IJCoL
Italian Journal of Computational Linguistics

5-1 | 2019
Emerging Topics from the Fifth Italian Conference on Computational Linguistics

On the Readability of Kernel-based Deep Learning Models in Semantic Role Labeling Tasks over Multiple Languages

Daniele Rossini, Danilo Croce and Roberto Basili

Electronic version
URL: http://journals.openedition.org/ijcol/453
DOI: 10.4000/ijcol.453
ISSN: 2499-4553

Publisher
Accademia University Press

Printed version
Number of pages: 11-31

Electronic reference
Daniele Rossini, Danilo Croce and Roberto Basili, "On the Readability of Kernel-based Deep Learning Models in Semantic Role Labeling Tasks over Multiple Languages", *IJCoL* [Online], 5-1 | 2019, Online since 01 June 2019, connection on 28 January 2021. URL: http://journals.openedition.org/ijcol/453 ; DOI: https://doi.org/10.4000/ijcol.453

On the Readability of Kernel-based Deep Learning Models in Semantic Role Labeling Tasks over Multiple Languages

Daniele Rossini[*]
Università di Roma, Tor Vergata

Danilo Croce[**]
Università di Roma, Tor Vergata

Roberto Basili[†]
Università di Roma, Tor Vergata

Sentence embeddings are effective input vectors for the neural learning of a number of inferences about content and meaning. Unfortunately, most of such decision processes are epistemologically opaque as for the limited interpretability of the acquired neural models based on the involved embeddings. In this paper, we concentrate on the readability *of neural models, discussing an embedding technique (the Nyström methodology) that corresponds to the reconstruction of a sentence in a kernel space, capturing grammatical and lexical semantic information. From this method, we build a Kernel-based Deep Architecture that is characterized by inherently high interpretability properties, as the proposed embedding is derived from examples, i.e., landmarks, that are both human readable and labeled. Its integration with an explanation methodology, the* Layer-wise Relevance Propagation, *supports here the automatic compilation of argumentations for the Kernel-based Deep Architecture decisions, expressed in form of analogy with activated landmarks. Quantitative evaluation against the Semantic Role Labeling task, both in English and Italian, suggests that explanations based on semantic and syntagmatic structures are rich and characterize convincing arguments, as they effectively help the user in assessing whether or not to trust the machine decisions.*

1. Introduction

Nonlinear methods such as Deep Neural Networks achieve state-of-the-art performances in several semantic Natural Language Processing (NLP) tasks (Goldberg 2016; Collobert et al. 2011). The wide spread of Deep Learning is supported by the impressive results and their feature learning capability (Bengio, Courville, and Vincent 2013; Kim 2014): input words and sentences are usually modeled as dense embeddings (i.e., vectors or tensors), whose dimensions correspond to latent semantic concepts acquired during an unsupervised pre-training stage. In similarity estimation, classification, emotional characterization of sentences as well as pragmatic tasks, such as question answering or dialogue, they largely demonstrated their effectiveness to model semantics.

[*] Dept. of Enterprise Engineering - Via del Politecnico 1, 00133 Rome, Italy.
E-mail: `rossini.danie@gmail.com`

[**] Dept. of Enterprise Engineering - Via del Politecnico 1, 00133 Rome, Italy.
E-mail: `croce@info.uniroma2.it`

[†] Dept. of Enterprise Engineering - Via del Politecnico 1, 00133 Rome, Italy.
E-mail: `basili@info.uniroma2.it`

Unfortunately, several drawbacks arise. First, most of the above approaches are epistemologically opaque as for the limited interpretability of the acquired neural models based on the involved embeddings. Second, injecting linguistic information into a Neural Network (NN) without degrading its transparency properties is still a problem with much room for improvement. Word embeddings are widely adopted as an effective pre-training approach, although there is no general agreement about how to provide deeper linguistic information to the NN. Some structured NN models have been proposed (Hochreiter and Schmidhuber 1997; Socher et al. 2013), although usually tailored to specific problems. Recursive NNs (Socher et al. 2013) have been shown to learn dense feature representations of the nodes in a structure, thus exploiting similarities between nodes and sub-trees. Also, Long-Short Term Memory networks (Hochreiter and Schmidhuber 1997) build intermediate representations of sequences, resulting in similarity estimates over sequences and their inner sub-sequences. However, such intermediate representations are strongly task dependent: this is beneficial from an engineering standpoint, but certainly controversial from a linguistic and cognitive point of view. In recent years, many approaches proposed extensions to the previous methods. Semi-supervised models within the multi-task learning paradigm have been investigated (Collobert et al. 2011). Context-aware dense representations (Pennington, Socher, and Manning 2014) and deep representations based on sub-words or characters (Peters et al. 2018; Devlin et al. 2019) successfully model syntactic and semantic information. Linguistically-informed mechanisms have been proposed to train the self-attention to attend syntactic information in a sentence, granting state-of-the-art results in Semantic Role Labeling (Strubell et al. 2018). However, in such approaches, the captured linguistic properties are never made explicit and the complexity of learned latent spaces only exacerbates the interpretability problem. Hence, despite state-of-the-art performances, the complexity of such approaches exacerbates the issue of a straightforward understanding of the linguistic aspects that are responsible for a network decisions. Attempts to solve the interpretability problem of NNs have been proposed in computer vision (Erhan, Courville, and Bengio 2010; Bach et al. 2015), but their extension to the NLP scenario is not straightforward.

We think that any effective approach to meaning representation should be at least epistemologically coherent, that is readable and justified through an argument theoretic lens on interpretation. This means that inferences based on vector embeddings should also naturally correspond to a clear and uncontroversial logical counterpart: in particular, neurally trained semantic inferences should be also epistemologically transparent. In other words, neural embeddings should support model readability, that is to trace back *causal connections* between the implicitly expressed linguistic properties of an input instance and the classification output produced by a model. Meaning representation should thus strictly support the (neural) learning of epistemologically well-founded models.

