Distributed Smoothed Tree Kernel

Lorenzo Ferrone  
University of Rome “Tor Vergata”  
Via del Politecnico 1  
00133 Roma, Italy  
lorenzo.ferrone@gmail.com

Fabio Massimo Zanzotto  
University of Rome “Tor Vergata”  
Via del Politecnico 1  
00133 Roma, Italy  
fabio.massimo.zanzotto@uniroma2.it

Abstract

English. In this paper we explore the possibility to merge the world of Compositional Distributional Semantic Models (CDSM) with Tree Kernels (TK). In particular, we will introduce a specific tree kernel (smoothed tree kernel, or STK) and then show that is possible to approximate such kernel with the dot product of two vectors obtained compositionally from the sentences, creating in such a way a new CDSM.

Italiano. In questo paper vogliamo esplorare la possibilità di unire il mondo dei metodi di semantica distribuzionale composizionale (CDSM) con quello dei tree Kernel (TK). In particolare introdurremo un particolare tree kernel e poi mostreremo che possibile approssimare questo kernel tramite il prodotto scalare tra due vettori ottenuti composizionalmente a partire dalle frasi di partenza, creando così di fatto un nuovo modello di semantica distribzionale composizionale.

1 Introduction

Compositional distributional semantics is a flourishing research area that leverages distributional semantics (see Baroni and Lenci (2010)) to produce meaning of simple phrases and full sentences (hereafter called text fragments). The aim is to scale up the success of word-level relatedness detection to longer fragments of text. Determining similarity or relatedness among sentences is useful for many applications, such as multi-document summarization, recognizing textual entailment (Dagan et al., 2013), and semantic textual similarity detection (Agirre et al., 2013; Jurgens et al., 2014). Compositional distributional semantics models (CDSMs) are functions mapping text fragments to vectors (or higher-order tensors). Functions for simple phrases directly map distributional vectors of words to distributional vectors for the phrases (Mitchell and Lapata, 2008; Baroni and Zamparelli, 2010; Zanzotto et al., 2010). Functions for full sentences are generally defined as recursive functions over the ones for phrases (Socher et al., 2011). Distributional vectors for text fragments are then used as inner layers in neural networks, or to compute similarity among text fragments via dot product.

CDSMs generally exploit structured representations $t^x$ of text fragments $x$ to derive their meaning $f(t^x)$, but the structural information, although extremely important, is obfuscated in the final vectors. Structure and meaning can interact in unexpected ways when computing cosine similarity (or dot product) between vectors of two text fragments, as shown for full additive models in (Ferrone and Zanzotto, 2013).

Smoothed tree kernels (STK) (Croce et al., 2011) instead realize a clearer interaction between structural information and distributional meaning. STKs are specific realizations of convolution kernels (Haussler, 1999) where the similarity function is recursively (and, thus, compositionally) computed. Distributional vectors are used to represent word meaning in computing the similarity among nodes. STKs, however, are not considered part of the CDSMs family. As usual in kernel machines (Cristianini and Shawe-Taylor, 2000), STKs directly compute the similarity between two text fragments $x$ and $y$ over their tree representations $t^x$ and $t^y$, that is, $STK(t^x, t^y)$. The function $f$ that maps trees into vectors is
only implicitly used, and, thus, \( STK(t^x, t^y) \) is not explicitly expressed as the dot product or the cosine between \( f(t^x) \) and \( f(t^y) \).

Such a function \( f \), which is the underlying reproducing function of the kernel [Aronszajn, 1950], is a CDSM since it maps trees to vectors by using distributional meaning. However, the huge nality of \( \mathbb{R}^n \) (since it has to represent the set of all possible subtrees) prevents to actually compute the function \( f(t) \), which thus can only remain implicit.

