MEDEA: Merging Event knowledge and Distributional vEctor Addition

Ludovica Pannitto
CoLing Lab, University of Pisa
ellepannitto@gmail.com

Alessandro Lenci
CoLing Lab, University of Pisa
alessandro.lenci@unipi.it

Abstract

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

Italiano. *La maggior parte dei modelli proposti nell'ambito della semantica disribuzionale composizionale si basa sull'utilizzo dei soli vettori lessicali. Proponiamo di arricchire il miglior modello presente in letteratura (la somma di vettori, che consideriamo come baseline) con informazione distribuzionale sugli eventi elicitati dalla frase, migliorando sistematicamente i risultati della baseline.*

1 Compositional Distributional Semantics: Beyond vector addition

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 et al., 2014; Coecke et al., 2010), as well as neural networks approaches (Socher et al., 2010; Mikolov et al., 2013).

Among all proposed compositional functions, vector addition still shows the best performances on various tasks (Asher et al., 2016; Blacoe and Lapata, 2012; Rimell et al., 2016), beating more complex methods, such as the Lexical Functional Model (Baroni et al., 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.

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 speakers' 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).

In this paper, we introduce **MEDEA**, a compositional distributional model of sentence meaning which integrates vector addition with GEK activated by lexical items. MEDEA is directly inspired by the model in Chersoni et al. (2017a) 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.

We test MEDEA on two datasets for compositional distributional semantics in which addition has proven to be very hard to beat. At least, before meeting MEDEA.

2 Introducing MEDEA

MEDEA consists of two main components: i.) a **Distributional Event Graph** (DEG) that models a fragment of semantic memory activated by lexical units (Section 2.1); ii.) 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

We assume a broad notion of *event*, corresponding to any **configuration of entities, actions, properties, and relationships**. Accordingly, an event can be a complex relationship between entities, as the one expressed by the sentence *The student read a book*, but also the association between an individual and a property, as expressed by the noun phrase *heavy book*.

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 cued by all the potential participants to the event.

The nodes of DEG are lexical embeddings, and edges link lexical items participating to the same events (i.e., its syntagmatic neighbors). Edges are weighted with respect to the statistical salience of the event given the item. Weights, expressed in terms of a statistical association measure such as *Local Mutual Information*, determine the event activation strength by linguistic cues.

In order to build DEG, we automatically harvested events from corpora, using syntactic relations as an approximation of semantic roles of event participants. From a dependency parsed sentence we identified an event by selecting a semantic head (verb or noun) and grouping all its syntactic dependents together (Figure 1). Since we expect each participant to be able to trigger the event and consequently any of the other participants, a relation can be created and added to the graph from each subset of each group extracted from sentence.

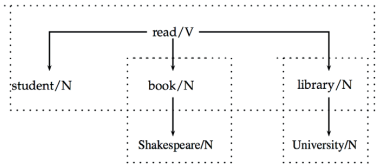

Figure 1: Dependency analysis for the sentence *The student is reading the book about Shakespeare in the university library*. Three events are identified (dotted boxes).

The resulting structure is therefore 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.

As graph nodes are embeddings, given a lexical cue w, DEG can be queried in two modes:

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

para. neighbors	essay/N, anthology/N, novel/N, author/N, publish/N, biography/N, autobiography/N, nonfiction/N, story/N, novella/N
synt. neighbors	publish/V, write/V, read/V, include/V, child/N, series/N, have/V, buy/V, author/N, contain/V

Table 1: The 10 nearest paradigmatic (top) and syntagmatic (bottom) neighbours of *book/N*, extracted from DEG. By further restricting the query on the graph neighbors, we can obtain for instance typical subjects of *book* as a direct object (*people/N, child/N, student/N, etc.*).

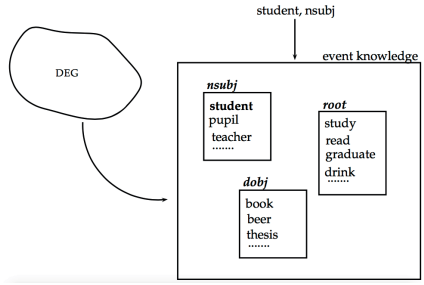

Figure 2: The image shows the internal architecture of a piece of EK retrieved from DEG. The interface with DEG is shown on the left side of the picture, each internal list of *neighbors* is labeled with their expected syntactic role in the sentence. All the items are intended to be embeddings.