A possible solution is to provide *explicit information* regarding semantics by relying on linguistic properties of sentences, i.e., by modeling the lexical, syntactic and semantic constraints implicitly encoded in the linguistic structure. That is achieved by learning methods based on tree kernels (Shawe-Taylor and Cristianini 2004; Moschitti 2012; Collins and Duffy 2002a) as the feature space they capture reflects linguistic patterns. Approximation method can then be used to successfully map tree structures into dense vector representations useful to train a neural network. As suggested in (Croce et al. 2017), the Nyström dimensionality reduction method (Williams and Seeger 2001) is of particular interest as it allows to reconstruct a low-dimensional embeddings of the rich kernel space by computing kernel similarities between input examples and a set of selected instances, called *landmarks*. If methods such as Nyström's are used over rich Tree Kernels (TKs), the projection vectors will encode information captured by such kernels, which have been proved to account for syntactic as well as semantic evidence (Croce, Moschitti, and Basili 2011). The resulting vectors can be then used as input of an effective neural learner, namely a *Kernel-based Deep Architecture* (KDA), which has been shown to achieve state-of-the-art results in different semantic tasks, such as question classification and semantic role labeling, and naturally favours the generation of explanations for its decisions: this is obtained by integrating it with a model of the activation state of a network, called *Layer-wise Relevance Propagation* (*LRP*), that traces back the contribution of input layers (and nodes) to the fired output. Such input components correspond, in a KDA, to landmarks, that are real and labeled examples. Thus it is possible to compile argumentations in favor or against its inference: each decision is in fact justified via an analogy with landmarks most linguistically related to the input instance. For example, consider a Question Classification (QA) (Li and Roth 2006) task and the question: *"What year did Oklahoma become a state ?"*, in which the information request is clearly a `NUMBER`. An argument supporting such claim can be constructed by providing an analogy that highlights the linguistic properties which are relevant for the task at hand. E.g.,

Example 1
I think "What year did Oklahoma become a state ?" refers to a `NUMBER` *since it recalls me of "The film Jaws was made in what year ?"*

A speaker presented with this argument would implicitly detect the important properties shared between the target example and the one used as comparison, e.g., the syntagma *"what year"*, and implicitly evaluate both the quality of the explanation and the trustfulness of the claim (i.e., the classification output) according to the ease of his properties-detection process. In fact, consider an alternative analogy:

Example 2
I think "What year did Oklahoma become a state ?" refers to a `NUMBER` *since it recalls me of "What is the population of Mozambique ?"*

While both arguments in Example 1 and 2 are supporting a correct claim (i.e., the question refers to a numeric quantity), the second is clearly less convincing as it is harder for a human to identify the connections between the two questions (which, in this case, is due to the fact that, on a finer grain, they refers to two different sub-categories of information).

In this paper, we extend such readability-enhancing approach, proposed in (Croce, Rossini, and Basili 2019), by conducting further experimental investigations on a Semantic Role Labeling task over English and Italian, in order to test its effectiveness in multiple languages. Quantitative evaluation of these outcomes shows that richer explanations based on semantic and syntagmatic structures characterize convincing arguments, in the sense that they provide right assistance to the user in accepting or rejecting the system output. This confirms the epistemological benefit that Nyström embeddings may bring, as linguistically rich and meaningful representations of useful causal connections in a variety of inference tasks.

We first survey approaches to improve the transparency of neural models in Section 2. In Section 3, we present the role of linguistic similarity principles as they are expressed by Semantic Kernels as well as an approximation technique, the Nyström method, to derive a low-rank matrix reconstructing the input space induced by a Semantic Kernel. The KDA is illustrated in Section 4.1 while the LRP is described in Section 4.2. Section 5 is dedicated to formalizing models that take as input the result of the LRP and are able to generate explanations for the KDA decisions by exploiting its inherent transparency properties, while in Section 5.1 a method for the quantitative evaluation of explanations is defined. The overall system is evaluated against the Argument Classification (AC) step in the Semantic Role Labeling (SRL) chain (Palmer, Gildea, and Xue 2010), in two different languages: English and Italian. Results are discussed in Section 6. Finally, Section 7 summarizes achievements, open issues and future directions of this work.

2. Related work on Interpretability

Advancements of Deep Learning are allowing the exploitation of data-driven models into areas that have profound impacts on society, as health care services, criminal justice systems and financial markets. Consequently, the traditional criticism of epistemological opaqueness of AI-based systems has recently drawn much attention from the research community, as the ability for humans to understand models and suitably weight the assistance they provide is a central issue for the correct adoption of such systems. However, to empower neural models with interpretability properties is still an open problem as it even lacks a broad consensus on the definitions of interpretability and explanation.

(Lipton 2018) analyzed definitions of interpretability and transparency found in literature and structured them across two main dimensions: *Model Transparency*, i.e., understanding the mechanism by which the model works, and *Post-Hoc Explanability* (or Model Functionality), i.e., the property by which the system conveys to its users information useful to justify its functioning such as intuitive evidences supporting the output decisions. The latter can be further divided into *global* explanations, i.e., a description of the full mapping the network has learned, and *local* explanations, i.e., motivations underlying a single output. Examples of global explanations are methods that use deconvolutional networks to characterize high-layer units in a CNN for image classification (Zeiler and Fergus 2014) and approaches that derive an identity for each filter in a CNN for text classification, in terms of the captured semantic classes (Jacovi, Sar Shalom, and Goldberg 2018).

Some Local Post-Hoc Explanation methods provide visual insights, for example through a GAN[1]-generated image to assess the information detail of deep representations extracted from the input text (Spinks and Moens 2018), however, as these methods stemmed from efforts into making neural image classifiers more *readable*, they 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, Courville, and Bengio 2010; Zeiler and Fergus 2014). 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

[1] A Generative Adversarial Network (GAN) (Goodfellow et al. 2014) is a class of machine learning systems in which two networks compete in a zero-sum game: the generator has to produce synthetic data, usually from a random input signal, while the discriminator has to detect if the input it is fed with comes from the real data or it has been produced from the generator.

layer (Simonyan, Vedaldi, and Zisserman 2014), 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 a 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, Singh, and Guestrin 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 establish the quality of the reasoning and the accountability, as a side effect of the quality of the selected features: this can be very hard in tasks where no strong theory about the decision is available or the boundaries between classes are not well defined. Sometimes, explanations are associated to vector representations as in (Ribeiro, Singh, and Guestrin 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. When embeddings are used to trigger neural learning the readability of the model is a clear proof of the consistency of the adopted vectors as meaning representations, as clear understanding of what a numerical representation is describing allows human inspectors to assess whether the machine correctly modelled the target phenomena or not. Readability here refers to the property of a neural network to support *linguistically motivated explanations* about its (textual) inference. A recent methodology exploits the coupling of the classifier with some sort of generator, or decoder, responsible for the selection of output justifications: (Lei, Barzilay, and Jaakkola 2016) propose a generator that provides rationales for a multi-aspect sentiment analysis prediction by highlighting short and self-sufficient phrases in the original text.

Concerns in the research area of deriving interpretable, sparse representations from dense embeddings (Faruqui et al. 2015; Subramanian et al. 2018) have recently grown: for example, in (Trifonov et al. 2018) an effective unsupervised approach to disentangle meanings from embedding dimensions as well as automatic evaluation method have been proposed. In this work, we present a model generating *local post-hoc explanations* through analogies with previous real examples by exploiting the Layer-wise Relevance Propagation extended to a linguistically motivated neural architecture, the KDA, that exhibits a promising level of epistemological transparency. With respect to the works above, our proposal holds a few nice properties. First, the instance representation (i.e., the embedding) is derived from similarity scores estimated against real examples in the training set. Those depend on the general linguistic material as well as the task-relevant information (i.e., the target class), according to the underlying kernel. This enforces full adherence between the explanation-generation process and the decision-making process executed by the neural discriminator. Second, it is well suited to deal with short texts, where it may be difficult to highlight meaningful, yet not trivial, portions of input as justifications, as well as with the classification of segments of longer text (e.g., multi-aspect sentiment analysis) in a fashion similar to the one described for SRL in Section 6. Moreover, it provides explanations that are easily interpretable even by non-expert users, as they are inspired and expressed at language level: these are done by entire sentences and allow the human inspector to implicitly detect lexical, semantic and syntactic connections in the comparison, and consequently judge the trustworthiness of the decision, relying only on his/her linguistic competence. Lastly, the explanation-generation process is computationally inexpensive, as the LRP corresponds to a single pass of backward propagation. As discussed in Section 5, it provides a transparent and epistemologically coherent view on the system's decision.

