Inside Outside Recursive Neural Network: A Unified Framework for Compositional Semantics and Meaning in Context

Phong Le, Remko Scha and **Willem Zuidema**

1 Introduction

The reader should not have any problem to understand *this* sentence although (s)he has never seen it. This is evidence that natural languages are compositional. In order to capture this phenomenon, compositional semantics which relies on the principle of compositionality "The meaning of a whole is a function of the meanings of the parts and of the way they are syntactically combined" (Partee, 1995) was introduced. Research on computational semantics therefore focuses on two problems (1) how to learn word representations, and (2) how to learn compositionality functions.

Phrase/word meaning in context, on the other hand, is about how meaning of a phrase/word is changed by affect of it contexts. For instance, at the word level, the word 'bank' in the two following constituents has different meanings

1. xyz
2. uvw

At the phrase level, depending on context, a phrase can have a literal meaning or figurative meaning

1. xyz
2. uvw

It is not difficult to realize that computational semantics and meaning in context have a strong relation. To meaning in context, compositionality is helpful to compute context representations. In addition, if the target is a phrase, compositionality is essential for computing its representation. To computational semantics, context can be used to disambiguate word senses, thus lead to more reliable composition. However, it is surprising that there are no attempts tackling both problems.

In this paper, we propose a new framework, namely Inside Outside Recursive Neural Network (IORNN). IORNN is able to compute phrase representations as well as context representations, thus tackles the both challenges at the same time in a unified framework.

2 Background

2.1 Compositional Semantics

Formal semantics, which uses formal languages to represent constituents, is the first attempt and well fits this principle [cite Montague]. It firstly assumes that words are represented by lambda expressions, e.g. John :- $\lambda x.john(x)$, walks :- $\lambda P \lambda y.walks(y) \wedge P(y)$. Then, using the lambda beta reduction rule as the compositionality function, it easily computes the meaning of the constituent "John walks" $\lambda y.john(y) \wedge walks(y)$. This approach, although being sound with human beings and having a simple but powerful compositionality function, is very challenging to computers since automatically learning word representation is a difficult task (Le and Zuidema, 2012). In addition, formal semantics only supports the truth values (True and False) and ignores lexical semantics restricts it to a short list of applications.

On the other extreme, distributional semantics, based on the *distributional hypothesis* (Lenci, 2008) "The degree of semantic similarity between two linguistic expressions A and B is a function of the similarity of the linguistic contexts in which A and B can appear", is widely used for learning word meaning, which is represented in a vector space. Firstly, it has strong support from not only linguistics but also cognitive science (Lenci, 2008). Secondly, word representations can be learnt from unannotated data, which are redundant and free on the Internet, thanks to the flourish of unsupervised learning techniques, such as Latent semantic analysis (Landauer et al., 1998), neural network language modelling (Collobert et al., 2011; Huang et al., 2012), Brown clustering algorithm (Brown et al., 1992), and spectral learning (Dhillon et al., 2012). And finally, the fact that it supports semantic similarity makes it have a wide range of application: information retrieval, sentiment analysis, machine translation, etc. The distributional hypothesis, however, is not able to apply to phrasal semantics because of the the sparsity of data: the number of semantically plausible phrases is infinitive whereas available data is finite.

The most visible approach to tackle compositional semantics problem is to combine formal semantics and distributional semantics: the former for compositionality, the latter for leaning word representations. [cite Erk, ...] It turns out this is not trivial since the the two kinds of semantics have very different forms of representations so that how to combine these forms is itself a difficult problem.

Another approach, which has been intensively studied recently, is distributional compositional semantics: if $\overrightarrow{a}, \overrightarrow{b}$ are vectors representing the meanings of the two items a, b, then the meaning vector $\overrightarrow{ab}$ of their constituent ab, yielded by the grammar rule R, is computed by (Mitchell and Lapata, 2010)

$$\overrightarrow{ab} = f(\overrightarrow{a}, \overrightarrow{b}, R, K) \tag{1}$$

where f is a to-be-defined function and K is background knowledge.