2.2 Meaning Composition Function

In MEDEA, we model sentence comprehension as the creation of a semantic representation SR, which includes two different yet interacting information tiers that are equally relevant in the overall representation of sentence meaning: i.) the *lexical meaning* component (LM), which is a context-independent tier of sentence meaning that accumulates the lexical content of the sentence, as traditional models do; ii.) an *active context* (AC), which aims at representing the most probable event, in terms of its participants, that can be reconstructed from DEG portions cued by lexical items. This latter component corresponds to the GEK activated by the single lexemes (or by other contextual elements) and integrated into a semantically coherent structure representing the sentence interpretation. It is incrementally updated during processing, when a new input is integrated into existing information.

2.2.1 Active Context

Each lexical item in the input activates a portion of GEK that is integrated into the current AC through a process of mutual re-weighting that aims at maximizing the overall semantic coherence of the SR.

At the outset, no information is contained in the AC of the sentence. When new *lexeme - syntactic role* pair $\langle w_i, r_i \rangle$ (e.g., *student - nsbj*) are encountered, expectations about the set of upcoming roles in the sentences are generated from DEG (figure 2). These include: i.) expectations about the role filled by the lexeme itself, which consists of its vector (and possibly its *p-neighbours*); ii.) expectations about sentence structure and other participants, which are collected in weighted list of vectors of its *s-neighbours*.

These expectations are then weighted with respect to what is already in the AC, and the AC is similarly adapted to the ewly retrieved information: each weighted list is represented with the weighted centroid of its top elements, and each element of a weighted lists is re-ranked according to its cosine similarity with the correspondent centroid (e.g., the newly retrieved weighted list of *subjects* is ranked according to the cosine similarity of each item in the list with the weighted centroid of *subjects* available in AC).

The final semantic representation of a sentence consists of two vectors, the **lexical meaning vector** ($\overrightarrow{LM}$) and the **event knowledge vector** ($\overrightarrow{AC}$), which is obtained by composing the weighted centroids of each role in AC.

3 Experiments

3.1 Datasets

We wanted to evaluate the contribution of activated event knowledge in a sentence comprehension task. For this reason, among the many 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 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 and test set, made up by a *target* noun labeled with a syntactic role (either *subject* or *direct object*) and a *property* expressed as *[head noun] that [verb] [argument]*. For instance, here are some example properties for the target noun *treaty*:

(1) a. OBJ treaty/N: document/N that delegation/N negotiate/V
b. SBJ treaty/N: document/N that grant/V independence/N

Transitive sentence similarity 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 (2) and low (3) similarity scores respectively:

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

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

3.2 Graph implementation

We tailored the construction of the DEG to this kind of simple syntactic structures, restricting it to the case of relations among pairs of event participants. Relations were automatically extracted from a 2018 dump of Wikipedia, BNC, and ukWaC corpora, parsed with the Stanford CoreNLP Pipeline (Manning et al., 2014).

Each $\langle(word_1, word_2), (r_1, r_2)\rangle$ pair was then weighted with a smoothed version of Local Mutual Information[1]:

$$LMI_\alpha(w_1, w_2, r_1, r_2) = f(w_1, w_2, r_1, r_2) log(\frac{\hat{P}(w_1,w_2,r_1,r_2)}{\hat{P}(w_1)\hat{P}_\alpha(w_2)\hat{P}(r_1,r_2)}) \quad (1)$$

where:

$$\hat{P}_\alpha(x) = \frac{f(x)^\alpha}{\sum_x f(x)^\alpha} \quad (2)$$

Each lexical node in DEG was then represented with its embedding. We used the same training parameters as in Rimell et al. (2016),[2], since we wanted our model to be directly comparable with their results on the dataset. While Rimell et al. (2016) built the vectors from a 2015 download of Wikpedia, we needed to cover all the lexemes contained in the graph and therefore we used the same corpora from which the DEG was extracted.

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.[3] Likewise, each sentence of the transitive sentences dataset is a triplet $((w_1, nsbj), (w_2, root), (w_3, dobj))$.

3.3 Active Context implementation

In MEDEA, the SR is composed of two vectors:

- $\overrightarrow{LM}$, as the sum of the word embeddings (as this was the best performing model in literature, on the chosen datasets);
- $\overrightarrow{AC}$, obtained by summing up all the weighted centroids of triggered participants. Each *lexeme - syntactic role* pair is used to retrieve its 50 top s-neighbors from the graph. The top 20 re-ranked elements were used to build each weighted centroid. These threshold were choosen empirically, after a few trials with different (i.e., higher) thresholds (as in Chersoni et al. (2017b)).