3. Kernel methods in semantic inferences

Prediction techniques such as Support Vector Machines learn decision surfaces that correspond to hyper-planes in the original feature space by computing inner products between input examples; consequently, they are inherently linear and cannot discover nonlinear patterns in data. A possible solution is to use a mapping $\phi : x \in \mathbb{R}^n \mapsto \phi(x) \in F \subseteq \mathbb{R}^N$ such that nonlinear relations in the original space become linearly separable in the target projection space, enabling the SVM to correctly separate the data by computing inner products $\langle \phi(x_i), \phi(x_j) \rangle$ in the new feature space. However, such projections can be computationally intense. *Kernel functions* are a class of functions that allow to compute $\langle \phi(x_i), \phi(x_j) \rangle$ without explicitly accessing the input representation in the projection space. Formally, given a feature space X and a mapping ϕ from X to F, a kernel κ is any function satisfying

$$\kappa(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle \quad \forall x_i, x_j \in X \tag{1}$$

An important generalization result is the Mercer Theorem (Shawe-Taylor and Cristianini 2004), stating that for any symmetric positive semi-definite function κ there exists a mapping ϕ such that 1 is satisfied. Hence, kernels include a broad class of functions (Shawe-Taylor and Cristianini 2004). Research community has been exploring kernel methods for decades and a wide variety of kernel paradigms have been proposed. In the following sub-sections, we will illustrate advancements in Tree-Kernels (TKs, (Collins and Duffy 2002b)), as they are well suited to encode formalisms, such as dependency graphs or grammatical trees, traditionally exploited in the linguistics communities.

3.1 Semantic Kernels

Learning to solve NLP tasks usually involves the acquisition of decision models based on complex semantic and syntactic phenomena. For instance, in Paraphrase Detection, verifying whether two sentences are valid paraphrases involves rewriting rules in which the syntax plays a fundamental role. In Question Answering, the syntactic information is crucial, as largely demonstrated in (Croce, Moschitti, and Basili 2011). Similar needs are applicable to the Semantic Role Labeling task, that consists in the automatic discovery of linguistic predicates (together with their corresponding arguments) in texts. A natural approach to such problems is to apply Kernel methods (Robert Müller et al. 2001; Shawe-Taylor and Cristianini 2004), that have been traditionally proposed to decouple similarity metrics and learning algorithms in order to alleviate the impact of feature engineering in inductive processes. Kernels may directly operate on complex structures and then be used in combination with linear learning algorithms, such as Support Vector Machines (SVM, (Vapnik 1998)). Sequences (Cancedda et al. 2003) or tree kernels (Collins and Duffy 2002a) are of particular interest as the feature space they capture reflects linguistic patterns. A sentence s can be represented as a parse tree that expresses the grammatical relations implied by s: parse trees are extracted by using the Stanford Parser (Manning et al. 2014). Tree kernels (TKs, (Collins and Duffy 2002a)) can be employed to directly operate on such parse trees, evaluating the tree fragments shared by the input trees. This operation corresponds to a dot product in the implicit feature space of all possible tree fragments. Whenever the dot product is available in the implicit feature space, kernel-based learning algorithms, such as SVMs (Cortes and Vapnik 1995), can operate in order to automatically generate robust prediction models. TKs thus allow estimating the similarity among texts, directly from sentence syntactic structures, that can be represented by parse trees. The underlying idea is that the similarity between two trees T_1 and T_2 can be derived from the number of shared tree fragments. Let the set $\mathcal{T} = \{t_1, t_2, \ldots, t_{|\mathcal{T}|}\}$ be the space of all the possible substructures and $\chi_i(n_2)$ be an indicator function that is equal to 1 if the target t_i is rooted at the node n_2 and 0 otherwise. A tree-kernel function over T_1 and T_2 is defined as follows: $TK(T_1, T_2) = \sum_{n_1 \in N_{T_1}} \sum_{n_2 \in N_{T_2}} \Delta(n_1, n_2)$ where N_{T_1} and N_{T_2} are the sets of nodes of T_1 and T_2 respectively, and $\Delta(n_1, n_2) = \sum_{k=1}^{|\mathcal{T}|} \chi_k(n_1)\chi_k(n_2)$ which computes the number of common fragments between trees rooted at nodes n_1 and n_2. The feature space generated by the structural kernels obviously depends on the input structures. Notice that different tree representations embody different linguistic evidence and theories, and may produce more or less effective syntactic/semantic feature spaces for a given task.

Many available linguistic resources are enriched with formalisms dictated by Dependency grammars. They can produce a significantly different representation as exemplified in Figure 1. Since tree kernels are not tailored to model the labeled edges that are typical of dependency graphs, these latter are rewritten into explicit hierarchical representations. Different rewriting strategies are possible, as discussed in (Croce, Moschitti, and Basili 2011): a representation that is shown to be effective in several tasks is the Grammatical Relation Centered Tree (GRCT) illustrated in Figure 2: the PoS-Tags are children of grammatical function nodes and direct ancestors of their associated lexical items. Another possible representation is the Lexical Only Centered Tree (LOCT) shown in Figure 3, which contains only lexical nodes and the edges reflect some dependency relations.

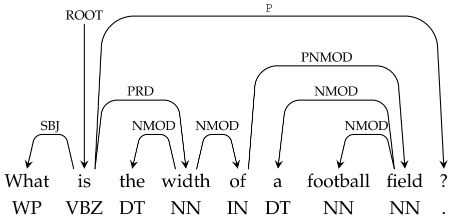
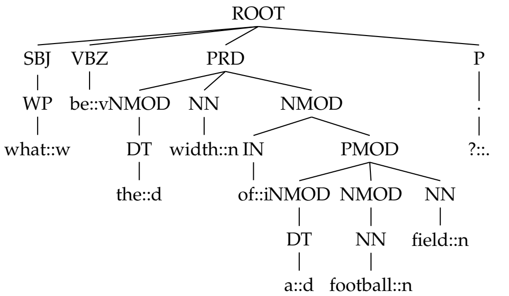

ROOT
P
PNMOD
PRD
NMOD
SBJ
NMOD
NMOD
NMOD
What is the width of a football field ?
WP VBZ DT NN IN DT NN NN .

