Event Knowledge in Compositional Distributional Semantics

Ludovica Pannitto*
Università di Pisa
Università di Trento

Alessandro Lenci**
Università di Pisa

The great majority of compositional models in distributional semantics present methods to compose vectors or tensors in a representation of the sentence. Here we propose to enrich one of the best performing methods (vector addition, which we take as a baseline) with distributional knowledge about events. The resulting model is able to outperform our baseline.

1. Compositional Distributional Semantics: Beyond vector addition

Linguistic competence entails the ability to understand and produce an unbounded number of novel, complex linguistic expressions. The comprehension of such expressions involves the construction of a semantic representation that, following a common statement for the so-called *principle of compositionality*, is said to be a function of the meaning of its parts and their syntactic modes of combination (Partee 1984).

These representations are needed to support human reasoning about the event or situation that is cued by language use. Consider for instance the different implications of sentences 1 and 2:

(1) After the landing, the pilot switched off the engine.

(2) After the rally, the pilot switched off the engine.

While the two sentences share the proposition *the pilot switched off the engine*, we are likely to infer different things, for instance, about the *engine* that is being swichted-off (i.e., the fact that in (1) it refers to an airplane or a ship while in (2) it refers to a car). Other aspects are involved as well: different inferences could be made upon which other participants are expected to perform further actions, for example *cabin crew*, *control tower*, *passengers* might be involved in the first scenario, but are definitely cut out from the second. Words like *landing* and *rally* cue in fact very different situations in the two sentences, creating different sets of expectations about the described event.

We expect our computational resources to be able to model such phenomena, that make up the very core of language use. In the last decades, distributional semantics has provided a solid framework for the representation of word meaning (Lenci 2018), and various approaches have been also introduced in order to extend vector models of meaning beyond the word level, for representing the meaning of more complex structures such as sentences. Compositional distributional semantics has mainly adopted

* CIMeC - Corso Bettini 31, 38068 Rovereto (TN), Italy. E-mail: ludovica.pannitto@unitn.it
** Dipartimento di Filologia, Letteratura e Linguistica - 56126 Pisa (PI), Italy.
E-mail: alessandro.lenci@unipi.it

a *syntactically transparent* model of semantic composition (Jackendoff 1997), and has addressed the problem of compositionality mainly relying on this standard, Fregean approach, namely considering the lexicon as a pretty much fixed set of word-meaning pairs, and representing sentence meaning as the algebraic composition of pre-computed semantic representations. Composing word representations into larger phrases and sentences notoriously represents a big challenge for distributional semantics (Lenci 2018). Various approaches have been proposed ranging from simple arithmetic operations on word vectors (Mitchell and Lapata 2008), to algebraic compositional functions on higher-order objects (Baroni, Bernardi, and Zamparelli 2014; Coecke, Clark, and Sadrzadeh 2010), as well as neural networks approaches that build so-called sentence embeddings (Kiros et al. 2015; Conneau et al. 2017).

Among all proposed compositional functions, vector addition still shows remarkable performances on various tasks, such as phrase similarity or paraphrase detection (Asher et al. 2016; Blacoe and Lapata 2012; Rimell et al. 2016), beating more complex methods, such as the Lexical Functional Model (Baroni, Bernardi, and Zamparelli 2014). However, the success of vector addition is quite puzzling from the linguistic and cognitive point of view: the meaning of a complex expression is not simply the sum of the meaning of its parts, and the contribution of a lexical item might be different depending on its syntactic as well as pragmatic context.

The majority of available models in literature assumes the meaning of complex expressions like sentences to be a vector (i.e., an embedding) projected from the vectors representing the content of its lexical parts. However, as pointed out by Erk and Padó (2008), while vectors serve well the cause of capturing the semantic relatedness among lexemes, this might not be the best choice for more complex linguistic expressions, because of the limited and fixed amount of information that can be encoded. Moreover events and situations, expressed through sentences, are by definition inherently complex and structured semantic objects. Actually, assuming the equation "meaning is vector" is eventually too limited even at the lexical level.

On the other hand, factors that have been long assumed to lie outside the lexicon, such as pragmatic or world knowledge, have proven to be processed together with lexical knowledge, playing a significant role in comprehension very early in processing, guiding the hearer's expectations about the upcoming input. Psycholinguistic evidence shows that lexical items activate a great amount of generalized event knowledge (GEK) (Elman 2011; Hagoort and van Berkum 2007; Hare et al. 2009), and that this knowledge is crucially exploited during online language processing, constraining the hearer' expectations about upcoming linguistic input (McRae and Matsuki 2009). GEK is concerned with the idea that the lexicon is not organized as a dictionary, but rather as a network, where words trigger expectations about the upcoming input, influenced by pragmatic knowledge along with lexical knowledge. Therefore sentence comprehension can be phrased as the identification of the event that best explains the linguistic cues used in the input (Kuperberg and Jaeger 2016).

Here, we introduce **MEDEA** (Merging Event knowledge and Distributional vEctor Addition), a structured compositional distributional model of sentence meaning which integrates vector addition with generalized event knowledge activated by lexical items (Section 2). MEDEA is directly inspired by the model in Chersoni, Lenci, and Blache (2017) and relies on two major assumptions:

- lexical items are represented with embeddings within a network of syntagmatic relations encoding prototypical knowledge about events;
- the semantic representation of a sentence is a structured object incrementally integrating the semantic information cued by lexical items.

