Distributed Structures and Distributional Meaning

Fabio Massimo Zanzotto
DISP University of Rome "Tor Vergata"
Via del Politecnico 1
00133 Roma, Italy
zanzotto@info.uniroma2.it

Lorenzo Dell'Arciprete
University of Rome "Tor Vergata"
Via del Politecnico 1
00133 Roma, Italy
lorenzo.dellarciprete@gmail.com

Abstract

Stemming from distributed representation theories, we investigate the interaction between distributed structure and distributional meaning. We propose a pure distributed tree (DT) and distributional distributed tree (DDT). DTs and DDTs are exploited for defining distributed tree kernels (DTKs) and distributional distributed tree kernels (DDTKs). We compare DTKs and DDTKs in two tasks: approximating tree kernels TK (Collins and Duffy, 2002); performing textual entailment recognition (RTE). Results show that DTKs correlate with TKs and perform in RTE better than DDTKs. Then, including distributional vectors in distributed structures is a very difficult task.

1 Introduction

Demonstrating that distributional semantics is a semantic model of natural language is a real research challenge in natural language processing. Frege's principle of compositionality (Frege, 1884), naturally taken into account in logic-based semantic models of natural language (Montague, 1974), is hardly effectively included in distributional semantics models. These models should compositionally derive distributional vectors for sentences and phrases from the distributional vectors of the composing words.

Besides vector averaging (Landauer and Dumais, 1997; Foltz et al., 1998), that can model distributional meaning of sentences, recent distributional compositional models focus on finding distributional vectors of word pairs (Mitchell and Lapata, 2010; Guevara, 2010; Baroni and Zamparelli, 2010; Zanzotto et al., 2010). Scaling up these 2-word sequence models to the sentence level is not trivial as syntactic structure of sentences plays a very important role. Understanding the relation between the structure and the meaning is needed for building distributional compositional models for sentences.

Research in Distributed Representations (DR) (Hinton et al., 1986) proposed models and methods for encoding data structures in vectors, matrices, or high-order tensors. Distributed Representations are oriented to preserve the structural information in the final representation. For this purpose, DR models generally use random and possibly orthogonal vectors for words and structural elements (Plate, 1994). As distributional semantics vectors are unlikely to be orthogonal, syntactic structure of sentences may be easily lost in the final vector combination.

In this paper, we investigate the interaction between distributed structure and distributional meaning by proposing a model to encode syntactic trees in distributed structures and by exploiting this model in kernel machines (Vapnik, 1995) to determine the similarity between syntactic trees. We propose a pure distributed tree (DT) and a distributional distributed tree (DDT). In line with the distributed representation theory, DTs use random vectors for representing words whereas DDTs use distributional vectors for words. Our interest is in understanding if the introduction of distributional semantic information in an inherently syntactic based model, such as distributed representations, leads to better performances in semantic aware tasks. DTs and DDTs are exploited for defining distributed tree ker-

Proceedings of the Workshop on Distributional Semantics and Compositionality (DiSCo'2011), pages 10–15,

Portland, Oregon, 24 June 2011.

nels (DTKs) and distributional distributed tree kernels (DDTKs). We study the interaction between structure and meaning in two ways: 1) by comparing DTKs and DDTKs with the classical tree similarity functions, i.e., the tree kernels TK (Collins and Duffy, 2002); 2) by comparing the accuracy of DTKs and DDTKs in a semantic task such as recognizing textual entailment (RTE). Results show that DTKs correlate with TKs and perform in RTE better than DDTKs. This indicates that including distributional vectors in distributed structures should be performed in a more complex fashion.

