Informatics

Evaluierung der LIME-basierten Erklärungen von Relationsextraktions-Modelle

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Data Science

eingereicht von

Thais Beham, B.Eng.
Matrikelnummer 11938269

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Ass. Gábor Recski, PhD

SIGNATURINFORMATIONEN	Signatory	Gabor Andras Recski
	Date/Time-UTC	2024-01-29T17:36:25+0100
	Verification	Information about the verification of the electronic signature can be found at: https://www.signaturpruefung.gv.at
Note	This document is signed with a qualified electronic signature. According to Art. 25 para. 2 of the Regulation (EU) No 910/2014 of 23. July 2014 ("eIDAS-Regulation") it shall have the equivalent legal effect of a handwritten signature.	

Wien, 29. Jänner 2024

Thais Beham

Thais Beham

Gábor Recski

Informatics

Evaluating LIME-based explanations of Relation Extraction models

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieurin

in

Data Science

by

Thais Beham, B.Eng.
Registration Number 11938269

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Ass. Gábor Recski, PhD

Vienna, January 29, 2024

Thais Beham

Thais Beham

Gábor Recski

Erklärung zur Verfassung der Arbeit

Thais Beham, B.Eng.

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 29. Jänner 2024

Thais Beham

Thais Beham

Danksagung

Ich möchte allen Professoren, die an meinem Masterstudiengang teilgenommen haben, für das Wissen danken, das ich in den Kursen und Projekten erworben habe.

Meinem Betreuer, Gábor Recski, danke ich vor allem für die interessanten Diskussionen über NLP, die zu dieser Arbeit geführt haben, und für seine Aufmerksamkeit und Verfügbarkeit, meine Fragen zu beantworten.

Vielen Dank für die von der Technischen Universität Wien zur Verfügung gestellten Ressourcen zur Berechnung meiner Ergebnisse.

Danke an Alex, der mich motiviert hat, eine Karriere im Bereich Data Science anzustreben und mir gesagt hat, dass mir dieses Fachgebiet wirklich Spaß machen würde.

Danke an alle meine Freunde, die mich auf dem schwierigen Weg des Umzugs in ein neues Land unterstützt haben, insbesondere während der Pandemie. Danke für den großen Spaß, den wir zusammen haben, und die Unterstützung in den schweren Zeiten.

Danke an meine Eltern, die, ohne dass ich es erwähnen muss, für mich unentbehrlich waren, um diesen Abschluss zu erreichen.

Acknowledgements

Thank you to all the Professors present in my master's journey for all the knowledge gained through the classes and projects.

Thank you very much to my supervisor, Gábor Recski, especially for the interesting discussions about NLP that resulted in this present work and for his attention and availability to answer my questions.

Thank you for the resources provided by the Vienna University of Technology for computing my results.

Thank you to my friend Alex, who motivated me to pursue a career in Data Science and told me that I would really enjoy the field.

Thank you to all my friends who supported me in the challenging journey of moving to a new country, especially during the pandemic. Thank you for the great fun we have together and the support in the hard times.

Thank you to my parents, who, with no need to say, were totally essential for me to accomplish this degree.

Kurzfassung

Die Bedeutung der Erklärung von Modellen des maschinellen Lernens hat in letzter Zeit aufgrund des Fortschritts des Deep Learning und der Anwendung dieser Modelle in verschiedenen Bereichen stark zugenommen[ABV+20]. Da die meisten dieser Modelle in einer *black-box* präsentiert werden, ist es wichtig, Vertrauen und Klarheit in ihr Verhalten zu gewährleisten, um ihre ordnungsgemäße Verwendung zu ermöglichen [TH22]. In diesem Zusammenhang ist LIME [RSG16] aufgrund seiner zufriedenstellenden Ergebnisse in mehreren Studien und seiner modellagnostischen Eigenschaft [HSM+20] eines der wichtigsten Werkzeuge im Bereich der XAI.

In Anbetracht dessen evaluiert diese Arbeit die Erklärbarkeit von LIME speziell für die Aufgabe der Relationsextraktion und stellt fest, dass LIME diese Aufgabe nicht korrekt behandelt, da er bei der Erstellung seiner Stichproben zufällig Wörter entfernt, was dazu führt, dass Beziehungsentitäten, die eine Relationsextraktionsaufgabe charakterisieren, entfernt werden. Daher wird in dieser Arbeit eine Lösung für dieses Problem vorgeschlagen, indem die interne Funktionalität von LIME so modifiziert wird, dass die Entfernung von Entitäten während des Sampling-Prozesses vermieden wird.

Qualitative und quantitative Metriken wurden verwendet, um die Erklärbarkeit von LIME in zwei separaten Modellen zur Relationsextraktion zu bewerten: ein neuronales Black-Box-Modell namens AGGCN [GZL19] und ein traditionelles maschinelles Lernmodell, Naive Bayes. Der verwendete Datensatz ist SemEval-10 Aufgabe 8 [HKK+10].

Die Metriken umfassen: Faithfulness[DJR+20], einschließlich der Berechnung von Sufficiency und Comprehensiveness, Stability[BCL23] mit Inherent und Parameter Stability, und Global Inference (Bewertung von SP-LIME[RSG16]). Sie werden anhand des Vergleichs zwischen den beiden Modellen bewertet.

Die Ergebnisse zeigten, dass die LIME-Rationale für die entsprechenden Vorhersagen (*faithful* Rationale) in dieser experimentellen Konfiguration sehr einflussreich zu sein scheint. Darüber hinaus zeigte LIME eine inhärente Stabilität (was die Ergebnisse der vorherigen Studie bestätigt[BCL23]) und wies für etwa 70% der getesteten Proben in beiden Modellen eine Parameterstabilität auf. Darüber hinaus zeigt LIME unterdurchschnittliche Ergebnisse bei der globalen Analyse mit SP-LIME, insbesondere beim Umgang mit Mehrklassenproblemen und textuellen Daten. Außerdem, haben die übermäßigen Laufzeiten von LIME für komplexe Modelle einen erheblichen Nachteil der Methode offenbart.

Abstract

The importance of explaining machine learning models has significantly grown recently due to the advance of Deep Learning and the application of these models in several fields [ABV+20]. Since most of these models are presented in a *black-box* matter, it is essential to ensure trustfulness and clarity in their behavior to enable their proper use [TH22]. In this context, LIME [RSG16] is one of the most important tools in the field of XAI due to its satisfactory results in several studies and its model-agnostic trait [HSM+20].

In consideration of the aforementioned, this work evaluates LIME explainability specifically for the task of Relation Extraction and identifies that LIME does not handle this task correctly since it randomly removes words when creating its samples, consequently resulting in the removal of relation entities that characterize a Relation Extraction task. Therefore, this work proposes a solution for the issue by modifying LIME's internal functionality to avoid the removal of the entities in the sampling process.

Qualitative and quantitative metrics were used to assess LIME explainability in two separate models for relation extraction: a black-box neural model named AGGCN [GZL19] and a traditional machine learning model, Naive Bayes. The dataset used is SemEval-10 task 8 [HKK+10].

The metrics encompass: Faithfulness[DJR+20], including the computation of Sufficiency and Comprehensiveness, Stability[BCL23] comprising Inherent and Parameter Stability, and Global Inference (evaluation of SP-LIME[RSG16]). They are assessed by considering the comparison between the two models .

The results showed that LIME rationales appear to be highly influential for the corresponding predictions (faithful rationales) in this experimental configuration. Furthermore, LIME presented inherent stability (confirming the previous study's findings [BCL23]) and exhibited parameter stability for approximately 70% of the tested samples in both models. Moreover, LIME demonstrates underwhelming results for global analysis using SP-LIME, especially for dealing with multi-class problems and textual data. Additionally, the excessive running times of LIME for complex models revealed a significant drawback of the method.

Contents

CHAPTER 1

Introduction

Artificial Intelligence (AI) and Deep Learning (DL) are powerful technologies that present great potential to provide benefits to society. Their recent advances have increased the number of tasks performed with high accuracy and success in complex problems such as image classification, face recognition, sentiment analysis, text classification, and speech understanding [Mat19]. Researchers have also started to explore how these approaches could highly benefit different domains such as healthcare, the criminal justice system, finance, and security [WA22]. Especially in healthcare, AI has shown to be relevant in discovering new uses for existing drugs, revealing cancer in tissues, detecting cardiac arrhythmia, and predicting hypoglycemic events in diabetics three hours before the medical industry average[Mat19].

However, these models are usually applied in a black-box manner, meaning that the internal functionality that led to a prediction is either unknown or known but uninterpretable by humans [GMR+18]. This is problematic due to lack of transparency, possible biases inherited by the algorithms from human prejudice embedded in the training data, and lack of trustworthiness [GMR+18]. Additionally, the European Union has recently been discussing the regulation of AI in what is known as the "AI Act", which intends to classify AI systems according to four levels of risk: from minimal to unacceptable[Mad21].

Therefore, to address trustfulness in AI, it is important to propose techniques to understand how the model behaves and assess if the decisions are made properly. This is what the field of study "Explainability in AI" (XAI) focused on. In this sense, XAI is a key part of applying ethics to AI because it tries to understand how machines perform computational work, reducing the imprint of unconscious biases and increasing trust in model decisions[Tur].

LIME [RSG16] (acronym for Local Interpretable Model-agnostic Explanations), presented in 2016, is one of the most popular tools for explainability, as well as one of the first techniques that emerged in the field [HSM+20]. One of the reasons why the method succeeded is its capability to function without requiring any information about the internals of the model to be explained, such as topology, learned parameters (weights, biases), and activation values [HSM+20].

Natural Language Processing (NLP) is a branch of AI involved with processing and analyzing human language [IBM]. It is placed at an intersection of computer science, artificial intelligence, and computational linguistics [LK17]. In other words, NLP is concerned with computers analyzing, understanding, and deriving meaning from human language in a smart and useful way [LK17]. Typical tasks of NLP include, but are not limited to: speech recognition, sentiment analysis, spam detection, machine translation, and relation extraction [IBM].

The great advances in computerized language processing led to the emergence of the Large Language Models (LLMs) [HQS+23], and their popularization in the form of user-friendly tools like Chat-GPT. They compress considerably potency in their capability to understand intricate linguistic patterns and provide coherent and contextually fitting responses [HQS+23]. However, those powerful models carry an inherent lack of explainability and transparency [HQS+23], meaning that the increase in the model complexity and the nature of their training process led to a deprecation of model understanding. The amount of parameters, in the degree of millions or billions, ensures the challenge of understanding the decision-making process of one prediction [HQS+23].

Relation Extraction (RE) is an important task within the NLP field since it focuses on extracting the semantic relationship between entities based on their related context, which is essential in the fields of Information Extraction (IE) and knowledge base construction [CZX+22]. RE plays an important role in domains like: automatic question-answering systems, retrieval systems, ontology learning, and semantic web labeling tasks[ZCL17].

1.1 Problem Statement

Given the aforementioned context, this master's thesis intends to address the following problem: are the current methods for explainability in ML/AI reliable? Specifically, the work focuses on evaluating the performance of LIME for the NLP task of Relation Extraction. The objective is to appraise how well it is possible to trust the explanations provided by LIME, focusing on this particular task. For this goal, the project is concentrated on Relation Extraction between a pair of nominals (entities), which consists of classifying the kind of relation that two nominals have in a sentence.

LIME is a method that works by creating samples around the instance to be explained, as perturbations of the original one [RSG16]. For textual data, this process occurs by randomly removing some words of a document [RSG16]. This approach is particularly dangerous when dealing with Relation Extraction problems since the relational entities may be removed when creating the samples. The issue comes from the fact that these created samples later obtain their prediction from the target model (the one which is intended to be explained), and since the entities are not present, the model is not able to predict the relation between them coherently. Moreover, some models even need to receive the relation entities together with the sentence as input to output a prediction [GZL19].

Thus, the thesis intends to address two Research questions:

- How to enable explanations of LIME in Relation Extraction tasks, in light of the LIME inherent process of removing words during its sampling step and potentially removing the relation entities?
- How well does LIME perform in Relation Extraction tasks regarding its explainability evaluation on several metrics?

For the first, we propose a modification of one of LIME's methods for text explanations to make it viable to provide the explanations without losing the relation entities in the process. For the second, LIME explanations will be analyzed individually for 2 different models, an explainable machine learning model, and a black box neural model. These explanations will be assessed using qualitative and quantitative metrics. Those metrics are: Faithfulness (Sufficiency and Comprehensiveness) [DJR$^+$20], Stability (Inherent and Parameter)[BCL23], and Global Inference (evaluation of SP-LIME [RSG16]) and are explained in details in section 3.2, section 3.3, and section 3.4, respectively.

The contributions of this present work include the thorough assessment of LIME explanations regarding RE tasks, the comparison of LIME performance among different models and metrics, and the proposal of LIME modifications to allow consistent explanations of RE tasks.

The document follows the structure: chapter 2, named Background and related work, discusses the main concepts of this study, among which are Explainability (encompassing explainability in NLP, techniques for evaluating explainability, and LIME), Relation Extraction and details about the dataset used in the experiments. The chapter 3, Methodology, analyzes LIME for Relation Extraction, explains the proposed modifications for the tool, presents in detail the metrics that will be used for assessment, and describes the Experimental setup. The chapter 4 refers to Results and Discussion and, lastly, chapter 5 closes with the conclusions and suggestions for future work.