Figure 1
Dependency Parse Tree of *"What is the width of a football field?"*.



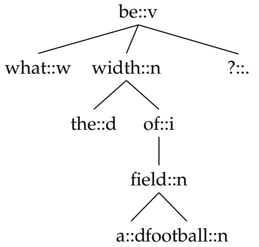


Figure 2
Grammatical Relation Centered Tree (GRCT) of *"What is the width of a football field?"*.

Figure 3
Lexical Only Centered Tree (LOCT) of *"What is the width of a football field?"*.

Different tree kernels can be defined according to the types of tree fragments considered in the evaluation of the matching structures. Subset of trees are exploited by the *Subset Tree Kernel* (Collins and Duffy 2002a), which is usually referred to as Syntactic Tree Kernel (STK); they are more general structures since their leaves can be also non-terminal symbols. The subset trees satisfy the constraint that grammatical rules cannot be broken and every tree exhaustively represents a CFG rule. *Partial Tree Kernel* (PTK, (Moschitti 2006)) relaxes this constraint considering partial trees, i.e., fragments generated by the application of partial production rules (e.g. sequences of non terminal nodes with gaps). The strict constraint imposed by the STK may be problematic especially when the training dataset is small and only few syntactic tree configurations can be observed. Overcoming this limitation, the PTK usually leads to higher accuracy, as shown by (Moschitti 2006).

Capitalizing lexical semantic information in Convolution Kernels. The tree kernels introduced in previous section perform a hard match between nodes when comparing two substructures. In NLP tasks, when nodes are words, this strict requirement reflects in a too strict lexical constraint, that poorly reflects semantic phenomena, such as the synonymy of different words or the polysemy of a lexical entry. To overcome this limitation, we adopt Distributional models of Lexical Semantics (Sahlgren 2006; Schütze 1993; Padó and Lapata 2007) to generalize the meaning of individual words by replacing them with geometrical representations (also called Word Embeddings) that are automatically derived from the analysis of large-scale corpora (Mikolov et al. 2013). These representations are based on the idea that words occurring in the same contexts tend to have similar meaning: the adopted distributional models generate vectors that are spatially close when the associated words exhibit a similar usage in large-scale document collections. Under this perspective, the distance between vectors reflects semantic relations between the represented words, such as paradigmatic relations, e.g., quasi-synonymy.[2] These word spaces allow to define meaningful soft matching between lexical nodes, in terms of the distance between their representative vectors. As a result, it is possible to obtain more informative kernel functions, which are able to capture syntactic and semantic phenomena through grammatical and lexical constraints. Notice that the supervised setting of a learning algorithm (such as SVM), operating over the resulting kernel, is augmented with the word representations generated by the unsupervised distributional methods, thus characterizing a cost-effective semi-supervised paradigm.

The *Smoothed Partial Tree Kernel* (SPTK) described in (Croce, Moschitti, and Basili 2011) exploits this idea extending the PTK formulation with a similarity function σ between nodes:

$$\Delta_{SPTK}(n_1, n_2) = \mu\lambda\sigma(n_1, n_2)\ , \ \text{ if } n_1 \text{ and } n_2 \text{ are leaves}$$

$$\Delta_{SPTK}(n_1, n_2) = \mu\sigma(n_1, n_2)\Big(\lambda^2 + \sum_{\vec{I}_1,\vec{I}_2:l(\vec{I}_1)=l(\vec{I}_2)} \lambda^{d(\vec{I}_1)+d(\vec{I}_2)} \prod_{k=1}^{l(\vec{I}_1)} \Delta_{SPTK}\left(c_{n_1}(i^1_k), c_{n_2}(i^2_k)\right)\Big) \tag{2}$$

In the SPTK formulation, the similarity function $\sigma(n_1, n_2)$ between two nodes n_1 and n_2 can be defined as follows:

- if n_1 and n_2 are both lexical nodes, then $\sigma(n_1, n_2) = \sigma_{LEX}(n_1, n_2) = \tau \frac{\vec{v}_{n_1} \cdot \vec{v}_{n_2}}{\|\vec{v}_{n_1}\| \|\vec{v}_{n_2}\|}$. It is the cosine similarity between

[2] In such spaces, vectors representing the nouns *football* and *soccer* will be near (as they are synonyms according to one of their senses) while *football* and *dog* are far

the word vectors $\vec{v}_{n_1}$ and $\vec{v}_{n_2}$ associated with the labels of n_1 and n_2, respectively. τ is called *terminal factor* and weights the contribution of the lexical similarity to the overall kernel computation.

- else if n_1 and n_2 are nodes sharing the same label, then $\sigma(n_1, n_2) = 1$.
- else $\sigma(n_1, n_2) = 0$.

The decay factors λ and μ are responsible for penalizing large child subsequences (that can include gaps) and partial sub-trees that are deeper in the structure, respectively.

3.2 The Nyström approximation

Given an input training dataset $\mathcal{D}$ of objects $o_i, i = 1 \dots N$, 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., $K(o_i, o_j) = \Phi\ (o_i) \cdot \Phi(o_j)$. The advantage of kernels is that the projection function $\Phi(o) = \vec{x} \in \mathbb{R}^n$ is never explicitly computed (Shawe-Taylor and Cristianini 2004). In fact, this operation may be prohibitive when the dimensionality n of the underlying kernel space is extremely large, as for Tree Kernels (Collins and Duffy 2002a). Kernel functions are used by learning algorithms, such as SVM, 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 o_i from $\mathcal{D}$ to derive representations, $\vec{x}_i$ denoting the i-*th* row of the matrix $\mathbf{X}$. The Gram matrix can always be computed as $\mathbf{G} = \mathbf{X}\mathbf{X}^\top$, with each single element corresponding to $\mathbf{G}_{ij} = \Phi\ (o_i)\Phi(o_j) = K(o_i, o_j)$. The aim of the Nyström method (Drineas and Mahoney 2005) is to derive a new low-dimensional embedding $\tilde{\vec{x}}$ in a l-dimensional space, with $l \ll n$ so that $\tilde{\mathbf{G}} = \tilde{\mathbf{X}}\tilde{\mathbf{X}}^\top$ and $\tilde{\mathbf{G}} \approx \mathbf{G}$. This is obtained by generating an approximation $\tilde{\mathbf{G}}$ of $\mathbf{G}$ using a subset of l columns of the Gram matrix, i.e., the kernel evaluations between all the objects $\in \mathcal{D}$ and a selection of a subset $L \subset \mathcal{D}$ of the available examples, called *landmarks*. Suppose we randomly sample l columns of $\mathbf{G}$, and let $\mathbf{C} \in \mathbb{R}^{N \times l}$ be the matrix of these sampled columns. Then, we can rearrange the columns and rows of $\mathbf{G}$ and define $\mathbf{X} = [\mathbf{X}_1\ \ \mathbf{X}_2]$ such that:

$$\mathbf{G} = \mathbf{X}\mathbf{X}^\top = \begin{bmatrix} \mathbf{W} & \mathbf{X}_1^\top\mathbf{X}_2 \\ \mathbf{X}_2^\top\mathbf{X}_1 & \mathbf{X}_2^\top\mathbf{X}_2 \end{bmatrix} \quad \text{and} \quad \mathbf{C} = \begin{bmatrix} \mathbf{W} \\ \mathbf{X}_2^\top\mathbf{X}_1 \end{bmatrix}$$

where $\mathbf{X}_1$ includes rows for the subset of $\mathbf{G}$ that contains only landmarks, $\mathbf{W} = \mathbf{X}_1^\top\mathbf{X}_1$ is their corresponding similarity matrix and finally $\mathbf{C}$ kernel evaluations between landmarks and the remaining examples. The Nyström approximation can be defined as:

$$\mathbf{G} \approx \tilde{\mathbf{G}} = \mathbf{C}\mathbf{W}^\dagger\mathbf{C}^\top \tag{3}$$

where $\mathbf{W}^\dagger$ denotes the Moore-Penrose inverse of $\mathbf{W}$. The Singular Value Decomposition (SVD) is used to obtain $\mathbf{W}^\dagger$ as follows. First, $\mathbf{W}$ is decomposed so that $\mathbf{W} = \mathbf{U}\mathbf{S}\mathbf{V}^\top$, where $\mathbf{U}$ and $\mathbf{V}$ are both orthogonal matrices, and $\mathbf{S}$ is a diagonal matrix containing the (non-zero) singular values of $\mathbf{W}$ on its diagonal. Since $\mathbf{W}$ is symmetric and positive definite, it holds that $\mathbf{W} = \mathbf{U}\mathbf{S}\mathbf{U}^\top$. Then, $\mathbf{W}^\dagger = \mathbf{U}\mathbf{S}^{-1}\mathbf{U}^\top = \mathbf{U}\mathbf{S}^{-\frac{1}{2}}\mathbf{S}^{-\frac{1}{2}}\mathbf{U}^\top$ and the Equation 3 can be rewritten as

$$\mathbf{G} \approx \tilde{\mathbf{G}} = \mathbf{C}\mathbf{U}\mathbf{S}^{-\frac{1}{2}}\mathbf{S}^{-\frac{1}{2}}\mathbf{U}^\top\mathbf{C}^\top = (\mathbf{C}\mathbf{U}\mathbf{S}^{-\frac{1}{2}})(\mathbf{C}\mathbf{U}\mathbf{S}^{-\frac{1}{2}})^\top = \tilde{\mathbf{X}}\tilde{\mathbf{X}}^\top \tag{4}$$

which explicitates the desired approximation of G in terms of the described decomposition. 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 $\mathbf{C}$ as

$$\tilde{\vec{x}} = \vec{c}\,\mathbf{U}\mathbf{S}^{-\frac{1}{2}} \tag{5}$$

where $\vec{c}$ is the vector whose j-th individual component contains the evaluation of the kernel function between o and the landmark $o_j \in L$. Therefore, the method produces l-dimensional vectors.

As discussed in the next section, the Nyströmmethod is a crucial step in our approach, as the resulting representation is inherently connected with the task at hand (each dimension of the input is linked with a real example, hence with a target class) and depends on the shared properties, between the original input example and the landmarks, captured by the exploited kernel. Notice that, while we investigated only tree linguistic kernels (as they are a natural choice in NLP), our approach can be in principle extended with any suitable kernel, as long as the captured similarities produce a satisfying representation (effectiveness of other kernel functions will be investigated in future works).

4. Interpretable Kernel-Based Deep Architectures

4.1 Kernel-based Deep Architectures

As discussed in Section 3.2, the Nyström representation $\tilde{\vec{x}}$ of any input example o is linear and can be adopted to feed a neural network architecture. We assume a labeled dataset $\mathcal{L} = \{(o, y) \mid o \in \mathcal{D},\ y \in Y\}$ is available, where o refers to a generic instance and y is its associated class. In this Section, we define a Multi-Layer Perceptron (MLP) architecture, with a specific Nyström layer based on the Nyström embeddings of Eq. 5. We will refer to this architecture, shown in Figure 4, as Kernel-based Deep Architecture (KDA). 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.

The *input* layer corresponds to the input vector $\vec{c}$, i.e., the row of the $\mathbf{C}$ matrix associated to an example o. Notice that, for adopting the KDA, the values of the matrix $\mathbf{C}$ should be all available. In the training stage, these values are in general cached. During the classification stage, the $\vec{c}$ vector corresponding to an example o is directly computed by l kernel computations between o and each of the l landmarks.

The input layer is mapped to the *Nyström* layer, through the projection in Equation 5. Notice that the embedding provides also the proper weights, defined by $\mathbf{U}\mathbf{S}^{-\frac{1}{2}}$, so that the mapping can be expressed through the Nyström matrix $\mathbf{H}_{Ny} = \mathbf{U}\mathbf{S}^{-\frac{1}{2}}$: it corresponds to a pre-trained stage derived through SVD, as discussed in Section 3.2. Equation 5 provides a static definition for $\mathbf{H}_{Ny}$ whose weights can be left invariant during the neural network training. However, the values of $\mathbf{H}_{Ny}$ can be made available for the standard back-propagation adjustments applied for training. Formally, the low-dimensional embedding of an input example o, is $\tilde{\vec{x}} = \vec{c}\,\mathbf{H}_{Ny} = \vec{c}\,\mathbf{U}\mathbf{S}^{-\frac{1}{2}}$.

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 $\mathbf{H}_t \in \mathbb{R}^{h_{t-1} \times h_t}$ and a bias vector $\vec{b}_t \in \mathbb{R}^{1 \times h_t}$, whereas h_t denotes the desired hidden layer dimensionality. Clearly, given that $\mathbf{H}_{Ny} \in \mathbb{R}^{l \times l}$, $h_0 = l$. The first hidden layer in fact receives in input $\tilde{\vec{x}} = \vec{c}\,\mathbf{H}_{Ny}$, that corresponds to $t = 0$ layer input $\vec{x}_0 = \tilde{\vec{x}}$ and its computation is formally expressed by

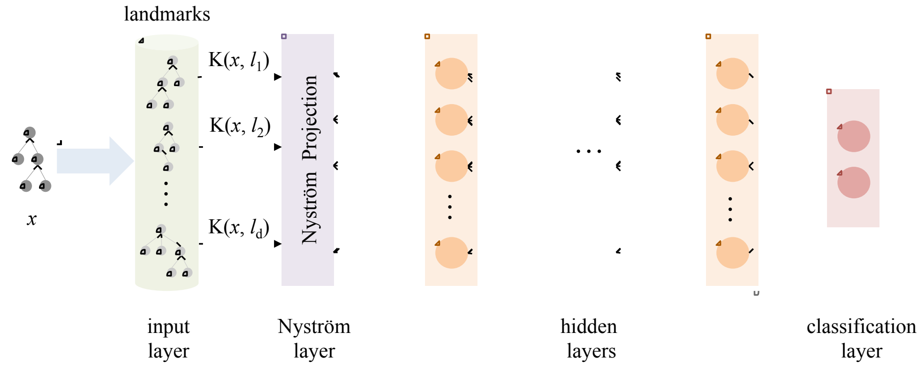


Figure 4
Kernel-based Deep Architecture.

