mechanism is the key component in transformers, being responsible for contextualizing the information within and across input sentences (Vaswani et al., 2017). We describe the multi-head attention mechanism in detail in §2.1.3.

7.3 Implemented Systems

The overall architecture of our models is shown in Figure 7.1. The machine translated sentence $\boldsymbol{t} = \langle t_1, ..., t_n \rangle$ and its source sentence counterpart $\boldsymbol{s} = \langle s_1, ..., s_m \rangle$ are concatenated and passed as input to the encoder, which produces d-dimensional hidden state vectors $\boldsymbol{H}_1, ..., \boldsymbol{H}_L$ for each layer $1 \leq \ell \leq L$, where $\boldsymbol{H}_\ell \in \mathbb{R}^{(n+m)\times d}$. Next, all hidden states are fed to a scalar mix module (Peters et al., 2018b) that learns a

weighted sum of the hidden states of each layer of the encoder, producing a new sequence of aggregated hidden states $\boldsymbol{H}_{\text{mix}}$ as follows:

$$\boldsymbol{H}_{\text{mix}} = \lambda \sum_{\ell=1}^{L} \beta_\ell \boldsymbol{H}_\ell, \tag{7.1}$$

where λ is a scalar trainable parameter, $\boldsymbol{\beta} \in \triangle^L$, is given by $\boldsymbol{\beta} = \mathsf{sparsemax}(\boldsymbol{\phi})$ using a sparse transformation (Martins and Astudillo, 2016), with $\boldsymbol{\phi} \in \mathbb{R}^L$ as learnable parameters. As it has been shown in (Rei et al., 2022a) not all layers are relevant and thus, using sparsemax we learn to ignore irrelevant layers.

For sentence-level models, the hidden state of the first token (`<cls>`) is used as sentence representation $\boldsymbol{H}_{\text{mix},0} \in \mathbb{R}^d$, which, in turn, is passed to a 2-layered feed-forward module in order to get a sentence score prediction $\hat{y} \in \mathbb{R}$. Moreover, attention matrices $\boldsymbol{A}_{1,1}, ..., \boldsymbol{A}_{L,H}$ for all layers and heads are also recovered as a by-product of the forward propagation.

Pretraining on Metrics Data. Every year, the WMT News Translation shared task organizers collect human judgments in the form of DAs. The collective corpora of 2017, 2018, and 2019 contain 24 LPs and a total of 657k samples with source, target, reference, and DA score. We follow the experiments from the previous edition carried by Zerva et al. (2021) and start by pretraining our QE models on this data using the learning objective proposed by UniTE (Wan et al., 2022), which incorporates reference translations into training and thus acts as data augmentation.

Setting pretrained transformers as encoders. We follow the recent trend (Kepler et al., 2019b; Ranasinghe et al., 2020) and experiment with three different pretrained multilingual transformers as the encoder layer of our models: XLM-R Large (Conneau et al., 2020),[3] InfoXLM Large (Chi et al., 2021),[4] and RemBERT (Chung et al., 2021).[5] XLM-R and InfoXLM consist of 24 encoder blocks with 16 attention heads each, whereas RemBERT has 32 encoder blocks with 18 attention heads each.

7.3.1 Explainable QE

The goal of the Explainable QE task is to identify machine translation errors without relying on word-level label information. In other words, it can be cast as an unsupervised word-level quality estimation problem, where explanations can be seen as highlights, representing the relevance of input words w.r.t. the model's prediction via continuous scores, aiming at identifying tokens that were not properly translated.

Several explainability methods can be used to extract highlights from a sentence-level model, such as post-hoc (Ribeiro et al., 2016; Arras et al., 2016) or inherently interpretable methods (Lei et al., 2016; Guerreiro and Martins, 2021). In our submission, we opted to use attention-based methods as they achieved the best results in the previous constrained track of the Explainable QE shared task (Fomicheva et al., 2021). Concretely, we take inspiration in the method we developed in the previous chapter (Treviso et al., 2021), which consists of scaling attention weights by the ℓ_2-norm of value vectors (Kobayashi et al., 2020) and finding the attention heads with the best performance on the dev set, and propose two new modifications:

[3] https://huggingface.co/xlm-roberta-large
[4] https://huggingface.co/microsoft/infoxlm-large
[5] https://huggingface.co/google/rembert

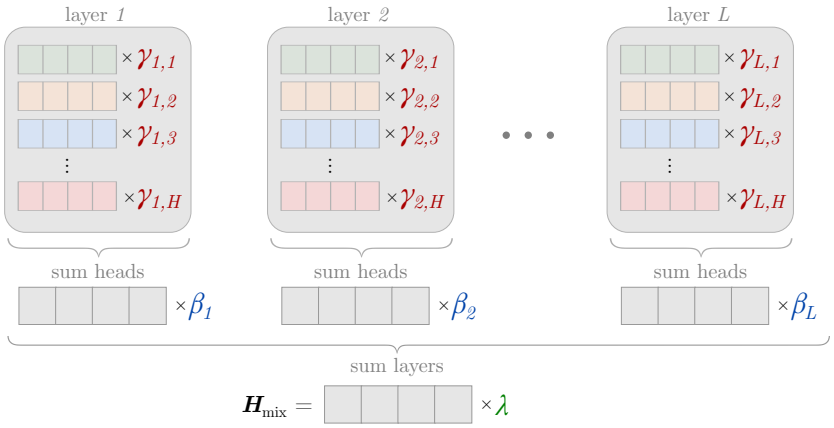


Figure 7.2: Our Head Mix component. At each layer ℓ, the hidden states $\boldsymbol{h}_{\ell,h} \in \mathbb{R}^{n \times d_h}$ for each head h are associated with learnable coefficients $\boldsymbol{\gamma}_\ell \in \triangle^H$, and are subsequently aggregated and multiplied by learnable coefficients $\boldsymbol{\beta} \in \triangle^L$, which are finally summed in order to produce a final bottlenecked representation $\boldsymbol{H}_{\text{mix}} \in \mathbb{R}^{n \times d}$ scaled by $\gamma \in \mathbb{R}$.

- **Attention $\times$ GradNorm:** Following the findings of Chrysostomou and Aletras (2022), we decided to extract explanations that consider both attention and gradient information. More precisely, we propose to scale the attention weights $\boldsymbol{A}_{\ell,h} \in \mathbb{R}^{n \times n}$ by the ℓ_2-norm of the gradient of value vectors $\boldsymbol{V}_{\ell,h} \in \mathbb{R}^{n \times d_h}$:

$$\boldsymbol{A}_{\ell,h} \odot \|\nabla_{\boldsymbol{V}_{\ell,h}}\|_2, \tag{7.2}$$

  where $\odot$ represents a broadcasted element-wise multiplication and $\boldsymbol{V}_{\ell,h} = \boldsymbol{V}_\ell \boldsymbol{W}^V_{\ell,h}$.

- **Head Mix:** We reformulate the scalar mix module (Eq. 7.1) to consider different weights for representations coming from different attention heads $\boldsymbol{h}_{\ell,h} \in \mathbb{R}^{n \times d_h}$ as follows:

$$\boldsymbol{H}_{\text{mix}} = \lambda \sum_{\ell=1}^{L} \beta_\ell \sum_{h=1}^{H} \gamma_{\ell,h} \boldsymbol{h}_{\ell,h}, \tag{7.3}$$

  where the *layer* mix weights $\boldsymbol{\beta} \in \triangle^L$ are given by $\boldsymbol{\beta} = \pi(\boldsymbol{\phi})$, and the *head* mix coefficients $\boldsymbol{\gamma}_\ell \in \triangle^H$ are given by $\boldsymbol{\gamma}_\ell = \pi(\boldsymbol{\theta}_\ell)$, with $\lambda \in \mathbb{R}$, $\boldsymbol{\phi} \in \mathbb{R}^L$ and $\boldsymbol{\theta} \in \mathbb{R}^{L \times H}$ as learnable parameters. We illustrate this component in Figure 7.2. We experimented both with a dense (π as softmax) and with a sparse transformation (π as sparsemax, Martins and Astudillo 2016), which may rule out the contribution of several heads and layers in the final representation. After training, the Head Mix coefficients can help to identify attention heads with high validation performance, which is helpful for explaining zero-shot LPs.

Furthermore, since all of our sentence-level models use subword tokenization, to get explanations for an entire word we follow Treviso et al. (2021) and sum the scores of its word pieces.

Ensembling explanations. We average the explanation scores of different attention heads for our final submissions. We decided which heads to aggregate together by taking the top-64 heads with the highest Head Mix coefficients $\beta_\ell \times \gamma_{\ell,h}$ and comparing their performance on the dev set, picking the top-5 with the highest results.

7.4 Experimental Results

For our experiments, we split the provided development sets into two equal size halves creating a new internal devset and an internal testset. The resulting sets contain $\approx$ 500 segments per language pair for both DA

Method	Direct Assessment						MQM			
	en-cs	en-ja	en-mr	km-en	ps-en	avg.	en-de	en-ru	zh-en	avg.
Baseline (Treviso et al., 2021)†	0.602	0.510	0.428	0.636	0.633	0.562	0.529	0.552	0.450	0.510
InfoXLM as encoder										
Attn × GradNorm	0.602	0.495	0.417	0.653	0.648	0.563	0.539	0.559	0.474	0.524
+ Soft Head Mix	0.600	0.495	0.426	0.656	0.653	0.566	0.532	0.563	0.467	0.521
+ Sparse Head Mix	0.604	0.503	0.421	0.658	0.660	0.569	0.541	0.551	0.454	0.515
Ensemble	0.641	0.521	0.440	0.669	0.667	0.588	**0.580**	**0.603**	**0.505**	**0.563**
+ Soft Head Mix	0.621	0.501	0.432	0.681	0.661	0.579	0.567	0.588	0.504	0.553
+ Sparse Head Mix	**0.645**	0.519	**0.450**	0.688	0.675	0.595	0.574	0.582	0.484	0.547
RemBERT as encoder										
Attn × GradNorm	0.596	0.511	0.427	0.675	0.676	0.577	0.474	0.532	0.448	0.485
+ Soft Head Mix	0.588	0.538	0.430	0.658	0.654	0.574	0.473	0.529	0.455	0.486
+ Sparse Head Mix	0.588	0.534	0.428	0.658	0.652	0.572	0.470	0.530	0.443	0.481
Ensemble	0.609	0.551	0.443	**0.702**	0.685	0.598	0.516	0.554	0.506	0.525
+ Soft Head Mix	0.613	**0.561**	0.448	0.699	0.692	0.603	0.521	0.558	0.498	0.526
+ Sparse Head Mix	0.620	0.557	0.447	**0.702**	**0.691**	**0.604**	0.511	0.551	0.503	0.522

Table 7.1: Explainable QE task results in terms of the average of AUC, AP and R@K. †We used InfoXLM to compute the results for the baseline.

and MQM, word and sentence-level. As for baselines we used our submitted explainer from previous shared task, concretely, we used the Attn × Norm explainer (Treviso et al., 2021). Moreover, we extract explanations from the best performing sentence-level models, as evaluated in our internal test set.

Since the explanations are given as continuous scores, they are evaluated against the ground-truth word-level labels in terms of the Area Under the Curve (AUC), Average Precision (AP), and Recall at Top-K (R@K) metrics only on the subset of translations that contain errors. Although R@K was considered the main metric for this task, we optimized internally for the average of all three metrics. Our internal results are shown in Table 7.1.

Discussion. The results highlight several contrasts between explanations for DA and MQM data: (i) while RemBERT is useful as an encoder for DA data (outperforms InfoXLM in 3 out of 5 LPs), it is outperformed by InfoXLM for all MQM LPs; (ii) the Head Mix component improves performance for DA, but it does not impact significantly the scores for MQM; and (iii) the Sparse Head Mix generally outperforms the Soft Head Mix for DA, but the trend flips for MQM. On what comes to the explainability methods, the baseline method (Attn × Norm – scaling the attention weights by the ℓ_2-norm of value vectors), which obtained the best results in the previous edition of the shared task (detailed in Chapter 6), is outperformed by our new method (Attn × GradNorm) for both DA and MQM data. Moreover, ensembling explanations from different heads brings further consistent improvements across the board for all LPs. For the zero-shot setting (*en-yo*), we build an ensemble of explanations by using the heads that were more common among the ensembles for all other LPs.

Explainability as word-level QE. Following the work by Fomicheva et al. (2020, 2022a) on exploiting inside information of a MT system to create an unsupervised approach to QE, we investigate the impact of using explainability scores to predict word-level tags. Concretely, for each MQM language pair, we use our top explainer to extract explanations scores $\boldsymbol{e}_i \in \mathbb{R}^L$ for all L tokens of the ith input example, and transform them

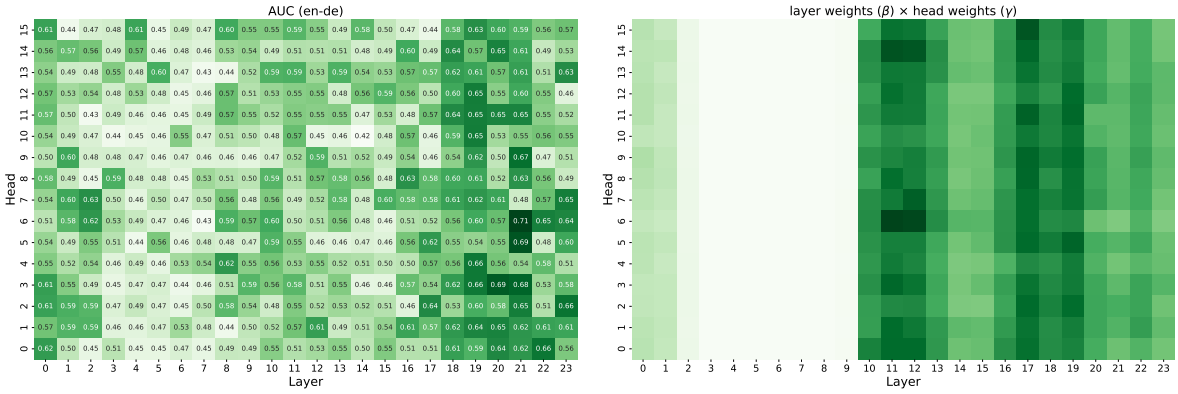


Figure 7.3: Investigation of attention heads from our InfoXLM model for the *en-de* language pair in terms of AUC scores (left) and sparse Head Mix coefficients $\beta_\ell \times \gamma_{\ell,h}$ (right).

to OK/BAD tags by tuning $\tau \in \mathbb{R}$ to maximize the following objective:

$$\frac{1}{N}\sum_{i=1}^{N} \text{MCC}(I[\boldsymbol{e}_i > \tau], \boldsymbol{y}_i), \tag{7.4}$$

where $\boldsymbol{y}_i \in \{0,1\}^L$ represents the ith ground-truth word-level tag sequence, and I is an indicator function applied element-wise. Once we find an optimal $\tau^\star$ on the dev set, we use it on test time to produce word-level predictions. We present results using this approach in Table 7.2 for all three MQM language pairs, comparing with COMETKIWI trained with word and sentence-level losses.[6] We note that thresholded explainability scores achieve competitive performance with COMETKIWI for *en-de* and *en-ru*, but lag behind for *zh-en* by a large margin. Surprisingly, we improve the results for *en-de* and *en-ru* by ensembling both approaches via a simple `OR` rule. These results suggest the potential applicability of explainability approaches for unsupervised QE, and demonstrate that explainability can be used to boost the performance of supervised word-level QE models.

System	en-de	en-ru	zh-en	avg.
COMETKIWI ($\hat{y}_A$)	0.2720	0.3279	**0.3676**	**0.3225**
Explainability ($\hat{y}_B$)	0.2550	0.3184	0.0787	0.2174
Both ($\hat{y}_A \vee \hat{y}_B$)	**0.2815**	**0.3418**	0.3381	0.3205

Table 7.2: Results for unsupervised word-level QE in terms of MCC.

7.5 Identifying Relevant Attention Heads

To circumvent the need of performing a time-consuming grid search to identify plausible attention heads, we designed the sparse Head Mix component to yield coefficients for each head representation. To better illustrate the benefits of this approach, we show the AUC scores obtained by each attention head for *en-de* alongside their Head Mix coefficients in Figure 7.3. In general, we note that heads from mid-up layers perform better while also deemed relevant by our sparse Head Mix module. In order to assess this concept more rigorously, we compute the Spearman's correlation between the AUC, AP, and R@K scores obtained by each head and the Head Mix coefficients. We present the results in Table 7.3. We note that the correlations are all positive, and for some

[6] Submitted to the shared task as part of this work (Rei et al., 2022b).

	Direct Assessment					MQM		
Metric	**en-cs**	**en-ja**	**en-mr**	**km-en**	**ps-en**	**en-de**	**en-ru**	**zh-en**
AUC	0.3344	0.2139	0.4014	0.5270	0.5173	0.5256	0.5091	0.4328
AP	0.3216	0.1144	0.3241	0.5428	0.5225	0.4717	0.4701	0.3413
R@K	0.3137	0.0910	0.3148	0.4700	0.5014	0.3954	0.4068	0.2101

Table 7.3: Spearman's correlation between plausibility scores obtained by each head (in terms of AUC, AP, and R@K) and the sparse Head Mix coefficients from our InfoXLM model.

language pairs they are above 0.5, suggesting a strong agreement. Overall, these results indicate that Head Mix coefficients can be successfully exploited to prune a large portion of the search space.

7.6 Official Results

We present the official results of our submissions alongside the results from other competitors in Table 7.4. Overall, we obtained the best results for all but two LPs (*km-en* and *ps-en*).

	Direct Assessment								MQM		
Team	**en-cs**	**en-ja**	**en-mr**	**en-yo**	**km-en**	**ps-en**	**all**	**all/yo**	**en-ru**	**en-de**	**zh-en**
Baseline: Random	0.363	0.336	0.167	0.144	0.565	0.614	0.365	0.409	0.135	0.124	0.093
Baseline: OpenKiwi+LIME	0.417	0.367	0.194	0.111	0.580	0.615	0.381	0.435	0.148	0.074	0.048
UT-QE (Azadi et al., 2022)	-	-	-	-	0.622	0.668	-	-	-	-	-
HW-TSC (Tao et al., 2022)	0.536	0.462	0.280	-	**0.686**	**0.715**	-	0.535	0.313	0.252	0.220
IST-Unbabel (*this work*)	**0.561**	**0.466**	**0.317**	**0.234**	0.665	0.672	**0.486**	**0.536**	**0.390**	**0.365**	**0.379**

Table 7.4: Official results in terms of R@K (Zerva et al., 2022).

7.7 Conclusions and Future Works

We presented the joint contribution of IST and Unbabel to the WMT 2022 Explainable QE shared task. By incorporating gradient information and designing a Head Mix component, we have refined the impact of attention heads towards the final prediction, leading to strong explainability performance and providing insights into the model's inner workings. In particular, our Head Mix component can be exploited to identify relevant attention heads at inference time, addressing the need for manual search from our previous method, detailed in Chapter 6. We have also found that explainability approaches can be applied as a form of unsupervised QE with a reasonably high accuracy when compared to a strong baseline, and may be used to further boost the performance of supervised word-level QE models. Overall, our submissions achieved the best official results for almost all LPs by a considerable margin.

One of the challenges in leveraging attention heads for deriving explanations is that they might not capture relevant information from other parts of the model. For future work we plan to explore approaches that aggregate attention weights from multiple layers, such as attention flows (Abnar and Zuidema, 2020), and that consider the entire layer block, such as ALTI (Ferrando et al., 2022). In the upcoming chapter, we discuss our final contribution to interpreting transformer-based QE models, where, instead of optimizing model performance, we focus on optimizing simulability, automatically identifying relevant attention heads in the process.

8

Learning to Scaffold: Optimizing Model Explanations for Quality Estimation

Contents

In this chapter, we bring together the contributions of multiple chapters in this thesis to propose a novel approach for improving the quality of explanations for transformer-based quality estimation (QE) models.

In Chapter 3, we laid the foundation for automatic simulability frameworks, which inspired Pruthi et al. (2022) to propose a new framework that rules out trivial protocols by design, rather than applying ad-hoc constraints on the explanation. We adopt their framework in this work to design a novel explainability method inspired by the success of attention-based methods, as detailed in Chapters 5, 6, and 7. Concretely, we propose an attention-based method that uses sparsity to automatically identify relevant attention heads in transformers. We achieve this by optimizing forward simulability, where a student model learns to simulate the predictions of a large teacher model. Our results show that students trained with explanations extracted from our new method are able to simulate the teacher model more effectively than those produced with previous approaches. Furthermore, through human annotations, we find that our learned explanations more closely align with how humans would explain the required decisions.

This chapter is based on (Fernandes et al., 2022), a work co-led with Patrick Fernandes. This chapter focuses on specific contributions made by the author of this thesis, which includes the co-design of the attention explainer and the experiments for QE. Nevertheless, it is worth to mention that one key contribution of this work—not covered in the chapter—is the formulation of the bi-level optimization problem, which we solve by borrowing ideas from the meta-learning literature and is detailed in our original paper.

8.1 Motivation

While deep learning's performance has led it to become the dominant paradigm in machine learning, its relative opaqueness has brought great interest in methods to improve *model interpretability*. Many recent works propose methods for extracting *explanations* from neural networks (§8.6), which vary from the highlighting of relevant input features (Simonyan et al., 2013; Arras et al., 2016; Ding et al., 2019) to more complex representations of the reasoning of the network (Mu and Andreas, 2020; Wu et al., 2021). However, are these methods actually achieving their goal of making models more interpretable? Some concerning findings have cast doubt on this proposition; different explanations methods have been found to disagree on the same model/input (Neely et al., 2021; Bastings et al., 2022) and explanations do not necessarily help predict a model's output and/or its failures (Chandrasekaran et al., 2018).

In fact, the research community is still in the process of understanding *what* explanations are supposed to achieve, and *how* to assess success of an explanation method (Doshi-Velez and Kim, 2017; Miller, 2019). Many early works on model interpretability designed their methods around a set of desiderata (Sundararajan et al., 2017; Lertvittayakumjorn and Toni, 2019) and relied on qualitative assessment of a handful of samples with respect to these desiderata; a process that is highly subjective and is hard to reproduce. In contrast, recent works have focused on more quantitative criteria: correlation between explainability methods for measuring *consistency* (Jain and Wallace, 2019; Serrano and Smith, 2019), *sufficiency* and *comprehensiveness* (DeYoung et al., 2020), and *simulability*: whether a human or machine consumer of explanations understands the model behavior well enough to predict its output on unseen examples (Lipton, 2018; Doshi-Velez and Kim, 2017). Simulability, in particular, has a number of desirable properties, such as being intuitively aligned with the goal of *communicating* the underlying model behavior to humans and being measurable in manual and automated

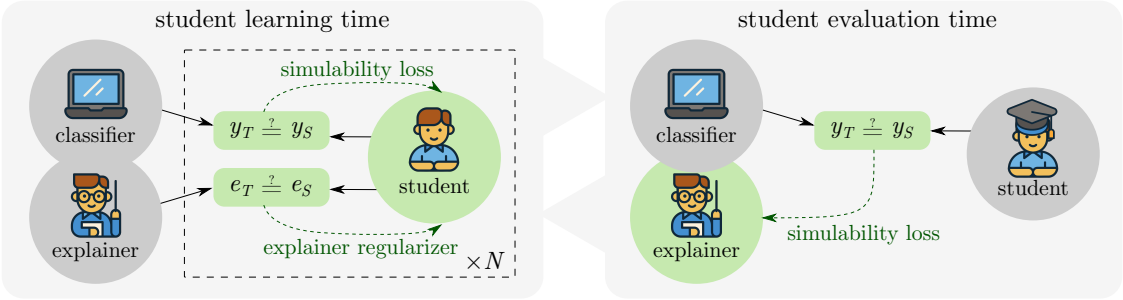


Figure 8.1: Illustration of our SMaT framework. First, a student model is trained to recover the classifier's predictions and to match the explanations given by the explainer. Then, the explainer is updated based on how well the trained student *simulates* the classifier (without access to explanations). In practice, we repeat these two consecutive processes for several steps. Green arrows and boxes represent learnable components.

experiments (Treviso and Martins, 2020; Hase and Bansal, 2020; Pruthi et al., 2022).