There are a wide range of approaches to learn compositionality functions f. The most simple approach is use vector addition and multiplication as compositionality functions (Mitchell and Lapata, 2008). Those functions, although no parameters need to be optimized, are too simple to capture real compositionality.

Socher and colleagues propose two neural network frameworks: recursive auto encoder (RAE) (Socher et al., 2011a) for unsupervised learning, and recursive neural network for supervised learning with task-based training signal (Socher et al., 2010) (e.g., for sentiment analysis, the training signal is the sentiment given by voters). The key idea of the RAE framework is that: a compositionality function is a compression function, such that an input is able to be recovered from the output by a decompression function.

(Baroni et al., 2012), (Grefenstette et al., 2013) and others attempt the challenge in a different way. They use tensors to represent functor words (i.e., verbs, adjectives, etc.), linear maps as compositionality functions, and use contexts (in a similar way with distributional lexical semantics) for estimating tensors' elements and functions' parameters.

2.2 Meaning in Context

3 Research Questions

Firstly, we ask ourselves "Which evidence is strong enough to be used for learning compositional semantics?" To answer this question, we rely on the following observation: a human being can guess the meaning of an unknown word by making use of the meaning of its context. In other words, he computes (in his brain) the meaning of the context and then use it to predict the meaning of the target unknown word (by setting some contrains to narrow down a list of possible meanings). Hence, if he correctly predicts the meaning of the unknown word, we can, at some level of belief, say that he coherends the meaning of the context. This idea is then encaptured in the following hypothesis

> *Hypothesis 1:* The agreement betwen words and contexts provides evidence for unsupervised compositional semantics learning.

Then, there are three questions need to be answered in order to implement the hypothesis

1. How to construct phrase representations?
2. How to construct context representations?
3. How to use the agreement betwen words and their contexts to learn compositionality functions?

In the next section, we answer these questions.

4 Inside Outside Recursive Neural Network

In this section, we answer the three questions raising in Section 1.

4.1 Recursive Neural Network (RNN)

(Socher et al., 2010) answer the first question "How to construct phrasal semantics?" by Recusive Neural Network (RNN) architecture, which is used to compute continuous phrase representations. In order to see how RNN works, let's conside the following example. Assuming that there is a consitutent with parse tree $(p_2\ (p_1\ x\ y)\ z)$ as in Figure 1. We will use a neural network, which contains a weight matrix $\mathbf{W}_1$ for left children and a weight matrix $\mathbf{W}_2$ for right children, to compute parents in a bottom up manner. Firstly, we use this network to compute p_1 based on its children x and y

$$\mathbf{p}_1 = f(\mathbf{W}_1\mathbf{x} + \mathbf{W}_2\mathbf{y} + \mathbf{b}) \tag{2}$$

where $\mathbf{b}$ is a bias, f is an activation function (e.g. *tanh* or *logistic*). Then, we use the same network to compute p_2 based on its children p_1 and z

$$\mathbf{p}_2 = f(\mathbf{W}_1\mathbf{p}_1 + \mathbf{W}_2\mathbf{z} + \mathbf{b}) \tag{3}$$

This process is continued until we reach the root node. This network is trained by a gradient-based optimization method (e.g., gradient descent) where the gradient over parameters is efficiently computed thanks to the backpropagation through structure (Goller and Kuchler, 1996). Using this architecture (and its extensions), Socher and colleagues succesfully reach state-of-the-art results in syntactic parsing (Socher et al., 2013a) and sentiment analysis (Socher et al., 2013b).

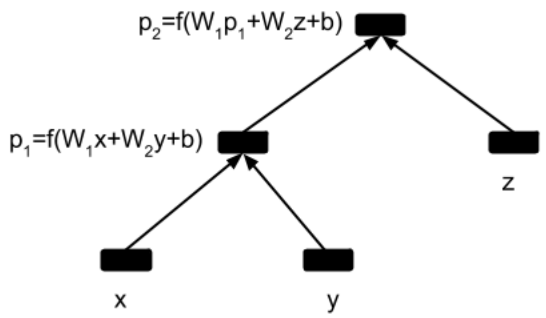

Figure 1: Recursive Neural Network (RNN).

