Evaluating Kernel-based Sentence Embeddings

Danilo Croce[1], Simone Filice[2], and Roberto Basili[1]

[1] University of Roma Tor Vergata, Department of Enterprise Engineering
{croce,basili}@info.uniroma2.it
[2] Amazon Research
filicesf@amazon.com

Abstract. Kernel-based and Deep Learning methods are two of the most popular approaches in Computational Natural Language Learning. Although these models are rather different and characterized by distinct advantages and limitations, they both had impressive impact on the accuracy of complex Natural Language Processing tasks. In this work, we consider a novel neural approach that can efficiently combine kernel methods and neural networks, in the attempt of squeezing the best from the two paradigms. As dimensionality reduction methods, such as the Nyström-based projection function, can be used approximate any valid kernel function by converting underlying structures (for instance linguistic structures, such as parse trees) into dense linear embeddings, we will show how they can be used to trigger deep neural learning. Moreover, we will investigate the linguistic implication underlying the distance measures resulting in such resulting dense spaces. Empricial evaluation on real datasets suggests that the *unsupervised* Nyström embeddings are more expressive than standard vectorial text representations, i.e., Bag-of-Words or lexical word embeddings.

1 Introduction

Nowadays, a variety of machine learning approaches to Natural Language Processing (NLP) are based on Deep Learning [14, 7]. This wide spread of Deep Learning is supported by the impressive results such methods achieve, and their feature learning capability [4, 16]: input words and sentences are usually modeled as dense embeddings (i.e., vectors or tensors), whose dimensions correspond to latent semantic concepts acquired during the training phase. This largely automatizes the feature engineering phase although, on the other side, it has some inherent drawbacks. In particular, injecting linguistic information into a NN is still an open problem. If pre-trained word embeddings are widely recognized as an effective approach for improving lexical generalization, there is no general agreement about how to provide syntactic information to the NN. Some structured NN models have been proposed [15, 29] although usually tailored to specific problems. Recursive NNs [29] 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 [15] build intermediate representations of sequences, resulting in similarity estimates over sequences and their inner sub-sequences. Usually such intermediate representations are strongly task dependent: this is beneficial from an engineering standpoint, but certainly

This work was done by Simone Filice prior to joining Amazon.

controversial from a linguistic and cognitive point of view. Moreover, the linguistic information captured by the learned models is never made explicit: it is embedded in a latent space whose dimensions cannot be easily interpreted. Understanding the linguistic aspects that are responsible for the network decision is not possible in very complex architectures. Few attempts to solve the interpretability problem of NNs have been proposed in computer vision [12, 3], but their extension to the NLP scenario is not straightforward.

A natural way to provide *explicit infromation* regarding the lexical, syntactic and semantic information about training cases is by mapping them to rich linguistic structures, such as dependency graphs or constituency trees. Kernel methods [28] directly operate on such structures and their use in combination with linear learning algorithms, such as Support Vector Machines (SVM) [30], allowed to achieve very good performances in several NLP tasks, as summarized in [23]. Sequence [5] or tree kernels [6] are of particular interest as the feature space they capture reflects linguistic patterns. A viable and general solution to represent linguistic structures (e.g., parse trees) in the training of a NN, that is in form of vectors or tensors, is provided by the Nyström method [32]. It allows to approximate the Gram matrix of a kernel function and to project input examples into low-dimensional embeddings: this correspond to the vector space of reconstruction coefficients against a set of selected instances, called *landmarks*. For example, if used over Tree Kernels (TKs), the Nyström projection corresponds to the embedding of trees into low-dimensional vectors, where each vector dimension reflects the kernel similarity between any input tree and the corresponding landmark. This kind of approximation has been shown beneficial in [9] where a Nyström based low-rank embedding of input examples has been used as the early layer of a deep feed-forward network, achieving state-of-the-art results in several tasks, ranging from question classification to semantic role labeling.

