Interpretability in sequence tagging models for Named Entity Recognition

by

SOFIA HERRERO VILLARROYA

11403861

August 14, 2018

36 ECTS
Feb-Aug 2018

Supervisor UvA: Dr. Giorgio Patrini

Supervisor Elsevier: Deep Kayal

Assessor: Dr. Zeynep Akata
Acknowledgements

I would like to thank my daily supervisor Giorgio Patrini and my supervisor at Elsevier Deep Kayal for their involvement in the project. Throughout the thesis, both of you have always made time to give me valuable advice and feedback. I have learned a lot from both of you and you have been of great support and help for this project.

I would also like to thank my friends and family for their unconditional support and inspiring optimism. Your advice and the time you dedicated to me helped me enormously throughout this thesis.
Abstract

The field of Explainable Artificial Intelligence has taken steps towards increasing transparency in the decision-making process of machine learning models for classification tasks. Understanding the reasons behind the predictions of models increases our trust in them and lowers the risks of using them. In an effort to extend this to other tasks apart from classification, this thesis explores the interpretability aspect for sequence tagging models for the task of Named Entity Recognition (NER). This work proposes two approaches for adapting LIME, an interpretation method for classification, to sequence tagging and NER. The first approach is a direct adaptation of LIME to the task, while the second includes adaptations following the idea that entities are conceived as a group of words and we would like one explanation for the whole entity. Given the challenges in the evaluation of the interpretation method, this work proposes an extensive evaluation from different angles. It includes a quantitative analysis using the AOPC metric; a qualitative analysis that studies the explanations at instance and dataset levels as well as the semantic structure of the embeddings and the explanations; and a human evaluation to validate the model’s behaviour. The evaluation has discovered patterns and characteristics to take into account when explaining NER models.
Chapter 1

Introduction

The field of Artificial Intelligence has taken enormous steps in the past years and is becoming ubiquitous in the decision-making process of numerous tasks, ranging from applications in our daily lives to more complex tasks. Typical machine learning algorithms that have been used for years could start being replaced by the new deep learning algorithms in common tasks such as recommendation algorithms [1] or malware detection [2]. For more complex tasks, such as image captioning [3] or machine translation [4], deep learning models have shown an impressive performance, making these tasks possible in industry and research. The main reason has been the availability of new resources such as larger data sets, more flexible frameworks and more powerful hardware, which has led to machine learning and deep learning being the crucial component in new applications.

These models in the area of Deep Learning have excelled in a number of complex applications for which they are able to extract hidden representations and patterns and improve prediction accuracy. They are characterized by the complexity of their structure which can include several non-linear layers stacked with hundreds of hidden units. Due to this structure and the size, the decision process that the model follows cannot be fully explained and the models are often used as black boxes. This directly compromises the interpretability of the model which, in turn, impacts its trustworthiness. The decision-making process in humans is backed up by our ability to explain the rationale behind it. We trust a decision more if the decision maker can explain the reasons behind it. For instance, we trust more a doctor’s prescription when she or he explains the reasons behind it.

However, the compromise of interpretability for complexity in the new deep learning models (and in the vast majority of the machine learning models) to obtain a higher accuracy is not always affordable. In domains where the consequences of the prediction can be catastrophic, such as the medical or the legal domain, applying models as black boxes is not an option since the model’s decision cannot be fully trusted. In a similar way as the human rationale, the line of research in Explainable Artificial Intelligence
and interpretable Machine Learning seeks to have transparent models whose behavior can be understandable and explainable and thus, trustworthy, in order for humans to use them. The efforts in this area have been mainly focused on explaining predictions for classification models. Therefore, the aim of this thesis is to broaden the interpretability area by exploring it in models for a different type of task, sequence tagging, instead of classification.

Sequence tagging or sequence labeling is a very common task in applications for Natural Language Processing (NLP) and Information Extraction (IE) such as Part-of-Speech tagging or Named Entity Recognition. The input for sequence tagging models is a sequence of tokens and the output is a sequence of labels, one per token. The task of Named Entity Recognition (NER) consists in finding entities in text data. These entities are proper nouns of places, organizations, people or any other category. It is an important task in NLP and IE since it is usually the first step of a pipeline for more complex tasks such as relation extraction, triplet extraction, assertion classification, summarization and any other application that builds on the entities recognized. To ensure the whole pipeline is explainable the very first step should be explainable. Therefore, this thesis explores, for the first time, the application of explainable AI techniques to sequence tagging models in the context of Named Entity Recognition. An attempt is made to adapt the interpretation methods made for classifiers to this task to determine the viability and the directions for interpretability in sequence tagging models.

1.1 Research questions

The field of Explainable AI has made progress towards the development of interpretation techniques for classifiers. This includes the family of attribution interpretation methods. The interpretation or explanation given by these methods attributes a relevance score to the different features of the input according to how much they impact the prediction. The initial attempts in the field of Explainable AI included model-dependent methods and it has been in the past two years that methods are shifting towards model-agnosticism since this is a desired characteristic to make interpretation possible in any domain. Therefore, for this thesis, we have chosen to use LIME (Linear model-agnostic explanations) since it is one of the few of them that is model-agnostic and is more flexible and general than others in that category.

Named Entity Recognition (NER) involves finding entities which can be composed of several words. Ideally, the interpretation of a prediction would give the user which words the model used to predict the entity, as a group. However, it is not clear whether this would be the most desirable way or how possible it is. With the aim of exploring this aspect, several research questions were formulated and answered in this thesis.
**RQ 1:** Is it possible to obtain explanations for sequence tagging models using LIME?

Given that a sequence labeling problem can be decomposed in independent classification problems, one per token, our hypothesis is that this is possible with an interpretation method for classification tasks.

**RQ 2:** How to get explanations for entities for the task of Named Entity Recognition?

An entity is a group of words that represent a concept (location, organization, person, etc). There are two types of entity according to their structure: single-word entities and multi-words entities. Even though the sequence tagging model tags each token individually, it is interesting to get an explanation from the interpretation method for the whole entity. Is it possible to adapt LIME to this requirement? To answer this question, two approaches of LIME are compared: one with the adaptations for this requirement and one without it.

**RQ 3:** How to evaluate explanations for entities?

This question arises from the need of evaluating the previous two approaches and answering the questions: do they bring value to the interpretation of entities? Are they necessary? Which method is best for sequence tagging and NER interpretability? For this, an extensive evaluation is carried out, including quantitative, qualitative and human evaluations.

### 1.2 Outline

The remainder of the document is organised as follows: Chapter 2 presents an overview of the related work in Explainable AI and in Named Entity Recognition; Chapter 3 explains necessary concepts that are used in the research, such as LIME and a sequence tagging model; Chapter 4 describes the approaches designed to obtain explanations for NER; Chapter 5 describes the evaluation approaches; Chapter 6 presents the results from the evaluation and Chapter 7 outlines the conclusions and the future work.
Chapter 2

Related Work

Literature supports the strong necessity that exists in achieving transparency in Artificial Intelligence (AI). Efforts to contribute to explainable AI have developed interpretation techniques to inspect models, mostly for classifiers, and to understand their prediction. The most relevant ones are outlined in this section. It also includes related work for the Named Entity Recognition task, which is posed as a sequence tagging problem, with an overview from the first models to tackle this problem to the current state-of-the-art.

2.1 Explainable Artificial Intelligence

The need for trustworthy models arises in domains where the predictions of the model can have important consequences, for instance, the medical, the military or the legal domain. One of the main driving forces of explainable AI is this need to validate the model for these domains where the user needs to trust the model in order to use it. Situations in which the models exhibit high predictive accuracy do not guarantee that the decision process is correct. An example of this is depicted in [5] for two use cases where interpretability shows that the model was basing its predictions on the wrong patterns. The prediction accuracy did not show this but the verification of the model allowed the doctors to ensure that the model could be used safely in the healthcare domain. Moreover, hadn’t this been checked, in the case of a wrong decision being executed, there would be a need to assign the responsibility. This also arises in situations that we are currently witnessing, for instance, self-driving cars. The integration of AI systems will require compliance with regulation and legal aspects which can be more smoothly achieved if the models are understandable, transparent and can be validated. This point is studied in [6] and is already a reality with the European Union’s new General Data Protection Regulation. This regulation project launched in 2016 and effective in May 2018 already has clauses regarding the use of machine learning algorithms for decision-making processes that significantly affect humans. The
potential impact of this regulation is analyzed in [7] where they conclude that it will enforce transparency and fairness in the machine learning algorithms in use. Therefore, the effects of this regulation itself naturally lead to the need for explainable AI.

Another driving force for explainable AI is the fact that we cannot always comprehend what we are building and the reason for its success. This does not allow conscious replication and application in other areas and researchers often find themselves in trial and error situations, basing their decisions in previous experiments without complete solid background. Moreover, understanding the model not only allows for validation but also aids in its improvement since more informed choices can be made. Model interpretation can aid model improvement by helping the developer spotting the weaknesses of the model or the data. For instance, identifying data leakage problems, which occur when accidentally introducing signals in the training set that will not occur in the data in which the model is deployed. Optimizing only for high accuracy will probably not discover this issue since the model scores a high accuracy basing its decisions on this signal. If optimized for interpretability, this problem could be spotted and the training data, in this case, could be fixed. In [8] they expose two situations that suffer from this problem: in one of them the patient IDs were heavily correlated with the diseases; and in the other, the removal of the word pneumonia created abnormal records that the model could spot as the ones that were positive examples for the classification. In their work, the authors explain more thoroughly the situation and remark that the data leakage occurred with them being completely unaware, which they judge to be the real danger. Amongst the methods they used to detect the data leakage, none of them included model interpretability which could have shown the erroneous decision process. Model interpretability could be also used to identify data-shift problems [9] that occur when the distribution of inputs and outputs for the training data is different to the one for the test dataset (or deployment data). Validating the decision process can guarantee the correct performance of the model and, in case the performance is different when deploying the model, the data shift can be spotted since the developer can trust the model’s behavior. Finally, validating the model’s decisions can also help in deciding between models since they might have different behaviors and use different features in different ways for prediction.

2.1.1 Methods for explainable AI

Reasons as the ones exposed have been outlined in the literature [10] [11] and have led to research on model interpretation. This included techniques developed towards more transparency in the decision process of machine learning and deep learning models. Initial attempts to open the black boxes started by providing global explanations for the whole dataset jointly. For instance, feature selection methods identify the inputs that contribute the most to the model’s generalization performance. However, this does not
Chapter 2. Related Work

represent feature relevance in an instance-basis and might not be faithful to the real decision process [12]. In the case of image data, an approach would be to identify the salient features that correspond to a certain class. These are only global explanations and they could be meaningless for individual explanations. As illustrated in [13], the salient features for the class bicycle could be the handlebar and the wheel. It could be the case that in some images one of these is not visible. In this case, the salient features cannot explain the model’s prediction.

Following attempts, mainly initiated in the Computer Vision domain, have focused on visualizing the individual components of the complex models. Successful approaches included visualizing the weights, activations and features [14][15]. Due to the graphical nature of Computer Vision tasks, the model’s behavior can easily be visualized, as demonstrated in the literature. This work inspired similar work for Natural Language Processing (NLP) as in [16] or in [17] where authors built saliency maps for hidden units in recurrent neural networks models for sentiment analysis and auto-encoding tasks. These techniques illustrate what the network is doing and, then, domain-experts of the task could derive conclusions from it. However, these techniques do not determine the reasons behind the predictions per se since individual neurons are not interpretable. Moreover, there is no indication on how or which neurons will activate, leaving the task as a brute-force exploration. This can become inefficient with the size of the model, the dimensionality of the hidden space or the number of layers. Finally, other visualization techniques are possible with dimensionality reduction techniques such as t-SNE [18] or PCA [19]. Their use is mainly for visualization of embeddings and has become a powerful tool in machine learning.