For instance, Pruthi et al. (2022) proposed a framework for automatic evaluation of simulability that, given a *teacher model* and explanations of this model's predictions, trains a *student model* to match the teacher's predictions. The explanations are then evaluated with respect to how well they help a student *learn to simulate* the teacher (§8.2). This is analogous to the concept in pedagogy of **instructional scaffolding** (Van de Pol et al., 2010), a process through which a teacher adds support for students to aid learning. More effective scaffolding—in our case, better explanations—is assumed to lead to better student learning. However, while this previous work provides an attractive way to *evaluate* existing explanation methods, it stops short of proposing a method to actually *improve* them.

In this work, we propose to *learn to explain* by directly learning explanations that provide better scaffolding of the student's learning, a framework we term ***S**caffold-**Ma**ximizing **T**raining* (**SMaT**). Figure 8.1 illustrates the framework: the explainer is used to *scaffold* the student training, and is updated based on how well the student does at *test* time at simulating the teacher model. We take insights from research on meta-learning (Finn et al., 2017; Raghu et al., 2021), formalizing our setting as a bi-level optimization problem and optimizing it based on higher-order differentiation (§8.3). Importantly, our high-level framework makes few assumptions about the model we are trying to explain, the structure of the explanations or the modalities considered. We then introduce a *parameterized* attention-based explainer optimizable with SMaT that works for any model with attention mechanisms (§8.4).

We experiment on 6 language pairs of the task of translation quality estimation (QE) using pretrained transformer models (§8.5). We find that our framework is able to effectively optimize explainers across all language pairs, where students trained with *learned* attention explanations achieve better simulability than baselines trained with *static* attention or gradient-based explanations. We further evaluate the *plausability* of our explanations (i.e., whether produced explanations align with how people would justify a similar choice) using human-labeled explanations and find that explanations learned with our learnable attention-based explainer are often more plausible than the static explainers considered. Overall, the results reinforce the utility of scaffolding as a criterion for evaluating and improving model explanations, and the effectiveness of attention-based methods for interpreting multilingual transformed-based models.

8.2 Background

Consider a model $T : \mathcal{X} \to \mathcal{Y}$ trained on some dataset $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{N}$. For example, this could be a text or image classifier that was trained on a particular downstream task (with $\mathcal{D}_{\text{train}}$ being the training data for that task). *Post-hoc* interpretability methods typically introduce an *explainer* module $E_T : \mathcal{T} \times \mathcal{X} \to \mathcal{E}$ that takes a model and an input, and produces an explanation $e \in \mathcal{E}$ for the output of the model given that input, where $\mathcal{E}$ denotes the space of possible explanations. For instance, interpretability methods using saliency maps define $\mathcal{E}$ as the space of *normalized* distributions of importance over L input elements $\boldsymbol{e} \in \triangle^L$ (where $\triangle^L$ is the $(L-1)$-probability simplex).

Pruthi et al. (2022) proposed an automatic framework for evaluating explainers that trains a *student* model $S_\theta : \mathcal{X} \to \mathcal{Y}$ with parameters θ to *simulate* the *teacher* (i.e., the original classifier) in a *constrained* setting. For example, the student can be constrained to have less capacity than the teacher by using a simpler model or trained with a subset of the dataset used for the teacher ($\hat{\mathcal{D}}_{\text{train}} \subsetneq \mathcal{D}_{\text{train}}$).

In this framework, a student S_θ is trained according to $\theta^* = \operatorname{argmin}_\theta \mathbb{E}_{(x,y)\sim\hat{\mathcal{D}}_{\text{train}}} [\mathcal{L}_{\text{sim}}(S_\theta(x), T(x))]$, and its simulability $\text{SIM}(S_{\theta^*}, T)$ is measured on an unseen test set. The actual form of $\mathcal{L}_{\text{sim}}$ and $\text{SIM}(S_{\theta^*}, T)$ is task-specific. For example, in a classification task, we use cross-entropy as the simulation loss $\mathcal{L}_{\text{sim}}$ over the teacher's predictions, while the simulability of a model S_{θ^*} can be defined as the simulation accuracy, i.e., what percentage of the student and teacher predictions match over a *held-out* test set $\mathcal{D}_{\text{test}}$:

$$\text{SIM}(S_{\theta^*}, T) = \mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{test}}}[\mathbb{1}\{S_{\theta^*}(x) = T(x)\}]. \tag{8.1}$$

Next, the training of the student is augmented with explanations produced by the explainer E. We introduce a student explainer $E_S : \mathcal{S} \times \mathcal{X} \to \mathcal{E}$, (the S-explainer) to extract explanations from the student, and *regularizing* these explanations on the explanations of teacher (the T-explainer), using a loss $\mathcal{L}_{\text{expl}}$ that takes explanations for both models:

$$\theta_E^* = \operatorname*{argmin}_\theta \mathbb{E}_{(x,y)\sim\hat{\mathcal{D}}_{\text{train}}} \Big[\underbrace{\mathcal{L}_{\text{sim}}\left(S_\theta(x), T(x)\right)}_{\text{simulability loss}} + \beta \underbrace{\mathcal{L}_{\text{expl}}\left(E_S(S_\theta, x), E_T(T, x)\right)}_{\text{explainer regularizer}} \Big]. \tag{8.2}$$

For example, Pruthi et al. (2022) considered as a teacher explainer E_T various methods such as LIME (Ribeiro et al., 2016), Integrated Gradients (Sundararajan et al., 2017), and attention mechanisms, and explored both attention regularization (using Kullback-Leibler divergence) and multi-task learning to regularize the student.

The key assumption surrounding this evaluation framework is that a student trained with *good* explanations should learn to simulate the teacher better than a student trained with bad or no explanations, that is, $\text{SIM}\left(S_{\theta_E^*}, T\right) > \text{SIM}\left(S_{\theta^*}, T\right)$. For clarity, we will refer to the simulability of a model $S_{\theta_E^*}$ trained using explanations as *scaffolded* simulability.

8.3 Optimizing Explainers for Teaching

In this work, we extend the previously described framework to make it possible to directly optimize the teacher explainer so that it can most effectively teach the student the original model's behavior. To this end, we

consider a *parameterized* T-explainer E_{ϕ_T} with parameters ϕ_T, and equivalently a *parameterized* S-explainer E_{ϕ_S} with parameters ϕ_S, leading to the following loss function for learning the student:

$$\mathcal{L}_{\text{student}}(S_\theta, E_{\phi_S}, T, E_{\phi_T}, x) = \mathcal{L}_{\text{sim}}\big(S_\theta(x), T(x)\big) + \beta\mathcal{L}_{\text{expl}}\big(E_{\phi_S}(S_\theta, x), E_{\phi_T}(T, x)\big). \tag{8.3}$$

While this framework is flexible enough to rigorously and automatically evaluate many types of explanations, calculating scaffolded simulability requires an optimization procedure to learn the student and S-explainer parameters θ, ϕ_S alongside finding the T-explainer parameters ϕ_T that optimize scaffolded simulability. To overcome this challenge, we draw inspiration from the extensive literature on meta-learning (Schmidhuber, 1987; Finn et al., 2017), and frame the optimization as a bi-level optimization problem, with an inner step for updating the student's parameters, θ, ϕ_S, and an outer update step for the teacher's parameters, ϕ_T. Further details on the optimization procedure can be found in (Fernandes et al., 2022).

8.4 Parameterized Attention Explainer

As a **key contribution** of this work, we introduce a novel *parameterized* attention-based explainer that can be learned with our framework. Transformer models (Vaswani et al., 2017) are currently the most successful deep-learning architecture across a variety of tasks (Shoeybi et al., 2019; Wortsman et al., 2022). Underpinning their success is the *multi-head attention mechanism*, which computes a *normalized* distribution over the $1 \leq i \leq L$ input elements in parallel for each head h:

$$\boldsymbol{A}^h = \mathsf{softmax}(\boldsymbol{Q}^h(\boldsymbol{K}^h)^\top), \tag{8.4}$$

where $\boldsymbol{Q}^h = [\boldsymbol{q}_1^h, \cdots, \boldsymbol{q}_L^h]$ and $\boldsymbol{K}^h = [\boldsymbol{k}_1^h, \cdots, \boldsymbol{k}_L^h]$ are the *query* and *key* linear projections over the input element representations for head h. Attention mechanisms have been used extensively for producing saliency maps (Wiegreffe and Pinter, 2019; Vashishth et al., 2019) and while some concerns have been raised regarding their faithfulness (Jain and Wallace, 2019), overall attention-based explainers have been found to lead to relatively good explanations in terms of *plausibility* and *simulability* (Treviso and Martins, 2020; Kobayashi et al., 2020; Pruthi et al., 2022).

However, to extract explanations from multi-head attention, we have two important design choices:

1. **Single distribution selection:** Since self-attention produces an attention matrix $\boldsymbol{A}^h \in \mathbb{R}^{L\times L}$, we need to *pool* these attention distributions to produce a single saliency map $\boldsymbol{e} \in \triangle^L$. Typically, the distribution from a single token (such as `[CLS]`) or the *average* of the attention distributions from all tokens $1 \leq i \leq L$ are used.

2. **Head selection:** We also need to *pool* the distributions produced by each head. Typical ad-hoc strategies include using the mean over all heads for a certain layer (Fomicheva et al., 2022a) or selecting a single head based on plausibility on validation set (Treviso et al., 2021). However, since transformers can have hundreds or even thousands of heads, these choices rely on human intuition or require large amounts of plausibility labels.

In this work, we approach the latter design choice in a more principled manner. Concretely, we associate each head with a weight and then perform a weighted sum over all heads. These weights are learned such

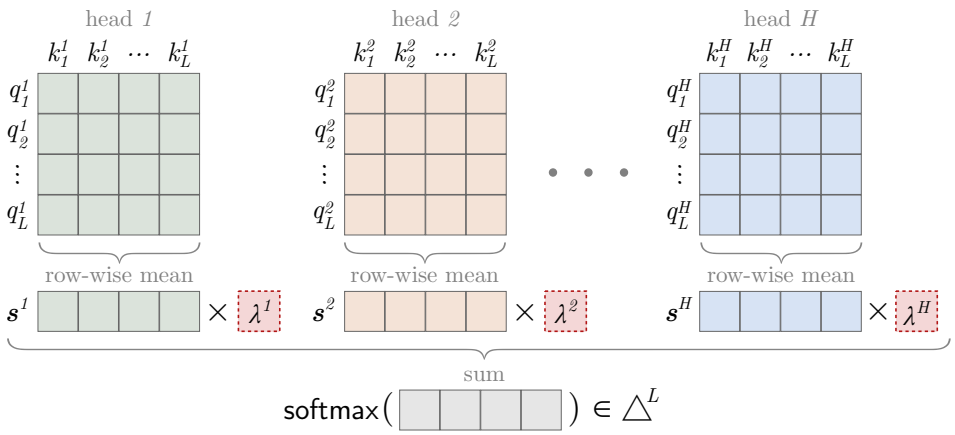


Figure 8.2: Our parameterized attention-based explainer. Dashed red boxes represent learned parameters $\boldsymbol{\lambda}_T = \mathsf{sparsemax}(\boldsymbol{\phi}_T) \in \triangle^H$, weighting average attention logits of each head $1 \leq h \leq H$. A softmax over the weighted sum generates the attention probabilities.

that the resulting explanation maximizes simulability, as described in §8.3. More formally, given a model T_{θ_T} and its query and key projections for an input x for each layer and head $h \leq H$, we define a *parameterized*, *differentiable* attention explainer E_{ϕ_T} as

$$\boldsymbol{s}^h = \frac{1}{L}\sum_{i=1}^{L}(\boldsymbol{q}_i^h)^\top \boldsymbol{K}^h, \qquad E_{\phi_T}(T, x) = \mathsf{softmax}\left(\sum_{h=1}^{H} \lambda_T^h \boldsymbol{s}^h\right), \tag{8.5}$$

where the teacher's head coefficients $\boldsymbol{\lambda}_T \in \triangle^H$ are defined as $\boldsymbol{\lambda}_T = \mathsf{normalize}(\boldsymbol{\phi}_T)$ with $\boldsymbol{\phi}_T \in \mathbb{R}^H$. Figure 8.2 illustrates each step of our parameterized attention explainer.

In this formulation, $\boldsymbol{s}^h \in \mathbb{R}^L$ represents the average *unnormalized attention logits* over all input elements, which are then combined according to $\boldsymbol{\lambda}_T$ and normalized with softmax to produce a distribution in $\triangle^L$. We apply a normalization function to head coefficients involved to create a *convex* combination over all heads in all layers. In this work we consider the sparse projection function $\mathsf{normalize} := \mathsf{sparsemax}$ (Martins and Astudillo, 2016) due to its benefits in terms of interpretability, since it leads to many heads having zero weight.

8.5 Experiments

We use JAX (Bradbury et al., 2018) to implement the higher-order differentiation, and use pretrained transformer models from the Huggingface Transformers library (Wolf et al., 2020), together with Flax (Heek et al., 2020). We train the teacher model with AdamW (Loshchilov and Hutter, 2019), and we train the student model with simple SGD updates (inner loop). We also use scalar mixing (Peters et al., 2018b) to pool representations from different layers automatically.[1] We train students with a teacher explainer in three settings:

- **No Explainer**: No explanations are provided, and no explanation regularization is used for training the student (i.e. $\beta = 0$ in Equation 8.3). We refer to students in this setting as **baseline** students.
- **Static Explainer**: Explanations for the teacher model are extracted with five commonly-used saliency-based explainers: (1) L2 norm of gradients; (2) a *gradient* $\times$ *input* explainer (Denil et al., 2014); (3) an *integrated gradients* explainer (Sundararajan et al., 2017); and *attention* explainers that uses the *mean*

[1] While scalar mixing reduced variance of student performance, SMaT also worked with other common pooling methods.

	2,100	4,200	8,400
No Explainer	.7457 [.7366:.7528]	.7719 [.7660:.7802]	.7891 [.7860:.7964]
Gradient L2	.8065 [.8038:.8268]	.8535 [.7117:.8544]	**.8638** [.8411:.8657]
Gradient × Input	.6846 [.6781:.6894]	.6922 [.6885:.6965]	.7141 [.7136:.7147]
Integrated gradients	.6686 [.6677:.6694]	.7086 [.6994:.7101]	.7036 [.6976:.7037]
Attention (*all layers*)	.8120 [.7955:.8125]	.8193 [.8186:.8280]	**.8467** [.8464:.8521]
Attention (*last layer*)	.7486 [.7484:.7534]	.7720 [.7672:.7726]	.7798 [.7717:.7814]
Attention (**learned**)	**.8156** [.8096:.8183]	**.8630** [.8412:.8724]	**.8561** [.8512:.8689]

Table 8.1: *Simulability* results, in terms of Pearson correlation, on the ML-QE dataset. *Underlined* values represent better performance than baseline with non-overlapping IQR.

pooling over attention from (4) all heads in the model and (5) from the heads of the last layer (Fomicheva et al., 2022a; Vafa et al., 2021). More details can be found in §F.1.

- **Learned Explainer**: Explanations are extracted with the explainer described in §8.4, with coefficients for each head that are trained with SMaT jointly with the student. We initialize the coefficients such that the model is initialized to be the same as the *static* attention explainer (i.e., performing the mean over all heads).

Independently of the T-explainer, we always use a learned attention-based explainer as the S-explainer, considering all heads except when the T-explainer is a static attention explainer that only considers the last layers' heads, where we do the same for the S-explainer. We use the Kullback-Leibler divergence as $\mathcal{L}_{\text{expl}}$, and we set $\beta = 5$ for attention-based explainers and $\beta = 0.2$ for gradient-based explainers (since we found smaller values to be better). We set $\mathcal{L}_{\text{sim}}$ as the mean squared error loss.

Data and evaluation. QE is the task of predicting a quality score given a sentence in a source language and a translation in a target language from a machine translation system. Interpreting quality scores of machine translated outputs is a problem that has received recent interest (Fomicheva et al., 2021) since it allows identifying which words were responsible for a bad translation. We use the MLQE-PE dataset (Fomicheva et al., 2022b), which contains 7,000 training samples for each of seven language pairs alongside word-level human annotation. We use as the base model a pretrained XLM-R-base (Conneau et al., 2020), a multilingual model with 12 layers and 12 heads in each (total of 144 heads).

We exclude one of the language pairs in the dataset (`si-en`) since the XLM-R model did not support it, leading to a training set with 42,000 samples. We reuse the same training set for both the teacher and student, sampling a subset for the latter. We vary the number of samples the student is trained with between 2,100 (5%), 4,200 (10%) and 8,400 (20%). Since this is a regression task, we evaluate simulability using the Pearson correlation coefficient between student and teacher's predictions.[2] The teacher achieves 0.63 correlation on the test set. For each setting, we train five students with different seeds. Since there is some variance in students' performance (we hypothesize due to the small training sets) we report the **median** and **interquantile range (IQR)** around it (relative to the 25-75 percentile).

[2]Pearson correlation is the standard metric used to evaluate sentence-level QE models.

	EN-DE		EN-ZH		ET-EN		NE-EN		RO-EN		RU-EN		OVERALL	
	src.	tgt.	src.	tgt.	src.	tgt.	src.	tgt.	src.	tgt.	src.	tgt.	src.	tgt.
Gradient L2	**0.64**	**0.65**	0.65	0.49	**0.67**	0.61	**0.68**	**0.55**	**0.72**	0.68	**0.65**	0.54	**0.67**	0.59
Gradient × Input	0.58	0.60	0.61	0.51	0.60	0.54	0.61	0.49	0.64	0.59	0.58	0.51	0.61	0.54
Integrated Gradients	0.59	0.60	0.63	0.49	0.60	0.52	0.64	0.48	0.64	0.59	0.60	0.51	0.62	0.53
Attention (*all layers*)	0.60	0.63	**0.68**	**0.52**	0.60	0.61	0.58	**0.55**	0.66	**0.70**	0.62	**0.55**	0.62	0.59
Attention (*last layer*)	0.51	0.49	0.61	0.49	0.51	0.50	0.55	0.48	0.52	0.57	0.56	0.50	0.54	0.50
Attention (**learned**)	**0.64**	**0.65**	**0.68**	**0.52**	0.66	**0.64**	0.66	0.54	0.71	**0.70**	0.61	0.54	0.66	**0.60**
Attention (*best layer*)*	0.64	0.65	0.69	0.64	0.64	0.68	0.68	0.68	0.71	0.76	0.64	0.59	0.65	0.65
Attention (*best head*)*	0.67	0.67	0.70	0.65	0.70	0.70	0.70	0.69	0.73	0.75	0.67	0.60	0.67	0.66

Table 8.2: Plausibility results for source and target inputs for each language pair of the MLQE-PE dataset in terms of AUC. * represents *supervised* methods that use human labels in some form.

Simulability results. Table 8.1 shows the results for the three settings. Similar to other tasks, the attention explainer trained with SMaT leads to students with higher simulability than baseline students and similar or higher than *static* explainer across all training set sizes. Curiously, the *Grad. L2* explainer achieves very high simulability for this task. It even has a higher *median* simulability score than SMaT for 8,400 samples. However, we attribute this to variance in the student training set sampling (that could lead to an imbalance in language pair proportions) which could explain why SMaT performance degrades with more samples. For this task, the gradient-based explainers always degrade simulability across the tested training set size. It also seems that using only the last layer's attention is also ineffective at teaching students, achieving the same performance as the baseline.

Plausibility analysis. We select the median model trained with 4,200 samples and follow the approach devised in the Explainable QE shared task to evaluate plausibility (Fomicheva et al., 2021), which consists of evaluating the human-likeness of explanations in terms of AUC only on the subset of translations that contain errors. The results are shown in Table 8.2. We note that for all language pairs, our learned explainer performs on par or better than static explainers, and only being surpassed by *Grad. L2* in the *source-side* over all languages. Comparing with the best attention layer/head, an approach used by Fomicheva et al. (2022a); Treviso et al. (2021), our explainer achieves similar AUC scores for source explanations, but lags behind the best attention layer/head for target explanations on *-EN language pairs. However, our approach sidesteps human annotation and avoids the cumbersome approach of independently computing plausibility scores for all heads.

Head coefficients. A key benefit of sparsemax is that it produces a small subset of *active* heads. The heatmaps of attention coefficients ($\boldsymbol{\lambda}_T$) learned after training, shown in Figure 8.3 (left), exemplify this. We can see that all heads of the first layer are *active* ($\lambda_T^h > 0$), whereas the rest active heads are spread throughout mid-up layers. We also found empirically that active heads are usually associated with attention heads that lead to top plausibility scores in Figure 8.3 (right), further reinforcing our good plausibility findings. To better quantify this notion, we computed Spearman's correlation results between our learned head coefficients and the AUC scores obtained by each head. Moreover, as suggested by Treviso et al. (2021), heads from mid-up layers are usually more plausible for QE. Thus, we computed correlations with and without the heads from the first layer, which were all deemed active by SMaT. We present the results in Table 8.3 for all language pairs. Except for *en-zh*,

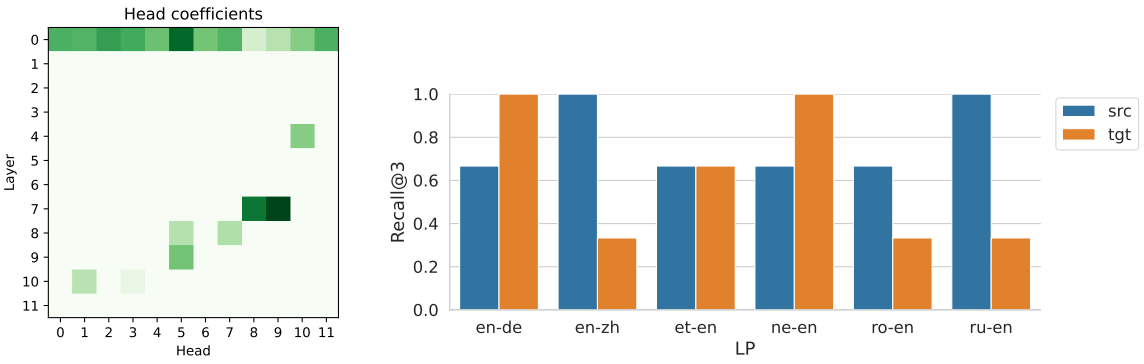


Figure 8.3: Head coefficients learned by the model, illustrating that only a small subset of attention heads are deemed relevant (left), and the recall at recovering the 3 most plausible attention heads from this subset (right).

SIDE	EN-DE	EN-ZH	ET-EN	NE-EN	RO-EN	RU-EN
Source (w/ 1st layer)	0.1039	0.1034	0.1660	0.2403	0.1643	-0.1017
Source (w/o 1st layer)	0.3039	-0.2662	0.2786	0.2841	0.2598	0.3023
Target (w/ 1st layer)	0.0812	0.0722	-0.0052	-0.0697	0.0362	-0.1052
Target (w/o 1st layer)	0.2996	-0.0868	0.2939	0.2974	0.3108	0.2515

Table 8.3: Spearman's correlation between AUC scores obtained by each head and the head coefficients learned by our explainer.

our findings indicate that the head coefficients of mid-up layers learned by SMaT exhibit a positive correlation with plausibility scores, typically falling within the range of $[0.2, 0.3]$. These findings suggest that examining the learned coefficients can serve as an effective approach for identifying plausible attention heads without the need for manual search.

8.6 Related Work

Explainability for text. Several works propose explainability methods to interpret decisions made by NLP models. Besides gradient and attention-based approaches already mentioned, some extract explanations by running the models with perturbed inputs (Ribeiro et al., 2016; Feng et al., 2018; Kim et al., 2020). Others even define custom backward passes to assign relevance for each feature (Bach et al., 2015). These methods are commonly employed together with post-processing heuristics, such as selecting only the top-k tokens with higher scores for visualization. Another line of work seeks to build a classifier with inherently interpretable components, such as methods based on attention mechanisms and rationalizers (Lei et al., 2016; Bastings et al., 2019).

Evaluation of explainability methods. As mentioned in the introduction, early works evaluated explanations based on properties such as *consistency*, *sufficiency* and *comprehensiveness*. Jacovi and Goldberg (2020) recommended the use of a graded notion of faithfulness, which the ERASER benchmark quantifies using the idea of sufficient and comprehensive rationales, alongside compiling datasets with human-annotated rationales for calculating plausibility metrics (DeYoung et al., 2020). Given the disagreement between explainability methods, Neely et al. (2021) showed that without a faithful ground-truth explanation it is impossible to determine which method is better. Diagnostic tests such as the ones proposed by Adebayo et al. (2018); Wiegreffe and

Pinter (2019) and Atanasova et al. (2020) are more informative yet they do not capture the main goal of an explanation: the ability to communicate an explanation to a practitioner.