In this paper, we will investigate the linguistic implication underlying the distance measures that hold within the resulting Semantic Kernel Space. Given a tree, we expect that the most similar tree is the one sharing most of the sub-tree structures, i.e. having a similar syntactic or semantic structure. The research question here is the following: *are such nice properties preserved in the low-dimensional embedding generated via the Nyström methodology?* The study of such issue requires the Nyström embeddings to preserve information by supporting the training and classification for a semantic task. We investigate the application of kernels and Nyström embeddings over the task of Semantic Textual Similarity [1] that is representative of the overall grammatical and semantic phenomena expressed by natural language sentences. First, we will test the quality by which the dot-product in the Semantic Kernel Spaces reflect semantic similarity between short texts. Then, we will use these similarities as the basis of a clustering process over the short texts (in particular questions): this will allow to verify if the topology of the embedding spaces is still able to group texts in agreement with human intuition. Results suggests that such *unsupervised* embeddings are more expressive than the standard vectorial representations used in NLP, i.e., Bag-of-Words and Word Embedding based representations.

2 Kernel-based Semantic Inference

Several NLP tasks require the explorations of complex semantic and syntactic phenomena. For instance, in Paraphrase Detection, verifying whether two sentences are valid paraphrases involves the analysis of some rewriting rules in which the syntax plays a fundamental role. In Question Answering, the syntactic information is crucial, as largely demonstrated in [10]. 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 exploit such linguistic information consists in applying kernel methods [24, 28] on structured representations of data objects, e.g., documents. A sentence s can be represented as a parse tree[3] that expresses the grammatical relations implied by s. Tree kernels (TKs) [6] 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.

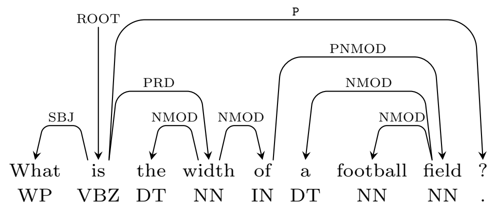

Fig. 1. Dependency Parse Tree of *"What is the width of a football field?"*.

Whenever the dot product is available in the implicit feature space, kernel-based learning algorithms, such as SVMs [8], can operate in order to automatically generate robust prediction models. TKs thus allow to estimate 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 theories and may produce more or less effective syntactic/semantic feature spaces for a given task.

Dependency grammars produce a significantly different representation which is 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,

[3] Parse trees can be extracted using automatic parsers. In our experiments, we used the Stanford Parser `https://nlp.stanford.edu/software/lex-parser.shtml`.

as discussed in [10]: 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) showed in Figure 3, which contains only lexical nodes and the edges reflect some dependency relations.

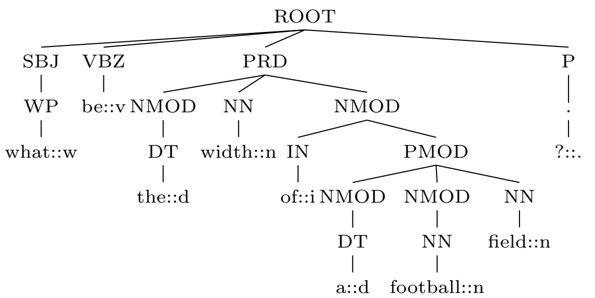

Fig. 2. Grammatical Relation Centered Tree (GRCT) of *"What is the width of a football field?"*.

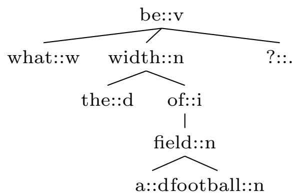

Fig. 3. Lexical Only Centered Tree (LOCT) of *"What is the width of a football field?"*.

Different tree kernels can be defined according to the types of tree fragments considered in the evaluation of the matching structures. In the *Subtree Kernel* [31], valid fragments are only the grammatically well formed and complete subtrees: every node in a subtree corresponds to a context free rule whose left hand side is the node label and the right hand side is completely described by the node descendants. Subset trees are exploited by the *Subset Tree Kernel* [6], which is usually referred to as Syntactic Tree Kernel (STK); they are more general structures since their leaves can be 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) [22] relaxes this constraint considering partial trees, i.e., fragments generated by the application of partial production rules (e.g. sequences of non terminal 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. The Partial Tree Kernel (PTK) overcomes this limitation, and usually leads to higher accuracy, as shown in [22].

