Explaining non-linear Classifier Decisions within Kernel-based Deep Architectures

Danilo Croce and **Daniele Rossini** and **Roberto Basili**
Department of Enterprise Engineering
University of Roma, Tor Vergata
{croce,basili}@info.uniroma2.it

Abstract

Nonlinear methods such as deep neural networks achieve state-of-the-art performances in several semantic NLP tasks. However epistemologically transparent decisions are not provided as for the limited interpretability of the underlying acquired neural models. In neural-based semantic inference tasks epistemological transparency corresponds to the ability of tracing back causal connections between the linguistic properties of a input instance and the produced classification output.

In this paper, we propose the use of a methodology, called *Layerwise Relevance Propagation*, over linguistically motivated neural architectures, namely *Kernel-based Deep Architectures* (KDA), to guide argumentations and explanation inferences. In such a way, each decision provided by a KDA can be linked to real examples, linguistically related to the input instance: these can be used to motivate the network output. Quantitative analysis shows that richer explanations about the semantic and syntagmatic structures of the examples characterize more convincing arguments in two tasks, i.e. question classification and semantic role labeling.

1 Introduction

Nonlinear methods such as deep neural networks achieve state-of-the-art performances in several challenging problems, such as image classification or natural language processing (NLP). However the traditional AI criticism still holds: they are not epistemologically transparent, as for the limited interpretability of the neural inferences.

In a question classification (QC) task, e.g. (Li and Roth, 2006), this is particularly evident. The category describing the target of a request is relevant in question answering to optimize the later stages of search and answer detection, and its interpretation depends on a variety of semantic and syntactic properties of the question. Epistemological transparency corresponds here to the ability of tracing back the connections between linguistic properties of the input question and the proposed question category. An example-driven machine learning model should be able to provide causal relations between the input semantic aspect and the properties of the question.
For example, given the prediction *"What is the capital of Zimbabwe?"* refers to a `Location`, we would like the system to motivate it with a sentence such as: *Since it seems similar to "What is the capital of California?" which also refers to a* `Location`.
Notice how in neural learning, as for example in Multilayer Perceptrons, Long Short-Term Memory Networks, (Hochreiter and Schmidhuber, 1997), or the more recent Attention-based Networks (Larochelle and Hinton, 2010), the network parameters have no clear conceptual counterpart.

Using the *Layerwise Relevance Propagation* (LRP) (Bach et al., 2015) approach, the classification decisions of a multilayer perceptron are decomposed backward across the network layers, and evidence about the contribution of individual input fragments (i.e. layer 0) to the final decision is gathered. Evaluation against images (i.e. the MNIST and ILSVRC data sets) suggests that LRP activates meaningful associations between input and output fragments, and this corresponds to tracing back meaningful causal connections.

In this paper, we propose the use of a similar mechanism over the linguistically motivated network architectures, as they have been recently proposed in (Croce et al., 2017): Kernel-based Deep network architectures aim at integrating syntactic/semantic information derived from the adoption of Tree Kernels (Collins and Duffy,

Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pages 16–24

2001) within neural-based learning. Here, we show that the inferences of such architectures can be motivated by simply applying the LRP method, which allows to trace back causal associations between the semantic classification and the examples expressed by parse tree-based metrics. Evaluation of the LRP algorithm to the problem of explaining the system decisions allows to demonstrate the meaningful impact of LRP on semantic transparency: users faced with explanations are better oriented to accept or reject the system decisions, thus improving the impact on the overall application accuracy.

In the rest of the paper, section 2 reports related works. In section 3 we describe the Kernel-based Deep Architecture (KDA) while section 4 illustrates the details of LRP and how it connects to KDAs. In section 5 we propose both a novel model to generate explanations of a network prediction and an evaluation methodology. In section 6 we provide experimental evidences of the overall system's effectiveness against two semantic tasks, question classification and frame-based argument classification in the semantic role labeling chain. Lastly, in section 7 conclusions are derived.

2 Related Work

Linguistically motivated *explanatory methods* should provide semantically clear justifications about a neural network textual inferences.

Methods making the neural learning more *readable* are usually designed to trace back the portions of the network input that mostly contributed to the output decision. Network propagation techniques are used to identify the patterns of a given input item (e.g., an image) that are linked to the particular deep neural network prediction as in (Erhan et al., 2010; Zeiler and Fergus, 2013). Usually, these are based on backward algorithms that layer-wise reuse arc weights to propagate the prediction from the output down to the input, thus leading to the re-creation of *meaningful* patterns in the input space. Typical examples are deconvolution heatmaps, used to approximate through Taylor series the partial derivatives at each layer (Simonyan et al., 2013), or the so-called Layer-wise Relevance Propagation (LRP), that redistributes back positive and negative evidence across the layers (Bach et al., 2015).

