Compositional Distributional Semantics Models in Chunk-based Smoothed Tree Kernels

Nghia The Pham
University of Trento
thenghia.pham@unitn.it

Lorenzo Ferrone
University of Rome "Tor Vergata"
lorenzo.ferrone@gmail.com

Fabio Massimo Zanzotto
University of Rome "Tor Vergata"
fabio.massimo.zanzotto@uniroma2.it

Abstract

The field of compositional distributional semantics has proposed very interesting and reliable models for accounting the distributional meaning of simple phrases. These models however tend to disregard the syntactic structures when they are applied to larger sentences. In this paper we propose the *chunk-based smoothed tree kernels* (CSTKs) as a way to exploit the syntactic structures as well as the reliability of these compositional models for simple phrases. We experiment with the recognizing textual entailment datasets. Our experiments show that our CSTKs perform better than basic compositional distributional semantic models (CDSMs) recursively applied at the sentence level, and also better than syntactic tree kernels.

1 Introduction

A clear interaction between syntactic and semantic interpretations for sentences is important for many high-level NLP tasks, such as question-answering, textual entailment recognition, and semantic textual similarity. Systems and models for these tasks often use classifiers or regressors that exploit convolution kernels (Haussler, 1999) to model both interpretations.

Convolution kernels are naturally defined on spaces where there exists a similarity function between terminal nodes. This feature has been used to integrate distributional semantics within tree kernels. This class of kernels is often referred to as *smoothed tree kernels* (Mehdad et al., 2010; Croce et al., 2011), yet, these models only use distributional vectors for words.

Compositional distributional semantics models (CDSMs) on the other hand are functions mapping text fragments to vectors (or higher-order tensors) which then provide a distributional meaning for simple phrases or sentences. Many CDSMs have been proposed for simple phrases like non-recursive noun phrases or verbal phrases (Mitchell and Lapata, 2008; Baroni and Zamparelli, 2010; Clark et al., 2008; Grefenstette and Sadrzadeh, 2011; Zanzotto et al., 2010). Non-recursive phrases are often referred to as chunks (Abney, 1996), and thus, CDSMs are good and reliable models for chunks.

In this paper, we present the *chunk-based smoothed tree kernels* (CSTK) as a way to merge the two approaches: the smoothed tree kernels and the models for compositional distributional semantics. Our approach overcomes the limitation of the smoothed tree kernels which only use vectors for words by exploiting reliable CDSMs over chunks. CSTKs are defined over a chunk-based syntactic subtrees where terminal nodes are words or word sequences. We experimented with CSTKs on data from the recognizing textual entailment challenge (Dagan et al., 2006) and we compared our CSTKs with other standard tree kernels and standard recursive CDSMs. Experiments show that our CSTKs perform better than basic compositional distributional semantic models (CDSMs) recursively applied at the sentence level and better than syntactic tree kernels.

The rest of the paper is organized as follows. Section 2 describes the CSTKs. Section 3 reports on the experimental setting and on the results. Finally, Section 4 draws the conclusions and sketches the future work.

2 Chunk-based Smoothed Tree Kernels

This section describes the new class of kernels. We first introduce the notion of the chunk-based syntactic subtree. Then, we describe the recursive formulation of the class of kernels. Finally, we introduce the basic CDSMs we use and we introduce two instances of the class of kernels.

*Proceedings of the Third Joint Conference on Lexical and Computational Semantics (*SEM 2014)*, pages 93–98,
Dublin, Ireland, August 23-24 2014.

2.1 Notation and preliminaries

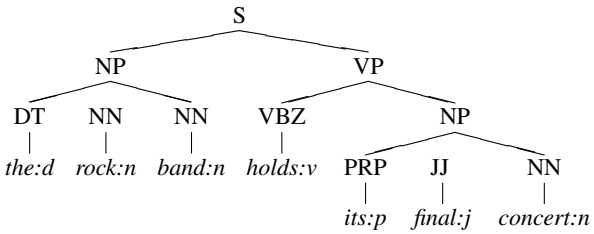


Figure 1: Sample Syntactic Tree

A *Chunk-based Syntactic Sub-Tree* is a subtree of a syntactic tree where each non-terminal node dominating a contiguous word sequence is collapsed into a chunk and, as usual in chunks (Abney, 1996), the internal structure is disregarded. For example, Figure 2 reports some chunk-based syntactic subtrees of the tree in Figure 1. Chunks are represented with a pre-terminal node dominating a triangle that covers a word sequence. The first subtree represents the chunk covering the second NP and the node dominates the word sequence *its:d final:n concert:n*. The second subtree represents the structure of the whole sentence and one chunk, that is the first NP dominating the word sequence *the:d rock:n band:n*. The third subtree again represents the structure of the whole sentence split into two chunks without the verb.