CHAPTER 2

Background and related work

2.1 Explainability

The advance of AI systems and the fact that researchers are indicating that those can even outperform humans in some analytical tasks (such as pattern recognition in imaging) is followed by the emerging importance of providing explanations for models' decisions[ABV+20]. The legal and ethical uncertainties surrounding these complex models make it difficult to advance this technology to its full potential [ABV+20], especially in fields such as medical or financial where the consequences of the decisions can cause considerable harm [YXHD23]. Therefore, there is a remarkable demand for research advances to provide more clarity and transparency in the model decision.

The growing importance of Explainability in various research communities is outstanding and ensured by workshops on: Explanation-aware Computing (ExaCt), Fairness, Accountability, and Transparency (FAT-ML), Workshop on Human Interpretability in Machine Learning (WHI), Interpretable ML for Complex Systems, Workshop on Explainable AI, Human-Centred Machine Learning, and Explainable Smart Systems [RR19].

Nevertheless, the work of Rosenfeld & Richardson (2019)[RR19] discusses that the term Explanability seems to have no consensus regarding its precise definition, as well as other related terms such as interpretability, transparency, explicitness, and faithfulness. However, the machine learning community frequently refers to explainability as the attempt to understand how machine learning algorithms make their decisions and how interpretations can be derived either directly or secondarily from machine learning components. Additionally, the authors discuss that papers seem to provide no difference between interpretable or explainable systems, and their opposite is usually referred to as "opacity" or "black-box". Transparency, on the other hand, is defined by the authors as a trait from a model that requires no additional information to be understood, such as Decision-trees for example.

Furthermore, a distinction is often made between two methods of achieving interpretability: interpreting existing models via post-hoc techniques (application of interpretation methods after model training) and designing inherently interpretable models [Mol22] [JG20]. This present work focuses on post-hoc techniques.

2.1.1 Explainability in NLP

Rationales are an important concept when analyzing Explainability in NLP. They can be defined as: given a model prediction of a document, it consists of the set of the most important words that contributed to this prediction [DJR+20][ZA22]. Explanations in NLP are usually carried out by identifying the rationales of a certain output [ZA22].

Some examples of techniques for Explainability in NLP are: Provenance-based, Surrogate model, Example-driven, and Feature importance [DQA+20]. To illustrate, Example-driven is a technique that explains an instance by identifying and presenting other instances, from available labeled data, that are semantically similar to the target one. For that, the work of [CRB19] proposes a method of selecting semantically similar examples by using *Layerwise Relevance Propagation*[BBM+15], an approach that assigns a relevance score for each feature.

Feature importance is a technique in which the explanation is given by providing the importance scores of the input features. One example is the research of Wallace *et al.* (2018) [WFBG18] that proposes a technique for feature importance in text classification of neural models using Deep k-Nearest Neighbors (DKNN) [PM18].

Surrogate model is an approach where the predictions of a model are explained by another model, usually more explainable, as a proxy [DQA+20]. Its use can be already seen in 1995, when the authors [CS95] proposed the use of Decision-trees as a surrogate model for neural models to explain their decisions. The huge advantage of this technique is that it can be used to explain any model (model-agnostic). Another benefit is that it can accomplish both local and global explanations [SHSRF19]. This is the principle used by LIME [RSG16].

Furthermore, the work of [ZA22] discusses the challenges of explainability in NLP. The authors argue that the field requires improvement regarding the use of rationales as explanations because words are a combination of syntax, semantics, and previous context. Therefore, they cannot easily be dissected from the input to interdependently serve as explanations [ZA22]. Nonetheless, explainable methods are intended as approximations, and they are inserted in an evolving field that should tackle these challenges during its progress.

2.1.2 Evaluating explainability

Several approaches currently exist to evaluate explainability techniques. One of those is referred to as "Plausibility". It seeks to assess how useful the explanation is to humans[JG20]. To accomplish this, a few metrics are employed to compare the rationales selected by the explainability tool with those chosen by humans [SMGBN22]. Examples are: BLEU [PRWZ02], MAXSIM[CN08], Intersection-over-Union (IOU)[DJR+20] and Area Under the Precision-Recall Curve (AUPRC)[SMGBN22]. The research of [SCNB23] proposes a framework to evaluate plausibility in language generation tasks. In addition, another study [NGS21] evaluates plausibility in a sentence comparison task.

Faithfulness is another relevant measure for gauging the explanations. It can be defined as an assessment of how accurately the explanation reflects the true reasoning process of a model [JG20]. The metric is presented by Deyoung *et al.* (2020) [DJR+20] which proposes the calculation of the metric by two evaluations: first, the degree to which the selected rationales were enough for the prediction, referred as "*Sufficiency*" and, secondly, the degree to which all the required features were selected as rationales, namely "*Comprehensiveness*" [DJR+20]. More details on this metric can be found in section 3.2.

2.1.3 LIME

One of the most popular approaches proposed for AI explainability is LIME[1] (Local Interpretable Model-agnostic Explanations)[HSM+20]. It is a method capable of explaining any model, irrespective of its level of complexity or configuration (model-agnostic). In addition, LIME operates locally, explaining the prediction of a single sample, rather than addressing the entire model globally. The explanations are carried out by showing which parts of the input (features) had the highest importance for the referred prediction; for example, a model that predicts a person's salary might show a high level of importance in features such as age and educational level when explained. Similarly, for images, the method identifies the most important patches, while for textual data, it highlights the most relevant words[RSG16].

LIME is a surrogate-based explanation technique [HSM+20]. This means it intends to explain the predictions of complex models (target models) by approximating a more interpretable model, named surrogate, in the locality of the instance to be explained. Ultimately, the decision is explained by interpreting the surrogate model prediction instead of the complex model. Through this approach, LIME ensures its ability to explain any complex model, as its intrinsic functionality is irrelevant.

The Figure 2.1 is a toy example that illustrates the LIME functionality [RSG16]. On the left side, there is a complex non-linear model to predict Diabetes, and the instance to be explained is selected. On the right side, we can observe that, in the vicinity around the instance, a simple linear model would be enough to explain the decisions. Thus, LIME explains the sample decision from the target model by approximating a surrogate model locally.

In summary, an explanation by LIME works as follows: given the prediction of an instance by a target model to be explained, LIME creates perturbations of the instance and weighs them according to their distance to the original instance. Then, the created samples are

[1] Github repository: https://github.com/marcotcr/lime

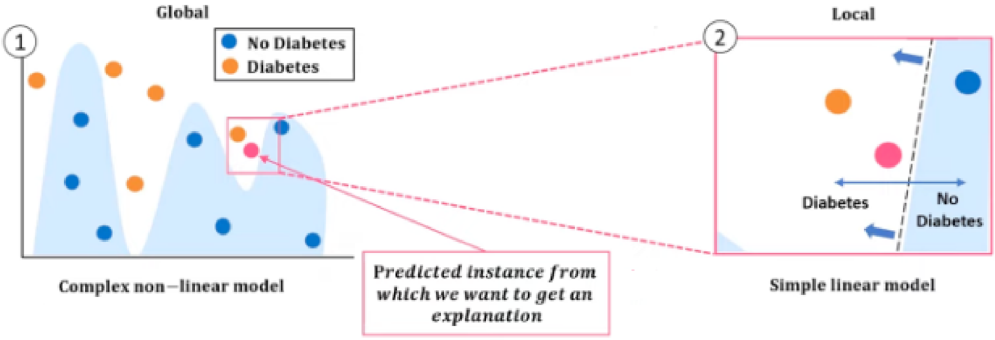


Figure 2.1: Toy example to illustrate the functionality of LIME via surrogate model. The left side indicates a complex model, and an instance is selected for explanation. On the right side, we can observe that in the vicinity of the sample, a linear model can be fitted to explain the decisions (the surrogate model). Modified from [Ale22].

labeled by the target model, resulting in the neighborhood dataset. The surrogate model is then created by being fitted in this dataset. Lastly, the prediction from the surrogate model of the target instance is explained[RSG16].

Additionally, the explanation involves a Fidelity-Interpretability trade-off. This means that the surrogate model should be simple enough to be understood by humans, implying high Interpretability. In contrast, it should also strive to approximate to the predictions of the target model locally, indicating high Fidelity. Therefore, the intention is to optimize the fidelity of the surrogate to the target model while keeping the complexity of the surrogate model low enough [RSG16].

Let g be the surrogate model, f be the target model, $\Omega(g)$ be the complexity of the surrogate model, and π_x be the neighborhood of perturbed samples around x, the instance to be explained. Moreover, let $\mathcal{L}(f, g, \pi_x)$ be a function that measures the infidelity of g to f in the locality defined by π_x, meaning the degree of how different the predictions of g are in relation to f in the neighborhood π_x. The explanation $\xi(x)$ is obtained as a minimization problem, as indicated in **??** [RSG16]. In order to guarantee both Interpretability and local Fidelity, $\mathcal{L}(f, g, \pi_x)$ should be minimized while ensuring that $\Omega(g)$ is low enough to allow interpretation by humans [RSG16].

$$\xi(x) = \operatorname*{argmin}_{g \in G} \left(\mathcal{L}(f, g, \pi_x) + \Omega(g)\right) \tag{2.1}$$

The measure of $\mathcal{L}(f, g, \pi_x)$ is computed by a locally weighted square loss of the f and g predictions in the locality π_x. Let z' represent elements of the set Z, which is the perturbed samples dataset, and z the recovered sample in its original representation; they are weighted by the distance in relation to x, denoted by $\pi_x(z)$. Furthermore, the loss is computed as the sum of the weighted squared difference in the prediction of the target model $f(z)$ and the surrogate model $g(z')$, for each z, z' pair from Z, as defined in Equation 2.2 [RSG16].

$$\mathcal{L}(f, g, \pi_x) = \sum_{z,z' \in Z} \pi_x(z) \left(f(z) - g(z')\right)^2 \tag{2.2}$$

Given $\pi_x(z)$ as a function that defines the weights of a perturbed sample z in relation to the original sample x and D as the distance function between them, the calculation of $\pi_x(z)$ is depicted in Equation 2.3. It consists of an exponential kernel with D as the distance function. For textual data, D is expressed as the cosine distance between x and z with width defined by σ. The fact that weights are considered and determined based on the distance makes the method quite resilient to sampling noise [RSG16].

$$\pi_x(z) = \exp\left(-\frac{D(x, z)^2}{\sigma^2}\right) \tag{2.3}$$

Moreover, the complexity of g, denoted as $\Omega(g)$, is defined depending on the type of the explainable model. For linear models, it may be the number of non-zero weights, and for decision trees, it may be the depth of the tree [RSG16].

LIME for textual data

The LIME explanations for textual data occur as follows: given an instance (referred to as the target sentence in this context) to be explained, it creates perturbed data points by randomly removing some of its words, resulting in a total of 5000 new samples. Further, each new perturbed sentence is weighted regarding the distance to the original one, defined by π_x, and calculated via Equation 2.3.

The new sentences are then fed into the black-box classifier and their predictions are obtained, resulting in a weighted neighborhood dataset. Afterward, a linear model is trained in this dataset and used to predict the target sentence. Finally, the LIME explanation is the interpretation of the linear model prediction of the target sentence[Mol22]. The interpretation is carried out by analyzing the weights of the model's features since they correspond to the respective importance of the words to the prediction [Ern18]. These are illustrated in the Figure 2.2.

For a given sentence, LIME will return the prediction probability of each class, as well as a list with the words that contributed the most for the prediction, the LIME rationales. The Figure 2.3 shows an example of LIME explanation for RE task, which the sentence is classified as "Product-Producer" relation (with probability of 1 and remaining classes with probability of 0) and the rationales are: "composed", "a", "for", "famous" and "was".