$\vec{x}_1 = f(\vec{x}_0\mathbf{H}_1 + \vec{b}_1)$, where f is a non-linear activation function, here a Rectified Linear Unit (ReLU). In general, the generic t-th layer is modeled as:

$$\vec{x}_t = f(\vec{x}_{t-1}\mathbf{H}_t + \vec{b}_t) \tag{6}$$

The final layer of KDA is the *classification layer*, realized through the output matrix $\mathbf{H}_O$ 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., $\mathbf{H}_O \in \mathbb{R}^{h_{O_{-1}} \times |Y|}$ 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 $\hat{y} = softmax(\vec{x}_{O_{-1}}\mathbf{H}_O + \vec{b}_O)$.

In order to avoid over-fitting, two different regularization schemes are applied. First, the dropout is applied to the input $\vec{x}_t$ of each hidden layer ($t \geq 1$) and to the input $\vec{x}_{O_{-1}}$ of the final classifier. Second, 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

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

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 $\mathcal{L}$.

4.2 Layer-wise Relevance Propagation

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 individual 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:

$$f(x) \approx \textstyle\sum_d R_d^{(1)} \tag{7}$$

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:

$$R_j^{(l+1)} = \sum_{i\in(l)} R_{i\leftarrow j}^{(l,l+1)} \tag{8}$$

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

$$R_i^{(l)} = \sum_{j\in(l+1)} R_{i\leftarrow j}^{(l,l+1)} \tag{9}$$

Note that 8 and 9 are such that 7 holds. In this work, we adopted the ϵ-rule defined in (Bach et al. 2015) to compute the messages $R_{i\leftarrow j}^{(l,l+1)}$:

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

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 w_{ij} are linked to the weighted activations of the input neurons.

If we apply the above process to the KDA applied to a linguistic inference task, e.g. sentence classification, then LRP implicitly traces back the syntactic, semantic and lexical relations between the decision and the landmarks: it thus selects the landmarks that were the most influential for the predicted structure, e.g. for deciding the class of the underlying sentence. Indeed, each landmark is uniquely associated to an entry of the input vector $\vec{c}$, as illustrated in Sec 4.1.

5. Generating explanations for predictions of deep models

Justifications for the KDA decisions can be obtained by explaining the evidence in favor or against a class using the set $\mathcal{L}$ of landmarks as examples. The idea is to select those $l \in \mathcal{L}$ that the LRP method detects as the most active elements in layer 0 during the classification. Once such active landmarks are detected, an *Explanatory Model* is a function in charge of compiling a 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.

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

Similarly, a landmark ℓ is *negatively activated* when:

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

k is a parameter used to make explanation depend 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). $a(\ell, s) = 0$ 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. The *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. Note that the value of k determines the amount of consistent landmarks and hence it regulates the tradeoff between the capacity of the system to produce an explanation at all and the adherence of such explanation to the machine inference process: low values of k grant that the Model generates explanations using landmarks with high activation scores only, however they may also result in the Model being unable to produce any explanation for some decisions, i.e., when no consistent landmark is available.

Of course several definitions for $M(e, \mathcal{L}_k)$ are possible. A general explanatory model would be:

$$M(e, \mathcal{L}_k) = M(\langle s, C,\tau \rangle, \mathcal{L}_k) = \begin{cases} \text{“}s \text{ is } C \text{ since it recalls me of } \ell\text{”} \\ \forall \ell \in \mathcal{L}^+_k \quad \text{if } \tau > 0 \\ \\ \text{“}s \text{ is not } C \text{ since it does not recall me of} \\ \ell \text{ which is } C\text{”} \\ \forall \ell \in \mathcal{L}^-_k \quad \text{if } \tau < 0 \\ \\ \text{“}s \text{ is } C \text{ but I don't know why”} \\ \text{if } \mathcal{L} \equiv \emptyset \end{cases}$$

where $\mathcal{L}^+_k$ and $\mathcal{L}^-_k$ are the partition of landmarks with positive and negative relevance scores in $\mathcal{L}_k$, respectively.
Here we defined 3 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 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', $\text{THEME}_{\text{PLACING}}, 1\rangle$, the model would produce:

Example
I think **"this plate"** *is* THEME *of* PLACING *in "Robot* put ***this plate*** *in the center of the table" since it reminds me of* **"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 active and consistent landmarks. Given the above explanation e_1, and $k_1 = 2$, it would return:

Example
I think **"this plate"** *is* THEME *of* PLACING *in "Robot* put ***this plate*** *in the center of the table" since it reminds me of* **"the soap"** *in "Can you* put ***"the soap"*** *in the washing machine?" and it also reminds me of* **"my coat"** *in "*hang ***my coat*** *in the closet in the bedroom".*

(Contrastive Model). The last model we propose is more complex, since it returns both a positive (where $\tau = 1$) and a negative ($\tau = -1$) analogy by selecting, respectively, the most positively relevant and the most negatively relevant consistent landmark. For instance it could result in the following explanation:

Example
I think **"this plate"** *is the* THEME *of* PLACING *in "Robot* put ***this plate*** *in the center of the table" since it reminds me of* **"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***".

All the three models find their foundations, from argumentation theory, in a "argument by analogy" schema (Walton, Reed, and Macagno 2008). Such kind of arguments gains strength proportionally to the linguistic plausibility of the analogy: the user exposed is thus expected to implicitly gauge evidences from the linguistic properties shared between the input sentence (or its parts) and the one used for comparison. Moreover, the higher their importance (e.g. strength of activation) with respect to the output decision the larger the amount of trust in the machine verdict, accordingly.

5.1 Evaluating the quality of an explanation

In general, judging the semantic coherence of an explanation is a very difficult task. In this section we propose an approach which aims at evaluating the quality of the explanations in terms of the amount of information that a user would gather given an explanation with respect to a scenario where such explanation is not made available. More formally, let $P(C|s)$ and $P(C|s, e)$ be, respectively, the prior probability of the user believing that the classification of s is correct and the probability of the user believing that the classification of s is 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 kinds of 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 *Information Gain* (Kononenko and Bratko 1991) achieved: 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 decision of accepting the system decision about 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 in clarifying 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:

$$I_r = \frac{1}{E}\frac{1}{\mid \mathcal{T}_s \mid}\sum_{j=1}^{|\mathcal{T}_s|} I(j) = \frac{I_a}{E} \tag{10}$$

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

6. Experimental Investigations

The effectiveness of the proposed approach has been measured against the Argument Classification task in the Semantic Role Labeling chain (SRL, (Palmer, Gildea, and Xue 2010)), consisting in the detection of 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. In particular, we tested our system against two datasets, consisting of domotic commands (i.e. HuRIC, (Bastianelli et al. 2014, 2016)), in English and Italian respectively.