2.1.1.1 Interpretable models

Nonetheless, there are classification models that are inherently interpretable since it is possible to inspect the reasoning path in a meaningful way. Models such as decision trees and decision rules provide a path (set of nodes) or a set of rules as a justification of the prediction. Models such as sparse linear models provide specific weights for the different parts of the input which help to understand the influence of them in the prediction. The interpretability of some of these models and others is discussed in [20]. However, the simplicity of these models prevents them from achieving high predictive accuracy, proving them unsuitable for new advanced tasks in which deep learning models excel. Also, in the wave of the new deep learning models, the attention mechanism [21] has been developed and it is conceptually interpretable since it assigns attention weights to the input. However, the specificity of the architecture does not make it task-independent. In [22] the authors employ the attention mechanism to determine the important words in the sentences but illustrating the results meaningfully becomes a difficult task due to the complexity of the data (it is text data and therefore they have thousands of time-
steps with tens- to hundreds-of-thousands of tokens). Their solution was to lower the complexity of the model and re-train it for a simpler task. This represents the trade-off mentioned previously between interpretability and model complexity.

2.1.1.2 Attribution methods for non-interpretable models

As opposed to inherently interpretable models, researchers are focusing in the past few years in local explanations aiming towards model-agnostic techniques, in an effort to provide explanations for any type of models. The family of **attribution methods** seeks to explain the relationships between neurons instead of just explaining what the network is doing, as feature visualization does. More precisely, these methods quantify the relationship between the inputs and the predictions by attributing to each input an attribution value, that is, a relevance or contribution score. The best explanation for a model’s prediction can be the model itself, which can be interpretable (e.g. decision trees) or not (e.g. deep learning models). For the latter, it is required an explainable approximation of the model. According to how this is achieved, methods can be divided into two categories: backpropagation-based methods and perturbation-based methods.

Backpropagation-based or gradient-based methods were earlier developed and consist on interpreting the gradients of neural networks. There are several variants of this idea that led to different methods. One of the issues with these methods is that the activation functions used in the neural networks, such as **Rectified Linear Units (ReLU)**, **sigmoid** or **tanh**, have zero gradients when they do not activate, for the ReLU, or when saturated, for the other two functions. This is lost information since the gradient is zero.

**Gradient with respect to the input** is a method to build salience or attribution maps that can be sharpened by multiplying the gradient of the output (with respect to the input) with the input. In [23], the authors showed this together with the similarity of this approach with deconvolution [24], another gradient-based method that is model-dependent (for Convolutional Neural Networks (CNN)).

**Integrated gradients** [25] addresses the zero-gradient issue described above by taking the derivatives over scaled values of the input and computing the average gradient. The original input is scaled in the interval between a baseline and itself. This baseline is set by the user, usually, to zero. An important characteristic of this method is that the scores assigned add up to the output of the prediction function \( f \) for the input minus the output of the prediction function \( f \) for the baseline. This property, known as **Completeness** or **Summation to Delta** [26], is also found in other methods described below.

**Layer-wise relevance propagation (LRP)**, proposed by [10], explains the classifier’s decisions by decomposition. Mathematically, it redistributes backward the output of the prediction function \( f \) for a given class using local redistribution rules until it assigns a relevance score \( R_i \) to each input variable (e.g., image pixel, word).
Chapter 2. Related Work

This relevance score can be positive, indicating it supports the prediction, or negative, indicating it is against the prediction, similar to the direction of the gradient. The redistribution process is performed via backward propagation satisfying the relevance conservation principle: the sum of all relevance scores per layer equals the prediction score for the class being considered. It was introduced for pixel-wise explanations in [27]. The redistribution rules need to be developed according to the connections each neural network has, so they are not readily available.

DeepLIFT [26] proceeds in the same fashion as LRP but includes the baseline concept of Integrated Gradients. Instead of redistributing backward the output of the prediction function $f$, it redistributes the difference of it with a baseline. Therefore, it has been designed to satisfy Completeness.

There are also methods which only consider positive attributions instead of both positive and negative, such as saliency maps [23] or Deep Taylor decomposition [28]. It is important to note that backpropagation-based methods are not model-agnostic since only models based on the backpropagation method can be used, by definition. Some of them are even more model-specific like Grad-CAM [29], deconvolutional networks and Guided Backpropagation [30]. Moreover, [31] recently proved the conditions under which the backpropagation-based methods explained above are equivalent and how they show correlations between them.

Perturbation-based methods compute the attribution score by perturbing the example to generate close by examples and recording the variation in the model’s prediction for these. The perturbation takes place in the locality of the example to approximate the model locally. It is hard to do so globally since simpler models that are interpretable are usually insufficient to accurately approximate globally the behavior of complex models. An advantage of the methods in this category is that they are model-agnostic.

LIME [32], which stands for Local Interpretatable Model-agnostic Explanations, provides explanations for individual model predictions by locally approximating the model around the prediction given. The locality is defined by a proximity measure and the model is approximated with a linear model that ensures the interpretability.

Anchors is a further extension to LIME by the same authors [33]. They argue that, given that the locality used in the explanations is instance-based, it is unclear whether the explanations could apply to unseen instances. They call coverage the region to which the explanations apply. The Anchors method is focused on this coverage and provides, as an IF-THEN rule, the elements in the input (anchors) that are required so that the explanation applies. For unseen instances, if they meet the conditions specified by the anchors, then the same explanations apply. A possible issue that could arise is the over-specification of the anchor. If this is too specific to the example then it narrows the number of examples that fit this anchor. This can be the case for complex domains with a high number of features. In this case, the author recommends the LIME
2.2 Named entity recognition

Named Entity Recognition (NER) is a task from Natural Language Processing (NLP) and Information Extraction (IE) consisting of identifying Named Entities (NE). These are proper names of locations, people, organizations and other, which usually convey the most important information in the document. NER is a classic task in NLP that has been extensively researched. It is usually the first step to other NLP and IE tasks, such as relation extraction, and it enhances numerous applications. For example, the identification of entities can help a search engine when indexing relevant websites or recommendation algorithms when indexing recommendations. In general, given that entities contain a great amount of information, identifying them eases the analysis of pieces of text (such as reviews, research papers, legal documents, etc) and enables quicker classification and understanding of them.

NER is usually posed as a sequence labeling or sequence tagging problem where each token in the sequence, for instance, each word in a sentence, is tagged to indicate whether it is part of an NE together with the type of NE. The first approaches to this task were rule-based techniques and statistical models such as Hidden Markov Models, Maximum Entropy or Conditional Random Fields (CRF) or regular classification algorithms such as Support Vector Machines. These methods heavily relied on feature engineering to obtain high performance. Features could range from spelling ones (capital letters, numbers, punctuation etc.), to morphological and syntactic ones (part-of-speech
Chapter 2. Related Work

(POS) tags) and context ones (n-grams). In addition, methods could also use knowledge bases to match words and apply rules.

The hand-crafted engineering is task-dependent and empirical, requiring an analysis for each new NLP task one faces. Therefore, in an effort to reduce the hand-crafted engineering, word-embedding-based methods were introduced by [35]. The model proposed trained embeddings to learn word representations and then used a multi-layer neural network architecture with a window approach to read the context. A drawback of this approach is that it limits the context to the window size and does not capture larger dependencies in the context. The success of LSTMs in speech recognition [36] motivated the authors of [37] to propose a different architecture for this task. They switched the model to a Bidirectional LSTM model which is more powerful than the multi-layer neural network. They also addressed the issue of capturing character-level information since this was captured by the hand-crafted features but not by the word embeddings used by [35]. They created character-level embeddings with a Convolutional Neural Network (CNN). Following this line, similar models were proposed and they currently match the state-of-the-art for statistical models without using external knowledge or hand-crafted features. Some of them replaced the CNN for another Bidirectional-LSTM [38] and included a CRF layer as the last step of the pipeline to include context in the final tagging [39].
Chapter 3

Preliminaries

This section discusses in detail the techniques used in this research. It includes a description of the interpretation method and the sequence tagging model chosen.

3.1 LIME

LIME (Local Interpretatable Model-agnostic Explanations) is an interpretation method for classifiers introduced in [32]. The main goal is to provide an explanation for the prediction of the classifier that is interpretable. More precisely, the author defines an explanation as a *textual or visual artifact that provides qualitative understanding of the relationship between the instance’s components (e.g. words in a text, patches in an image) and the model’s prediction*. Therefore, the explanation must be interpretable and understandable by the user while representing the model’s behavior. LIME explanations are local since they are obtained for individual model predictions. To compute an explanation, LIME approximates the original model locally around the example given with the aim of faithfully representing the model’s behavior in the neighborhood of this example. A global approximation, instead of a local, would imply global and local fidelity. However, this is still a challenge for complex models. A global interpretation could be non- interpretable due to the complexity of the model or could be very simple to capture the behavior of the model [40]. In addition, global explanations tend to represent average behavior, which can lead to explanations of limited value for individual examples.

From a formal point of view, the explanation is given by an explainer which is an interpretable model \( g \in G \) from the set \( G \) of potentially interpretable models (inherently interpretable models). In order for the explanations to be interpretable, the input features of an example must use an interpretable representation even though the model internally uses another representation. For instance, for text classification, the interpretable representation can be a binary vector indicating the presence or absence of each word. The model, however, can then represent each word with word embeddings.
or other representation. Using this interpretable representation, for a given example $x \in \mathbb{R}^d$, then a binary vector indicating the presence of a word is denoted as $x' \in \mathbb{R}^{d'}$ and the domain of $g$ would be $\{0,1\}^{d'}$. Also, let us have $\Omega(g)$ as a measure of complexity of the explanation model since not all of them will be simple enough to interpret (trade-off between complexity and interpretability). In the case of LIME, the interpretable model used for the explanations is a sparse linear model and the complexity of it would be the number of nonzero weights.

Given a model $f : \mathbb{R}^d \to \mathbb{R}$ to interpret, the neighbourhood for a data example $x$ is defined by $\pi_x(z)$ which specifies the distance between instances $x$ and $z$ according to a proximity measure. In practice, LIME uses an exponential kernel defined on the cosine distance function for text input and on the L2 distance for image input.