Regarding the pros and cons of the method, the ability to explain any model is a significant advantage and it can be found numerous successful applications of LIME in different

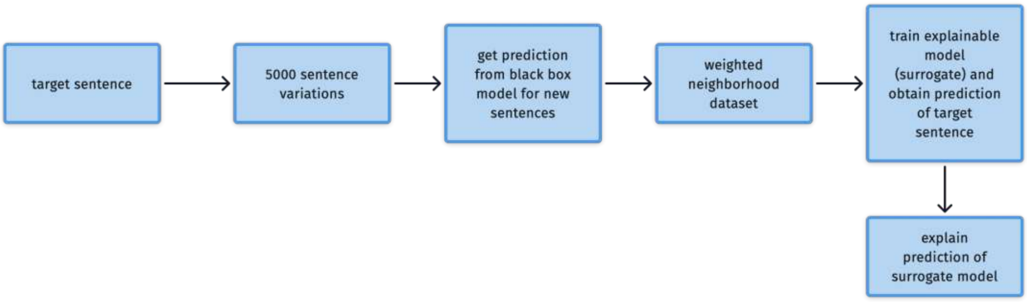
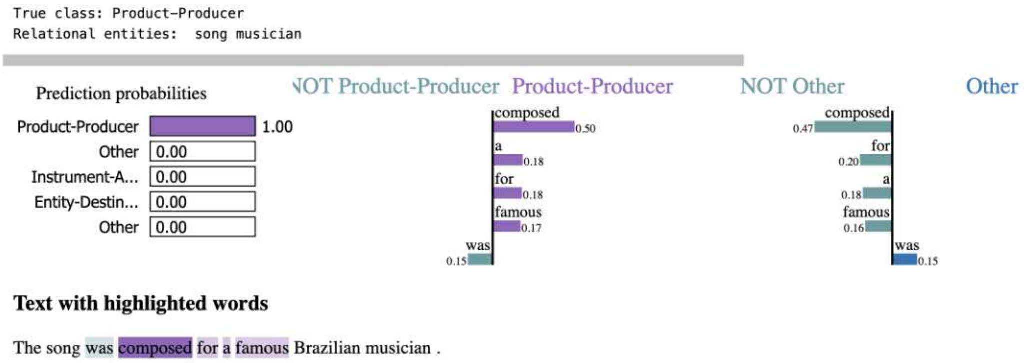

target sentence
5000 sentence variations
get prediction from black box model for new sentences
weighted neighborhood dataset
train explainable model (surrogate) and obtain prediction of target sentence
explain prediction of surrogate model

Figure 2.2: LIME functionality for textual data. The diagram indicates a simplified version of the steps LIME takes when generating an explanation. The steps show that LIME actually explains the prediction of a surrogate model, instead of the model itself, and this approximation is made locally.



```
True class: Product-Producer
Relational entities:  song musician
```

Text with highlighted words

The song was composed for a famous Brazilian musician .

Figure 2.3: Example of LIME explanation. The figure shows on the left side the prediction probabilities for each class, in the middle, the weights of each rational for the prediction of the class Product-Producer, and below the sentence with the highlighted words as the rationales, with color saturation proportional to its weights.

domains. However, using a surrogate model only indirectly solves the problem since the explanation of the target model is highly dependable on the quality of the surrogate fit, which requires dense sampling, resulting in excessive computational costs. Additionally, sampling leads to instability where the same instance can be susceptible to different explanations from different runs[HSM+20].

2.1.4 Related work on LIME for NLP

Within the literature review, one of the studies [TSB+21] suggests employing LIME to assess a model intended to classify the misuse of opioids in textual clinical data. The work obtains the explanations from LIME in all individual predictions to assess for differences in features between race/ethnic groups. Additionally, it indicates that LIME can also be used to detect bias in models.

Another research [THZ+22], focusing on detecting toxicity in social media text, used LIME to detect the toxic spans, meaning the parts of the text that are toxic. First, the documents are classified by a Long Short Term Memory (LSTM) model with GloVE as toxic or not. Later, the selection of the toxic parts is carried out by the LIME explanation of the model decision. The text selection is the LIME rationales since it encompasses the set of the most important words for the prediction. The successful results demonstrate an accuracy of 98% for LIME in detecting toxic spans.

Lastly, additional work [ZGW+19] proposes the calculation of semantic similarity between medical text pairs, using Convolutional Neural Networks (CNN) and LIME to detect which words were decisive for the prediction results, aiming to enhance the model interpretability.

2.2 Relation Extraction

The amount of textual data generated due to the advent of Web 2.0 is growing exponentially [NJM21] [MF16]. This data comprises a variety of sources, such as social networks, online blogs, magazines, news articles, research publications, and question-answering forums [NJM21] [PPB17]. The possibility to analyze this huge amount of data holds the potential to yield valuable insights for a variety of purposes, thereby elevating its significance as a task of great importance[NJM21]. Especially for medical data, it could generate numerous benefits by the processing of electronic health records (EHRs) [WWRM+18]. The NLP domain dedicated to addressing this challenge is known as Information Extraction (IE), which encompasses the transformation of unstructured or semi-structured data into structured data [NJM21]. Name Entity Recognition (NER) and Relation Extraction (RE) are disciplines part of this field.

Specifically, Relation Extraction pertains to the problem of extracting semantic relationships between entities. For example, in a sentence that has as entities a person and an organization, they may be linked through relations such as "employed at" [PPB17].To illustrate, the phrase: “Fizzy [drinks] and meat cause heart disease and [diabetes]”, has annotated entities as e1 = “drinks” and e2 = “diabetes”, the goal of the task would be to automatically detect the cause-effect relationship, indicated with the notation Cause-Effect(e1,e2) [WCDML16].

RE carries huge importance for advanced NLP, since it is used in tasks such as Machine Translation (MT), Question Answering (QA) systems, and Event Extraction [NJM21]. Other examples of its utilization are the detection of interactions between drugs to build a medical database and the extraction of relationships among people to create an easily searchable knowledge base[HRG23].

Regarding the related work in the field, the research of [LRW+18] deals with the task of Relation Extraction in multiple relations among multiple entities in unstructured text. It uses the SemEval 2017 dataset and proposes to solve the problem using a dynamic Long Short-Term Memory (LSTM) network. To train the model, entity features, entity position, and part of speech features were used. In another study, the reference [YL10] employs a graph-based model to extract relationships from data sourced from Wikipedia.

2.2.1 Dataset

The Dataset comprising the Relation Extraction task chosen for this work is called SemEval 2010 task 8. Semantic Evaluation (SemEval) Corpus is a yearly workshop focused on semantic-oriented problems [NJM21] and its repositories possess several datasets extensively employed for different Information Extraction tasks. A prominent one is SemEval 2010 task 8, for Relation Extraction[NJM21].

The dataset was introduced by Hendrickx *et al.* (2010) [HKK+10]. It consists of a Multi-class classification of semantic relations between pairs of nominals (noun, noun phrase, or any word or group of words that functions as a noun [Nor19]), consisting of 9 relations plus class ’Other’ and 10,717 annotated examples. The average length of the sentences is 19 words. The following list displays the classes and provides an example of one sample from each:

- **Cause-Effect (CE)**: "The <e1>burst</e1> has been caused by water hammer <e2>pressure</e2>."
- **Instrument-Agency (IA)**: "The <e1>author</e1> of a keygen uses a <e2>disassembler</e2> to look at the raw assembly code."
- **Product-Producer (PP)**: "The <e1>court</e1> decided the objection by making the instalment <e2>order</e2> as sought."
- **Content-Container (CC)**: "The <e1>lawsonite</e1> was contained in a <e2>platinum crucible</e2> and the counter-weight was a plastic crucible with metal pieces."
- **Entity-Origin (EO)**: "The technology is available to produce and transmit <e1>electricity</e1> economically from OTEC <e2>systems</e2>."

- **Entity-Destination (ED)**: "<e1>People</e1> have been moving back into <e2>downtown</e2>."
- **Component-Whole (CW)**: "The system as described above has its greatest application in an arrayed <e1>configuration</e1> of antenna <e2>elements</e2>."
- **Member-Collection (MC)**: "The <e1>student</e1> <e2>association</e2> is the voice of the undergraduate student population of the State University of New York at Buffalo."
- **Message-Topic (MT)**: "The Pulitzer Committee issues an official <e1>citation</e1> explaining the <e2>reasons</e2> for the award."
- **Other**: "Unlike other fish, grunion come out of the water completely to lay their eggs in the wet <e1>sand</e1> of the <e2>beach</e2>."

The distribution of the classes by their percentage and frequency of samples can be seen in Table 2.1

Table 2.1: Classes distribution in the dataset SemEval 10 - task 8. For each class is indicated the amount of samples (Frequency), and its Percentage.

Relation	Frequency	Percentage
Cause-Effect	1331	12.4%
Component-Whole	1253	11.7%
Entity-Destination	1137	10.6%
Entity-Origin	974	9.1%
Product-Producer	948	8.8%
Member-Collection	923	8.6%
Message-Topic	895	8.4%
Content-Container	732	6.8%
Instrument-Agency	660	6.2%
Other	1864	17.4%
Total	10717	100%

The dataset has been utilized by several researchers. For instance, the work of[GCH+20] introduced a model named Tree-Structured LSTM with Attention, achieving state-of-the-art results with an F1 score of 0.871; another study by Wang *et al.* (2016) [WCDML16], presents a Convolutional Neural Network architecture with two levels of attention; lastly, the model called Attention Guided Graph Convolutional Networks for Relation Extraction (AGGCN)[GZL19] achieves F1 score of 0.8513 on this dataset. The latter is one of the models chosen to be evaluated in this present work since it enabled clear and easy reproducibility of its results with a good performance.

CHAPTER 3

Methodology

This chapter will explain the LIME modifications carried out for Relation Extraction, the metrics used in our experiments, the experimental setup, and the models used in the evaluations.

3.1 LIME for Relation Extraction

As referred to in subsection 2.1.3, the process of using LIME for textual data includes creating perturbations of the sentence to be explained, which is carried out by randomly removing some of its words. The following step is to obtain the predictions of the newly created samples from the target model.

The problem when dealing with Relation Extraction is that the relational entities may be removed during this process, so the model predictions will not work coherently. Given that the goal of the task is to predict the relation between the entities, if those are not present, the task is inconsistent. It is especially damaging when the model requires as input not only the sentence but the position of the relational entities or the entities themselves, as in the case of our experiments for the AGGCN model (further details about the preprocessing of the dataset in subsection 3.5.1).

The strategy to overcome this issue was the following: to guarantee that the predictions work correctly, we added the constraint that the relation entities can not be removed during LIME sampling process. With this limitation, we still ensure that the variations among the created samples are different enough since only 2 words are blocked from being removed. Hence, those variations consistently enable a good fit from the surrogate model. To achieve it, we present modifications for the LIME method used for explaining textual data: `LimeTextExplainer()`.

3.1.1 Modifications of `LimeTextExplainer()`

This present project proposes a solution to enable coherent functionality of LIME explanations for Relation Extraction by modifying the method `LimeTextExplainer()`[1]. In summary, we implemented a new input field in the function that provides explanations for adding the relational entities. Thus, when the user requests an explanation from a sentence, it can optionally add the relational entities in the new parameter `exception_words`, as displayed below:

```
explainer = ExtendedLimeTextExplainer()
exp = explainer.explain_instance(sentence, classifier_fn,
exception_words=(entity1, entity2))
```

After receiving this information as input, the modified `LimeTextExplainer` method (namely `ExtendedLimeTextExplainer()`) has internal constraints to avoid the removal of the entities during the step of creating perturbed sentences.

The steps in details are:

1. Modification of the vector that defines the number of words to be omitted in each perturbed sample: the vector is of size $(1, n)$, where n is the number of perturbed sentences and each of its elements indicates the number of words to be removed for the respective sentence. Originally, each element of the vector is randomly chosen in the range from 1 to the total length of the sentence. Now, the range is from 1 to total length subtracted by 2 (thus, all words, except 2, can be omitted).

2. LIME creates a matrix of ones, where each column represents a word and each row represents a sentence. Each row of this matrix is iterated, and some of the ones are replaced by zeros, meaning that some words are removed for each sentence. The number of words to be removed in each sentence is defined by the previously described vector. In this step, we added the constraint that no value from the columns referred to the entities can be replaced by zero.

3. By the end of the loop, the final matrix has several $0s$ in random positions, but the columns referred to the relation entities only have $1s$, meaning that they will remain in the sentences.

3.2 Faithfulness

The first metric to be used for the evaluation of LIME explanations is called Faithfulness. The metric is presented in the paper "*ERASER: A Benchmark to Evaluate Rationalized NLP Models*" [DJR+20]. The authors start by discussing the growing importance of

[1] The details of the implementation can be found in the repository, together with results of some of the experiments carried out in this work: `https://github.com/thaisbeham/RE_Lime_evaluation`

explainability, specifically in the NLP field, and that the research has been carried out non-uniformly (different datasets, aims, and metrics). To overcome it, they propose a benchmark where multiple datasets and models are evaluated with specific metrics and human annotation. The work focuses on the evaluation of the rationales.

The benchmark is called **ERASER**: **E**valuating **R**ationales **A**nd **S**imple **E**nglish **R**easoning. To evaluate the rationales, it presents the metric Faithfulness, used to assess if the rationales of a model, in fact, informed its prediction. It is also measured if the model's rationales agree with the ones provided by humans. The assessment is carried out for different datasets, models, and NLP tasks (Sentiment analysis, QA tasks, Natural Language inference, etc), enabling comparison from different criteria [DJR+20].

Moreover, Faithful rationales are the set of words from a sentence that has a meaningful influence on its prediction from a model. To ascertain the degree to which the rationales are faithful, the authors introduce two metrics: Comprehensiveness and Sufficiency. The details about them are:

- Comprehensiveness: the degree to which all the required features were selected as rationales [DJR+20].
- Sufficiency: the degree to which the selected rationales are enough for the prediction. [DJR+20].

The rationales analyzed in the experimental part of this present work are the ones provided by LIME through its explanation process, as the method outputs a list of the most important words that led to the prediction.

3.2.1 Comprehensiveness

Comprehensiveness is calculated by the difference between the prediction probabilities of a sentence and its modified version without the rationales. The rationales and prediction probabilities are obtained via the LIME explanation.

The details of the implementation are:

- Given a sentence prediction by a model, the prediction is explained using LIME.
- LIME returns the class, rationales, and prediction probability for that prediction.
- A modified sentence is created by removing the rationales.
- The prediction of the modified sentence is explained by LIME which returns class, rationales, and prediction probability for that prediction.
- The difference between the prediction probabilities of the original and modified sentence is calculated for the outputted class of the original sentence. This final value is the Comprehensiveness score.