Capitalizing lexical 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 [26, 19] 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. 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 similar 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[4]. 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. Moreover, 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 [10] exploits this idea extending the PTK formulation with a similarity function σ between nodes:

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

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

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 $\sigma(n_1, n_2) = \sigma_{LEX}(n_1, n_2) = \tau \frac{\mathbf{v}_{n_1} \cdot \mathbf{v}_{n_2}}{\|\mathbf{v}_{n_1}\| \|\mathbf{v}_{n_2}\|}$. It is the cosine similarity between the word vectors $\mathbf{v}_{n_1}$ and $\mathbf{v}_{n_2}$ associated with the labels of n_1 and n_2, respectively. τ is called *terminal factor* and weighs 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 $\sigma(n_1, n_2) = 0$.

Dealing with Compositionality in Tree Kernels. The main limitations of the SPTK are that (i) lexical semantic information only relies on the vector metrics applied to the leaves in a context free fashion and (ii) the semantic compositions between words is neglected in the kernel computation, that only depends on their grammatical labels.

In [2] a solution for overcoming these issues is proposed. The pursued idea is that the semantics of a specific word depends on its context. For example, in the sentence, "*What instrument does Hendrix play?*", the role of the word *instrument* is fully captured if its composition with the verb *play* is taken into account. Such combination of lexical semantic information can be directly expressed into the tree structures, as shown in Figure 4. The resulting representation is a compositional extension of a GRCT structure, where the original label d_n of grammatical

[4] 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

function nodes n (i.e., dependency relations in the tree) are augmented by also denoting their corresponding head/modifier pairs (h_n, m_n).

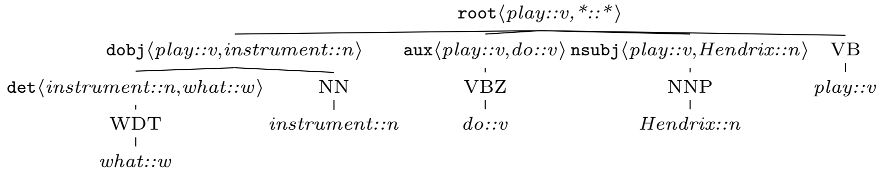

Fig. 4. Compositional Grammatical Relation Centered Tree (CGRCT) of the sentence "*What instrument does Hendrix play?*".

In CGRCTs, (sub)tree rooted at dependency nodes can be used to provide a contribution to the kernel that is a function of the composition of vectors, $\mathbf{h}$ and $\mathbf{m}$, expressing the lexical semantics of the head h and modifier m, respectively. Several algebraic functions have been proposed in [2] to compose the vectors of h=l^h::pos^h and m=l^m::pos^m into a vector $\mathbf{c}^{h,m}$ representing the head modifier pair $c = \langle \mathrm{l}^h$::pos^h,l^m::$\mathrm{pos}^m \rangle$, in line with the research on Compositional Distributional Semantics (e.g., [21]). In this work, we investigated the additive function (according to the notation proposed in [21]) that assigns to a head/modifier pair c the vector resulting from the linear combination of the vectors representing the head and the modifier, i.e., $\mathbf{c}^{h,m} = \alpha \mathbf{h} + \beta \mathbf{m}$. Although this composition method is very simple and efficient, it actually produces very effective kernel functions, as demonstrated in [2, 13]. According to the CGRCT structures, [2] defines the Compositionally Smoothed Partial Tree Kernel (CSPTK). The core novelty of the CSPTK is the compositionally enriched estimation of the function σ. The function σ can be applied to lexical nodes, to POS tag nodes as well as to augmented dependency nodes. In the algorithm the three cases are defined. For simple lexical nodes, σ consists of a lexical kernel σ_{LEX}, such as the cosine similarity between word vectors (sharing the same POS-tag): this is equivalent to [10]. For POS nodes σ consists of the identity function that is 1 only when the same POS is matched and it is 0 elsewhere.