Several efforts have been made in the perspective of providing explanations of a neural classifier, often by focusing into highlighting an handful of crucial features (Baehrens et al., 2010) or deriving simpler, more readable models from a complex one, e.g. a binary decision tree (Frosst and Hinton, 2017), or by local approximation with linear models (Ribeiro et al., 2016). However, although they can explicitly show the representations learned in the specific hidden neurons (Frosst and Hinton, 2017), these approaches base their effectiveness on the user ability to study the quality of the reasoning and of the accountability as a side effect of the quality of the selected features: this can be very hard in tasks where boundaries between classes are not well defined. Sometimes, explanations are associated to vector representations as in (Ribeiro et al., 2016), i.e. bag-of-word in case of text classification, which is clearly weak at capturing significant linguistic abstractions, such as the involved syntactic relations. In this work, we propose a model which allows to provide explanations that are easily interpretable even by non-expert users, as they are expressed in natural language and are hence a more natural solution. It implicitly captures lexical, semantic and syntactic generalizations through the generation of a linguistically fluent explanation of predictions: as this is exploit linguistic analogies it provides a more transparent and epistemologically coherent view on the system's decision.

3 A Kernel-based Deep Architecture

In this section, we will first describe the Nyström method for generating low dimensional embeddings that approximate high dimensional kernel spaces. Then we will review the Kernel-based Deep Architecture discussed in (Croce et al., 2017), that efficiently combines kernel methods and deep learning by using a Nyström layer into a neural architecture.

Given an input dataset $\mathcal{D}$, a kernel $K(o_i, o_j)$ is a similarity function over $\mathcal{D}^2$ that corresponds to a dot product in the implicit kernel space, i.e., $K(o_i, o_j) = \Phi(o_i) \cdot \Phi(o_j)$. Kernel functions are used by learning algorithms, such as Support Vector Machines (Shawe-Taylor and Cristianini, 2004), to operate only implicitly on instances in the kernel space, by never accessing their explicit definition. Let us apply the projection function Φ over all examples from $\mathcal{D}$ to derive representations, $\vec{x}$ denoting the rows of the matrix

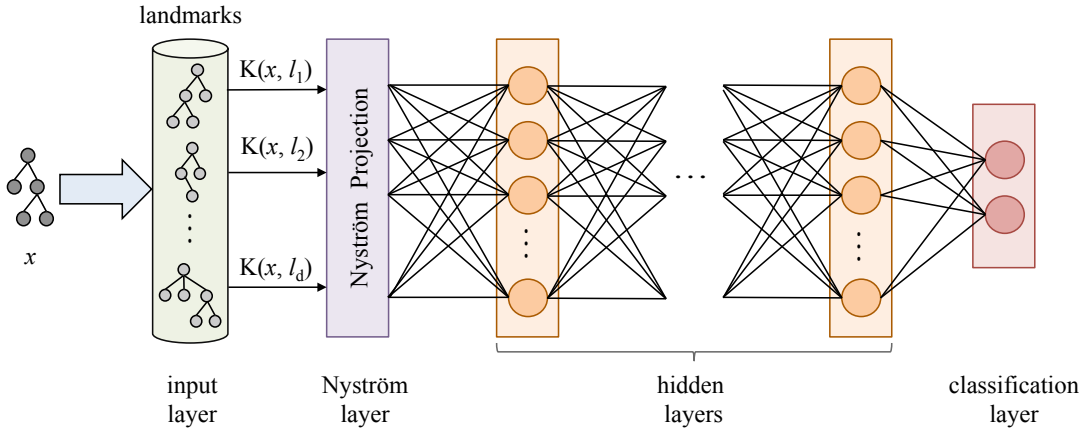

Figure 1: Kernel-based Deep Architecture.

X. The Gram matrix can always be computed as $G = XX^\top$, with each single element corresponding to $G_{ij} = \Phi(o_i)\Phi(o_j) = K(o_i, o_j)$. The aim of the Nyström method is to derive a new low-dimensional embedding $\tilde{x}$ in a l-dimensional space, with $l \ll n$ so that $\tilde{G} = \tilde{X}\tilde{X}^\top$ and $\tilde{G} \approx G$. This is obtained by generating an approximation $\tilde{G}$ of G using a subset of l columns of the matrix, i.e., a selection of a subset $L \subset \mathcal{D}$ of the available examples, called *landmarks*. Suppose we randomly sample l columns of G, and let $C \in \mathbb{R}^{|D| \times l}$ be the matrix of these sampled columns. Then, we can rearrange the columns and rows of G and define $X = [X_1 \;\; X_2]$ such that:

$$G = XX^\top = \begin{bmatrix} W & X_1^\top X_2 \\ X_2^\top X_1 & X_2^\top X_2 \end{bmatrix}$$
$$\text{and} \quad C = \begin{bmatrix} W \\ X_2^\top X_1 \end{bmatrix}$$

where $W = X_1^\top X_1$, i.e., the subset of G that contains only landmarks. The Nyström approximation can be defined as:

$$G \approx \tilde{G} = CW^\dagger C^\top \tag{1}$$

where $W^\dagger$ denotes the Moore-Penrose inverse of W. The Singular Value Decomposition (SVD) is used to obtain $W^\dagger$ as it follows. First, W is decomposed so that $W = USV^\top$, where U and V are both orthogonal matrices, and S is a diagonal matrix containing the (non-zero) singular values of W on its diagonal. Since W is symmetric and positive definite, $W = USU^\top$. Then $W^\dagger = US^{-1}U^\top = US^{-\frac{1}{2}}S^{-\frac{1}{2}}U^\top$ and the Equation 1 can be rewritten as

$$G \approx \tilde{G} = CUS^{-\frac{1}{2}}S^{-\frac{1}{2}}U^\top C^\top$$
$$= (CUS^{-\frac{1}{2}})(CUS^{-\frac{1}{2}})^\top = \tilde{X}\tilde{X}^\top$$

Given an input example $o \in \mathcal{D}$, a new low-dimensional representation $\tilde{\vec{x}}$ can be thus determined by considering the corresponding item of C as

$$\tilde{\vec{x}} = \vec{c}US^{-\frac{1}{2}} \tag{2}$$

where $\vec{c}$ is the vector whose dimensions contain the evaluations of the kernel function between o and each landmark $o_j \in L$. Therefore, the method produces l-dimensional vectors.

Notice that an optimal selection of landmarks can be expected to reduce the Gram Matrix approximation error. However, the uniform sampling without replacement policy is adopted: it is in fact theoretically and empirically shown in Kumar et al. (2012) to achieve results comparable with alternative but (more complex) selection policies.

In (Croce et al., 2017), the Nyström representation $\tilde{\vec{x}}$ has been used as input within neural network architectures. In fact, given a labeled dataset $\mathcal{L} = \{(o, y) \mid o \in \mathcal{D}, \; y \in Y\}$, where o refers to a generic instance and y is its associated class, a Multi-Layer Perceptron (MLP) architecture can be defined, with a specific Nyström layer based on the Nyström embeddings of Eq. 2. Such Kernel-based Deep Architecture (KDA) has an *input layer*, a *Nyström layer*, a possibly empty sequence of non-linear *hidden layers* and a final *classification layer*, which produces the output, as shown in Figure 1.

The *input* layer corresponds to the input vector $\vec{c}$, i.e., the row of the C matrix associated to an example o. The input layer is mapped to the *Nyström* layer, through the projection in Equation 2. Notice that the embedding provides also the proper weights, defined by $US^{-\frac{1}{2}}$, so that the mapping can be expressed through the Nyström matrix $H_{Ny} = US^{-\frac{1}{2}}$: it corresponds to a pre-trained stage derived through SVD. Formally, the low-dimensional embedding of an input example o, is $\tilde{\vec{x}} = \vec{c}\, H_{Ny} = \vec{c}\, US^{-\frac{1}{2}}$.

The resulting outcome $\tilde{\vec{x}}$ is the input to one or more non-linear *hidden* layers. Each t-th hidden layer is realized through a matrix $H_t \in \mathbb{R}^{h_{t-1} \times h_t}$ and a bias vector $\vec{b}_t \in \mathbb{R}^{1 \times h_t}$, where h_t denotes the desired hidden layer dimensionality. Clearly, given that $H_{Ny} \in \mathbb{R}^{l \times l}$, $h_0 = l$. The first hidden layer in fact receives in input $\tilde{\vec{x}} = \vec{c} H_{Ny}$, that corresponds to the $t = 0$ layer input $\vec{x}_0 = \tilde{\vec{x}}$ and its computation is formally expressed by $\vec{x}_1 = f(\vec{x}_0 H_1 + \vec{b}_1)$, where f is a non-linear activation function. In general, the generic t-th layer is modeled as:

$$\vec{x}_t = f(\vec{x}_{t-1} H_t + \vec{b}_t) \tag{3}$$

The final layer of KDA is the *classification layer*, realized through the output matrix H_O and the output bias vector $\vec{b}_O$. Their dimensionality depends on the dimensionality of the last hidden layer (called O_{-1}) and the number $|Y|$ of different classes, i.e., $H_O \in \mathbb{R}^{h_{O_{-1}} \times |Y|}$ and $\vec{b}_O \in \mathbb{R}^{1 \times |Y|}$, respectively. In particular, this layer computes a linear classification function with a softmax operator so that $\hat{y} = softmax(\vec{x}_{O_{-1}} H_O + \vec{b}_O)$.