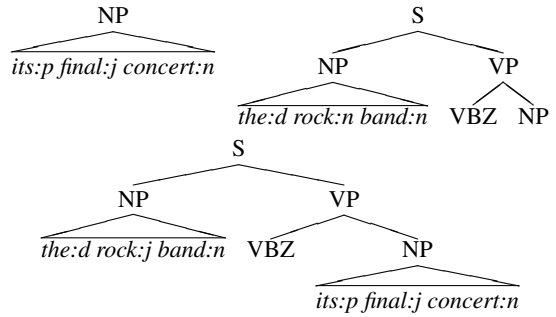


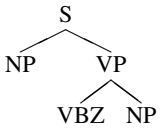
Figure 2: Some Chunk-based Syntactic Sub-Trees of the tree in Figure 1

In the following sections, generic 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 s_n representing its syntactic tag. As usual for constituency-based parse trees, pre-terminal nodes are nodes that have a single terminal node as child. Terminal nodes of trees are words denoted with *w:pos* where w is the actual token and pos is its postag. The structure of these trees is represented as follows. Given a tree t, $c_i(n)$ denotes i-th child of a node n in the set of nodes $N(t)$. The production rule headed in node n is $\mathsf{prod}(n)$, that is, given the node n with m children, $\mathsf{prod}(n)$ is:

$$\mathsf{prod}(n) = s_n \rightarrow s_{c_1(n)} \dots s_{c_m(n)}$$

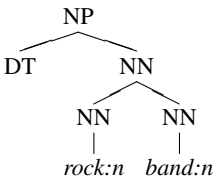
Finally, for a node n in $N(t)$, the function $d(n)$ generates the word sequence dominated by the non-terminal node n in the tree t. For example, *d(VP)* in Figure 1 is *holds:v its:p final:j concert:n*.

Chunk-based Syntactic Sub-Trees (CSSTs) are instead denoted with the letter τ. Differently from trees t, CSSTs have terminal nodes that can represent subsequences of words of the original sentence. The explicit syntactic structure of a CSST is the structure not falling in chunks and it is represented as $\mathsf{s}(\tau)$. For example, $\mathsf{s}(\tau_3)$ is:

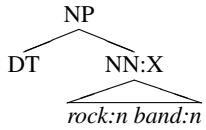


where τ_3 is the third subtree of Figure 2.

Given a tree t, the set $\mathcal{S}(t)$ is defined as the set containing all the relevant CSSTs of the tree t. As for the tree kernels (Collins and Duffy, 2002), the set $\mathcal{S}(t)$ contains all CSSTs derived from the subtrees of t such that if a node n belongs to a subtree t_s, all the siblings of n in t belongs to t_s. In other words, productions of the initial subtrees are complete. A CSST is obtained by collapsing in a single terminal nodes a contiguous sequence of words dominated by a single non-terminal node. For example:



is collapsed into:



Finally, $\vec{w_n} \in \mathbb{R}^m$ represent the *distributional* vectors for words w_n and $f(w_1 \dots w_k)$ represents a compositional distributional semantics model applied to the word sequence $w_1 \dots w_k$.

2.2 Smoothed Tree Kernels on Chunk-based Syntactic Trees

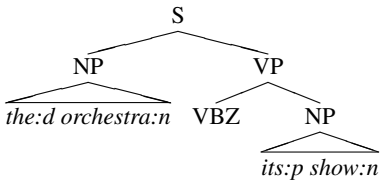
As usual, a tree kernel, although written in a recursive way, computes the following general equation:

$$K(t_1, t_2) = \sum_{\substack{\tau_i \in \mathcal{S}(t_1) \\ \tau_j \in \mathcal{S}(t_2)}} \lambda^{|N(\tau_i)|+|N(\tau_j)|} K_F(\tau_i, \tau_j) \quad (1)$$