The novelty of CSPTK corresponds to the compositional treatment of two dependency nodes, $n_1 = \langle d_1, h_1, m_1 \rangle$ and $n_2 = \langle d_2, h_2, m_2 \rangle$. The similarity function σ in this case corresponds to a compositional function σ_{Comp} between the two nodes. σ_{Comp} is not null only when the two nodes exhibit the same dependency relation, i.e. $d = d_1 = d_2$, so that also the respective heads and modifiers share the same POS labels. In all these cases a compositional metric is applied over the two involved (h_i, m_i) compounds. In the simple case, the cosine similarity between the two vectors $\mathbf{c_i}^{h_i,m_i} = \alpha \mathbf{h_i} + \beta \mathbf{m_i}$, i=1,2, is applied. Other metrics corresponds to more complex compositions $\Psi((\mathbf{h_1}, \mathbf{m_1}), (\mathbf{h_2}, \mathbf{m_2}))$ that account for linear algebra operators among the four vectors.

3 Approximating kernel spaces through Nyström

Given an input training 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., $K(o_i, o_j) =$

$\Phi(o_i) \cdot \Phi(o_j)$. The advantage of kernels is that the projection function $\Phi(o) = \mathbf{x} \in \mathbb{R}^n$ is never explicitly computed [28]. In fact, this operation may be prohibitive when the dimensionality n of the underlying kernel space is extremely large, as for Tree Kernels [6]. 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 from $\mathcal{D}$ to derive representations, $\mathbf{x}$ denoting the rows 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 [11] is to derive a new low-dimensional embedding $\tilde{\mathbf{x}}$ 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 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 $\mathbf{G}$, and let $\mathbf{C} \in \mathbb{R}^{|D| \times l}$ 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:

$$\mathbf{G} = \mathbf{X}\mathbf{X}^\top = \begin{bmatrix} \mathbf{W} & \mathbf{X}_1^\top \mathbf{X}_2 \\ \mathbf{X}_2^\top \mathbf{X}_1 & \mathbf{X}_2^\top \mathbf{X}_2 \end{bmatrix} \quad \text{and} \quad \mathbf{C} = \begin{bmatrix} \mathbf{W} \\ \mathbf{X}_2^\top \mathbf{X}_1 \end{bmatrix} \tag{2}$$

where $\mathbf{W} = \mathbf{X}_1^\top \mathbf{X}_1$, i.e., the subset of $\mathbf{G}$ that contains only landmarks. 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 it 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 $\mathbf{W} = \mathbf{U}\mathbf{S}\mathbf{U}^\top$. 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

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

Given an input example $o \in \mathcal{D}$, a new low-dimensional representation $\tilde{\mathbf{x}}$ can be thus determined by considering the corresponding item of $\mathbf{C}$ as $\tilde{\mathbf{x}} = \mathbf{c}\mathbf{U}\mathbf{S}^{-\frac{1}{2}}$ where $\mathbf{c}$ 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.

4 On the expressiveness of Semantic Kernel Spaces

In this Section, we want to support the kernel formulations provided in the previous chapters via an empirical analysis that aims at confirming that (i) adopted semantic kernels are very effective in capturing semantic and syntactic aspects of sentences, (ii) the low dimensional embeddings produced by the Nyström method preserve the expressiveness of the original kernel spaces.

High Performance Semantic Textual Similarity Estimation. Semantic Textual Similarity (STS) is the task of measuring the degree of equivalence in the underlying semantics of two snippets of text. This assessment is performed using an ordinal scale that ranges from complete semantic equivalence to complete semantic dissimilarity. State-of-the-art systems in STS are based on supervised methods that exploits rich features sets, complex alignment models and deep learning techniques (e.g., [25]). In this analysis we do not aim at competing with such systems. We just want to demonstrate that the adopted kernel functions provide a good indicator of the semantic relatedness between two sentences: in a *completely unsupervised fashion*, we will evaluate the semantic similarity between two sentences by directly using the tree kernel functions. Then, we will verify whether such similarity correlates with the similarity scores provided by the annotators.