2 Related Work

Distributed Representations (DR) (Hinton et al., 1986) are models and methods for encoding data structures as trees in vectors, matrices, or high-order tensors. DR are studied in opposition to symbolic representations to describe how knowledge is treated in connectionist models (Rumelhart and Mcclelland, 1986). Basic symbolic elements, e.g., *John* or *car*, as well as eventually nested structures, e.g., *buy(John,car,in(1978))*, are represented as vectors, matrices, or higher order tensors. Vectors of basic elements (words, or concepts) can be randomly generated (e.g. (Anderson, 1973; Murdock, 1983)) or, instead, they may represent their attributes and can be manually built (e.g. (McRae et al., 1997; Andrews et al., 2009)). Vectors, matrices, or tensors for structures are compositionally derived using vectors for basic elements.

Good compositionally obtained vectors for structures are *explicit and immediately accessible*: information stored in a distributed representation should be easily accessible with simple operations (Plate, 1994). Circular convolution in Holographic Reduced Representations (HRRs) (Plate, 1994) is designed to satisfy the immediate accessibility property. It supports two operations for producing and accessing the compact representations: the circular convolution and the correlation. Given that component vectors are obtained randomly (as in (Anderson, 1973; Murdock, 1983)), correlation is the inverse of composition. Yet, distributed representations offer an informative way of encoding structures if basic vectors are nearly orthogonal.

3 Distributed Trees and Distributional Distributed Trees

Stemming from distributed representations, we propose a way to encode syntactic trees in distributed vectors. These vectors can be pure distributed tree vectors (DT) or distributional distributed tree vectors (DDT). Once defined, these vectors can be used as a tree similarity function in kernel machines (Vapnik, 1995). We can build pure distributed tree kernels (DTK) or distributional distributed tree kernels (DDTK) to be used in recognizing textual entailment (RTE).

The rest of the section is organized as follows. We firstly present the distributed trees and the distributed tree kernels (Sec. 3.1). We then describe how to obtain DTs and DDTs (Sec. 3.2). Finally, we describe how the related kernels can be used for the recognizing textual entailment task (Sec. 3.2.1).

3.1 Distributed Trees and Distributed Tree Kernels

We define a distributed vector in order to finally produce a similarity function between trees (i.e., a kernel function) as the classical tree kernel (Collins and Duffy, 2002). A distributed vector $\vec{\tilde{T}}$ is a vector representing the subtrees of a tree T. The final function is:

$$\vec{\tilde{T}} = \sum_{n \in N(T)} s(n) \quad (1)$$

where $N(T)$ is the set of nodes of the tree T, n is a node, and $s(n)$ is the sum of the distributed vectors of the subtrees of T rooted in the node n. The function $s(n)$ is recursively defined as follows:

- $s(n) = \vec{n} \otimes \vec{w}$ if n is a pre-terminal node $n \rightarrow w$ where $\vec{n}$ is the vector representing n and $\vec{w}$ is the one representing the word w.
- $s(n) = \vec{n} \otimes (\vec{c_1} + s(c_1)) \otimes \ldots \otimes (\vec{c_n} + s(c_n))$ where n is not a pre-terminal node, $n \rightarrow c_1 \ldots c_n$ is the first production of the tree rooted in n, $\vec{n}$ is the vector of the node n, and $\vec{c_i}$ are the vectors of the nodes c_i.

The distributed vectors of the nodes only depend on tags of the nodes.

The function $\otimes$ is defined as the reverse element-wise product $\vec{v} = \vec{a} \otimes \vec{b}$ as:

$$v_i = \gamma a_i b_{n-i+1} \quad (2)$$

where v_i, a_i, and b_i are the elements of, respectively, the vectors $\vec{v}$, $\vec{a}$, and $\vec{b}$; n is the dimension of the space; and γ is a value to ensure that the operation $\otimes$ approximate the property of vector module preservation. This function is not commutative and this guarantees that different trees t have different vectors $\vec{t}$. It is possible to demonstrate that:

$$\vec{\tilde{T}} = \sum_{t \in S(T)} \vec{t} \quad (3)$$

where $S(T)$ is the set of the subtrees of T, t is one of its subtrees, and $\vec{t}$ is its distributed representation.