When approximating $f$ with $g$ in the locality defined by $\pi_x$, let $\mathcal{L}(f,g,\pi_x)$ be a measure of how inaccurate and non-representative - or as the authors say, unfaithful - the approximation is. Then, to guarantee interpretability and local fidelity this measure $\mathcal{L}(f,g,\pi_x)$ must be minimized while having $\Omega(g)$ simple enough to obtain explanations that can be interpreted by humans. Therefore, the optimization process to learn the explanation is ruled by the locality-aware loss:

$$\epsilon(x) = \arg \min_{g \in \mathcal{G}} \mathcal{L}(f,g,\pi_x) + \Omega(g)$$ (3.1)

$\mathcal{L}(f,g,\pi_x)$ is approximated by drawing examples in the locality of the example $x$ weighting them by $\pi_x$. A sampled example is obtained by drawing nonzero elements of $x'$ (the binary vector that is the interpretable representation) uniformly at random to obtain a perturbed example $z' \in \mathbb{R}^{d'}$ (also in the interpretable representation). This perturbed example is labeled with $f(z)$, obtained by passing to the original model $z \in \mathbb{R}^d$, the true representation of $z'$. These perturbed instances create a new dataset $Z$ used to approximate Equation 3.1 and get the explanations $\epsilon(x)$. LIME uses a locally weighted square loss for $\mathcal{L}$ as:

$$\mathcal{L}(f,g,\pi_x) = \sum_{z,z' \in Z} \pi_x(z)(f(z) - g(z'))^2$$ (3.2)

In practice, LIME defines $\mathcal{G}$ as the class of linear models such that $g(z') = w_g \cdot z'$. For text classification, they use a bag of words as the interpretable representation and set a limit $K$ on the number of words so that it remains interpretable for the user. Therefore, the complexity is defined by $\Omega(g) = \infty 1[||w_g||_0 > K]$. Equation 3.1 becomes intractable with this $\Omega(g)$. To approximate it, they first run LASSO, which consists on minimizing an L1 loss of the feature weights, and select the top-K features to truncate the model. Then, the wights are learned with least squares.
3.2 Bidirectional LSTM for sequence tagging

In the current state-of-the-art for sequence tagging, one of the models used to process text sequences is the Bidirectional LSTM (BLSTM) [41]. Its success is mainly due to the fact that LSTM (Long Short-term memory) units are able to capture the long-term dependencies present in text data. In addition, the bidirectional pass captures these dependencies from both sides and enhances the performance.

The LSTM unit was introduced by [42] as a different type of unit for Recurrent Neural Networks (RNN). An RNN is a type of network to process sequences where the current output depends on the previous output for every element in the sequence. Figure 3.1 depicts a layer of an RNN where $x$ is the input, $h$ is the hidden state and $o$ is the output. The parameters to optimize are the matrices $U, W, V$. Each element $t$ of the sequence is passed through this model where the network stores information from previous inputs in the hidden state $h$, given by:

$$h_t = \sigma(U \cdot x_t + W \cdot h_{t-1}) \tag{3.3}$$

where $\sigma$ is a non-linear activation such as the $\text{sigmoid}$ or the $\text{tanh}$. Then, $h$ is used to calculate the output $o_t = \text{softmax}(V \cdot h_t)$.

Figure 3.1: Recurrent Neural Network model.

The first RNNs presented one main issue: they could not capture long-term dependencies due to the vanishing gradient problem, as demonstrated in [43]. LSTM units were proposed as a solution to this problem. They essentially differ on how they compute the hidden state but the architecture is the same as for an RNN. Figure 3.2 depicts the mechanism of an LSTM unit. The unit is called cell, $C$, and it decides which information to keep from the previous state and which one to pass for the next. It uses a system of gates as follows:

- Forget gate $f_t$: uses a sigmoid activation to decide how much information to keep from the previous cell,

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f). \tag{3.4}$$
• Input gate $i_t$: uses a sigmoid activation to decide how much information to keep from the candidate cell state (computed in the candidate $C'_t$),

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i).$$ \hspace{1cm} (3.5)

• Candidate $C'_t$: it is the new candidate values for the new cell state for step $t$,

$$C'_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C).$$ \hspace{1cm} (3.6)

• New cell state $C_t$: the final new cell state $C_t$ for the current step $t$ is determined by how much information the forget gate keeps from the previous cell and how much information the input cell adds from the candidate values,

$$C_t = f_t \ast C_{t-1} + i_t \ast C'_t.$$ \hspace{1cm} (3.7)

• Output gate $o_t$: controls the output by deciding how much information of the new cell state is used,

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o),$$ \hspace{1cm} (3.8)

$$h_t = o_t \ast \tanh(C_t).$$ \hspace{1cm} (3.9)

Figure 3.2: Long short-term memory unit for a RNN.

The BLSTM model is used to capture forward and backward text dependencies. It is composed of a forward LSTM unit and a backward LSTM unit whose outputs are concatenated. The forward and backward LSTMs can have several layers of LSTMs units that would be stacked. In this project, only one layer was used. The concatenated output is then passed through a feed-forward layer and then through a softmax function to obtain the probabilities per class for each element in the sequence.
3.3 Sequence tagging for Named Entity Recognition

Named Entity Recognition identifies the entities (proper nouns) and their type in text. The approach to this task consists in using a sequence tagging model whose prediction indicates where those entities are by assigning a label to each word in the example that indicates whether it is part of the entity or not. Entities can be classified in single-word entities, when the entity consist on one word, e.g: Australia, and multi-word entities, when the entity consist on several words, e.g.: New York. Therefore, it is necessary to indicate the boundaries in order to differentiate them in text and know where the entity starts and finishes, if it is multi-word. For this purpose, the words in the sequence are tagged following the BIO tagging scheme introduced in [44]. This scheme uses the label B to indicate that the word is the beginning of the entity; label I to indicate that the word is inside an entity; and the label O to indicate that the word is outside an entity. In addition, the entity type is indicated appended to the label as B-TYPE and I-TYPE. Figure 3.3 illustrates the labelling schema. There exist other schemes, such as the BIOES which also indicates with the label E that the word ends the entity and with the label S that the entity is single-word. However, even though some researchers have compared both schemes for certain fields, for instance in Biomedical NER [45] and [46], the BIO scheme is widely used and the most common, as well as the one use in worldwide competitions for Named Entity Recognition as the one hosted by the Conference on Computational Natural Language Learning (CoNLL) in the years 2002 [47] and 2003 [48], which continues to be the reference for NER.

Figure 3.3: Example of a labelled sentence using the BIO tagging scheme. The entity is New York City of type location (LOC).

```
The president went to New York City after two months
```

```
     0  0  0  0  B-LOC I-LOC I-LOC 0  0  0
```
Chapter 4

Interpretable Named Entity Recognition

As previously discussed, the aim of this research is to determine whether current model-agnostic interpretation methods for classification tasks, more precisely LIME, can be applied to tasks modeled as sequence tagging problems, more precisely, Named Entity Recognition. Firstly, sequence tagging can be seen as a set of individual classification tasks -which may or may not be independent depending on the task. Therefore, the intuition is that it is possible to apply LIME to it. Answering RQ 1, Section 4.1 explains this idea. Secondly, Named Entity Recognition consist of identifying the entities in a given text. For this, let us have a model that receives as input the words of the text. In this situation, an explanation with an attribution interpretation method will return for every word a relevance score indicating how (positively or negatively) relevant it is to identify the entity. In other situations, the model could have had a different input space and receive, for instance, boolean features describing each word. However, this situation requires a feature engineering process to determine the features. With the aim of training models that do not rely on extra steps, the first situation is preferred and thus we assume that the input is the raw text data. With this, to answer RQ 2, two approaches to interpreting NER predictions are described in Section 4.2 from a high-level perspective. They assume to have a model able to identify entities with a reasonable accuracy. The model returns the probabilities for each word to be part of an entity (B and I labels in the BIO labeling scheme, as described in 3.3) or not (O label in the BIO labeling scheme). The final prediction for the word is given by the label with the highest probability of all three. Finally, Section 4.3 summarizes the approaches taken.
4.1 LIME for sequence tagging models

The main difference between classification models and sequence tagging models is the label they have to predict. For the former, there is one label for one example, regardless of the format of the example (image, text, numbers, etc). However, for the latter this is a structured prediction and, in the case of NER, it is a sequence of labels. The main issue when explaining structured predictions is its heterogeneity. Firstly, the size or structure might be different for every example. Secondly, different parts of the structure might convey different information which might not be relevant for the explanation. Deciding on which parts of the prediction to explain is also part of the explanation process.

Current interpretation methods for classification tasks receive the model’s prediction as input. For instance, LRP for Neural Networks backpropagates the relevance starting from the neuron in the last layer that had the highest activation, which indicates the class predicted. LIME requires a vector of probabilities (one per class) per example that indicates how likely it is for the model to classify the entire example as each class. Therefore, these expect a model’s prediction corresponding to the requirements of a classification task, one label per example, instead of a structured prediction. However, it is possible to use LIME with structured output for text input as illustrated in [32]. In the case of sequence labeling, each token in the input sequence will have a corresponding label in the output sequence. The idea is to estimate the model around one element of the sequence by passing to LIME the model’s probabilities of just that element. This implies recording the changes in the probability of that element of the sequence when perturbing the example. This is, essentially, considering the problem as a classification per element and getting an explanation for every element in the sequence, which in this case are words (also referred to as tokens). Therefore, a sentence would have as many per-words explanations as words and one could decide which ones are relevant depending on the task.

4.2 LIME for Named Entity Recognition

When explaining the predictions of a Named Entity Recognition system we are interested in explaining the entities predicted and why they were predicted. Therefore, when using LIME, the predicted entities are the elements that are chosen from the sequence to explain. Two situations arise then: explaining single-word entities and explaining multi-word entities.

In the case of single-word entities, the explanation is straightforward since the entity is composed of one word. LIME will approximate the interpretable linear model around the changes in the probabilities for the word that is the entity. In the case of multi-word entities, the situation is different. The entity is a set of words that represent the same concept. One can imagine that, ideally, the explanation should be for the whole
entity as a group and LIME would estimate one model that represents the behavior of the complete entity, instead of retrieving a separate explanation per word in the entity. To introduce this concept of entity in LIME, two adaptations are proposed.

First, for LIME to estimate one model for the entity it should receive the changes in the probabilities such that they represent the entity altogether. Let us have a sequence tagging model \( f \) for Named Entity Recognition. This model maps an input sequence of length \( L \), e.g. a list of words (a sentence, a paragraph, a document), to an output sequence of length \( L \), e.g. a list of the predicted probabilities of assigning each possible tag \( t \in T \) to an element, for each element \( l \) in the input sequence. For every element \( l \), the predicted probability is a vector \( p_l \in \mathbb{R}^{|T|} \). To illustrate it better, let the model be able to recognize entities of type X and type Y. Then \( T = \{ O, B-X, B-Y, I-Y, I-Y \} \), meaning that a word can be classified into five possible classes. Then, the probability \( p_l^O \) corresponds to the probability of assigning tag O to the word at position \( l \), \( w_l \); the probability \( p_l^{B-X} \) corresponds to the probability of assigning tag B-X to \( w_l \), etc. In the case of a multi-word entity, there is a probability vector \( p \) for every word in the entity. With the aim of representing the entity as a whole, we aggregate the probabilities of the words to obtain a single probability vector. This probability vector would represent the entity at an entity-level and not at a word-level. Therefore, it consists of the probability of the entity being tagged as the possible classes, which would be as a non-entity, as an entity of type X or as an entity of type Y. Thus, \( T' = \{ O, X, Y \} \). To illustrate it better, let us have an input sequence with a multi-word entity \( e \) of length \( |e| = 3 \), that is, three words. Let these words be at positions 1, 2 and 3 in the input sequence, so \( w_1, w_2 \) and \( w_3 \). From the output sequence, let us consider the corresponding probability vectors \( p_1, p_2 \) and \( p_3 \). Then, the probability vector \( p_e \in \mathbb{R}^{|T'|} \) is computed by aggregating the probabilities as follows:

- The probability of the entity of **not being an entity**, denoted as \( p_e^O \), is computed as the probability that every word has of getting assigned the class O averaged over the words in the entity,

\[
p_e^O = \frac{1}{|I|} \sum_{i \in I} p_i^O,
\]

where \( I = \{1, 2, 3\} \) are the positions in the input sequence of the words that belong to an entity.