The logic behind this approach is that a higher prediction probability for a certain class indicates that a model has more confidence in its prediction. Comparably, a lower prediction probability reflects reduced certainty in the model's prediction for that class.

Given that $m(x_i)_j$ indicates the prediction probability for sentence x_i and class j and $m\left(\frac{x_i}{r_i}\right)_j$ indicates the prediction probability of class j from the sentence x_i without its rationals r_i, the Comprehensiveness metric is calculated by the difference in those predictions probabilities, as demonstrated in Equation 3.1.

$$\text{Comprehensiveness} = m(x_i)_j - m\left(\frac{x_i}{r_i}\right)_j \tag{3.1}$$

When calculating the difference between the prediction probabilities for the whole sentence and that of the sentence without the rationals, a positive and high value is expected. The intent is that this difference should be as big as possible, indicating that the model is much more confident in predicting a sentence with the rationales compared with a sentence without them.

Contrarily, a lower value of Comprehensiveness means that the rationales were less influential for the prediction. Moreover, a negative value indicates more confidence without the rationales [DJR+20]. Additionally, Comprehensiveness specifically focuses on the degree to which all the needed features for the prediction were selected.

Furthermore, constraints are added when creating the modified sentences to ensure that the relational entities are not removed.

Example of a LIME explanation of Sentence (1):

Ten million quake ***<survivors>*** *moved into makeshift* ***<houses>****.* (Sentence 1)

This sentence is extracted from the test set and the words marked in green are the rationales defined by LIME, where the importance of the word is represented by their level of saturation. The relational entities are "survivors" and "houses".

For this case, the modified sentence would be Sentence (2):

Ten ***<survivors>*** *into* ***<houses>*** (Sentence 2)

Sentence 2 removed the rationales but still kept the entities. Subsequently, the LIME explanation of it is carried out, and its prediction probability for the class outputted in Sentence 1 is used in the Equation 3.1.

3.2.2 Sufficiency

For Sufficiency, it complements the Comprehensiveness metric by proposing the opposite: instead of removing the rationals, it keeps them and removes all the other words. The constraint that the relational entities must continue in the sentence, whether they are rationales or not, is also maintained for this metric.

Given that $m(x_i)_j$ stands for the prediction probability of sentence x_i for class j, and let $m(r_i)_j$ be the prediction probability for class j of a sentence composed only with the rationales r_i, the Sufficiency is computed by the difference of the prediction probabilities as indicated in Equation 3.2

$$\text{Sufficiency} = m(x_i)_j - m(r_i)_j \tag{3.2}$$

The Equation 3.2 shows that the difference between the prediction probability of the original and the modified sentence is again calculated (for the class with the highest prediction probability of the original sentence). Sufficiency is expected to be the lowest possible (near zero or negative) since it indicates that only the rationales are already sufficient to make the prediction. Negative values for Sufficiency can indicate that the model is even more confident when having only the rationales.

Given the Sentence (1) from 3.2.1, the modified sentence with only the rationales (plus entities) would be:

million quake ***<survivors>*** *moved makeshift* ***<houses>*** (Sentence 3)

3.2.3 AOPC

Considering that LIME requires the user to set the number of rationales returned per explanation, we use an approach called AOPC where the metric (Sufficiency or Comprehensiveness) is calculated repetitively for different numbers of rationales, and the average of these results is taken as the final value of the metric for a given sentence. This is the same methodology followed by [DJR+20], and its name means "Area Over the Perturbation Curve” (AOPC), a method that derives from the ROAR, "RemOve And Retrain" from Hooker *et al.*(2019) [HEKK19].

The choice of the number of rationales is made by percentages of the total number of words in a sentence, referred to as bins. The authors [DJR+20] used the following bins: 1%, 5%, 10%, 20%, and 50%. Given k as the bins, $\beta + 1$ as the total number of bins, and i as the instance, the method applied for Comprehensiveness is defined by Equation 3.3. The equation shows the summation of the Comprehensiveness values (for different numbers of rationales) ranging from 0 until β; then the total is divided by $\beta + 1$, closing the average calculation. The analogous version is computed for Sufficiency.

$$\frac{1}{|\beta| + 1} \left(\sum_{k=0}^{\beta} m(x_i)_j - m\left(\frac{x_i}{r_{ik}}\right)_j \right) \tag{3.3}$$

Since our dataset presents a much smaller number of words compared to the paper (our average length is 19 words), many bins would result in the same and smaller number of words. For example, 1% and 5% of 19 words would both mean only one word, when rounding to the highest integer. Therefore, different bins were chosen to provide more meaningful results. They were: 20%, 50%, 70%, and 90%.

Furthermore, the random version of the metrics (Sufficiency and Comprehensiveness) is calculated as a way to compare the obtained results. Wherein the rationales are substituted by a set of randomly chosen words when creating the modified sentences. Likewise, the *AOPC* approach is applied in this configuration.

3.2.4 Faithfulness results for other tasks and related work

DeYoung et al. (2020) [DJR+20] provides an extensive list of Faithfulness metrics' results for different tasks and models, that can be found in Table 3.1. It demonstrated an interesting comparison by providing values for Attention and Gradient weights, LIME, and Random. The column Performance (Perf.) stands for macro F1 or accuracy.

Is notable that LIME performed well across all the tasks and got the best scores for "BoolQ", "Movies", "FEVER", "MultiRC", "CoS-E" and "e-SNLI". Additionally, it can be observed that the values for Comprehensiveness are included in the range -0.002 until 0.437 and Sufficiency, from -0.079 until 0.583.

Another work [EZMMA22] evaluates a different version of Faithfulness for Sentiment Analysis Explanations. Faithfulness is evaluated, in this case, by the difference in accuracy between the model prediction of whole sentences and the sentence with only the rationales (or referred to in the paper as "explanation"). Moreover, the authors also introduce the concept of *plausability*, which consists of measuring the agreement between explainable methods and human judgment. Together with LIME, the tools SHAP and *Anchors* are also evaluated. The results conclude that there is a "remarkable discrepancy" between the results of the three methods and that LIME considerably outperforms the others.

The work of [KDI+21] evaluates explanations in terms of Sufficiency and Comprehensiveness for the task of hate speech detection in Bengali language. Instead of using LIME to select the rationales, those are selected by permutation feature importance. The explainability is measured in 5 different models and the results can be found in Table 3.2.

Additionally, the work of [BKG23] calculates, among other metrics, Faithfulness to evaluate different explainable methods: SFFA [BKG23], L-Shapley and C-Shapley [CSWJ18] and IntGrad[STY17] for different datasets and two models: attention bi-directional LSTM (Attbilstm) and Convolutional Neural Network (CNN). The metrics are also calculated using the AOPC methodology. The results are displayed in Table 3.3. It can be seen that LIME outperformed some methods in some configurations and underperformed in others; in general, there was no huge discrepancy among the results of the methods, however SFFA had the best results in all experiments.

3.3 Stability

The second metric used for evaluating LIME explanations is called Stability, introduced by Burger, C., Chen, L., & Le, T. (2023)[BCL23]. It consists of evaluating how stable are the explanations of LIME based on the change of parameters considered of low importance. It is divided into two separate evaluations: inherent Stability and parameter Stability.

3.3.1 Inherent Stability

Inherent Stability comprises the observation of changes in the LIME output, specifically in the ranked list of rationales, by varying the random seed and the sampling rate. The *default* sampling rate in LIME is 5000, this means that for each sentence, in which the prediction is to be explained, LIME creates 5000 modified sentences that are used in the process of generating the explanation. There are no justifications for why the number 5000 is chosen but it seems to work well in practice [BCL23].

The metric consists of varying the rate from 500 to 10000, with 500 as the step. In each round the LIME explanation will return a ranked list of rationales and it will be compared with the list outputted by the explanation with the *default* sampling rate (5000). The comparison is carried out by using RBO (Rank Biased Overlap) [WMZ10]. Further, the whole process is repeated for another random seed.

The authors [BCL23] did this analysis for only one sample per model, however, it was observed that the choice of this sample could result in significantly different outcomes. Therefore, it was chosen to calculate the metric for 10 samples for each model, using the same 2 random seeds, and averaging the results of all samples for each sampling rate.

RBO

RBO is a statistical metric that measures the similarity between two ranked lists [WMZ10]. It calculates the expected average overlap between two ranked lists, considering progressively deeper levels of comparison. The degree of depth to be considered is tuned by the parameter p (persistence), varying from 0 to 1. The lower value of p gives more importance to the top elements of the ranked list, and higher p would distribute the emphasis more evenly across all ranks. Additionally, p equals zero means that only the first element of the list will be considered[WMZ10].

Let S and T be two ranked lists being compared, p the persistence, and A_d the agreement at depth d between the two lists; RBO calculation is defined by Equation 3.4 [WMZ10].

$$\mathrm{RBO}(S, T, p) = (1-p)\sum_{d=1}^{\infty} p^{(d-1)} \cdot A_d \tag{3.4}$$

RBO is a metric that ranges from 0 to 1. When equal to 0, it means that the two lists are totally disjoint, and when equal to 1, it means that they are identical[WMZ10].

Therefore, when analyzing inherent Stability of LIME explanations, given two lists of rationales to be compared, higher values of RBO would indicate that the lists are more similar and, therefore, the LIME explanations are more stable.

We followed the methodology of [BCL23] for our experiments, where the parameter p was set to 0.8. It means that higher importance is given to elements at the top of the list, however, all the elements can have an impact on the calculation.

3.3.2 Parameter Stability

The second type of Stability metric is called parameter Stability. It consists of replacing irrelevant words with their synonyms and testing the difference in the LIME output (the meaning of irrelevant is explained below). It is expected that LIME would not be significantly sensitive to irrelevant changes in the sentence. The implementation consisted of the following steps: replacement of words, calculation of angular similarity and comparison of rationales' lists. Those are presented in the next subsections.

Replacement of words

The metric process starts by selecting a sentence to be analyzed and obtaining its explanation by LIME, in order to define the rationales. Further, we define the words capable of being replaced by their synonyms, referred to as unimportant words, as all the words with exception of the rationales and the relational entities.

The paper that introduces the metric [BCL23] carried out the replacement of unimportant words by using the embedding *paragram-sl999*. However, it did not present very good results during our tests since words were frequently replaced by their translation in different languages instead of their synonyms. Thus, it was decided to use the embedding *paragram-ws353* (using the library *gensim*[ŘS10]) in our experiments. An example of replacement using these two embeddings in one of the sentences from the test set is shown below:

Original: "Skype, a free software, allows a hookup of multiple computer users to join in an online conference call without incurring any telephone costs."

WS353: Skype, another free-of-charge software, enables another hook-up de diverse computers customers towards joining across another on-line conferences calls unless incurs everything phone costs.

SL999: Skype, une freie software, enables une hook-up du diverse computers customers pour joining at either on-line conferences calls ohne incurs everything phone costs.

After defining the embedding for the replacement, some constraints were adopted, following the methodology of [BCL23]. Those were:

- Ensure that a word can only be replaced by one of the same Part of the Speech (POS). Additionally, it is allowed to change verbs with nouns (and vice-versa).

- The number of words to be replaced per sentence is constraint by either (what happens first):
  - Angular Similarity (explained below) between the original and modified sentence should remain above 0.5.
  - Maximum number of modified words is 5.

Angular similarity

The angular similarity measures the degree to which two pieces of text carry the same meaning [Kag20] and is calculated to ensure that the difference between the original and modified sentence is not so prominent. Thus, both sentences should remain with Angular Similarity above 0.5.The calculation follows the methodology presented in the paper [BCL23].

First, the sentences are converted into an embedded form using the *Universal Sentence Encoder* presented in the paper [CYyK+18]. Finally, the Angular similarity is computed by the inner product of the embedded sentences' vectors.

Comparing Rationales lists

After the previous steps, two sentences are obtained: the original and the perturbed one. Subsequently, an analysis is conducted to determine whether it impacted the explanations of LIME by examining if the perturbed sentence yields a significantly different output compared to the original sentence.

For that, the list of LIME rationales of both sentences are compared using RBO (with $p = 0.8$), as they consist of ranked lists. If the RBO calculation is below 0.5, it means that the output of the modified sentence was considerably different. In this situation, it means that the LIME explanation was not stable and it is referred to as a "successful attack".

3.4 Global Inference

The third metric, Global Inference, is intended to evaluate if it is possible to have a broad understanding of how the model works based on the local analysis of specially picked instances. LIME provides a tool for this task, called SP-LIME, which stands for "Sub-modular Pick-LIME" [RSG16]. Therefore, the Global Inference metric intends to evaluate qualitatively the effectiveness of understanding the model predictions globally using the SP-LIME tool, specifically for the Relation Extraction task.