Our aim is to integrate the evidence on the role played by event knowledge during language processing in a linguistically motivated model for compositional semantic representations.

We test MEDEA (Section 3) on two datasets for compositional distributional semantics in which addition has proven to be hard to beat: the first is RELPRON (Rimell et al. 2016), a popular dataset for the similarity estimation between compositional distributional representations; the second is the transitive sentences similarity dataset (Kartsaklis and Sadrzadeh 2014). Our results (Section 4) show that event knowledge plays an important role in the compositional process and that it retains more or different information than what is generally encoded in distributional vectors.

2. Introducing MEDEA

Like in Chersoni, Lenci, and Blache (2017), the model is inspired by Memory, Unification and Control (MUC), proposed by Hagoort (2013, 2016) as a general model for the neurobiology of language. MUC incorporates three main functional components: i.) *Memory* corresponds to knowledge stored in long-term memory; ii.) *Unification* refers to the process of combining the units stored in *Memory* to create larger structures, with contributions from the context; and iii.) *Control* is responsible for relating language to joint action and social interaction. Similarly, our model distinguishes between:

- a **Distributional Event Graph** (DEG) that models a fragment of semantic memory activated by lexical units (Section 2.1);
- a **Meaning Composition Function** that dynamically integrates information activated from DEG to build a sentence semantic representation (Section 2.2)

2.1 Distributional Event Graph

In order to represent the GEK cued by lexical items during sentence comprehension, we explored a graph based implementation of a distributional model, for both theoretical and methodological reasons: in graphs, structural-syntactic information and lexical information can naturally coexist and be related, moreover vectorial distributional models often struggle with the modeling of dynamic phenomena, as it is often difficult to update the recorded information, while graphs are more suitable for situations where relations among items change overtime.

The data structure would ideally keep track of each event automatically retrieved from corpora, thus indirectly containing information about schematic or underspecified events, by abstracting over one or more participants from each recorded instance. Events are extracted from parsed sentences, using syntactic relations as an approxima-

tion of semantic roles (e.g., the subject relation for the agent, the direct object relation for the patient, etc.).[1]

Given a lexical head (e.g., a verb or a noun), all its syntactic dependents are grouped together, similarly to the syntactic joint contexts for verb representation that were proposed by Chersoni et al. (2016).[2] More schematic events are also generated by abstracting from one or more event participants for every recorded instance (cf. Figure 1). We assume a very broad notion of *event*, as an n-ary relation between entities. Accordingly, an event can be a complex situation involving multiple participants, such as *The student reads a book in the library*, but also the association between an entity and a property expressed by the noun phrase *heavy book* (in accordance to what psychologists call *situation knowledge* or *thematic associations* (Binder 2016)). With respect to psycholinguistic research (McRae and Matsuki 2009), DEG can be regarded as a model of the generalized knowledge about events that can be derived from linguistic input, while in general GEK can be acquired from a richer array of inputs (e.g., including sensorimotor experience).

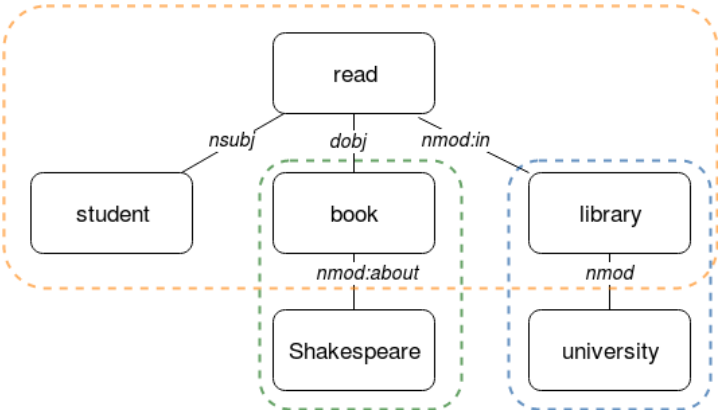

Figure 1
Reduced version of the parsing for the sentence *The student is reading the book about Shakespeare in the university library*. Three events are identified, each represented with a dotted box.

The nodes of DEG are lexical embeddings, and edges link lexical items participating to the same events (i.e., its syntagmatic neighbors, Figure 2). Edges are weighted with respect to the statistical salience of the event (i.e., the labeled link) given the item (i.e.,

1 We chose to use syntactic labels as a proxy for semantic relations for both general and practical reasons: from the practical point of view, syntactic annotation is much easier to obtain than semantic parsing, and many more resources are available with this kind of annotation. Moreover, dependency parsing, especially in the Universal Dependencies framework, provides a practical way to isolate relations between semantically full words such as nouns, verbs and adjectives, that are used here to cue the relevant subsets of DEG. In this sense, the choice poses some problems when it comes to more fine-grained semantic distinctions not easily captured through syntax (e.g., some prepositional complements may be ambiguous between *time* vs. *location* interpretation), but at the same time it offers a valuable opportunity and a simple way to deal with the notion of semantic role in a distributional approach.

2 *Syntactic joint contexts* are defined as an abstraction over joint contexts [Melamud et al. 2014], where each feature of the vector corresponds to a full argument constellation of a verb, to approximate the knowledge about typical event participants. For instance, a joint context for the verb *eat* in the *The dog eats the bone* is formed by both the subject *dog* and the direct object *bone*. The present work applies the same sort of notion to different categories than verbs and, more importantly, deals with the contextualization of the obtained representations.