We provide an example of the re-weighting process with the property *document that store maintains*, whose target is *inventory*: i.) at first the head noun *document* is encountered: its vector is activated as event knowledge for the *object* role of the sentence and constitutes the contextual information in AC against which GEK is re-weighted; ii.) *store* as a subject triggers some *direct object* participants, such as *product*, *range*, *item*, *technology*, etc. If the centroid were built from the top of this list, the cosine similarity with the target would be around 0.62; iii.) *s-neighbours* of *store* are re-weighted according to the fact that AC contains some information about the target already, (i.e., the fact that it is a document). The re-weighting process has the effect of placing on top of the list elements that are more similar to *document*. Thus, now we find *collection*, *copy*, *book*, *item*, *name*, *trading*, *location*, etc., improving the cosine similarity with the target, that goes up to 0.68; iv.) the same happens for *maintain*: its *s-neighbors* are retrieved and weighted against the complete AC, improving their cosine similarity with *inventory*, from 0.55 to 0.61.

3.4 Evaluation

We evaluated our model on RELPRON development set using Mean Average Precision (MAP), as

[1]The smoothed version (with $\alpha = 0.75$) was chosen in order to alleviate PMI's bias towards rare words (Levy et al., 2015), which arises especially when extending the graph to more complex structures than pairs.

[2]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]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).

in Rimell et al. (2016). We produced the compositional representation of each property in terms of SR, and then ranked for each target all the 518 properties of the dataset portion, according to their similarity to the target. Our main goal was to evaluate the contribution of event knowledge, therefore the similarity between the target vector and the property SR was measured as the sum of the cosine similarity of the target vector with the $\overrightarrow{LM}$ of the property, and the cosine similarity of the target vector with the $\overrightarrow{AC}$ cued by each property. As shown in Table 2, the full MEDEA model (last column) achieves top performance, above the simple additive model LM.

	RELPRON		
	LM	AC	LM+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**

Table 2: The table shows results in terms of MAP for the development subset of RELPRON. Except for the case of verb+arg, the models involving event knowledge in AC always improve the baselines (i.e., LM models).

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 LM vectors plus the cosine between their EK vectors. MEDEA is in the last column of Table 3 and again outperforms simple addition.

	transitive sentences dataset		
	LM	AC	LM+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**

Table 3: The table shows results in terms of Spearman's ρ on the transitive sentences dataset. Except for the case of sbj+root, the models involving event knowledge in AC always improve the baselines. *p-values* are not shown because they are all equally significant ($p < 0.01$).

4 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 straight forward mapping between syntactic label and semantic roles, more naturalistic datasets show a much wider range of syntactic phenomena that would allow us to test how expectations jointly work on syntactic structure and semantic roles.

References

Nicholas Asher, 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.

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

William Blacoe 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. Association for Computational Linguistics.

Emmanuele Chersoni, Alessandro Lenci, and Philippe Blache. 2017a. Logical metonymy in a distributional model of sentence comprehension. In *Sixth Joint Conference on Lexical and Computational Semantics (* SEM 2017)*, pages 168–177.

Emmanuele Chersoni, Enrico Santus, Philippe Blache, and Alessandro Lenci. 2017b. Is structure necessary for modeling argument expectations in distributional semantics? In *12th International Conference on Computational Semantics (IWCS 2017)*.

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

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

Katrin Erk 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. Association for Computational Linguistics.

Peter Hagoort and Jos van Berkum. 2007. Beyond the sentence given. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1481):801–811.

Mary Hare, Michael Jones, Caroline Thomson, Sarah Kelly, and Ken McRae. 2009. Activating event knowledge. *Cognition*, 111(2):151–167.

Dimitri Kartsaklis 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.

Gina R Kuperberg and T Florian Jaeger. 2016. What do we mean by prediction in language comprehension? *Language, cognition and neuroscience*, 31(1):32–59.

Alessandro Lenci. 2018. Distributional Models of Word Meaning. *Annual Review of Linguistics*, 4:151–171.

Omer Levy, 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.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Ken McRae 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.

Tomas Mikolov, 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*, pages 3111–3119.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition.

Laura Rimell, 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.

Richard Socher, Christopher D Manning, and Andrew Y Ng. 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*, volume 2010, pages 1–9.