The idea behind SP-LIME is to display to the user some explanations that are specially selected to provide meaningful information instead of randomly looking at some of them. The chosen explanations should cover important components, they should not be redundant and the number of explanations should be low enough to allow quality

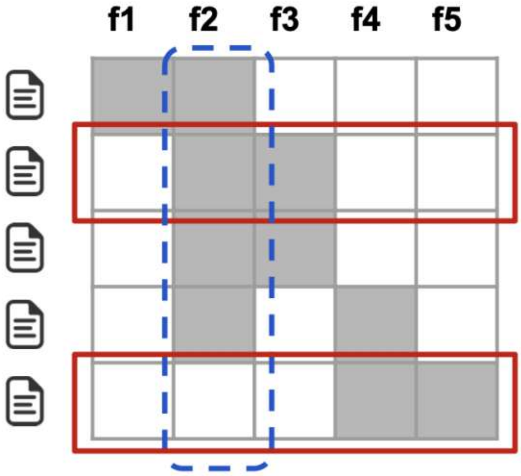


Figure 3.1: Representation of SP-LIME for textual data, where each row represents a document, and each column a feature (words). Given that feature 2 has the highest importance by belonging in most explanations, the documents chosen are the second and last, so all the features, except f1, would be covered in the explanations. Extracted from: [RSG16].

analysis. Important components are those features that are present in a high number of explanations.

Firstly, it works by assigning an importance score to the features in the dataset based on the number of explanations they are included. For textual data, it means that words used to explain several sentences have a higher importance score. The next step is to select explanations that present different rationales. Finally, the result is a set of explanations that comprise important words and those are not repeated [RSG16].

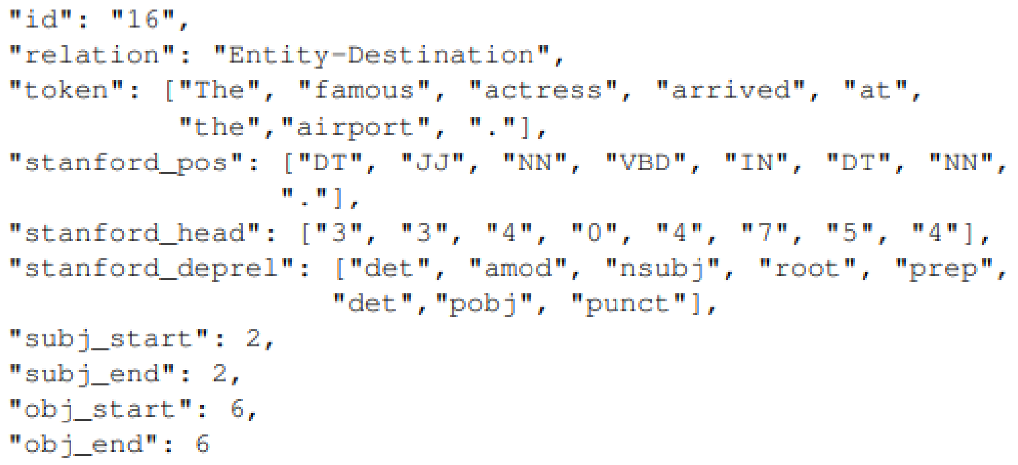
The Figure 3.1 extracted from [RSG16] illustrates this method for textual data. The columns stand for the features (words), and the rows for the documents. Feature 2 is considered the most important as it explains the highest number of documents (4 out of 5). Therefore, at least one of the selected documents should contain this word. The second and last documents are the ones selected since both contain information about all the features (especially f2), besides f1.

3.5 Experimental setup

3.5.1 Pre-processing

One of the models used in the experiments, the AGGCN, requires as input not only the sentence but other information such as dependency parsing and the locations of the relational entities. To illustrate, the Figure 3.2 shows an example of a sample from the test set.

The paper that presented the model [GZL19] does not specify how the preprocessing was carried out. Since LIME creates modified samples and those obtain their prediction from the target model, they consist of strings and do not carry information about dependency parsing, contrary to the dataset used to train the model. Therefore, it was necessary to analyze how it was done and replicate it to apply on the created samples from LIME so those can be inputted in the target model.



```
"id": "16",
"relation": "Entity-Destination",
"token": ["The", "famous", "actress", "arrived", "at",
          "the","airport", "."],
"stanford_pos": ["DT", "JJ", "NN", "VBD", "IN", "DT", "NN",
                 "."],
"stanford_head": ["3", "3", "4", "0", "4", "7", "5", "4"],
"stanford_deprel": ["det", "amod", "nsubj", "root", "prep",
                    "det","pobj", "punct"],
"subj_start": 2,
"subj_end": 2,
"obj_start": 6,
"obj_end": 6
```

Figure 3.2: Example of an instance from the dataset SemEval 2010 - task 8 [HKK+10] used in the AGGCN model. It can be noted that the model not only receives the sentence as input but also information about dependency parsing and the position of the relational entities.

Given the analysis of Figure 3.2, the field `stanford_pos` refers to POS (Part-of-Speech), `stanford_head` stands for the tag head and `stanford_deprel` indicates the dependency relations. The names suggested that the Stanford dependency parsing was used. Therefore, the library stanza [QZZ+20] from Stanford NLP Group, was utilized to calculate the dependency parsing "POS", "head" and "DepRel" for the samples created by LIME.

Previously, the Standford NLP Group used the called "Stanford Dependency Parsing" which only contemplated the English language. Later on, it adopted instead the Universal Dependency (UD) Parsing, which enables its use for any language [DMM08]. The stanza library comtemplates the Universal Depency parsing.

The observance of samples from the dataset revealed that the field "DepRel" was referred to the Stanford dependency parsing, instead of UD. Since the differences are minimal, it was decided to still continue with the stanza library. On the other hand, POS was likewise obtained using the Stanford dependency parsing, which, in this case, is quite different from UD tags. In this situation, the stanza library allows additionally to use POS from the Stanford dependency parsing nomenclature by setting the parameter to "xpos" instead of "upos" when requesting the parsing[Gro20]. Their difference can be seen in Table 3.4. Regarding the field "head", there is no significant difference among both methods.

The "subj_start", "subj_end", "obj_start", "obj_end" fields refer to the position of the relational entities respectively. In the dataset, there is no differentiation between start and end since the entities always consist of only one word. The position count starts at zero and punctuation is also taken into consideration.

3.6 Models

3.6.1 AGGCN

The Neural Model chosen for the evaluation is Attention Guided Graph Convolutional Networks (AGGCNs)[GZL19]. The parameters chosen were the same as the ones provided in their repository [2]. The reason is that it reached almost the state-of-the-art performance from the chosen dataset and provided enough information to allow the reproducibility of the results correctly.

3.6.2 Naive Bayes

In comparison, a Machine Learning model that does not use neural networks was chosen. Naive Bayes was the final choice since it is a very popular method and performed fast and significantly well in our dataset. The algorithm comes from the Bayes theorem that describes the conditional probability P(A|B). The naive part comes from the assumption that the features have no correlation with each other and all contribute to the probability of the class[Cam22]. It was implemented using the scikit-learn library [PVG+11] with the function "GaussianNB" that enables classification.

3.7 Using LIME with AGGCN

The LIME function to provide explanations for textual data is called "explain_instance" and can be observed below:

```
explain_instance(text_instance, classifier_fn, labels=(1,),
top_labels=None, num_features=10, num_samples=5000,
distance_metric='cosine', model_regressor=None)
```

It can be noticed that it requires from the user two inputs: a sentence ("text_instance") and a function ("classifier_fn"). The latter should be one that must only receive as

[2] https://github.com/Cartus/AGGCN/tree/master

input a sentence or list of sentences and outputs the prediction probabilities for all classes of the inputted sentence using the target model (model that wants to be explained).

In detail, the function `classifier_fn` takes as input a list of strings (the sentences) and outputs the prediction probabilities as a NumPy array with format (d, k), where d stands for the number of strings and k, the number of classes. For scikit-learn classifiers, it is simply the `classifier.predict_proba` function. For complex models like AGGCN, it is necessary to create a separate function that follows these constraints. The AGGCN model makes it especially challenging since it not only requires a sentence as input but also several fields, such as tags, parsing, and position of the entities (as shown in Figure 3.2). Therefore, it was necessary to make a workaround to enable LIME functionality since it strictly only receives a sentence as input .

Thus, to make possible the use of LIME to explain AGGCN predictions, the function `classifier_fn` was implemented comprising several steps inside it. Those steps were:

1. The function `classifier_fn` receives as input several sentences, as expected.
2. Each sentence is preprocessed and added all the required information (dependency parsing, position of entities, tokens, etc) to fit the format of the original dataset, as displayed in Figure 3.2.
3. The preprocessed samples are saved together in a newly created JSON file.
4. The model reads the JSON file and outputs the prediction probabilities in the format of a (d, k) numpy array, where d is the number of sentences and k the number of classes.

These steps can be visualized in the Figure 3.3 where the two first elements of the diagram stand for the LIME process of creating the perturbed samples and are carried out inside the `LimeTextExplainer` method. The next 3 steps are done inside the `classifier_fn` function and should be constructed by the user. In our case the approach used was to develop the preprocessing of each sentence and save them in a JSON file, then provide this file as input to the AGGCN model and return the prediction probabilities.

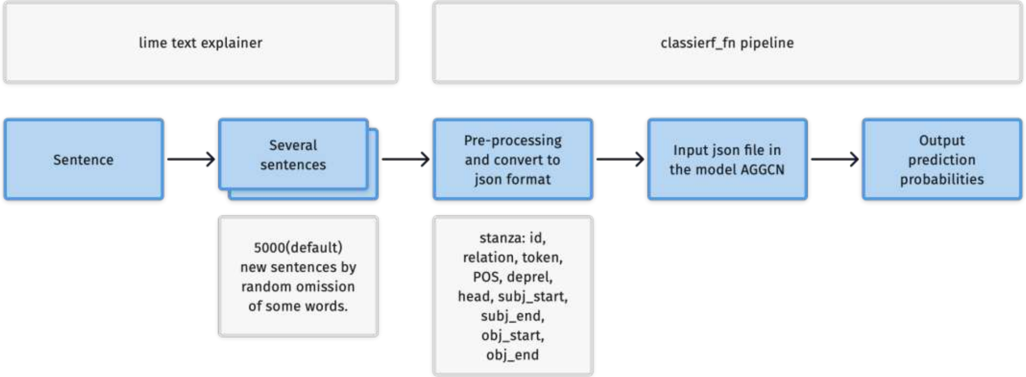

lime text explainer
classierf_fn pipeline
Sentence
Several sentences
Pre-processing and convert to json format
Input json file in the model AGGCN
Output prediction probabilities
5000(default) new sentences by random omission of some words.
stanza: id, relation, token, POS, deprel, head, subj_start, subj_end, obj_start, obj_end

Figure 3.3: LIME explanations steps for AGGCN model. Thu diagram indicates the steps done inside the created function "`classifier_fn`", which includes the preprocessing of each sentence and storing them in a JSON file, providing this file as input to the model, and returning the prediction probabilities in required format. The steps carried out by LIME text explainer include: taking a sentence and creating variations of it.

Table 3.1: Faithfulness metrics (Comprehensiveness and Sufficiency) for different tasks, models, and datasets. Perf. stands for performance in macro F1 or accuracy - Modified from [DJR+20]

		Perf.	Comp. ↑	Suff. ↓
Evidence Inference				
GloVe + LSTM	Attention	0.429	-0.002	-0.023
GloVe + LSTM	Gradient	0.429	0.046	-0.138
GloVe + LSTM	Lime	0.429	0.006	-0.128
GloVe + LSTM	Random	0.429	-0.001	-0.026
BoolQ				
GloVe + LSTM	Attention	0.471	0.010	0.022
GloVe + LSTM	Gradient	0.471	0.024	0.031
GloVe + LSTM	Lime	0.471	0.028	-0.154
GloVe + LSTM	Random	0.471	0.000	0.005
Movies				
BERT+LSTM	Attention	0.970	0.129	0.097
BERT+LSTM	Gradient	0.970	0.142	0.112
BERT+LSTM	Lime	0.970	0.187	0.093
BERT+LSTM	Random	0.970	0.058	0.330
FEVER				
BERT+LSTM	Attention	0.870	0.037	0.122
BERT+LSTM	Gradient	0.870	0.059	0.136
BERT+LSTM	Lime	0.870	0.212	0.014
BERT+LSTM	Random	0.870	0.034	0.122
MultiRC				
BERT+LSTM	Attention	0.655	0.036	0.052
BERT+LSTM	Gradient	0.655	0.077	0.064
BERT+LSTM	Lime	0.655	0.213	-0.079
BERT+LSTM	Random	0.655	0.029	0.081
CoS-E				
BERT+LSTM	Attention	0.487	0.080	0.217
BERT+LSTM	Gradient	0.487	0.124	0.226
BERT+LSTM	Lime	0.487	0.223	0.143
BERT+LSTM	Random	0.487	0.072	0.224
e-SNLI				
BERT+LSTM	Attention	0.960	0.105	0.583
BERT+LSTM	Gradient	0.960	0.180	0.472
BERT+LSTM	Lime	0.960	0.437	0.389
BERT+LSTM	Random	0.960	0.081	0.487

Table 3.2: Faithfulness Performance in Hate Speech detection for Bengali language task, extracted from [KDI+21]

Classifier	Comprehensiveness	Sufficiency
GBT	0.79	0.25
Conv-LSTM	0.73	0.15
Bangla BERT	0.78	0.25
XML-RoBERTa	0.84	0.44
mBERT-uncased	0.81	0.35
mBERT-cased	0.76	0.28

Table 3.3: Comparison of Comprehensiveness and Sufficiency across 2 models and 3 datasets, evaluating the rationales extracted from several methods, including LIME. Extracted from [BKG23]

Model	Dataset		SFFA	L-Shapley	C-Shapley	IntGrad	LIME
Attbilstm	IMDB	Compr. ↑	0.643	0.4136	0.127	0.423	0.459
		Suff. ↓	0.020	0.083	0.101	0.061	0.185
	YELP	Compr. ↑	0.631	0.406	0.394	0.402	0.439
		Suff. ↓	0.110	0.266	0.268	0.150	0.234
	AG news	Compr. ↑	0.721	0.295	0.259	0.483	0.291
		Suff. ↓	0.003	0.070	0.089	0.031	0.103
CNN	IMDB	Compr. ↑	0.476	0.438	0.418	0.408	0.375
		Suff. ↓	-0.134	-0.125	-0.118	-0.115	0.014
	YELP	Compr. ↑	0.513	0.468	0.466	0.472	0.207
		Suff. ↓	-0.138	-0.133	-0.132	-0.141	0.011
	AG news	Compr. ↑	0.684	0.212	0.167	0.351	0.275
		Suff. ↓	-0.021	0.134	0.162	0.044	0.111

Table 3.4: Difference between xpos and upos for an example sentence (modified from [Gro20])

Word	**upos**	**xpos**
Barack	PROPN	NNP
Obama	PROPN	NNP
was	AUX	VBD
born	VERB	VBN
in	ADP	IN
Hawaii	PROPN	NNP
.	PUNCT	.