In order to use this architecture in an unsupervised learning manner, (Socher et al., 2011b) replace neural network in RNN by autoencoder (and hence the new architecture is called Recursive Autoencoder - RAE), which is a feedforward neural network trained by forcing output equal to input (see Figure 2). Training a RAE is therefore to minmize the sum of reconstruction errors (i.e., $\|[\mathbf{x}';\mathbf{y}'] - [\mathbf{x};\mathbf{y}]\|^2$) at all internall nodes.

4.2 IORNN

None of the above architectures, RAE or RNN, compute contextual semantics. However, they give us a hint to do that. In this section, we will answer the second question “How to construct contextual semantics?” by a new neural network architecture, namely Inside Outside Recursive Neural Network (IORNN). We also present this architecture by using an example of a constituent and parse tree $(p_2\ (p_1\ x\ y)\ z)$ (see Figure 3).

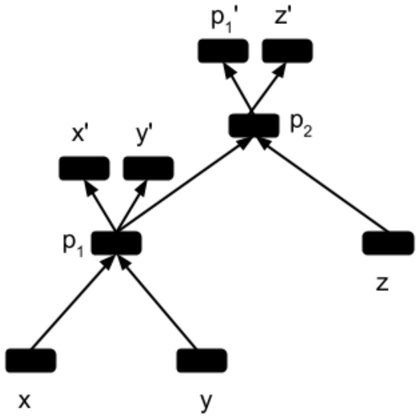

Figure 2: Recursive Autoencoder (RAE).

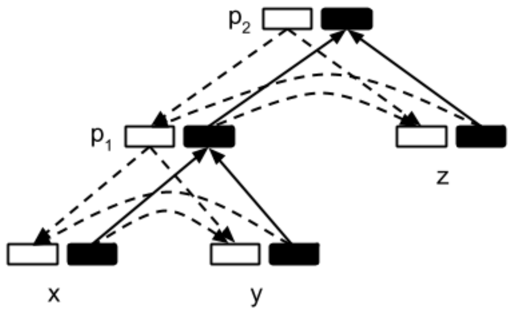

Figure 3: Inside-Outside Recursive Neural Network (IORNN). Black rectangles correspond to inner meanings, white rectangles correspond to outer meanings.

Each node u is assigned two vectors $\mathbf{o}_u$ and $\mathbf{i}_u$. The first one, called *outer meaning*, denotes the meaning of the context; the second one, called *inner meaning*, denotes the meaning of the phrase that the node covers.

Word embeddings (e.g., $\mathbf{i}_x$) Similar to (Socher et al., 2010), and (Collobert et al., 2011), given a string of binary representations of words $(a, b, ..., w)$ (i.e., all of the entries of w are zero except the one corresponding to the index of the word in the dictionary), we first compute a string of vectors $(\mathbf{i}_a, ..., \mathbf{i}_w)$ representing inner meanings of those words by using a look-up table (i.e., word embeddings) $\mathbf{L} \in \mathbb{R}^{n\times|V|}$, where $|V|$ is the size of the vocabulary and n is the dimensionality of the vectors. This look-up table $\mathbf{L}$ could be seen as a storage of lexical semantics where each column is a vector representation of a word. Hence,

$$\mathbf{i}_w = \mathbf{L}w \in \mathbb{R}^n \tag{4}$$

Computing inner meaning The inner meaning of a non-terminal node, say p_1, is given by

$$\mathbf{i}_{p_1} = f(\mathbf{W}_1^i \mathbf{i}_x + \mathbf{W}_2^i \mathbf{i}_y + \mathbf{b}^i) \tag{5}$$

where $\mathbf{W}_1^i, \mathbf{W}_2^i$ are $n \times n$ real matrices, $\mathbf{b}^i$ is a bias vector, and $f(.)$ is an activation function, e.g. $tanh$ function. Intuitively, the inner meaning of a parent node is the function of the inner meanings of its children. This is similar to what (Socher et al., 2010) call recursive neural network.