To run this analysis we adopted the question-question portion of the STS dataset from SemEval-2016 [1]. It includes 209 question pairs extracted from the Stack Exchange Data Dump, whose topics range from highly technical areas such as programming and mathematics, to more casual topics like cooking and fitness. Table 1 reports the Pearson correlation to the gold labels of different kernel similarities. We include two baselines model to better assess our results. The $\cos_{BoW}$ is the cosine similarity of bag-of-words vectors. These vectors consider only lexical information as their dimensions reflect the occurrences of words into a text, totally ignoring word ordering or syntactic information. This produces high-dimensional sparse space (with as many dimensions as words in a dictionary) in which matching between different but semantically related words are completely neglected. Word Spaces can capture this linguistic information, where words are represented via low-dimensional embeddings where distance reflects semantic relations among represented lexical items ([26]). Here, $\cos_{W2V}$ is the cosine similarities of the vectors obtained by averaging the word vectors associated to the words of each sentence. We used 250-dimensional word vectors generated by applying the Word2vec tool with a Skip-gram model [19] to the entire Wikipedia.

Model	**Pearson**
$\cos_{BoW}$	0.077
$\cos_{W2V}$	0.086
PTK	0.202
Ny^{PTK}_{300}	0.189
Ny^{PTK}_{400}	0.202
SPTK	0.262
Ny^{SPTK}_{300}	0.252
Ny^{SPTK}_{400}	0.263

Table 1. Analysis of the Semantic Textual Similarity task.

The poor result achieved by the $\cos_{BoW}$ suggests that lexical overlap between texts is not particularly beneficial in this task at least when only the test data are considered. $\cos_{W2V}$ obtains a similar Pearson correlation: word embeddings need a better way to be combined, by using for instance the syntactic information (the SPTK is actually a way to achieve such target). We then experimented tree kernels[5] on LOCT tree representation, where all nodes are words, and edges reflect some dependency relations. Such syntactic information is crucial: both

[5] We used default values for the kernel parameters λ and μ, both set to 0.4. The terminal factor has been tuned via grid-search

PTK and SPTK significantly improve the baselines. The similarity score between two questions in thus measured in terms of the kernel function between the corresponding parse trees, without any kind of supervision. Most importantly, when the Nyström approximation of the kernel spaces is generated, overall results are not impacted. An approximated semantic kernel space generated by using only 300 landmarks, i.e., Ny_{300}^{PTK} and Ny_{300}^{SPTK}, achieve a Pearson Correlation which is only slightly lower than the one achieved by the corresponding tree kernels, while using 400 landmarks, i.e., Ny_{400}^{PTK} and Ny_{400}^{SPTK}, no difference is observed. This demonstrates that the embeddings derived by applying the Nyström method to tree kernel spaces are a semantically rich representation for text, which is largely more expressive than common text representations, such as the Bag-of-words model.

		Rank			
Sentence 1	**Sentence 2**	**Gold**	$\mathbf{cos}_{BoW}$	$\mathbf{cos}_{W2V}$	$\mathbf{Ny}_{300}^{SPTK}$
How do I remove paint from a wood floor?	*How do I remove paint from a porous table top?*	4	1	3	4
How do I remove paint from a wood floor?	*How do I remove a thick layer of paint from tiles?*	3	3-4	1	3
How do I remove paint from a wood floor?	*How can I remove paint from a deck?*	2	2	4	2
How do I remove paint from a wood floor?	*How can I remove small paint specks from a wooden floor?*	1	3-4	2	1

Table 2. Some pairs from the STS dataset. They are sorted with respect to their gold label similarity. The last four columns indicate their ranking position with respect to different models. In case of ties multiple positions are reported. The Ny_{300}^{SPTK} ranking corresponds to the one produced by the SPTK

To better appreciate the impact of different representations we reported few example pairs in Table 2. Pairs are sorted w.r.t. their gold label similarity (in these examples the gold labels range from 1 to 4). While $\cos_{BoW}$ and $\cos_{W2V}$ models introduce many errors in their rankings, the Ny_{300}^{SPTK} produces the correct ranking. In particular, the $\cos_{BoW}$ cannot match semantically similar words such as *wood* and *wooden*, resulting in a poor similarity between the last pair, i.e., the one with the highest gold label similarity. Conversely, $\cos_{W2V}$ can capture this kind of matches, however its results are still low. Probably using the average vector for combining word embeddings is not a good choice: the syntactic information of the question is completely ignored and the word embeddings have the same contribution, regardless their syntactic/semantic role in the sentence. The Ny_{300}^{SPTK}, approximating a tree kernel operating on syntactic trees, overcomes this limit, as demonstrated by its good results.