To evaluate the performances of proposed explanation models, we associated values for the posteriori probability $P(C|s,e)$ to five defined labels: *Very Good* if the provided explanation is clearly convincing, *Good* if the explanation is convincing but it is

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

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 2
Weights for the Cohen's Kappa κ_w statistics.

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 3
Information gains for the three Explanatory Models applied to the SRL-AC datasets in English and Italian, respectively.

	Basic	Multiplicative	Contrastive	*accuracy*
SRL-AC eng	0.669	0.663	0.667	0.961
SRL-AC ita	0.561	0.632	0.651	0.912

not completely related to the input example so that some doubts about the system decision still remain, *Uncertain* if the explanation is not useful to increase the confidence of the user with respect to the system decision, *Bad* if the explanation makes the annotator believe that the system decision is not correct while *Incoherent* corresponds to the case where the explanation is clearly inconsistent with the input example and suggests a clear error of the system in providing its answer. Corresponding values are shown in Table 1.

We gathered into explanation datasets hundreds of explanations from the three models for each task and asked annotators to perform independent labeling of them, with external knowledge only about the overall balance between explanations from incorrect and correct predictions. In case of multiple annotators, we addressed their consensus by measuring a weighted Cohen's Kappa (adopting the weights reported in Table 2).

6.1 Argument Classification in English

For the English language, the dataset included over 650 annotated transcriptions of spoken robotic commands, organized in 18 frames and about 60 arguments. We extracted single arguments from each HuRIC example, for a total of $1,300$ instances. We performed extensive 10-fold cross-validation, optimizing network hyper-parameters via grid-search for each test set. Consistently with (Croce, Moschitti, and Basili 2011), we generated Nyström representation from 200 landmarks exploiting a equally-weighted linear combination of Smoothed Partial Tree Kernel function, operating over Grammatical Relation Centered Tree (GRCT) derived from dependency grammar, with default parameters $\mu = \lambda = 0.4$, and a linear kernel function applied to sparse vector representing the instance's frame. With these settings, the KDA accuracy was 96.1%. Among the available examples, we sampled 692 explanations equally balanced among true positives, false positives, false negatives and true negatives. Due to the required balanced representation of all classes, the prior probability of the sample thus corresponds to an entropy ~ 1. Two annotators (measured Cohen's kappa is 0.783) were exposed to partially overlapping selections from the overall collection of explanations, such that explanations from the 3 models were equally distributed between the two, in order to mitigate human biases. Results are shown in Table 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.

In a scenario such as domotic Human Robotic Interfaces, the quality of individual explanatory models is very important as the robot is made capable of using explanation in a dialogue with the user. Let us consider the following examples obtained by the contrastive model:

Example
I think **"the washer"** *is the* CONTAINING OBJECT *of* CLOSURE *in "Robot can you* open ***the washer****?" since it reminds me of* **"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".*

This argumentation is very rich. It must be observed that it is not just the result of the text similarity between the examples and the question, something that is usually directly expressed by the kernel. In the example, the lexical overlap between the command and the explanation is very limited. Rather, the explanation is strictly dependent on the model and on the instance. The command cited is the one activated by the feedforward process in the KDA, i.e. the one that has been found useful in the inference. This is a dynamic side effect of the KDA model and has a dynamic nature that changes across different cases. In the situation

Example
I think **"me"** *is the* BENEFICIARY *of* BRINGING *in "I would like some cutlery can you* get ***me*** *some?" since reminds me of* **"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 role of grammatical information is more explicit also in the counterargument regarding the sentence *Would you please* follow ***me*** *to the kitchen?'*

Both the above commands have limited lexical overlap with the retrieved landmarks. Nevertheless, the retrieved analogies make the explanations quite effective: an explanatory model such as the contrastive one seems to successfully capture semantic and syntactic relations among input instances and closely related landmarks that are meaningful and epistemologically clear.

6.2 Argument Classification in Italian

Evaluation also targeted a second dataset, that is the italian HuRIC dataset (Bastianelli et al. 2016), including about 240 domotic commands, comprising of about 450 roles. As GRCT representations of instances were not available, we designed a tree representation reflecting the semantic frame structure, i.e. each sentence span corresponds to a non-lexical node labeled either as *lexical unit*, *argument* (assigned to the argument to be classified) or *other* (assigned to all other arguments). The Nyström projection was performed from a sampling of 100 landmarks. Again, this was combined with a linear kernel on sparse vector representations reflecting the instance's frame. The measured accuracy is 91.2% on about 90 examples, while the training and development set have a size of, respectively, 270 and 90 examples. Due to the small size of the data set and the high accuracy, we choose to generate an explanation dataset with a uncorrect-to-correct ratio of about 0.3, that is 144 explanations from correct predictions and 48 explanations from wrong predictions. A single annotator performed the manual labeling task, whose results are shown in Table 3: again, the information gain scores suggest that the generated explanations were able to correctly assist the human inspector in trusting or not the network decision. The slightly lower performances may be due to the skewness of the dataset, which penalizes good explanations from correct predictions, being the prior probability higher (in this case, $P(C, s) = 0.75$).
Nevertheless, the produced explanations for decisions over italian sentences exhibit similar properties to their english counterpart. For example, consider

Example
I think **"lo"** *is the* PHENOMENON *of* LOCATING *in "Ho bisogno del mio orologio* cerca ***lo*** *per favore" since reminds me of* **"una maglietta"** *in " Puoi* cercare ***una maglietta rossa*** *nell'armadietto"*
.