Simulability. A new dimension for evaluating explainability methods relies on the forward simulation proposed by Lipton (2018) and Doshi-Velez and Kim (2017), which states that humans should be able to correctly simulate the model's output given the input and the explanation. Chandrasekaran et al. (2018); Hase and Bansal (2020); Arora et al. (2022) analyze simulability via human studies across text classification datasets. Treviso and Martins (2020) designed an automatic framework where students (machine or human) have to predict the model's output given an explanation as input. Similarly, Pruthi et al. (2022) proposed the simulability framework that was extended in our work, where explanations are used to regularize the student rather than passed as input.

Learning to explain. The concept of simulability also opens a path to learning explainers. In particular Treviso and Martins (2020) learn an attention-based explainer that maximizes simulability. However, directly optimizing for simulability sometimes led to explainers that learned trivial protocols (such as selecting only punctuation symbols or stopwords to leak the label). Our approach of optimizing a teacher-student framework is similar to approaches that optimize for model distillation (Zhou et al., 2022). However, these approaches modify the original model rather than introduce a new explainer module. Raghu et al. (2021) propose a framework similar to ours for learning *commentaries* for inputs that speed up and improve the training of a model. However commentaries are model-independent and are optimised to improve performance on the real task. Rationalizers (Chen et al., 2018; Jacovi and Goldberg, 2021; Guerreiro and Martins, 2021) also directly learn to extract explanations, but can also suffer from trivial protocols.

8.7 Conclusion and Future Works

We proposed a framework for directly optimizing explanations of the model's predictions to improve the training of a student *simulating* the said model. Concretely, to this end, we introduced a parameterized attention-based explainer that is optimizable by our framework. By experimenting on QE, we found that explanations learned with our explainer both lead to students that simulate the original model more accurately and are more aligned with how people explain similar decisions when compared to previously proposed methods. On top of that, our parameterized attention explainer provides a principled way for discovering relevant attention heads in transformers.

We only explored learning attention-based explainers, but our method can also be used to optimize other types of explainability methods, including gradient-based ones, by introducing learnable parameters in their formulations. Another promising future research direction is to explore using SMaT to learn explanations other than saliency maps, such as free-text explanations produced by large language models (Yordanov et al., 2022; Ross et al., 2022a).

9

Conclusions

Contents

In this concluding chapter, we provide a succinct summary of the key findings and contributions made in this thesis. Additionally, we examine open problems alongside identifying promising paths for subsequent research.

9.1 Summary of Contributions

This thesis explored the role of simulability and sparsity to improve the interpretability of neural networks, particularly in the context of quality estimation (QE) for machine translation. In Chapter 3, we designed a flexible framework for evaluating explainability methods in terms of forward simulability (Doshi-Velez and Kim, 2017) in an automatic way, allowing us to compare several approaches under a single, human-aligned perspective. Building on this framework, we introduced a new explainability method that leverages sparse attention to maximize simulability while also leading to plausible explanations.

Chapter 4 introduced CREST, a framework that incorporates learnable sparse signals to guide the generation of synthetic counterfactuals using masked language models, and subsequently leverages these counterfactuals to regularize selective rationales. We conducted a rigorous evaluation and showed that CREST counterfactuals often lead to improvements in terms of model robustness and rationale quality. With the counterfactual generator in place, we also proposed an automatic approach to evaluate explainability methods based on counterfactual simulability, revealing that our approach is especially effective in learning contrastive behavior.

We presented *Sparsefinder* in Chapter 5, an efficient and interpretable alternative to the multi-head attention mechanism found in transformers. This novel approach trains a compact student model to predict the sparse attention patterns of a larger teacher model trained with α-entmax attention, effectively reducing computation time when compared to a vanilla transformer. Notably, while *Sparsefinder* seeks to reduce computational cost, it also designed to preserve the interpretable behavior of learned attention heads, standing out from related approaches.

In the second part of this thesis, in Chapter 6, we collaborated with the Unbabel AI team to participate in the Explainable Quality Estimation Shared Task. Driven by prior research on connecting explainability with word-level QE (Fomicheva et al., 2022a), we investigated various explainability methods, such as gradient, erasure, attention, and rationalization approaches, while also proposing new ways to interpret decisions made by modern QE models based on pretrained transformers. In particular, we explored the role of single attention heads in identifying word-level translation errors, leading to winning submissions in nearly all language pairs.

Building on these findings, more plausible and practical explainers that leverage sparsity were developed in Chapter 7. More precisely, we addressed the weakness of manual search of attention heads by designing a sparse bottleneck layer called *Sparse Head Mix*, which aggregates hidden states from selected attention heads for sentence-level predictions. By combining attention with gradient information and leveraging coefficients from Sparse Head Mix, we automatically identified relevant attention heads, improving interpretability and alleviating manual search. These innovations heavily contributed to our success in winning the shared task for 7 out of 9 language pairs.

In Chapter 8, we integrated the contributions from multiple chapters to propose a novel approach for improving explanations for transformer-based QE models. Building on the foundations of automatic simulability, we developed an attention-based method that uses sparsity to automatically identify relevant attention heads in transformers by optimizing forward simulability. We showed that our explainer not only learns to maximize simulability, but also produces explanations that better align with human intuition.

In summary, we have found that simulability serves as a valuable tool for evaluating and designing more plausible and robust explainers, while sparsity can be an important factor for improving the explainability of transformer-based models. Our empirical evaluations reveal that attention-based methods often outperform other approaches for explaining QE models, and that sparsity can be effectively employed to identify relevant inner components of the model, such as attention heads, as well as identifying influential words in the input. These sparse signals not only guided the creation of efficient attention mechanisms, but also offered valuable information for counterfactual generation. Our successful strategies in this area led to winning submissions in two consecutive editions of the Explainable Quality Estimation Shared Task, in 2021 and 2022, further highlighting the relevance and effectiveness of our approaches.

9.2 Open Problems and Future Directions

The work presented in this thesis can serve as a base for future research on interpretability in neural networks, particularly in the context of the QE task. We believe that the insights gained from our investigation of simulability and sparsity can be applied to other tasks and domains, such as language modeling and image recognition, potentially broadening the impact of these techniques. In spite of this, many open problems and limitations remain.

Theoretical Foundation for Automatic Simulability. Despite our initial efforts in showing the feasibility and effectiveness of our automatic simulability framework in Chapter 3, a more comprehensive theory that clarifies its training dynamics, possibly generalizing to related frameworks, is currently missing. Such a theory could elucidate the emergence of trivial protocols and suggest potential solutions without resorting to ad-hoc strategies. A promising future direction on this line involves examining $\mathcal{V}$-information (Xu et al., 2020), a generalization of information theory to the case where agents are constrained to be in a function family $\mathcal{V}$. Importantly, we may constrain $\mathcal{V}$ to an arbitrary function space, making it possible to *produce* $\mathcal{V}$-information through data processing. This generalization contrasts with classical information theory, where the data processing inequality ensures that new information can never be produced through processing (Cover and Thomas, 2012). That is, for inputs X and outputs Y, and a certain $\mathcal{V}$, it is possible that

$$I_{\mathcal{V}}(h(X), Y) > I_{\mathcal{V}}(X, Y), \tag{9.1}$$

where h is a data processing function. This is well-suitable for our simulability framework, as the layperson can be conceptualized as the family class $\mathcal{V}$, and the explainer can be cast as the processing step h. In other words, this structure may offer a theoretical basis for learning explanations that are both informative and non-trivial within the context of the communication game.

Counterfactuals for QE. In Chapter 4 we presented a framework for generating textual counterfactuals, which we use for text classification and natural language inference. However, it remains an open problem how this framework can be used to generate counterfactuals for more complex tasks, such as QE, for which generating counterfactuals is not straightforward due to the asymmetry in the semantics of ground truth scores. That is, although transforming a good translation into a bad one is relatively simple, correcting a bad translation is more

demanding—a task known as Automatic Post-Editing. Large Language Models (LLMs) could potentially address this issue, as their ability to perform many multilingual tasks might enable them to develop the necessary capabilities for this problem, possibly with the guidance of sparse signals provided by explainability methods.

Effectively Exploiting Sparsity in Modern Hardware. Effectively exploiting sparsity to improve efficiency in neural networks is challenging, as it is often unclear beforehand which network components will be required for subsequent computations, and practical implementations tend to be reliant on specific hardware. Sparsefinder, presented in Chapter 5, represents an initial effort to leverage sparsity patterns to improve the efficiency of self-attention in transformers. Even though it enjoys subquadratic time complexity and runs faster than a vanilla transformer, it remains slower than alternative methods in practice due to hardware constraints. Nevertheless, given its flexibility, the potential applications of sparsity in large transformers are promising. A possible path is to leverage this adaptable property and empower sparsity to identify a subset of efficient attention kernels that can reproduce specific attention patterns in linear time, reducing the overall computational load when compared to materializing the standard quadratic attention matrix. To this end, we can consider a set of k efficient kernels $\mathcal{K}_1, ..., \mathcal{K}_k$ that receive query $\boldsymbol{Q} \in \mathbb{R}^{n \times d}$ and key $\boldsymbol{K} \in \mathbb{R}^{n \times d}$ matrices in order to produce a mixture of attention logits:

$$\begin{aligned} \boldsymbol{p} &= \alpha\text{-}\mathsf{entmax}(\boldsymbol{\theta}), \text{with } \boldsymbol{\theta} \in \mathbb{R}^k, \\ \boldsymbol{Z} &= p_1 \mathcal{K}_1(\boldsymbol{Q}, \boldsymbol{K}) + p_2 \mathcal{K}_2(\boldsymbol{Q}, \boldsymbol{K}) + \cdots + p_k \mathcal{K}_k(\boldsymbol{Q}, \boldsymbol{K}), \end{aligned}$$

which are then used to compute the regular attention operation, as defined in Eq. 2.6. In this way, a high efficiency can be achieved by only computing kernels for which $p_i > 0$, as long as $\|\boldsymbol{p}\|_0 \leq k \ll n$. Another avenue involves leveraging deterministic sparse transformations, such as α-entmax, to better manage instability issues during the training of mixture-of-experts models (Fedus et al., 2022), which struggle during backpropagation due to the differentiability properties of top-k operations (Zoph et al., 2022).

Explanations for Closed LLMs. With the rapid rise of LLMs in popularity, it is likely that they will be increasingly employed for a variety of multilingual tasks, including MT and QE, possibly involving the use of closed LLMs—models accessible only through APIs or interfaces without the ability to inspect or modify their underlying architecture. Not having access to the models' internal components directly affects the approaches proposed in this thesis, as nearly all of them depend on extracting information from built-in mechanisms, such as attention weights. An exception is our simulability framework proposed in Chapter 3, which is flexible and can be applied to learn *post-hoc* explainers without requiring the access to the inner workings of the teacher, allowing them to operate on top of closed LLMs. Therefore, an interesting direction is to cast the layperson (student) as a smaller LM, leading to a communication game between two LMs mediated by a learnable sparse explainer.

For instance, consider a large LM T that receives a prompt x as input and produces an outcome y, along with an explainer module E that constructs a message m for S, a smaller LM that tries to replicate T's outcome given the information conveyed by m. The communication between these three agents depends on the task prompted to T, which influences the format of its outcome (single or multiple tokens) and paves the way for numerous explainability directions:

- **Highlighting decisions**: If the outcome y is a single result (e.g., a quality assessment) or a sequence of tokens (e.g., a translation), the explainer E could be a differentiable attention module—as proposed in Chapter 3—that learns a distribution over the prompt tokens $\boldsymbol{x}$, guiding S towards the same decisions as T.

- **Highlighting rationales**: If the outcome includes a decision y and a rationale z (e.g., when asking the model to explain its own decision), the explainer E could also focus on specific parts of the rationale z, being cautious not to leak the label y to S. The communication part of the framework used in Chapter 8 is suitable in this case, as it only uses explanations during training to regularize S.

- **Producing free-text explanations**: In addition to highlighting input tokens, LLMs' self-rationalization capabilities can be employed to generate informative explanations. To this end, we could use the previous approaches to create a message m, and ask S to predict both the outcome y and the rationale z. This strategy resembles knowledge distillation (Kim and Rush, 2016), but differs in handling the message m: using it to both to regularize S and as a gold target.

- **Evaluating self-rationales**: Besides creating new explainers, we can also assess the quality of self-rationalization by examining the success of the communication.

Although highlights and free-text explanations are given as examples, our simulability framework can handle different message types, such as prototypes or structured explanations, leading to promising research avenues. Furthermore, this direction is linked to the theoretical foundation for simulability, which can assist in providing better theoretical guarantees prior to costly experiments.

Bibliography

Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4197, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.385. URL https://aclanthology.org/2020.acl-main.385.

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31, 2018.

David Alvarez-Melis and Tommi Jaakkola. A causal framework for explaining the predictions of black-box sequence-to-sequence models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 412–421, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1042. URL https://www.aclweb.org/anthology/D17-1042.

Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Sy21R9JAW.

Siddhant Arora, Danish Pruthi, Norman Sadeh, William Cohen, Zachary Lipton, and Graham Neubig. Explain, edit, and understand: Rethinking user study design for evaluating model explanations. In *Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI)*, Vancouver, Canada, February 2022. URL https://arxiv.org/abs/2112.09669.

Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. Explaining predictions of non-linear classifiers in NLP. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 1–7, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-1601. URL https://aclanthology.org/W16-1601.

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. A diagnostic study of explainability techniques for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3256–3274, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.263. URL https://aclanthology.org/2020.emnlp-main.263.

Fatemeh Azadi, Heshaam Faili, and Mohammad Javad Dousti. Mismatching-aware unsupervised translation quality estimation for low-resource languages. *arXiv preprint arXiv:2208.00463*, 2022. URL https://arxiv.org/abs/2208.00463.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):1–46, 07 2015. doi: 10.1371/journal.pone.0130140. URL https://doi.org/10.1371/journal.pone.0130140.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 2015 International Conference on Learning Representations (ICLR)*, 2015.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5301. URL https://aclanthology.org/W19-5301.

Jasmijn Bastings and Katja Filippova. The elephant in the interpretability room: Why use attention as explanation when we have saliency methods? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 149–155, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.blackboxnlp-1.14. URL https://aclanthology.org/2020.blackboxnlp-1.14.

Jasmijn Bastings, Wilker Aziz, and Ivan Titov. Interpretable neural predictions with differentiable binary variables. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2963–2977, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1284. URL https://aclanthology.org/P19-1284.

Jasmijn Bastings, Sebastian Ebert, Polina Zablotskaia, Anders Sandholm, and Katja Filippova. "will you find these shortcuts?" a protocol for evaluating the faithfulness of input salience methods for text classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 976–991, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.emnlp-main.64.

Aurélien Bellet, Amaury Habrard, and Marc Sebban. Metric learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 9(1):1–151, 2015.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv:2004.05150*, 2020. URL https://arxiv.org/abs/2004.05150.

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. Confidence estimation for machine translation. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 315–321, Geneva, Switzerland, aug 23–aug 27 2004. COLING. URL https://aclanthology.org/C04-1046.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL https://aclanthology.org/D15-1075.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. e-snli: Natural language inference with natural language explanations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 9539–9549. Curran Associates, Inc., 2018. URL http://papers.nips.cc/paper/8163-e-snli-natural-language-inference-with-natural-language-explanations.pdf.

Samuel Carton, Anirudh Rathore, and Chenhao Tan. Evaluating and characterizing human rationales. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9294–9307, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.747. URL https://aclanthology.org/2020.emnlp-main.747.

Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Niehues Jan, Stüker Sebastian, Sudoh Katsuitho, Yoshino Koichiro, and Federmann Christian. Overview of the iwslt 2017 evaluation campaign. In *Proceedings of the 14th International Workshop on Spoken Language Translation (IWSLT)*, pages 2–14, 2017.

Arjun Chandrasekaran, Viraj Prabhu, Deshraj Yadav, Prithvijit Chattopadhyay, and Devi Parikh. Do explanations make VQA models more predictable to a human? In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1036–1042, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1128. URL https://aclanthology.org/D18-1128.

Shiyu Chang, Yang Zhang, Mo Yu, and Tommi Jaakkola. Invariant rationalization. In *International Conference on Machine Learning*, pages 1448–1458. PMLR, 2020.

Howard Chen, Jacqueline He, Karthik Narasimhan, and Danqi Chen. Can rationalization improve robustness? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3792–3805, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.278. URL https://aclanthology.org/2022.naacl-main.278.

Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan. Learning to explain: An information-theoretic perspective on model interpretation. In *International Conference on Machine Learning*, pages 883–892. PMLR, 2018.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. InfoXLM: An information-theoretic framework for cross-lingual language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3576–3588, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.280. URL https://aclanthology.org/2021.naacl-main.280.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019. URL https://arxiv.org/abs/1904.10509.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar, October 2014a. Association for Computational Linguistics. doi: 10.3115/v1/W14-4012. URL https://aclanthology.org/W14-4012.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014b. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. URL https://aclanthology.org/D14-1179.

Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. Rethinking attention with performers. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://openreview.net/forum?id=Ua6zuk0WRH.

George Chrysostomou and Nikolaos Aletras. An empirical study on explanations in out-of-domain settings. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6920–6938, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.477. URL https://aclanthology.org/2022.acl-long.477.

Hyung Won Chung, Thibault Fevry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. Rethinking Embedding Coupling in Pre-trained Language Models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=xpFFI_NtgpW.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416v5*, 2022. URL https://arxiv.org/abs/2210.11416v5.

Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. In *27th Annual Meeting of the Association for Computational Linguistics*, pages 76–83, Vancouver, British

Columbia, Canada, June 1989. Association for Computational Linguistics. doi: 10.3115/981623.981633. URL https://www.aclweb.org/anthology/P89-1010.

Alexis Conneau and Guillaume Lample. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems*, 32:7059–7069, 2019.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.747. URL https://aclanthology.org/2020.acl-main.747.

Gonçalo M. Correia, Vlad Niculae, and André F. T. Martins. Adaptively sparse transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2174–2184, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1223. URL https://aclanthology.org/D19-1223.

Thomas M Cover and Joy A Thomas. *Elements of Information Theory*. John Wiley & Sons, 2012.

Renato Cordeiro de Amorim. Constrained clustering with minkowski weighted k-means. In *2012 IEEE 13th International Symposium on Computational Intelligence and Informatics (CINTI)*, pages 13–17. IEEE, 2012.

Nicola De Cao, Michael Sejr Schlichtkrull, Wilker Aziz, and Ivan Titov. How do decisions emerge across layers in neural models? interpretation with differentiable masking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3243–3255, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.262. URL https://aclanthology.org/2020.emnlp-main.262.

Misha Denil, Alban Demiraj, and Nando de Freitas. Extraction of salient sentences from labelled documents. *ArXiv*, abs/1412.6815, 2014.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.408. URL https://aclanthology.org/2020.acl-main.408.

Shuoyang Ding, Hainan Xu, and Philipp Koehn. Saliency-driven word alignment interpretation for neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 1–12, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5201. URL https://aclanthology.org/W19-5201.

Tanay Dixit, Bhargavi Paranjape, Hannaneh Hajishirzi, and Luke Zettlemoyer. CORE: A retrieve-then-edit framework for counterfactual data generation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2964–2984, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.findings-emnlp.216.

Finale Doshi-Velez and Been Kim. Towards A Rigorous Science of Interpretable Machine Learning. *arXiv preprint arXiv:1702.08608*, 2017. URL http://arxiv.org/abs/1702.08608.

Melda Eksi, Erik Gelbing, Jonathan Stieber, and Chi Viet Vu. Explaining errors in machine translation with absolute gradient ensembles. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 238–249, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eval4nlp-1.23. URL https://aclanthology.org/2021.eval4nlp-1.23.

Miquel Esplà, Mikel Forcada, Gema Ramírez-Sánchez, and Hieu Hoang. ParaCrawl: Web-scale parallel corpora for the languages of the EU. In *Proceedings of Machine Translation Summit XVII: Translator, Project and User Tracks*, pages 118–119, Dublin, Ireland, August 2019. European Association for Machine Translation. URL https://aclanthology.org/W19-6721.

Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah A. Smith. Sparse overcomplete word vector representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1491–1500, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1144. URL https://www.aclweb.org/anthology/P15-1144.

William Fedus, Barret Zoph, and Noam Shazeer. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022. URL http://jmlr.org/papers/v23/21-0998.html.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. Pathologies of neural models make interpretations difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1407. URL https://www.aclweb.org/anthology/D18-1407.

Patrick Fernandes, Kayo Yin, Graham Neubig, and André F. T. Martins. Measuring and increasing context usage in context-aware machine translation. In *Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, Virtual, August 2021. URL https://arxiv.org/abs/2105.03482.

Patrick Fernandes, Marcos Treviso, Danish Pruthi, Andre Martins, and Graham Neubig. Learning to scaffold: Optimizing model explanations for teaching. In *Thirty-Sixth Conference on Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=V5rlSPsHpkf.

Javier Ferrando, Gerard I. Gállego, and Marta R. Costa-jussà. Measuring the mixing of contextual information in the transformer. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8698–8714, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.emnlp-main.595.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/finn17a.html.

Jakob Foerster, Ioannis Alexandros Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2137–2145. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6042-learning-to-communicate-with-deep-multi-agent-reinforcement-learning.pdf.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555, 2020. doi: 10.1162/tacl_a_00330. URL https://aclanthology.org/2020.tacl-1.35.

Marina Fomicheva, Piyawat Lertvittayakumjorn, Wei Zhao, Steffen Eger, and Yang Gao. The Eval4NLP shared task on explainable quality estimation: Overview and results. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 165–178, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eval4nlp-1.17. URL https://aclanthology.org/2021.eval4nlp-1.17.

Marina Fomicheva, Lucia Specia, and Nikolaos Aletras. Translation error detection as rationale extraction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 4148–4159, Dublin, Ireland, May 2022a. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.327. URL https://aclanthology.org/2022.findings-acl.327.

Marina Fomicheva, Shuo Sun, Erick Fonseca, Chrysoula Zerva, Frédéric Blain, Vishrav Chaudhary, Francisco Guzmán, Nina Lopatina, Lucia Specia, and André F. T. Martins. MLQE-PE: A multilingual quality estimation and post-editing dataset. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4963–4974, Marseille, France, June 2022b. European Language Resources Association. URL https://aclanthology.org/2022.lrec-1.530.

Erick Fonseca, Lisa Yankovskaya, André F. T. Martins, Mark Fishel, and Christian Federmann. Findings of the WMT 2019 shared tasks on quality estimation. In *Proceedings of the Fourth Conference on Machine*

Translation (Volume 3: Shared Task Papers, Day 2), pages 1–10, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5401. URL https://aclanthology.org/W19-5401.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.117. URL https://aclanthology.org/2020.findings-emnlp.117.

L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 80–89, 2018.

Max Glockner, Vered Shwartz, and Yoav Goldberg. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2103. URL https://aclanthology.org/P18-2103.

Taisiya Glushkova, Chrysoula Zerva, Ricardo Rei, and André F. T. Martins. Uncertainty-aware machine translation evaluation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3920–3938, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.330. URL https://aclanthology.org/2021.findings-emnlp.330.

Yoav Goldberg and Graeme Hirst. *Neural Network Methods in Natural Language Processing*. Morgan & Claypool Publishers, 2017. ISBN 1627052984.

João Graça. Unbabel: how to combine ai with the crowd to scale professional-quality translation. In *Proceedings of the AMTA 2018 Workshop on Translation Quality Estimation and Automatic Post-Editing*, pages 41–85, 2018.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. Continuous measurement scales in human evaluation of machine translation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 33–41, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL https://aclanthology.org/W13-2305.

Christopher Grimsley, Elijah Mayfield, and Julia R.S. Bursten. Why attention is not explanation: Surgical intervention and causal reasoning about neural models. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1780–1790, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.220.

Nuno M. Guerreiro and André F. T. Martins. SPECTRA: Sparse structured text rationalization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6534–6550, Online

and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.525. URL https://aclanthology.org/2021.emnlp-main.525.

Nitish Gupta, Sameer Singh, Matt Gardner, and Dan Roth. Paired examples as indirect supervision in latent decision models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5774–5785, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.466. URL https://aclanthology.org/2021.emnlp-main.466.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2017. URL https://aclanthology.org/N18-2017.

Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3(null):1157–1182, March 2003. ISSN 1532-4435.

Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1–3):389–422, March 2002. ISSN 0885-6125. doi: 10.1023/A:1012487302797. URL https://doi.org/10.1023/A:1012487302797.

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. Self-attention attribution: Interpreting information interactions inside transformer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12963–12971, May 2021. doi: 10.1609/aaai.v35i14.17533. URL https://ojs.aaai.org/index.php/AAAI/article/view/17533.

Peter Hase and Mohit Bansal. Evaluating explainable AI: Which algorithmic explanations help users predict model behavior? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5540–5552, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.491. URL https://aclanthology.org/2020.acl-main.491.

Peter Hase, Shiyue Zhang, Harry Xie, and Mohit Bansal. Leakage-adjusted simulatability: Can models generate non-trivial explanations of their behavior in natural language? In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4351–4367, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.390. URL https://aclanthology.org/2020.findings-emnlp.390.

Serhii Havrylov and Ivan Titov. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 2149–2159. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/6810.pdf.

Jonathan Heek, Anselm Levskaya, Avital Oliver, Marvin Ritter, Bertrand Rondepierre, Andreas Steiner, and Marc van Zee. Flax: A neural network library and ecosystem for JAX, 2020. URL http://github.com/google/flax.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NeurIPS Deep Learning and Representation Learning Workshop*, 2015. URL http://arxiv.org/abs/1503.02531.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019. URL https://proceedings.mlr.press/v97/houlsby19a.html.

William Huang, Haokun Liu, and Samuel R. Bowman. Counterfactually-augmented SNLI training data does not yield better generalization than unaugmented data. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pages 82–87, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.insights-1.13. URL https://aclanthology.org/2020.insights-1.13.

Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.386. URL https://www.aclweb.org/anthology/2020.acl-main.386.

Alon Jacovi and Yoav Goldberg. Aligning faithful interpretations with their social attribution. *Transactions of the Association for Computational Linguistics*, 9:294–310, 2021. doi: 10.1162/tacl_a_00367. URL https://aclanthology.org/2021.tacl-1.18.

Sarthak Jain and Byron C. Wallace. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1357. URL https://www.aclweb.org/anthology/N19-1357.

Sarthak Jain, Sarah Wiegreffe, Yuval Pinter, and Byron C. Wallace. Learning to faithfully rationalize by construction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4459–4473, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.409. URL https://aclanthology.org/2020.acl-main.409.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=rkE3y85ee.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351, 2017. doi: 10.1162/tacl_a_00065. URL https://aclanthology.org/Q17-1024.

Nitish Joshi and He He. An investigation of the (in)effectiveness of counterfactually augmented data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3668–3681, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.256. URL https://aclanthology.org/2022.acl-long.256.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia, July 2018. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P18-4020.

Tasnim Kabir and Marine Carpuat. The UMD submission to the explainable MT quality estimation shared task: Combining explanation models with sequence labeling. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 230–237, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eval4nlp-1.22. URL https://aclanthology.org/2021.eval4nlp-1.22.

A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Sklgs0NFvr.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, António Góis, M. Amin Farajian, António V. Lopes, and André F. T. Martins. Unbabel's participation in the WMT19 translation quality estimation shared task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 78–84, Florence, Italy, August 2019a. Association for Computational Linguistics. doi: 10.18653/v1/W19-5406. URL https://aclanthology.org/W19-5406.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. OpenKiwi: An open source framework for quality estimation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 117–122, Florence, Italy, July 2019b. Association for Computational Linguistics. doi: 10.18653/v1/P19-3020. URL https://aclanthology.org/P19-3020.

Daniel Khashabi, Tushar Khot, and Ashish Sabharwal. More bang for your buck: Natural perturbation for robust question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language*

Processing (EMNLP), pages 163–170, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.12. URL https://aclanthology.org/2020.emnlp-main.12.

Siwon Kim, Jihun Yi, Eunji Kim, and Sungroh Yoon. Interpretation of NLP models through input marginalization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3154–3167, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.255. URL https://aclanthology.org/2020.emnlp-main.255.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1139. URL https://aclanthology.org/D16-1139.

Nikita Kitaev, Steven Cao, and Dan Klein. Multilingual constituency parsing with self-attention and pre-training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3499–3505, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1340. URL https://aclanthology.org/P19-1340.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://openreview.net/forum?id=rkgNKkHtvB.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. Attention is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7057–7075, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.574. URL https://aclanthology.org/2020.emnlp-main.574.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In Doina Precup and Yee Whye Teh, editors, *International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL http://proceedings.mlr.press/v70/koh17a.html.

Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artif. Intell.*, 97(1–2):273–324, December 1997. ISSN 0004-3702. doi: 10.1016/S0004-3702(97)00043-X. URL https://doi.org/10.1016/S0004-3702(97)00043-X.

Julia Kreutzer, Jasmijn Bastings, and Stefan Riezler. Joey NMT: A minimalist NMT toolkit for novices. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 109–114, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-3019. URL https://www.aclweb.org/anthology/D19-3019.

Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. Multi-agent cooperation and the emergence of (natural) language. In *International Conference on Learning Representations*, 2016. URL https://openreview.net/forum?id=Hk8N3Sclg.

Eric Lehman, Jay DeYoung, Regina Barzilay, and Byron C. Wallace. Inferring which medical treatments work from reports of clinical trials. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3705–3717, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1371. URL https://aclanthology.org/N19-1371.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1011. URL https://aclanthology.org/D16-1011.

Christoph Wolfgang Leiter. Reference-free word- and sentence-level translation evaluation with token-matching metrics. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 157–164, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eval4nlp-1.16. URL https://aclanthology.org/2021.eval4nlp-1.16.

Piyawat Lertvittayakumjorn and Francesca Toni. Human-grounded evaluations of explanation methods for text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5195–5205, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1523. URL https://aclanthology.org/D19-1523.

David Lewis. *Convention: A Philosophical Study*. John Wiley & Sons, 2008.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. Visualizing and understanding neural models in NLP. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 681–691, San Diego, California, June 2016a. Association for Computational Linguistics. doi: 10.18653/v1/N16-1082. URL https://www.aclweb.org/anthology/N16-1082.

Jiwei Li, Will Monroe, and Dan Jurafsky. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*, 2016b. URL https://arxiv.org/abs/1612.08220.

Zachary C. Lipton. The mythos of model interpretability. *Commun. ACM*, 61(10):36–43, 2018.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. URL https://arxiv.org/abs/1907.11692.

Arle Lommel, Aljoscha Burchardt, Maja Popović, Kim Harris, Eleftherios Avramidis, and Hans Uszkoreit. Using a new analytic measure for the annotation and analysis of MT errors on real data. In *Proceedings of the*

17th Annual conference of the European Association for Machine Translation, pages 165–172, Dubrovnik, Croatia, June 16-18 2014. European Association for Machine Translation. URL https://aclanthology.org/2014.eamt-1.38.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688v2*, 2023. URL https://arxiv.org/abs/2301.13688v2.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://aclanthology.org/P11-1015.

Nishtha Madaan, Inkit Padhi, Naveen Panwar, and Diptikalyan Saha. Generate your counterfactuals: Towards controlled counterfactual generation for text. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 13516–13524, 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/17594/17401.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=S1jE5L5gl.

Marianna Martindale and Marine Carpuat. Fluency over adequacy: A pilot study in measuring user trust in imperfect MT. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 13–25, Boston, MA, March 2018. Association for Machine Translation in the Americas. URL https://aclanthology.org/W18-1803.

Andre Martins and Ramon Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *International Conference on Machine Learning (ICML)*, volume 48 of *Proceedings of Machine Learning Research*, pages 1614–1623, New York, New York, USA, 20–22 Jun 2016. PMLR. URL http://proceedings.mlr.press/v48/martins16.html.

André Martins, António Farinhas, Marcos Treviso, Vlad Niculae, Pedro Aguiar, and Mario Figueiredo. Sparse and continuous attention mechanisms. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20989–21001. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/f0b76267fbe12b936bd65e203dc675c1-Paper.pdf.

André Martins, Marcos Treviso, António Farinhas, Pedro MQ Aguiar, Mário AT Figueiredo, Mathieu Blondel, and Vlad Niculae. Sparse continuous distributions and fenchel-young losses. *Journal of Machine Learning Research*, 23:1–74, 2022. URL https://www.jmlr.org/papers/v23/21-0879.html.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *5th International Conference on Learning Representations (ICLR)*, 2017. URL https://openreview.net/forum?id=Byj72udxe.

Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1 – 38, 2019. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2018.07.007. URL http://www.sciencedirect.com/science/article/pii/S0004370218305988.

Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1 – 15, 2018. ISSN 1051-2004. doi: https://doi.org/10.1016/j.dsp.2017.10.011. URL http://www.sciencedirect.com/science/article/pii/S1051200417302385.

Jesse Mu and Jacob Andreas. Compositional explanations of neurons. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. Explainable prediction of medical codes from clinical text. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1101–1111, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1100. URL https://aclanthology.org/N18-1100.

Michael Neely, Stefan F Schouten, Maurits JR Bleeker, and Ana Lucic. Order in the court: Explainable ai methods prone to disagreement. In *ICML Workshop on Theoretic Foundation, Criticism, and Application Trend of Explainable AI*, 2021. URL https://arxiv.org/abs/2105.03287.

Vlad Niculae and Mathieu Blondel. A regularized framework for sparse and structured neural attention. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/2d1b2a5ff364606ff041650887723470-Paper.pdf.

Vlad Niculae and Andre Martins. LP-SparseMAP: Differentiable relaxed optimization for sparse structured prediction. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7348–7359. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/niculae20a.html.

Vlad Niculae, Andre Martins, Mathieu Blondel, and Claire Cardie. SparseMAP: Differentiable sparse structured inference. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on*

Machine Learning, volume 80 of *Proceedings of Machine Learning Research*, pages 3799–3808. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/niculae18a.html.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.441. URL https://aclanthology.org/2020.acl-main.441.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1–9, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6301. URL https://aclanthology.org/W18-6301.

Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. doi: 10.3115/1219840.1219855. URL https://aclanthology.org/P05-1015.

Bhargavi Paranjape, Mandar Joshi, John Thickstun, Hannaneh Hajishirzi, and Luke Zettlemoyer. An information bottleneck approach for controlling conciseness in rationale extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1938–1952, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.153. URL https://aclanthology.org/2020.emnlp-main.153.

Bhargavi Paranjape, Matthew Lamm, and Ian Tenney. Retrieval-guided counterfactual generation for QA. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1670–1686, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.117. URL https://aclanthology.org/2022.acl-long.117.

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1244. URL https://aclanthology.org/D16-1244.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research (JMLR)*, 12:2825–2830, 2011.

Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL https://www.aclweb.org/anthology/D14-1162.

Ben Peters, Vlad Niculae, and André F. T. Martins. Interpretable structure induction via sparse attention. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 365–367, Brussels, Belgium, November 2018a. Association for Computational Linguistics. doi: 10.18653/v1/W18-5450. URL https://www.aclweb.org/anthology/W18-5450.

Ben Peters, Vlad Niculae, and André F. T. Martins. Sparse sequence-to-sequence models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1504–1519, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1146. URL https://www.aclweb.org/anthology/P19-1146.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018b. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://aclanthology.org/N18-1202.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. AdapterHub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.7. URL https://aclanthology.org/2020.emnlp-demos.7.

Peter Polák, Muskaan Singh, and Ondřej Bojar. Explainable quality estimation: CUNI Eval4NLP submission. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 250–255, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eval4nlp-1.24. URL https://aclanthology.org/2021.eval4nlp-1.24.

Matt Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6319. URL https://aclanthology.org/W18-6319.

Danish Pruthi, Rachit Bansal, Bhuwan Dhingra, Livio Baldini Soares, Michael Collins, Zachary C. Lipton, Graham Neubig, and William W. Cohen. Evaluating explanations: How much do explanations from the teacher aid students? *Transactions of the Association for Computational Linguistics*, 10:359–375, 2022. doi: 10.1162/tacl_a_00465. URL https://aclanthology.org/2022.tacl-1.21.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Alessandro Raganato and Jörg Tiedemann. An analysis of encoder representations in transformer-based machine translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting*

Neural Networks for NLP, pages 287–297, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5431. URL https://aclanthology.org/W18-5431.

Alessandro Raganato, Yves Scherrer, and Jörg Tiedemann. Fixed encoder self-attention patterns in transformer-based machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 556–568, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.49. URL https://aclanthology.org/2020.findings-emnlp.49.

Aniruddh Raghu, Maithra Raghu, Simon Kornblith, David Duvenaud, and Geoffrey Hinton. Teaching with commentaries. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=4RbdgBh9gE.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. TransQuest at WMT2020: Sentence-level direct assessment. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1049–1055, Online, November 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.wmt-1.122.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online, November 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.213. URL https://aclanthology.org/2020.emnlp-main.213.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. Unbabel's participation in the WMT20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 911–920, Online, November 2020b. Association for Computational Linguistics. URL https://aclanthology.org/2020.wmt-1.101.

Ricardo Rei, Ana C Farinha, José G.C. de Souza, Pedro G. Ramos, André F.T. Martins, Luisa Coheur, and Alon Lavie. Searching for COMETINHO: The little metric that could. In *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation*, pages 61–70, Ghent, Belgium, June 2022a. European Association for Machine Translation. URL https://aclanthology.org/2022.eamt-1.9.

Ricardo Rei, Marcos Treviso, Nuno M. Guerreiro, Chrysoula Zerva, Ana C Farinha, Christine Maroti, José G. C. de Souza, Taisiya Glushkova, Duarte Alves, Luisa Coheur, Alon Lavie, and André F. T. Martins. CometKiwi: IST-unbabel 2022 submission for the quality estimation shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 634–645, Abu Dhabi, United Arab Emirates (Hybrid), December 2022b. Association for Computational Linguistics. URL https://aclanthology.org/2022.wmt-1.60.

Ricardo Rei, Nuno Guerreiro, Marcos Treviso, Luisa Coheur, Alon Lavie, and André F. T. Martins. The inside story: Towards better understanding of machine translation neural evaluation metrics. In *Currently under review.*, 2023.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association*

for Computational Linguistics, pages 1085–1097, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1103. URL https://aclanthology.org/P19-1103.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, KDD '16, page 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939778. URL https://doi.org/10.1145/2939672.2939778.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.442. URL https://aclanthology.org/2020.acl-main.442.

Mark O Riedl. Human-centered artificial intelligence and machine learning. *Human Behavior and Emerging Technologies*, 1(1):33–36, 2019.

Marcel Robeer, Floris Bex, and Ad Feelders. Generating realistic natural language counterfactuals. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3611–3625, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.306. URL https://aclanthology.org/2021.findings-emnlp.306.

Alexis Ross, Ana Marasović, and Matthew Peters. Explaining NLP models via minimal contrastive editing (MiCE). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3840–3852, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.336. URL https://aclanthology.org/2021.findings-acl.336.

Alexis Ross, Matthew Peters, and Ana Marasovic. Does self-rationalization improve robustness to spurious correlations? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7403–7416, Abu Dhabi, United Arab Emirates, December 2022a. Association for Computational Linguistics. URL https://aclanthology.org/2022.emnlp-main.501.

Alexis Ross, Tongshuang Wu, Hao Peng, Matthew Peters, and Matt Gardner. Tailor: Generating and perturbing text with semantic controls. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3194–3213, Dublin, Ireland, May 2022b. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.228. URL https://aclanthology.org/2022.acl-long.228.

Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics*, 9:53–68, 2021. doi: 10.1162/tacl_a_00353. URL https://aclanthology.org/2021.tacl-1.4.

Raphael Rubino, Atsushi Fujita, and Benjamin Marie. Error identification for machine translation with metric embedding and attention. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Sys-*

tems, pages 146–156, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eval4nlp-1.15. URL https://aclanthology.org/2021.eval4nlp-1.15.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019. URL https://www.nature.com/articles/s42256-019-0048-x.

Jurgen Schmidhuber. Evolutionary principles in self-referential learning. on learning now to learn: The meta-meta-meta...-hook. Diploma thesis, Technische Universitat Munchen, Germany, 14 May 1987. URL http://www.idsia.ch/~juergen/diploma.html.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online, July 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.704. URL https://aclanthology.org/2020.acl-main.704.

Thibault Sellam, Amy Pu, Hyung Won Chung, Sebastian Gehrmann, Qijun Tan, Markus Freitag, Dipanjan Das, and Ankur Parikh. Learning to evaluate translation beyond English: BLEURT submissions to the WMT metrics 2020 shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 921–927, Online, November 2020b. Association for Computational Linguistics. URL https://aclanthology.org/2020.wmt-1.102.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL https://aclanthology.org/P16-1162.

Sofia Serrano and Noah A. Smith. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1282. URL https://www.aclweb.org/anthology/P19-1282.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism, 2019. URL https://arxiv.org/abs/1909.08053.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034v2*, 2013. URL https://arxiv.org/abs/1312.6034v2.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine*

Translation in the Americas: Technical Papers, pages 223–231, Cambridge, Massachusetts, USA, August 8-12 2006. Association for Machine Translation in the Americas. URL https://aclanthology.org/2006.amta-papers.25.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1170.

Lucia Specia, Gustavo Paetzold, and Carolina Scarton. Multi-level Translation Quality Prediction with QuEst++. In *Proc. of ACL-IJCNLP 2015 System Demonstrations*, pages 115–120, Beijing, China, 2015. URL http://www.aclweb.org/anthology/P15-4020.

Lucia Specia, Carolina Scarton, and Gustavo Henrique Paetzold. Quality Estimation for Machine Translation. *Synthesis Lectures on Human Language Technologies*, 11(1):1–162, 2018.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André F. T. Martins. Findings of the WMT 2020 shared task on quality estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 743–764, Online, November 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.wmt-1.79.

Anant Subramanian, Danish Pruthi, Harsh Jhamtani, Taylor Berg-Kirkpatrick, and Eduard Hovy. Spine: Sparse interpretable neural embeddings. In *Proceedings of the Thirty Second AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Gradients of counterfactuals. *arXiv preprint arXiv:1611.02639*, 2016. URL https://arxiv.org/abs/1611.02639.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/sundararajan17a.html.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://papers.nips.cc/paper_files/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf.

Shimin Tao, Su Chang, Ma Miaomiao, Hao Yang, Xiang Geng, Shujian Huang, Min Zhang, Jiaxin Guo, Minghan Wang, and Yinglu Li. CrossQE: HW-TSC 2022 submission for the quality estimation shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 646–652, Abu Dhabi, United Arab Emirates (Hybrid), December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.wmt-1.61.

Yi Tay, Dara Bahri, Liu Yang, Donald Metzler, and Da-Cheng Juan. Sparse sinkhorn attention. In *International Conference on Machine Learning (ICML)*, pages 9438–9447. PMLR, 2020.

Damien Teney, Ehsan Abbasnedjad, and Anton van den Hengel. Learning what makes a difference from counterfactual examples and gradient supervision. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pages 580–599. Springer, 2020. URL https://www.ecva.net/papers/eccv_2020/papers_ECCV/papers/123550579.pdf.

Robert Tibshirani. Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996. doi: 10.1111/j.2517-6161.1996.tb02080.x. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1996.tb02080.x.

Marcos Treviso and André F. T. Martins. The explanation game: Towards prediction explainability through sparse communication. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 107–118, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.blackboxnlp-1.10. URL https://aclanthology.org/2020.blackboxnlp-1.10.

Marcos Treviso, Nuno M. Guerreiro, Ricardo Rei, and André F. T. Martins. IST-unbabel 2021 submission for the explainable quality estimation shared task. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 133–145, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eval4nlp-1.14. URL https://aclanthology.org/2021.eval4nlp-1.14.

Marcos Treviso, António Góis, Patrick Fernandes, Erick Fonseca, and Andre Martins. Predicting attention sparsity in transformers. In *Proceedings of the Sixth Workshop on Structured Prediction for NLP*, pages 67–81, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.spnlp-1.7. URL https://aclanthology.org/2022.spnlp-1.7.

Marcos Treviso, Ji-Ung Lee, Tianchu Ji, Betty van Aken, Qingqing Cao, Manuel R. Ciosici, Michael Hassid, Kenneth Heafield, Sara Hooker, Colin Raffel, Pedro H. Martins, André F. T. Martins, Jessica Zosa Forde, Peter Milder, Edwin Simpson, Noam Slonim, Jesse Dodge, Emma Strubell, Niranjan Balasubramanian, Leon Derczynski, Iryna Gurevych, and Roy Schwartz. Efficient methods for natural language processing: A survey. *arXiv preprint arXiv:2209.00099v2*, 2023a. URL https://arxiv.org/abs/2209.00099v2. Accepted at TACL.

Marcos Treviso, Alexis Ross, Nuno Guerreiro, and André F. T. Martins. A joint framework for rationalization and counterfactual text generation. In *Currently under review.*, 2023b.

Valentin Trifonov, Octavian-Eugen Ganea, Anna Potapenko, and Thomas Hofmann. Learning and evaluating sparse interpretable sentence embeddings. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 200–210, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5422. URL https://www.aclweb.org/anthology/W18-5422.

Constantino Tsallis. Possible generalization of boltzmann-gibbs statistics. *Journal of Statistical Physics*, 1988.

Keyon Vafa, Yuntian Deng, David Blei, and Alexander Rush. Rationales for sequential predictions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10314–10332, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.807. URL https://aclanthology.org/2021.emnlp-main.807.

Janneke Van de Pol, Monique Volman, and Jos Beishuizen. Scaffolding in teacher–student interaction: A decade of research. *Educational psychology review*, 22(3):271–296, 2010.

Shikhar Vashishth, Shyam Upadhyay, Gaurav Singh Tomar, and Manaal Faruqui. Attention Interpretability Across NLP Tasks. *arXiv preprint arXiv:1909.11218*, 2019. URL https://arxiv.org/abs/1909.11218.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Sahil Verma, John Dickerson, and Keegan Hines. Counterfactual explanations for machine learning: A review. In *NeurIPS 2020 Workshop: ML Retrospectives, Surveys & Meta-Analyses (ML-RSA)*, 2020.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1580. URL https://aclanthology.org/P19-1580.

Elena Voita, Rico Sennrich, and Ivan Titov. Analyzing the source and target contributions to predictions in neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1126–1140, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.91. URL https://aclanthology.org/2021.acl-long.91.

Apoorv Vyas, Angelos Katharopoulos, and François Fleuret. Fast transformers with clustered attention. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21665–21674. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/f6a8dd1c954c8506aadc764cc32b895e-Paper.pdf.

Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schrödl. Constrained k-means clustering with background knowledge. In *International Conference on Machine Learning (ICML)*, page 577–584, 2001.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. Universal adversarial triggers for attacking and analyzing NLP. In *Proceedings of the 2019 Conference on Empirical Methods in Natural*

Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 2153–2162, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1221. URL https://aclanthology.org/D19-1221.

Yu Wan, Dayiheng Liu, Baosong Yang, Haibo Zhang, Boxing Chen, Derek Wong, and Lidia Chao. UniTE: Unified translation evaluation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8117–8127, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.558. URL https://aclanthology.org/2022.acl-long.558.

Sinong Wang, Belinda Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020. URL https://arxiv.org/abs/2006.04768.

Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research (JMLR)*, 10(2), 2009.

Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1002. URL https://www.aclweb.org/anthology/D19-1002.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL https://aclanthology.org/N18-1101.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time, 2022. URL https://arxiv.org/abs/2203.05482.

Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel Weld. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*

(Volume 1: Long Papers), pages 6707–6723, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.523. URL https://aclanthology.org/2021.acl-long.523.

Eric P Xing, Andrew Y Ng, Michael I Jordan, and Stuart Russell. Distance metric learning with application to clustering with side-information. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 15, page 12, 2002.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In Francis Bach and David Blei, editors, *International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2048–2057, Lille, France, 07–09 Jul 2015. PMLR. URL http://proceedings.mlr.press/v37/xuc15.html.

Yilun Xu, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon. A theory of usable information under computational constraints. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=r1eBeyHFDH.

Yordan Yordanov, Vid Kocijan, Thomas Lukasiewicz, and Oana-Maria Camburu. Few-shot out-of-domain transfer learning of natural language explanations in a label-abundant setup. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3486–3501, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.findings-emnlp.255.