In our case, the basic similarity $K_F(t_i, t_j)$ is defined to take into account the syntactic structure and the distributional semantic part. Thus, we define it as follows in line with what done with several other smoothed tree kernels:

$$K_F(\tau_i, \tau_j) = \delta(\mathsf{s}(\tau_i), \mathsf{s}(\tau_j)) \prod_{\substack{a \in PT(\tau_i) \\ b \in PT(\tau_j)}} \langle f(a), f(b) \rangle$$

where $\delta(\mathsf{s}(\tau_i), \mathsf{s}(\tau_j))$ is the Kroneker's delta function between the the structural part of two chunk-based syntactic subtrees, $PT(\tau)$ are the nodes in τ directly covering a chunk or a word, and $\langle \vec{x}, \vec{y} \rangle$ is the cosine similarity between the two vectors $\vec{x}$ and $\vec{y}$. For example, given the chunk-based subtree τ_3 in Figure 2 and

$\tau_4 =$ 

the similarity $K_F(\tau_3, \tau_4)$ is: $\langle f(\textit{the:d orchestra:n}), f(\textit{the:d rock:n band:n}) \rangle \cdot \langle f(\textit{its:p show:n}), f(\textit{its:p final:j concert:n}) \rangle$.

The recursive formulation of the Chunk-based Smoothed Tree Kernel (CSTK) is a bit more complex but very similar to the recursive formulation of the syntactic tree kernels:

$$K(t_1, t_2) = \sum_{\substack{n_1 \in N(t_1) \\ n_2 \in N(t_2)}} C(n_1, n_2) \quad (2)$$

where $C(n_1, n_2) =$

$$\begin{cases} \langle f(d(n_1)), f(d(n_2)) \rangle \text{ if } label(n_1) = label(n_2) \\ \quad \text{and } \mathsf{prod}(n_1) \neq \mathsf{prod}(n_2) \\ \langle f(d(n_1)), f(d(n_2)) \rangle \\ \quad + \prod_{j=1}^{nc(n_1)} (1 + C(c_j(n_1), c_j(n_2))) \\ \quad - \prod_{j=1}^{nc(n_1)} \langle f(d(c_j(n_1))), f(d(c_j(n_2))) \rangle \\ \quad \text{if } n_1, n_2 \text{ are not pre-terminals and} \\ \quad \mathsf{prod}(n_1) = \mathsf{prod}(n_2) \\ 0 \text{ otherwise} \end{cases}$$

where $nc(n_1)$ is the lenght of the production $\mathsf{prod}(n_1)$.

2.3 Compositional Distributional Semantic Models and two Specific CSTKs

To define specific CSTKs, we need to introduce the basic compositional distributional semantic models (CDSMs). We use two CDSMs: the Basic Additive model (BA) and teh Full Additive model (FA). We thus define two specific CSTKs: the CSTK+BA that is based on the basic additive model and the CSTK+FA that is based on the full additive model. We describe the two CDSMs in the following.

The Basic Additive model (BA) (introduced in (Mitchell and Lapata, 2008)) computes the distibutional semantics vector of a pair of words $a = a_1 a_2$ as:

$$ADD(a_1, a_2) = \alpha \vec{a_1} + \beta \vec{a_2}$$

where α and β weight the first and the second word of the pair. The basic additive model for word sequences $s = w_1 \dots w_k$ is recursively defined as follows:

$$f_{BA}(s) = \begin{cases} \vec{w_1} & \text{if } k = 1 \\ \alpha \vec{w_1} + \beta f_{BA}(w_2 \dots w_k) & \text{if } k > 1 \end{cases}$$

The Full Additive model (FA) (used in (Guevara, 2010) for adjective-noun pairs and (Zanzotto et al., 2010) for three different syntactic relations) computes the compositional vector $\vec{a}$ of a pair using two linear tranformations A_R and B_R respectively applied to the vectors of the first and the second word. These matrices generally only depends on the syntactic relation R that links those two words. The operation follows:

$$f_{FA}(a_1, a_2, R) = A_R \vec{a_1} + B_R \vec{a_2}$$

	RR					RRTWS				
	RTE1	**RTE2**	**RTE3**	**RTE5**	**Average**	**RTE1**	**RTE2**	**RTE3**	**RTE5**	**Average**
Add	0.541	0.496	0.507	0.520	0.516	0.560	0.538	0.643	0.578	0.579
FullAdd	0.512	0.516	0.507	0.569	0.526	0.571	0.608	0.643	0.643	0.616
TK	0.561	0.552	0.531	0.54	0.546	0.608	0.627	0.648	0.630	0.628
CSTK+BA	0.553	0.545	0.562	0.568	$0.557^{\dagger}$	0.626	0.616	0.648	0.628	$0.629^{\dagger}$
CSTK+FA	0.543	0.550	0.574	0.576	$\mathbf{0.560}^{\dagger}$	0.628	0.616	0.652	0.630	$\mathbf{0.631}^{\dagger}$

Table 1: Task-based analysis: Accuracy on Recognizing Textual Entailment († is different from both ADD and FullADD with a stat.sig. of $p > 0.1$.)

The full additive model for word sequences $s = w_1 \dots w_k$, whose node has a production rule $s \rightarrow s_{c_1} \dots s_{c_m}$ is also defined recursively:

$$f_{FA}(s) = \begin{cases} \overrightarrow{w_1} & \text{if } k = 1 \\ A_{vn}\overrightarrow{V} + B_{vn} f_{FA}(NP) & \\ \quad \text{if } s \rightarrow V\ NP & \\ A_{an}\overrightarrow{A} + B_{an} f_{FA}(N) & \\ \quad \text{if } s \rightarrow A\ N & \\ \sum f_{FA}(s_{c_i}) & \text{otherwise} \end{cases}$$

where A_{vn}, B_{vn} are matrices used for verb and noun phrase interaction, and A_{an}, B_{an} are used for adjective, noun interaction.

3 Experimental Investigation

3.1 Experimental set-up

We experimented with the Recognizing Textual Entailment datasets (RTE) (Dagan et al., 2006). RTE is 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 (see (Dagan et al., 2013)). We used four datasets: RTE1, RTE2, RTE3, and RTE5, with the standard split between training and testing. The dev/test distribution for RTE1-3, and RTE5 is respectively 567/800, 800/800, 800/800, and 600/600 T-H pairs.

Distributional vectors are derived with DISSECT (Dinu et al., 2013) from a corpus obtained by the concatenation of ukWaC (wacky.sslmit.unibo.it), a mid-2009 dump of the English Wikipedia (en.wikipedia.org) and the British National Corpus (www.natcorp.ox.ac.uk), for a total of about 2.8 billion words. We collected a 35K-by-35K matrix by counting co-occurrence of the 30K most frequent content lemmas in the corpus (nouns, adjectives and verbs) and all the content lemmas occurring in the datasets within a 3 word window. 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.

We built the matrices for the full additive models using the procedure described in (Guevara, 2010). We considered only two relations: the Adjective-Noun and Verb-Noun. The full additive model falls back to the basic additional model when syntactic relations are different from these two.

To build the final kernel to learn the classifier, we followed standard approaches (Dagan et al., 2013), that is, we exploited two models: a model with only a rewrite rule feature space (RR) and a model with the previous space along with a token-level similarity feature (RRTWS). The two models use our CSTKs and the standard TKs in the following way as kernel functions: (1) $RR(p_1, p_2) = \kappa(t_1^a, t_2^a) + \kappa(t_1^b, t_2^b)$; (2) $RRTWS(p_1, p_2) = \kappa(t_1^a, t_2^a) + \kappa(t_1^b, t_2^b) + (TWS(a_1, b_1) \cdot TWS(a_2, b_2) + 1)^2$ where TWS is a weighted token similarity (as in (Corley and Mihalcea, 2005)).

3.2 Results

Table 1 shows the results of the experiments, the table is organised as follows: columns 2-6 report the accuracy of the RTE systems based on rewrite rules (RR) and columns 7-11 report the accuracies of RR systems along with token similarity (RRTS). We compare five differente models: ADD is the Basic Additive model with parameters $\alpha = \beta = 1$ (as defined in 2.3) applied to the words of the sentence (without considering its tree structure), the same is done for the Full Additive (FullADD), defined as in 2.3. The Tree Kernel (TK) as defined in (Collins and Duffy, 2002) are applied to

