UNIVERSITÀ DI PISA

DIPARTIMENTO DI FILOLOGIA, LETTERATURA E LINGUISTICA

CORSO DI LAUREA MAGISTRALE IN INFORMATICA UMANISTICA

TESI DI LAUREA

Event Knowledge in Compositional Distributional Semantics

RELATORE
Prof. Alessandro Lenci
CONTRORELATORE
Dott. Felice Dell'Orletta

CANDIDATA
Ludovica Pannitto

ANNO ACCADEMICO 2017/2018

Contents

1 | Introduction

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.

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 sentence 1 it refers to an airplane while in sentence 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.

The aim of this work is to investigate the use of distributional methods in a model of compositional meaning that is both **linguistically motivated** and **cognitively inspired**.

Chapter 2 provides a brief overwiev of **Distributional Semantics**, a usage based model of meaning that has been widely applied to cognitive tasks, such as judgements on semantic similarity or identification of semantic or analogical relations between words. Distributional semantics also provides a natural framework for the computational implementation of linguistic theories that rely on the claims that context of use is a proxy to linguistic meaning.

The fact that the meaning of words can be inferred from use is something intuitively true, which can be demonstrated with a set of examples like the following ones[1]:

(3) He handed her glass of *bardiwac*.

(4) Beef dishes are made to complement the *bardiwacs*.

(5) Nigel staggered to his feet, face flushed from too much *bardiwac*.

(6) Malbec, one of the lesser-known *bardiwac* grapes, responds well to Australia's sunshine.

(7) I dined off bread and cheese and this excellent *bardiwac*.

(8) The drinks were delicious: blood-red *bardiwac* as well as light, sweet Rhenish.

Although we could not possibly know the referential meaning of the word *bardiwac*, we can provide quite a few inferences: it is probably a liquid substance (sentence 3), usually consumed during meals (sentences 4-7), it is probably alcoholic (sentence 5), and so on. If one were presented with this set of sentences, and then asked what *bardiwac* means, the most likely answer would probably be something like *a kind of wine*. And these sentences were in fact created by substituting the word *claret*, a french wine, with the fake word *bardiwac*.
One of the main criticism often posed to the distributional theories is that of being

[1] Examples from Stefan Evert, handpicked and edited from the British National Corpus

primarly a theory of lexical meaning. The issue of compositionality and complex meaning representation has a long-standing history in linguistics, and various theories about how complex meaning is derived have been proposed.
Because of the traditional distinction between syntax and semantics, a *syntactically transparent semantic composition* theory has been the most widely accepted and employed.
Following this statement:

- all elements of content in the meaning of a sentence are found in the semantic representations of the lexical items composing the sentence;
- the way the semantic representations are combined is a function only of the way the lexical items are combined in syntactic structure. In particular, the internal structure of individual semantic representations plays no role in determining how the semantic representations are combined, and pragmatics plays no role in determining how semantic representations are combined.

Distributional semantics has approached 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. Chapter 3 provides a short review of available models for compositionality within the distributional semantic framework.

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 speaker's expectations about the upcoming input.
The metaphor of the lexicon as a dictionary is no longer suitable, and augmenting the lexicon with structures like Pustejovsky's *qualia* has been proven not to be a feasible option: these are not flexible enough to account for the wide variety of interpretations that can be retrieved (Lascarides and Copestake, 1998; Zarcone et al., 2014) and that are influenced by factors such as the subject choice, the general syntactic and discourse context, and by our *world knowledge*. Chapter 4 summarizes some of the theories in the **Generalized Event Knowledge** framework that aims at overcoming such limitations.

Based on the importance of event knowledge in the comprehension process, we developed a general framework that allows for the integration of this *generalized event knowledge* into standard compositional distributional models, such as the sum model. Our architecture, described in chapter 5, is fairly customizable, and generally based on a twofold structure: a *storage component*, called DEG (distributional event graph), which is a repository of both traditional lexical meanings and knowledge coming from events, and an *activation and integration component*, which accounts for meaning composition.

Chapter 6 is concerned with the evaluation process. We aimed at evaluating the contribution of activated event knowledge in a sentence comprehension task. For this reason, we implemented a reduced version of the hypothesized framework and, among the many existing datasets concerning entailment or paraphrase detection (a brief review is provided in section 3.2), 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 in fact an intermediate level of grammatical complexity, as they involve complete sentences (while other datasets only consider smaller phrases), while being composed of fixed length sentences with similar syntactic constructions (i.e., transitive sentences). The two datasets differ with respect to size and construction method.
RELPRON was also the object of a pilot study, described in the same chapter, whose aim was to validate the dataset and enrich it with human judgements, thereby providing a different perspective on model testing.
Some more in-depth qualitative discussion on the models is reported in chapter 7, in particular with respect to RELPRON dataset, which is the one we developed our models on.

Chapter 8 draws some conclusions on the work and suggest further developments and analysis that are needed in order to gain a deeper insight on the treatment of compositionality within distributional semantics.

2 | Meaning and Linguistic Contexts

2.1 Distributional Semantics, meaning from usage

Distributional semantics is a usage-based model of meaning, lying on the assumption that *meaning* in language is an abstraction over the linguistic contexts in which linguistic items are used.
Relying on the hypothesis that the semantic similarity of lexical items is a function of their distribution in linguistic contexts, a mathematical encoding of the hypothesis is easily provided by means of a vectorial representation of co-occurrences.
This kind of representations, thanks to their scalar properties, have proven themselves to be able to at least partially overcome some of the well known problems left out of the picture by formal theories of meaning representation. These can be collectively addressed as issues concerning the relationship between *word meaning* and *word usage in context*, and include a great amount of phenomena, ranging from meaning acquisition to logical metonymy, that have long been considered at the periphery of linguistic competence, while being central to how speakers actually use language to convey meaning.

2.2 Distributional Hypothesis

The theoretical foundations of distributional semantics lie in the so called *Distributional Hypothesis* (henceforth, DH), namely the fact that

> *”lexemes with similar linguistic contexts have similar meaning”* (Lenci, 2008).

While the most famous formulation of the DH is to be found in Firth (1957),

> *"you shall know a word by the company it keeps"*,

the first complete theorization of the distributional method is to be found in Harris (1954). Harris, claiming that similarity in distributions should be taken as an *explanans* for meaning, provides a solid empirical methodology for the semantic analysis, including meaning among the entities that are part of linguistic studies.

Distributionalism as a more general theory of meaning has anyway broader foundations, influenced also by Wittgenstein (1953) and well developed within behavioral psychology (e.g., according to Deese (1966), meaning is acquired thanks to association or co-occurrence of stimuli) and cognitive sciences: here, studies such as Rubenstein and Goodenough (1965) showed how similarity judgements and linguistic contexts overlap significantly. Miller and Charles (1991), in a similar distributional fashion, claim that

> *"words contextual representation is an abstract cognitive structure that accumulates from encounters with the word in various (linguistic) contexts. Two words are semantically similar to the extent that their contextual representations are similar"*.

Given this twofold nature of the DH, as an empirical methodology as well as a cognitive hypothesis, Lenci (2008) distinguishes between a **weak** and a **strong** version of the DH.
The former is in fact taken as a quantitative method for semantic analysis: as in Harris distributional procedures, a certain correlation between semantic content and linguistic distribution is assumed, thus allowing the *weak* DH to coexist within many research programs in theoretical linguistics (e.g., distributional methodologies were employed within the theory of the *Generative Lexicon* by Pustejovsky (1991)).
The *strong* DH deals instead with the possibility that the form and origin of semantic representations is distributional: in this approach contexts have a specific role in the formation of cognitive representations. This version of the DH pairs with

usage-based theories of meaning. Quoting Miller and Charles (1991), the distributional behaviour of lexemes has a role in explaining the semantic cognitive content, as

> *"what people know when they know a word is not how to recite its dictionary definition they know how to use it (when to produce it and how to understand it) in everyday discourse"*.

In this stronger formulation, the distributional hypothesis is able to address the problem of language comprehension as a whole, also being concerned with how linguistic knowledge interacts with perceptual or non-linguistic reasoning, whose behaviour could be acquired and modeled distributionally as well.
Some recent research goes towards a distributional and yet more comprehensive model of language, tackling issues such as multi-modality (Feng and Lapata, 2010; Bruni et al., 2014) and the integration of non-linguistic distributional information into distributional semantic models, In this area, many models aim at enriching linguistic representations with visual information, paving the way for more and more complex and multidimensional models of meaning.

2.3 Distributional representations

While symbolic semantic representations are discrete and categorical, distributional representations are **graded** (words differ not only in the contexts in which they appear, but also in the salience of these contexts) and **distributed** (semantic properties derive from global vector comparison).

The most popular framework that implements DH for semantic analysis is based on *Vector Space Models* (Salton et al., 1975), first developed in Information Retrieval (IR) in the early '60s.
This was introduced to represent documents in terms of the words they are composed of, statistically weighted upon their relevance, thus allowing to measure document similarity in terms of distance in the vector space, but was later employed also to determine term similarity in order to perform IR tasks such as query expansion. Terms are here considered similar if they tend to appear together in documents.

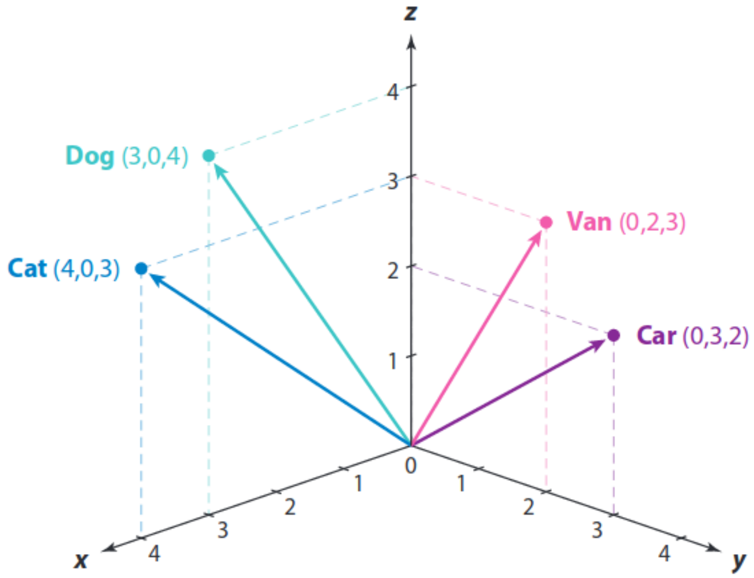

Figure 2.1: Image from Lenci (2018), showing distributional vectors of the lexemes *car*, *cat*, *dog* and *van*. Each dimension (in this example only 3 dimensions are considered) represents a relevant context for the linguistic items. Similarity among lexemens is intuitively related to their proximity in vector space.

Classical **Distributional Semantic Models** (henceforth: DSMs), also known as **Matrix Models** or **Count Models**, generalize the idea of Salton et al. (1975) Vector Space Model: a lexeme is represented as a $n-$dimensional vector, where the distributional components are features representing co-occurrences among linguistic contexts (see table 2.1).