The distributed kernel $\widetilde{TK}$ function over trees then easily follows as:

$$\widetilde{TK}(T_1, T_2) = \vec{\tilde{T}_1} \cdot \vec{\tilde{T}_2} = \sum_{t_1 \in S(T_1)} \sum_{t_2 \in S(T_2)} \vec{t_1} \cdot \vec{t_2} \quad (4)$$

If the different trees are orthogonal, $\widetilde{TK}(T_1, T_2)$ counts approximately the number of subtrees in common between the two trees T_1 and T_2.

3.2 Pure Distributed vs. Distributional Distributed Trees

For producing the distributed trees, we use basic random vectors representing tree nodes $\vec{n}$. These are generated by independently drawing their elements from a normal distribution N(0,1) with mean 0 and variance 1. The vectors are then normalized so that they have unitary Euclidean length. This generation process guarantees that, for a high enough number of dimensions, the vectors are statistically expected to be nearly orthogonal, i.e. the dot product among pairs of different vectors is expected to be 0.

We can obtain the pure distributed trees (DT) and the distributional distributed trees (DDT) along with their kernel functions, DTK and DDTK, by using different word vectors $\vec{w}$. In the DTs, these vectors are random vectors as the other nodes. In DDTs, these vectors are distributional vectors obtained on a corpus with an LSA reduction (Deerwester et al., 1990).

3.2.1 Entailment-specific Kernels

Recognizing textual entailment (RTE) is a complex semantic task often interpreted as a classification task. Given the text T and the hypothesis H determine whether or not T entails H. For applying the previous kernels to this classification task, we need to define a specific class of kernels. As in (Zanzotto and Moschitti, 2006; Wang and Neumann, 2007; Zanzotto et al., 2009), we encode the text T and the hypothesis H in two separate syntactic feature spaces. Then, given two pairs of text-hypothesis $P_1 = (T_1, H_1)$ and $P_2 = (T_2, H_2)$, the prototypical kernel PK is written as follows:

$$PK(P_1, P_2) = K(T_1, T_2) + K(H_1, H_2) \quad (5)$$

where $K(\cdot, \cdot)$ is a generic kernel. We will then experiment with different PK kernels obtained using: the original tree kernel function (TK) (Collins and Duffy, 2002), DTK, and DDTK.

Along with the previous task specific kernels, we use a simpler feature (Lex) that is extremely effective in determining the entailment between T and H. This simple feature is the lexical similarity between T and H computed using WordNet-based metrics as in (Corley and Mihalcea, 2005). This feature, hereafter called *Lex*, encodes the similarity between T and H, i.e., $sim(T, H)$. This feature is used alone or in combination with the previous kernels and it gives an important boost to their performances. In the task experiment, we will then also have: Lex+TK, Lex+DTK, and Lex+DDTK.

4 Experimental Evaluation

In this section, we experiment with the distributed tree kernels (DTK) and the distributional distributed tree kernels (DDTK) in order to understand whether or not the syntactic structure and the distributional meaning can be easily encoded in the distributed trees. We will experiment in two ways: (1) direct comparison of the distances produced by the original tree kernel (TK) (Collins and Duffy, 2002) and the novel kernels DTK and DDTK; (2) task driven evaluation of DTK and DDTK using the RTE task.

The rest of the section is organized as follows. We firstly introduce the experiment set up that is used for the two settings (Sec. 4.1). Secondly, we report on the experimental results (Sec. 4.2).

4.1 Experimental Set-up

We have the double aim of producing a direct comparison of how the distributed tree kernel (DTK) is approximating the original tree kernel (TK) and a task based comparison for assessing if the approximation is enough effective to similarly solve the task that is textual entailment recognition. For both experimental settings, we take the recognizing textual entailment sets ranging from the first challenge (RTE-1) to the fifth (RTE-5) (Dagan et al., 2006; Bar-Haim et al., 2006; Giampiccolo et al., 2007; Bentivogli et al., 2009).