the constituency-based tree representation of the tree, without the intervening collapsing step described in 2.2. These three models are the baseline against which we compare the CSTK models where the collapsing procedure is done via Basic Additive (CSTK + BA, again with $\alpha = \beta = 1$) and FullAdditive (CSTK + FA), as described in section 2.2, again, with the aforementioned restriction on the relation considered. For RR models we have that CSTK+BA and CSTK+FA both achieve higher accuracy than ADD and FullAdd, with a statistical significante greater than 93.7%, as computed with the sign test. Specifically we have that CSTK+BA has an average accuracy 7.94% higher than ADD and 5.89% higher than FullADD, while CSTK+FA improves on ADD and FullADD by 8.52% and 6.46%, respectively. The same trend is visible for the RRTS model, again both models are statistically better than ADD and FullADD, in this case we have that CSTK+BA is 8.63% more accurate then ADD and 2.11% more than FullADD, CSTK+FA is respectively 8.98% and 2.43% more accurate than ADD and FullADD. As for the TK models we have that both CSTK models achieve again an higher average accuracy: for RR models CSTK+BA and CSTK+FA are respectively 2.01% and 0.15% better than TK, while for RRTS models the number are 2.54% and 0.47%. These results though are not statistically significant, as is the difference between the two CSTK models themselves.

4 Conclusions and Future Work

In this paper, we introduced a novel sub-class of the convolution kernels in order exploit reliable compositional distributional semantic models along with the syntactic structure of sentences. Experiments show that this novel subclass, namely, the Chunk-based Smoothed Tree Kernels (CSTKs), are a promising solution, performing significantly better than a naive recursive application of the compositional distributional semantic models. We experimented with CSTKS equipped with the basic additive and the full additive CDSMs but these kernels are definitely open to all the CDSMs.

Acknowledgments

We acknowledge ERC 2011 Starting Independent Research Grant n. 283554 (COMPOSES).

References

Steven Abney. 1996. Part-of-speech tagging and partial parsing. In G.Bloothooft K.Church, S.Young, editor, *Corpus-based methods in language and speech*. Kluwer academic publishers, Dordrecht.

Marco Baroni and Roberto Zamparelli. 2010. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1183–1193, Cambridge, MA, October. Association for Computational Linguistics.

Stephen Clark, Bob Coecke, and Mehrnoosh Sadrzadeh. 2008. A compositional distributional model of meaning. *Proceedings of the Second Symposium on Quantum Interaction (QI-2008)*, pages 133–140.

Michael Collins and Nigel Duffy. 2002. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In *Proceedings of ACL02*.

Courtney Corley and Rada Mihalcea. 2005. Measuring the semantic similarity of texts. In *Proc. of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 13–18. Association for Computational Linguistics, Ann Arbor, Michigan, June.

Danilo Croce, Alessandro Moschitti, and Roberto Basili. 2011. Structured lexical similarity via convolution kernels on dependency trees. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1034–1046, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In Quionero-Candela et al., editor, *LNAI 3944: MLCW 2005*, pages 177–190. Springer-Verlag, Milan, Italy.

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. 2013. *Recognizing Textual Entailment: Models and Applications*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Georgiana Dinu, Nghia The Pham, and Marco Baroni. 2013. DISSECT: DIStributional SEmantics Composition Toolkit. In *Proceedings of ACL (System Demonstrations)*, pages 31–36, Sofia, Bulgaria.

Edward Grefenstette and Mehrnoosh Sadrzadeh. 2011. Experimental support for a categorical compositional distributional model of meaning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1394–1404, Stroudsburg, PA, USA. Association for Computational Linguistics.

Emiliano Guevara. 2010. A regression model of adjective-noun compositionality in distributional semantics. In *Proceedings of the 2010 Workshop on GEometrical Models of Natural Language Semantics*, pages 33–37, Uppsala, Sweden, July. Association for Computational Linguistics.

David Haussler. 1999. Convolution kernels on discrete structures. Technical report, University of California at Santa Cruz.

Yashar Mehdad, Alessandro Moschitti, and Fabio Massimo Zanzotto. 2010. Syntactic/semantic structures for textual entailment recognition. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 1020–1028, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of ACL-08: HLT*, pages 236–244, Columbus, Ohio, June. Association for Computational Linguistics.

Fabio Massimo Zanzotto, Ioannis Korkontzelos, Francesca Fallucchi, and Suresh Manandhar. 2010. Estimating linear models for compositional distributional semantics. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, August,.