Although table 2.1 shows raw frequencies, for exemplification reasons, these turn out to be far from the optimal solution to estimate the salience of linguistic contexts. In order to overcome this problem, which is due to the natural distribution of linguistic input, distributional models employ several refined statistical measures to weigh co-occurrences and get higher scores for more informative contexts.

	bite	buy	drive	eat	get	live	park
van	0	9	0	0	12	0	8
car	0	13	8	0	15	0	5
dog	6	0	0	9	10	7	0
cat	0	0	0	6	8	3	0

Table 2.1: The table shows a co-occurrence matrix. Each lexical item, on the rows, is represented by a $7-$dimensional vector, where each dimension is labeled with a relevant linguistic context selected during the modeling phase.

The correspondence between distributional features and linguistic context can be bijective in principle, where each vector component can correspond to a distinct context, although in practice this methodology shows some drawbacks due to the so-called *curse of dimensionality*. Linguistic contexts tend, in fact, to be very sparse and objects in the vector space end up being almost equidistant, making it difficult to discriminate similar from dissimilar ones. Moreover, these explicit vectors fail at taking into consideration the fact that context themselves are similar to one another, missing some crucial conceptual generalization. For these reasons, implicit vector have soon been introduced (Deerwester et al., 1990), where lexemes are represented by dense, low dimensional vectors composed of latent features extracted from co-occurrences, through dimensionality reduction techniques.

When applied to very large matrices, dimensionality reduction algorithms such as Singular Value Decomposition (SVD) can be computationally quite onerous. A major issue concernes the fact that these algebraic techniques rely on the global statistics collected in the co-occurrence matrix. If new distributional data is added, the whole semantic space must be built again from scratch, making the model unfeasible for an incremental study.
Still in the domain of count DSMs, there are also **Random Encoding Models** (Kanerva et al., 2000; Sahlgren, 2006; Jones and Mewhort, 2007; Recchia et al., 2010): rather than collecting global co-occurrence statistics into a matrix and then

optionally reducing them to dense vectors, they directly learn low-dimensional implicit vectors by assigning each lexical item a random vector that is incrementally updated depending on the co-occurring contexts.

A whole different approach to learning distributional vectors are **Prediction Models**: instead of counting co-occurrences, prediction DSMs (also commonly known as *word embeddings*) are created through network algorithms that provide low-dimensional, implicit distributional representations.

These algorithms learn how to optimally predict the context of the target item (*Skip-Gram with Negative Sampling* (SGNS), figure 2.2) or the item vector based on the context (*Continuous Bag of Words* (CBOW), figure 2.3), both introduced in Mikolov et al. (2013a).

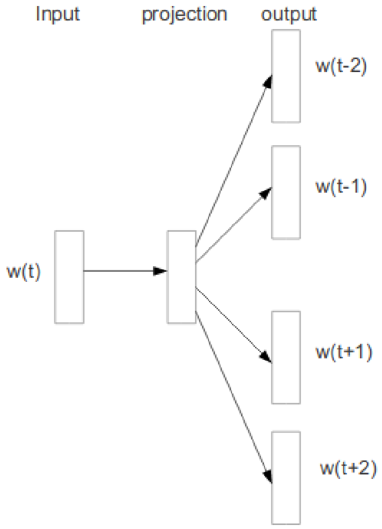

Figure 2.2: Image from Mikolov et al. (2013a), showing the skip-gram model architecture. The training objective is to learn word vector representations that are good at predicting the nearby words.

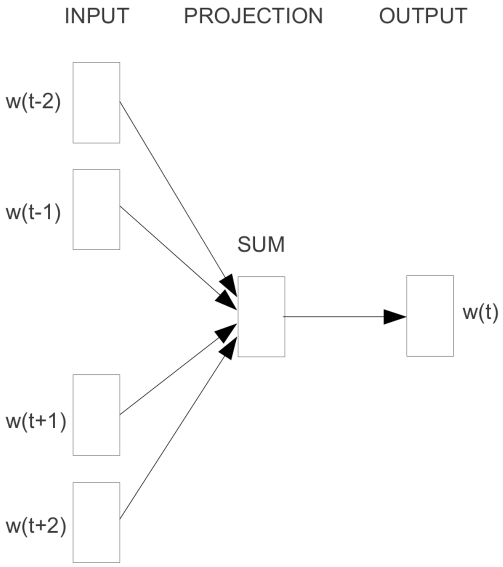

Figure 2.3: Image from Mikolov et al. (2013a), showing the cbow model architecture. As in the skip-gram case, the training objective is to learn word vector representations that are good at predicting the nearby words.

Starting from Mikolov et al. (2013a), many other architectures for the creation of dense word vectors have been introduced in literature, such as *GloVe* (*Global Vectors*) (Pennington et al., 2014), a popular implementation that produces dense vectors based on a weighted least squares model trained on a global matrix of word-word co-occurrence counts, and the recently introduced *FastText* (Bojanowski et al., 2017), which exploits the idea of learning word representations as the sum of the embeddings of the character n-grams composing them.

Despite their increasing popularity, due to the fact that various types of ”linguistic regularities” have been claimed to be identifiable by neural embeddings (see figure 2.4), the question whether neural embeddings are really a breakthrough with respect to more traditional DSMs is far from being set (Levy and Goldberg, 2014), largely depending also on the dataset size (Sahlgren and Lenci, 2016).

A summary of vectorial distributional representations is provided in figure 2.5.

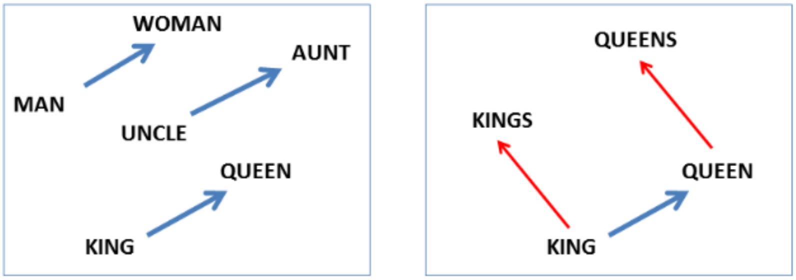

Figure 2.4: Image from Mikolov et al. (2013b), left panel shows vector offsets for three word pairs illustrating the gender relation. Right panel shows a different projection, and the singular/plural relation for two words

2.3.1 Beyond vectorial DSMs

Most approaches to distributional semantics rely on dense vector representations. Apart from technical reasons mainly related to available libraries and efficiency of computations, there seems to be no theoretical reason to rely solely on vector space models.

Some efforts towards the de-biasing of DSMs from this assumption have been made, with the creation of a graph based distributional model (Biemann et al., 2013; Biemann, 2016).

2.4 Parameters for distributional representations

DSMs can be seen as tuples $< T, C, W, S >$, where:

- T are the target elements (i.e., the lexemes for which the DSM provides a representation);
- C are the linguistic contexts with which T co-occur;
- W is a context weighting function in the case of count models, or the objective function in the case of predictive models;
- S is a similarity measure between the produced representations.

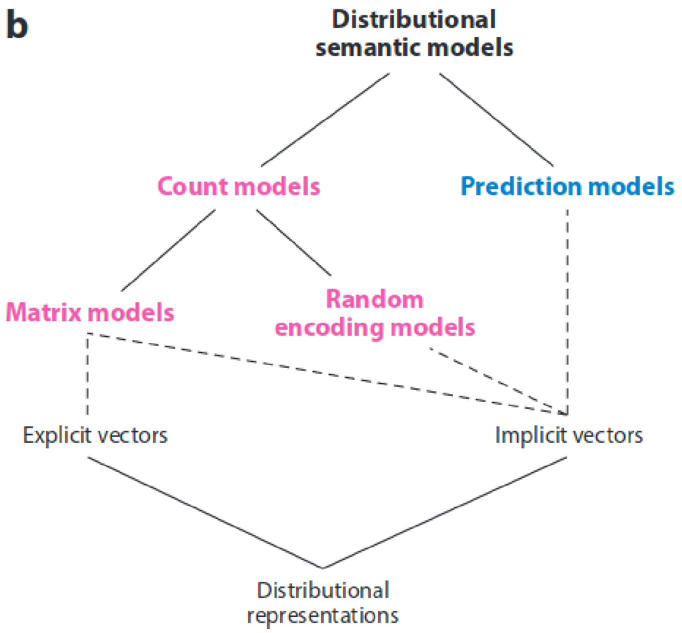

Figure 2.5: Image from Lenci (2018), showing how different DSMs get categorized with respect to the different approaches presented.

Each of the component of the distributional model introduces a certain amount of variability, which makes it possible to tailor the construction of the DSM to specific purposes.

2.5 Semantic relations in DSMs

Unsupervised methods which induce vector representations from large textual corpora coalesce several types of information, in particular not being able to distinguish notions of semantic **similarity** and **relatedness**.
As distributional semantics is being applied to a broader range of tasks than lexical representation, a more fine grained distinction between different kinds of relations would be beneficial in order to provide the right kind of inference depending on the context imposed by the task.

While many datasets have been released in order to evaluate distributional models, the relationship between semantic similarity and DSMs is much more complex than it appears.

If not explcitally built in order to avoid this, DSMs typically provide high similarity scores for words that are semantically similar (i.e., *car* and *van*) as well as for words that are semantically related or associated (i.e., *car* and *driver*).
Just mentioning two among the most popular evaluation datasets commonly employed for evaluating the performances of distributional models (WS-353 by Finkelstein et al. (2001) and MEN by Bruni et al. (2012)), these have been shown (Hill et al., 2015) to measure word association rather than proper similarity, and they both seem to suffer from the fact that dissimilar pairs (i.e., non-synonyms such as antonyms) receive high ratings because of the instructions provided to the human annotators, while associated but dissimilar concepts do not receive low ratings (i.e., *summer* and *drought* are rated 7.16 out of 10).

Moreover, while state-of-the-art models are able to match human performances on the available datasets, the ability of DSMs to distinguish between different lexical relations (i.e, hypernymy, antonymy, meronymy as well as non standard semantic relations, which in turn show very different inferential properties) is far from being set.
As pointed out in Lenci (2018), the outcome of DSMs resembles a *network of word associations*, rather than a semantically structured space, which is an important shortcoming for applications and for linguistic research as well, as the limitations in properly distinguishing different semantic relation influences the ability to model logical inferences, which are crucial to human reasoning (Erk, 2016).

On the opposite side, formal approaches to meaning representation and composition provide fine grained distinctions among different kinds of semantic relations (i.e., lexical ontologies) and mathematically well-defined frameworks to perform inferences. The most promising perspective seems to be the integration of distributional information with symbolic models, in order to merge the greater flexibility demostrated by statistically-induced representations with the rich inferential power of formal structures.

3 | Composition

3.1 Compositional Distributional Semantics

The representation of complex linguistic expression is a challenging and widely debated topic for distributional semantics. The entire set of approaches to the representation of complex meaning in distributional space can be roughly split in two families:

- the first group of approaches is concerned with **extending** the distributional hypothesis to broader portions of text;
- the second group proposes to obtain a representation of the distributional meaning of a complex expression by **composing** a lower-level object, such as the distributional representation for word-units.