CHAPTER 4

Results & Discussion

This chapter will present the results of the metrics: Faithfulness, Stability, and Global Inference. It will also depict the models' performance and running times to compute the metrics.

4.1 Models' performance

Achieving high performance in the models was not the objective of this project. However, it is worth mentioning to assess whether their variations impact LIME outputs. Table 4.1 shows the classification metrics Precision, Recall, F1-Score, and Support for each one of the classes and their Macro Average measure. The same is displayed for AGGCN model at Table 4.2.

Table 4.1: Naive Bayes Model Performance for the task Relation Extraction with the SemEval 2010 - task 8 dataset[HKK+10]

Category	Precision	Recall	F1-Score	Support
Cause-Effect	0.705	0.838	0.766	328
Component-Whole	0.499	0.603	0.546	312
Content-Container	0.780	0.573	0.661	192
Entity-Destination	0.785	0.613	0.688	292
Entity-Origin	0.770	0.376	0.505	258
Instrument-Agency	0.78	0.090	0.161	156
Member-Collection	0.723	0.369	0.489	233
Message-Topic	0.769	0.318	0.450	261
Other	0.245	0.595	0.347	454
Product-Producer	0.582	0.277	0.375	231
Macro Avg	0.664	0.465	0.499	

Table 4.2: AGGCN Model[GZL19] Performance for the task Relation Extraction with the SemEval 2010 - task 8 dataset [HKK+10]

Category	Precision	Recall	F1-Score	Support
Cause-Effect	0.918	0.918	0.918	328
Component-Whole	0.839	0.837	0.838	312
Content-Container	0.803	0.891	0.844	192
Entity-Destination	0.882	0.925	0.903	292
Entity-Origin	0.837	0.837	0.837	258
Instrument-Agency	0.768	0.744	0.756	156
Member-Collection	0.820	0.901	0.859	233
Message-Topic	0.819	0.935	0.873	261
Product-Producer	0.802	0.858	0.829	231
Macro avg	0.832	0.872	0.851	

4.2 Faithfulness

This section comprises the results of the Faithfulness metric. Since the running times for the LIME explanations were extremely high, with an approximate duration of 30 minutes for the explanation of one sentence for the AGGCN model, it was decided to reduce the parameter sampling rate of LIME from 5000 (*default*) to 2000 to speed up the computation. It means that, for each sentence, LIME will create 2000 perturbed sentences instead of previously 5000. Nevertheless, the running time still lasted almost 4 days in this configuration, as can be seen in the section 4.5. Additionally, the metric "Inherent Stability" also confirmed that the sampling rate reduction does not significantly affect the outputs (presented in subsection 4.3.1).

It can be observed in Table 4.3 the results of the Faithfulness metric, divided into Comprehensiveness and Sufficiency. Each value represents the average result computed for 100 samples. The F1 score was calculated using the whole test set. Additionally, the AOPC method is applied for each sample for both normal and random configurations. The latter consists of the metric being calculated several times for each sentence, with a different number of rationales per round, and then the result is averaged. The number of rationales is defined by percentages of the total words, called bins. The bins chosen were 20%, 50%, 70%, and 90%. More details about AOPC in subsection 3.2.3.

Comprehensiveness a metric that higher results represent better performance, as it is calculated by the difference between the prediction probabilities of the whole sentence and the sentence without the rationals (refer to subsection 3.2.1). Consequently, the expectation is for this difference to be maximized, meaning that the model exhibits reduced confidence without the inclusion of rationales.

The Comprehensiveness results for both models (AGGCN and Naive Bayes) are positive and greater than random calculations, indicating a favorable performance. Moreover, when analyzing the outcomes provided in the ERASER paper [DJR+20], in Table 3.1,

Table 4.3: Faithfulness results (Comprehensiveness and Sufficiency) and F1 score for AGGCN and Naive Bayes models, using LIME sampling rate of 2000. Random models stand for random selection of rationales.

Model	F1	Comprehensiveness ↑	Sufficiency ↓
AGGCN	0.85	0.3669	0.0471
AGGCN random		0.2222	0.1478
Naive Bayes	0.50	0.3772	0.0633
Naive Bayes random		0.2572	0.1875

for different tasks, the average performance of LIME is 0.186 for this metric, which shows that our outputs are higher than this. Thus, it seems to indicate that the degree to which all the required features were selected as rationales is high.

For Sufficiency, it is observed that values are near zero for both models, with AGGCN's outcome being slightly below the one from Naive Bayes and both of them being under the random results. Since Sufficiency is a metric calculated by the difference between the prediction probabilities of the whole sentence and the sentence with only the rationales, values near zero indicate that the model could achieve similar confidence with only the rationales. This fact is crucial as it highlights LIME's ability to choose rationales that are sufficient for making the prediction.

The analysis of Comprehensiveness and Sufficiency outputs points to a high degree of all the required features being selected as rationales and those being enough for the prediction. The findings suggest that the rationales are faithful, thus indicating that the degree to which those words contributed to the prediction is high. This demonstrates that LIME can indeed provide good explanations by generating faithful rationales, confirming the findings of [DJR+20] and [MSY+21].

It is worth stating that the metric does not provide exact thresholds to define what would precisely represent faithful rationales since the results vary a lot regarding the model and dataset used. Nonetheless, as previously mentioned, the outcomes indicate a positive overall performance. Additionally, the idea proposed by [DJR+20] which introduced the metrics, is that researchers should submit their results to a common leaderboard[1] to allow a broader comparison of values. Currently, no values for this exact task or dataset are there.

4.3 Stability

This section will exhibit the results for inherent and parameter Stability metrics.

[1] The leaderboard can be accessed at https://www.eraserbenchmark.com.

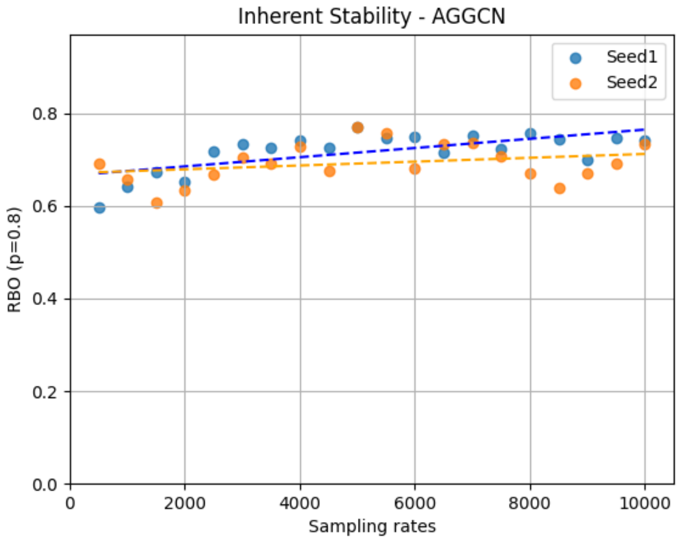


Figure 4.1: Inherent Stability results for AGGCN model: each data point corresponds to the RBO score, comparing the ranked rationales list generated by LIME at various sampling rates indicated on the x-axis against the baseline with a sampling rate of 5000 (*default*). The process is repeated for another random seed. The data points presented the average computed over 10 samples. The parameter p in RBO is set to 0.8.

4.3.1 Inherent Stability

The graphs in Figure 4.1 and Figure 4.2 present the results of the inherent Stability computation for evaluating the impact in LIME explanations when varying the sampling rate in 20 different rates ranging from 500 to 10000 (with 500 as the step) in 2 different seeds (defined as 1 and 28989) for 10 sentences from the test set. Each data point in the visualizations refers to the averaged calculation of RBO of the 10 sentences for the respective seed and sampling rate. The parameter p from RBO is set to 0.8.

RBO is a metric used for calculations of similarity between ranked lists. The lists compared are: the rationales derived from an explanation using a given sampling rate against the rationales obtained using the *default* sampling rate.

The Table 4.4 summarizes the data presented in the visualizations by showcasing the mean and median of their data points. The values are expressed in terms of RBO calculations, ranging from 0 to 1. A value of 1 signifies that the rationales of the compared sentences are identical, while a value of 0 indicates that they are entirely different. The obtained values of 0.70 to 0.74 suggest that the LIME explanations were not significantly affected by the change in the sampling rate and random seed. The referred work [BCL23], presented mean and median results above 0.80% in 3 out of 4 experiments, confirming our findings.

Furthermore, the graph Figure 4.1 revealed that, for the AGGCN model, the LIME

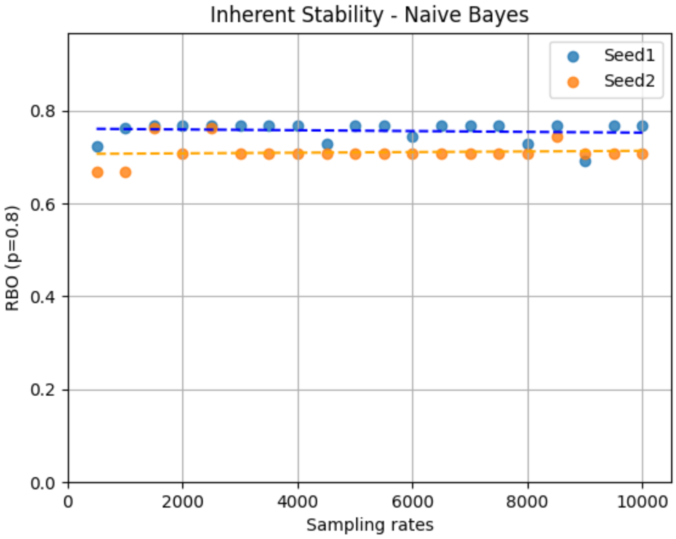


Figure 4.2: Inherent Stability results for Naive Bayes model: each data point corresponds to the RBO score, comparing the ranked rationales list generated by LIME at various sampling rates indicated on the x-axis against the baseline with a sampling rate of 5000 (*default*). The process is repeated for another random seed. The data points presented the average computed over 10 samples. The parameter p in RBO is set to 0.8.

Table 4.4: Inherent Stability metric calculated in terms of similarity among different sampling rates and seeds. Similarity is calculated with RBO. The mean and Median were computed by averaging across 10 samples.

Model	Mean RBO	Median RBO
AGGCN	0.70	0.71
Naive Bayes	0.73	0.74

explanations that used sampling rates ranging from 2000 to 8000 presented minimal differences among each other. This consideration is important for efficiency, especially given the highly time-consuming nature of explanations. Scaling down from 5000 to 2000 already yields significant benefits in reducing computational demands. Consequently, we selected 2000 as the sampling rate for the Faithfulness experiments in section 4.2.

The results for Naive Bayes, exhibited in Figure 4.2, indicate that the reduction or increase of the sampling rate did not significantly impact the LIME explanations output (with some exceptions for the rates 500 and 1000). Therefore, it seems that complex models can suffer more from the choice of the correct sampling rate than simpler models and that the sampling rate of 2000 appears to be a good choice to deliver an output similar to the sampling rate of 5000.

Examining the effect of changing the random seed in our experiments depicts that it does not seem to have a significant impact on the explanations in both models. Nonetheless, this impact was even less significant in the Naive Bayes.

4.3.2 Parameter Stability

The parameter Stability metric is calculated in terms of the rate of "*attacks*" effectiveness. It compares the LIME explanations of the original sentence and the perturbed one (with replacement of some of its words by their synonyms). If their explanations are significantly different (in terms of the list of rationales), an attack was successful, as explained in depth in subsection 3.3.2. The results are displayed as the percentage of sentences that suffered an effective attack. Consequently, lower values indicate that LIME explanations were more stable regarding parameter Stability.