Table 1
The five nearest paradigmatic and syntagmatic neighbors for the lexical item book, extracted from DEG.

Paradigmatic Neighbors	essay, story, novel, author, biography
Syntagmatic Neighbors	publish, write, read, child, series

the node). Weights, expressed in terms of a statistical association measure such as *Local Mutual Information*, determine the strength with which linguistic cues activate event information from the DEG. The resulting structure can therefore be seen as a weighted hypergraph, as it contains relations holding among groups of nodes, and a labeled multigraph, since each edge or hyperedge is labeled in order to represent the *syntactic pattern* holding in the group. Given the same group of words, in fact, different syntactic patterns are possible: for instance, considering the triplet *cat - chase - dog*, the pattern *nsubj - root - dobj* would indicate the event *The cat chases the dog*, while the pattern *dobj - root - nsubj* refers to the opposite situation where *The dog chases the cat*, and the pattern *nsubj - root - obl* could refer to a passive formulation such as *The cat is chased by the dog*.[3] The weights are derived from co-occurrence statistics and measure the association strengths between event nodes. They are intended as salience scores that identify the most prototypical events associated with an entity (e.g., the typical actions performed by a student).

As graph nodes are embeddings, given a lexical cue w, DEG can be used to retrieve two kinds of information:

- the most similar nodes to w (i.e., its paradigmatic neighbors), using a vector similarity measure like the cosine[4] (Table 1, top row);
- the closest associates of w (i.e., its syntagmatic neighbors), using the weights on the graph edges (Table 1, bottom row).

2.2 Meaning Composition Function

In MEDEA, we model sentence comprehension as the creation of a semantic representation SR (Figure 3), which includes two different yet interacting information tiers that are equally relevant in the overall representation of sentence meaning:

- *linguistic conditions* (LC) - a context-independent tier of meaning that accumulates the embeddings associated with the lexical items, as traditional compositional distributional models do;

3 The syntactic labels (e.g. *root*, *nsubj*, etc.) conform to the Universal Dependencies tagset available at `https://universaldependencies.org/`.

4 Cosine similarity is one of the most widely employed measures in vector space and quantifies the similarity of two non-zero vectors in terms of the angle between them:

$$cos(\theta) = cos(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \left\|\vec{b}\right\|} = \frac{\sum_i a_i b_i}{\sqrt{\sum_i a_i^2}\sqrt{\sum_i b_i^2}} \quad (1)$$

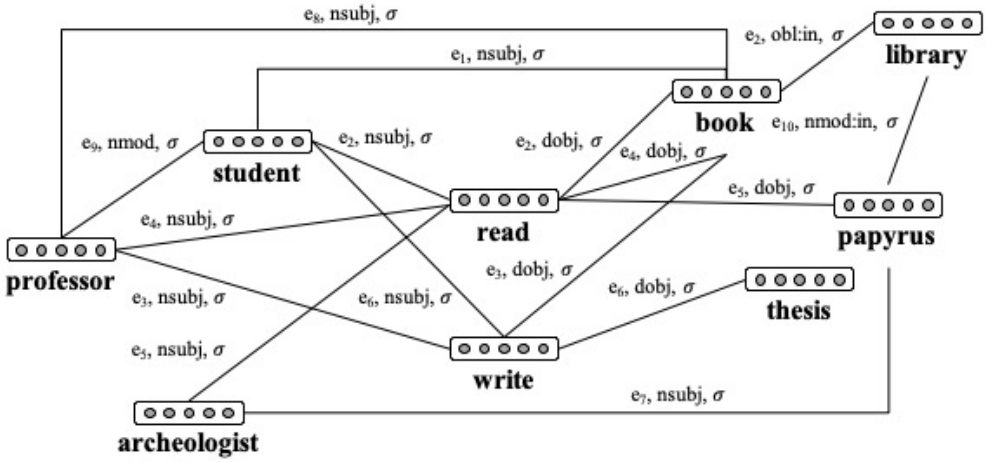

Figure 2
Toy example of DEG showing several instances of events, each represented by a sequence of co-indexed *e*. For example, e_2 corresponds to the event of students reading books in libraries, while e_1 and e_8 represent schematic events of students and professors performing some generic action on books (e.g., reading, consulting, studying, etc.). Each direct labeled edge is associated with its salience weight σ.

- *active context* (AC) - which aims at representing the most probable event, in terms of its participants, that can be reconstructed from DEG subsets cued by lexical items. More specifically, we assume that AC contains the embeddings activated from DEG by the single lexemes (or by other contextual elements) and integrated into a semantically coherent structure. The Active Context makes it possible to enrich the semantic content of the sentence with contextual information, predict other elements of the event, and generate expectations about incoming input. For instance, given the AC in Figure 3, we can predict that the student is most likely to be drinking a coffee at the cafeteria and that he/she is drinking it for breakfast or in the morning. The ranking of each element in AC depends on two factors: i.) its degree of activation by the lexical items, ii.) its overall coherence with respect to the information already available in the AC.

Let SR_{i-1} be the semantic representation built for the linguistic input $w_1, \ldots, w_{i-1}$. When we process a new pair $\langle w_i, r_i \rangle$ with a lexeme w_i and syntactic role r_i:

1. LC in SR_{i-1} is updated with the embedding $\overrightarrow{w_i}$;
2. AC in SR_{i-1} is updated with the embeddings of the syntagmatic neighbors of w_i extracted from DEG.