- The probability of the entity **being of type X** takes into account the following: for the entity to be correctly predicted as an entity of type X the first word should be predicted as B-X and the rest as I-X. That is, for the first word in the entity, the highest probability should be the one for B-X, while for the other words in the entity it should be I-X. Therefore, we only consider the probabilities that determine whether it was predicted as an entity of type X: \( p_e^X \) is the average of
4.3 Approaches

Considering LIME for sequence tagging and the adaptations explained, there are two approaches to get explanations for entities in Named Entity Recognition tasks.

The first one consist on applying LIME-as-is to the sequences with entities regarding the words in an entity separately. This would mean obtaining explanations for each word in the entity, as explained in section 4.1, estimating a model per word and returning an explanation per word in the entity. If we were given several explanations to explain one entity, one would compare them to spot the common and non-common elements to obtain a global picture. Applying this idea naively, we can obtain a joint explanation from the per-word explanations. When doing so, it could happen that a
word appears in several explanations with different relevance scores. To address this, the naive approach assumes the following:

- If the word has a positive score for all of the explanations it appears on, then the highest is kept. Same applies for negative scores where the smallest is kept. It is important to keep the magnitude of the relevance to be able to determine how influential it was for a part of the entity. Lower relevance scores, but still positives, are still indicating that it is relevant but with less influence so keeping the highest provides more information.

- If the word has positive and negative scores in the explanations it appears on, then the one with highest absolute value is kept since that effect is considered to be stronger (note that a zero relevance score means that there is no effect).

- The previous assumptions are only applied for relevance scores above a threshold that indicates when a relevance score is considered significant. If the values are lower than that threshold then we consider the effects to be so low that they represent the same. The threshold was set to 0.1 for relevance scores between 0 and 1. In cases where the relevance scores are different but below the threshold, the relevance score assigned is the one of the first explanation examined. Other options would be to consider the relevance score of the last explanation examined, or any other.

By joining the explanations for each word in the entity we can naively obtain one explanation for the entity. However, with the adaptations explained in section 4.2, a single explanation can be obtained by introducing the concept of entity from the beginning of the pipeline. This would be the second approach, denoted as LIME-ner. It requires (1) passing to LIME the aggregated probabilities to estimate a model that captures the behavior of the model for the entity as a whole; and (2) enforcing the concept of entity when estimating the model.

Each approach returns a single explanation for the whole entity, that is, a list of words with an assigned relevance score. Table 4.1 summarises the characteristics of each. In the next section, the evaluation methodology for them is described.
Table 4.1: Characteristics of the two approaches of LIME for NER. It includes the input to the explainer LIME, the perturbations used by the explainer LIME and the output of the explainer. In the case of LIME-as-is the explainer outputs one explanation per word in the entity and then these are joined to have one for the whole entity.

<table>
<thead>
<tr>
<th></th>
<th>LIME-as-is</th>
<th>LIME-ner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>Probabilities per word in the entity</td>
<td>Probabilities for the whole entity</td>
</tr>
<tr>
<td>Perturbation</td>
<td>Breaking entities</td>
<td>Not breaking entities</td>
</tr>
<tr>
<td>Output</td>
<td>Joint explanations for the whole entity</td>
<td>One explanation for the whole entity</td>
</tr>
</tbody>
</table>
Chapter 5

Evaluation

The evaluation of interpretation methods is a challenging task [49]. No defined benchmark exists and it is usually tackled with metrics specific to the interpretation method. This is due to the lack of ground-truth for the explanations, which makes it hard to use any standard metric. Furthermore, the interpretation of the explanations is subjective. An explanation illustrates the model’s point of view which may not coincide with our point of view. Then, we have to interpret it and different people can do it in different ways depending on their expectations of the model’s behaviour. In a qualitative evaluation, correctly accounting for this might be hard and a quantitative evaluation should ideally not be affected by this. These reasons lead to open interpretations of the explanations, making it difficult to find an evaluation method that accounts for this or that measures beyond this. Moreover, it is important to remark that the quality of the explanations is limited by the quality of the model and the data. In some situations, it could be hard to distinguish whether an error comes from the interpretation method or from the classifier. However, with higher accuracy in the model, the explanations tend to become more intuitive since the patterns picked up by the model are more visible.

Related work on evaluating explanations with text input uses assumptions that seem logical about which characteristics an explanation should have. For instance, in [50] they assume that the model should not use words belonging to other documents. For a document classification task, they build a hybrid document by introducing in a document parts of other documents. The metric measures whether the explanation includes words from the other document. However, this might only work for some problems where this assumption is valid or where creating hybrid examples makes sense or is possible. This reflects the difficulty in achieving a unified evaluation metric. In [13] they assume that relevant words influence the most the prediction and that removing them will cause the highest change in the prediction. Therefore, if the interpretation method has identified the actual relevant words, then removing them should lead to the highest change in the prediction of the model. This assumption is more applicable than the previous one since it does not depend as much on the type of examples and will be
Therefore, to answer RQ 3, this chapter provides an extensive evaluation focused on text input data with a two-fold aim. On one hand, we would like to determine whether the two approaches with the adaptations are identifying the relevant words in the text sentences. On the other hand, we would like to obtain insights to characterize an explanation for sequence tagging and validate the assumptions made determining if they add any value. The results of this proposed evaluation are discussed in the next chapter.

5.1 Qualitative evaluation

A qualitative evaluation can be carried out at different levels. Firstly, at an instance-level, one can observe the explanations -the list of words with the assigned relevance scores- for individual examples and derive conclusions. LIME provides a visualization of the interpretations where the words are highlighted according to the relevance score. Secondly, at dataset-level one can analyze the most relevant words for one type of entity across all the examples. This will determine which words represent or are indicative of a type of entity for the model. These inspections can reveal the soundness of the explanations and quickly spot discrepancies and strange or biased behaviors.

Following the lines of [51], another interesting experiment is to explore the explanations together with the semantics captured by the word embeddings. Given that the explanations return for each word of the sentence a relevance score, this could be used to weight a linear combination of the word2vec word embeddings. This would represent the sentence in terms of the entity as a sentence embedding. Since the word2vec embeddings have shown to maintain the semantic relationships when manipulated using vector arithmetic [52], this experiment explores if these linear regularities are maintained when creating an embedding of the sentence using the relevance scores. To illustrate this better, let us have the example $i$ from the data set, which is a sentence $s_i = [w_1, w_2, ..., w_N]$ of length $N$ where $w_n \in \mathbb{R}^{\text{emb\_dims}}$ is the corresponding word2vec word embedding of word $w_n$. The explanation of the sentence for class $k$ returns the relevance scores, which can be expressed in a vector $r_i = [r_1, r_2, ..., r_N]$ of length $N$ where $r_n \in \mathbb{R}$. The sentence $s_i$ embedding vector $d_i \in \mathbb{R}^{\text{emb\_dims}}$ is computed as:

$$d_i = \sum_{j}^{N} r_j \cdot w_j. \quad (5.1)$$

We expect that the words that an NER model uses to predict an entity are different for each entity type. If these words are discriminant enough and the semantics of the embeddings are kept, then an exploratory analysis of the embedding space with PCA should reveal clusters given by the semantics and the type of entity. Note that the relevance scores are obtained for the predicted class so the clusters will be given by the
model’s explanation of the predicted class and not by the model’s explanation for the ground-truth class, thus, representing the behavior of the model without including the real label.

5.2 Quantitative evaluation

The common approach to quantitatively assessing the explanations of post hoc interpretation methods consist of a perturbation process that progressively removes the information from the example. This has been applied in the Computer Vision and the NLP domain in [13] [53] [51] [54] [55]. The idea is based on the assumption that relevant words will have a greater impact on the prediction probability, and thus removing them will show a higher change in it, compared to removing less relevant words. Therefore, when measuring the change in the prediction probability, accurate explanations that spot the right relevant words should show this behavior. While in other papers they use as metric the plain difference between probabilities, in [13] they formalized this idea as a metric and named it as Area over the perturbation curve (AOPC). They applied it for a classification problem in Computer Vision.

For the task explored here, we have used the AOPC metric and adapt it for NER where the interest is in the entities. Let us define an explanation $E$ as an ordered list of words in the sentence:

$$E = [w_1, w_2, ..., w_N], \quad (5.2)$$

where $N$ is the length of the sentence. The order is induced by a scoring function $s_i = S(x, f, w_i, c)$ where $x$ is the sentence, $f$ is the prediction function that returns the probabilities, $w_i$ is the word and the scoring function is with respect to class $c$ (for simplicity, this $c$ is not used in notation). The score $s_i$ indicates how important is the word $w_i$ to predict the class $c$. In this case, the scoring function is each of the approaches of LIME explained in the previous chapter, and both return a list of scores per entity. The scoring function induces an order on the list indices $I$ such that:

$$(i < j) \iff (S(x, f, w_i) > S(x, f, w_j)) \quad \forall (i, j) \in I \text{ s.t. } i \neq j. \quad (5.3)$$

In this ordering, the locations in the sentence which support the most the class $c$ will be at the beginning of $E$. The ones that are against predicting class $c$ will have a negative score and will be at the end of the list. Words with no effect have a score of zero. A most relevant first (MoRF) perturbation process is now applied to the sentence following the ordering in $E$ as follows:

$$x_{MoRF}^{(0)} = x$$

$$\forall \, 1 \geq k \geq L : \, x_{MoRF}^{(k)} = g(x_{MoRF}^{(k-1)}, i_k), \quad (5.4)$$
where \( g \) is a function that removes the information, in this case, a word, of the sentence \( x^{(k-1)}_{MoRF} \) at index \( i_k \) and \( L \) is the number of iterative removals. The perturbation chosen is to remove the word by setting its embedding to the zero embedding. Because of the ordering, the focus is mainly on the highly relevant features. Using a fixed \( g(x, i_k) \), different explanations can be compared according to the Area over the MoRF perturbation curve (AOPC):

\[
AOPC = \frac{1}{L + 1} \left(\sum_{k=0}^{L} f_{gt}(x^{(0)}_{MoRF}) - f_{gt}(x^{(k)}_{MoRF})\right)_{p(x)}, \tag{5.6}
\]

where \( \langle \cdot \rangle_{p(x)} \) indicates the average over all the examples in the data set and \( f_{gt}(x) \) is the predicted probability of the ground-truth class \( gt \). When applying the AOPC for NER, the interest is on the entities. Therefore, the average \( \langle \cdot \rangle_{p(x)} \) is over the entities in the data set and the AOPC is per entity. If a sentence has several entities, then it is perturbed for every entity and what changes is the ordering since it is given by the explanation of each entity. For multi-word entities, the final difference is the average over the differences for every word in it.

Another possible perturbation process is the least relevant first (LeRF) where the order is reversed and the least relevant words are considered first. The expression for LeRF is:

\[
x^{(0)}_{LeRF} = x \\
\forall 1 \geq k \geq L : x^{(k)}_{LeRF} = g(x^{(k-1)}_{LeRF}, i_L + 1 - k). \tag{5.8}
\]

Two experiments are conducted using the AOPC metric. The first one consists on applying the MoRF perturbation on correctly classified entities (where all tokens of the entity have been predicted correctly) using the explanation for the true class. In this case, if the relevant words (the ones with positive relevance score) are truly the relevant ones then we expect that removing them first will cause a drop in the target output (the ground-truth class). Similarly, the second one consists on applying the LeRF perturbation on incorrectly classified entities (where at least one word in the entity has not been predicted correctly). We use, as well, the explanation for the true class, thus, the words with positive relevance score support the prediction of the true class while the ones with negative are against predicting that class, so they support predicting another class. Since the prediction is wrong, then the negative ones are actually supporting that prediction of the model. In this case, removing first the negative words will change the wrong prediction to the correct one causing a rise in the target output (the ground-truth class). Calculating the AOPC for different consecutive values of \( L \) results in a curve that is analyzed according to two criteria [54]:

- The curve’s initial steepness. The steepness represents the change in the prediction probability. The stronger it is the better the interpretation method is at capturing
the important features. Since we assume that those will yield the greatest changes in the prediction, then the point in the curve when those are removed should be the steepest. For the two experiments, these are removed first, so the initial steepness is examined.