In addition to standard dropout, a L_2 regularization is applied to the norm of each layer.

Finally, the KDA is trained by optimizing a loss function made of the sum of two factors: first, the cross-entropy function between the gold classes and the predicted ones; second the L_2 regularization, whose importance is regulated by a meta-parameter λ. The final loss function is thus

$$L(y, \hat{y}) = \sum_{(o,y) \in \mathcal{L}} y\, log(\hat{y}) + \lambda \sum_{H \in \{H_t\} \cup \{H_O\}} ||H||^2$$

where $\hat{y}$ are the softmax values computed by the network and y are the true one-hot encoding values associated with the example from the labeled training dataset $\mathcal{L}$.

As shown in Figure 1, it is worth noticing that the network is stimulated with an input vector c which contains the kernel evaluations $K(s, l_i)$ between each example and the landmarks. When using linguistic kernels (such as Semantic Tree Kernels) this measure corresponds to a syntactic/semantic similarity between the x and the subset of examples used for the space reconstruction (made available through the Nyström method). Once stimulated, the network will provide an output. In order to give an explanation to a network decision, we will discuss in the following section how to revert the propagation process connecting output and input. As a side effect we will be able to determine those landmarks mostly affecting the final decision and which are more semantically related to the input instance.

4 Layer-wise Relevance Propagation in Kernel-based Deep Architectures

Layer-wise Relevance propagation (LRP, presented in (Bach et al., 2015)) is a framework which allows to decompose the prediction of a deep neural network computed over a sample, e.g. an image, down to relevance scores for the single input dimensions of the sample such as subpixels of an image.

More formally, let $f : \mathbb{R}^d \rightarrow \mathbb{R}^+$ be a positive real-valued function taking a vector $x \in \mathbb{R}^d$ as input. The function f can quantify, for example, the probability of x being in a certain class. The Layer-wise Relevance Propagation assigns to each dimension, or feature, x_d a relevance score $R_d^{(1)}$ such that:

$$f(x) \approx \sum_d R_d^{(1)} \tag{4}$$

Features whose score is $R_d^{(1)} > 0$ or $R_d^{(1)} < 0$ correspond to evidence in favor or against, respectively, the output classification. In other words, LRP allows to identify fragments of the input playing key roles in the decision, by propagating relevance backwards. Let us suppose to know the relevance score $R_j^{(l+1)}$ of a neuron j at network layer $l+1$, then it can be decomposed into messages $R_{i \leftarrow j}^{(l,l+1)}$ sent to neurons i in layer l:

$$R_j^{(l+1)} = \sum_{i \in (l)} R_{i \leftarrow j}^{(l,l+1)} \tag{5}$$

Hence it derives that the relevance of a neuron i at layer l can be defined as:

$$R_i^{(l)} = \sum_{j \in (l+1)} R_{i \leftarrow j}^{(l,l+1)} \tag{6}$$

Note that 5 and 6 are such that 4 holds. In this work, we adopted the ϵ-rule defined in (Bach et al., 2015) to compute the messages $R_{i\leftarrow j}^{(l,l+1)}$:

$$R_{i\leftarrow j}^{(l,l+1)} = \frac{z_{ij}}{z_j + \epsilon \cdot \text{sign}(z_j)} R_j^{(l+1)}$$

where $z_{ij} = x_i w_{ij}$ and $\epsilon > 0$ is a numerical stabilizing term and must be small. The informative value is justified by the fact that the weights z_{ij} are linked to the activation weights w_{ij} of the input neurons.

If we apply it to a KDA processing linguistic observations, then LRP implicitly traces back the syntactic, semantic and lexical relations between the example and the landmarks, thus it selects the landmarks whose presences were the most influential to identify the predicted structure in the sentence. Indeed, each landmark is uniquely associated to an entry of the input vector $\vec{c}$, as illustrated in Sec 3.

5 Explanatory Models

Justifications for the KDA emissions can be obtained by explaining the evidence in favour or against a class using landmarks $\{\ell\}$ as examples. The idea is to select those $\{\ell\}$ that the LRP method produces as the most active elements in layer 0. Once such active landmarks are detected, an *Explanatory Model* is a function in charge to compile the linguistically fluent explanation by using analogies or differences with the input case. The semantic expressiveness of such analogies makes the resulting explanation clear and increases the user confidence on the system reliability. When a sentence s is classified, LRP assigns activation scores r_ℓ^s to each individual landmark ℓ: let $\mathcal{L}^{(+)}$ (or $\mathcal{L}^{(-)}$) denote the set of landmarks with positive (or negative) activation score.