Figures 4 and 5 exemplify the update of the SR for the subject *student* with the information activated by the verb *drink*. The update process is defined as follows:

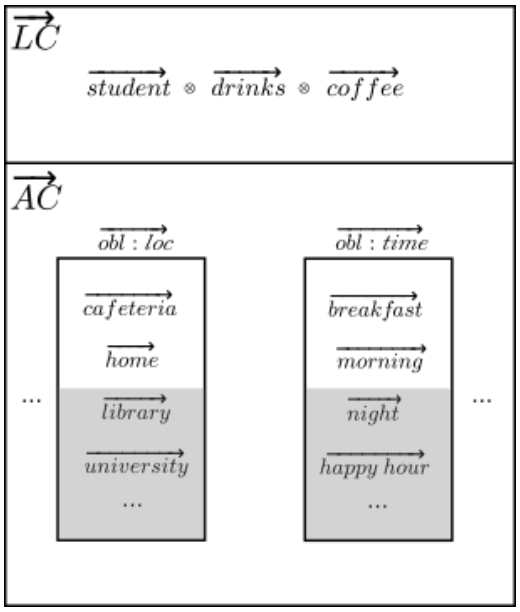

Figure 3
Sample SR for the sentence *The student drinks the coffee*. The LC includes the embeddings of the lexical items in the sentence and a generic composition function, while AC consists of lists of embeddings attached to a syntactic label: these have been activated from DEG and ranked by their salience with respect to the current content in the SR. Syntactic labels are taken as a surface approximation of their semantic role (e.g., the items listed under "obl:*loc*" are a set of possible locations of the event expressed by the sentence).

1. LC is represented with the vector $\overrightarrow{LC}$ obtained from the combination of the embeddings of the words contained in the sentence. Therefore, when $\langle w_i, r_i \rangle$ is processed, the embedding $\overrightarrow{w_i}$ is simply added to $\overrightarrow{LC}$;
2. for each syntactic role r_i, AC contains a set of ranked lists (one for each processed pair that triggers that syntactic role) of embeddings corresponding to the most likely words expected to fill that role. For instance, the AC for the chunk *The student* in Figure 4 contains a list of the embeddings of the most expected main verbs and direct objects associated with *student*, a list of the embeddings of the most expected locations, etc. Each list of expected role fillers is itself represented with a centroid vector[5] (e.g., $\overrightarrow{dobj}$) of their k most prominent items (with k a model hyperparameter). For instance, setting $k = 2$, the $\overrightarrow{dobj}$ centroid in the AC in figure 4 is built just from $\overrightarrow{book}$ and $\overrightarrow{research}$; less salient elements (the gray areas in Figures 3, 4 and 5) are kept in the list of likely direct objects, but at this stage do not contribute to the centroid representing the expected fillers

5 A centroid vector is generally obtained as a (weighted) average of a set of vectors, namely $\overrightarrow{X} = \frac{\sum_{i=0}^{|V|} p_i \overrightarrow{v_i}}{\sum_{i=0}^{|V|} p_i}$, where V is the set of vectors $\overrightarrow{v_i}$ and p_i is a scalar representing the weight attributed to each vector.

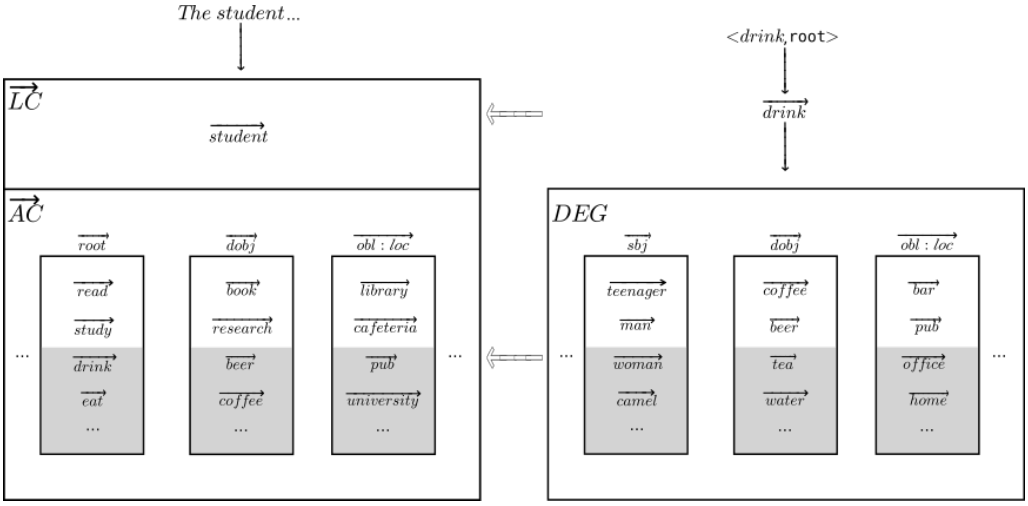

Figure 4
On the left, the SR generated after having processed the first chunk (i.e., *The student...*). On the right, the embedding and DEG subsets activated by the verb *drinks*.

for that role. AC is then updated with the DEG subset activated by the new lexeme w_i (e.g., the verb *drink*):