Distributed tree kernels (DTK) [Zanzotto and Dell’Arciprete, 2012] partially solve the last problem. DTKs approximate standard tree kernels (such as (Collins and Duffy, 2002)) by defining an explicit function \( DT \) that maps trees to vectors in \( \mathbb{R}^m \) where \( m \ll n \) and \( \mathbb{R}^n \) is the explicit space for tree kernels. DTKs approximate standard tree kernels (TK), that is, \( \langle DT(t^x), DT(t^y) \rangle \approx TK(t^x, t^y) \), by approximating the corresponding reproducing function. Thus, these distributed trees are small vectors that encode structural information. In DTKs tree nodes \( u \) and \( v \) are represented by nearly orthonormal vectors, that is, vectors \( \vec{u} \) and \( \vec{v} \) such that \( \langle \vec{u}, \vec{v} \rangle \approx \delta(\vec{u}, \vec{v}) \) where \( \delta \) is the Kroneker’s delta. This is in contrast with distributional semantics vectors where \( \langle \vec{u}, \vec{v} \rangle \) is allowed to be any value in \([0,1]\) according to the similarity between the words \( v \) and \( u \).

In this paper, leveraging on distributed trees, we present a novel class of CDSMs that encode both structure and distributional meaning: the distributed smoothed trees (DST). DSTs carry structure and distributional meaning on a rank-2 tensor (a matrix): one dimension encodes the structure and one dimension encodes the meaning. By using DSTs to compute the similarity among sentences with a generalized dot product (or cosine), we implicitly define the distributed smoothed tree kernels (DSTK) which approximate the corresponding STKs. We present two DSTs along with the two smoothed tree kernels (STKs) that they approximate. We experiment with our DSTs to show that their generalized dot products approximate STKs by directly comparing the produced similarities and by comparing their performances on two tasks: recognizing textual entailment (RTE) and semantic similarity detection (STS). Both experiments show that the dot product on DSTs approximates STKs and, thus, DSTs encode both structural and distributional semantics of text fragments in tractable rank-2 tensors. Experiments on STS and RTE show that distributional semantics encoded in DSTs increases performance over structure-only kernels. DSTs are the first positive way of taking into account both structure and distributional meaning in CDSMs. The rest of the paper is organized as follows. Section 2.1 introduces the basic notation used in the paper. Section 2 describe our distributed smoothed trees as compositional distributional semantic models that can represent both structural and semantic information. Section 4 reports on the experiments. Finally, Section 5 draws some conclusions.

2 Distributed Smoothed Tree Kernel

We here propose a model that can be considered a compositional distributional semantic model as it transforms sentences into matrices that can then be used by the learner as feature vectors. Our model is called Distributed Smoothed Tree Kernel [Ferrone and Zanzotto, 2014] as it mixes the distributed trees (Zanzotto and Dell’Arciprete, 2012) representing syntactic information with distributional semantic vectors representing semantic information.

Figure 1: A lexicalized tree

2.1 Notation

Before describing the distributed smoothed trees (DST) we introduce a formal way to denote constituency-based lexicalized parse trees, as DSTs exploit this kind of data structures. Lexicalized trees are denoted with the letter \( t \) and \( N(t) \) denotes the set of non terminal nodes of tree \( t \). Each non-terminal node \( n \in N(t) \)
has a label $l_n$ composed of two parts $l_n = (s_n, w_n)$: $s_n$ is the syntactic label, while $w_n$ is the semantic headword of the tree headed by $n$, along with its part-of-speech tag. Terminal nodes of trees are treated differently, these nodes represent only words $w_n$ without any additional information, and their labels thus only consist of the word itself (see Fig 1). The structure of a DST is represented as follows: Given a tree $t$, $h(t)$ is its root node and $s(t)$ is the tree formed from $t$ but considering only the syntactic structure (that is, only the $s_n$ part of the labels), $c_i(n)$ denotes $i$-th child of a node $n$. As usual for constituency-based parse trees, pre-terminal nodes are nodes that have a single terminal node as child.