Computing outer meaning The outer meaning of the root node, $\mathbf{o}_{root}$, is initially set randomly, and then learnt later. To a node which is not the root, say p_1, the outer meaning is given by

$$\mathbf{o}_{p_1} = g(\mathbf{W}_1^o \mathbf{o}_{p_2} + \mathbf{W}_2^o \mathbf{i}_z + \mathbf{b}^o) \tag{6}$$

where $\mathbf{W}_1^o, \mathbf{W}_2^o$ are $n \times n$ real matrices, $\mathbf{b}^o$ is a bias vector, and $g(.)$ is an activation function, e.g. $tanh$ function. Informally speaking, the outer meaning of a node (i.e., the meaning of its context) is the function of the outer meaning of its parent and the inner meaning of its sister.

The reader, if familiar with syntactic parsing, could recognizes the similarity between Equation 5, 6 and the inside, outside probabilities given a parse tree. Therefore, we name the architecture Inside-Outside Recursive Neural Network.

4.3 Training IORNN

This section is to answer the final question "How to use contextual semantics and lexical semantics to learn compositionality functions?".

According to Hypothesis 1, there must be a strong correlation between $\mathbf{o}_w$ and $\mathbf{i}_w$ where w is any word in a given sentence. The simplest way to train the network is to force $\mathbf{o}_{w_j} = \mathbf{i}_{w_j}$; hence, learning is to minimize the following loss function

$$J(\theta) = \sum_{s \in D} \sum_{w \in s} \|\mathbf{o}_w - \mathbf{i}_w\| \tag{7}$$

where D is a set of training sentences and θ are the network parameters. However, that could be problematic because the meaning of context is not necessary the meaning of the target word.

Here, based on the observation that the meaning of context sets contraints on selecting a word to fill in the blank, one could suggest put a *softmax* neuron unit on the top of each $\mathbf{o}_w$ in order to compute the probability $P(x|\mathbf{o}_w)$. Unfortunately, as pointed out by (Collobert et al., 2011), it might not work.

Using the same method proposed by (Collobert et al., 2011), we train the network such that it gives a higher score to the correct target word rather than to incorrect ones. The score $s(x, \mathbf{o}_w)$ given to a candidate word x for a specific context $\mathbf{o}_w$ is computed by

$$u(x, \mathbf{o}_w) = f(\mathbf{W}_u^o \mathbf{o}_w + \mathbf{W}_u^i \mathbf{i}_x + \mathbf{b}_u) \tag{8}$$

$$s(x, \mathbf{o}_w) = \mathbf{W}_s u(x, \mathbf{o}_w) + \mathbf{b}_s \tag{9}$$

where $\mathbf{W}_u^o, \mathbf{W}_u^i$ are $n \times k$ real matrices, $\mathbf{W}_s$ is a $k \times 1$ matrix, and $\mathbf{b}_u, \mathbf{b}_s$ are bias vectors. (We fix $k = 2n$.) Now, the objective function is the ranking criterion with respect to θ

$$J(\theta) = \sum_{s \in D} \sum_{w \in s} \sum_{x \in V} \max\{0, 1 - s(w, \mathbf{o}_w) + s(x, \mathbf{o}_w)\} \tag{10}$$

To minimize the above objective function, we randomly pick up a word in the dictionary as a corrupt example, compute the gradient, and update the parameters by a gradient descent method. Thanks to the backpropagation through structure (Goller and Kuchler, 1996), the gradient is efficiently computed. Following (Socher et al., 2013b), we use AdaGrad (Duchi et al., 2011) to update the parameters.

5 Experiments

In order to examine how IORNN performs on both compositional semantics learning and meaning in context, we evaluate it on two tasks: phrase similarity and word meaning in context. Because, to our knowledge, there are no frameworks that tackle both problems, we use vector addition and pair-wise multiplication as our baselines. Although these methods are simple, choosing them as baselines are reasonable since (1) (Blacoe and Lapata, 2012) show that they perform better than RAE in the phrase similarity task, (2) they are widely used in many applications requiring compositional semantics [cite...], and (3) vector addition is widely used as a method to compute contextual meaning vector (Thater et al., 2011)

In the all experiments, we implemented IORNN in Torch-lua (Collobert et al., 2012). We initialled the network with the 50-dim word embeddings[1] from (Collobert et al., 2011). Then we trained it on a dataset containing 1.5M sentences from the BNC corpus (about one fourth of the whole corpus), which were parsed by the Berkeley parser (Petrov et al., 2006) and binarized.