Concerning the first family, we cite the work from Lin and Pantel (2001) as an example. However interesting, this kind of approach shows many drawbacks from our perspective: the first is a computational one, as the arbitrary length of sequences results in data sparseness and less reliable estimates. Moreover, it is a linguistically and cognitively implausible approach, which does not take into account well established principles of comprehension such as the most accredited theories of conceptualization and compositionality.

As far as the second family of approaches is concerned, a general framework for vector-based models of compositionality is presented in Mitchell and Lapata (2008). In their framework, semantic composition is defined primarily as a function of two vectors, $\vec{u}$ and $\vec{v}$. These two constituents will stand in some semantic relation R. Moreover, while processing a complex structure, we might add any amount of additional knowledge, which is represented by another component K. The general class of models for this process of composition is therefore expressed by the relation:

$$\vec{p} = f(\vec{u}, \vec{v}, R, K) \tag{3.1}$$

Through the variation of these parameters, the framework expresses a wide variety of models, in the spirit of using distributional information to build meaningful representations for more complex linguistic structures.

While common methods for combining vectors involve simple operation such as sum or average, which are insensitive to word order or syntactic structure, they point out the importance of syntactic relations for sentence and discourse processing, as well as sentence priming or inference tasks.

3.1.1 Full-algebraic approaches

A first set of approaches to the issue of compositionality in the distributional semantic space can be regarded as a set of purely algebraic techniques, which phrase the problem of meaning composition as the composition of content vectors only. The result of the composition function is therefore itself a vector living in the same space of the original vectors.

Although effective in practice, as shown in Blacoe and Lapata (2012) and many other works, these models have many limitations from the linguistic point of view.

Additive model

In the additive model, the composition is obtained through linear combination of word vectors.

$$p = Au + Bv \tag{3.2}$$

where A and B are weight matrices or, in the simplified version

$$p = \alpha u + \beta v \tag{3.3}$$

α and β are weights generally set to 1.

The resulting vector can be seen as a sort of union vector obtained by summing up the relevant contexts for the word vectors.
Several issues may arise concerning this method:

- intuitively, the meaning of a complex expression is not easily approximated by the general meaning of its parts. This is due to the polysemous nature of lexemes: when composing a lexical item in context, we select the relevant traits of the item, thus ideally excluding some aspects from the resulting meaning;

- the method does not, in principle, take into account word order, while the semantic operation of composition certainly does: this effect can be mitigated via the introduction of coefficients in order to allow for different weighting of items;

- depending on the techniques used to build the vector space, different type of words may end up living in different sub-spaces of the semantic space: summing different vectors together may result in a vector concatenation operation.

Zanzotto et al. (2010) introduce a novel approach for estimating parameters for additive compositional distributional semantics.
The generic additive model is defined as

$$\vec{z} = A\vec{x} + B\vec{y} \tag{3.4}$$

where A and B which determine the contributions of u and v to p, and should therefore capture the syntactic relation and any background knowledge needed.
Their approach focuses on solving a regression model with multiple dependent variables: given a set of training examples E of triples $(\vec{x}, \vec{y}, \vec{z})$, we can write the basic regression equation as:

$$\vec{z} = (A, B) \begin{pmatrix} \vec{x} \\ \vec{y} \end{pmatrix} \tag{3.5}$$

The approximate solution is computed through Least Square Estimation.

Multiplicative model

As the additive model behaves like a union over the salient contexts, the idea behind the multiplicative model is the one of intersection between contexts.

$$p = Cuv^T \tag{3.6}$$

where C is a weight tensor that projects the uv tensor product into the space of p.

In a simplified version, the operation is performed component-wise:[1]

$$p_i = u_i \times v_i \tag{3.7}$$

this way the salience of a context for one item involved in the composition functions as a weight on the other item.

A major issue comes from the same observations reported above: when applying the multiplicative composition to vectors living in different subspaces of the vector space, the result will be null or too sparse since there are no or few common contexts.

Another issue, pointed out in Bride (2017), derives from the fact that the vector obtained through component-wise multiplication depends on the base of the vector space: while this could be an interesting point from the linguistic perspective, it certainly arises some problems from the mathematical one, as many mathematical operations on vector spaces derive from the fact that we assume the possibility to change the base of the representation.

[1] This is equivalent to performing the outer product of the two vectors u and v and considering C as the projection of the main diagonal.

Matrix models

Rudolph and Giesbrecht (2010) propose a *Compositional Matrix-Space Model* (CMSMs) which uses matrices instead of vectors and matrix multiplication as the one and only composition operation (see figure 3.1).

$$\begin{tikzcd}
\sigma_1 \arrow[rrrr, bend left, "\text{concatenation } \cdot"] \arrow[d, "{[\![\cdot]\!]}"] & \sigma_2 \arrow[rrr, bend left] \arrow[d, "{[\![\cdot]\!]}"] & \cdots & \sigma_n \arrow[r, bend left] \arrow[d, "{[\![\cdot]\!]}"] & \sigma_1\sigma_2\ldots\sigma_n \arrow[d, "{[\![\cdot]\!]}"] \\
{[\![\sigma_1]\!]} \arrow[rrrr, bend right, "\text{composition } \bowtie"'] & {[\![\sigma_2]\!]} \arrow[rrr, bend right] & \cdots & {[\![\sigma_n]\!]} \arrow[r, bend right] & {[\![\sigma_1\sigma_2\ldots\sigma_n]\!]}
\end{tikzcd}$$

Figure 3.1: Image from Rudolph and Giesbrecht (2010). Given a mapping $|\cdot| : \Sigma \rightarrow S$ from a set of tokens (words) Σ into some semantical space S, the semantic composition operation $\bowtie$: $S* \rightarrow S$ maps sequences of semantic objects into semantic objects, such that the meaning of a sequence of tokens can be obtained by applying $\bowtie$ to the sequence of semantic objects, thus creating an isomorphism between $(\Sigma*, \cdot)$ and $(S*, \bowtie)$.

The model is said to be both algebraically plausible, as matrix multiplication is associative but non-commutative, and cognitively or psychologically plausible, as mental states are represented as vectors of numerical values and changes in mental states as linear applications that shift the vectors. A semantic space of matrices can thus be seen as a function space of mental transformations.
Rudolph and Giesbrecht show how CMSMs is able to cover a variety of distributional and symbolic aspects of natural languages, yet crucial questions remain open: besides hand-crafted encodings, it must be shown how to acquire CMSMs for large token sets, and non-linear aspects cannot be expressed by matrices, that narrow the possibilities down to linear mappings.

3.1.2 Taking order into consideration

One of the main issues when dealing with distributional models of meaning and compositionality is that of taking into account the order in which words appear in the sentence.
As stated about the problem of compositionality itself, integrating syntactic information into vector spaces seems to be a fundamental step in order to achieve good results in compositionality.
Many approaches have been proposed to integrate this kind of information, and here we present a few of them.

In Giesbrecht (2010) a matrix-based distributional model is proposed, in order to overcome limitations due to word order encoding in vector space.
Given a vocabulary V, a context window $w = m$ and a series of tokens $t_1, ...t_1 \in V$, for a token t_i a matrix of size $V \times V$ is generated that has nonzero values in cells where t_i appears between t_{i-m} and t_{i+m}.
This procedure defines a tensor T where $T(i, j, k)$ is the number of occurrences of $L(j)sL(i)s'L(k)$ in the corpus, where s and s' are sequences of at most $w - 1$ tokens.
Giesbrecht (2010) provides the following example: given a corpus made up by the following three sentences,

(9) Paul kicked the ball slowly.

(10) Peter kicked the ball slowly.

(11) Paul kicked Peter.

the matrix representation for the item *kick* is shown in table 3.1, assuming a window of size 1 and prior stop-words removal.

The obtained space is clearly very sparse, so dimensionality reduction needs to be performed in order to obtain feasible objects.

A different approach is presented in Jones and Mewhort (2007) and in Sahlgren et al. (2008), inspired by the former.
The method introduced by Jones and Mewhort, named BEAGLE (*Bound Encoding of the Aggregate Language Environment*), represents each word with three

KICK	PETER	PAUL	KICK	BALL	SLOWLY
PETER	0	0	0	1	0
PAUL	1	0	0	1	0
KICK	0	0	0	0	0
BALL	0	0	0	0	0
SLOWLY	0	0	0	0	0

Table 3.1: The table provides the matrix representation obtained for the lexical item *kick*. Each cell represents the occurrence of a left and right context for the item.

memory vectors: one encoding co-occurrences, one encoding word order, and the last one combining the two. The vectors are computed through *environmental auxiliary vectors*, which are random vectors whose components are normally distributed i.i.d. (independent and identically distributed) random variables.
We report an example from the authors: consider the sentence

(12) a dog bit the mailman

and *dog* as the target word.
When processing the sentence, we would have:

- ***dog***, the environmental vector set once at the beginning of the procedure;
- ***_dog_***, the *context* information for the present sentence;
- <***dog***>, the *order* information for the present sentence;
- ***_DOG_***, the vector accumulating *context* information;
- <***DOG***>, the vector accumulating *order* information;
- ***DOG***, the final memory vector for *dog*;

The context information is computed like standard co-occurrence information, thus registering co-occurrences from the current sentence, normalizing it and adding to the *context* vector.
The order information is computed by adding up all the possible n-grams vectors

obtained from the sentence through a convolution operation[2] between a placeholder vector fixed in advance and vectors from co-occurring words.

At the end of the procedure, the vector ***DOG*** is obtained summing up the two vectors ***_DOG_*** and <***DOG***>, and it's sensitive to both word order and proximity.

Similarly to Jones and Mewhort, in Sahlgren et al. (2008) *Random Indexing* is used to get *environment* vectors and *permutations* are used as convolution operations. These have the advantage of being easier to compute. Moreover, Sahlgren et al. claim that replacing products with sums gives a better representation of word order since it allows the vectors to tolerate similarity between slightly varied sentences (e.g. *dog bit a man* vs. *dog bit the man*).

3.1.3 Introducing structure in vector space

The model presented in Erk and Padó (2008) stems again from the failure to consider syntax as a relevant part in the representation of meaning in vector space. The authors underline two important aspects: in the first place, the syntactic relation between the target u and its context v is often ignored during the construction of the vector space and the composition operation. In the second place the result of a vector composition operation is itself a vector: Erk and Padó claim that this might not be the best choice because single vectors can only encode a fixed amount of information and thus may not be able to scale up to the meaning of entire phrases or sentences.
Their proposal is a *structured vector space* model for word meaning: the representation for a lemma comprises several vectors representing the word's lexical meaning along with its *selectional preferences*. The meaning of a word u in a context v is computed combining u with the selectional preferences expressed by v, which are specific to the relation holding between the two items. The contextualized vectors can then be combined with a representation of the structure's expression, such as a parse tree, to address the issue of compositionality.

[2] A *convolution operation* is a mathematical operation between functions. In this case, it is used in the form of *circular convolution* in order to project back the tensor product between two vectors in the original vector space.