- Target output variation. Following the same line, better methods will show a target output with larger variation across $L$ since they are identifying the most discriminant features.

It is important to remark that the interpretation method (LIME) and the AOPC metric rely on the same idea: a perturbation process. They differ on the sampling of the words to perturb. For LIME this is random and for the AOPC this is given by the explanation. In addition, while LIME approximates a model by optimizing for all classes, the AOPC only focuses on the ground-truth class for the average.

5.3 Human evaluation

A human evaluation is an interesting step to compare the explanations provided by algorithms with human ones. This shows the subjective aspect of them as well as how different are the reasoning processes of models and humans.

For this study, the aim was to determine the similarity between human explanations and the model explanations. The task given to participants was the same as for the model and the explainer: first, identify the entity and then identify the words that lead to that. In practice, the question was:

*Given a sentence which contains an entity of type either GEO or PER, specify which words in the sentence, if any, could help you determine that there is one.*

The task includes 31 randomly sampled sentences for which the user has to answer the question. The complete task can be found in Appendix B.

The human responses are then evaluated by counting the overlapping words with the model’s explanation top three most relevant words. Only the three first are used since from the qualitative evaluation one could see that the relevance distribution is quite skewed usually and only a few words are truly relevant and have a significant score different than zero. This is mainly in part because the sentences are short so there are not many words to distribute the relevance across and also because of the nature of the problem of NER. This is further illustrated and explained in the next chapter with the results.
Chapter 6

Results

This chapter provides the results from the evaluation outlined in the previous chapter. Firstly, it describes the experimental setup, including the data and the model chosen. Secondly, it provides the results for the different experiments carried out to evaluate and explore the explanations.

6.1 Experimental set up

6.1.1 Data preprocessing

The data used for the experiments is the annotated corpus for Named Entity Recognition using the Groningen Meaning Bank (GMB) corpus\(^1\). This corpus has eight types of entities: Geographical (geo), Organization (org), Person (per), Geopolitical Entity (gpe), Time (tim), Artifact (art), Event (eve) and Natural Phenomenon (nat). Table 6.1 provides more information about the data set. We decided to use only two types of entities to avoid compromising in excess the performance of the model, since the more entities the harder the problem is and a worse performance would lower the quality of the explanations. Given that the proportions are not balanced, we chose to use the geo entity, since it is the most frequent one, and the per entity. Other entities were also frequent such as org or tim, but the former is similar to geo and the latter has a very specific format that identifies them.

The data is in the CoNLL format which is already tokenized and labeled per word. We applied common preprocessing for NER that includes replacing numbers by the digit 0 and leaving uppercase letters. We removed the entities that were not the chosen ones by setting their label to O. Table 6.2 shows the counts for the final data set. As it can be seen, almost all entities of the type person are composed of two or more words since the number of I-per labels is larger than the number of B-per labels. This is because they are usually composed of first name and last name.

\(^1\)Groningen Meaning Bank data, \url{http://gmb.let.rug.nl/data.php}, last visit July, 2018
Table 6.1: Count of labels for the annotated corpus.

<table>
<thead>
<tr>
<th>Entity</th>
<th>B label</th>
<th>I label</th>
</tr>
</thead>
<tbody>
<tr>
<td>'geo'</td>
<td>37644</td>
<td>7414</td>
</tr>
<tr>
<td>'org'</td>
<td>20143</td>
<td>16784</td>
</tr>
<tr>
<td>'per'</td>
<td>16990</td>
<td>17251</td>
</tr>
<tr>
<td>'gpe'</td>
<td>15870</td>
<td>198</td>
</tr>
<tr>
<td>'tim'</td>
<td>20333</td>
<td>6528</td>
</tr>
<tr>
<td>'art'</td>
<td>402</td>
<td>297</td>
</tr>
<tr>
<td>'eve'</td>
<td>308</td>
<td>253</td>
</tr>
<tr>
<td>'nat'</td>
<td>201</td>
<td>51</td>
</tr>
</tbody>
</table>

Table 6.2: Training, validation and test distribution label counts for the final data set.

<table>
<thead>
<tr>
<th>Label</th>
<th>Training</th>
<th>Validation</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>'O'</td>
<td>414619</td>
<td>103191</td>
<td>129303</td>
</tr>
<tr>
<td>Label 'B-geo'</td>
<td>24112</td>
<td>5948</td>
<td>7584</td>
</tr>
<tr>
<td>Label 'I-geo'</td>
<td>4805</td>
<td>1185</td>
<td>1424</td>
</tr>
<tr>
<td>Label 'B-per'</td>
<td>10929</td>
<td>2736</td>
<td>3325</td>
</tr>
<tr>
<td>Label 'I-per'</td>
<td>11260</td>
<td>2655</td>
<td>3336</td>
</tr>
</tbody>
</table>

The experiments were carried out over a subset of the test data because computing the explanations is very expensive. Therefore, only the entities with a prediction probability of more than 95% were used since the confidence of the model would increase the explanations quality and then the pattern picked might be more visible and determining.

### 6.1.2 NER Model

As outlined in Section 2.2, initial models for NER relied on hand-crafted features obtained through expensive processes of feature engineering. In the recent years and with the appearance of word embeddings, the feature engineering is being replaced with the aim of automatic the pipeline and eliminating this cost. Following this line, the model chosen for this project is a BLSTM with word embedding features.

Word embeddings were chosen because they mapped directly to words which can then be interpreted, as opposed to character-level embeddings, which map to characters, which is harder to interpret by a human. If the explanation shows that the character $a$ is relevant to predict certain entity it is hard for a human to interpret that since characters per se do not always have a meaning, compared to words. Different input features could not be explored further in this project. The word embeddings used were *word2vec* embeddings because these carry semantic information and have shown to perform better.
than other embeddings such as the ones based on TF-IDF. The embeddings used were pre-trained ones from [38] which were trained on the English Gigaword version 4 (with the LA Times and NY Times portions removed) with an embedding dimension of 100.

A one-layer Bidirectional LSTM was implemented with Tensorflow\(^2\) and trained by minimizing the cross-entropy loss via mini-batch stochastic gradient descent. Regularization was achieved with dropout and gradient clipping. Table 6.3 lists the hyperparameters values after optimizing them for the F1 score with Bayesian optimization [56]. The results are shown in Table 6.4 in terms of F1 score over all entities and per entity type. The F1 score is computed at entity-level where an entity is considered to be correctly predicted if all of the tokens have been correctly predicted (exact match). The precision is the proportion of entities found that is correct. The recall is the proportion of existing entities that are found. The F1 score is the harmonic mean of precision and recall. It was computed using the guidelines and script from the CoNLL conference shared task. Attribution methods should be applicable to any model that returns meaningful predictions, therefore, it was not the goal to obtain state-of-the-art performance.

Table 6.3: Optimized hyperparameters values

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch size</td>
<td>64</td>
</tr>
<tr>
<td>Number of epochs</td>
<td>15</td>
</tr>
<tr>
<td>Learning rate</td>
<td>0.01</td>
</tr>
<tr>
<td>Dropout</td>
<td>0.6</td>
</tr>
<tr>
<td>Maximum norm (gradient clipping)</td>
<td>5.0</td>
</tr>
<tr>
<td>LSTM hidden dimensions</td>
<td>128</td>
</tr>
</tbody>
</table>

Table 6.4: Results for the trained models on the test data set.

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>F1 score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall</td>
<td>0.848094</td>
<td>0.832157</td>
<td>0.84005</td>
</tr>
<tr>
<td>Geo</td>
<td>0.884233</td>
<td>0.879219</td>
<td>0.881719</td>
</tr>
<tr>
<td>Per</td>
<td>0.761935</td>
<td>0.724812</td>
<td>0.74291</td>
</tr>
</tbody>
</table>

6.1.3 LIME configuration

LIME allows setting some parameters to configure the explanations for text input. The \texttt{split_expression} parameter is used to define the split expression that LIME uses to get the input features from the input string (a sentence in this case). This was set to the

\(^2\url{https://www.tensorflow.org/}\)
space character. The bow parameter indicates whether the location of the word matters. If true, then words should be treated as a bag-of-words and each is the same everywhere in the text so that multiple occurrences of the same word are counted as one. This was set to false so that the same word is different according to the location it has in the sentence. This was required since there are words such as the which are likely to appear several times. However, the impact is different if they are close to an entity or not. In addition, modifications were made to LIME to work with entities and to include the second type of perturbations.

6.2 Qualitative results

6.2.1 Instance-level analysis

The qualitative analysis on an instance-level allows to quickly validate and understand the explanations. Figures 6.1a and 6.1b present a visualization of the explanations for LIME-as-is, estimating a model per word in the entity. Figure 6.1c is the explanation for LIME-ner where only one model has been estimated for the entity. The entity is Mark McClellan and the labels on the top show the class (BPER, IPER or PER) for which the explanation has been computed. In the text on the right, the words are highlighted according to the relevance score. The words with positive relevance are supporting the class for which the explanation is computed, so they are situated on the right side of the graph. The words with negative relevance score do not support that prediction and are situated on the left side of the graph. The same is presented for an entity of type geo in Figure 6.2.

The impact of LIME-ner can be seen in the relevance score assigned to the tokens in the entity: all of them get the same score. This is due to the perturbations that do not break the entity and therefore both tokens are equally relevant since they are considered a group. On the other hand, in the explanation for LIME-as-is, the influence of the B token in the I token is captured by the explanation showing how the fact that there is a B highly affects the prediction of an I token. This is because it is impossible to find a word with tag I without a word with tag B or I before. The model is able to learn this pattern and in this situations, the words inside the entity obtain very high relevance scores but, in general, the relevance scores for this problem tend to be between 0 and 0.5. Therefore, LIME-ner does not account for the effect that words inside an entity have on other words inside the entity since it considers the entity as one word and does not break it, mitigating this influence.

In general, the explanations from both approaches allowed to understand better the model behavior, showing that the model memorizes some of the entities. This might be due to the frequency of the entity in the training data. The explanations for frequent entities include the words in the entity with a high relevance score and almost no other
of the surrounding words. Also, when comparing individual examples of the two types of entities, it is noticeable that when predicting an entity of type geo the model relies more on the words around the entity and when predicting an entity of type per the model relies more on the tokens that belong to the entity (the entity itself). It could be because for locations there exist patterns that do not rely on the entity, since a lot of words can introduce them, e.g.: in, from, to etc. But, not as many for a person. This is seen in the examples previously discussed where both entities have a similar frequency in the training data set, Mark McClellan (11 times) and Gulf of Aden (10). For entity Mark McClellan the explanations include from context the word said (rel. score 0.07) with a low relevance score. However, for entity Gulf of Aden the explanations include other words such as across (0.15), the (0.13) or days (0.1).

6.2.2 Dataset-level analysis

The data-set level analysis included compiling the representative words per entity according to the model. To do this, we obtained the explanations for the entities for a given type (per or geo) and sort out the words across all of them by their relevance score in decreasing order. If a word appeared in different explanations with different relevance scores, the maximum relevance score is picked for the word. The idea is then to pick the top K words as representative word for that entity type.