The Table 4.5 illustrates the results, where 100 sentences from the test set were evaluated. It can be noticed that approximately 30% of both models' explanations of modified sentences were susceptible to significantly different outputs when compared to the original ones. The outcome partly agrees with the study [BCL23], where the metric is introduced, since they obtained rates of 37.06%, 55.56% for IMDB dataset and 6.25%, 3.37% for HateSpeech dataset, for FNN and BERT models each, respectively. Due to the results varying significantly when compared with the previous work, it can imply that the stability of the explanations is highly dependent on the chosen dataset and model.

The average of RBO calculations is also presented in Table 4.5. They are quite similar between both models and differ by only 0.01. The authors [BCL23] obtained 0.417, 0.436 for IMDB and 0.328, 0.442 for HateSpeech, for FNN and BERT, respectively, indicating that our RBO calculations reveal a better performance on average compared to their values, depicting that we obtained more stable explanations regarding parameter Stability.

Table 4.5: Parameter Stability results expressed in terms of Attack Success Rate. A success attack is considered when the perturbed sentence provides a significantly different output from LIME compared with the original one.

Model	Attack Success Rate ↓	RBO ↑
AGGCN	32%	0.5715
Naive Bayes	27%	0.5761

It can also be observed that Naive Bayes performed slightly better than AGGCN, with 27% against 32%. It confirms that models with less complexity are more robust and less susceptible to an attack.

4.4 Global Inference

Global Inference is a metric that intends to evaluate the performance of LIME in providing a global understanding of the model. It assesses the outcomes provided by the SP-LIME, a tool inside LIME intended for global explanations, by choosing a set of representative samples using submodular pick optimization [RSG16].

The function SP-LIME was run for the first 2000 samples of the test set and with the model Naive Bayes since this tool showed to be highly expensive computationally, thus the simpler model would provide faster running times. Nonetheless, it needed 10 minutes to finalize, ten times more than the calculations of the other metrics for this model. It led to the conclusion that the computation would be impractical for the AGGCN model.

The number of explanations outputted is a parameter set by the user with the *default* value of 5. It was chosen 10 explanations since the dataset consists of 10 different classes and was expected more or less an even distribution among them. The number of rationales per explanation was set to 5 since it showed as a good value in previous experiments and it agrees with the documents' average length.

The result was that SP-LIME returned the explanation of 10 samples, intended to be a represented set of the whole 2000 samples. However, the class distribution was significantly unbalanced, given that 7 out of 10 explanations were from the class "*Other*", 2 from class "*Cause-Effect*" and 1 from "*Container-Content*". The outputs of the latter classes and one from "*Other*" can be observed in Figure 4.3, Figure 4.4, Figure 4.5 and Figure 4.6.

The first conclusion inferred from the results was that, specifically for multi-class problems, it is challenging to gain a complete understanding of the model when 70% of the samples chosen by SP-LIME belong to the same class. In our case, it was especially negative since the class "*Other*" does not intend to be meaningful and simply refers to enclosing all the relations that do not fit in the specified classes.

Regarding the positive sides of the metric, both explanations for "*Cause-Effect*" class (Figure 4.4 and Figure 4.5), showed the highest importance for the words "cause" and "caused", respectively, which provides a good understanding of how the model is interpreting this class and indicates "*plausibility*" (term explained in subsection 3.2.4) of the explanation since these rationales would probably agree with an explanation provided by a human. For Figure 4.4, the results are even more interesting since both relational entities are also selected as rationales ("fat" and "disease").

The results plausibility is likewise presented in Figure 4.6, where the word of highest importance is "bottle", which indicates a container, suggesting that the model can detect these features and relate them with the class "*Content-Container*". Additionally, this word is one of the relational entities. The word "in", at third place of importance, also demonstrates interesting results since it is usually used to express the relation between a "Content" and a "Container".

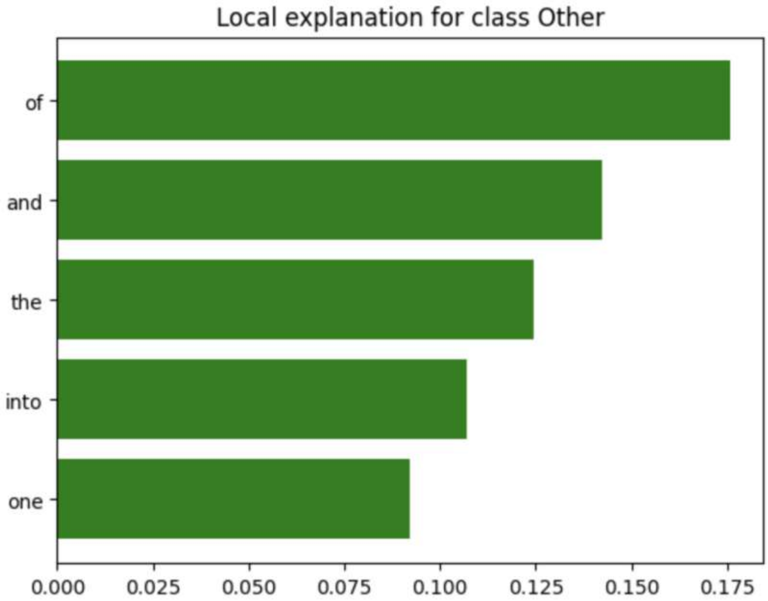


Figure 4.3: LIME-SP output for class Other. The original sentence is: "*A year later, Arlonzia married Bizzell Pettway and moved into one of the new <e1>houses</e1> built by the <e2>government</e2>.*", the entities are "houses" and "government" and it was incorrectly predicted, since the original class is "Product-Producer".

One of the goals of the method is that the explanations should not be redundant. Thus, they should have different rationales. However, the class "*Cause-Effect*" had two explanations where the most important word was a variation of the verb "cause" in each, resulting in a certain redundancy. It is noticeable that if lemmatization had been performed on the rationales before giving their importance scores, it would have prevented the selection of explanations containing both the words "cause" and "caused", potentially delivering more interesting results.

Lastly, the result of class "Other" (Figure 4.3), specifically for our problem, can not be used for interpretation since the class itself does not present a significant meaning. Nevertheless, it could be noticed that the explanation chosen was actually from a sample that got a wrong prediction (the original class is "Product-Producer"). Understanding whether the model's decisions were correct or incorrect can provide valuable information for assessing the model's weaknesses. Furthermore, the method SP-LIME does not display the sentence referring to the explanation to the user, only the explanation itself. Hence, to be able to identify from which sample that explanation belongs, one has to manually look for the sample that matches the words in the explanation.

In light of all the above stated, some suggestions for improvements are presented:

- In multi-class problems, ensure a more balanced distribution of the classes when selecting the explanations to avoid most of them being from only one class.

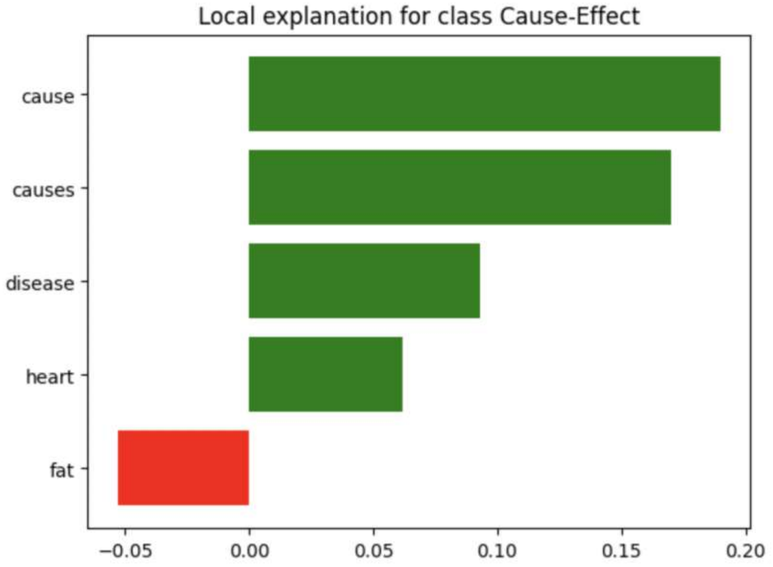


Figure 4.4: SP-LIME output for class Cause-Effect. The original sentence is: "*The <e1>fat</e1> and cholesterol cause heart <e2>disease</e2>; the animal protein causes cancer.*", where "fat" and "disease" are the relational entities

- Indicate if the outputted class is the correct one or not.
- Apply lemmatization before giving importance scores to the rationales since it would avoid redundant explanations, as in the case of our results that presented the words "causes," "cause," and "caused".
- Indicate the sentence associated with the given explanation.

4.5 Running time

LIME is shown as a highly expensive computational method. Since it operates through a surrogate model, maintaining a high sampling rate is crucial to replicate the target model locally effectively. However, this elevated sampling rate leads to excessively long running times for complex models. The Table 4.6 depicts the running times for all the metrics separated by model.

The time for Faithfulness refers to one metric (among Comprehensiveness, Sufficiency, and the random version of them) since their running times were quite similar and each of them was calculated in the same configuration of 100 samples and 2000 as the sampling rate. Ultimately, the consumed time in total was 4 times the computation of approximately 80 hours, resulting in about 320 hours. Inherent Stability was calculated with 10 samples for each model and Global Inference with 2000 samples.

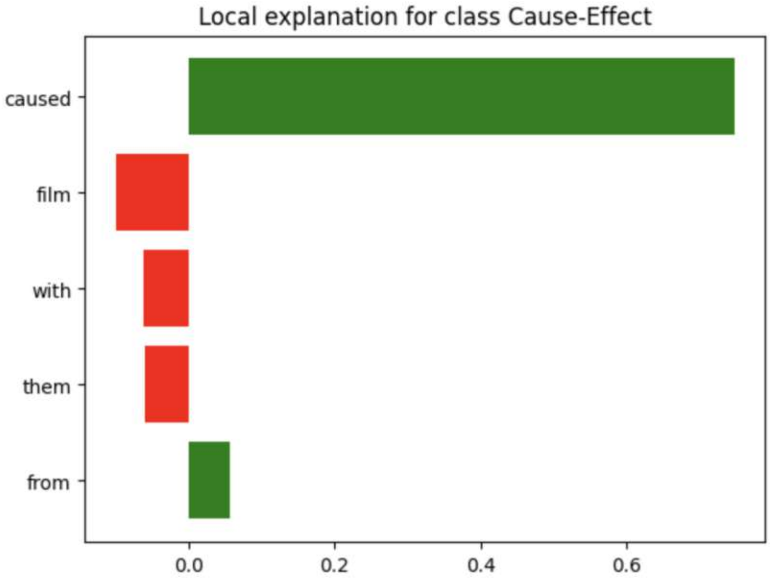


Figure 4.5: Second SP-LIME output for class Cause-Effect, the original sentence is: "*The <e1>visit</e1> caused a <e2>sensation</e2> on the Whangaparaoa Peninsula with people reporting the orca and rushing out to photograph and film them, from a safe distance.*", the relational entities are: "visit" and "sensation".

Table 4.6: Running times per metric and per model. The Faithfulness metric indicates the average time of each of the metrics (Comprehensiveness, Sufficiency and Random version of both

Metric	AGGCN	NB
Faithfulness	79.5 h	1 min
Inherent Stability	60.8 h	1 min
Parameter Stability	20 h	1 min
Global Inference	-	10 min

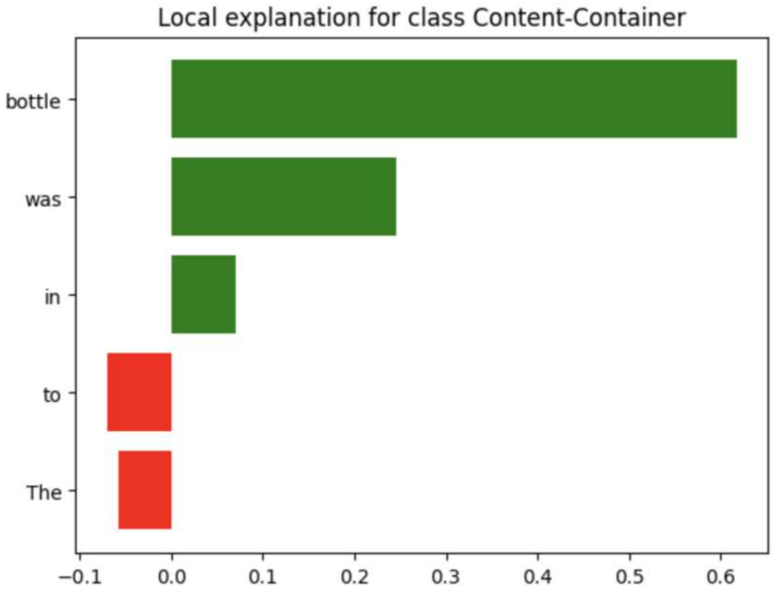


Figure 4.6: SP-LIME output for class Content-Container. The original sentence is: "*The <e1>drug</e1> was in a <e2>bottle</e2> that was not prescribed to her*", the relational entities are "drug" and "bottle".