5.1 Qualitative Evaluations

In order to show that IORNN is cabable to use contex to predict the meaning of a unknown word, we run the trained network on the WSJ section 22 and measured the average predicted rank of target/gold-standard words. (We did not use sentences from the BNC corpus because we also wanted to examine the generality of IORNN: "Does it work on different domains?") For each target word, we create a list of 20,000 candidate words consisting of 19,999 words randomly selected from the vocabulary and the target word itself. Then, we use the scores given in Equation 9 to rank those candidates.

We found the average predicted rank of target words is 187.3 over 20,000 ($< 1\%$), which means that the context meaning computed by IORNN tends to prefer the correct word. Looking into details (see Table 1), we discovered that IORNN tends to predict correctly word class (e.g., noun in examples 1 and 2, verb in example 4, adverb in example 3), word form (e.g., plural in example 1 and 3, bare verb in example 4). This is interesting because gramatical categories are totally ignored in both training and test phases. In addition, in some cases, high rank words seems to be logical in corresponding contexts (e.g., example 3).

5.2 Phrase Similarity

Phrase similarity is the task in which one is asked to compute the meanings of (short) phrases and measure their semantic similarities. Its goodness is measured by comparing its judgements with human judgements. In this experiment, we used the dataset[2] from (Mitchell and Lapata, 2010) which contains 5832 human judgements on semantic similarity for noun-noun, verb-object, and adjective-noun phrases. There are 108 items; each contains a phrase pair and human ratings from 1 (very low similarity) to 7 (very high similarity) (see Table 2).

In this task, we use the cosine distance to measure the semantic similarity, i.e. $d(a, b) = \cos(\mathbf{i}_a, \mathbf{i}_b)$. Following (Blacoe and Lapata, 2012), (Hermann and Blunsom, 2013) and many others, we compute Spearman correlation coefficient ρ between model scores and human judgements.

type	phrase 1	phrase 2	rating
v-obj	remember name	pass time	3
adj-n	dark eye	left arm	5
n-n	county council	town hall	4

Table 2: Items in the dataset from (Mitchell and Lapata, 2010).

First of all, we focus on the results reported by (Blacoe and Lapata, 2012). Blacoe & Lapata claims that RAE performs worse than addition and pair-wise multiplication in all of their three settings. Here, we used their third setting, and, to be fair, we trained IORNN with their neural language model word embeddings[3]. The results are given in Table 3. IORNN is the best on adj-n and v-obj, and second on noun-noun.

dim.	model	adj-n	n-n	v-obj
50	add.	0.28	**0.26**	0.24
50	mult.	0.26	0.22	0.18
100	RAE	0.20	0.18	0.14
50	IORNN	**0.30**	0.23	**0.28**

Table 3: Spearman correlation coefficients of model predictions for the phrase similarity task with neural language model word embeddings from (Blacoe and Lapata, 2012).

(Hermann and Blunsom, 2013) extend RAE with the help of Combinatory Categorial Grammar (CCG). Their models, named Combinatory Categorial Autoencoder (CCAE), are similar to RAE but use different parameter sets for different grammatical rules and grammatical types. Thank to this semantic-related grammar, their models outperform RAE and score towards the upper end of the range of addition and pair-wise multiplication. Table 4 shows the comparison between IORNN and other methods whose results are copied from the corresponding papers.