The resulting lists of words had a lot of entities in the top positions mixed with some non-entity words. This shows that the model is also memorizing the entities to identify them by name and the most frequent ones are in the top positions. However, it is also interesting to analyze the first 10 non-entity words in the sorted list. Table 6.5 contains the list for both type of entities for LIME-as-is and Table 6.6 for LIME-ner. The resulting words for each type of entity are related to the concept of the entity, validating the model’s reasoning, to some extent. For the per entity, the model uses words related to roles. For the geo entity it uses words related to locations. Also, both approaches of LIME are able to identify the same relevant words.

Further inspecting the tables 6.5a and 6.6a for the entity per, it reinforces the idea that for entities of type per the model relies more on the words inside the entity. When this influence (the influence of entity words on other entity words) is accounted (LIME-as-is) the words in the first 150 positions are all entities while for LIME-ner there is already one non-entity word at position 65. This is not the case for the entity geo where other non-entity words are more relevant in general. The effect of considering the entities as a group for LIME-ner is visible in the relevance score. In Table 6.6 the relevance scores are in general lower than in Table 6.5. This might be because the relevance scores for LIME-ner are with respect to the whole entity. Individual effects on each token of the entity, which could be different in sign, are estimated as a whole, resulting in a compensated score.
Figure 6.1: Explanations retrieved from LIME-as-is are Figures 6.1a and 6.1b. Explanation for LIME-ner is in 6.1c. The three images have been cut to fit better the space. The complete images are available in Appendix A.

(a) Explanation for token *Mark*.

(b) Explanation for token *McClellan*.

(c) Explanation for entity *Mark McClellan*.
Figure 6.2: Explanations retrieved from LIME-as-is are Figures 6.2a, 6.2b and 6.2c. Explanation for LIME-ner is in 6.2d. The three images have been cut to fit better the space. The complete images are available in Appendix A.

(a) Explanation for token $Gulf$.

(b) Explanation for token $of$

(c) Explanation for token $Aden$

(d) Explanation for entity $Gulf$ of $Aden$
Table 6.5: Top 10 non-entity words representative of each type of entity with explanations obtained with LIME-as-is. The position is relative to the ranking including entity words and the relevance score is the largest for that word.

(a) Top 10 non-entity words for PER for LIME-as-is.

<table>
<thead>
<tr>
<th>Rank position</th>
<th>Word (rel. score)</th>
</tr>
</thead>
<tbody>
<tr>
<td>151</td>
<td>rider (0.55)</td>
</tr>
<tr>
<td>160</td>
<td>legend (0.52)</td>
</tr>
<tr>
<td>166</td>
<td>leader (0.48)</td>
</tr>
<tr>
<td>173</td>
<td>and (0.45)</td>
</tr>
<tr>
<td>181</td>
<td>adviser (0.41)</td>
</tr>
<tr>
<td>187</td>
<td>said (0.38)</td>
</tr>
<tr>
<td>188</td>
<td>says (0.37)</td>
</tr>
<tr>
<td>190</td>
<td>activist (0.37)</td>
</tr>
<tr>
<td>192</td>
<td>respects (0.37)</td>
</tr>
<tr>
<td>193</td>
<td>businessman (0.36)</td>
</tr>
</tbody>
</table>

(b) Top 10 non-entity words for GEO for LIME-as-is.

<table>
<thead>
<tr>
<th>Rank position</th>
<th>Word (rel. score)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>in (0.85)</td>
</tr>
<tr>
<td>64</td>
<td>outside (0.69)</td>
</tr>
<tr>
<td>73</td>
<td>northern (0.67)</td>
</tr>
<tr>
<td>76</td>
<td>of (0.67)</td>
</tr>
<tr>
<td>80</td>
<td>neighbouring (0.66)</td>
</tr>
<tr>
<td>81</td>
<td>to (0.66)</td>
</tr>
<tr>
<td>88</td>
<td>and (0.64)</td>
</tr>
<tr>
<td>94</td>
<td>from (0.62)</td>
</tr>
<tr>
<td>99</td>
<td>near (0.59)</td>
</tr>
<tr>
<td>112</td>
<td>restive (0.57)</td>
</tr>
</tbody>
</table>

Table 6.6: Top 10 non-entity words representative of each type of entity with explanations obtained with LIME-ner. The position is relative to the ranking including entity words and the relevance score is the largest for that word.

(a) Top 10 non-entity words for PER for LIME-ner.

<table>
<thead>
<tr>
<th>Position</th>
<th>Word (rel. score)</th>
</tr>
</thead>
<tbody>
<tr>
<td>65</td>
<td>legend (0.43)</td>
</tr>
<tr>
<td>97</td>
<td>leader (0.39)</td>
</tr>
<tr>
<td>104</td>
<td>rider (0.39)</td>
</tr>
<tr>
<td>222</td>
<td>adviser (0.32)</td>
</tr>
<tr>
<td>282</td>
<td>cleric (0.26)</td>
</tr>
<tr>
<td>285</td>
<td>businessman (0.26)</td>
</tr>
<tr>
<td>291</td>
<td>said (0.25)</td>
</tr>
<tr>
<td>310</td>
<td>singer (0.24)</td>
</tr>
<tr>
<td>314</td>
<td>worker (0.23)</td>
</tr>
<tr>
<td>318</td>
<td>journalist (0.23)</td>
</tr>
</tbody>
</table>

(b) Top 10 non-entity words for GEO for LIME-ner.

<table>
<thead>
<tr>
<th>Position</th>
<th>Word (rel. score)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>in (0.85)</td>
</tr>
<tr>
<td>68</td>
<td>of (0.66)</td>
</tr>
<tr>
<td>73</td>
<td>to (0.65)</td>
</tr>
<tr>
<td>74</td>
<td>neighbouring (0.65)</td>
</tr>
<tr>
<td>81</td>
<td>and (0.64)</td>
</tr>
<tr>
<td>87</td>
<td>outside (0.6)</td>
</tr>
<tr>
<td>89</td>
<td>from (0.6)</td>
</tr>
<tr>
<td>92</td>
<td>near (0.6)</td>
</tr>
<tr>
<td>105</td>
<td>at (0.57)</td>
</tr>
<tr>
<td>106</td>
<td>restive (0.56)</td>
</tr>
</tbody>
</table>
6.2 Qualitative results

6.2.3 Semantic analysis

As explained in section 5.1, the semantic experiments are performed by computing a sentence embedding per entity using the explanation for this entity for the predicted class. Then, they are projected with PCA in a two-dimensional space. To evaluate the effect of the relevance score on the sentence embeddings we use a baseline where the words for the sentence embedding are weighted by the TF-IDF value on the test set instead of the relevance score. The PCA projections are shown in Figure 6.3. Each sentence embedding point is colored by the true class of the entity: orange for per and blue for geo. An interactive projection is also available for the embeddings created with explanations from LIME-as-is\(^3\) and LIME-ner\(^4\). In this interactive projection, label them by the String and color them by the Type to obtain the same projection as in Figure 6.3.

The projections show two clear clusters for each entity type, which are not present in the baseline. This shows that the words captured by the interpretation method are discriminant of the type of entity. In addition, further exploration inside each cluster shows that there are smaller semantic clusters where related entities are close by. Figure 6.3 includes some labels for some data points to illustrate this concept (better seen in the electronic version of the document with zoom). An inspection using the interactive projection showed that, for instance, Latin American locations are close by, as well as Middle East countries. For the per type, presidents are close by, as well as names with title such as Mr. In addition, LIME-ner explanations further cluster better the type per by representing close by Mr.Bush, President Bush, President Mahmoud Ahmadinejad, and other names where for each the closest neighbors are the same name. For LIME-as-is, there is a big cluster of presidents, for instance, without differentiating between them as much. This makes sense since LIME-ner is representing the sentences per entities, while LIME-as-is is doing it per words and therefore entities with common words are not differentiated. This situation does not take place in the geo entity because the entities cannot be further clustered since words in different entities do not overlap as in the person type with entities such as President.

\(^3\)https://goo.gl/e4DKux
\(^4\)https://goo.gl/NsscQc
Figure 6.3: PCA projections of the sentence embeddings weighted by TF-IDF (6.3a) and by relevance scores from LIME-as-is (6.3b) and from LIME-ner (6.3c). For both approaches of LIME, 6.3b and 6.3c, they conserve the semantic structure of the words and they are not too dissimilar between each other. The explanations are able to separate the entities by type: in orange are data points of sentence embeddings with a person entity and in blue with a geo entity. In addition, inside each type of entity, the points are also organized in small clusters. In the case of geographic locations, nearby countries are close by in the projections. In the case of the person names, they are clustered by presidents, Mr. and regular names.
6.3 Quantitative results

The quantitative evaluation was carried out using the AOPC metric as described in section 5.2. For the first experiment, the examples were perturbed according to the MoRF perturbation on correctly classified entities with explanations obtained for the true class. For the second experiment, the examples were perturbed according to the LeRF perturbation on incorrectly classified entities with the explanations obtained as well for the true class. Figure 6.4 shows the AOPC curves per type of entity and all together for both experiments. A rise in the target output should be a drop in the AOPC curve due to how the difference in equation 5.6 is taken. For a drop in the target output, it should be a rise. The baseline consists of removing features randomly and it is averaged over 10 runs with a standard deviation of less than 0.001 in all of the experiments.

Figure 6.4: Results for the two experiments with the AOPC. The first row shows the results of the first experiment (AOPC + MoRF) and the second column for the second experiment (AOPC + LeRF). For both, the first column shows the results only on the entities of type geo, the second on the entities of type per and the third on both type of entities together. The experiment has been done for LIME as-is and the corresponding random baseline and for LIME-ner and the corresponding baseline. For both LIME-ner and LIME-as-is the steepness of the curve shows that they capture the relevant words.

The results are analyzed according to the criteria described in section 5.2. For both experiments, both of the interpreting methods, LIME-as-is and LIME-ner, perform better than the baseline, showing how the sampling of words for the interpretation...
method has the behavior assumed for relevant words. In addition, this confirms that the modifications added for LIME-ner did not alter the ability to spot relevant words and provide meaningful explanations. Regarding the steepness, for all experiments, it is the same for LIME-as-is and LIME-ner and they stabilize as words get less relevant. This means that both are capturing the relevant words. For LIME-as-is this was expected since it is performing regular LIME. For LIME-ner, which is approximating only one model, this confirms that the relevant words are also identified. Notice that this behavior was expected since both the interpretation methods and the AOPC are based in a perturbation method.

Focusing on the first experiment, there is a clear difference between the results of the entity \textit{geo} and the ones for the entity \textit{person}. As previously observed in the qualitative evaluation, one can observe a different behavior of the model depending on the entity. To predict the multi-word entities of type \textit{geo} the model relies more on the words around the entity. However, to predict the multi-word entities of type \textit{person} the model relies more on the words inside the entity. Therefore, for the entities of type \textit{geo} the mitigating effect of LIME-ner (introduced in section 6.2.1) does not affect as much the final list since the model does not rely as much on the tokens inside an entity. However, for the case of the \textit{per} entity, the mitigating effect is more visible. LIME-ner is not accounting for the effect between tokens inside an entity since it is considering the entities as a group when perturbing. The relevance score obtained is, in this case, with respect to the whole entity and not per token. Due to this, the variation in the target output is smaller for LIME-ner because the greatest effect has been mitigated. Finally, the AOPC graph for both entity types shows the average behavior of the two types.