- the event knowledge activated by w_i for a given role r_i is ranked according to cosine similarity with the vector $\overrightarrow{r_i}$ available in AC: in our example, the direct objects activated by the verb *drink* (e.g., $\overrightarrow{beer}$, $\overrightarrow{coffee}$, etc.) are ranked according to their cosine similarity to the $\overrightarrow{dobj}$ vector of the AC;
- the ranking process works also in the opposite direction: the newly retrieved information is used to update the centroids in AC. For example, the direct objects activated by the verb *drink* are aggregated into centroids and the corresponding weighted lists in AC are re-ranked according to the cosine similarity with the new centroids, in order to maximize the semantic coherence of the representation. At this point, $\overrightarrow{book}$ and $\overrightarrow{research}$, which are not as salient as $\overrightarrow{coffee}$ and $\overrightarrow{beer}$ in the *drinking* context, are downgraded in the ranked list and are therefore less likely to become part of the $\overrightarrow{dobj}$ centroid at the next step.

The newly retrieved information is now added to the AC: as shown in Figure 5, once the pair $\langle drink, root \rangle$ has been fully processed, the AC contains lists for each triggered syntactic role, containing for example just one list for the *sbj* role, which was only triggered by the verb, and two ranked lists for the *dobj* role, that was triggered by both previous elements. The whole AC is represented with the centroid vector $\overrightarrow{AC}$ built out of a subset of the role vectors $\overrightarrow{r_1}, \ldots, \overrightarrow{r_n}$ available in AC.

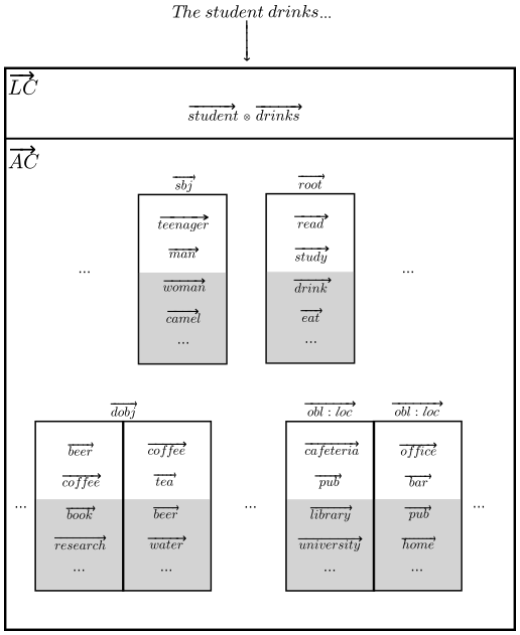

Figure 5
The original semantic representation SR for *The student...* is updated with the information activated by the verb, producing the SR for *The student drinks ...* The new event knowledge is re-ranked with respect to the previous content of AC.

3. Experiments

3.1 Datasets and Tasks

Our aim is to evaluate the contribution of activated event knowledge in a sentence comprehension task. For this reason, among the several existing datasets concerning entailment or paraphrase detection, we chose RELPRON (Rimell et al. 2016), a dataset of subject and object relative clauses, and the transitive sentence similarity (TSS) dataset presented in Kartsaklis and Sadrzadeh (2014). These two datasets show an intermediate level of grammatical complexity, as they involve complete sentences (while other datasets include smaller phrases), but have fixed length structures featuring similar syntactic constructions (i.e., transitive sentences). The two datasets differ with respect to size and construction method.

RELPRON consists of 1,087 pairs, split in development (518 items) and test set (579 items), made up by a *target* noun labeled with a syntactic role (either *subject* or *direct object*) and a *property* expressed as *head noun* followed by a relative clause composed by a *verb* and a *nominal argument*. For instance, here are some example properties for the target noun *treaty*:

(3) a. OBJ treaty: document that delegation negotiate

b. SBJ treaty: document that grant independence

For each target *t*, the representations for the 518 properties in the dataset[6] are built and ranked according to their similarity to *t*. Like Rimell et al. (2016), we use Mean Average Precision (henceforth MAP) to evaluate our models on RELPRON. Formally, MAP is defined as

$$MAP = \frac{1}{N}\sum_{i=1}^{N} AP(t_i) \tag{2}$$

where N is the number of target nouns in RELPRON, and $AP(t)$ is the Average Precision for target t, defined as:

$$AP(t) = \frac{1}{P_t}\sum_{k=1}^{M} Prec(k) \times rel(k) \tag{3}$$

Here, P_t is the number of correct properties for target t in the dataset, M is the total number of properties in the dataset, $Prec(k)$ is the precision at rank k, and $rel(k)$ is a function equal to one if the property at rank k is a correct property for t, and zero otherwise. Intuitively, $AP(t)$ will be 1 if, for the target t, all the correct properties associated to it are ranked in the top positions, and the value becomes lower when the correct properties are ranked farther from the head of the list.

We represented each property in RELPRON as a triplet $((hn, r), (w_1, r_1), (w_2, r_2))$ where hn is the head noun, w_1 and w_2 are the lexemes that compose the proper relative clause, and each element of the triplet is associated with its syntactic role in the property sentence.[7]

The TSS dataset consists of 108 pairs of transitive sentences, each annotated with human similarity judgments collected through the Amazon Mechanical Turk platform. Each transitive sentence in composed by a triplet *subject verb object*. Here are two pairs with high (4) and low (5) similarity scores respectively:

(4) a. government use power
    b. authority exercise influence

(5) a. team win match
    b. design reduce amount

Similarly to RELPRON properties, each sentence of the TSS is represented a triplet $((w_1, sbj), (w_2, root), (w_3, dobj))$. We built a compositional vector representation for both sentences of each item of the dataset, and then we measured the similarity between the resulting representations. Models are evaluated in terms of the Spearman correlation between the similarity scores and the human ratings.