The distributional vectors used for DDTK have been obtained by an LSA reduction of the word-by-word cooccurrence matrix generated on the UKWaC corpus (Ferraresi et al., 2008), using a context window of size 3. An appropriate size for the LSA reduction was deemed to be 250. Thus, in the experiments we used 250 dimensions both for distributional and random vectors, to allow a correct comparison between DTK and DDTK models.

For the direct comparison, we used tree pairs derived from the RTE sets. Each pair is derived from a T-H pair where T and H are syntactically analyzed and each RTE set produces the corresponding set of tree pairs, e.g., the development set of RTE1 produces a set of 567 tree pairs. To determine whether or not a distributed kernel, DTK or DDTK, is behaving similarly to the original TK kernel, given a set of tree pairs, we produce two ranked lists of tree pairs: the first is ranked according to the original TK applied to the tree pairs and the second according to the target distributed kernel. We evaluate the correlation of the two ranked lists according to the spearman's correlation. Higher correlation corresponds to a better approximation of TK.

For the task driven comparison, we experimented with the datasets in the classical learning setting: the development set is used as training set and the final classifier is tested on the testing set. We used a support vector machine (Joachims, 1999) with an implementation of the original tree kernel (Moschitti, 2006). The classifiers are evaluated according to the accuracy of the classification decision on the testing set, i.e., the ratio of the correct decisions over all the decisions to take.

	Average Spearman's Correlation
DTK	0.8335
$DDTK$	0.7641

Table 1: Average Spearman's correlations of the tree kernel (TK) with the distributed tree kernel (DTK) and the distributed distributional tree kernel (DDTK) in a vector space with 250 dimensions

	avg	RTE1	RTE2	RTE3	RTE5
TK	55.02%	55.50%	53.38%	55.88%	55.33%
DTK	55.63%	57.25%	54.88%	54.38%	56.00%
DDTK	55.11%	54.00%	53.88%	55.38%	57.17%
Lex+TK	62.11%	59.75%	61.25%	66.62%	60.83%
Lex+DTK	63.25%	61.12%	62.12%	66.25%	63.50%
Lex+DDTK	62.90%	60.62%	61.25%	66.38%	63.33%

Table 2: Accuracies of the different methods on the textual entailment recognition task

4.2 Experimental results

In the first experiment of this set, we want to investigate which one between DTK and DDTK correlates better with original TK. Table 1 reports the spearman's correlations of tree kernels with DTK and DDTK in a vector space with 250 dimensions. These correlations are obtained averaging the correlations over the 9 RTE sets. According to these results, DTK better correlates with TK with respect to DDTK. Distributional vectors used for words are not orthogonal as these are used to induce the similarity between words. Yet, this important feature of these vectors determines a worse encoding of the syntactic structure.

In the task driven experiment, we wanted to investigate whether the difference in correlation has some effect on the performance of the different systems. Accuracy results on the RTE task are reported in Table 2. The columns RTE1, RTE2, RTE3, and RTE5 represent the accuracies of the different kernels using the traditional split of training and testing. The column avg reports the average accuracy of the different methods in the 4 sets. Rows represent the different kernels used in this comparative experiment. These kernels are used with the task specific kernel PK by changing the generic kernel K. The first 3 rows represent the *pure* kernels while the last 3 rows represent the kernels boosted with the lexical similarity (Lex), a simple feature computed using WordNet-based metrics, as in (Corley

and Mihalcea, 2005). Looking at the first 3 rows, we derive that there is not a significant difference between TK, DTK, and DDTK. DTK and DDTK can then be used instead of the TK. This is an important result, since the computation of DTK (or DDTK) is much faster than that of TK, due to TK's complexity being quadratic with respect to the size of the trees, and DTK requiring a simple dot product over vectors that can be obtained with linear complexity with respect to the tree size. The second fact is that there is no difference between DTK and DDTK: more semantically informed word vectors have the same performance of random vectors.