Formally, every explanation is characterized by a triple $e = \langle s, C, \tau \rangle$ where s is the input sentence, C is the predicted label and τ is the modality of the explanation: $\tau = +1$ for positive (i.e. acceptance) statements while $\tau = -1$ correspond to rejections of the decision C.

A landmark ℓ is *positively activated* for a given sentence s if there are not more than $k-1$ other active landmarks ℓ' whose activation value is higher than the one for ℓ, i.e.

$$|\{\ell' \in \mathcal{L}^{(+)} : \ell' \neq \ell \wedge r_{\ell'}^s \geq r_\ell^s > 0\}| < k$$

Similarly, a landmark is *negatively activated* when:

$$|\{\ell' \in \mathcal{L}^{(-)} : \ell' \neq \ell \wedge r_{\ell'}^s \leq r_\ell^s < 0\}| < k$$

where k is a parameter used to make explanation depending on not more than k landmarks, denoted by $\mathcal{L}_k$. Positively (or negative) active landmarks in $\mathcal{L}_k$ are assigned to an activation value $a(\ell, s) = +1$ (-1), while $a(\ell, s) = 0$ for all other not activated landmarks.

Given the explanation $e = \langle s, C, \tau \rangle$, a landmark ℓ whose (known) class is C_ℓ is *consistent* (or *inconsistent*) with e according to the fact that the following function:

$$\delta(C_\ell, C) \cdot a(\ell, q) \cdot \tau$$

is positive (or negative, respectively), where $\delta(C', C) = 2\delta_{kron}(C' = C) - 1$ and δ_{kron} is the Kronecker delta.

An *explanatory model* is then a function $M(e, \mathcal{L}_k)$ which maps an explanation e, a sub set $\mathcal{L}_k$ of the active *and* consistent landmarks $\mathcal{L}$ for e into a sentence f in natural language. Of course several definitions for $M(e, \mathcal{L}_k)$ are possible. A general explanatory model would be:

$$M(e, \mathcal{L}_k) = \begin{cases} \text{'}s \text{ is } C \text{ since it is similar to } \ell\text{'} \\ \forall \ell \in \mathcal{L}_k^+ \quad \text{if } \tau > 0 \\ \\ \text{'}s \text{ is not } C \text{ since it is different} \\ \text{from } \ell \text{ which is } C\text{'} \\ \forall \ell \in \mathcal{L}_k^- \quad \text{if } \tau < 0 \\ \\ \text{'}s \text{ is } C \text{ but I don't know why'} \\ \text{if } \mathcal{L} \equiv \emptyset \end{cases}$$

where $\mathcal{L}_k^\pm$ are the partition of landmarks with positive and negative relevance scores in $\mathcal{L}_k$, respectively.

Here we introduce three explanatory models we used during experimental evaluation:

(Basic Model) The first model is the simplest. It returns an analogy only with the (unique) consistent landmark with the highest positive score if $\tau = 1$ and lowest negative score when $\tau = -1$. In case no active and consistent landmark can be found, the Basic Model returns a phrase stating only the predicted class, with no explanation. As an example the explanation of an accepted decision in an argument classification task, described by the triple $e_1 = \langle$'Put this plate in the center of the table', $\text{THEME}_{\text{PLACING}}, 1\rangle$, would be mapped by the model into:

I think **"this plate"** *is* THEME *of* PLACING *in "Robot* PUT ***this plate*** *in the center of the table" since similar to* **"the soap"** *in "Can you* PUT ***the soap*** *in the washing machine?".*

(Multiplicative Model) In a second model, denoted as *multiplicative*, the system makes reference to up to $k_1 \leq k$ analogies with positively (or negatively) active and consistent landmarks. Given the above explanation e_1, and $k_1 = 2$, it would return:

I think **"this plate"** *is* THEME *of* PLACING *in "Robot* PUT ***this plate*** *in the center of the table" since similar to* **"the soap"** *in "Can you* PUT ***"the soap"*** *in the washing machine?" and it is also similar to* **"my coat"** *in "*HANG ***my coat*** *in the closet in the bedroom".*

(Contrastive Model) The last proposed model is more complex since it returns both a positive (whether $\tau = 1$) and a negative ($\tau = -1$) analogy by selecting, respectively, the most positively relevant and the most negatively relevant consistent landmark: For instance, given $e1$, it could return:

I think **"this plate"** *is the* THEME *of* PLACING *in "Robot* PUT ***this plate*** *in the center of the table" since similar to* **"the soap"** *which is in "Can you* PUT ***the soap*** *in the washing machine" and it is not the* GOAL *of* PLACING *since different from* **"on the counter"** *in "*PUT *the plate* ***on the counter****".*