Mo Yu, Shiyu Chang, Yang Zhang, and Tommi Jaakkola. Rethinking cooperative rationalization: Introspective extraction and complement control. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4094–4103, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1420. URL https://aclanthology.org/D19-1420.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 2020.

Omar Zaidan and Jason Eisner. Modeling annotators: A generative approach to learning from annotator rationales. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 31–40, Honolulu, Hawaii, October 2008. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D08-1004.

Chrysoula Zerva, Daan van Stigt, Ricardo Rei, Ana C Farinha, Pedro Ramos, José G. C. de Souza, Taisiya Glushkova, Miguel Vera, Fabio Kepler, and André F. T. Martins. IST-unbabel 2021 submission for the quality estimation shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 961–972, Online, November 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.wmt-1.102.

Chrysoula Zerva, Frédéric Blain, Ricardo Rei, Piyawat Lertvittayakumjorn, José G. C. de Souza, Steffen Eger, Diptesh Kanojia, Duarte Alves, Constantin Orăsan, Marina Fomicheva, André F. T. Martins, and Lucia Specia. Findings of the WMT 2022 shared task on quality estimation. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 69–99, Abu Dhabi, United Arab Emirates (Hybrid), December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.wmt-1.3.

Biao Zhang, Ivan Titov, and Rico Sennrich. Sparse attention with linear units. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6507–6520, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.523. URL https://aclanthology.org/2021.emnlp-main.523.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/250cf8b51c773f3f8dc8b4be867a9a02-Paper.pdf.

Guangxiang Zhao, Junyang Lin, Zhiyuan Zhang, Xuancheng Ren, Qi Su, and Xu Sun. Explicit sparse transformer: Concentrated attention through explicit selection. *arXiv preprint arXiv:1912.11637*, 2019. URL https://arxiv.org/abs/1912.11637.

Wangchunshu Zhou, Canwen Xu, and Julian McAuley. BERT learns to teach: Knowledge distillation with meta learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7037–7049, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.485. URL https://aclanthology.org/2022.acl-long.485.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, page 1097–1100, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356572. doi: 10.1145/3209978.3210080. URL https://doi.org/10.1145/3209978.3210080.

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. ST-MoE: Designing Stable and Transferable Sparse Expert Models. *arXiv preprint arXiv:2202.08906v2*, 2022. URL https://arxiv.org/abs/2202.08906v2.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005. doi: 10.1111/j.1467-9868.2005.00503.x. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2005.00503.x.

Supplemental Material for Chapter 3

A.1 Classifiers experimental setup (Table 3.3)

We chose our classifiers so that they are close to the models used by related works (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019; Bastings et al., 2019). For all models, we calculated their accuracy on the dev set after each epoch. At the end of training we selected the model with the best validation accuracy. We experimented with two classes of classifiers: a simple RNN with attention as in Jain and Wallace (2019); Wiegreffe and Pinter (2019); and the rationalizer models of Lei et al. (2016) and Bastings et al. (2019) which sample binary masks from Bernoulli and HardKuma distributions, respectively.

A.1.1 RNNs with attention

For the text classification experiments, each input word x_i is mapped to 300D-pretrained GloVe embeddings (Pennington et al., 2014) from the 840B release,[1] kept frozen, followed by a bidirectional LSTM layer (BiLSTM) resulting in vectors $\mathbf{h}_1, \ldots, \mathbf{h}_n$. We score each of these vectors using the additive formulation of Bahdanau et al. (2015), applying an attention transformation to convert the resulting scores $\mathbf{s} \in \mathbb{R}^n$ to a probability distribution $\pi \in \triangle^n$. We use this to compute a contextual vector $\mathbf{c} = \sum_{i=1}^{n} \pi_i \mathbf{h_i}$, which is fed into the output softmax layer that predicts $\hat{y}$. For NLI, the input x is a pair of sentences (a premise and an hypothesis), and the classifier C is similar to the the above, but with two independent BiLSTM layers, one for each sentence. In the attention layer, we use the last hidden state of the hypothesis as the query and the premise vectors as keys.

We used the AdamW Loshchilov and Hutter (2019) optimizer for all experiments. We tuned two hyperparameters: learning rate within $\{0.003, \mathbf{0.001}, 0.0001\}$, and l_2 regularization within $\{0.01, 0.001, \mathbf{0.0001}, 0\}$. We picked the best configuration by doing a grid search and by taking into consideration the accuracy on the validation set (selected values in bold). Table A.1 shows all hyperparameters set for training.

HYPERPARAM.	SST	IMDB	AGNEWS	YELP	SNLI
Word embeddings size	300	300	300	300	300
BiLSTM hidden size	128	128	128	128	128
Merge BiLSTM states	concat	concat	concat	concat	concat
Batch size	8	16	16	128	32
Number of epochs	10	10	5	5	10
Early stopping patience	5	5	3	3	5
Learning rate	0.001	0.001	0.001	0.001	0.001
ℓ_2 regularization	0.0001	0.0001	0.0001	0.0001	0.0001

Table A.1: RNNs training hyperparameters for text classification and NLI datasets.

A.1.2 Bernoulli and HardKuma

We used the implementation of Bastings et al. (2019),[2] which includes a reimplementation of the generator-encoder model from (Lei et al., 2016). The model used for text classification is a RNN-based generator followed by a RNN-based encoder, whereas for NLI is a decomposable attention classifier from (Parikh et al., 2016), for which only the HardKuma implementation was available. In order to faithfully compare the frameworks, we adapted the HardKuma code and implemented a Bernoulli version of the same classifier, taking into consideration the sparsity and fused-lasso loss penalties, and the deterministic strategy used during test time. For

[1] http://nlp.stanford.edu/data/glove.840B.300d.zip
[2] https://github.com/bastings/interpretable_predictions

simplicity, we used the independent variant of the generator of Lei et al. (2016). Table A.2 lists only the hyperparameters that we set during training. We refer to the original work of Bastings et al. (2019) to see all other hyperparamers, for which we kept the default values.

HYPERPARAM.	SST	IMDB	AGNEWS	YELP	SNLI
Latent selection (HardKuma)	0.3	0.1	0.3	0.3	0.1
Sparsity penalty (Bernoulli)	0.01	0.001	0.01	0.01	0.0003
Lasso penalty	0	0	0	0	0
Batch size	25	25	25	256	64
Number of epochs	25	25	25	10	100
Early stopping patience	5	5	5	5	100
Learning rate	0.0002	0.0002	0.0002	0.001	0.0002
ℓ_2 regularization	10^{-5}	10^{-5}	10^{-5}	10^{-5}	10^{-6}

Table A.2: Rationalizer models training hyperparameters for text classification and NLI datasets.

A.1.3 Validation set results and model statistics

Table A.3 shows the accuracy of each classifier on the validation set, their number of trainable parameters and the average training time per epoch.

	SST			IMDB			AGNEWS		
CLF.	# P	t	ACC	# P	t	ACC	# P	t	ACC
C	474K	10s	85.32	474K	2m	95.64	474K	2m	98.09
C_{ent}	474K	10s	84.29	474K	2m	95.84	474K	2m	98.54
C_{sp}	474K	10s	84.17	474K	2m	95.44	474K	2m	98.51
C_{bern}	1.1M	15s	80.16	1.1M	2m	87.40	1.1M	2m	96.26
C_{hk}	1.1M	15s	84.40	1.1M	2m	91.84	1.1M	2m	96.74

Table A.3: Classifier results on the validation set and model statistics. # P is the number of trainable parameters, and is t the average training time per epoch.

	YELP			SNLI		
CLF.	# P	t	ACC	# P	t	ACC
C	474K	3h	77.03	998K	4m	78.74
C_{ent}	474K	3h	76.72	998K	4m	79.38
C_{sp}	474K	3h	76.84	998K	4m	79.69
C_{bern}	1.1M	5h	69.99	382K	2m	79.79
C_{hk}	1.1M	5h	74.29	462K	2m	86.04

Table A.4: Continuation of Table A.3.

A.2 Communication experimental setup (Table 3.4)

Training the communication under our framework consists on training a layperson L on top of explanations (message) produced by E about C's decision. With the exception of the explainer E trained jointly with L, none of the other explainers have trainable parameters. Therefore, in these cases, the communication between E and L consists only on training L. For all models, we calculated its CSR on the dev set after each epoch. At

the end of training we selected the model with the best validation CSR. Table A.5 shows the communication hyperparameters. Note that for SNLI we still need to train a BiLSTM to encode the hypothesis.

HYPERPARAM.	SST	IMDB	AGNEWS	YELP	SNLI
Word embeddings size	-	-	-	-	300
BiLSTM hidden size	-	-	-	-	128
Merge BiLSTM states	-	-	-	-	concat
Batch size	16	16	16	112	64
Number of epochs	10	10	10	5	10
Early stopping patience	3	3	3	3	3
Learning rate	0.001	0.001	0.001	0.003	0.001
ℓ_2 regularization	10^{-5}	10^{-5}	10^{-5}	10^{-5}	10^{-5}

Table A.5: Communication hyperparameters for text classification and NLI datasets.

A.2.1 Validation set results and model statistics

Table A.6 shows the CSR and ACC_L for each explainer on the validation set, the number of trainable parameters of L and the average training time per epoch.

	SST				IMDB				AGNEWS			
EXPLAINER	# P	t	CSR	ACC_L	# P	t	CSR	ACC_L	# P	t	CSR	ACC_L
Random	38K	10s	63.76	62.84	247K	1m	61.36	61.24	120K	2m	85.26	84.58
Erasure	38K	10s	81.88	79.82	247K	2m	94.00	91.40	120K	3m	98.41	96.98
Top-k gradient	38K	10s	76.72	75.57	247K	1m	91.88	89.52	120K	2m	98.23	96.97
Top-k softmax	38K	20s	84.29	80.62	247K	1m	96.60	93.60	120K	2m	98.54	97.14
Top-k 1.5-entmax	38K	20s	85.44	80.28	247K	1m	97.88	94.92	120K	2m	98.22	97.37
Top-k sparsemax	38K	20s	85.44	81.54	247K	1m	96.76	93.32	120K	2m	96.46	95.72
Select. 1.5-entmax	38K	10s	85.55	80.62	247K	1m	97.44	94.56	120K	1m	98.30	97.41
Select. sparsemax	38K	10s	85.44	81.54	247K	1m	97.04	93.36	120K	1m	96.46	95.72
Bernoulli	38K	5s	84.75	78.21	247K	1m	91.80	87.36	120K	1m	97.12	94.82
HardKuma	38K	5s	87.50	81.76	247K	1m	95.36	91.20	120K	1m	97.38	96.05

Table A.6: Communication results on the validation set and explainer statistics. # P is the number of trainable parameters, and is t the average training time per epoch.

	YELP				SNLI			
EXPLAINER	# P	t	CSR	ACC_L	# P	t	CSR	ACC_L
Random	1.8M	3h	52.55	48.21	560K	9m	31.04	33.11
Erasure	1.8M	4h	79.63	69.59	560K	10m	78.72	70.60
Top-k gradient	1.8M	3h	71.81	63.59	560K	10m	77.55	69.41
Top-k softmax	1.8M	3h	81.49	70.67	560K	9m	79.10	70.95
Top-k 1.5-entmax	1.8M	3h	82.80	71.31	560K	9m	80.30	73.57
Top-k sparsemax	1.8M	3h	82.97	71.46	560K	9m	83.25	75.34
Select. 1.5-entmax	1.8M	2h	82.90	70.99	560K	6m	77.46	71.66
Select. sparsemax	1.8M	2h	84.67	72.25	560K	6m	82.33	75.11
Bernoulli	1.8M	2h	84.93	66.77	560K	2m	75.75	68.61
HardKuma	1.8M	2h	87.43	71.57	560K	3m	75.10	71.10

Table A.7: Continuation of Table A.6.

A.3 Joint E and L setup

A.3.1 Communication

According to §3.4.2, in this model we have two set of parameters to train, one for the explainer E and other for the layperson L, whereas the classifier is a frozen model that we want to explain. Here, we set C as the RNN with softmax classifier (see §3.3). We design E with the same architecture of the RNNs with attention from §A.1.1 but without a final output layer, and L have the same architecture as the laypersons in §3.5. In short, the architecture of E is composed of: (i) embedding layer; (ii) BiLSTM; (iii) attention mechanism. As before, the message is constructed with the words extracted from the attention mechanism.

We use sparsemax attention during training to ensure end-to-end differentiability, and we recover the top-k attended words during test time. We used $k = 5$ for IMDB and $k = 4$ for SNLI in all experiments. In order to encourage faithful explanations, we set $h = \frac{1}{L}\sum_i C_{\text{RNN}}(x_i)$ and $\tilde{h} = \frac{1}{L}\sum_i \text{FFN}(E_{\text{RNN}}(x_i))$, where FFN is a simple feed-forward layer, and $C_{\text{RNN}}(x_i)$ and $E_{\text{RNN}}(x_i)$ are the BiLSTM states from the classifier and the explainer, respectively. In other words, we are approximating the average of the BiLSTM states of C and E. We set $\lambda = 1$ and $\beta = 0.2$ and used the same hyperparameters as in Table A.5. The list of stopwords used in our experiments contains 127 English words extracted from NLTK.

A.3.2 Analysis of β

A potential problem of this model is for the two agents to agree on a trivial protocol, ensuring a high CSR even with bad quality explanations (e.g. punctuations or stopwords). Besides preventing stopwords to be in the message,[3] we set a different probability β of the explainer accessing the predictions of the classifier $\hat{y}$. Intuitively, these strategies should encourage explanations to have higher quality. One way to quantitatively access the quality of the explanations is by aggregating the relative frequencies of each selected word in the validation set, and calculating its Shannon's entropy. If the entropy is low, then the explanations have a high number of repetitions and the explainers are focusing on a very small subset of words, denoting a trivial protocol. To check for a reasonable entropy score that resembles a good quality explanation, we investigate the entropy of the other explainers, for which we had confirmed their quality via human evaluation.

In order to see the impact of β, we carried an experiment with increasing values of β and looked at the CSR, ACC_L and the entropy (H) of the generated explanations. Results are shown in Table A.8 for each explainer on IMDB and SNLI.

When $\beta = 0$ no information about the label predicted by the classifier is being exposed to the explainer, and as a result we have a model that resembles a combination of selective (during training) and top-k (during test time) sparsemax explainers. This means that the results between these explainers are expected to be very similar in terms of CSR.[4] Overall, for both datasets, we can see a tradeoff between CSR and entropy H as β increases, suggesting that CSR is not able to capture the notion of quality (which was expected due to the subjective nature of an explanation). For IMDB the entropy values were lower than our previous explainers, but for SNLI they were very similar. A potential reason for this is the particularity of the two datasets: IMDB have long documents (280 words on average) with a large set of repetitive words which are not stopwords and

[3] In practice, we simply set attention scores associated with stopwords to $-\infty$.

[4] Note that this also depends on the performance of C and C_{sp}, which are indeed very similar in this case: 95.64 and 95.44.

		IMDB			SNLI		
CLF.	EXPLAINER	H	CSR	ACC_L	H	CSR	ACC_L
C	Random	9.13	59.20	58.92	8.21	31.04	33.11
C	Erasure	9.40	96.32	93.48	9.75	78.72	70.60
C	Top-k gradient	9.49	85.84	83.72	9.39	77.55	69.41
C	Top-k softmax	9.38	94.44	91.84	9.76	78.66	71.00
C_{ent}	Top-k 1.5-entmax	9.62	95.20	93.36	9.54	80.30	73.57
C_{sp}	Top-k sparsemax	9.56	95.28	92.56	8.79	83.25	75.34
C_{ent}	Select. 1.5-entmax	10.76	97.44	94.56	8.49	77.46	71.66
C_{sp}	Selec. sparsemax	10.41	97.04	93.36	8.38	82.33	75.11
C_{bern}	Bernoulli	10.66	91.88	87.36	8.27	75.75	68.61
C_{hk}	HardKuma	11.38	95.36	91.20	9.93	75.10	71.10
-	Human highlights	-	-	-	8.72	87.97	87.97
C	Joint E and L ($\beta = 0.0$)	6.16	93.04	90.84	9.81	80.74	72.38
C	Joint E and L ($\beta = 0.2$)	6.05	98.52	94.56	9.81	93.44	77.20
C	Joint E and L ($\beta = 0.5$)	5.63	99.68	95.64	9.45	95.81	77.54
C	Joint E and L ($\beta = 1.0$)	3.72	99.92	95.56	9.01	97.49	77.23

Table A.8: Entropy of the explanations for all explainers on the validation set of IMDB and SNLI. Entropy for human highlights was calculated based on non-neutral examples.

are strongly correlated with the labels (e.g. good, ok, bad, etc.); SNLI premises are very short (14 words on average) without a large set of repetitive words. Finally, due to this tradeoff, we selected $\beta = 0.2$ for all of our experiments since it induces a very high CSR with a reasonably good entropy.

A.4 Machine Translation experiments

A.4.1 Data

To compare explainers on a more challenging task with large $|\mathcal{Y}|$, we ran an experiment on neural machine translation (NMT), adapting the JoeyNMT framework (Kreutzer et al., 2019). We used the EN→DE IWSLT 2017 dataset (Cettolo et al., 2017), with the standard splits (Table 3.2).

A.4.2 Classifier

We replicated the work of Peters et al. (2019) with the exception that we used raw words as input instead of byte-pair encodings. The implementation is based on Joey-NMT (Kreutzer et al., 2019). We employed beam search decoding with beam size of 5, achieving a BLEU score of 20.49, 21.12 and 20.75 for softmax (C), 1.5-entmax (C_{ent}) and sparsemax (C_{sp}), respectively. We refer to the work of Peters et al. (2019) for more training details. Table A.9 shows the classifier hyperparameters.

A.4.3 Communication

The layperson is a model that uses an unidirectional LSTM with 256 hidden units to encode the translation prefix, and a feed-forward layer to encode the concatenation of k source word embeddings (the message) to a vector of 256 dimensions. The two vectors are concatenated and passed to a linear output layer to predict the next word $\tilde{y} \in \mathcal{Y}$ from the target vocabulary. We used 300D-pretrained GloVe embeddings to encode source

HYPERPARAM.	VALUE
Word embeddings size	512
BiRNN hidden size	512
Attention scorer	(Bahdanau et al., 2015)
Batch size	32
Optimizer	Adam
Number of epochs	100
Early stopping patience	8
Learning rate	0.001
Decrease factor	0.5
ℓ_2 regularization	0
RNN type	LSTM
RNN layers	2
Dropout	0.3
Hidden dropout	0.3
Maximum output length	100
Beam size	5

Table A.9: Classifier hyperparmeters for neural machine translation.

words (EN), and 300D-pretrained FastText embeddings to encode target words (DE).[5] Table A.10 shows the communication hyperparameters.

HYPERPARAM.	VALUE
Word embeddings size	300
LSTM hidden size	256
Merge LSTM states	concat
Batch size	16
Number of epochs	10
Early stopping patience	5
Learning rate	0.003
ℓ_2 regularization	10^{-5}

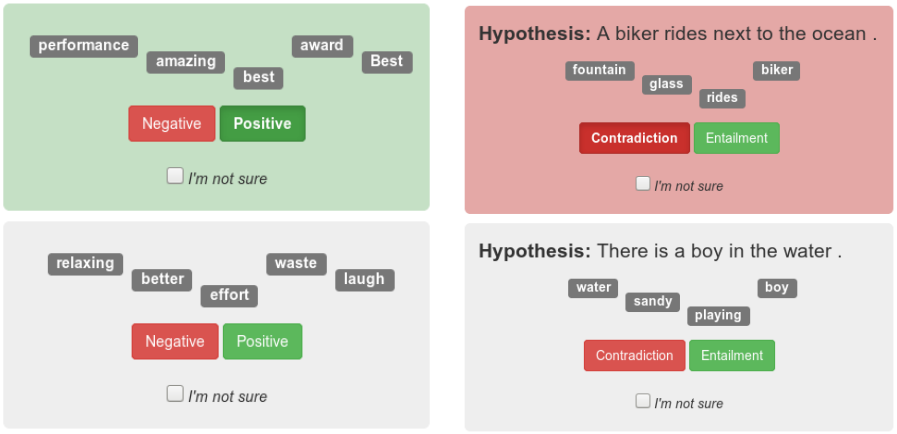
Table A.10: Communication hyperparmeters for neural machine translation.

A.5 Human annotation

We had four different human annotators, two for IMDB and two for SNLI. No information was given about the explainers which produced each message, and documents were presented in random order. Since in our experiments we define the message as being a bag-of-words, which does not encode order information, the explanations (i.e. the selected words) were shuffled and displayed as a cloud of words. The annotators were asked to predict the label of each document, when seeing only these explanations. For SNLI, we show the entire hypothesis as raw text and the premise as a cloud of words. We selected top-k explainers with $k = 5$ for IMDB and $k = 4$ for SNLI. Figure A.1 shows a snapshot of the annotation interface used for the experiments described in §3.6.

By directly looking at the explanations, we observed that some of them are very ambiguous with respect to the true label, so we decided to include a checkbox to be marked in case the annotator was not sure by his/her decision. The unsure checkbox also helps to capture the notion of sufficiency, that is, if the explanations are sufficient for a human predict some label. A similar approach was employed by Yu et al. (2019) using a

[5] https://dl.fbaipublicfiles.com/fasttext/vectors-crawl/cc.de.300.bin.gz



(a) IMDB **(b)** SNLI

Figure A.1: Snapshot of the annotation interface.

two-stage annotation method, explicitly asking the human annotator if the rationale was sufficient for his/her decision. Furthermore, we calculated the agreement between explainers using the Cohen's kappa coefficient and the relative observed agreement ratio (or accuracy, p_o). Table A.11 shows statistics for the unsure checkbox and agreement between annotators.

CLF.	EXPLAINER	IMDB					SNLI				
		u	p_o	κ	CSR_H	ACC_H	u	p_o	κ	CSR_H	ACC_H
C	Erasure	0.05	0.92	0.83	89.25	86.25	0.25	0.83	0.66	72.50	83.50
C	Top-k gradient	0.17	0.76	0.51	73.50	73.00	0.32	0.80	0.59	65.75	76.75
C	Top-k softmax	0.23	0.91	0.81	89.25	88.25	0.25	0.78	0.55	72.00	82.75
C_{ent}	Top-k 1.5-entmax	0.09	0.91	0.81	89.25	85.75	0.29	0.82	0.64	70.00	80.50
C_{sp}	Top-k sparsemax	0.09	0.88	0.76	89.00	87.50	0.38	0.80	0.59	68.25	80.25
C_{ent}	Selec. 1.5-entmax	0.13	0.80	0.60	86.50	84.00	0.21	0.84	0.67	75.25	87.00
C_{sp}	Selec. sparsemax	0.10	0.89	0.77	87.75	86.75	0.35	0.83	0.66	72.25	85.00
C_{bern}	Bernoulli	0.25	0.72	0.43	79.00	75.00	0.24	0.85	0.69	74.50	86.75
C_{hk}	HardKuma	0.17	0.81	0.61	83.75	80.75	0.18	0.86	0.72	79.25	87.50
C	Joint E and L	0.12	0.96	0.91	96.75	89.25	0.65	0.71	0.44	58.00	70.00
-	Human highlights	-	-	-	-	-	0.34	0.88	0.74	83.25	83.25
Average		**0.14**	**0.85**	**0.70**	-	-	**0.31**	**0.82**	**0.63**	-	-

Table A.11: Results for human evaluation. κ is the Cohen's kappa coefficient, p_o is the relative observed agreement, and u represents the average of the portion of examples where annotators were unsure about their decisions.

A.6 Examples of explanations

Tables A.12 and A.13 show the average word overlap between explainers' messages (m) for IMDB and SNLI. Looking at the statistics we observed that, in general, top-k attention-based classifiers produce similar explanations among themselves, and the erasure explainer produces messages similar to top-k softmax. Major differences are observed for top-k gradient and rationalizers, while selective attention produces, by definition, more words than top-k attention (i.e. $m_{\text{top-k}} \subseteq m_{\text{selective}}$). It is worth noticing that although explainers with similar messages are expected to have a similar CSR (e.g. top-k attention and erasure), including/excluding a

single word in the explanation might impact the layperson decision, as we can see in the next examples. Tables A.14 and A.15 show the output of erasure, gradient, attention, and joint explainers for IMDB, along with the prediction made by the classifier (y_C) and the layperson (y_L). In Tables A.16 and A.17 we also include the human highlights explainer for SNLI.