Their intuition is to view the interpretation of a word in context as guided by expectations about typical events linked to that word. This is a well established line of research both in cognitive science (situation models) and computational linguistics (selectional preferences and selectional restrictions).
Each word is encoded (see figure 3.2) as a combination of:

- a vector that models the lexical meaning of the word;
- a set of vectors, each representing the semantic expectations for each particular relation supported by the word.

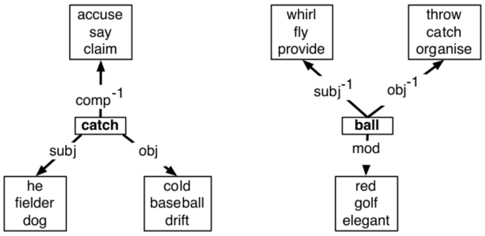

Figure 3.2: Image from Erk and Padó (2008). Lexical information is enriched with selectional preferences modeled through syntactic relations.

Formally, let D be a vector space and L a set of relation labels, the meaning of a lemma w would be a triple (v, R, R^{-1}) where $v \in D$ is a standard lexical vector, $R : L \rightarrow D$ maps each relation into a vector that describes the selectional preferences of w and $R^{-1} : L \rightarrow D$ maps from labels to inverse selectional preferences of w.
Referring to figure 3.2, the meaning of *ball* would be composed of the vector of *ball* (bold in the picture), its selectional preferences associtated to the syntactic label *mod* (i.e., *red, golf, elegant*) and its inverse selectional preferences as a *subject* (i.e., *whirl, fly, provide*) and as an *object* (i.e., *throw, catch, organize*).
Given $a = (v_a, R_a, R_a^{-1})$, $b = (v_b, R_b, R_b^{-1})$ and $l \in L$ the relation holding between the two, it is possible to compute the contextualized representation of the two vectors as $a' = (v_a * R_b^{-1}(l), R_a - \{l\}, R_a^{-1})$ and $b' = (v_b * R_a(l), R_b, R_b^{-1} - \{l\})$ where $*$ is a standard composition operation between vectors such as sum or component-wise multiplication (see figure 3.3).

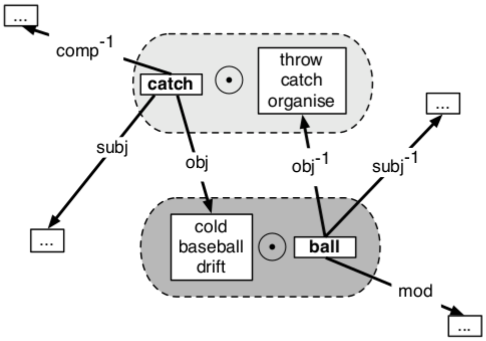

Figure 3.3: Image from Erk and Padó (2008). Predicate and argument are combined via relation-specific semantic expectations. In this example, the representation for *catch* and the representation for *ball* are contextualized in the compositional segment *catch ball*. The selectional preferences of *catch* for the label *obj* are composed with the distributional vector of *ball*, while the inverse selectional preferences for *ball* are composed with the distributional vector for *catch*. The selectional preferences for the *object* relation are then removed from the representation of *ball* and *catch*.

Along with this line of research Erk and Padó (2010) argue that the problem of representing contextualized meaning is closely related to the problem of *concept combination* in cognitive science: many approaches are based on prototype models, and type vectors can be regarded as an implementation of the prototype-theory. Erk and Padó propose instead an **exemplar model**, which memorizes each encountered instance of a category and uses the current context in order to activate relevat exemplars.
Each target is therefore represented by a set of exemplars (i.e. all the sentences in which the target occurs). Given a set of exemplars of a lemma E, relevant exemplars are selected as follows:

$$act(E, s) = \{e \in E | sim(e, s) > \theta(E, s)\} \tag{3.8}$$

where s is the point of comparison, sim is a similarity function and θ is a threshold function.
As pointed out by Erk and Padó, a major drawback to this approach is the consistent computational overhead needed during the comparisons.

3.1.4 Neural Networks

A different kind of approach is the one described in Socher et al. (2012): a compositional vector representation for sequences of arbitrary length is constructed via a recursive neural network that operates over a parse tree.
Each constituent of the expression (words in the first step, internal nodes afterwards) is associated with both a matrix and a vector. The latter captures the lexical meaning of the constituent, while the former is used to modify the words it combines with, as a function.

The model is inspired by the work of Socher et al. (2011), where each parent vector of the parse tree is obtained through the formula $p = g\left(W\begin{bmatrix}a\\b\end{bmatrix}\right)$, where W is a matrix and g is a non-linear function. The representation for the entire phrase is then built up recursively, by navigating upwards the parse tree.
Socher et al. extend this idea by assigning a matrix to each word and learning a non-linear composition function for any syntactic type.
This approach appears able to approximate the behavior of words which function mainly as operators and lack a proper semantic representation, and the behavior of *content* words as well.
The final function proposed by the authors is the following:

$$p = f_{A,B}(a,b) = f(Ba, Ab) = g\left(W\begin{bmatrix}Ba\\Ab\end{bmatrix}\right) \tag{3.9}$$

where A, B are matrices for single words and W maps both transformed words in the same n-dimensional space.
After two words are merged in the parse tree, the new constituent can be merged with another one by applying the same functions (see figure 3.4).

3.1.5 Formal methods with distributional meaning

Coecke et al. (2010) propose a framework based on the algebra of *pregroups*, as introduced by Lambek (1997), in order to unify the distributional theory of meaning and the compositional theory of grammatical types.

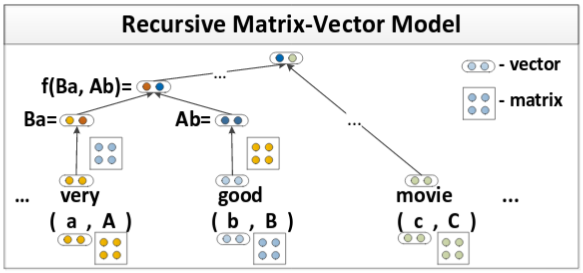

Figure 3.4: Image from Socher et al. (2012), showing how intermediate nodes of the parse tree are composed. The whole sentenc is represented by the root of the parse tree.

The use of pregroups is motivated by the common structure they share with vector spaces and tensor products: this makes it possible to join methods from both formal and distributional semantics, as the meaning of words are represented by vectors whereas grammatical roles are types in the pregroup.
Pregroups formalize the grammar of natural language in the same way as type-categorial logics do. One starts by fixing a set of basic grammatical roles and a partial ordering between them. A type is assigned to each word of the dictionary and reduction rules are applied to get one type from the composition of others. The model is therefore category-theoretic, as this allows us to store additional information needed for a quantitative approach, and to reason about grammatical structures of a phrase.
The authors provide two examples (reported below), for which they fix the following basic types, from which compound types are formed by adjoints of juxtaposition:

- n: noun;
- j: infinitive of the verb;
- s: declarative statement;
- σ: glueing type.

If the juxtaposition of the types of the words within a sentence reduces to the basic type s, then the sentence is said to be grammatical.

The sentence *John likes Mary* has the following type assignment[3]:

- *John*: n
- *likes*: $n^r s n^l$
- *Mary*: n

and therefore reduces to the type s as follows (diagrammatically shown in figure 3.5):

$$n(n^r s n^l)n \to 1 s n^l n \to 1 s 1 \to s \tag{3.10}$$

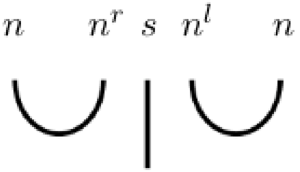

Figure 3.5: Image from Coecke et al. (2010), showing the reduction diagram for the sentence *John likes Mary*.

They similarly provide a reduction for the sentence *John does not like Mary*, for which we only show the diagrammatical reduction in figure 3.6.

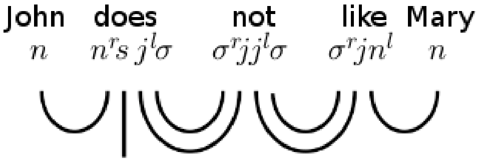

Figure 3.6: Image from Coecke et al. (2010), showing the reduction diagram for the sentence *John does not like Mary*.

Asher et al. (2017) present a method to integrate *Type Composition Logic* (TCL, as defined in Asher (2011)), which provides a formal model of the interaction between composition and lexical meaning, with distributional semantics.

[3] n^r and n^l are left and right adjoints for an element n

In TCL, a logical form is assigned to the target combination, where each unit is contextualised through a logical function depending on the others. As an example, let's consider the combination of *heavy* with *traffic*. In the TCL formalism, this would result in a logical form such as:

$$\lambda x.(O(\text{heavy})(x) \wedge M(\text{traffic})(x)) \tag{3.11}$$

where O is a functor induced by the noun and M does the same for the adjective. Functors are meant to contextualize the lexical meaning of the items. TCL however does not provide any method to construct functors or lexical meaning, and that's where distributional semantics comes in handy.

Like most formal semantic theories, TCL distinguishes between the *external content* of a word, which is the appropriate extension at time of evaluation, and a *type* or *internal content*, which can be regarded as the proper lexical meaning of the word, or the set of features associated with it.

In the model proposed by Asher et al. (2017), each *type* is an algebraic object built through distributional semantics: individual types will be vectors, while functors will be transformations that allow for vector or type shifts.

Asher et al. (2017) present two methods to compute a contextualised weighting of lexical units: the first one, latent vector weighting, is based on a matrix factorization technique in which a latent space is constructed and used to determine which dimensions are important for a particular expression, while the second one is based on tensor factorization.

Another effort in bridging formal and distributional semantics comes from Clark and Pulman (2007) and is furtherly elaborated in Clark et al. (2008): here the authors analyze the work of Smolensky and Legendre (2006), which aimed at integrating the connectionist and the symbolic models of comprehension. A symbolic structure s is defined by a collection of structural roles r_i each of which may be occupied by a filler f_i, s is therefore a set of constituents, each being a filler-role binding f_i/r_i. In the connectionist approach, roles and fillers are vectors. Smolensky and Legendre propose a tensor product representation to obtain a connectionist vector for the whole symbolic structure.

Clark and Pulman use this same approach to combine symbolic and distributional models of meaning, where the symbolic structure is a sort of representation of

the sentence, such as a parse tree or a set of predicate-argument relations, and the distributional representation is a set of context vectors.
The final representation is obtained through a tensor product between roles and fillers, whose order is determined by some traversal of the symbolic structure.
For example, for the parse tree in figure 3.7, the tensor product proposed to obtain the sentence meaning is the following:

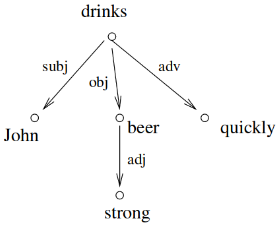

Figure 3.7: Image from Clark and Pulman (2007), showing the parse tree for the sentence *John drinks strong beer quickly*.

$$\text{drinks} \otimes \text{subj} \otimes \text{John} \otimes \text{obj} \otimes \left(\text{beer} \otimes \text{adj} \otimes \text{strong}\right) \otimes \text{adv} \otimes \text{quickly} \tag{3.12}$$

assimung that vectors for dependency relations (i.e., *sbj, obj*, etc...) form an orthonormal basis in a ”*relation space*”.
This allows us to encode order information in the final representation (i.e., the representation for *dog bites man* is different from the one for *man bites dog*), but leaves some issues open, such as how to obtain vectors for the symbolic relations (e.g. dependency relations such as *sbj*, *obj*...): these are supposed to be an orthonormal basis in a *relation space*, but this leaves space to another major drawback, namely the fact that the semantic comparison between two sentences only makes sense if the two sentences can be represented by the same parse tree, therefore living in the same tensor space, and this makes the model less flexible.