6 Similarly to the Rimell et al. (2016) original paper, we only considered the items contained in the development set.

7 The relation for the head noun is assumed to be the same as the target relation (either *subject* of *direct object* of the relative clause).

3.2 MEDEA settings: data

We used the same corpora both to train the embeddings and to extract the syntactic relations for DEG. The training data comes from the concatenation of three dependency-parsed corpora: BNC (Leech 1992), ukWaC (Baroni et al. 2009) and a 2018 dump of the English Wikipedia, for a combined size of approximately 4 billion tokens. The corpora were parsed with Stanford CoreNLP (Manning et al. 2014).[8]

The embeddings associated to DEG lexical nodes were trained using the same parameters as in Rimell et al. (2016): we created lemmatized 100-dim vectors with *skip-gram* with *negative sampling* (SGNS) (Mikolov et al. 2013), setting minimum item frequency at 100 and context window size at 10.

3.3 MEDEA settings: DEG

We tailored the construction of DEG to the kind of simple syntactic structures required by the datasets (i.e., at most triplets of nodes), restricting it to the case of relations among pairs of event participants.

We included in the graph only events with a minimum frequency of 5 in the training corpora. The edges of the graph were weighted with *Smoothed LMI*. Given a triple composed by the words w_1 and w_2, and a syntactic relation s linking them, we computed its weight by using a smoothed version of the Local Mutual Information (Evert 2004):

$$LMI_\alpha(w_1, w_2, s) = f(w_1, w_2, s) * log(\frac{P(w_1, w_2, s)}{P(w_1) * P_\alpha(w_2) * P(s)}) \tag{4}$$

where the smoothed probabilities are defined as follows:

$$P_\alpha(x) = \frac{f(x)^\alpha}{\sum_x f(x)^\alpha} \tag{5}$$

Local Mutual information is often employed to balance the effects of frequency and to quantify the discrepancy between the chance of co-occurrence of two elements based on their individual and joint distributions. This type of smoothing, with $\alpha = 0.75$, was chosen to mitigate the bias of MI statistical association measures towards rare events (Levy, Goldberg, and Dagan 2015). While this formula only involves pairs (as only pairs were employed in the experiments), it is easily extensible to more complex tuples of elements.

3.3.1 LC

We implemented the additive model as a baseline, by considering only the LC tier of the SR and using addition as a composition function:

8 Note that in Rimell et al. (2016) the training corpus was a 2015 dump of Wikpedia.

$$\overrightarrow{LC} = \sum_{w \in sent} \vec{w} \tag{6}$$

3.3.2 AC content and re-ranking settings

In the present experiments, we did not use the predictions on non-expressed arguments to compute $\overrightarrow{AC}$. Moreover, we built the AC differently in the two tasks:

- as far as RELPRON is concerned, we restricted the evaluation to the representation of the target argument: for example, for the property *document that delegation negotiate*, the $\overrightarrow{AC}(sent)$ only contains the $\overrightarrow{dobj}$ centroid;
- for the transitive sentences similarity dataset, the $\overrightarrow{AC}(sent)$ results from the summation of the centroids corresponding to the overtly filled roles in the sentence (i.e., *sbj*, *root*, *dobj*).

For each word in the dataset items, the top 50 associated words were retrieved from DEG. Both for the re-ranking phase and for the construction of the final representation, the event knowledge vectors (i.e., the role vectors $\overrightarrow{r}$ and the $\overrightarrow{AC}$ vector) are built from the top 20 elements of each weighted list. As detailed in Section 2.2, the ranking process in MEDEA can be performed forward and backward at the same time (i.e., the AC can be used to re-rank newly retrieved information and vice versa, respectively), but for simplicity we only implemented the forward ranking.

3.4 Scoring

We evaluated the performances of the LC component (i.e., our baseline), of the AC component alone and of the whole SR, as a summation of the first two scores.

Thus, in the case of RELPRON, given a $target$ word in a sentence $sent$, the score for MEDEA is computed as a summation of two cosine scores:

$$score(target, sent) = cosine(\overrightarrow{target}, \overrightarrow{LC}(sent)) + cosine(\overrightarrow{target}, \overrightarrow{AC}(sent)) \tag{7}$$

whereas in the case of the transitive sentences similarity dataset, given two sentences s_1, s_2 the score for MEDEA is computed as:

$$score(s_1, s_2) = cosine(\overrightarrow{LC}(s_1), \overrightarrow{LC}(s_2)) + cosine(\overrightarrow{AC}(s_1), \overrightarrow{AC}(s_2)) \tag{8}$$

In all settings, we assume the model to be aware of the syntactic parse of the test items. In the transitive sentences similarity dataset, word order fully determines the syntactic constituents, as the sentences are always in the *subject verb object* order. In RELPRON, on the other hand, the item contains information about the relation that is being tested: in the *subject* relative clauses, the properties always show the *verb* followed by the *argument* (e.g., *telescope: device that detects planets*), while in the *object* relative clauses the properties always present the opposite situation (e.g., *telescope: device that observatory has*).