	Erasure	Top-k gradient	Top-k softmax	Top-k 1.5-entmax	Top-k sparsemax	Selec. 1.5-entmax	Selec. sparsemax	Bernoulli	HardKuma	Joint E and L
Erasure	*1.00*	0.34	0.85	0.56	0.55	0.20	0.37	0.14	0.23	0.20
Top-k gradient	0.34	*1.00*	0.35	0.30	0.30	0.16	0.26	0.11	0.18	0.11
Top-k softmax	0.85	0.35	*1.00*	0.57	0.55	0.20	0.37	0.14	0.24	0.20
Top-k 1.5-entmax	0.56	0.30	0.57	*1.00*	0.61	0.21	0.39	0.12	0.24	0.19
Top-k sparsemax	0.55	0.30	0.55	0.61	*1.00*	0.20	0.43	0.13	0.24	0.20
Selec. 1.5-entmax	0.20	0.16	0.20	0.21	0.20	*1.00*	0.45	0.24	0.44	0.08
Selec. sparsemax	0.37	0.26	0.37	0.39	0.43	0.45	*1.00*	0.21	0.41	0.13
Bernoulli	0.14	0.11	0.14	0.12	0.13	0.24	0.21	*1.00*	0.28	0.06
HardKuma	0.23	0.18	0.24	0.24	0.24	0.44	0.41	0.28	*1.00*	0.08
Joint E and L	0.20	0.11	0.20	0.19	0.20	0.08	0.13	0.06	0.08	*1.00*

Table A.12: Average word overlap (%) between explainers for IMDB.

	Erasure	Top-k gradient	Top-k softmax	Top-k 1.5-entmax	Top-k sparsemax	Selec. 1.5-entmax	Selec. sparsemax	Bernoulli	HardKuma	Joint E and L
Erasure	*1.00*	0.38	0.77	0.55	0.41	0.35	0.37	0.32	0.49	0.38
Top-k gradient	0.38	*1.00*	0.40	0.36	0.31	0.34	0.33	0.32	0.35	0.26
Top-k softmax	0.77	0.40	*1.00*	0.56	0.41	0.36	0.37	0.32	0.49	0.38
Top-k 1.5-entmax	0.55	0.36	0.56	*1.00*	0.46	0.36	0.42	0.32	0.46	0.34
Top-k sparsemax	0.41	0.31	0.41	0.46	*1.00*	0.35	0.54	0.32	0.38	0.29
Selec. 1.5-entmax	0.36	0.34	0.36	0.36	0.35	*1.00*	0.64	0.88	0.48	0.26
Selec. sparsemax	0.37	0.33	0.37	0.42	0.54	0.64	*1.00*	0.60	0.45	0.26
Bernoulli	0.32	0.32	0.32	0.32	0.32	0.88	0.60	*1.00*	0.46	0.24
HardKuma	0.49	0.35	0.49	0.46	0.38	0.48	0.45	0.46	*1.00*	0.38
Joint E and L	0.38	0.26	0.38	0.34	0.29	0.26	0.26	0.24	0.38	*1.00*

Table A.13: Average word overlap (%) between explainers for SNLI.

(positive) Mardi Gras : Made in china is an excellent movie that depicts how two cultures have much in common but , are not even aware of the influence each society has on one another . David Redmon open your eyes and allows you to see how the workers in china manufactures beads that cost little to nothing and are sold in America for up to 20 dollars . When Redmon questions Americans about where these beads come from they had no clue and seemed dumb founded . When he told them that they are made in China for less then nothing with horrible pay and unacceptable working conditions , Americans seemed sad , hurt , and a little remorseful but didn ' t really seem that they would stop purchasing the beads after finding out the truth . When Redmon questioned the workers in china they did not know that Americans were wearing them over their necks and paid so much for these beads . The workers laughed at what the purpose was behind beads and couldn ' t believe it . This movie is a great film that gives us something to think about in other countries besides our own . < br > < br > M . Pitts

EXPLAINER	y_C	y_L	EXPLANATION
Erasure	pos	pos	excellent great film besides hurt
Top-k gradient	pos	neg	hurt horrible a excellent couldn
Top-k softmax	pos	pos	excellent great film movie besides
Top-k 1.5-entmax	pos	pos	great excellent couldn that besides
Top-k sparsemax	pos	pos	excellent great couldn gives besides
Select. entmax15	pos	pos	great excellent couldn that besides hurt didn that horrible is china Pitts gives us Redmon stop is not for t
Select. sparsemax	pos	pos	excellent great couldn gives besides china hurt that is
Bernoulli	pos	neg	an excellent movie another dumb horrible unacceptable sad remorseful movie great br br Pitts
HardKuma	pos	pos	excellent movie depicts America dumb horrible a great gives us besides our Pitts
Joint E and L	pos	pos	great excellent

(negative) I don ' t remember " Barnaby Jones " being no more than a very bland , standard detective show in which , as per any Quinn Martin show , Act I was the murder , Act II was the lead character figuring out the murder , Act III was the plot twist (another character murdered), Act IV was the resolution and the Epilogue was Betty (Lee Meriwether) asking her father - in - law Barnaby Jones (Buddy Ebsen) how he figured out the crime and then someone saying something witty at the end of the show . < br > < br > One thing I do remember was the late , great composer Jerry Goldsmith ' s excellent theme song . Strangely , the opening credit sequence made me want to see the show off and on for the seven seasons the show was on the air . I will also admit that it was nice to see Ebsen in a role other than Jed Clampett despite Ebsen being badly miscast . I just wished the show was more entertaining than when I first remembered it . < br > < br > Update (1 / 11 / 2009): I watched an interview with composer Jerry Goldsmith on YouTube through their Archive of American Television channel . Let ' s just say that I was more kind than Goldsmith about the show " Barnaby Jones ."

EXPLAINER	y_C	y_L	EXPLANATION
Erasure	neg	pos	wished excellent remembered miscast Strangely
Top-k gradient	neg	neg	miscast excellent remembered it badly
Top-k softmax	neg	pos	wished excellent remembered miscast figuring
Top-k 1.5-entmax	neg	neg	wished remembered Strangely miscast excellent
Top-k sparsemax	neg	neg	Strangely miscast wished badly excellent
Select. entmax15	neg	neg	wished remembered Strangely miscast excellent admit bland no character figuring say badly figured credit , the < the witty want just thing <
Select. sparsemax	neg	neg	Strangely miscast wished badly excellent remembered bland
Bernoulli	neg	neg	very bland , lead character plot character Epilogue witty show br late composer excellent theme song Strangely seasons nice badly miscast entertaining remembered br (1 / composer American Television
HardKuma	neg	neg	bland figuring saying excellent Strangely credit admit miscast wished remembered (1 11
Joint E and L	neg	neg	bland badly something

(positive) Yes ... I ' m going with the 1 - 0 on this and here ' s why . In the last few years , I have watched quite a few comedies and only left with a few mild laughs and a couple video rental late fees because the movies were that easy to forget . Then I stumble upon " Nothing ". Looked interesting , wasn ' t expecting much though . I was wrong . This was probably one of the funniest movies I have ever had the chance to watch . Dave and Andrew make a great comedic pair and the humor was catchy enough to remember , but not over complex to the point of missing the joke . I don ' t want to remark on any of the actual scenes , because I do feel this is a movie worth seeing for once . With more and more pointless concepts coming into movies (you know , like killer military jets and " fresh " remakes that are ruining old classics), This movie will make you happy to say it ' s OK to laugh at " Nothing ".

EXPLAINER	y_C	y_L	EXPLANATION
Erasure	pos	pos	funniest worth great wrong pointless
Top-k gradient	pos	pos	comedic funniest OK worth joke
Top-k softmax	pos	pos	funniest worth great wrong pointless
Top-k 1.5-entmax	pos	pos	funniest great wrong worth not
Top-k sparsemax	pos	pos	funniest worth great catchy wrong
Select. entmax15	pos	neg	funniest great wrong worth not catchy do probably pointless easy feel ruining movie OK joke ever Yes seeing stumble comedic mild don wasn enough), forget because 0 for
Select. sparsemax	pos	neg	funniest worth great catchy wrong ruining 0 feel easy OK not pointless
Bernoulli	neg	neg	- few comedies few mild laughs couple movies stumble interesting wrong probably funniest movies Dave great comedic humor catchy joke scenes movie pointless movies fresh remakes ruining movie Nothing ".
HardKuma	neg	neg	0 stumble wrong probably one funniest great catchy not joke a movie worth seeing pointless ruining OK Nothing
Joint E and L	pos	neg	funniest pointless worth

(negative) I ' m not to keen on The Pallbearer , it ' s not too bad , but just very slow at the times . As the movie goes on , it gets a little more interesting , but nothing brilliant . I really like David Schwimmer and I think he ' s good here . I ' m not a massive Gwyneth Paltrow fan , but I don ' t mind her sometimes and she ' s okay here . The Pallbearer is not a highly recommended movie , but if you like the leads then you might enjoy it .

EXPLAINER	y_C	y_L	EXPLANATION
Erasure	neg	pos	brilliant slow recommended nothing good
Top-k gradient	neg	pos	not nothing recommended slow brilliant
Top-k softmax	neg	pos	brilliant slow nothing recommended good
Top-k 1.5-entmax	neg	neg	slow brilliant nothing not recommended
Top-k sparsemax	pos	pos	slow brilliant nothing recommended good
Select. entmax15	neg	pos	slow brilliant nothing not recommended good enjoy highly very if you goes don okay , little it bad gets really
Select. sparsemax	pos	pos	slow brilliant nothing recommended good enjoy very bad highly
Bernoulli	neg	neg	Pallbearer , too bad slow times movie , brilliant good massive okay Pallbearer highly movie enjoy
HardKuma	neg	pos	slow nothing brilliant good okay highly recommended might enjoy
Joint E and L	neg	neg	nothing bad slow okay highly

Table A.14: Examples of extracted explanations for IMDB.

(positive) Ok , when I rented this several years ago I had the worst expectations . Yes , the acting isn ' t great , and the picture itself looks dated , but as I sat there , a strange thing happened . I started to like it . The action is great and there are few scenes that make you jump . Brion James , maybe one of the greatest B - grade actors next to Bruce Campbell , is great as always . The story isn ' t bad either . Now I wouldn ' t rush out and buy it , but you won ' t waste your time at least watching this good b - grade post apocalyptic western .

EXPLAINER	y_C	y_L	EXPLANATION
Erasure	pos	pos	good great great grade waste
Top-k gradient	pos	neg	waste worst greatest grade t
Top-k softmax	pos	neg	good great great worst grade
Top-k 1.5-entmax	pos	pos	great waste great good greatest
Top-k sparsemax	pos	neg	great waste great good grade
Select. entmax15	pos	pos	great waste great good greatest great always Ok apocalyptic Yes make buy t grade isn worst but wouldn strange is
Select. sparsemax	pos	neg	great waste great good grade greatest your worst Yes Ok
Bernoulli	neg	neg	worst , acting , looks strange great scenes greatest actors great story bad , waste watching good apocalyptic western
HardKuma	pos	neg	worst great great always waste good apocalyptic
Joint E and L	pos	neg	great worst

(negative) I have read each and every one of Baroness Orczy ' s Scarlet Pimpernel books . Counting this one , I have seen 3 pimpernel movies . The one with Jane Seymour and Anthony Andrews i preferred greatly to this . It goes out of its way for violence and action , occasionally completely violating the spirit of the book . I don ' t expect movies to stick directly to plots , i gave up being that idealistic long ago , but if an excellent movie of a book has already been made , don ' t remake it with a tv movie that includes excellent actors and nice costumes , but a barely decent script . Sticking with the 80 ' s version .... Rahne

EXPLAINER	y_C	y_L	EXPLANATION
Erasure	neg	pos	excellent excellent script barely decent
Top-k gradient	neg	neg	barely decent script if but
Top-k softmax	neg	pos	excellent excellent script decent barely
Top-k 1.5-entmax	neg	pos	barely excellent excellent have Sticking
Top-k sparsemax	neg	pos	excellent excellent barely pimpernel decent
Select. entmax15	neg	pos	barely excellent excellent have Sticking preferred decent . It don to t script way if costumes Counting pimpernel Rahne , nice greatly t have
Select. sparsemax	neg	pos	excellent excellent barely pimpernel decent preferred nice t Sticking It
Bernoulli	pos	pos	Baroness Orczy pimpernel movies greatly occasionally movies plots excellent movie tv excellent actors nice costumes barely decent script ....
HardKuma	neg	pos	have pimpernel preferred way excellent excellent barely decent Sticking Rahne
Joint E and L	neg	neg	barely expect decent preferred completely

(negative) While I agree that this was the most horrendous movie ever made , I am proud to say I own a copy simply because myself and a bunch of my friends were extras (mostly in the dance club scenes , but a few others as well . This movie had potential with Bolo and the director of Enter the Dragon signed on , but as someone who was on set most every day I can tell you that Robert Clouse was an old and confused individual , at least during the making of this movie . It was a wonder he could find his way to the set everyday . I would also like to think that this might have been a better movie if a lot of it had not been destroyed in a fire at Morning Calm studios . I can ' t say that it would have been for sure , but it would be nice to think so . I was actually surprised that it was ever released , and that someone like Bolo would attach his name to it without a fight . Oh well . Also look at the extras for pro wrestler Scott Levy , AKA Raven . He was a wrestler in Portland at the time ... nice guy , very smart .

EXPLAINER	y_C	y_L	EXPLANATION
Erasure	neg	pos	horrendous well well nice nice
Top-k gradient	neg	pos	well horrendous this well very
Top-k softmax	neg	pos	horrendous well Oh well nice
Top-k 1.5-entmax	neg	neg	horrendous Oh surprised had agree
Top-k sparsemax	neg	pos	horrendous smart nice Oh had
Select. entmax15	neg	pos	horrendous Oh surprised had agree nice smart others ever well ever but most nice movie proud like wonder . way few without . find but It making well actually be everyday
Select. sparsemax	neg	pos	horrendous smart nice Oh had ever ever few . wonder nice
Bernoulli	neg	pos	most horrendous bunch extras mostly scenes few This movie old movie everyday lot nice extras wrestler wrestler nice guy
HardKuma	neg	neg	horrendous bunch few confused wonder Oh guy very smart
Joint E and L	neg	neg	horrendous without

(positive) Having read some of the other comments here I was expecting something truly awful but was pleasantly surprised . REALITY CHECK : The original series wasn ' t that good . I think some people remember it with more affection than it deserved but apart from the car chases and Daisy Duke ' s legs the scripts were weak and poorly acted . The Duke boys were too intelligent and posh for backwood hicks , the shrunken Boss Hog was too cretinous to be evil and Rosco was just hyper throughout every screen moment . It ' s amazing the series actually lasted as long as it did because it ran out of story lines during the first series . < br > < br > Back to the movie . If you watch this film in it ' s own right , not as a direct comparison to however you remember the TV series , then it ' s not bad at all . The real star is of course the General Lee . The car chases and stunts are excellent and that ' s really what D . O . H . is all about . Johnny Knoxville is his usual eccentric self and along with Seann William Scott as Cousin Bo the pair make this film really funny in a hilarious Dumb - And - Dumber sort of way the TV series never achieved . The lovely Jessica Simpson is a natch as Miss Daisy , Burt Reynolds makes a much improved Boss Hog and M . C . Gainey makes a believably nasty Rosco P . Coltrane , the way he always should have been . < br > < br > If you don ' t like slapstick humour and crazy car stunts then you wouldn ' t be watching this film anyway because you should know what to expect . Otherwise if you want an entertaining car - action movie with a few good laughs that ' s not too taxing on the brain then go see this enjoyable romp with an open mind .

EXPLAINER	y_C	y_L	EXPLANATION
Erasure	pos	pos	horrendous well well nice nice
Top-k gradient	pos	neg	well horrendous this well very
Top-k softmax	pos	pos	horrendous well Oh well nice
Top-k 1.5-entmax	pos	pos	horrendous Oh surprised had agree
Top-k sparsemax	pos	neg	horrendous smart nice Oh had
Select. entmax15	pos	pos	horrendous Oh surprised had agree nice smart others ever well ever but most nice movie proud like wonder . way few without . find but It making well actually be everyday
Select. sparsemax	pos	pos	horrendous smart nice Oh had ever ever few . wonder nice
Bernoulli	pos	pos	most horrendous bunch extras mostly scenes few This movie old movie everyday lot nice extras wrestler wrestler nice guy
HardKuma	neg	neg	horrendous bunch few confused wonder Oh guy very smart
Joint E and L	pos	pos	horrendous without

Table A.15: (continuation) Examples of extracted explanations for IMDB.

(entailment)
Premise: Children and adults swim in large pool with red staircase .
Hypothesis: A group of people are swimming .

EXPLAINER	y_C	y_L	EXPLANATION
Erasure	ent	ent	swim pool staircase adults
Top-k gradient	ent	con	adults pool swim large
Top-k softmax	ent	ent	swim pool large staircase
Top-k 1.5-entmax	ent	ent	swim pool large staircase
Top-k sparsemax	ent	ent	swim pool large adults
Select. entmax15	ent	ent	swim pool large staircase adults Children in and with
Select. sparsemax	ent	ent	swim pool large adults in
Bernoulli	ent	ent	Children and adults swim in large pool with red staircase .
HardKuma	ent	con	swim large pool staircase
Joint E and L	ent	con	pool swim staircase

(contradiction)
Premise: A group of Asian children are gathered around in a circle listening to an older male in a white shirt .
Hypothesis: A man is wearing a black shirt .

EXPLAINER	y_C	y_L	EXPLANATION
Erasure	con	ent	Asian white male children
Top-k gradient	con	ent	circle children gathered to
Top-k softmax	con	ent	Asian white male children
Top-k 1.5-entmax	con	con	white older a male
Top-k sparsemax	con	con	a male shirt Asian
Select. entmax15	con	con	white older a male Asian listening circle shirt of children around gathered a an group in in . A to are
Select. sparsemax	con	con	a male shirt Asian . an
Bernoulli	ent	ent	A group of Asian children are gathered around in a circle listening to an older male in a white shirt .
HardKuma	con	ent	group Asian male white shirt
Joint E and L	con	con	male group white

(contradiction)
Premise: A woman is pushing her bike with a baby carriage in front .
Hypothesis: A woman is pushing groceries in a cart .

EXPLAINER	y_C	y_L	EXPLANATION
Erasure	con	con	baby woman bike pushing
Top-k gradient	con	con	carriage bike her with
Top-k softmax	con	neu	baby woman carriage pushing
Top-k 1.5-entmax	con	con	carriage woman her baby
Top-k sparsemax	ent	con	baby carriage woman front
Select. entmax15	con	con	carriage woman her baby pushing front is A . a bike with
Select. sparsemax	ent	ent	baby carriage woman front pushing is
Bernoulli	con	con	A woman is pushing her bike with a baby carriage in front .
HardKuma	con	con	woman pushing bike carriage
Joint E and L	con	con	woman baby

(neutral)
Premise: A woman in a gray shirt working on papers at her desk .
Hypothesis: Lady working in her desk tensely to completed the task

EXPLAINER	y_C	y_L	EXPLANATION
Erasure	neu	neu	desk papers woman .
Top-k gradient	neu	neu	desk on shirt at
Top-k softmax	neu	neu	desk papers woman .
Top-k 1.5-entmax	neu	neu	desk papers working woman
Top-k sparsemax	neu	neu	desk papers woman working
Select. entmax15	neu	ent	desk papers working woman . on shirt her at in a
Select. sparsemax	neu	ent	desk papers woman working her A
Bernoulli	neu	neu	A woman in a gray shirt working on papers at her desk .
HardKuma	neu	neu	woman working papers at desk
Joint E and L	neu	neu	working desk woman papers

(neutral)
Premise: A brown dog with a blue muzzle is running on green grass .
Hypothesis: A mean dog is wearing a muzzle to keep it from attacking cats

EXPLAINER	y_C	y_L	EXPLANATION
Erasure	neu	neu	dog brown running muzzle
Top-k gradient	neu	neu	with brown on green
Top-k softmax	neu	neu	dog brown running blue
Top-k 1.5-entmax	con	con	dog blue brown muzzle
Top-k sparsemax	neu	neu	dog muzzle with is
Select. entmax15	con	neu	dog blue brown muzzle running is . A grass green with on a
Select. sparsemax	neu	neu	dog muzzle with is A a running on brown
Bernoulli	neu	con	A brown dog with a blue muzzle is running on green grass .
HardKuma	neu	neu	dog muzzle running
Joint E and L	neu	neu	dog running muzzle

Table A.16: Examples of extracted explanations for SNLI.

(contradiction)
Premise: A man sits at a table in a room .
Hypothesis: A woman sits .

EXPLAINER	y_C	y_L	EXPLANATION
Erasure	con	ent	sits table . at
Top-k gradient	con	ent	. sits table A
Top-k softmax	con	ent	sits table . room
Top-k 1.5-entmax	con	ent	table . sits man
Top-k sparsemax	con	ent	man sits A at
Select. entmax15	con	ent	table . sits man A room a a at in
Select. sparsemax	con	ent	man sits A at in a a
Bernoulli	con	ent	A man sits at a table in a room .
HardKuma	con	con	man sits at
Joint E and L	con	con	man
Human Highlights	con	ent	man

(entailment)
Premise: Elderly woman climbing up the stairs .
Hypothesis: The old lady was walking up the stairs .

EXPLAINER	y_C	y_L	EXPLANATION
Erasure	ent	ent	stairs woman Elderly climbing
Top-k gradient	ent	con	Elderly stairs . the
Top-k softmax	ent	con	stairs woman Elderly climbing
Top-k 1.5-entmax	ent	con	stairs Elderly woman climbing
Top-k sparsemax	ent	con	stairs Elderly woman climbing
Select. entmax15	ent	con	stairs Elderly woman climbing up . the
Select. sparsemax	ent	con	stairs Elderly woman climbing the
Bernoulli	con	con	Elderly woman climbing up the stairs .
HardKuma	ent	con	Elderly woman climbing up stairs
Joint E and L	ent	ent	stairs Elderly climbing woman
Human Highlights	ent	con	Elderly woman climbing

(entailment)
Premise: A woman with a blond ponytail and a white hat is riding a white horse , inside a fence with a horned cow .
Hypothesis: The woman is riding a horse .

EXPLAINER	y_C	y_L	EXPLANATION
Erasure	ent	con	horse riding . fence
Top-k gradient	ent	ent	cow horse fence riding
Top-k softmax	ent	ent	horse riding fence cow
Top-k 1.5-entmax	ent	con	horse riding woman a
Top-k sparsemax	ent	con	horse riding a is
Select. entmax15	ent	con	horse riding woman a cow fence horned a is ponytail , a with inside blond A . hat
Select. sparsemax	ent	con	horse riding a is with A ,
Bernoulli	ent	con	A woman with a blond ponytail and a white hat is riding a white horse , inside a fence with a horned cow .
HardKuma	ent	con	woman ponytail riding horse inside horned cow
Joint E and L	ent	ent	cow horse fence inside
Human Highlights	ent	ent	woman blond horse fence horned cow

(contradiction)
Premise: A woman in a black coat eats dinner while her dog looks on .
Hypothesis: A woman is wearing a blue coat .

EXPLAINER	y_C	y_L	EXPLANATION
Erasure	con	ent	coat black woman dog
Top-k gradient	con	ent	dog eats black looks
Top-k softmax	con	ent	coat black woman dinner
Top-k 1.5-entmax	con	con	black coat woman dog
Top-k sparsemax	con	con	black a woman A
Select. entmax15	con	con	black coat woman dog a looks in . dinner eats her A on while
Select. sparsemax	con	ent	black a woman A coat in her
Bernoulli	con	ent	A woman in a black coat eats dinner while her dog looks on .
HardKuma	con	ent	woman black coat
Joint E and L	con	con	woman black
Human Highlights	con	con	black

Table A.17: (continuation) Examples of extracted explanations for SNLI.

B

Supplemental Material for Chapter 4

B.1 Datasets

The revised IMDB and SNLI datasets, which we refer to as rIMDB and rSNLI respectively, were created by Kaushik et al. (2020). They contain counterfactuals consisting of revised versions made by humans on the Amazon's Mechanical Turk crowdsourcing platform. For both datasets, the authors ensure that (a) the counterfactuals are valid; (b) the edited reviews are coherent; and (c) the counterfactuals do not contain unnecessary modifications. For SNLI, counterfactuals were created either by revising the premise or the hypothesis. We refer to (Kaushik et al., 2020) for more details on the data generation process. Table B.1 presents statistics for the datasets used for training models in this work.