[1] http://ronan.collobert.com/senna/

[2] http://homepages.inf.ed.ac.uk/s0453356/share

[3] http://homepages.inf.ed.ac.uk/s1066731/dl.php?file=wordVectors.emnlp2012.zip&db=1

ID	word	top 10 candidates
1	Institutional investors and bankers [...] were cautiously optimistic after the mild 1.8% decline in Tokyo stock *prices*.	standards, hours, projects, roads, members, duty, restrictions, *prices*, house, locations
2	That is *why* everybody was a little surprised by the storm of sell orders from small private investors [...]	at, over, when, since, *why*, why, one, was, a
3	That is why everybody was a little surprised by the storm of sell orders from small private *investors*, " said Norbert Braeuer [...]	men, companies, courts, camps, businesses, sports, jobs, parks, partners, air
4	[...] most investors wanted to see what would *happen* in New York before acting.	go, say, try, and, want, ',', work, invest, to, forget

Table 1: Words, their contexts, and top 10 candidates.

model	adj-n	n-n	v-obj
Blacoe & Lapata			
a./m.	0.21 - 0.48	0.22 - 0.50	0.18 - 0.35
RAE	0.19 - 0.31	0.24 - 0.30	0.09 - 0.28
Hermann & Blunsom			
CCAEs	0.38 - 0.41	0.41 - 0.44	0.23 - 0.34
Our implementation			
add.	0.30	0.43	0.30
mult.	0.14	0.24	0.16
IORNN	0.38	0.36	0.32

Table 4: Spearman correlation coefficients of model predictions for the phrase similarity task.

IORNN's performance lies in the range of CCAEs on adj-n and v-obj, but worse on noun-noun. However, it is worth emphasizing that IORNN uses one parameter set for all grammatical rules (similarly to RAE). From the difference of performance between RAE and CCAEs, we expect that extending IORNN in the same way (i.e., using CCG and different parameter sets for different grammatical rules and grammatical types) will lead to better performance.

5.3 Word Similarity in Context

Differing from the first task, this task focuses on word meaning in context: it examines how well a model can make use of context to disambiguate word meaning. In this experiment, we use the Stanford Word Similarity in Context (SWSC) dataset from (Huang et al., 2012) which contains 2003 word pairs, their sentential contexts, and human ratings from 0 to 10 (see Table 5).

For IORNN, we represent the meaning of a word in its sentential context by concatenate its inner and outer meanings, i.e. $\mathbf{m}_w = [\mathbf{i}_w; \mathbf{o}_w]$. For vector addition, we compute the context meaning by averaging the meaning vectors of 5 words on the left and 5 words on the right and then concatenate it with the meaning vector of the target word.

Similarly to the first experiment, we also use the cosine distance to measure the semantic similarity and compute Spearman correlation coefficient ρ between model scores and human judgements.

We compare IORNN and vector addition with the method proposed by (Huang et al., 2012) (HSMN-M AvgSimC) and the one proposed by (Reisinger and Mooney, 2010) (Pruned tf-idf-M AvgSimC). These two methods are multi-prototype approaches: the meaning of a word is represented by multiple vectors (i.e., prototypes). In order to extract multipe prototypes for a word, they compute a vector for each context that the word is in, then cluster those context vectors. In this way, a prototype corresponds to the centroid of a cluster, and hence represent a sense of the word. In order to compute the word meaning similarity with context, they use AvgSimC metric

$$\text{AvgSimC}(w, w') = \frac{1}{k^2} \sum_{i=1}^{k} \sum_{j=1}^{k} p(c, w, i) p(c', w', j) d(\mu_i(w), \mu_j(w'))$$

where k is the number of prototypes of each word, $p(c, w, i)$ is the likelihood that word w is in its cluster i given context c, $\mu_i(w)$ is the i-th cluster centroid of w and $d(v, v')$ is a function computing similarity between two vectors.

Table 6 shows the comparison between the methods. It is not surprising to see HSMN-M AvgSimC and Pruned tf-idf-M AvgSimC perform

word 1	word 2	human ratings
Located downtown along the east *bank* of the Des Moines River, the plaza is available for parties, ...	This is the basis of all *money* laundering, ...	0.0, 0.0, 3.0, 10.0, 8.0, 0.0, 4.0, 0.0, 0.0, 0.0

Table 5: An example in the SWSC dataset.

best since disambuguiating word sense is the key to success and these methods use different vectors to represent different senses of a word. However, IORNN, which represents the meaning of a word by a vector, performs comparably with Pruned tf-idf-M AvgSimC, and higher than the vector addition. It is worth noting the improvement from without using context (C&W) to using context meaning computed by IORNN (from 58.0 to 60.2), thus confirming the abbility of IORNN in computing word meaning in context.