Except for the lexical unit, there is little lexical overlapping between the two sentences, which also have a quite different syntax. The system is also able to distinguish between different roles sharing the same lexical surface, as in:

Example
I think **"dal tavolo"** *is not the* SOURCE *of* BRINGING *in "Vai nella salda da pranzo* prendi *tutti i piatti* ***dal tavolo*** *e mettili nella lavastoviglie" since does not remind me of* **"dal tavolo"** *in "* Porta *mi il mio libro* ***dal tavolo"*** *but rather it is the* SOURCE *of* TAKING *since it reminds me of* ***"dall'appendiabiti"*** *in "* Prendi *il mio giacchetto* ***dall'appendiabiti*** *nella mia camera da letto".*

7. Conclusions

This paper extends the approach proposed in (Croce, Rossini, and Basili 2019), investigating its effectiveness in Semantic Role Labeling both for English and Italian. The proposed methodology exploits (Nyström) approximations for semantic kernel to derive vector embeddings able to support the explanation of quantitative linguistic inferences, such as those provided by neural networks. Produce explanations correspond to argumentations in natural language using analogy schemas with activated real training examples, i.e., landmarks. In order to assess the quality of compiled justification, an evaluation methodology based on Information Theory is applied. In particular, performances are measured with respect to increase in the information gain, that is decrease in entropy. Experimental results show how explanatory models provide helpful contribution to the confidence of the user in the network decision for both languages: the explanations augment confidence in correct decisions and lower down the confidence for the network errors. Given that KDA and in particular the proposed Nyström embeddings can be largely used for epistemologically clear neural learning in natural language processing, we think that they correspond to meaningful embeddings with huge potential for better neural learning models. First, they promote language semantics in a natural way and create associations between input instances and decisions that are harmonic with respect human (logical) intuition. In a sense, linguistic inferences are explained without necessarily moving out of the language level. Second, they are mathematically solid models for different levels of language semantics according to different kernel formulations. In this way the embeddings can be fine tuned to tasks, without impacting on the learning architecture but only by modeling different aspects of language syntax and semantics in the kernel function. Finally, the explanations proposed in this paper correspond just to an early stage of the research. In fact, there are many ways in which activated landmarks can be made useful in the explanation process and we are in a very early stage of such an exploration. For example, argumentation theory, as applied to the landmarks active in a decision and the source input example, can provide very rich ways to compile justification, i.e. short texts that argue for a decision.

References

Bach, Sebastian, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, Wojciech Samek, and Óscar Déniz Suárez. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. In *PloS one*, volume 10.

Baehrens, David, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. 2010. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11.

Bastianelli, Emanuele, Giuseppe Castellucci, Danilo Croce, Luca Iocchi, Roberto Basili, and Daniele Nardi. 2014. Huric: a human robot interaction corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, May. European Language Resources Association (ELRA).

Bastianelli, Emanuele, Danilo Croce, Andrea Vanzo, Roberto Basili, and Daniele Nardi. 2016. A discriminative approach to grounded spoken language understanding in interactive robotics. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 2747–2753, New York, New York, USA, July.

Bengio, Yoshua, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, August.

Cancedda, Nicola, Éric Gaussier, Cyril Goutte, and Jean-Michel Renders. 2003. Word-sequence kernels. *Journal of Machine Learning Research*, 3:1059–1082.

Collins, Michael and Nigel Duffy. 2002a. Convolution kernels for natural language. In *Advances in Neural Information Processing Systems 14 (NIPS 2001)*. pages 625–632.

Collins, Michael and Nigel Duffy. 2002b. New ranking algorithms for parsing and tagging: kernels over discrete structures, and the voted perceptron. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 263–270, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Collobert, Ronan, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12.

Cortes, Corinna and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning*, 20(3).

Croce, Danilo, 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)*, Vancouver, Canada, July. Association for Computational Linguistics.

Croce, Danilo, Alessandro Moschitti, and Roberto Basili. 2011. Structured lexical similarity via convolution kernels on dependency trees. In *Proceedings of the 2011 Conference on Empirical*

Methods in Natural Language Processing, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Croce, Danilo, Daniele Rossini, and Roberto Basili. 2019. Neural embeddings: accurate and readable inferences based on semantic kernels. *Natural Language Engineering*, 25(4):519–541.

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

Drineas, Petros and Michael W. Mahoney. 2005. On the nyström method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6.

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

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

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

Frosst, Nicholas and Geoffrey E. Hinton. 2017. Distilling a neural network into a soft decision tree. In Tarek R. Besold and Oliver Kutz, editors, *Proceedings of the First International Workshop on Comprehensibility and Explanation in AI and ML 2017 co-located with 16th International Conference of the Italian Association for Artificial Intelligence (AI*IA 2017), Bari, Italy, November 16th and 17th, 2017*, volume 2071 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Goldberg, Yoav. 2016. A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57.

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., pages 2672–2680.

Hochreiter, Sepp and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8).

Jacovi, Alon, Oren Sar Shalom, and Yoav Goldberg. 2018. Understanding convolutional neural networks for text classification. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 56–65, Brussels, Belgium, November. Association for Computational Linguistics.

Kim, Yoon. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, October. Association for Computational Linguistics.

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

Lei, Tao, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, Austin, Texas, November. Association for Computational Linguistics.

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

Lipton, Zachary C. 2018. The mythos of model interpretability. *Queue*, 16(3), June.

Manning, Christopher, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland, June. Association for Computational Linguistics.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Moschitti, Alessandro. 2006. Efficient convolution kernels for dependency and constituent syntactic trees. In *Proceedings of the 17th European Conference on Machine Learning*, Berlin, Germany, September.

Moschitti, Alessandro. 2012. State-of-the-art kernels for natural language processing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Tutorial*

Abstracts, page 2, Jeju Island, Korea, July. Association for Computational Linguistics.

Padó, Sebastian and Mirella Lapata. 2007. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2), June.

Palmer, M.S., D. Gildea, and N. Xue. 2010. *Semantic Role Labeling*. IEEE Morgan & Claypool Synthesis eBooks Library. Morgan & Claypool Publishers.

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October. Association for Computational Linguistics.

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

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*, pages 1135–1144, San Francisco, California, USA, August, 13-17. ACM.

Robert Müller, Klaus, Sebastian Mika, Gunnar Rätsch, Koji Tsuda, and Bernhard Schölkopf. 2001. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–201.

Sahlgren, Magnus. 2006. *The Word-Space Model*. Ph.D. thesis, Stockholm University.

Schütze, Hinrich. 1993. Word space. In *Advances in Neural Information Processing Systems 5*. Morgan-Kaufmann.

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

Simonyan, Karen, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Workshop Track Proceedings*, Banff, AB, Canada, April 14-16.

Socher, Richard, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, Washington, USA, October. Association for Computational Linguistics.

Spinks, Graham and Marie-Francine Moens. 2018. Evaluating textual representations through image generation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 30–39, Brussels, Belgium, November. Association for Computational Linguistics.

Strubell, Emma, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038, Brussels, Belgium, October-November.

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

Trifonov, Valentin, Octavian-Eugen Ganea, Anna Potapenko, and Thomas Hofmann. 2018. Learning and evaluating sparse interpretable sentence embeddings. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 200–210, Brussels, Belgium, November. Association for Computational Linguistics.

Vapnik, Vladimir N. 1998. *Statistical Learning Theory*. Wiley-Interscience.

Walton, Douglas, Christopher Reed, and Fabrizio Macagno. 2008. *Argumentation Schemes*. Cambridge University Press.

Williams, Christopher K. I. and Matthias Seeger. 2001. Using the nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems 13 (NIPS 2000)*. MIT Press, pages 682–688.

Zeiler, Matthew D. and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *13th European Conference on Computer Vision, ECCV 2014*, pages 818–833, Zurich, Switzerland, September.