Finally, we use $w_n \rightarrow \in \mathbb{R}^k$ to denote the distributional vector for word $w_n$.

2.2 The method at a glance

We describe the approach in a few sentences. In line with tree kernels over structures (Collins and Duffy, 2002), we introduce the set $S(t)$ of the subtrees $t_i$ of a given lexicalized tree $t$. A subtree $t_i$ is in the set $S(t)$ if $s(t_i)$ is a subtree of $s(t)$ and, if $n$ is a node in $t_i$, all the siblings of $n$ in $t$ are in $t_i$. For each node of $t_i$ we only consider its syntactic label $s_n$, except for the head $h(t_i)$ for which we also consider its semantic component $w_n$ (see Fig 2). The functions DSTs we define compute the following:

$$DST(t) = T = \sum_{t_i \in S(t)} T_i$$

where $T_i$ is the matrix associated to each subtree $t_i$. The similarity between two text fragments $a$ and $b$ represented as lexicalized trees $t^a$ and $t^b$, can be computed using the Frobenius product between the two matrices $T^a$ and $T^b$, that is:

$$\langle T^a, T^b \rangle_F = \sum_{t_i^a \in S(t^a)} \sum_{t_j^b \in S(t^b)} \langle T_i^a, T_j^b \rangle_F$$

We want to obtain that the product $\langle T_i^a, T_j^b \rangle_F$ approximates the dot product between the distributional vectors of the head words $(\langle h(t_i^a), h(t_j^b) \rangle_{F})$ whenever the syntactic structure of the subtrees is the same (that is $s(t_i^a) = s(t_j^b)$), and $\langle T_i^a, T_j^b \rangle_F \approx 0$ otherwise. This property is expressed as:

$$\langle T_i^a, T_j^b \rangle_F \approx \delta(s(t_i^a), s(t_j^b)) \cdot \langle h(t_i^a), h(t_j^b) \rangle$$

To obtain the above property, we define

$$T_i = s(t_i)w_{h(t_i)} \rightarrow$$

where $s(t_i)$ are distributed tree fragment (Zanzotto and Dell'Arciprete, 2012) for the subtree $t_i$ and $w_{h(t_i)}$ is the distributional vector of the head of the subtree $t_i$.

Distributed tree fragments have the property that $s(t_i)s(t_j) \approx \delta(t_i, t_j)$. Thus, exploiting the fact that: $\langle a \rightarrow w_\rightarrow b v \rightarrow \rangle_F = \langle a, b \rangle \cdot \langle w, v \rangle$, we have that Equation[2] is satisfied as:

$$\langle T_i, T_j \rangle_F = \langle s(t_i), s(t_j) \rangle \cdot \langle w_{h(t_i)}, w_{h(t_j)} \rangle$$

$$\approx \delta(s(t_i), s(t_j)) \cdot \langle w_{h(t_i)}, w_{h(t_j)} \rangle$$

It is possible to show that the overall compositional distributional model $DST(t)$ can be obtained with a recursive algorithm that exploits vectors of the nodes of the tree.

3 The Approximated Smoothed Tree Kernels

The CDSM we proposed approximates a specific tree kernel belonging to the smoothed tree kernels class. This recursively computes (but, the recursive formulation is not given here) the following general equation:

$$STK(t^a, t^b) = \sum_{t_i \in S(t^a)} \sum_{t_j \in S(t^b)} \omega(t_i, t_j)$$
\[ \omega(t_i, t_j) = \alpha \cdot \langle w_{h(t_i)}, w_{h(t_j)} \rangle \cdot \delta(s(t_i), s(t_j)) \]

Where \( \alpha = \sqrt{\lambda |N(t_i)| + |N(t_j)|} \) and \( \lambda \) is a parameter.