5.1 Using information theory for validating explanations

Let $P(C|s)$ and $P(C|s,e)$ be, respectively, the prior probability of the classification of s being correct and the probability of the classification being correct given an explanation. Note that both indicate the level of confidence the user has in the classifier (i.e. the KDA) given the amount of available information, i.e. with and without explanation. Three explanations are possible:

- **Useful explanations:** these are explanations such that C is correct and $P(C|s,e) > P(C|s)$ or C is not correct and $P(C|s,e) < P(C|s)$
- **Useless explanations:** they are explanations such that $P(C|s,e) = P(C|s)$
- **Misleading explanations:** they are explanations such that C is correct and $P(C|s,e) < P(C|s)$ or C is not correct and $P(C|s,e) > P(C|s)$

The core idea is that semantically coherent and exhaustive explanations must indicate correct classifications whereas incoherent or non-existent explanations must hint towards wrong classifications.

Given the above probabilities, we can measure the quality of an explanation by computing the achieved *Information Gain* (Kononenko and Bratko, 1991): the *posterior* probability is expected to grow w.r.t. to the *prior* one for correct decisions when a good explanation is available against the input sentence, while decreasing for bad or confusing explanations. The intuition behind Information Gain is that it measures the amount of information (provided in number of bits) gained by the explanation about the user decision of accepting the system classification on an incoming sentence s. A positive gain indicates that the probability amplifies towards the right decisions, and declines with errors. We will let users to judge the quality of the explanation and assign them a posterior probability that increases along with better judgments. In this way we have a measure of how convincing the system is about its decisions as well as how weak it is to clarify erroneous cases. To compare the overall performance of the different explanatory models M, the Information Gain is measured against a collection of explanations generated by M and then normalized throughout the collection's entropy E as follows:

$$I_r = \frac{1}{E}\frac{1}{\mid \mathcal{T}_s \mid}\sum_{j=1}^{|\mathcal{T}_s|} I(j) = \frac{I_a}{E} \qquad (7)$$

where $\mathcal{T}_s$ is the explanations collection and $I(j)$ is the Information Gain of explanation j.

6 Experimental Evaluation

The effectiveness of the proposed approach has been measured against two different semantic processing tasks,i.e. question classification and argument classification in semantic role labeling. The Nystrom projection has been implemented in the KeLP framework (Filice et al., 2018)[1], the neural network and LRP have been implemented in Tensorflow[2], with 1 and 2 hidden layers, respectively, whose dimensionality corresponds to the number of involved Nystrom landmarks (500 and 200, re-

[1] http://www.kelp-ml.org
[2] https://www.tensorflow.org

Category	$P(C\|s,e)$	$1-P(C\|s,e)$
V.Good	0.95	0.05
Good	0.8	0.2
Weak	0.5	0.5
Bad	0.2	0.8
Incoher.	0.05	0.95

Table 1: Posterior probabilities w.r.t. quality categories

Class	Incoher.	Bad	Weak	Good	V.Good
Incoher.	1.00	0.83	0.50	0.16	0.00
Bad	0.83	1.00	0.66	0.33	0.16
Weak	0.50	0.66	1.00	0.66	0.50
Good	0.16	0.33	0.66	1.00	0.83
V.Good	0.00	0.16	0.50	0.83	1.00

Table 2: Weights for the Cohen's Kappa κ_w statistics

spectively, randomly selected[3]), and the adoption of dropout regularization in hidden and final layers. For both tasks, hyper-parameters have been optimized via grid-search. The Adam optimizer has been applied to minimize the cross-entropy loss function, with a multi-epoch (500) training, each fed with batches of size 256. We adopted an early stop strategy, where the best model was selected according to the performance over the development set.

For evaluating our explanation method, we defined five quality categories and associated them to values for the posteriori probability $P(C|s,e)$, as shown in Table 1. We gathered into explanation datasets hundreds of explanations from the three models for each task and presented them to a pool of annotators (further details in related subsections) for independent labeling; annotators had no information of the correctness of the system emissions but just knowledge about the dataset entropy. We addressed their consensus by measuring a weighted Cohen's Kappa.

6.1 Question Classification

In our first evaluation, we replicated the experiments reported by (Croce et al., 2017) with respect to the question classification task. We thus used the UIUC dataset (Li and Roth, 2006), including a training and test set of 5452 and 500 questions, respectively, organized in 6 coarse-grained classes (as `ENTITY` or `HUMAN`). We generated Nystrom representation of the Compositionally Smoothed Partial Tree Kernel (Annesi et al., 2014) function with default parameters $\mu = \lambda = 0.4$. Using 500 landmarks, the KDA accuracy was 92.6%.