3.1.6 Lexical Function Model

Baroni and Zamparelli (2010) propose a model for *adjective-noun* composition, representing nouns as vectors and adjective as matrices, which can be interpreted as functions over vectors.

Adjectives in attributive position are considered functions from one meaning to another, that is to say linear maps from n-dimensional vectors to n-dimensional vectors. Each adjective is therefore encoded in a weight matrix, which multiplies the noun vector $\vec{v}$ in the formula

$$\vec{p} = B\vec{v} \tag{3.13}$$

The matrices are estimated through partial least square regressions, approximating target adj-N vectors automatically extracted from the corpus.

The model, which can be expressed by means of the framework introduced in Mitchell and Lapata (2008), is only apparently simple: in the first place each adjective matrix is trained separately, requiring a great deal of time or computational resources to obtain the complete model, moreover the model is not easily scalable to more complex composition operation. While it is valuable in cases such as affixation or *determiner + N* cases, it would be difficult to apply it to different sentential structures such as *verbs + arguments*. This results in both linguistics and computational pitfalls, as each verb can vary in the number of constituents it requires, and building larger constituents would end up in data sparsity problems.

In order to overcome issues such as the need for an individual function for each adjective, Bride et al. (2015) attempt a generalization of the Lexical Function Model introduced by Baroni and Zamparelli. They automatically learn a *generalized lexical function* which has to be composed with both the adjective vector and the noun vector in order to obtain the phrase representation.

The generalized lexical function would be a tensor $\mathcal{A}$, which is multiplied by the vector for the adjective and the vector for the noun following. The product between the tensor and the vector produces a matrix, which would be the LF matrix for the adjective.

As a practical matter, the problem can be formalized as follows:

$$\text{Find } \mathcal{A} \text{ so that: } \sum_{adj}(\mathcal{A} \times adj - ADJ)^2 \text{ is minimal} \tag{3.14}$$

therefore the learning phase still needs to acquire a certain number of matrices for adjectives.

Paperno et al. (2014) try to overcome some issues of Lexical Function Models introducing the *Practical Lexical Function Model*: the representation for a semantic representation for a unit is an ordered tuple of a vector and n matrices. Matrices are used in substitution for tensors in lexical function models, as they encode the *arity* of the linguistic unit: each matrix corresponds to a function-argument relation and words have as many matrices as many arguments they take. In this framework semantic composition is obtained through two composition rules: *function application* and *simmetric composition*.

The former is employed when it is necessary to handle structures with different *arity* (e.g., verb plus noun). The latter applies when two syntactic sisters are of the same *arity* (such as the noun-noun composition): this simply sums the two objects in a new tuple.

As an example (figure 3.8) let's consider the sentence *user writes software*: in the practical lexical function model, *user* and *software* would be just vectors, while *write* will be represented by a tuple made up of a vector and two matrices, one for the subject role and the other for the object role. The composition would proceed as follows: in the first step consists of transforming *write* into the resulting vector obtained by summing up the vector of *write* with the product $write_s \times \overrightarrow{user}$, where $write_s$ is the matrix of *write* for the subject role. This new *write* needs now to be composed with *software* in order to obtain the final vector.

While overcoming many of the pitfalls of lexical function model, this approach still presents some issues, both from the computational and the theoretical side: Gupta et al. (2015) acknowledge some inconsistencies in the method proposed by Paperno et al., which lead to overestimate the predicate lexical contribution to the composite meaning. Paperno et al. (2014) propose two ways of re-establishing the correct influence of the predicate, and show that a more balanced compositional function turns out to be more effective.

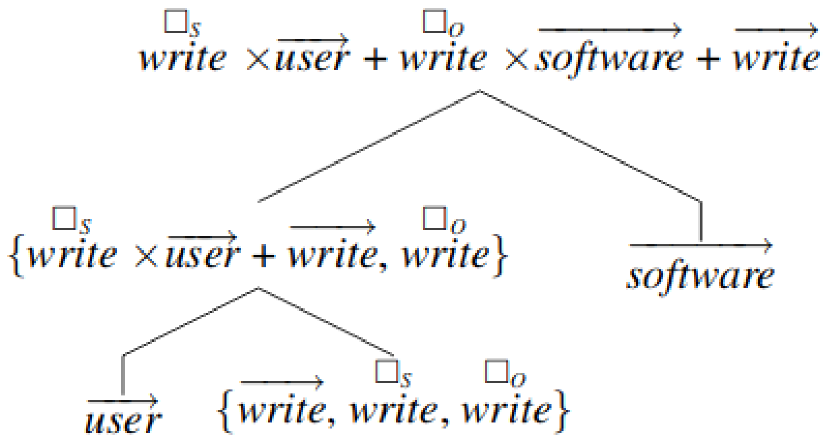
$\overset{\square_S}{write} \times \overrightarrow{user} + \overset{\square_O}{write} \times \overrightarrow{software} + \overrightarrow{write}$
$\{\overset{\square_S}{write} \times \overrightarrow{user} + \overrightarrow{write}, \overset{\square_O}{write}\}$
$\overrightarrow{software}$
$\overrightarrow{user}$
$\{\overrightarrow{write}, \overset{\square_S}{write}, \overset{\square_O}{write}\}$

Figure 3.8: Image from Gupta et al. (2015), showing the Practical Lexical Function model derivation for the noun-verb-noun phrase *user writes software*.

3.2 Evaluating Compositional Distributional Semantics

In order to compare different approaches to the composition problem, it is essential to provide datasets that allow for evaluation at various levels of complexity and make it possible to address the variety of phenomena that can arise during a compositional process.

The range of proposed datasets is wide and tackles precise grammatical phenomena as well as general relatedness between full complex sentences. Here we offer a brief review of the most widely employed ones.

Most of the mentioned datasets collect similarity judgments elicited from human annotators. It must be noticed that, altough humans can reliably judge wether two phrases are similar, their agreement tends to be lower compared to judgments for simple word pairs.

3.2.1 Compositional phrases

While compositional structures have a long-standing history in distributional semantics, the first standard evaluation framework for compositional structures is probably the one proposed in Mitchell and Lapata (2008), derived from Kintsch (2001), where the authors predict human similarity judgements on noun-verb phrases. The similarity mainly depends on the verb sense disambiguation in context: for example, *the sales slumped* should be more similar to *the sales declined* than to *the sales slouched*, whereas *the shoulder slumped* is judged more similar to *the shoulders slouched* than to *the shoulders declined*.
Mitchell and Lapata (2010) extend the dataset to a wider range of phenomena, including adjective-noun and noun-noun, along with verb-object pairs, and focus more specifically of phrase similarity.

Still addressing small compositional phrases, the following datasets address more specifically the distinction between compositional and non-compositional meaning.
Reddy et al. (2011) introduce, motivated by the task proposed in Biemann and Giesbrecht (2011), a dataset of noun-noun pairs, evaluating both the literality of the phrase and the extent to which the use of each component was literal within the proposed phrases, thus attempting to provide a dataset which has both scalar compositionality judgments of the phrase as well as the literality score for each component word. table 3.2 shows some examples from the dataset.

Compound	**Word1**	**Word2**	**Phrase**
swimming pool	4.80 ± 0.40	4.90 ± 0.30	4.87 ± 0.34
spelling bee	4.81 ± 0.77	0.52 ± 1.04	2.45 ± 1.25
cloud nine	0.47 ± 0.62	0.23 ± 0.42	0.33 ± 0.54

Table 3.2: Phrases showing high, medium and low compositionality ratings from the dataset proposed in Reddy et al. (2011)

Similarly, Boleda et al. (2012, 2013) focus on intersective adjectives in compositional phrases, providing items such as *white towel*, compositional and involving and intersective adjective, and *false floor* or *black hole*, where the meaning is not strictly compositional and the adjective does not behave intersectively.

Bernardi et al. (2013) focus instead on noun paraphrasing, investigating in particular the semantics of determiner phrases when they act as components of the stable and generic meaning of a content word (as opposed to situation-dependent deictic and anaphoric usages). For example a *trilogy* has to be recognized as a series of *three books*. Their datasets is made of more than 200 nouns (the final set contains 173 nouns and 23 determiner phrases) strongly related to a determiner phrase. Matching each noun with its associated DP (target DP), two *foil* DPs sharing the same noun as the target but combined with other determiners (same-N foils), one DP made of the target determiner combined with a random noun (same-D foil), the target determiner (D foil), and the target noun (N foil). Examples are shown in table 3.3.

noun	**target DP**	**same-N foil 1**	**same-N foil 2**	**same-D foil**	**D foil**	**N foil**
duel	two opponents	various opponents	three opponents	two engineers	two	opponents
homeless	no home too	few homes	one home	no incision	no	home
polygamy	several wives	most wives	fewer wives	several negotiators	several	wives
opulence	too many goods	some goods	no goods	too many abductions	too many	goods

Table 3.3: Examples from the noun-DP relatedness benchmark Bernardi et al. (2013)

3.2.2 Small, complete sentences

Similarly to Mitchell and Lapata (2008) noun-verb pairs, Grefenstette and Sadrzadeh (2011) and Kartsaklis and Sadrzadeh (2014) introduced a task for full, yet simple transitive sentences, involving *subject-verb-object* triples. Sentence pairs were rated by humans: some of the ratings are summarized in table 3.4.

Another dataset presenting simple yet complete sentences is RELPRON Rimell et al. (2016), which focuses on relative clauses deawing the attention to a composition phenomenon that involves functional words (i.e., *that*) as well as lexical ones.

	sentence 1	**sentence 2**	**score (avg)**
high score	medication achieve result	drug produce effect	6.16
	pupil achieve end	student reach level	6.08
medium score	commitee consider matter	study pose problem	3.16
	company suffer loss	firm cause injury	3.13
low score	program offer support	road cross line	1.00
	people remember name	physician pass time	1.00

Table 3.4: High, medium and low similarity ratings of sentence pairs from the dataset proposed in Kartsaklis and Sadrzadeh (2014)

The dataset consists of 1087 manually selected relative sentences, paired with a target that the sentences are supposed to paraphrase. An example is given in table 3.5.

OBJ	cell/N: room/N that prison/N have/V
SBJ	study/N: room/N that contain/V bookshelf/N
OBJ	bull/N: mammal/N that rodeo/N feature/V
SBJ	horse/N: mammal/N that pull/V wagon/N

Table 3.5: Examples of subject and object relative sentences extracted from the dataset RELPRON.

3.2.3 Full sentence similarity and entailment

Larger datasets for sentence similarity and entailment have been developed mainly for SemEval or *SEM Shared Tasks.