4 Experimental investigation

Generic settings We experimented with two datasets: the Recognizing Textual Entailment datasets (RTE) \cite{Dagan2006} and the the Semantic Textual Similarity 2013 datasets (STS) \cite{Agirre2013}. The STS task consists of determining the degree of similarity (ranging from 0 to 5) between two sentences. The STS datasets contains 5 datasets: headlines, OnWN, FNWN and SMT which contains respectively 750, 561, 189 and 750 RTE is instead the task of deciding whether a long text \( T \) entails a shorter text, typically a single sentence, called hypothesis \( H \). It has been often seen as a classification task. We used four datasets: RTE1, RTE2, RTE3, and RTE5. We parsed the sentence with the Stanford Parser \cite{Klein2003} and extracted the heads for use in the lexicalized trees with Collins’ rules \cite{Collins2003}. Distributional vectors are derived with DISSECT \cite{Dinu2013} from a corpus obtained by the concatenation of ukWaC, a mid-2009 dump of the English Wikipedia and the British National Corpus for a total of about 2.8 billion words. The raw count vectors were transformed into positive Pointwise Mutual Information scores and reduced to 300 dimensions by Singular Value Decomposition. This setup was picked without tuning, as we found it effective in previous, unrelated experiments. To build our DTSKs we used the implementation of the distributed tree kernels\cite{STK}. We used 1024 and 2048 as the dimension of the distributed vectors, the weight \( \lambda \) is set to 0.4 as it is a value generally considered optimal for many applications \cite{Zanzotto2012}. To test the quality of the approximation we computed the Spearman’s correlation between values produced by our DSTK and by the standard versions of the smoothed tree kernel. We obtained text fragment pairs by randomly sampling two text fragments in the selected set. For each set, we produced exactly the number of examples in the set, e.g., we produced 567 pairs for RTE1, etc.

Table 1 reports the results for the correlation experiments. We report the Spearman’s correlations over the different sets (and different dimensions of distributed vectors) between our DSTK and the STK. The correlation is above 0.80 in average for both RTE and STS datasets. The approximation also depends on the size of the distributed vectors. Higher dimensions yield to better approximation: if we increase the distributed vectors dimension from 1024 to 2048 the correlation between DSTK and STK increases. This direct analysis of the correlation shows that our CDSM are approximating the corresponding kernel function and there is room of improvement by increasing the size of distributed vectors.

<table>
<thead>
<tr>
<th>STK vs DSTK</th>
<th>RTE1</th>
<th>RTE2</th>
<th>RTE3</th>
<th>RTE5</th>
<th>headl</th>
<th>FNWN</th>
<th>OnWN</th>
<th>SMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1024</td>
<td>0.86</td>
<td>0.84</td>
<td>0.90</td>
<td>0.84</td>
<td>0.87</td>
<td>0.65</td>
<td>0.95</td>
<td>0.77</td>
</tr>
<tr>
<td>2048</td>
<td>0.87</td>
<td>0.84</td>
<td>0.91</td>
<td>0.84</td>
<td>0.90</td>
<td>0.65</td>
<td>0.96</td>
<td>0.77</td>
</tr>
</tbody>
</table>

**Results** Table 1 reports the results for the correlation experiments. We report the Spearman’s correlations over the different sets (and different dimensions of distributed vectors) between our DSTK and the STK. The correlation is above 0.80 in average for both RTE and STS datasets. The approximation also depends on the size of the distributed vectors. Higher dimensions yield to better approximation: if we increase the distributed vectors dimension from 1024 to 2048 the correlation between DSTK and STK increases. This direct analysis of the correlation shows that our CDSM are approximating the corresponding kernel function and there is room of improvement by increasing the size of distributed vectors.

5 Conclusions and future work

Distributed Smoothed Trees (DST) are a novel class of Compositional Distributional Semantics Models (CDSM) that effectively encode structural information and distributional semantics in tractable rank-2 tensors, as experiments show. The paper shows that DSTs contribute to close the gap between two apparently different approaches: CDSMs and convolution kernels. This contribute to start a discussion on a deeper understanding of the representation power of structural information of existing CDSMs.
References