Model	$\rho \times 100$
C&W (w/o context)	58.0
Huang et al.	
HSMN-M AvgSimC	65.7
Pruned tf-idf-M AvgSimC	60.5
Our implementation	
add.	59.0
IORNN	60.3

Table 6: Spearman correlation coefficients of model predictions on the SWCS dataset. C&W is the method to use the Collobert & Wetson word embeddings without taking context into account.

5.4 Summary

Now, we combine the experimental results presented above to compare IORNN with the two baselines, vector addition and vector pair-wise multiplication in Table 7. In phrase similarity task, IORNN outperforms the both baselines on adj-n and v-obj and worse than addition on noun-noun [why???]. In word similarity in context task, IORNN also outperforms the both baselines. These results show us that we can tackle the two problems, compositional semantics and meaning in context, with the unified framework IORNN.

6 Discussion

In this section, we will discuss two important issues. The first one is about the cognitive plausibility of IORNN. The second is about potential extensions for it.

Model	Phrase similarity			WSC
	adj-n	n-n	v-obj	
add.	0.30	**0.43**	0.30	59.0
mult.	0.14	0.24	0.16	-
IORNN	**0.38**	0.36	**0.32**	**60.3**

Table 7: Comparison of IORNN against vector addition and pair-wise multiplication in the two tasks.

6.1 Cognitive Plausibility

We found that it is cognitively plausible to represent context meaning seperatively from phrase/word meaning, which we have called *outer meaning* and *inner meaning* respectively. The key point here is what is called *dynamic binding* in connectionsim [cite...] where, in our case, outer meaning could be seen as slots and inner meaning as fillers (see Figure 4). Similar to hierarchical prediction network proposed by (Borensztajn et al., 2009), if the outer meaning and inner meaning are strongly correlated, a dynamic binding occurs and connects the neurons at the tree root node of the phrase to the neurons at the corresponding node of the tree of the context. This explains why a human can use context meaning to predict the meaning of a unknown word, and why (s)he can select a word/phrase to fill in a blank in a uncomplete sentence.

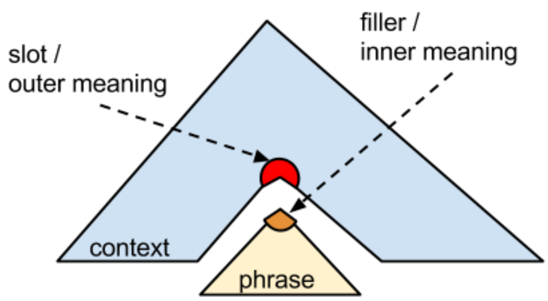

Figure 4: If there is a strong correlation between outer meaning and inner meaning, a dynamic binding occurs.

6.2 Potential Extensions / Future Work

Our new architecture IORNN has many potental extensions, some of which are promising to improve the performance in those two tasks presented above, other could lead to important applications.

At lexicon level In Subsection 5.3, we have seen multi-prototype approaches are potental for computing word meaning in context. Certainly, we can combine these approaches and our framework IORNN at the lexical level in order to reduce ambiguity introducing by lexical semantics.

At syntax level IORNN only uses parse trees without grammatical categories. However, Herman & Blunsom emperically show the important role of syntax in vector space models of compositional semantics. It turns out that it is also easy to extend IORNN in the same way, i.e. using CCG and different parameter sets for different grammatical rules and grammatical types. Thanks to some degree of similarity between IORNN and RAE, we expect that this extension helps IORNN imporve its capacity in capturing compositionality.

At discourse level In this paper, we propose IORNN as an architecture processing individual sentences; therefore, the outer meaning at the root node is alway a null-context outer meaning vector (i.e., $\mathbf{o}_{root} = \mathbf{o}_{\emptyset}$). It turns out that it is easy to extend IORNN to make use of discourse context. Figure 5 illustrates how to connect inner and outer meanings of sentences in a discourse. Intuitively, the outer meaning of a sentence is the function of the inner meanings of its neighbour sentences.