The first pilot task on semantic textual similarity (STS) was proposed at SemEval 2012 Agirre et al. (2012), collecting examples from the Microsoft Research Paraphrase dataset (Dolan et al., 2004), the Microsoft Research Video Paraphrase Corpus (Chen and Dolan, 2011) and the translation shared task of the 2007 and 2008 ACL Workshops on Statistical Machine Translation (WMT) (Callison-Burch et al., 2007, 2008), along with glosses from pairs of glosses from OntoNotes 4.0

and WordNet 3.1.
The dataset items were judged through the Amazon Mechanical Turk platform (examples in 3.2.3, in decreasing order of similarity).

(13) a. The bird is bathing in the sink.
b. Birdie is washing itself in the water basin.

(14) a. In May 2010, the troops attempted to invade Kabul.
b. The US army invaded Kabul on May 7th last year, 2010.

(15) a. John said he is considered a witness but not a suspect.
b. He is not a suspect anymore. John said.

(16) a. They flew out of the nest in groups.
b. They flew into the nest together.

(17) a. The woman is playing the violin.
b. The young lady enjoys listening to the guitar.

(18) a. John went horse back riding at dawn with a whole group of friends.
b. Sunrise at dawn is a magnificent view to take in if you wake up early enough for it.

The same data was used for the 2013 *SEM shared task Agirre et al. (2013), where also a more specific task investigating the reason or type of similarity was introduced.

The SICK dataset (Bentivogli et al., 2016), used in SemEval-2014 shared task on compositional distributional semantics (Marelli et al., 2014), consisting of about 10,000 English sentence pairs annotated for relatedness in meaning and entailment, was created including general knowledge about concepts and categories, but avoiding encyclopedic knowledge about specific instances, which requires the identification of phenomena such as named entities, that are a non-central issue to compositionality.

Each pair in the SICK data set is rated with respect to:

- meaning *relatedness* between the two sentences;
- meaning *entailment* between the two sentences;

resulting in a relatedness score and three possible labels for entailment (examples are given in table 3.6):

entailment - if sentence A is true, sentence B is true
contradiction - if A is true, then B is false
neutral - the truth of B cannot be determined on the basis of A

Sentences	Entailment Label	Relatedness Score
Two teams are competing in a football match Two groups of people are playing football	ENTAILMENT	4.7
The brown horse is near a red barrel at the rodeo The brown horse is far from a red barrel at the rodeo	CONTRADICTION	3.6
A man in a black jacket is doing tricks on a motorbike A person is riding the bicycle on one wheel	NEUTRAL	3.7
A boy is playing a keyboard A boy is standing in a room by a lamplight	NEUTRAL	2.1

Table 3.6: Examples of SICK sentences pairs with their gold entailment labels and gold relatedness scores (measurements are intended out of 5)

More specifically addressing *entailment* and *contradiction* is the Stanford Natural Language Inference (SNLI) Corpus (Bowman et al., 2015), which provides a larger (570k human-written) and more ecological set of sentence pairs, improving the control over event and entity coreference, which was identified as a problem in the existing datasets.
SNLI was created though a crowdsourcing platform, where each worker was presented with premise scene descriptions from a pre-existing corpus, and asked to provide hypotheses for each of the three possible labels (entailment, neutral and contradiction). A portion of the obtained pairs was then validated following the same procedure as the SICK dataset (examples shown in table 3.7).

Text	**Judgements**	**Hypothesis**
A man inspects the uniform of a figure in some East Asian country.	contradiction C C C C C	The man is sleeping
An older and younger man smiling.	neutral N N E N N	Two men are smiling and laughing at the cats playing on the floor
A black race car starts up in front of a crowd of people.	contradiction C C C C C	A man is driving down a lonely road.
A soccer game with multiple males playing.	entailment E E E E E	Some men are playing a sport.
A smiling costumed woman is holding an umbrella.	neutral N N E C N	A happy woman in a fairy costume holds an umbrella.

Table 3.7: Example pairs taken from the devtest portion of the SNLI corpus. Each pair has the judgments of five workers and a consensus judgment (if any of the three labels was chosen by at least three of the five workers, it was chosen as the consensus label).

Among these datasets, also the one presented in Cheung and Penn (2012) and Pham et al. (2013) are worth mentioning, as they restrict the task to specific syntactic or semantic phenomena such as nominalization or word order variation. In the Pham et al. (2013) dataset, for example, for the sentence *A man plays a guitar* a paraphrase such as *The man plays a guitar* is provided, as well as a foil sentence such as *A guitar plays a man*.

4 | Generalized Event Knowledge

Research in semantics has for a long time been addressing the problem of compositionality as a distinction between two classes of sentences, namely those that do not violate any combinatorial constraint and are therefore suitable for a semantic interpretation (see example 19), and those that are judged as semantically impossible or not interpretable, because a semantic violations occurs (see example 20).

(19) The musician plays the flute in the theater.

(20) * The nominative plays the global map in the pot.

The first class, however, consists of a great amount of phenomena, coalescing in particular typical (example 19) and atypical sentences (see example 21), whose status aroused much interest in recent experimental research[1].

(21) The gardener plays the castanets in the cave.

The classical binary classification has long been supported by a two-step process for language interpretation: a first step concerned the computation of the meaning of the sentence in a context-free manner, solely based on syntactic and *core* lexical knowledge, the second step concerned the integration of meaning with discourse or world-knowledge information. This model has however been proved wrong by experimental researches on event-related potentials (ERP) (Hagoort and van Berkum, 2007), that showed how both semantic violations and world-knowledge or discourse violations are marked by a greater amplitude of the N400 component[2] (see figure 4.1).

[1] Examples are taken from Chersoni (2018).

[2] As the degree of semantic fit between a word and its context increases, the amplitude of the N400 goes down.

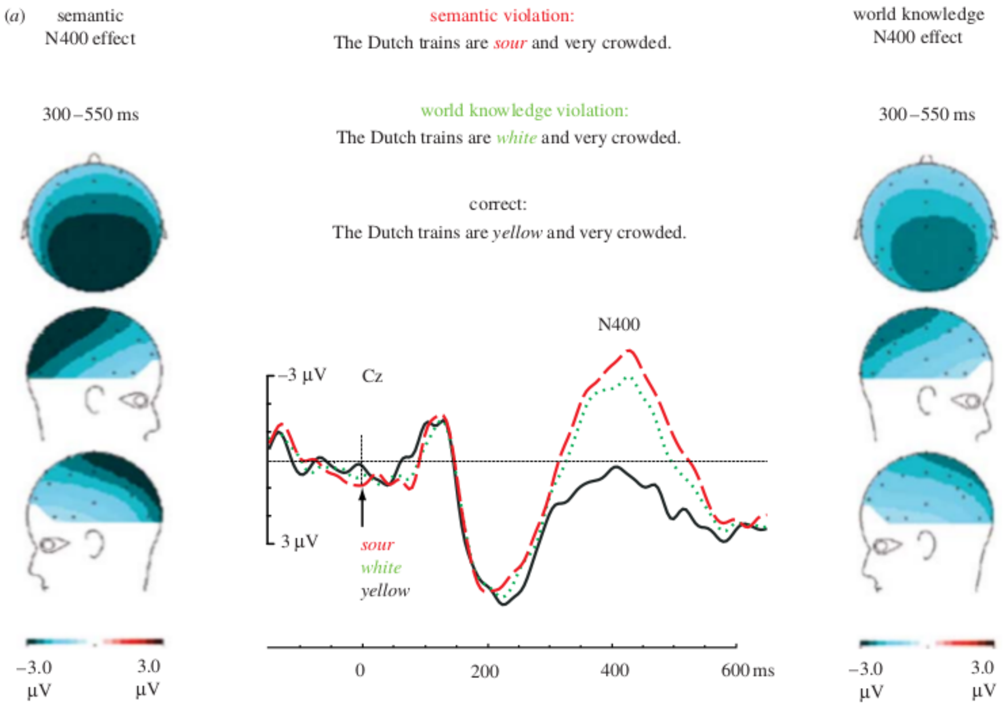

Figure 4.1: Image from Hagoort and van Berkum (2007), showing how the violations of semantic constraints (*train - sour*) and the violations of world knowledge constraints (*Dutch train - white*) behave similarly at a cognitive level.

From the processing point of view, it seems that novel and unexpected combinations require a bigger cognitive effort than typical associations. Nonetheless, events and situations, expressed through sentences, are by definition inherently complex and structured semantic objects, which can be decomposed as stated by the standard, Fregean principle of compositionality.
A long standing line of research in psycholinguistic studies, compatible with the results discussed above, suggests that processing is driven by event contingencies and typical concepts combinations. Therefore, there exist a tradeoff between **storage and computation** (Baggio et al., 2012), that leads us to the need for a deeper analysis about the nature and structure of *lexical* information.

In this area, we believe there is enough evidence to reject the hypothesis of a modular lexicon, which paired with processing theories that placed syntax at the core of sentence composition, in favor of the **Generalized Event Knowledge** (GEK) framework and constraint-based models of language, where comprehension exploits all available sources of information simultaneously.

In particular McRae et al. (1998) studied the influence of the *thematic-fit* (close to the notion of *selectional preference*, (Lebani and Lenci, 2018)) constraint on the selection of two equally possible syntactic interpretations of a sentence, such as *The man arrested...* versus *The cop arrested...*, which are both open to two possible continuations, depending on the interpretation of *arrested* as the main verb of an active sentence (*Main Clause* interpretation) or as a past participle (*Reduced Relative* interpretation).
Their findings showed the role played by the initial noun (i.e., *man* or *cop*) in the syntactic interpretation of the sentence, and found a model in which all constraints were used at the same processing stage to be more in line with reading time data of experimental subjects, thus showing that generalized knowledge about events and their typical participants is activated during processing and guides the unfolding representation of the processed sentence. In other words, the fact that *cop* is more likely to be the subject of the *arrest*, while *man* is more likely to be the object, seems to be considered at once during the assignment of the syntactic interpretation to the sentence.

McRae et al. (2005) and McRae and Matsuki (2009) tested event knowledge priming in more detail, with their results (summarized in figure 4.2) supporting the hypothesis of a mental lexicon arranged as a *web of mutual expectations*, which are in turn able to influence comprehension (figure 4.3).