4. Results and Discussion

4.1 RELPRON

Given the targets and the composed vectors of all the definitions in RELPRON, we assessed the cosine similarity of each pair and computed the Mean Average Precision scores shown in Table 2.

Table 2
The table shows results in terms of MAP for the development subset of RELPRON.

	RELPRON		
	LC	AC	LC+AC
verb	0,18	0,18	**0,20**
arg	0,34	0,34	**0,36**
hn+verb	0,27	0,28	**0,29**
hn+arg	0,47	0,45	**0,49**
verb+arg	**0,42**	0,28	0,39
hn+verb+arg	0,51	0,47	**0,55**

Following the original evaluation in Rimell et al. (2016), we tested six different combinations for each composition model: the verb only, the argument only, the head noun and the verb, the head noun and the argument, the verb and the argument and all three of them. In all cases but one, the models built on the complete SR (i.e., involving the LC level and the AC level) show significant improvements, outperforming the simple additive baseline. Most interestingly, the models involving only the AC tier of the semantic representation still show comparable performances to the baseline.

The only model that lags behind is the *verb+arg* model. As also shown in Rimell et al. (2016), models involving only the sum of lexical vectors show balanced results (Table 3). Things are instead different for the AC component. Here, the composition of event knowledge elicited by verb and argument seems much better at predicting the object than the subject.

Table 3
The table shows MAP results, for each model involving only the LC component, for subject and object relations separately.

LC	verb	arg	hn+verb	hn+arg	verb+arg	hn+verb+arg
subject	0,21	0,44	0,30	0,55	0,49	0,60
object	0,20	0,39	0,30	0,52	0,48	0,59
Δ	0,01	0,06	0,00	0,04	0,01	0,01

Table 4
The table shows MAP results, for each model involving only the AC component, for subject and object relations separately.

AC	verb	arg	hn+verb	hn+arg	verb+arg	hn+verb+arg
subject	0,19	0,41	0,29	0,47	0,22	0,48
object	0,19	0,34	0,29	0,51	0,38	0,52
Δ	0,00	0,06	0,00	-0,04	**-0,16**	-0,04

One relevant parameter of the models is that they work in the linear order in which words are found in the sentence. The *verb+arg* model, therefore, works differently when run on *subject* clauses than on *object* clauses. In the *subject* case, in fact, the verb is found first, and then its expectations are used to reweigh the ones of the object. In the *object* case, on the other hand, things go the opposite way: at first the subject is found, and then its expectations are used to reweigh the ones of the verb (see table 5). When testing the same model, but in reverse order of activation (the second word of the property and then the first one), we find opposite results, with a MAP of 0.41 for *subjects* and 0.21 for *objects*. It seems that, when arguments, which are nouns, are encountered first, event knowledge is more precise and better at predicting the target. This is in line with the fact that *arguments* alone perform better than *roots* alone, and could be related to the fact that verb perform in general distributionally worse than nouns on standard similarity tasks.

Table 5
The table shows the differences between standard linear order (first row) and reverse order (second row) for *subject* and *object* relative clauses. Values in bold refer to the models that show best performances.

	subject clause	*object* clause
$w_1\ w_2$ order	V - O	**S - V**
$w_2\ w_1$ order	**O - V**	V - S

4.2 Transitive sentences dataset

For the transitive sentences dataset, we evaluated the correlation of our scores with human ratings with Spearman's ρ. The similarity between a pair of sentences s_1, s_2 is defined as the cosine between their LC vectors plus the cosine between their AC vectors. We tested seven different combinations for each composition model, evaluating the contribution of each subset of the sentence (i.e., *subject* alone, *verb* alone, *subject+verb*, etc., up to the full sentence).

MEDEA is in the last column of Table 6 and again outperforms simple addition in most cases. Event knowledge alone (i.e., AC column of Table 6) outperforms the baseline in the *sbj* and *root* models, suggesting that information on event knowledge is not properly encoded in distributional vectors, and possibly captures different aspects of compositional meaning. Except for the case of *sbj+root*, the models involving event knowledge in AC always improve the baselines.

5. Conclusion

We provided a basic implementation of a meaning composition model, which aims at being incremental and cognitively plausible. While still relying on vector addition, our results suggest that distributional vectors do not encode sufficient information about event knowledge, and that, in line with psycholinguistic results, activated GEK plays an important role in building semantic representations during online sentence processing.

Our ongoing work focuses on refining the way in which this event knowledge takes part in the processing phase and testing its performance on more complex datasets: while both RELPRON and the transitive sentences dataset provided a straightforward mapping between syntactic label and semantic roles, more naturalistic datasets show a

Table 6
The table shows results in terms of Spearman's ρ on the transitive sentences dataset. *p-values* are not shown because they are all equally significant ($p < 0.01$).

	transitive sentences dataset		
	LC	AC	LC+AC
sbj	0.432	0.475	0.**482**
root	0.525	0.547	0.**555**
obj	0.628	0.537	0.**637**
sbj+root	0.**656**	0.622	0.648
sbj+obj	0.653	0.605	0.**656**
root+obj	0.732	0.696	0.**750**
sbj+root+obj	0.732	0.686	0.**750**

much wider range of syntactic phenomena that would allow us to test how expectations jointly work on the event structure, both at the syntactic level and with respect to the semantic roles filled by participants. Similarly, we will consider more complex tasks such as entailment or inference, for which a variety of datasets are available in literature, in order to evaluate the model's performances on broader language understanding benchmarks.