5 Conclusions

Distributed structures and distributional meaning are largely correlated. In this paper, we analyzed this correlation with respect to the research challenge of producing compositional models for distributional semantics. In the studies of distributed representation, compositionality is a big issue that has produced many models and approaches. Compositional distributional semantics poses the same issue. We empirically showed that a methodology for including distributional meaning in distributed representation is possible, but it must be furtherly developed to be an added value. Distributional semantics has been positively added in traditional tree kernels (Mehdad et al., 2010). Yet, the specific requirement of distributed tree kernels (i.e., the orthogonality of the vectors) reduces this positive effect.

References

James A. Anderson. 1973. A theory for the recognition of items from short memorized lists. *Psychological Review*, 80(6):417 – 438.

Mark Andrews, Gabriella Vigliocco, and David Vinson. 2009. Integrating experiential and distributional data to learn semantic representations. *Psychological Review*, 116(3):463 – 498.

Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second pascal recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*. Venice, Italy.

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.

Luisa Bentivogli, Ido Dagan, Hoa T. Dang, Danilo Giampiccolo, and Bernardo Magnini. 2009. The Fifth PASCAL Recognizing Textual Entailment Challenge. In *Proceedings of TAC'2009*.

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.

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.

Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407.

Adriano Ferraresi, Eros Zanchetta, Marco Baroni, and Silvia Bernardini. 2008. Introducing and evaluating ukwac, a very large web-derived corpus of english. In *In Proceed-ings of the WAC4 Workshop at LREC 2008*, Marrakesh, Morocco.

P. Foltz, W. Kintsch, and T. Landauer. 1998. The measurement of textual coherence with latent semantic analysis. *Discourse Processes*, 25(2&3):285–307.

Gottlob Frege. 1884. *Die Grundlagen der Arithmetik (The Foundations of Arithmetic): eine logisch-mathematische Untersuchung ber den Begriff der Zahl*. Breslau.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9. Association for Computational Linguistics, Prague, June.

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.

G. E. Hinton, J. L. McClelland, and D. E. Rumelhart. 1986. Distributed representations. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations*. MIT Press, Cambridge, MA.

Thorsten Joachims. 1999. Making large-scale svm learning practical. In B. Schlkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods-Support Vector Learning*. MIT Press.

Thomas K. Landauer and Susan T. Dumais. 1997. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240, April.

K. McRae, V. R. de Sa, and M. S. Seidenberg. 1997. On the nature and scope of featural representations of word meaning. *J Exp Psychol Gen*, 126(2):99–130, June.

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. 2010. Composition in distributional models of semantics. *Cognitive Science*.

Richard Montague. 1974. English as a formal language. In Richmond Thomason, editor, *Formal Philosophy: Selected Papers of Richard Montague*, pages 188–221. Yale University Press, New Haven.

Alessandro Moschitti. 2006. Making tree kernels practical for natural language learning. In *Proceedings of EACL'06*, Trento, Italy.

Bennet B. Murdock. 1983. A distributed memory model for serial-order information. *Psychological Review*, 90(4):316 – 338.

T. A. Plate. 1994. *Distributed Representations and Nested Compositional Structure*. Ph.D. thesis.

David E. Rumelhart and James L. Mcclelland. 1986. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition : Foundations (Parallel Distributed Processing)*. MIT Press, August.

Vladimir Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer.

Rui Wang and Günter Neumann. 2007. Recognizing textual entailment using sentence similarity based on dependency tree skeletons. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 36–41, Prague, June. Association for Computational Linguistics.

Fabio Massimo Zanzotto and Alessandro Moschitti. 2006. Automatic learning of textual entailments with cross-pair similarities. In *Proceedings of the 21st Coling and 44th ACL*, pages 401–408, Sydney, Australia, July.

Fabio Massimo Zanzotto, Marco Pennacchiotti, and Alessandro Moschitti. 2009. A machine learning approach to textual entailment recognition. *NATURAL LANGUAGE ENGINEERING*, 15-04:551–582.

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,.