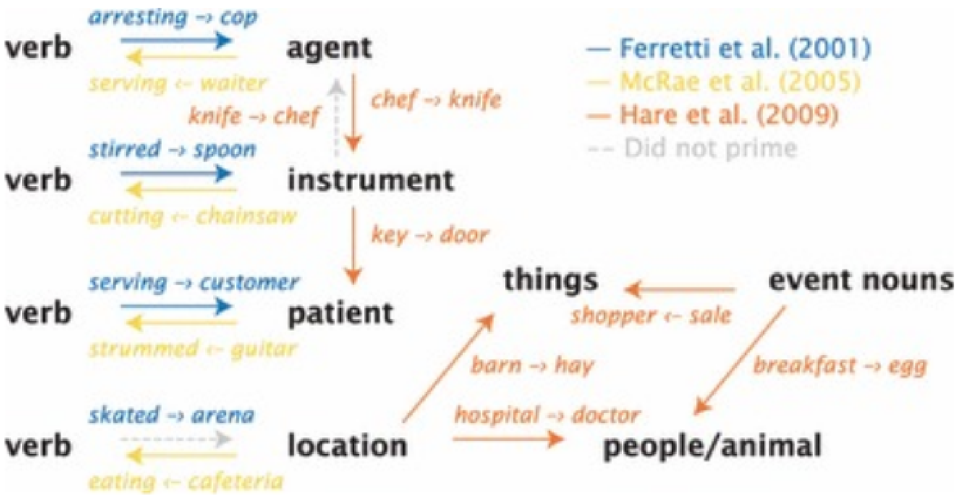

Figure 4.2: Image from McRae and Matsuki (2009) summarizing the experimental studies on event-based verb-argument and argument-argument priming

4.1 The lexicon is the event repository

The majority of available models for meaning composition 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.
As already mentioned, 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, and there are no theoretical reasons why different representations should not be considered.

Actually, assuming the equation *"meaning is vector"* is eventually too limited even at the lexical level. As supported by the mentioned psycholinguistic evidence, lexical items activate a great amount of generalized event knowledge (Elman, 2011; Hagoort and van Berkum, 2007; Hare et al., 2009), and this knowledge is crucially exploited during online language processing, constraining the speakers expectations about upcoming linguistic input (McRae and Matsuki, 2009).

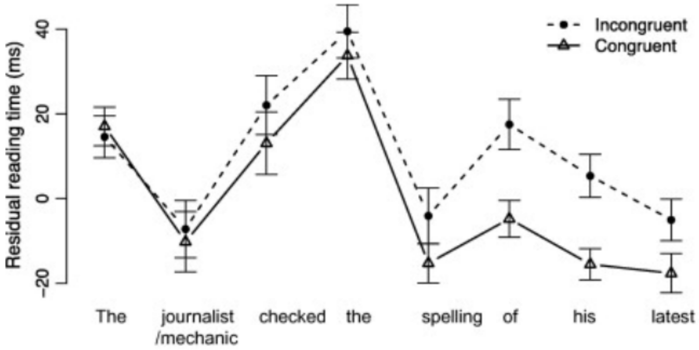

Figure 4.3: Image from Bicknell et al. (2010) showing the difference in the mean residual reading times between the congruent and the incongruent condition

In this framework, sentence comprehension can be phrased as the identification or creation of the event that best explains the linguistic cues used in the input (Kuperberg and Jaeger, 2016): as typical events or conceptual associations are likely to be already stored in the lexicon, they need to be properly *retrieved*, rather than built from scratch. Hence the need to find a new balance between storage and computation in compositional language comprehension.
Along with Rumelhart (1979) and Elman (2009, 2011, 2014), the metaphor of the lexicon as a dictionary is no longer suitable: words are not seen anymore as elements in a data structure that must be retrieved from memory (Elman, 2009),

> *"but rather as stimuli that alter mental states (which arise from processing prior words) in lawful ways. In this view, words are not mental objects that reside in a mental lexicon. They are operators on mental states. From this perspective, words do not have meaning; they are rather cues to meaning".*

In order to populate the memory component and guide unification Elman proposes a Simple Recurrent Network (SRN) that learns the contingent relationships between activities and participants that are involved in events that unfold over time.

In its *strong* version (Lenci, 2008), the Distributional Hypothesis appears as a suitable framework for the computational modeling of generalized event knowledge. Distributional models have in fact been widely used to model psychological phenomena such as similarity judgements and semantic/associative priming and have been developed as cognitively plausible models of language comprehension.

Distributional constraint-based models have been successfully applied to thematic fit modelling (Erk et al., 2010; Baroni and Lenci, 2010).
The *thematic fit* θ of a lexical item a given another lexical item b and a syntactic role s is typically assessed by means of vector cosine between $\vec{a}$ and the prototype vector built out of the k top values $\vec{c_1}...\vec{c_k}$ such that each c_i co-occurs with b in the relation expressed by s. For example, the tematic fit of *student* as the subject of *reading* is given by the cosine similarity between the distributional vector $\overrightarrow{student}$ and the centroid vector built over the most salient subjects of *read*.

A similar approach is adopted in Chersoni et al. (2016, 2017a,b), showing good results in the field of logical metonymy and argument typicality: in the first two works, semantic coherence is assessed as the product of all the partial thematic fit scores for all the event-participant combinations within a sentence, while in Chersoni et al. (2017b) semantic coherence is assessed as the cosine similarity between the arguments of the sentence and the prototype vector of current argument expectations, which is dynamically updated as new information from newly-saturated arguments comes in.

5 | Model

The main purpose of this work is to introduce a compositional distributional model of sentence meaning which integrates vector addition with GEK activated by lexical items.

The model is directly inspired by Chersoni et al. (2016, 2017a,b), whose architecture is based on the *Memory, Unification and Control* (MUC) model (see figure 5.1), introduced in the field of neurosciences by Peter Hagoort (Hagoort, 2013), which includes the following components:

- **Memory** corresponds to linguistic knowledge stored in long-term memory. More than a traditional lexicon, this component resembles a repository of constructions; (Goldberg, 2003), as it is made up by unification-ready structures, which bear different degrees of internal complexity, as in the *Parallel Architechture* framework (Culicover and Jackendoff, 2006);

- **Unification** refers to the constraint-based assembly in working memory of the constructions stored in Memory into larger structures, with contributions from the linguistic and extra-linguistic context;

- **Control** is responsible for relating language to joint action and social interaction.

The model by Chersoni et al. (2016) specializes the content of the memory component, against the hypothesis of a modular lexicon, and populates it with generalized knowledge about events.

In particular, it relies on two major assumptions:

- lexical items are represented with distributional vectords (i.e., embeddings) within a network of 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 model follows this approach, therefore consisting of two main components:

- a **Distributional Event Graph** (DEG) that models a fragment of semantic memory and is activated by lexical units (Section 5.1);
- a **Meaning Composition Function**, that dynamically integrates information activated from DEG to build the semantic representation for the whole sentence (Section 5.2).

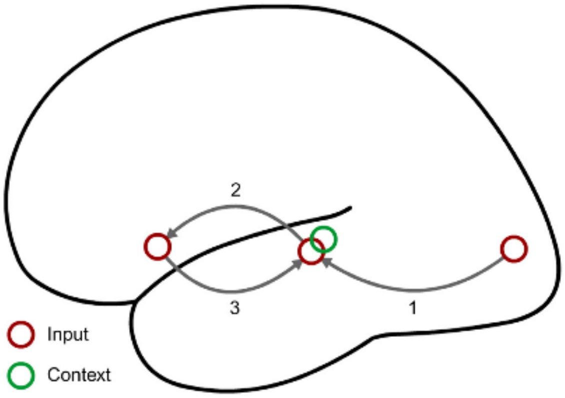

Figure 5.1: Image from Hagoort (2015), showing the processing cycle subserving semantic unification: inputs are conveyed from to the inferior, middle, and superior temporal gyri (1), where lexical information is activated. Signals are hence relayed to the inferior frontal gyrus (2), where neurons respond with a sustained firing pattern. Signals are then fed back into the same regions in temporal cortex from where they were received (3).
A recurrent network is thus set-up, which allows information to be maintained online, a context (green circle) to be formed during subsequent processing cycles, and incoming words to be unified within the context.

We expect the model to be able to reflect the processing differences shown in picture 4.3. The hypothesis is that, when facing sentences like (22) and (23), that share the same main verb *check*, the facilitatory effect observed during processing for the first sentence could be modeled if distributional information about the expected object is triggered by the subject in the first place.

(22) The journalist checked the spelling.

(23) The mechanic checked the spelling.

5.1 Distributional Event Graph

In order to represent GEK cued by lexical items during sentence comprehension, we explored a graph-based representation of distributional information, for both theoretical and methodological reasons: in a graph-like data structure, 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 graph-like data structures are more suitable for situations where relations among items change over time, thus making graph-based structures more suitable for cognitively inspired models.

The data structure would ideally keep track of each event automatically retrieved from corpora, indirectly containing information about schematic or underspecified events, by abstracting over one or more participants from each recorded instance (see figure 5.2).

Events can be therefore cued by all the potential participants, in line with psycholinguistic research, with a strength that depends on the distributional (i.e., statistical) association between the triggered event and the participant.

Because DEG is supposed to contain distributional information, we automatically harvested **events** from corpora, using syntactic relations as an approximation of semantic roles for event participants.

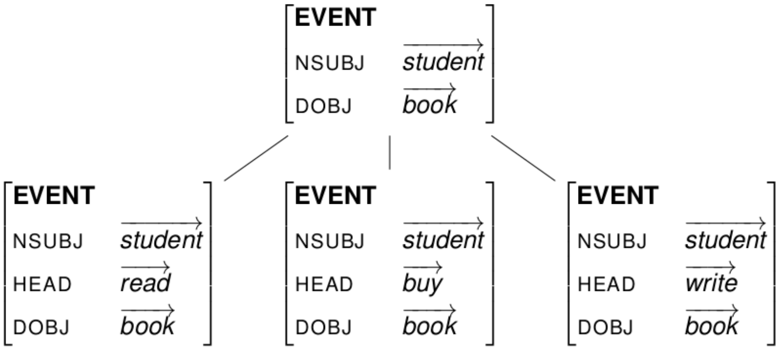
EVENT
NSUBJ $\overrightarrow{student}$
DOBJ $\overrightarrow{book}$
EVENT
NSUBJ $\overrightarrow{student}$
HEAD $\overrightarrow{read}$
DOBJ $\overrightarrow{book}$
EVENT
NSUBJ $\overrightarrow{student}$
HEAD $\overrightarrow{buy}$
DOBJ $\overrightarrow{book}$
EVENT
NSUBJ $\overrightarrow{student}$
HEAD $\overrightarrow{write}$
DOBJ $\overrightarrow{book}$

Figure 5.2: Image courtesy of Alessandro Lenci. The picture shows how an underspecified event such as *student (sbj) - book (obj)*, representing generic events of students performing actions on books, is derived by several fully specified event involving the same participants.

From a dependency parsed sentence we identified an event by selecting a semantic head and grouping all its syntactic dependents together (figure 5.3).

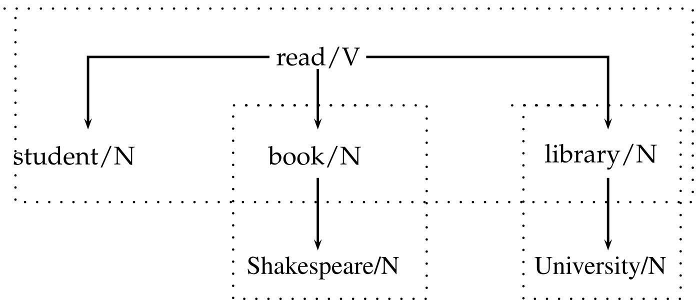

Figure 5.3: Reduced version of the dependency parsing for the sentence *The student is reading the book about Shakespeare in the university library*. Three events are identified.

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 (table 5.1).