CHAPTER 5

Conclusion

This work proposed to create modifications for the LIME library to enable the correct functionality for Relation Extraction tasks which was successfully achieved. The implementation was tested with two different models and for calculating the metrics of Faithfulness and Stability. The results and details of the implementation are available in the repository: https://github.com/thaisbeham/RE_Lime_evaluation.

Regarding the metric Faithfulness, the results for Sufficiency exhibited values near zero, which suggests that the sentences with only the rationales could almost reproduce the same performance as the prediction in the original sentences, revealing that the rationales were indeed representative of the important information in the document and practically sufficient to provide same results. These findings suggest that LIME was able to select faithful rationales in our experiments, confirming the conclusion of [RSG16] and [MSY$^+$21]. Furthermore, the metric Comprehensiveness showed favorable results, surpassing random calculations and indicating that the model performance got more confident (higher prediction probability) in sentences with rationales compared with sentences without them. Nonetheless, Faithfulness consists of a metric highly dependent on the model and dataset, culminating in a non-straightforward comparison with values from other works.

Moreover, our findings on the Stability metric reveal that LIME explanations possess inherent Stability. This conclusion was delivered by additional comparison with results from the study of Burger, C. *et al.* (2023)[BCL23]. Concerning parameter Stability, our determinations demonstrate that the majority of LIME explanations were stable regarding changes in unimportant features for our experiment setup. Nevertheless, the comparison with the previously mentioned work [BCL23] showed divergent outcomes, which can indicate that this stability presents a correlation with the type of task, model, and dataset used, thus diminishing the reliability of the method.

Subsequently, the Global Inference showed that explaining a model's predictions globally using a local method is challenging, especially when dealing with textual data and multi-class problems. Some suggestions to make the output of SP-LIME more meaningful are: ensuring a more or less even distribution of the explanations for all the classes, considering lemmatization before defining the most important words, indicating the sentence referred to in the explanation and if the explanation was for a sample correctly predicted or not.

In conclusion, LIME, when evaluating RE tasks, is a method that needed a correction in its functionality to return the explanations properly. After the modifications, the method depicted several positive performances, such as: indication that LIME rationales are faithful and its explanation are stable (observed for our experimental conditions). On the other hand, high computational costs and the deficits of SP-LIME to effectively understand the model globally are factors that need to be balanced out when choosing LIME for explanations.

Lastly, the main contribution of this work includes: the proposal of modifications for coherent functionality of LIME explanations in Relation Extraction problems and evaluation of Faithfulness of LIME rationales, stability of its explanations, and global explanation via SP-LIME. Suggestions for future works are: repetition of the experimental setup for a different dataset and the implementation of the proposed improvements for the tool SP-LIME.

List of Figures

List of Tables

Bibliography

[ABV+20] Julia Amann, Alessandro Blasimme, Effy Vayena, Dietmar Frey, Vince I Madai, and Precise4Q Consortium. Explainability for artificial intelligence in healthcare: a multidisciplinary perspective. *BMC medical informatics and decision making*, 20:1–9, 2020.

[Ale22] Aleixnieto. Understanding lime | explainable ai, 2022.

[BBM+15] 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):e0130140, 2015.

[BCL23] Christopher Burger, Lingwei Chen, and Thai Le. Are your explanations reliable? investigating the stability of lime in explaining textual classification models via adversarial perturbation. *arXiv preprint arXiv:2305.12351*, 2023.

[BKG23] Housam KB Babiker, Mi-Young Kim, and Randy Goebel. From intermediate representations to explanations: Exploring hierarchical structures in nlp. In *ECAI 2023*, pages 157–164. IOS Press, 2023.

[Cam22] Data Base Camp. What is the naive bayes algorithm?, Apr 2022.

[CN08] Yee Seng Chan and Hwee Tou Ng. Maxsim: A maximum similarity metric for machine translation evaluation. In *Proceedings of ACL-08: HLT*, pages 55–62, 2008.

[CRB19] Danilo Croce, Daniele Rossini, and Roberto Basili. Auditing deep learning processes through kernel-based explanatory models. 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 4037–4046, 2019.

[CS95] Mark Craven and Jude Shavlik. Extracting tree-structured representations of trained networks. In D. Touretzky, M.C. Mozer, and M. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8. MIT Press, 1995.

[CSWJ18] Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan. Learning to explain: An information-theoretic perspective on model interpretation. 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 883–892. PMLR, 10–15 Jul 2018.

[CYyK$^+$18] Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Universal sentence encoder, 2018.

[CZX$^+$22] Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 2778–2788, New York, NY, USA, 2022. Association for Computing Machinery.

[DJR$^+$20] 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, 2020.

[DMM08] Marie-Catherine De Marneffe and Christopher D Manning. Stanford typed dependencies manual. Technical report, Technical report, Stanford University, 2008.

[DQA$^+$20] Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. A survey of the state of explainable AI for natural language processing. In Kam-Fai Wong, Kevin Knight, and Hua Wu, editors, *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 447–459, Suzhou, China, December 2020. Association for Computational Linguistics.

[Ern18] Ernst. Explaining text classification predictions with lime, Dec 2018.

[EZMMA22] Julia El Zini, Mohamad Mansour, Basel Mousi, and Mariette Awad. On the evaluation of the plausibility and faithfulness of sentiment analysis explanations. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 338–349. Springer, 2022.

[GCH$^+$20] ZhiQiang Geng, GuoFei Chen, YongMing Han, Gang Lu, and Fang Li. Semantic relation extraction using sequential and tree-structured lstm with attention. *Information Sciences*, 509:183–192, 2020.

[GMR+18] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM Comput. Surv.*, 51(5), aug 2018.

[Gro20] Stanford NLP Group. Part-of-speech morphological features, 2020.

[GZL19] Zhijiang Guo, Yan Zhang, and Wei Lu. Attention guided graph convolutional networks for relation extraction. *CoRR*, abs/1906.07510, 2019.

[HEKK19] Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks, 2019.

[HKK+10] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[HQS+23] Muhammad Usman Hadi, Rizwan Qureshi, Abbas Shah, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, Seyedali Mirjalili, et al. A survey on large language models: Applications, challenges, limitations, and practical usage. *Authorea Preprints*, 2023.

[HRG23] Computer Science Department Stanford University Hazy Research Group, InfoLab. Relation extraction. http://deepdive.stanford.edu/relation_extraction, 2023. Accessed: 22.12.2023.

[HSM+20] Andreas Holzinger, Anna Saranti, Christoph Molnar, Przemyslaw Biecek, and Wojciech Samek. Explainable ai methods-a brief overview. In *International Workshop on Extending Explainable AI Beyond Deep Models and Classifiers*, pages 13–38. Springer, 2020.

[IBM] IBM. What is natural language processing? https://www.ibm.com/topics/natural-language-processing. Accessed: 20.05.23.

[JG20] Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness?, 2020.

[Kag20] Dec 2020.

[KDI+21] Md. Rezaul Karim, Sumon Kanti Dey, Tanhim Islam, Sagor Sarker, Mehadi Hasan Menon, Kabir Hossain, Md. Azam Hossain, and Stefan Decker. Deephateexplainer: Explainable hate speech detection in under-resourced bengali language. In *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–10, 2021.

[LK17] Marc Moreno Lopez and Jugal Kalita. Deep learning applied to NLP. *CoRR*, abs/1703.03091, 2017.

[LRW$^+$18] Jin Liu, Haoliang Ren, Menglong Wu, Jin Wang, and Hye-jin Kim. Multiple relations extraction among multiple entities in unstructured text. *Soft Computing*, 22:4295–4305, 2018.

[Mad21] Tambiama André Madiega. Artificial intelligence act. *European Parliament: European Parliamentary Research Service*, 2021.

[Mat19] Sherin Mary Mathews. Explainable artificial intelligence applications in nlp, biomedical, and malware classification: a literature review. In *Intelligent Computing: Proceedings of the 2019 Computing Conference, Volume 2*, pages 1269–1292. Springer, 2019.

[MF16] Brent Daniel Mittelstadt and Luciano Floridi. The ethics of big data: current and foreseeable issues in biomedical contexts. *The ethics of biomedical big data*, pages 445–480, 2016.

[Mol22] Christoph Molnar. *Interpretable Machine Learning*. 2 edition, 2022.

[MSY$^+$21] Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. Hatexplain: A benchmark dataset for explainable hate speech detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 14867–14875, 2021.

[NGS21] Duc Hau Nguyen, Guillaume Gravier, and Pascale Sébillot. A study of the plausibility of attention between rnn encoders in natural language inference. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1623–1629. IEEE, 2021.

[NJM21] Zara Nasar, Syed Waqar Jaffry, and Muhammad Kamran Malik. Named entity recognition and relation extraction: State-of-the-art. *ACM Comput. Surv.*, 54(1), feb 2021.

[Nor19] Richard Nordquist. What are nominals in english grammar?, May 2019.

[PM18] Nicolas Papernot and Patrick McDaniel. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning, 2018.

[PPB17] Sachin Pawar, Girish K. Palshikar, and Pushpak Bhattacharyya. Relation extraction : A survey, 2017.

[PRWZ02] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[PVG+11] 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*, 12:2825–2830, 2011.

[QZZ+20] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2020.

[RR19] Avi Rosenfeld and Ariella Richardson. Explainability in human–agent systems. *Autonomous Agents and Multi-Agent Systems*, 33:673–705, 2019.

[ŘS10] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. http://is.muni.cz/publication/884893/en.

[RSG16] 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*, pages 1135–1144, 2016.

[SCNB23] Gabriele Sarti, Grzegorz Chrupała, Malvina Nissim, and Arianna Bisazza. Quantifying the plausibility of context reliance in neural machine translation, 2023.

[SHSRF19] Kacper Sokol, Alexander Hepburn, Raul Santos-Rodriguez, and Peter Flach. blimey: surrogate prediction explanations beyond lime. *arXiv preprint arXiv:1910.13016*, 2019.

[SMGBN22] Viktor Schlegel, Erick Mendez-Guzman, and Riza Batista-Navarro. Towards human-centred explainability benchmarks for text classification, 2022.

[STY17] 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.

[TH22] Nakul Tanwar and Yasha Hasija. Explainable ai; are we there yet? In *2022 IEEE Delhi Section Conference (DELCON)*, pages 1–6. IEEE, 2022.

[THZ+22] Mohammed Taleb, Alami Hamza, Mohamed Zouitni, Nabil Burmani, Said Lafkiar, and Noureddine En-Nahnahi. Detection of toxicity in social media based on natural language processing methods. In *2022 International Conference on Intelligent Systems and Computer Vision (ISCV)*, pages 1–7. IEEE, 2022.

[TSB+21] Hale M Thompson, Brihat Sharma, Sameer Bhalla, Randy Boley, Connor McCluskey, Dmitriy Dligach, Matthew M Churpek, Niranjan S Karnik, and Majid Afshar. Bias and fairness assessment of a natural language processing opioid misuse classifier: detection and mitigation of electronic health record data disadvantages across racial subgroups. *Journal of the American Medical Informatics Association*, 28(11):2393–2403, 2021.

[Tur] Dr. Matt Turek. Explainable artificial intelligence (xai), https://www.darpa.mil/program/explainable-artificial-intelligence, access: 20.05.23.

[WA22] Darrell M. West and John R. Allen. How artificial intelligence is transforming the world, Mar 2022.

[WCDML16] Linlin Wang, Zhu Cao, Gerard De Melo, and Zhiyuan Liu. Relation classification via multi-level attention cnns. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1298–1307, 2016.

[WFBG18] Eric Wallace, Shi Feng, and Jordan Boyd-Graber. Interpreting neural networks with nearest neighbors. In Tal Linzen, Grzegorz Chrupała, and Afra Alishahi, editors, *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 136–144, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[WMZ10] William Webber, Alistair Moffat, and Justin Zobel. A similarity measure for indefinite rankings. *ACM Trans. Inf. Syst.*, 28(4), nov 2010.

[WWRM+18] Yanshan Wang, Liwei Wang, Majid Rastegar-Mojarad, Sungrim Moon, Feichen Shen, Naveed Afzal, Sijia Liu, Yuqun Zeng, Saeed Mehrabi, Sunghwan Sohn, and Hongfang Liu. Clinical information extraction applications: A literature review. *Journal of Biomedical Informatics*, 77:34–49, 2018.

[YL10] Xiaofeng Yu and Wai Lam. Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, page 1399–1407, USA, 2010. Association for Computational Linguistics.

[YXHD23] Zongbao Yang, Yinxin Xu, Jinlong Hu, and Shoubin Dong. Generating knowledge aware explanation for natural language inference. *Information Processing Management*, 60(2):103245, 2023.

[ZA22] Julia El Zini and Mariette Awad. On the explainability of natural language processing deep models. *ACM Computing Surveys*, 55(5):1–31, 2022.

[ZCL17] Qianqian Zhang, Mengdong Chen, and Lianzhong Liu. A review on entity relation extraction. In *2017 Second International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*, pages 178–183, 2017.

[ZGW^{+}19] Tao Zheng, Yimei Gao, Fei Wang, Chenhao Fan, Xingzhi Fu, Mei Li, Ya Zhang, Shaodian Zhang, and Handong Ma. Detection of medical text semantic similarity based on convolutional neural network. *BMC medical informatics and decision making*, 19:1–11, 2019.