[3]More complex policies have been applied to select landmarks but statistically significant results have not been measured (not reported here due to space limitations).

	QC	SRL-AC
Basic	0.548	0.669
Multiplicative	0.514	0.662
Contrastive	0.576	0.667
κ_w	0.677	0.783
accuracy	0.926	0.961

Table 3: Information gains for the three Explanatory Models applied to the SRL-AC and QC datasets. k_w is the weighted Cohen's Kappa κ_w.

A group of 3 annotators evaluated an explanation dataset of 300 explanations (perfectly balanced between correct and not correct classification), composed of 100 explanations for each model. Performances are shown in Table 3.

All three explanatory models were able to gain more than half the required information in order to ascertain the correctness of the classification.

Consider:

I think "What year did Oklahoma become a state ?" refers to a `NUMBER` *since similar to "The film Jaws was made in what year ?"*

The model provided an evidently coherent analogy, but this is a easy case due to the occurrence in both questions of very discriminative words, i.e *"what year"*. However, the system is also able to capture semantic similarities when both syntactic and lexical features are different. E.g.:

I think "Where is the Mall of the America ?" refers to a `LOCATION` *since similar to "What town was the setting for The Music Man ?".*

This is an high-quality explanation since the system provided an analogy with a landmark requesting the same fine-grained category but with little sharing of lexical and syntactic information (note, for example, the absence in the landmark of the very discriminative word *"where"*). Let us now consider the case of wrong classifications:

I think "Mexican pesos are worth what in U.S. dollars ?" refers to a `DESCRIPTION` *since similar to "What is the Bernoulli Principle ?"*

The system provided an explanation that is not possible to easily interpret: indeed it was labeled as [Incoherent] by all the annotators.

However, system effectiveness is limited in case of negative modality for correct classifications. In these cases explanations, albeit coherent, can be trivial and do not actually help in reducing uncertainty about the correct target class. The explanation

I think "What is angiotensin ?" does not refer to a NUM *since different from "What was Einstein 's IQ ?".*

is correct but obvious. As an alternative, a negative analogy with a very likely class, i.e. ENTITY or DESCRIPTION, would have provided more useful information for disambiguation. A second challenge is represented by inherently ambiguous questions. The following explanation

I think "What is the sales tax in Minnesota ?" refers to a NUMBER *since similar to "What is the population of Mozambique ?" and does not refer to a* ENTITY *since different from "What is a fear of slime ?".*

tells why NUMBER is a more likely class than ENTITY. Although seemingly correct, this is a mistake, as ENTITY is the proper decision. However, the explanation is perfectly fine, as it well expresses the decision's rationale: lack of contextual information in the question is here the main cause of the error.

6.2 Argument Classification

Semantic role labeling (SRL (Palmer et al., 2010)) consists in detecting the semantic arguments associated with the predicate of a sentence and their classification into their specific roles (Fillmore (1985)). For example, given the sentence "*Bring the fruit onto the dining table*", the task would be to recognize the verb "*bring*" as evoking the BRINGING frame, with its roles, THEME for "*the fruit*" and GOAL for "*onto the dining table*". Argument classification corresponds to the subtask of assigning labels to the sentence fragments spanning individual roles.

As proposed in (Moschitti et al., 2008), SRL can be modeled as a multi classification task over each parse tree node n, where argument spans reflect sub-sentences covered by the tree rooted at n. Consistently with (Croce et al., 2011), in our experiments the KDA has been empowered with a Smoothed Partial Tree Kernel, operating over Grammatical Relation Centered Tree (GRCT) derived from dependency grammar.

We used the HuRIC dataset (Bastianelli et al., 2014), including over 650 annotated transcriptions of spoken robotic commands, organized in 18 frames and about 60 arguments[4]. We extracted single arguments from each HuRIC example, for a total of $1,300$ instances. We run experiments with a methodology similar to the one described in Sec 6.1, but due to the limited data size we performed extensive 10-fold cross-validation, optimizing network hyper-parameters via grid-search for each test set. We generated Nystrom representation of a equally-weighted linear combination of SPTK function with default parameters $\mu = \lambda = 0.4$ and of linear kernel function applied to sparse vector representing the instance frame. With these settings, the KDA accuracy was 96.1%. We sampled 692 explanations almost equally distributed among the 3 explanatory models. Two annotators were involved.