Focusing on the second experiment, the results are different per entity type as well. This experiment is carried out on incorrectly classified entities. However, notice that with just one of the tokens incorrectly classified the entity is considered incorrectly classified. However, it could be the case that the model was wrong on one token and correct in the rest. If this is the case, the explanation for each token for the true class will yield different results for each token, since for one the model was able to classify it correctly while the other it wasn’t. For correctly predicted tokens, removing negative words would not change the prediction of the token. If we observe only the \textit{geo} entity, we see that LIME-as-is has a greater variation. This might be because for the token(s) that is wrongly classified, there might be more words with negative scores than for the token(s) correctly classified. Given that the perturbations are breaking the entity, these negative words will be in the first positions of the explanation and will be removed, causing a change in the prediction of the wrongly classified token(s), as reflected in the AOPC. For the case of LIME-ner, these negative words are not captured in the same way since the perturbations are considering the entity as a whole and only retrieve words that do not support the entity as a whole, mitigating the effect of single words in the entity. This leads to a smaller variation in the target output. For the \textit{per} entity the
situation is the opposite. For this type of entity, the model relies more on the tokens inside the entity. When joining the explanations for LIME-as-is, if not all tokens are correctly predicted, the individual explanations per token disagree in the sign of the relevances for the tokens inside the entity, i.e.: the most relevant word for a token, might have a negative relevance score in the explanation of a wrongly predicted token. Because of this, it is not always guaranteed that the entity will be in the first positions of the explanation for this experiment. However, the model usually relies on it. On the other hand, for LIME-ner, given that the entity is considered as a whole, it is likely that the full entity will be one of the first words being removed since the contradictory effects just explained are mitigated. This causes then a greater variance in the probability for LIME-ner as the AOPC shows. Again, the graph for both entity types shows the average behavior of the two types. The AOPC graphs do not show significant differences between the two LIME approaches, making it hard to determine which one is better quantitatively. However, it spots aspects of the model and explanations, such as the mitigating effect, to take into account when addressing sequence tagging explanations for NER. These are discussed in the conclusions in the next chapter.

6.4 Human evaluation results

The human evaluation was carried out with 52 English-speaking participants that filled in the evaluation form provided in Appendix B. As explained in section 5.3, the form contains a description of the task and the 31 randomly sampled sentences that the user had to answer. The human responses are evaluated by counting the overlapping words with the model’s explanation top three most relevant words. Because the participants did not always consider three words to be relevant, the responses for a sentence could have less than three words. We normalized them by dividing by the length of the response, so 1, 2 or 3, depending on how many words it has. Therefore, a human response obtains a score of 1 if all the top 3 words of the model’s explanation were in the human response and a score of 0 if none of them were. The mean and standard deviation for every sentence are reported in Table 6.7a for LIME-as-is and in Table 6.7b for LIME-ner.

The results are very similar for both approaches of LIME. Both of them are able to capture the same relevant words and they differ in the order of them and the relevance scores. However, this study does not account for the relevance scores and the order since participants were not asked to rank them given that this would be a very difficult task. The form also included a question to evaluate how difficult the task provided was for the participant. Figure 6.5 shows the responses for this question. From it, we can observe that for 42% of the participants the task was not trivial and probably asking for a relevance ranking would not only be harder but also would require from them more
Observing Table 6.7, for almost all of the sentences, the mean is above 0.5 meaning that at least half of the words overlap, and for several of them it is above 0.66, meaning that at least 2/3 of the response overlapped. Given that the average is around 0.6 this means that the model’s explanations are not too far away from what humans would do, validating to some extent the patterns that the model is using. However, note that human evaluation is not a ground-truth for the explanations since human and model points of views and patterns might be different. Participants pointed out that they consider relevant also the combination of words (which was not possible to specify in the survey) and their prior knowledge. The model builds its prior knowledge from the training step, and this has been shown in examples showing that it had memorized frequent entities. LIME accounts for the effect of combination of words in its perturbation but does not report relevance on bi-grams and tri-grams. This user study shows that this could be an improvement for the interpretation method.

Analyzing some sentences more in detail, sentence number 7 had a really low mean of 0.05 for both approaches. The sentence is "The Associated Press says an early vote count in crucial Diyala province showed 78 percent of voters saying "yes" to the constitution." While for users the word "province" let them know that the entity was Diyala, for the model this was not the case. It hadn’t seen before Diyala since it is an infrequent entity, and maybe the word province as well, so it used other words to predict it. Moreover, for Diyala, the model had a confidence of 0.58 for predicting the B-GEO tag and a confidence of 0.4 for predicting the O tag so the words in the sentence were not discriminant enough for the model to get the entity with high confidence. The visual explanation can be found in Appendix A. This resulted in a low overlapping score for this sentence.

Observing the results we also find two situations: (1) sentences where the overlapping was greater for LIME-as-is than for LIME-ner and (2) sentences where the overlapping was greater for LIME-ner than for LIME-as-is. In these situations, the two
approaches returned explanations with different words in the top 3, usually one word different. For situation (1) we can observe, for instance, sentence 10, which is *Catherine Drew brings us this new form of music* where the entity is *Catherine Drew* and the explanation for LIME-as-is has in the top 3 {Catherine, Drew, brings} while LIME-ner has {Catherine, Drew, this}. For sentence 12, which is *In addition to Palestinian opposition Mr. Olmert faces another obstacle to implementing the pullout plan* where the entity is *Mr. Olmert*, the explanation for LIME-as-is has in the top 3 {Mr., Olmert, faces} while LIME-ner has {Mr., Olmert, in}. For these cases, the overlapping was greater for LIME-as-is because its explanations resembled better human expectations, since the verb in the 3rd person of singular denotes a person, while *this* and *in* do not. Nevertheless, attending to the relevance scores for this words, they are around 0.04 so very close to 0.0 and thus it would be up to the user evaluating the explanation whether to trust *in* or *this* as relevant proof. The inverse case took place in situation (2). Sentence 19 is *The military issued a statement saying the suspects and weapons were seized during a raid on a rural farm compound in eastern Iraq* where the entity is *eastern Iraq* and the explanation for LIME-as-is has in the top 3 {in, Iraq, compound} while LIME-ner has {in, Iraq, eastern}. Sentence 22 is *In his weekly radio address President Bush said he understands the skepticism some members of Congress have about certain parts of the legislation* where the entity is *President Bush* and the explanation for LIME-as-is has in the top 3 {President, said, in} while LIME-ner had {President, Bush, said}. The overlapping here was greater for LIME-ner since it was using the verb *said* which indicates person and the word *eastern* which humans used to determine that there is a location.

To sum up, the human evaluation has shown an overlapping of 0.66 words on average, implying that the model’s explanations resemble what humans would expect. Both methods for different examples included in their top 3 words that are non-relevant to the users with a very low relevance score of around 0.0. However, the human responses were also skewed and only selected very few words for their explanations showing the same skewed distribution of relevance that the task of NER has. Results show that both LIME-ner or LIME-as-is perform very similar in this top 3.
Chapter 6. Results

Table 6.7: Table 6.7a shows the human evaluation results for LIME-as-is and Table 6.7b for LIME-ner. They include the mean and standard deviation for every sentence and the mean of the means (Total) and its standard deviation. All numbers are between 0 and 1. Results show an overlapping of 0.66 words on average, validating the model’s behaviour to some extent. Also, both approaches performed similarly.

(a) Top 10 non-entities words for PER for LIME-ner.

<table>
<thead>
<tr>
<th>Sentence #</th>
<th>Mean</th>
<th>Std. dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.73</td>
<td>0.38</td>
</tr>
<tr>
<td>2</td>
<td>0.63</td>
<td>0.32</td>
</tr>
<tr>
<td>3</td>
<td>0.84</td>
<td>0.29</td>
</tr>
<tr>
<td>4</td>
<td>0.68</td>
<td>0.26</td>
</tr>
<tr>
<td>5</td>
<td>0.84</td>
<td>0.30</td>
</tr>
<tr>
<td>6</td>
<td>0.78</td>
<td>0.35</td>
</tr>
<tr>
<td>7</td>
<td>0.05</td>
<td>0.12</td>
</tr>
<tr>
<td>8</td>
<td>0.57</td>
<td>0.43</td>
</tr>
<tr>
<td>9</td>
<td>0.79</td>
<td>0.32</td>
</tr>
<tr>
<td>10</td>
<td>0.78</td>
<td>0.37</td>
</tr>
<tr>
<td>11</td>
<td>0.72</td>
<td>0.31</td>
</tr>
<tr>
<td>12</td>
<td>0.79</td>
<td>0.33</td>
</tr>
<tr>
<td>13</td>
<td>0.59</td>
<td>0.34</td>
</tr>
<tr>
<td>14</td>
<td>0.77</td>
<td>0.41</td>
</tr>
<tr>
<td>15</td>
<td>0.74</td>
<td>0.37</td>
</tr>
<tr>
<td>16</td>
<td>0.73</td>
<td>0.31</td>
</tr>
<tr>
<td>17</td>
<td>0.37</td>
<td>0.31</td>
</tr>
<tr>
<td>18</td>
<td>0.47</td>
<td>0.34</td>
</tr>
<tr>
<td>19</td>
<td>0.51</td>
<td>0.30</td>
</tr>
<tr>
<td>20</td>
<td>0.81</td>
<td>0.31</td>
</tr>
<tr>
<td>21</td>
<td>0.66</td>
<td>0.37</td>
</tr>
<tr>
<td>22</td>
<td>0.53</td>
<td>0.34</td>
</tr>
<tr>
<td>23</td>
<td>0.57</td>
<td>0.24</td>
</tr>
<tr>
<td>24</td>
<td>0.52</td>
<td>0.28</td>
</tr>
<tr>
<td>25</td>
<td>0.29</td>
<td>0.25</td>
</tr>
<tr>
<td>26</td>
<td>0.80</td>
<td>0.29</td>
</tr>
<tr>
<td>27</td>
<td>0.69</td>
<td>0.26</td>
</tr>
<tr>
<td>28</td>
<td>0.75</td>
<td>0.37</td>
</tr>
<tr>
<td>29</td>
<td>0.77</td>
<td>0.32</td>
</tr>
<tr>
<td>30</td>
<td>0.84</td>
<td>0.31</td>
</tr>
<tr>
<td>31</td>
<td>0.77</td>
<td>0.31</td>
</tr>
</tbody>
</table>

Total 0.662 0.176

(b) Top 10 non-entities words for GEO for LIME-ner.