	Train		Val.		Test	
Dataset	**docs**	**tokens**	**docs**	**tokens**	**docs**	**tokens**
IMDB	22.5K	6M	2.5K	679K	25K	6M
Revised IMDB	3414	629K	490	92K	976	180K
SNLI	549K	12M	10K	232K	10K	231K
Revised SNLI	4165	188K	500	24K	1000	48K

Table B.1: Datasets statistics.

Additionally, we incorporate various contrastive and out-of-domain datasets to evaluate our models. For IMDB, we use the contrast IMDB (Gardner et al., 2020), RottenTomatoes (Pang and Lee, 2005), SST-2 (Socher et al., 2013), Amazon Polarity and Yelp (Zhang et al., 2015). For SNLI, we evaluate on the Hard SNLI (Gururangan et al., 2018), break (Glockner et al., 2018), MultiNLI (Williams et al., 2018), and Adversarial NLI (Nie et al., 2020). We refer to the original works for more details.

B.2 CREST Details

B.2.1 Masker

For all datasets, the masker consists of a SPECTRA rationalizer that uses a T5-small encoder as the backbone for the encoder and predictor (see §4.2.1). Our implementation is derived directly from its original source (Guerreiro and Martins, 2021). We set the maximum sequence length to 512, truncating inputs when necessary. We employ a contiguity penalty of 10^{-4} for IMDB and 10^{-2} for SNLI. We train all models for a minimum of 3 epochs and maximum of 15 epochs along with early stopping with a patience of 5 epochs. We use AdamW (Loshchilov and Hutter, 2019) with a learning rate of 10^{-4} and weight decay of 10^{-6}.

B.2.2 Editor

For all datasets, CREST and MiCE editors consist of a full T5-small model (Raffel et al., 2020), which includes both the encoder and the decoder modules. We use the T5 implementation available in the *transformers* library (Wolf et al., 2020) for our editor. We train all models for a minimum of 3 epochs and maximum of 20 epochs along with early stopping with a patience of 5 epochs. We use AdamW (Loshchilov and Hutter, 2019) with a learning rate of 10^{-4} and weight decay of 10^{-6}. For both CREST and MiCE, we generate counterfactuals with beam search with a beam of size 15 and disabling bigram repetitions. We post-process the output of the editor to trim spaces and repetitions of special symbols (e.g., `</s>`).

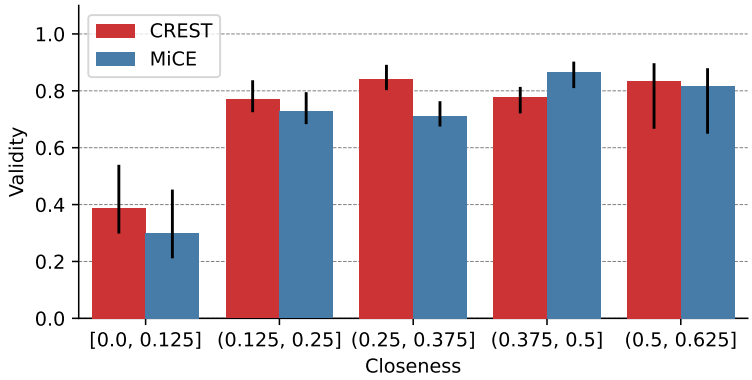


Figure B.1: Validity by binned closeness ranges for MiCE (30% masking) and CREST (30% masking). At lower closeness ranges, CREST produces more valid counterfactuals than does MiCE.

B.2.3 SPECTRA rationalizers

All of our SPECTRA rationalizers share the same setup and training hyperparameters as the one used by the masker in §4.4, but were trained with distinct random seeds. We tuned the counterfactual loss weight α within $\{1.0, 0.1, 0.01, 0.001, 0.0001\}$, and λ within $\{1.0, 0.1, 0.01, 0.001\}$ for models trained with agreement rationalization. More specifically, we performed hyperparameter tuning on the validation set, with the goal of maximizing in-domain accuracy. As a result, we obtained $\alpha = 0.01$ and $\lambda = 0.001$ for IMDB, and $\alpha = 0.01$ and $\lambda = 0.1$ for SNLI.

B.3 Validity vs. Closeness

To better assess the performance of CREST and MiCE by varying closeness, we plot in Figure B.1 binned-validity scores of CREST and MiCE with 30% masking on the revised SNLI dataset. Although CREST is deemed less valid than MiCE overall (*cf.* Table 4.1), we note that CREST generates more valid counterfactuals in lower minimality ranges. This provides further evidence that CREST remains superior to MiCE on closeness intervals of particular interest for generating counterfactuals in an automatic way.

B.4 Human Annotation

The annotation task was conducted by four distinct individuals, all of whom are English-fluent PhD students. Two annotators were employed for IMDB and two for SNLI. The annotators were not given any information regarding the methods used to create each counterfactual, and the documents were presented in a random order to maintain source anonymity. The annotators were presented with the reference text and its corresponding gold label. Subsequently, for each method, they were asked to assess both the validity and the naturalness of the resulting counterfactuals using a 5-point Likert scale. We provided a guide page to calibrate the annotators' understating of validity and naturalness prior the annotation process. We presented hypothetical examples with different levels of validity and naturalness and provided the following instructions regarding both aspects:

- "If every phrase in the text unequivocally suggests a counterfactual label, the example is deemed fully valid and should receive a top score of 5/5."

Method	IMDB			SNLI		
	v	n	r_o	v	n	r_o
Manual	4.60	4.36	0.83	4.89	4.90	0.95
MiCE	2.76	2.29	0.71	4.35	4.71	0.94
CREST	4.06	3.44	0.76	4.89	4.89	0.96
Overall	3.81	3.36	0.77	4.71	4.83	0.95

Table B.2: Annotation statistics. v and n represent the averaged validity and naturalness scores, whereas r_o is the relative observed agreement computed with a normalized and inverted MAD.

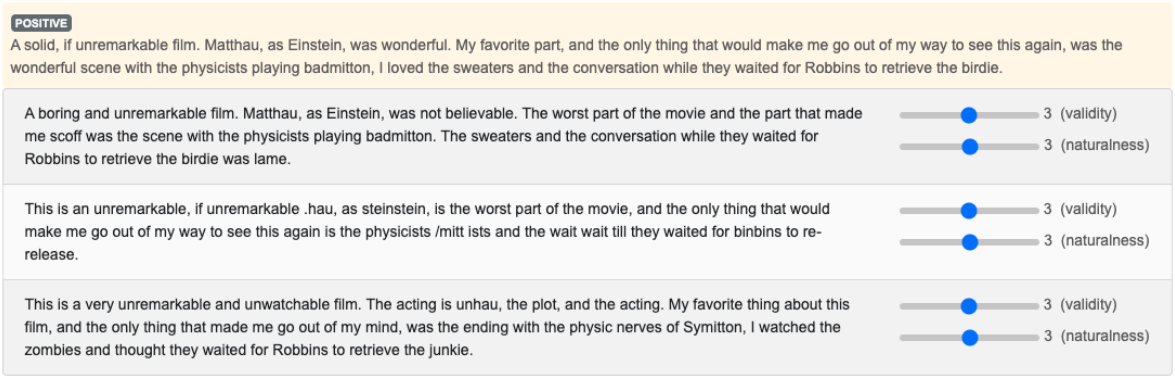


Figure B.2: Snapshot of the annotation interface.

- "If the counterfactual text aligns with the style, tone, and grammar of real-world examples, it's considered highly natural and deserves a score of 5/5."

We measure inter-annotator agreement with a normalized and inverted Mean Absolute Difference (MAD), which computes a "soft" accuracy by averaging absolute difference ratings and normalizing them to a 0-1 range. We present the annotation results in Table B.2. Our results show that humans agreed more on manual examples than on automatic approaches. On the other hand, for SNLI, annotators assigned similar scores across all methods. In terms of overall metrics, including validity, naturalness, and agreement, the scores were lower for IMDB than for SNLI, highlighting the difficulty associated with the generation of counterfactuals for long movie reviews.

Annotation interface. Figure B.2 shows a snapshot of the interface used for the annotation, which is publicly available at https://www.github.com/mtreviso/TextRankerJS.

B.5 Counterfactual Data Augmentation Analysis

Previous studies on counterfactual data augmentation have found that model performance highly depends on the number and diversity of augmented samples (Huang et al., 2020; Joshi and He, 2022). To account for this, we investigate the effect of adding increasingly larger portions of CREST counterfactuals for data augmentation on the IMDB dataset. Our findings are summarized in Table B.3.

Setup	Data size	RotTom	SST-2	Amazon	Yelp
F	100%	76.5 ± 1.6	79.8 ± 1.6	86.0 ± 0.7	88.5 ± 0.7
With data augmentation:					
$F + C_H$	+8%	76.6 ± 1.5	80.7 ± 1.3	86.3 ± 1.0	89.1 ± 1.2
$F + C_{S,V}$	+1%	77.2 ± 1.1	80.5 ± 0.5	86.1 ± 0.2	88.8 ± 0.3
$F + C_{S,V}$	+2%	76.2 ± 1.2	<u>80.8</u> ± 0.8	86.7 ± 0.5	<u>89.6</u> ± 0.5
$F + C_{S,V}$	+4%	**77.7** ± 0.8	<u>80.8</u> ± 0.7	**87.0** ± 0.6	**89.8** ± 0.6
$F + C_{S,V}$	+8%	76.6 ± 2.2	80.2 ± 1.7	86.1 ± 0.9	88.2 ± 1.0
$F + C_{S,V}$	+85%	76.8 ± 0.9	79.3 ± 0.3	85.2 ± 0.9	88.0 ± 1.0
$F + C_S$	+100%	76.7 ± 1.0	80.6 ± 0.6	86.4 ± 0.6	89.1 ± 0.5
With agreement regularization:					
F & $C_{S,V}$	85%	76.3 ± 1.4	80.2 ± 1.3	86.3 ± 0.7	88.9 ± 0.7
F & C_S	100%	<u>77.3</u> ± 2.3	**81.1** ± 2.4	<u>86.8</u> ± 0.8	89.3 ± 0.7

Table B.3: OOD accuracy of SPECTRA rationalizers with different portions of augmented counterfactuals. Values in **Bold**: top results; <u>underlined</u>: second-best.

Discussion. We find that incorporating human-crafted counterfactuals ($F + C_H$) improves SPECTRA performance on all OOD datasets. On top of that, we note that using a small proportion (4% of the full IMDB) of valid CREST counterfactuals ($F + C_{S,V}$) through data augmentation also leads to improvements on all datasets and outweighs the benefits of manual counterfactuals. This finding confirms that, as found by PolyJuice (Wu et al., 2021), synthetic counterfactuals can improve model robustness. Conversely, as the number of augmented counterfactuals increases (85%), the performance on OOD datasets starts to decrease, which is also consistent with the findings of Huang et al. (2020). When augmenting the entire training set we obtain an increase of accuracy, suggesting that the counterfactual loss weight (α) was properly adjusted on the validation set. Finally, we observe that while applying CREST-Rationalization only on valid examples (F & $C_{S,V}$) degrades performance, applying CREST-Rationalization on all paired examples (F & C_S) maintains a high accuracy on OOD datasets and concurrently leads to strong results on in-domain and contrast datasets (see Table 4.2).

B.6 Examples of Counterfactuals

Table B.4 shows examples of counterfactuals produced by MiCE and CREST, both with 30% masking.

	Sentiment Classification:
Input:	If you haven't seen this, it's terrible. It is pure trash. I saw this about 17 years ago, and I'm still screwed up from it.
MiCE:	If you haven't seen this, it's a great movie. I saw this about 17 years ago, and I'm still screwed up from it.
CREST:	If you haven't seen this movie, it's worth seeing. It's very funny. I saw it about 17 years ago, and I'm still screwed up from it.
Input:	Touching; Well directed autobiography of a talented young director/producer. A love story with Rabin's assassination in the background. Worth seeing !
MiCE:	Watching abiography of a very young writer/producer. A great story of Rabin's assassination in the background! Worth seeing!!
CREST:	This is the worst film of a talented young director/producer. And Rabin's assassination in the background is even worse!
Input:	A solid, if unremarkable film. Matthau, as Einstein, was wonderful. My favorite part, and the only thing that would make me go out of my way to see this again, was the wonderful scene with the physicists playing badmitton, I loved the sweaters and the conversation while they waited for Robbins to retrieve the birdie.
MiCE:	This is an unremarkable, if unremarkable .hau, as steinstein, is the worst part of the movie, and the only thing that would make me go out of my way to see this again is the physicists /mitt ists and the wait wait till they waited for binbins to re-release.
CREST:	This is a very unremarkable and unwatchable film. The acting is unhau, the plot, and the acting. My favorite thing about this film, and the only thing that made me go out of my mind, was the ending with the physic nerves of Symitton, I watched the zombies and thought they waited for Robbins to retrieve the junkie.
Input:	I saw this film earlier today, and I was amazed at how accurate the dialog is for the main characters. It didn't feel like a film - it felt more like a documentary (the part I liked best). The leading ladies in this film seemed as real to me as any fifteen year-old girls I know. All in all, a very enjoyable film for those who enjoy independent films.
MiCE:	I saw this film earlier today, and I was amazed at how bad the film is for the sake of a film - it feels more like thanthe part I played in this film. To me - fifteen year-old s I don't know. All in all this is a bad film for those who like independent films :
CREST:	I saw this movie earlier today, and I was surprised at how bad it is for the first time. It's not a good movie - it's just a bad movie (the only thing I can say about it). The acting is awful to me as any fifteen year-old as I can. All in all, the movie is a waste of time for me.
	Natural Language Inference:
Prem:	A large group of people walking in a busy city at night.
Hyp:	People are outside in a park.
MiCE:	People are walking in a city at night
CREST:	People walking in a city.
Prem:	Players from two opposing teams wearing colorful cleats struggle to gain control over a ball on an AstroTurf field.
Hyp:	The players are playing a sport.
MiCE:	The players are playing chess at home
CREST:	The players are sitting on a couch.
Prem:	A woman is in the middle of hitting a tennis ball.
Hyp:	A woman is playing tennis.
MiCE:	A woman is playing basketball at home
CREST:	A woman is playing basketball.
Prem:	Two boys with blond-hair, wearing striped shirts on a bed.
Hyp:	Children playing in the park.
MiCE:	Children are on the bed.
CREST:	Boys are on the bed.
Prem:	Bubbles surround a statue in the middle of a street.
Hyp:	There are bubbles around the statue.
MiCE:	There are bubbles surround the statue.
CREST:	Bubbles are in the ocean.
Prem:	A young girl is standing in a kitchen holding a green bib.
Hyp:	A boy is playing with a firetruck.
MiCE:	A child is in a fire place
CREST:	A girl is holding a bib.

Table B.4: Examples of original inputs from the IMDB and SNLI datasets followed by synthetic counterfactuals produced by MiCE and CREST with 30% masking.

C

Supplemental Material for Chapter 5

C.1 Machine Translation

C.1.1 Setup

Data. Statistics for all datasets used in MT experiments can be found below in Table C.1.

DATASET	# TRAIN	# TEST	AVG. SENTENCE LENGTH
IWSLT17 (EN→DE)	206K	1080	20 ±14 / 19 ±13
IWSLT17 (EN→FR)	233K	1210	20 ±14 / 21 ±15

Table C.1: Statistics for MT datasets.

Training and Model. We replicated the sentence-level model of Fernandes et al. (2021) with the exception that we used α-entmax with $\alpha = 1.5$ instead of softmax in all attention heads and layers. Table C.2 shows some architecture (transformer large) and training hyperparameters used for MT experiments. We refer to the original work of Fernandes et al. (2021) for more training details.

HYPERPARAM.	VALUE
Hidden size	1024
Feedforward size	4096
Number of layers	6
Number of heads	16
Attention mapping π	1.5-entmax
Optimizer	Adam
Number of epochs	20
Early stopping patience	10
Learning rate	0.0005
Scheduling	Inverse square root
Linear warm-up steps	4000
Dropout	0.3
CoWord dropout	0.1
Beam size	5

Table C.2: Hyperparmeters for neural machine translation models.

C.1.2 Projections setup

Data. Statistics for the subsets of IWSLT used in the projection analysis can be found below in Table C.3.

	TRAIN			VALIDATION		
PAIR	# SENT.	# POS. PAIRS	AVG. SENT. LENGTH	# SENT.	# POS. PAIRS	AVG. SENT. LENGTH
EN→DE	9K	8M ±1M	35 ±16	1K	330K ±56K	36 ±17
EN→FR	9K	9M ±1M	37 ±17	1K	334K ±58K	37 ±16

Table C.3: Statistics for subsets of IWSLT used for training and evaluating projections.

Training. After extracting the α-entmax graphs, we optimize the learnable parameters of Equation 5.7 with Adam over a single epoch. Moreover, we used the k-means implementation from scikit-learn (Pedregosa et al.,

2011) for our clustering-based approach. The hyperparameters used both for training the projections and for clustering with k-means are shown in Table C.4.

HYPERPARAM.	VALUE
Projection dim. r	4
Loss margin ω	1.0
Batch size	16
Optimizer	Adam
Number of epochs	1
Learning rate	0.01
ℓ_2 regularization	0
k-means init	k-means++
k-means max num. inits	10
k-means max iters	300

Table C.4: Hyperparmeters for MT projections.

Projection analysis. We compare Sparsefinder, varying $B \in \{2, 4, 6, 8, 10, 12\}$ for bucket-based methods, and $t \in \{0.5, 1.0, 1.5, 2.0, 2.5\}$ for the distance-based variant, with the following methods:

- **Window baseline:** connect all query and key pairs within a sliding window of size $w \in \{0, 1, 3, 5, 7, 9, 11, 15, 19, 23, 27\}$.
- **Learnable patterns:** Reformer by varying the number of buckets within $\{2, 4, 6, 8, 10, 12\}$; Routing transformer by varying the number of clusters within $c \in \{2, 4, 6, 8, 10\}$ with top-k set to $\lceil n/c \rceil$ (i.e. balanced clusters).
- **Fixed patterns:** BigBird by varying the number of random blocks within $\{2, 4, 6, 8, 10\}$ with a block size of 1; Longformer by varying the number of random global tokens within $\{4, 8, 12, 16, 20\}$.

Sparsity-recall tradeoff per layer and head (EN→DE). Plots are shown in Figures C.1, C.2.

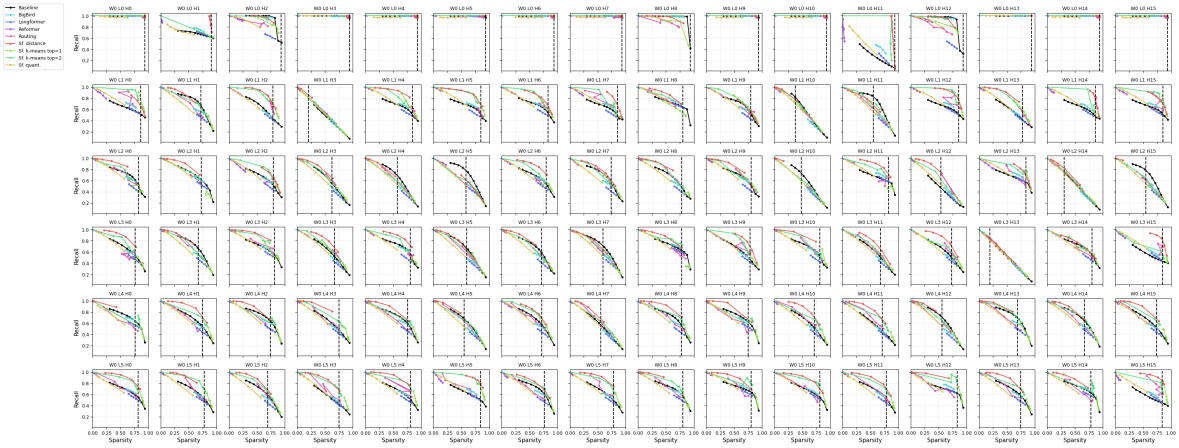


Figure C.1: Sparsity-recall tradeoffs without a fixed window pattern for EN→DE.

Sparsity-recall tradeoff per layer and head (EN→FR). Plots are shown in Figures C.3, C.4.

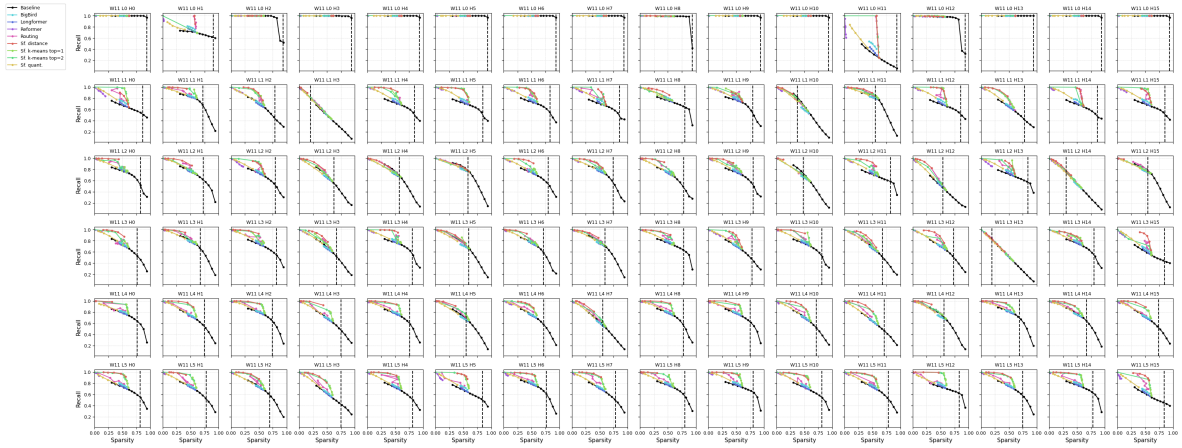


Figure C.2: Sparsity-recall tradeoffs with a fixed window pattern of size 11 for EN→DE.

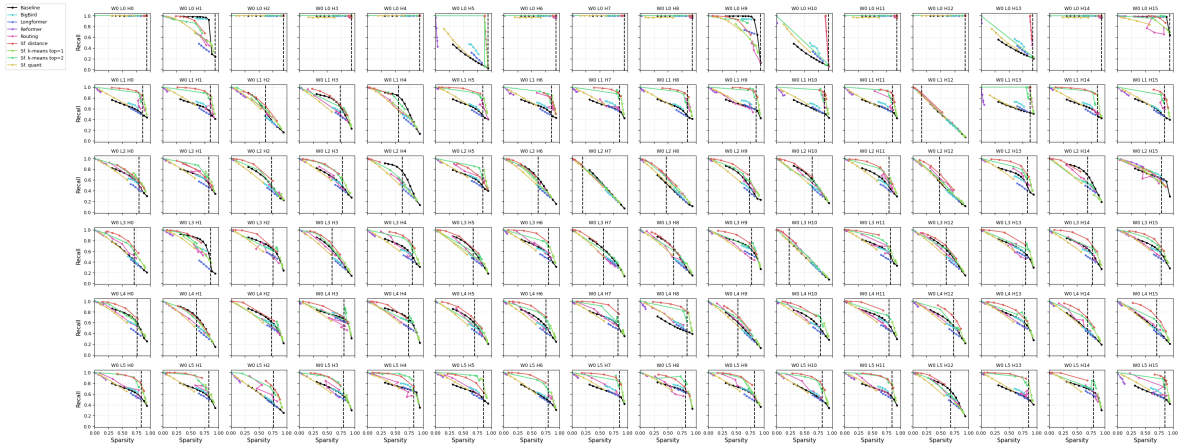


Figure C.3: Sparsity-recall tradeoffs without a fixed window pattern for EN→FR.