Results are shown in Tab 3. In this task, all models were able to gain more than two thirds of needed information. The alike scores of the three models are probably due to the narrow linguistic domain of the corpus and the well-defined semantic boundaries between the arguments. To show the capability of such models, let us consider:

I think **"the washer"** *is the* CONTAINING OBJECT *of* CLOSURE *in "Robot can you* OPEN ***the washer****?" since similar to* **"the jar"** *in "*CLOSE ***the jar****" and it is not the* THEME *of* BRINGING *since different from* **"the jar"** *in "*TAKE ***the jar*** *to the table of the kitchen".*

I think **"me"** *is the* BENEFICIARY *of* BRINGING *in "I would like some cutlery can you* GET ***me*** *some?" since similar to* **"me"** *in "*BRING ***me*** *a fork from the press." and it is not the* COTHEME *of* COTHEME *since different from* **"me"** *in "Would you please* FOLLOW ***me*** *to the kitchen?".*

The above commands have very limited lexical overlap with retrieved landmarks. Nevertheless, the analogies make explanations quite effective: explanatory models seems to successfully capture semantic and syntactic relations among input instances and closely related landmarks.

7 Conclusion

This paper investigated the effectiveness of a novel method to generate epistemologically transparent and linguistically fluid explanations for a neural predictor emissions. The proposed approach applies LRP to a KDA to backpropagate and redistribute the prediction to input entries. It then produces a sentence exploiting analogies with landmarks, according to different explanatory models. Moreover a novel evaluation methodology based on Information Theory is provided. Empirical investigations carried out against the QC and AC tasks confirm that the explanatory models contribute to increase the user confidence in the machine correct responses.

[4] http://sag.art.uniroma2.it/lu4r.html

References

Paolo Annesi, Danilo Croce, and Roberto Basili. 2014. Semantic compositionality in tree kernels. In *Proceedings of CIKM 2014*. ACM.

Sebastian Bach, Alexander Binder, Gregoire Montavon, Frederick Klauschen, Klaus-Robert Mller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7).

David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. 2010. How to explain individual classification decisions. *J. Mach. Learn. Res.*, 11:1803–1831.

Emanuele Bastianelli, Giuseppe Castellucci, Danilo Croce, Luca Iocchi, Roberto Basili, and Daniele Nardi. 2014. Huric: a human robot interaction corpus. In *LREC*, pages 4519–4526. European Language Resources Association (ELRA).

Michael Collins and Nigel Duffy. 2001. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL '02), July 7-12, 2002, Philadelphia, PA, USA*, pages 263–270. Association for Computational Linguistics, Morristown, NJ, USA.

Danilo Croce, Simone Filice, Giuseppe Castellucci, and Roberto Basili. 2017. Deep learning in semantic kernel spaces. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 345–354, Vancouver, Canada. Association for Computational Linguistics.

Danilo Croce, Alessandro Moschitti, and Roberto Basili. 2011. Structured lexical similarity via convolution kernels on dependency trees. In *Proceedings of EMNLP '11*, pages 1034–1046.

Dumitru Erhan, Aaron Courville, and Yoshua Bengio. 2010. Understanding representations learned in deep architectures. Technical Report 1355, Université de Montréal/DIRO.

Simone Filice, Giuseppe Castellucci, Giovanni Da San Martino, Alessandro Moschitti, Danilo Croce, and Roberto Basili. 2018. Kelp: a kernel-based learning platform. *Journal of Machine Learning Research*, 18(191):1–5.

Charles J. Fillmore. 1985. Frames and the semantics of understanding. *Quaderni di Semantica*, 6(2):222–254.

Nicholas Frosst and Geoffrey Hinton. 2017. Distilling a neural network into a soft decision. *CEUR Workshop Proceedings*, 2071.

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

Igor Kononenko and Ivan Bratko. 1991. Information-based evaluation criterion for classifier's performance. *Machine Learning*, 6(1):67–80.

Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. 2012. Sampling methods for the nyström method. *J. Mach. Learn. Res.*, 13:981–1006.

Hugo Larochelle and Geoffrey E. Hinton. 2010. Learning to combine foveal glimpses with a third-order boltzmann machine. In *Proceedings of Neural Information Processing Systems (NIPS)*, pages 1243–1251.

Xin Li and Dan Roth. 2006. Learning question classifiers: the role of semantic information. *Natural Language Engineering*, 12(3):229–249.

Alessandro Moschitti, Daniele Pighin, and Roberto Basili. 2008. Tree kernels for semantic role labeling. *Computational Linguistics*, 34.

M.S. Palmer, D. Gildea, and N. Xue. 2010. *Semantic Role Labeling*. Online access: IEEE (Institute of Electrical and Electronics Engineers) IEEE Morgan & Claypool Synthesis eBooks Library. Morgan & Claypool Publishers.

Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should I trust you?": Explaining the predictions of any classifier. *CoRR*, abs/1602.04938.

John Shawe-Taylor and Nello Cristianini. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, UK.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034.

Matthew D. Zeiler and Rob Fergus. 2013. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901.