Clustering linguistic structures in Semantic Kernel Spaces. In order to prove the expressiveness of the generated semantic space, we also investigated the application of clustering techniques within the approximated Nyström spaces. The positive impact of Kernel-Based clustering methods has been already demonstrated in several works, such as [27] and [17] where kernel functions enable the clustering of data even when complex and/or non-linear topologies are involved. We selected a collection of questions from the UIUC dataset [18], composed of a training and test set of 5,452 and 500 questions, respectively. We adopted the clustering methods formulated in [17] and implemented in KeLP[6].

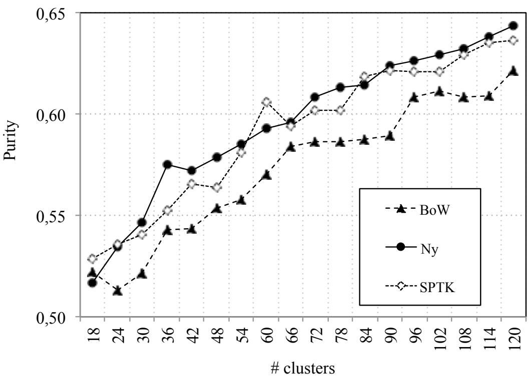

Fig. 5. Cluster purity w.r.t. the number of clusters on the task of Question Classification.

We first applied a traditional K-mean algorithm in the explicit geometrical space generated by the BoW representation of questions. Then, we evaluated a Kernel-based K-means formulation empowered with the configuration achieving best results in [9]: a CSPTK kernel applied to the CGRCT representation. Finally, we approximated the above kernel function by using 500 landmarks. We evaluated the clustering quality in terms of purity, i.e., the percentage of the most frequent class in each cluster. In fact, questions are organized in 6 classes which reflect the intent of the question itself (like `ENTITY` or `HUMAN`); as an example, given the question "*Who is the President of Pergament?*", a user would expect an answer referring to a `HUMAN` being. Figure 5 shows the purity obtained with different values of the clustering parameter K. Since the seed of the K-means formulation and the selection of the landmarks are random, we iterated this evaluation 5 times and reported the average purity across the iterations. The plot shows that the adoption of SPTK improves the purity w.r.t. the *BoW* representation. Noticeably, the results achieved in the approximated space (the Ny curve) overlap the ones achieved by the kernel counterpart.A deeper analysis of the clusters obtained in the reduced space is reported in Table 4, which reports 4 of the 100 clusters obtained by the standard K-mean algorithm over the approximated Nyström space. The syntactic information captured by the tree kernel is clearly shown by the items in the first two clusters, that are in the form "*What are/is*" and "*What is*". Most noticeably, in the second cluster all questions refer to *leader*s, *president*s or *prime minister*, as these are semantically related according the adopted lexical word embedding. The third and forth clusters are more interesting because they do not contain questions sharing the same structure, but the linguistic generalization is more evident, where locations (such as *countries* and *mountains*) and group of people are addressed by the questions. Most importantly, this combination of lexical, syntactic and semantic information is coded in the 500-dimension of the approximated kernel space. The derived clusters are very expressive in linguistic terms. In fact almost all clusters correspond more or less explicitly to one or more syntactic-semantic patterns, such as **Cluster 3.**

What [`LOC`] {*border, surround, is bounded by, comprise*} [`LOC`] ?

or **Cluster 4.**

[6] `http://www.kelp-ml.org/?page_id=799`

What [HUM] *(did)* [HUM] {*become*} ?
What [HUM] {*did, does*} [HUM] { { *play* } [sport]}, *advertise*} {*for*}?