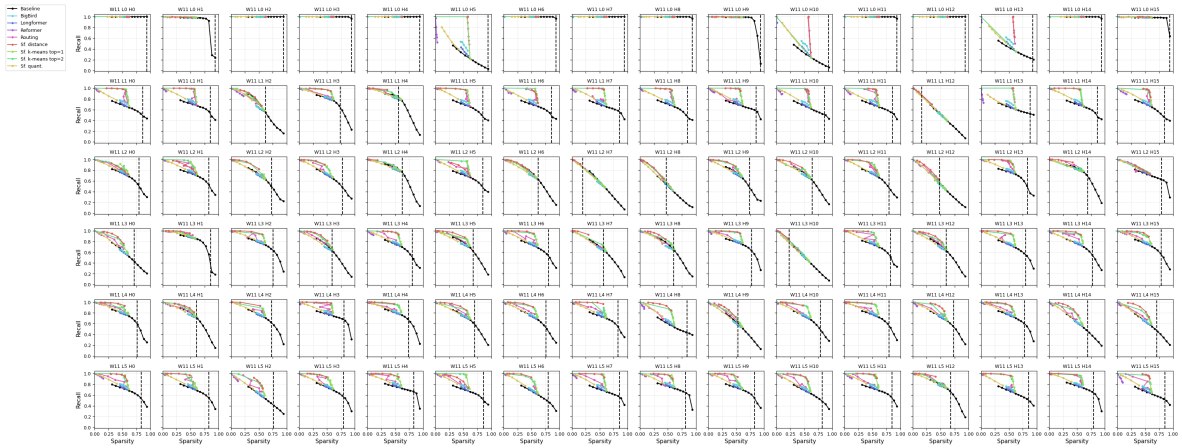


Figure C.4: Sparsity-recall tradeoffs with a fixed window pattern of size 11 for EN→FR.

C.2 Masked Language Modelling

C.2.1 Setup

Data and model. In order to have a transformer model trained with α-entmax, we finetuned RoBERTa-Base (Liu et al., 2019) on WikiText-103 (Merity et al., 2017) over 3000 steps with Adam (learning rate of

3×10^{-5}). To mimic the finetuning approach adopted by Longformer, we employed a batch size of 2 by accumulating gradients over 32 steps due to GPU memory constraints. At the end of training we obtained a perplexity of 1.2529 with an overall sparsity of 0.9804. Table C.5 shows some architecture (transformer large) and training hyperparameters used for MT experiments. We refer to the original work of Liu et al. (2019) for more architecture details.

HYPERPARAM.	VALUE
Hidden size	64
Feedforward size	3072
Max input length	514
Number of layers	12
Number of heads	12
Attention mapping π	1.5-entmax
Optimizer	Adam
Number of steps	3000
Learning rate	0.00003

Table C.5: Hyperparmeters for masked language modelling models.

C.2.2 Projections setup

Data and training. The subset used for Masked LM projections experiments contains 500 instances for training and 500 instances for validation. Moreover, all instances have a sentence length of 512 tokens. We got 3M (±1M) positive pairs for training and 2.5M (±1M) for validation. The hyperparameters for Masked LM are the same as the ones used in the MT experiments, shown in Table C.4.

Projection analysis. We perform the same analysis as in MT, but now we vary the window size of the baseline within {0, 1, 3, 7, 11, 25, 31, 41, 51, 75, 101, 125, 151, 175, 201, 251, 301, 351, 401, 451, 501, 512}.

Sparsity-recall tradeoff per layer and head. Plots are shown next.

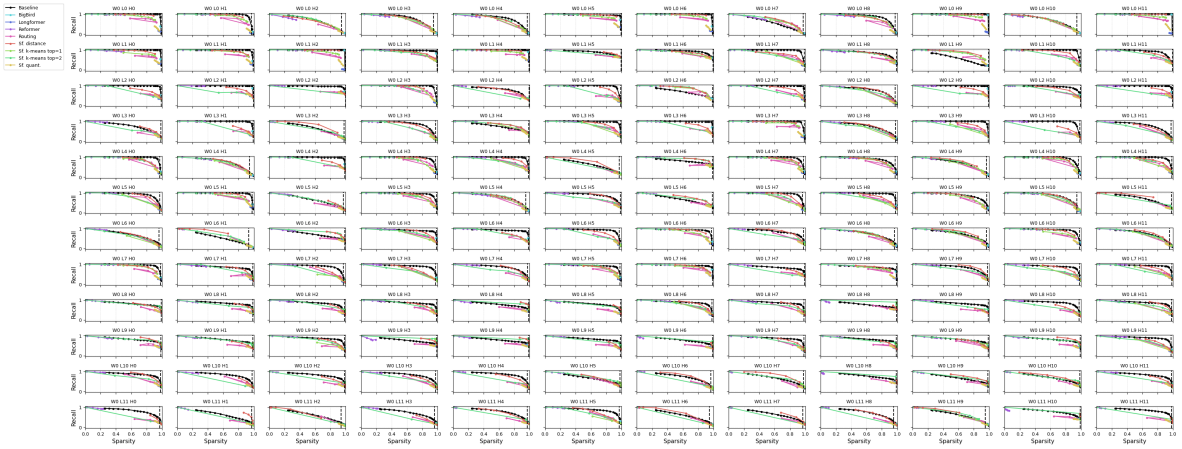


Figure C.5: Sparsity-recall tradeoffs without a fixed window pattern for MLM.

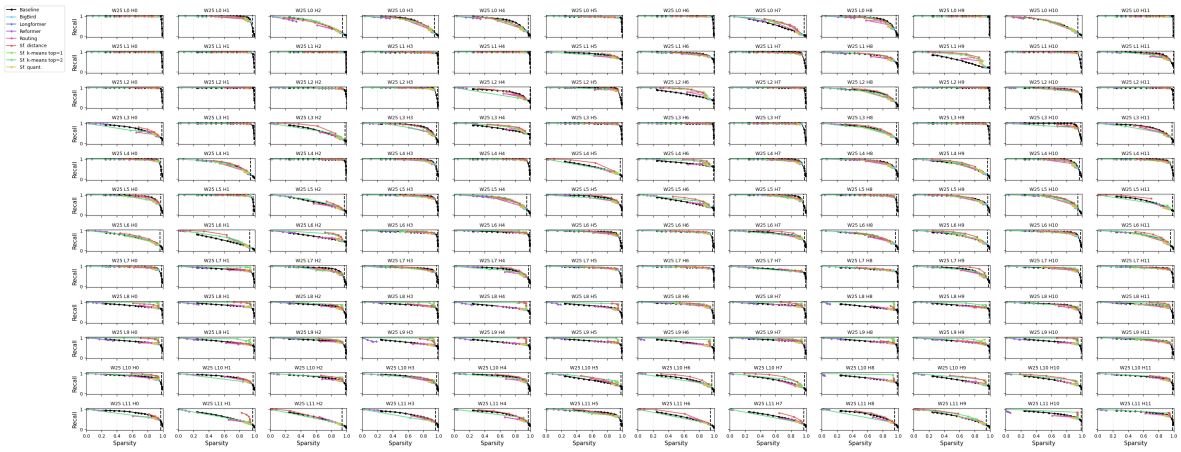


Figure C.6: Sparsity-recall tradeoffs with a fixed window pattern of size 25 for MLM.

C.3 Attention plots

Examples of attention maps can be seen in Figure C.7 and C.8.

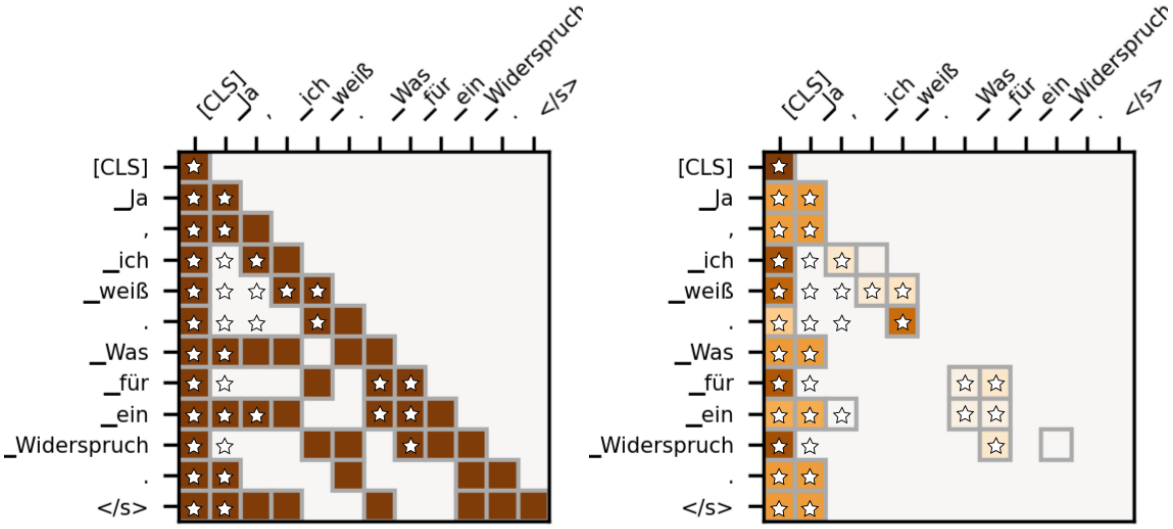


Figure C.7: Learned patterns by Sparsefinder k-means (left) and the subsequent attention weights (right). Starred blocks represent ground-truth edges.

C.4 Efficient Sparsefinder

Plots for block size within {1,2,4,8,16} are shown in C.9. For these experiments, we used a window size of 3 for all methods in order to better measure the impact of others hyper-parameters.

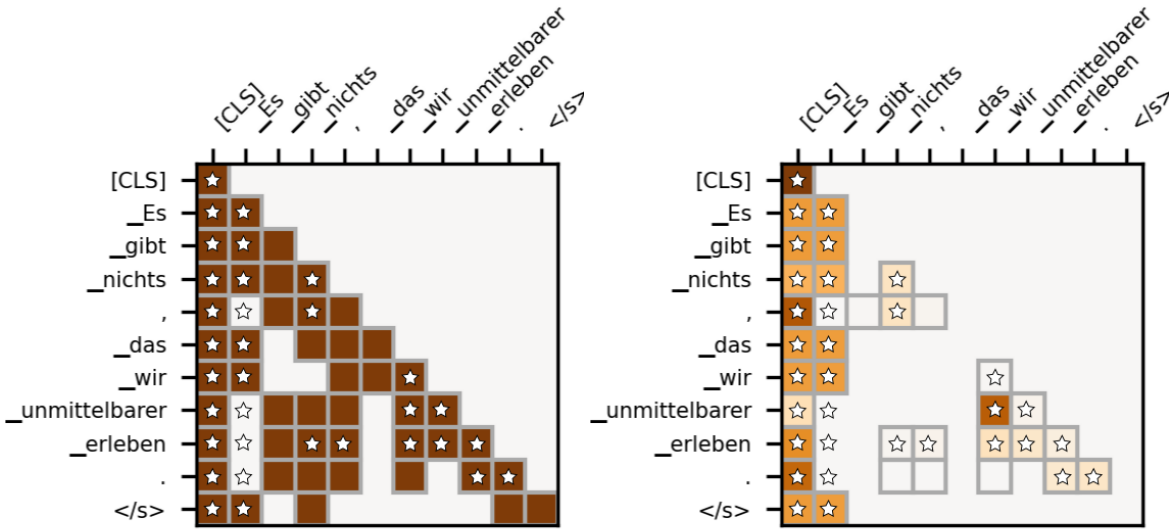


Figure C.8: Learned patterns by Sparsefinder k-means (left) and the subsequent attention weights (right). Starred blocks represent ground-truth edges.

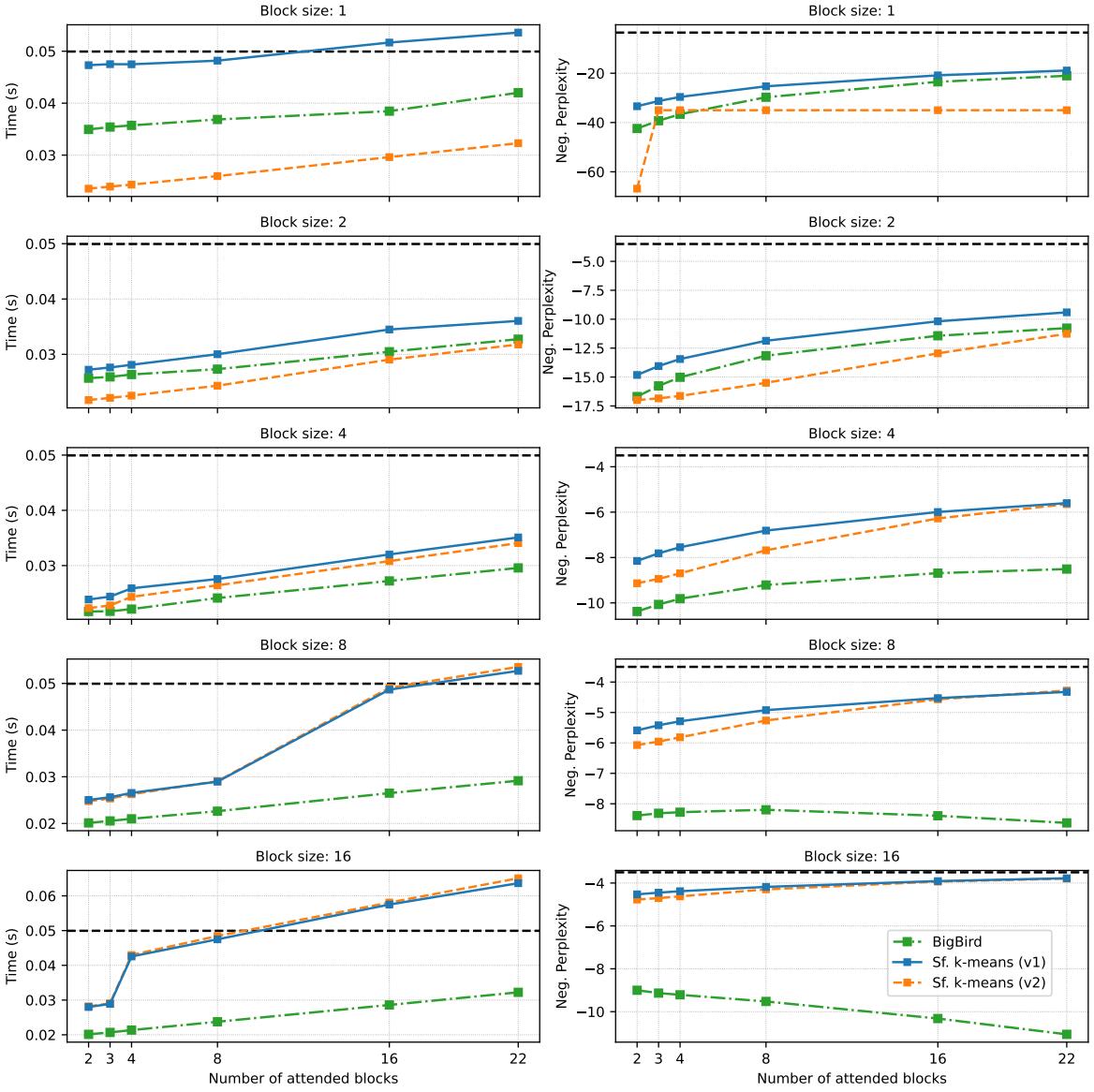


Figure C.9: Comparison between Sparsefinder and BigBird in terms of running time and (negative) perplexity as a function of the number of random blocks for several block sizes. The horizontal dashed line represents the results obtained by the full α-entmax transformer.

D

Supplemental Material for Chapter 6

D.1 Training hyperparameters

The hyperparameters used for training are shown in Table D.1.

HYPERPARAM.	XLM-R	XLM-R-M	REMBERT
Feed-forward size	1024	1024	1024
Batch size	2	2	1
Optimizer	Adam	Adam	Adam
Number of epochs	10	10	10
Early stopping patience	3	3	3
Encoder learning rate	1×10^{-4}	1×10^{-4}	3×10^{-5}
Feed-forward learning rate	1×10^{-4}	1×10^{-4}	1×10^{-5}
Gradient accumulation	4	4	8
Dropout	0.05	0.05	0.05

Table D.1: Hyperparmeters used for training sentence (constrained) and word-level (unconstrained) QE systems.

D.2 Full results for the constrained track

Following the analysis described in §6.5.1, we report the best results for each explainability method for XLM-R-based models in Table D.2 on the validation set of RO-EN and Table D.3 on the validation set of ET-EN. We also report the best explainers based on Attention $\times$ Norm for XLM-R-M and RemBERT-based models. For explainability methods based on attention weights, we show two attention heads: one with the best performance on source AUC and another with the best performance on target AUC. Besides submitting ensembled explanations, we also made submissions with Attention $\times$ Norm heads that achieve the top performance on the validation set of RO-EN and ET-EN.

			Source			Target		
#	ENCODER	EXPLAINER	AUC	AP	R@K	AUC	AP	R@K
1	XLM-R	Attention - Layer 18 - Head 3	0.6555	0.4569	0.3509	0.7894	0.7189	0.6054
2	XLM-R	Attention - Layer 18 - Head 0	0.7445	0.6353	0.5164	0.7462	0.6488	0.5197
3	XLM-R	Cross-attention - Layer 18 - Head 3	0.7092	0.5461	0.4139	0.8066	0.7378	0.6293
4	XLM-R	Cross-attention - Layer 18 - Head 0	0.7514	0.6345	0.5170	0.7374	0.6254	0.4883
5	XLM-R	Attention $\times$ Norm - Layer 18 - Head 3	0.7178	0.5686	0.4372	**0.8136**	**0.7432**	**0.6342**
6	XLM-R	Attention $\times$ Norm - Layer 19 - Head 2	**0.7851**	**0.6875**	**0.5701**	0.8099	0.7301	0.6153
7	XLM-R	Gradient $\times$ Hidden States - Layer 15	0.6949	0.5629	0.4399	0.6780	0.5388	0.4044
8	XLM-R	Gradient $\times$ Attention - Layer 17	0.7104	0.5942	0.4913	0.7618	0.6747	0.5628
9	XLM-R	Integrated Gradients - Layer 15	0.6539	0.5251	0.4059	0.6560	0.5148	0.3853
10	XLM-R	LIME	0.6470	0.5160	0.3922	0.5892	0.4576	0.3300
11	XLM-R	Leave-one-out	0.6970	0.5673	0.4409	0.5921	0.4752	0.3567
12	XLM-R	Relaxed-Bernoulli Rationalizer	0.4803	0.3638	0.2483	0.5434	0.4043	0.2914
13	XLM-R-M	Attention $\times$ Norm - Layer 23 - Head 3	0.6993	0.5824	0.4571	0.7686	0.6932	0.5932
14	XLM-R-M	Attention $\times$ Norm - Layer 23 - Head 1	0.7530	0.6612	0.5479	0.7612	0.6841	0.5802
15	RemBERT	Attention $\times$ Norm - Layer 23	0.7824	0.6987	0.5901	0.7904	0.6865	0.5723
16	RemBERT	Attention $\times$ Norm - Layer 22 - Head 5	0.7842	0.6822	0.5752	0.7167	0.5549	0.4278
1	Ensemble	(5) + (6) + (15)	0.8043	0.7137	0.5970	0.8398	0.7695	0.6606
2	Ensemble	(5) + (6) + (14) + (15)	**0.8074**	**0.7203**	**0.6071**	**0.8421**	**0.7725**	**0.6624**

Table D.2: Full constrained track results on the validation set of RO-EN.

#	ENCODER	EXPLAINER	Source			Target		
			AUC	AP	R@K	AUC	AP	R@K
1	XLM-R	Attention - Layer 18 - Head 3	0.6406	0.5205	0.3811	0.7094	0.6210	0.5037
2	XLM-R	Attention - Layer 18 - Head 0	0.6656	0.5619	0.4438	0.7055	0.6011	0.4779
3	XLM-R	Cross-attention - Layer 18 - Head 3	0.6587	0.5335	0.3947	0.7270	0.6396	0.5226
4	XLM-R	Cross-attention - Layer 17 - Head 13	0.7090	0.5927	0.4673	0.6788	0.5760	0.4599
5	XLM-R	Attention × Norm - Layer 18 - Head 3	0.6697	0.5540	0.4228	0.7257	0.6373	0.5200
6	XLM-R	Attention × Norm - Layer 19 - Head 2	0.7335	0.6181	0.4857	**0.7404**	**0.6477**	**0.5303**
7	XLM-R	Gradient × Hidden States - Layer 14	0.6567	0.5403	0.4156	0.6041	0.4837	0.3619
8	XLM-R	Gradient × Attention - Layer 17	0.6613	0.5597	0.4322	0.6891	0.5983	0.4798
9	XLM-R	Integrated Gradients - Layer 15	0.6194	0.4995	0.3699	0.5705	0.4649	0.3489
10	XLM-R	LIME	0.6221	0.4968	0.3606	0.5405	0.4297	0.3222
11	XLM-R	Leave-one-out	0.6584	0.5375	0.4082	0.5493	0.4494	0.3412
12	XLM-R	Relaxed-Bernoulli Rationalizer	0.4933	0.3794	0.2481	0.5406	0.4277	0.3211
13	XLM-R-M	Attention × Norm - Layer 21 - Head 8	0.6235	0.5041	0.3670	0.7122	0.6254	0.5133
14	XLM-R-M	Attention × Norm - Layer 21 - Head 9	0.5510	0.4106	0.2738	0.7068	0.6175	0.5059
15	RemBERT	Attention × Norm - Layer 23	0.7465	**0.6382**	**0.5229**	0.7085	0.5954	0.4756
16	RemBERT	Attention × Norm - Layer 23 - Head 8	**0.7501**	0.6203	0.4912	0.6758	0.5486	0.4418
1	Ensemble	(5) + (6) + (15)	**0.7467**	**0.6368**	**0.5113**	0.7545	0.6662	0.5512
2	Ensemble	(5) + (6) + (14) + (15)	0.7441	0.6366	0.5089	**0.7639**	**0.6805**	**0.5688**

Table D.3: Full constrained track results on the validation set of ET-EN.

E

Supplemental Material for Chapter 7

E.1 Data Information

The data used for finetuning our QE systems is shown in Table E.1. For DA data, we split the original development set to generate a new dev/test split, therefore the reported numbers in the table correspond to this "internal" dev split.

LP	Samples	Source Tokens	Target Tokens	Target OK / BAD
TRAIN				
en-de	9000	147870	153656	0.84 / 0.16
en-mr	26000	690516	561371	0.90 / 0.10
en-zh	9000	148657	163308	0.65 / 0.35
et-en	9000	126877	185491	0.75 / 0.25
ne-en	9000	135205	181707	0.41 / 0.59
ro-en	9000	154538	167471	0.71 / 0.29
ru-en	9000	104423	132006	0.85 / 0.15
si-en	9000	141283	166914	0.42 / 0.58
en-de†	54681	1571090	1926444	0.90 / 0.10
en-ru†	15628	312185	354871	0.95 / 0.05
zh-en†	75327	134165	2789907	0.87 / 0.13
DEV				
en-de	500	8262	8555	0.84 / 0.16
en-mr	500	13803	11216	0.91 / 0.09
en-zh	500	8422	9302	0.75 / 0.25
et-en	500	7081	10257	0.73 / 0.27
ne-en	500	7542	10247	0.38 / 0.62
ro-en	500	8550	9202	0.78 / 0.22
ru-en	500	5984	7511	0.84 / 0.16
si-en	500	7866	9415	0.41 / 0.59
en-cs	500	10302	9302	0.75 / 0.25
en-ja	500	10354	13287	0.73 / 0.27
km-en	495	9015	8843	0.45 / 0.55
ps-en	500	13463	12160	0.51 / 0.49
en-de†	503	10535	12454	0.96 / 0.04
en-ru†	503	10767	11911	0.91 / 0.09
zh-en†	509	980	19192	0.98 / 0.02

Table E.1: DA and MQM (†) data for all LPs.

F

Supplemental Material for Chapter 8

F.1 Explainer Details

With the *integrated gradients* explainer (Sundararajan et al., 2017), we use 10 iterations for the integral in the *simulability* experiments (due to the computation costs) and 50 iterations for the *plausability* experiments. We use zero vectors as baseline embeddings, since we found little variation in changing this. For both gradients-based explainers, we project into the simplex by using the softmax function, similar to the attention-based explainers. This results in very negative values having low probability values. Moreover, for evaluating plausibility on translation quality estimation, we followed Treviso et al. (2021) and computed the explanation score of a single word by summing the scores of its word pieces.

We would like to note that, unlike the setting in Pruthi et al. (2022), we **do not** apply a top-k post-processing heuristic on gradients/attention logits, instead directly projecting them to the simplex. This might explain the difference in results to the original paper, particularly for the low simulability performance of static explainers.