<table>
<thead>
<tr>
<th>Sentence #</th>
<th>Mean</th>
<th>Std. dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.73</td>
<td>0.38</td>
</tr>
<tr>
<td>2</td>
<td>0.63</td>
<td>0.32</td>
</tr>
<tr>
<td>3</td>
<td>0.84</td>
<td>0.29</td>
</tr>
<tr>
<td>4</td>
<td>0.68</td>
<td>0.26</td>
</tr>
<tr>
<td>5</td>
<td>0.84</td>
<td>0.30</td>
</tr>
<tr>
<td>6</td>
<td>0.78</td>
<td>0.35</td>
</tr>
<tr>
<td>7</td>
<td>0.05</td>
<td>0.12</td>
</tr>
<tr>
<td>8</td>
<td>0.57</td>
<td>0.43</td>
</tr>
<tr>
<td>9</td>
<td>0.79</td>
<td>0.32</td>
</tr>
<tr>
<td>10</td>
<td>0.78</td>
<td>0.37</td>
</tr>
<tr>
<td>11</td>
<td>0.72</td>
<td>0.31</td>
</tr>
<tr>
<td>12</td>
<td>0.79</td>
<td>0.33</td>
</tr>
<tr>
<td>13</td>
<td>0.59</td>
<td>0.34</td>
</tr>
<tr>
<td>14</td>
<td>0.77</td>
<td>0.41</td>
</tr>
<tr>
<td>15</td>
<td>0.74</td>
<td>0.37</td>
</tr>
<tr>
<td>16</td>
<td>0.73</td>
<td>0.31</td>
</tr>
<tr>
<td>17</td>
<td>0.37</td>
<td>0.31</td>
</tr>
<tr>
<td>18</td>
<td>0.47</td>
<td>0.34</td>
</tr>
<tr>
<td>19</td>
<td>0.51</td>
<td>0.30</td>
</tr>
<tr>
<td>20</td>
<td>0.81</td>
<td>0.31</td>
</tr>
<tr>
<td>21</td>
<td>0.66</td>
<td>0.37</td>
</tr>
<tr>
<td>22</td>
<td>0.53</td>
<td>0.34</td>
</tr>
<tr>
<td>23</td>
<td>0.57</td>
<td>0.24</td>
</tr>
<tr>
<td>24</td>
<td>0.52</td>
<td>0.28</td>
</tr>
<tr>
<td>25</td>
<td>0.29</td>
<td>0.25</td>
</tr>
<tr>
<td>26</td>
<td>0.80</td>
<td>0.29</td>
</tr>
<tr>
<td>27</td>
<td>0.69</td>
<td>0.26</td>
</tr>
<tr>
<td>28</td>
<td>0.75</td>
<td>0.37</td>
</tr>
<tr>
<td>29</td>
<td>0.77</td>
<td>0.32</td>
</tr>
<tr>
<td>30</td>
<td>0.84</td>
<td>0.31</td>
</tr>
<tr>
<td>31</td>
<td>0.77</td>
<td>0.31</td>
</tr>
</tbody>
</table>

Total 0.666 0.175

Interpretability in sequence tagging models for Named Entity Recognition
Chapter 7

Conclusion and Future Work

7.1 Conclusion

During this thesis, first attempts were made towards interpretability in sequence tagging models for Named Entity Recognition (NER) and it has been shown that it is possible to obtain explanations for sequence tagging and NER. Following reasonable assumptions, two adaptations (LIME-as-is and LIME-ner) were presented to use LIME (a model-agnostic interpretation method for classification) for this task. Then, an extensive evaluation has led to understanding the model’s behavior for NER and the specific aspects of this task to take into account when identifying which characteristic should an explanation have for NER and sequence tagging.

Following the intuition that Named Entities constitute a group of words, LIME-ner implemented the necessary steps to enforce this assumption and explain entities as a group. On the other hand, LIME-as-is is a naive direct adaptation of LIME to sequence tagging. The evaluation has shown the existence of a predominant effect of a word inside the entity on other words inside the entity. This is mainly due to the fact that the model predicts token by token with a BIO tagging scheme. This considers each token separately, even inside an entity (by tagging the tokens differently with B and I). When enforcing the LIME-ner assumption this effect is mitigated. In addition, the data used exposed two opposite situations where the effect of this mitigation had different results. The entities of type geo had less of this effect and LIME-ner performed as good on them, while the entities of people had this effect much stronger and LIME-ner mitigated it most, causing quantitatively worse results than LIME-as-is. However, the mitigating effect affects mostly the relevance score of the entity words and not of other words in the sentence. Therefore, the approach of LIME-ner provides an explanation for the whole entity with the trade-off of masking this effect. For applications where the entities are very long or where the interest is on the words of the sentence outside the entity, this approach can be of benefit. However, we have also shown that applying directly LIME as-is is possible. It will lead to an explanation per word, explaining
the entity separately. This might return contradictory results for words since they can help to predict certain words but do not help to predict other words. These can become confusing when getting an idea of the words that affect the whole entity. Nevertheless, as shown in the experiments, these can be naively joined and obtain results that represent the entity.

These conclusions have been obtained through the extensive proposed evaluation. The qualitative evaluation has validated the model showing how the relevant words make sense per entity type and that are discriminant enough to identify them. Also, it has led to insights about the model’s behavior and the power of explanations thanks to comparing both approaches. For instance, the fact that the model memorizes the Named Entities and little context is used to predict it or how it is possible to characterize the entities and the relation between them and the words. The quantitative evaluation has provided means to compare both approaches and confirmed the efficacy of the explanations in identifying the relevant words even with the adaptations. It also has confirmed results for the qualitative evaluation and has shown the mitigation effect LIME-ner has. The AOPC quantitative metric was used in an effort to continue with previous work towards a unifying benchmark. Finally, the human evaluation has shown a significant overlapping of the models and human explanations, where on average 2/3 of the words overlapped. This validates, to some extent, the model’s patterns as being reasonable according to human expectations. In addition, they revealed that users also have a skewed distribution of relevance that the explanations with LIME also had, where only a few words are really relevant and other from context are not. This is an aspect to take into account for further interpretation methods for NER.

7.2 Future work

In the course of this research, other aspects were thought of that could be interesting to more extensively explore the topic.

Following the lines of adapting LIME, there are other aspects of it that, while not being directly related to NER, could improve the explanations. The perturbation function used by LIME removes words by setting them to the zero embedding. This creates artificial examples that the original data set would not have. To perturb more realistically, the perturbations could replace words by other words with the same part-of-speech tag or by a similar word, measured by a chosen similarity function.

Another option that could not be considered in this study is the use of another interpretation method. It is of value to aim for a model-agnostic method, instead of a model-dependent such as gradient-based methods, since one of the main issues in the field of explainable AI is the specificity of the methods and metrics [49], [57]. From the outlined methods in the related work section presented previously, Kernel SHAP [34] would be an interesting different method to explore. In the same aspect, experimentation
with other data set would lead to insights that would help to create a better idea on how to address explanations for NER and generalize this conclusions.
Bibliography


Chapter 7. Conclusion and Future Work


7.2 Future work


Appendix . Conclusion and Future Work


Appendix A

Individual examples visualizations

Section 6.2 included Figures that were cut to improve visualization. The complete images are included here for the first sentence (Figures A.1, A.2 and A.3) and for the second sentence (Figures A.4, A.5, A.6 and A.7). Also, Section 6.4 included the analysis of a sentence with the entity *Diyala*. Figure A.8 is the visualization for this.

Figure A.1: LIME-as-is complete explanation for token *Mark*. 
Figure A.2: LIME-as-is complete explanation for token *McClellan*.
Figure A.3: LIME-ner complete explanation for entity *Mark McClellan*.

Figure A.4: LIME-as-is complete explanation for token *Gulf*.
Appendix A. Individual examples visualizations

Figure A.5: LIME-as-is complete explanation for token *of*.

Figure A.6: LIME-as-is complete explanation for token *Aden*.
Figure A.7: LIME-ner complete explanation for entity *Gulf of Aden*.

Figure A.8: LIME-ner complete explanation for entity *Diyala*. 
Appendix B

Human evaluation

Here is presented the exercise given to the users for the human evaluation as described in 5.3.

INTRODUCTION
Dear user,
We are studying how context words can help in identifying specific types of words. This form presents 30 sentences from old newspapers headlines. Each sentence has ONLY ONE of two types of words:

• A NAME (a proper noun) of a LOCATION (the name of a city, a country, or any other location)

• Or a NAME (a proper noun) of a PERSON (title and first name, or first name and last name, etc).

along with any other possible adjectives like nationalities (lebanese) or pronouns (he, she).

TASK
Please, specify which words in the sentence, if any, could help you determine that there is one of the two types of words: the name of a Location or the name of a Person, i.e.: if you saw those words you would know that there is a Location or a Person in the sentence. It is no problem to include the Location or the Person itself (sometimes that is the only word that helps you identify it), but please, pay attention if there are any other words that would help you.

Pick as many words as you want and in case none is helpful write "none". There is no correct answers, this is just to understand human behaviour.

FORMAT EXAMPLE (this is an example for the format with a dummy sentence):
Sentence: Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Response: amet, elit, dolor
Or, if no word helps you, then:
Sentence: Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Response: none

Disclaimer: the sentences presented here have been obtained from publicly available data collected from newspapers headlines. Numbers have been replaced with 0. The survey is completely anonymous. *Required field

Sentences begin here:

More than half of China ’s reported cases have been detected since late last month.*

Officials from the Yekiti Kurdish Party say hospital officials in northeastern Syria found signs of torture on the cleric ’s body ” after it was recovered.*

Despite Mr. Zelaya ’s statements ” envoys from the two sides say negotiations will continue.*

The Lebanese government says it will cooperate with a U.N. investigation into the killing of former prime minister Rafik Hariri.*

The spokesman ” Andras Batiz ” said Tuesday three wild swans found last week have tested positive for the 0 virus.*

The Italian aid group ” Emergency ” said in a statement the videotape of kidnapped journalist Daniele Mastrogiacomo was received Wednesday.*

The Associated Press says an early vote count in crucial Diyala province showed 78 percent of voters saying ” yes ” to the constitution .*

The two men are expected to discuss recent signs North Korea may be willing to return to six-nation talks on ending its nuclear weapons program.*

Hardline Islamic cleric Ahmad Khatami on Tuesday also denounced those still opposing the Guardian ’s Council decision as against the government.*

Catherine Drew brings us this new form of music.*
Officials say the separatists have given up their demand for independence from Indonesia.*

In addition to Palestinian opposition * Mr. Olmert faces another obstacle to implementing the pullout plan.*

Last week * he led a gathering of clerics from across Afghanistan who condemned the Taliban and called on people to support the government.*

Rice welcomed the move as an important step.*

Nearby * thousands of backers of the opposition Nationalists marched for improved ties with mainland China.*

They raided her vacation home in Greece * where they found dozens of antiquities that had not been registered with the Greek government.*

There’s a new Boss in town - and his name is Simon Cowell.*

Los Angeles is one of the world’s most diverse cities * and a summer music series there brings together the city’s many ethnic communities.*

The military issued a statement saying the suspects and weapons were seized during a raid on a rural farm compound in eastern Iraq.*

Mr. Katsav resigned shortly before his term ended in 0 at the end of his presidential term * under a plea bargain that would have required him to admit to lesser charges of sexual misconduct.*

The vote came a short while after several dozen Palestinian police stormed the Palestinian parliament building in Gaza City to press demands for a security crackdown on Hamas.*

In his weekly radio address * President Bush said he understands the scepticism some members of Congress have about certain parts of the legislation.*

Two were shot and killed late last month in the southern port city of Karachi.*

They were illegally carrying a total of 0 Somalis and Ethiopians across the Gulf of Aden a few days ago.*
He first announced the plan last week during a visit by U.S. Secretary of State Condoleezza Rice.*

The newspaper says the companies have been storing information on calling patterns since the September 0 0 terrorist attacks in the United States.*

Parliament speaker Nino Burdzhanadze had said the choice was quite unexpected and other candidates had been under consideration.*

Security is tight across Kabul as Afghans commemorate the fifth anniversary of the assassination of anti-Taliban guerrilla leader.*

Last year President Gloria Arroyo ordered a massive crackdown on kidnap gangs leading to the arrests or killing of several gang leaders.*

A third British soldier died from injuries suffered in combat days ago in Helmand province.*

Cuban President Fidel Castro has walked for the first time in public since suffering a damaging fall two months ago.*

Please, rate how difficult or non-intuitive was this task on a scale from 1 (easy/intuitive) to 10 (difficult/non-intuitive).*