References

Asher, Nicholas, Tim Van de Cruys, Antoine Bride, and Márta Abrusán. 2016. Integrating Type Theory and Distributional Semantics: A Case Study on Adjective–Noun Compositions. *Computational Linguistics*, 42(4):703–725.

Baroni, Marco, Raffaela Bernardi, and Roberto Zamparelli. 2014. Frege in Space: A Program of Compositional Distributional Semantics. *Linguistic Issues in Language Technology*, 9(6):5–110.

Baroni, Marco, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The WaCky Wide Web: A Collection of Very Large Linguistically Processed Web-Crawled Corpora. *Language Resources and Evaluation*, 43(3):209–226.

Binder, Jeffrey R. 2016. In Defense of Abstract Conceptual Representations. *Psychonomic Bulletin & Review*, 23:1096–1108.

Blacoe, William and Mirella Lapata. 2012. A comparison of vector-based representations for semantic composition. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 546–556, Jeju Island, Korea, July 12-14. Association for Computational Linguistics.

Chersoni, Emmanuele, Alessandro Lenci, and Philippe Blache. 2017. Logical Metonymy in a Distributional Model of Sentence Comprehension. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 168–177, Vancouver, Canada, August 3-4.

Chersoni, Emmanuele, Enrico Santus, Alessandro Lenci, Philippe Blache, and Chu-Ren Huang. 2016. Representing Verbs with Rich Contexts: An Evaluation on Verb Similarity. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1967—1972, Austin, TX, USA, November 1–5.

Coecke, Bob, Stephen Clark, and Mehrnoosh Sadrzadeh. 2010. Mathematical foundations for a compositional distributional model of meaning. Technical report.

Conneau, Alexis, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark, September, 7-11.

Elman, Jeffrey L. 2011. Lexical knowledge without a lexicon? *The mental lexicon*, 6(1):1–33.

Erk, Katrin and Sebastian Padó. 2008. A structured vector space model for word meaning in context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 897–906, Honolulu, Hawaii, USA, October 25–27. Association for Computational Linguistics.

Evert, Stefan. 2004. *The Statistics of Word Cooccurrences Word Pairs and Collocations*. Ph.D. thesis, University of Stuttgart.
Hagoort, Peter. 2013. MUC (Memory, Unification, Control) and Beyond. *Frontiers in Psychology*, 4(JUL):1–13.
Hagoort, Peter. 2016. MUC (Memory, Unification, Control): A Model on the Neurobiology of Language beyond Single Word Processing. In Gregory Hickok and Steve Small, editors, *Neurobiology of Language*, volume 28. Elsevier, Amsterdam, pages 339–347.
Hagoort, Peter and Jos van Berkum. 2007. Beyond the sentence given. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1481):801–811.
Hare, Mary, Michael Jones, Caroline Thomson, Sarah Kelly, and Ken McRae. 2009. Activating Event Knowledge. *Cognition*, 111(2):151–167.
Jackendoff, Ray. 1997. *The Architecture of the Language Faculty*. MIT Press, Cambridge, MA.
Kartsaklis, Dimitri and Mehrnoosh Sadrzadeh. 2014. A study of entanglement in a categorical framework of natural language. In *Proceedings of the 11th Workshop on Quantum Physics and Logic (QPL)*. Kyoto, Japan, 4-6th June.
Kiros, Ryan, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-Thought Vectors. In *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, pages 3294–3302.
Kuperberg, Gina R. and T. Florian Jaeger. 2016. What do we mean by prediction in language comprehension? *Language, cognition and neuroscience*, 31(1):32–59.
Leech, Geoffrey Neil. 1992. 100 Million Words of English: The British National Corpus (BNC).
Lenci, Alessandro. 2018. Dynamic Distributional Semantics. Unpublished manuscript.
Levy, Omer, Yoav Goldberg, and Ido Dagan. 2015. Improving Distributional Similarity with Lessons Learned from Word Embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.
Manning, Christopher, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of ACL 2014, the 52nd Annual Meeting of the Association for Computational Linguistics (System Demonstrations)*, pages 55–60, Baltimore, MD, USA, 22-27 June.
McRae, Ken and Kazunaga Matsuki. 2009. People Use their Knowledge of Common Events to Understand Language, and Do So as Quickly as Possible. *Language and Linguistics Compass*, 3(6):1417–1429.
Melamud, Oren, Ido Dagan, Jacob Goldberger, Idan Szpektor, and Deniz Yuret. 2014. Probabilistic modeling of joint-context in distributional similarity. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 181–190, Baltimore, Maryland, USA, 26-27 June.
Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26 (NIPS 2013)*, pages 3111–3119, Lake Tahoe, NV, USA, December 5–10.
Mitchell, Jeff and Mirella Lapata. 2008. Vector-based models of semantic composition.
Partee, Barbara H. 1984. Compositionality. In *Varieties of Formal Semantics*. Foris, Dordrecht, pages 281–311.
Rimell, Laura, Jean Maillard, Tamara Polajnar, and Stephen Clark. 2016. RELPRON: A Relative Clause Evaluation Data Set for Compositional Distributional Semantics. *Computational Linguistics*, 42(4):661–701.