Cluster 1	
DESC	*What are vermicilli, rigati, zitoni, and tubetti?*
DESC	*What are liver enzymes?*
DESC	*What are amaretto biscuits?*
DESC	*What are tonsils for?*
DESC	*What are hook worms?*
DESC	*What are some chemical properties of mendelevium?*
ENTY	*What are birds descendents of?*
DESC	*What are some children 's rights?*

Cluster 2	
HUM	*Who is the President of Pergament?*
HUM	*Who is the leader of Brunei?*
HUM	*Who is the president of Bolivia?*
HUM	*Who is the President of Ghana?*
HUM	*Who is the leader of India?*
HUM	*Who was the president of Vichy France?*
HUM	*Who was the 1st U.S. President?*
HUM	*Who is the prime minister of Japan?*
HUM	*Who was the oldest U.S. president?*

Cluster 3	
LOC	*What two countries ' coastlines border the Bay of Biscay?*
LOC	*What country is bounded in part by the Indian Ocean and Coral and Tasman seas?*
LOC	*What country do the Galapagos Islands belong to?*
LOC	*What part of Britain comprises the Highlands, Central Lowlands, and Southern Uplands?*
LOC	*What two Caribbean countries share the island of Hispaniola?*
LOC	*What country surrounds San Marino, the world 's smallest Republic?*
LOC	*What mountain range marks the border of France and Spain?*
LOC	*What strait links the Mediterranean Sea and the Atlantic Ocean?*
LOC	*What U.S. state includes the San Juan Islands?*

Cluster 4	
ENTY	*What basketball maneuver did Bert Loomis invent?*
HUM	*What college did Joe Namath play football for?*
HUM	*What hockey team did Wayne Gretzky play for?*
HUM	*What dumb-but-loveable character did Maurice Gosfield play on The Phil Silvers Show?*
HUM	*What Cruise Line does Kathie Lee Gifford advertise for?*
HUM	*What team did baseball 's St. Louis Browns become?*
ENTY	*What war did Johnny Reb and Billy Yank fight?*
HUM	*What feathered cartoon characters do Yugoslavians know as Vlaja, Gaja, and Raja?*
...	...

Table 3. Example of question clusters in the Semantic Kernel Space.

5 Conclusions

Quantitative approaches to language semantics are often difficult to evaluate and explain as for the lack of explicit interpretation functions acting on the models acquired through supervised or unsupervised learning. In this paper Nyström embeddings, proposed as approximation of distance metrics (i.e. kernel functions) able to support dimensionality reduction, are proposed as linear representations for syntactic and semantic phenomena in natural languages. The so-called Nyström embeddings thus correspond to vectors in semantic spaces, determined by the reference semantic kernels. Specifically, the vector corresponds to the reconstruction coefficients against a set of landmarks. In line with results presented in previous papers, this work explores the linguistic readability of such vector representations.

In particular, two NLP tasks are studied and the adoption of such linear representation is compared against other linear methods, namely bag-of-words, largely used in Information Retrieval, and lexical embeddings, i.e. [20], often applied as a pre-training mechanism in neural learning. The first task is semantic similarity estimation and helps in observing the impact of the adopted Nyström vectors as linear correspondents of patterns corresponding to semantically similar sentences. Results suggest that correlations between sentence pairs as estimated by semantic tree kernels improve significantly with respect to other lexical embeddings, e.g. neural language models such as [20]. Unsupervised clustering of NL questions is the second task that shows how semantic phenomena (e.g. the class of questions in natural language in a Question Answering task) behave regularly in the kernel space, even when the Nyström approximations are used. In this way, the linear representations obtained through the Nyström vectors cluster in the space in a semantically coherent way.

Along this line of research, more NL inference tasks and different natural languages will be involved in the future experiments in order to assess the semantic coherence of the Nyström embeddings on a wider set of linguistic phenomena and generalize them, correspondingly. Moreover, our aim is using these vectors not only as triggers for neural learning, as proposed in [9], but mostly as flexible representations for semantic phenomena. They can be retrieved from a neural models: they in fact are isomorphic to the parameters of one or more layers in a network and can be thus adopted to explain the neural model as encoded in the network layers: this enables to explain a decision according to its resemblance to know examples and patterns.