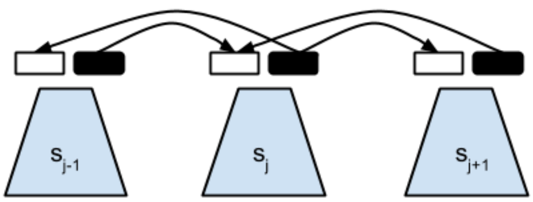

Figure 5: IORNN with Discourse Context. Black rectangles correspond to inner meanings, white rectangles correspond to outer meanings.

7 Conclusion

References

Baroni, M., Bernardi, R., and Zamparelli, R. (2012). Frege in space: A program for compositional distributional semantics.

Blacoe, W. and Lapata, M. (2012). A comparison of vector-based representations for semantic composition. *RAE*, 70(68.29):6910.

Borensztajn, G., Zuidema, W., and Bod, R. (2009). The hierarchical prediction network: towards a neural theory of grammar acquisition. In *31th Annual Conference of the Cognitive Science Society*. Citeseer.

Brown, P. F., Desouza, P. V., Mercer, R. L., Pietra, V. J. D., and Lai, J. C. (1992). Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

Collobert, R., Kavukcuoglu, K., and Farabet, C. (2012). Implementing neural networks efficiently. In *Neural Networks: Tricks of the Trade*, page 537557. Springer.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:24932537.

Dhillon, P., Rodu, J., Foster, D., and Ungar, L. (2012). Two step cca: A new spectral method for estimating vector models of words. *arXiv preprint arXiv:1206.6403*.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 999999:2121–2159.

Goller, C. and Kuchler, A. (1996). Learning task-dependent distributed representations by backpropagation through structure. In *Neural Networks, 1996., IEEE International Conference on*, volume 1, page 347352.

Grefenstette, E., Dinu, G., Zhang, Y.-Z., Sadrzadeh, M., and Baroni, M. (2013). Multi-step regression learning for compositional distributional semantics. *arXiv preprint arXiv:1301.6939*.

Hermann, K. M. and Blunsom, P. (2013). The role of syntax in vector space models of compositional semantics. *Proceedings of ACL, Sofia, Bulgaria, August. Association for Computational Linguistics*.

Huang, E. H., Socher, R., Manning, C. D., and Ng, A. Y. (2012). Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics.

Landauer, T. K., Foltz, P. W., and Laham, D. (1998). An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284.

Le, P. and Zuidema, W. (2012). Learning compositional semantics for open domain semantic parsing. In *COLING*, pages 1535–1552.

Lenci, A. (2008). Distributional semantics in linguistic and cognitive research. *Italian journal of linguistics*, 20(1):131.

Mitchell, J. and Lapata, M. (2008). Vector-based models of semantic composition. *proceedings of ACL-08: HLT*, page 236244.

Mitchell, J. and Lapata, M. (2010). Composition in distributional models of semantics. *Cognitive Science*, 34(8):13881429.

Partee, B. (1995). Lexical semantics and compositionality. *An invitation to cognitive science: Language*, 1:311360.

Petrov, S., Barrett, L., Thibaux, R., and Klein, D. (2006). Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 433–440. Association for Computational Linguistics.

Reisinger, J. and Mooney, R. J. (2010). Multi-prototype vector-space models of word meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 109–117. Association for Computational Linguistics.

Socher, R., Bauer, J., Manning, C. D., and Ng, A. Y. (2013a). Parsing with compositional vector grammars. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Socher, R., Manning, C. D., and Ng, A. Y. (2010). Learning continuous phrase representations and syntactic parsing with recursive neural networks. In *Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop*.

Socher, R., Pennington, J., Huang, E. H., Ng, A. Y., and Manning, C. D. (2011a). Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, page 151161.

Socher, R., Pennington, J., Huang, E. H., Ng, A. Y., and Manning, C. D. (2011b). Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 151–161. Association for Computational Linguistics.

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. (2013b). Recursive deep models for semantic compositionality over a sentiment treebank. EMNLP.

Thater, S., Fürstenau, H., and Pinkal, M. (2011). Word meaning in context: A simple and effective vector model. In *IJCNLP*, pages 1134–1143.