References

1. Agirre, E., Banea, C., Cer, D.M., Diab, M.T., Gonzalez-Agirre, A., Mihalcea, R., Rigau, G., Wiebe, J.: Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In: Bethard, S., Cer, D.M., Carpuat, M., Jurgens, D., Nakov, P., Zesch, T. (eds.) SemEval@NAACL-HLT. pp. 497–511 (2016)
2. Annesi, P., Croce, D., Basili, R.: Semantic compositionality in tree kernels. In: Proceedings of CIKM 2014. ACM (2014)
3. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W., Suárez, Ó.D.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. In: PloS one (2015)
4. Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell. 35(8), 1798–1828 (Aug 2013)
5. Cancedda, N., Gaussier, É., Goutte, C., Renders, J.M.: Word-sequence kernels. Journal of Machine Learning Research 3, 1059–1082 (2003)
6. Collins, M., Duffy, N.: Convolution kernels for natural language. In: Proceedings of Neural Information Processing Systems (NIPS'2001). pp. 625–632 (2001)
7. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. Journal of Machine Learning Research (JAIR) 12, 2493–2537 (Nov 2011)
8. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. 20(3), 273–297 (Sep 1995)
9. Croce, D., Filice, S., Castellucci, G., Basili, R.: Deep learning in semantic kernel spaces. In: Proceedings of ACL2017. pp. 345–354 (2017)
10. Croce, D., Moschitti, A., Basili, R.: Structured lexical similarity via convolution kernels on dependency trees. In: Proceedings of EMNLP '11. pp. 1034–1046 (2011)

11. Drineas, P., Mahoney, M.W.: On the nyström method for approximating a gram matrix for improved kernel-based learning. J. Mach. Learn. Res. 6, 2153–2175 (Dec 2005)
12. Erhan, D., Courville, A., Bengio, Y.: Understanding representations learned in deep architectures. Tech. Rep. 1355, Université de Montréal/DIRO (Oct 2010)
13. Filice, S., Castellucci, G., Croce, D., Basili, R.: Kelp: a kernel-based learning platform for natural language processing. In: Proceedings of ACL: System Demonstrations. Beijing, China (July 2015)
14. Goldberg, Y.: A primer on neural network models for natural language processing. Journal of Artificial Intelligence Research (JAIR) 57, 345–420 (2016)
15. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. 9(8), 1735–1780 (Nov 1997)
16. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings EMNLP 2014. pp. 1746–1751. Doha, Qatar (October 2014)
17. Kulis, B., Basu, S., Dhillon, I., Mooney, R.: Semi-supervised graph clustering: A kernel approach. In: Proc. of ICML. pp. 457–464. ACM (2005)
18. Li, X., Roth, D.: Learning question classifiers: the role of semantic information. Natural Language Engineering 12(3), 229–249 (2006)
19. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. CoRR abs/1301.3781 (2013), `http://arxiv.org/abs/1301.3781`
20. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. CoRR (2013)
21. Mitchell, J., Lapata, M.: Composition in distributional models of semantics. Cognitive Science 34(8), 1388–1429 (2010)
22. Moschitti, A.: Efficient convolution kernels for dependency and constituent syntactic trees. In: ECML. Berlin, Germany (September 2006)
23. Moschitti, A.: State-of-the-art kernels for natural language processing. In: ACL (Tutorial Abstracts). p. 2. The Association for Computer Linguistics (2012)
24. Robert Müller, K., Mika, S., Rätsch, G., Tsuda, K., Schölkopf, B.: An introduction to kernel-based learning algorithms. IEEE Transactions on Neural Networks 12(2), 181–201 (2001)
25. Rychalska, B., Pakulska, K., Chodorowska, K., Walczak, W., Andruszkiewicz, P.: Samsung poland nlp team at semeval-2016 task 1: Necessity for diversity; combining recursive autoencoders, wordnet and ensemble methods to measure semantic similarity. In: Proceedings of SemEval-2016. pp. 614–620 (June 2016)
26. Sahlgren, M.: The Word-Space Model. Ph.D. thesis, Stockholm University (2006)
27. Schölkopf, B., Smola, A., Müller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput. 10(5), 1299–1319 (Jul 1998)
28. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, New York, NY, USA (2004)
29. Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of EMNLP '13 (2013)
30. Vapnik, V.N.: Statistical Learning Theory. Wiley-Interscience (1998)
31. Vishwanathan, S., Smola, A.J.: Fast kernels on strings and trees. In: Proceedings of Neural Information Processing Systems. pp. 569–576 (2002)
32. Williams, C.K.I., Seeger, M.: Using the nyström method to speed up kernel machines. In: Proceedings of NIPS 2000 (2001)