Groups	**hyper-edges added to the graph**
read/V, student/N, book/N, library/N	(read/V, student/N, book/N, library/N) (read/V, student/N, book/N) (read/V, student/N, library/N) (read/V, book/N, library/N) (student/N, book/N, library/N) (read/V, student/N) (read/V, book/N) (read/V, library/N) (student/N, book/N) (student/N, library/N) (book/N, library/N)
book/N, Shakespeare/N	(book/N, Shakespeare/N)
library/N, university/N	(library/N, university/N)

Table 5.1: Relations extracted from the dependency parsed sentence shown in Figure 5.3. Syntactic relations are omitted for clarity reasons.

It is worth remarking that we do not stick to the *verbal* notion of *event* as to an *occuring situation*, which is usually described by a verb with its arguments, but we take a broader perspective on the notion of *event* as more general notion of **relation holding among entities that constitue and event, state or situation.** Any *content* word can be the head of an event structure, and this allows us to extend our model to a larger range of compositional phenomena, including also those included within a noun phrase, such as *adjectives* or *relative clauses*).

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.

An *hypergraph* (figure 5.4) is a generalization of a graph in which an edge can join any number of vertices. More formally, an hypergraph H is a pair $H = (V, E)$ where V is a set of vertices and E is a subset of $\mathcal{P}(V) \setminus \emptyset$, namely the set of all possible subsets of V, excluding the empty set $\emptyset$.

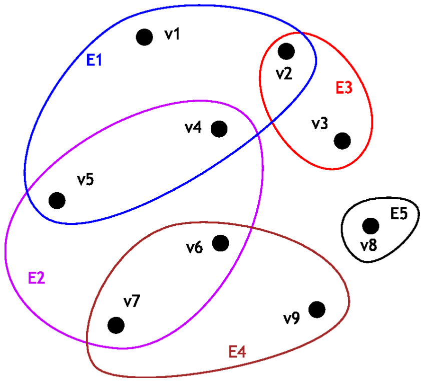

Figure 5.4: A hypergraph $H = (V, E)$ with 9 vertices and 5 hyperedges.

DEG is also a *multigraph* as, given a set of vertices, there could be more than one relation holding among them, which means having more than one event with the same participants (e.g., the two events *dog bites man* and *man bites dog* have the same set of participants and can be both represented in the graph trough different sets of labels).

In DEG, each node is a lexical embedding, and edges link lexical items participating to the same events (i.e., its syntagmatic neighbors), thus representing the event itself. Edges are weighted with respect to the statistical salience of the event given the item. Weights determine the event activation strength by linguistic cues.

As graph nodes are embeddings, given a lexical cue w, DEG can be queried on two different levels:

- retrieving the most similar nodes to w (i.e., its paradigmatic neighbors), using a standard vector similarity measure like the cosine (table 5.2, top row);
- retrieving the closest associates of w (i.e., its syntagmatic neighbors), using the weights on the graph edges (table 5.2, bottom row).

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

Table 5.2: 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.*).

This outset resembles in a sense the twofold representation introduced in Erk and Padó (2008), while generalizes the idea to non explicit relations holding within event participants. Moreover, it allows for the activation of strictured sets of participants rather than binary relations alone, and makes it possible to modulate semantic preferences in context.

5.1.1 Graph Implementation

Event extraction

Although extensible to larger spans of texts, we tailored the construction of the DEG to simple syntactic structures, restricting to the definition of an event as a verbal syntactic head and its main nominal dependents (i.e., *subject*, *direct object*, *prepositional modifiers*).

A number of normalization operations are involved in the process, in particular: passive sentences are brought back to the active voice, enhanced dependencies[1] are also considered for the event construction and only relations within a fixed linear distance from the head are kept into account.

[1] *Enhanced dependencies* are a formalism introduced in the Universal Dependencies schema, in order to make some of the implicit relations between words more explicit, and augment some of the dependency labels to facilitate the disambiguation of types of arguments and modifiers. These are useful for handling phenomena such as *propagation of conjuncts* (in the basic repre-

Relations were automatically extracted from the concatenation of a 2018 dump of Wikipedia, BNC, and ukWaC corpora, parsed with the Stanford CoreNLP Pipeline (Manning et al., 2014), considering sentences longer than 5 tokens and shorter than 100, in order to avoid possible parsing errors.
We restricted to verbs and nouns with a frequency greater than 100, removing those that contained more than half of non alphabetic characters and that started with special symbols (e.g., *!*, *#*, *@*...). Proper nouns were substituted with a generic placeholder, or with the recognised entity (e.g., *person*, *city*, *organization*...) when provided by the parsing.
From this set of relations, for reasons due to the evaluation setting described in Chapter 6, we only considered relations among pairs of entities, labeled with a syntactic pattern that reflected the role of the entity in the group it belongs to. In many of the following examples, the resulting structure will be therefore a directed, labeled multigraph (see figure 5.5).

Each lexical node in DEG was associated with its embedding. For reasons that will be made explicit in the following chapter, we used the same training parameters as in Rimell et al. (2016), since we wanted our model to be directly comparable with the previous results on the dataset. We built a lemmatized 100-dim vector space model, with *skip-gram with negative sampling* (SGNS, Mikolov et al. (2013a)), setting minimum item frequency at 100 and the context window at 10.

sentation, the governor and dependents of a conjoined phrase are all attached to the first conjunct, often leading to very long dependency paths between content words. The enhanced representation contains dependencies between the other conjuncts and the governor and dependents of the phrase), *controlled/raised subjects* (the basic trees lack a subject dependency between a controlled verb and its controller or between an embedded verb and its raised subject, in the enhanced graph, there is an additional dependency between the embedded verb and the subject of the matrix clause) and *relative clauses* (relative pronouns are attached to the main predicate of the relative clause, typically with a nsubj or obj relation, while in the corresponding enhanced graphs, the relative pronoun is attached to its antecedent with the special *ref* relation and the antecedent is attached as an argument to the main predicate of the relative clause).

While Rimell et al. (2016) built their 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.[2].

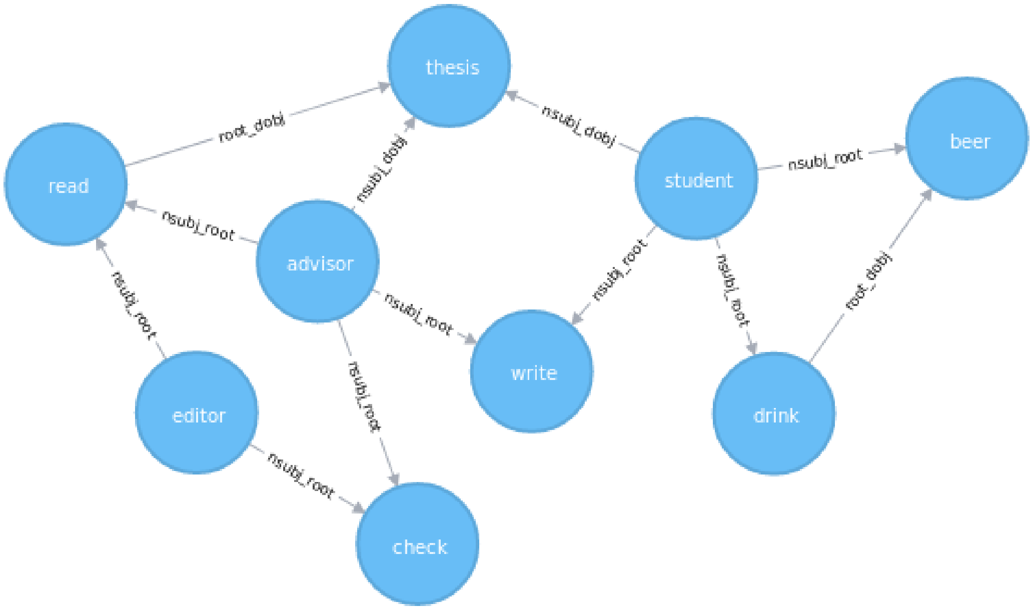

Figure 5.5: The picture shows an example of DEG created from pairs of items co-occurring in the same event. Lexical items (nodes) are linked though syntactic patterns that express their semantic role in the original, complete event.

Weighing Scheme

Each *event - syntactic labels* pairs were then weighted with a smoothed version of Local Mutual Information (LMI) in order to allow for the activation of the event with respect to the salience that the event has for each involved participant.
Each event is here represented as a list of lexemes $e = w_1, ...w_n$, each associated with a label in the list of syntactic labels $p = r_1, ...r_n$.

[2] Results obtained by our embedding on standard datasets are collected in section 6.3.2

We are interested in computing

$$LMI_{\alpha}(e,p) = f(e,p) log \frac{\hat{P}(e,p)}{\hat{P}_{\alpha}(e)\hat{P}(p)} \tag{5.1}$$

in the case of pairs, this formula rephrases as:

$$LMI_{\alpha}(w_1,w_2,r_1,r_2) = f(w_1,w_2,r_1,r_2) log \frac{\hat{P}(w_1,w_2,r_1,r_2)}{\hat{P}(w_1)\hat{P}_{\alpha}(w2)\hat{P}(r_1,r_2)} \tag{5.2}$$

where, for each item x:

$$\hat{P}_{\alpha}(x) = \frac{f(x)^{\alpha}}{\sum_x f(x)^{\alpha}} \tag{5.3}$$

Through some easy calculations, formula 5.2 reduces to the following:

$$\begin{aligned} &LMI_{\alpha}(w_1,w_2,r_1,r_2) = \\ &= f(w_1,w_2,r_1,r_2) log(\frac{f(w_1,w_2,r_1,r_2)}{f(w_1)f(w2)^{\alpha}f(r_1,r_2)} \times C \times C_{\alpha}) \end{aligned} \tag{5.4}$$

where $C = \sum_x f(x)$ is the total number of pairs and $C_{\alpha} = \sum_x f(x)^{\alpha}$ is the sum of element frequencies in the smoothed version.

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

For each lexeme and for each event, weights were normalized using *z-scores* in order to alleviate discrepancies due to huge frequency differences.

Storage and Query

The focus of the representation is a fine-grained representation of the relations holding among entities.
The most traditional way to represent relational information is by means of a *Relational Databases*, namely sets of tables representing highly structured data. In the relational framework, relations are typically inferred through the interaction of particular fields such as foreign keys.

This approach has many drawbacks:

- the organization of a relational database is rigid and must be defined in advance, making it difficult to introduce new relations overtime;
- JOINS are computed at query time by matching primary and foreign keys of all rows in the connected tables, producing compute-heavy and memory-intensive operations;
- many-to-many relations are modelled through the introduction of *join tables*, making in a sense entities more important than relations in the representation.

Graph databases (Robinson et al., 2013) try to overcome these issues by providing a model that more closely resembles the real data organization.
Each node in the graph database model directly (i.e., *physically*) contains a list of relationship records that represent the relationships to other nodes, thus allowing direct access to the connected nodes in case of traditional JOIN operations, allowing a huge advantage in performance, as well as a more intuitive data modeling phase.

For these reasons, we employed a graph database management system, *Neo4j*[3], for the actual implementation of DEG.
Neo4j does not support hyperedges, therefore these must be modeled in a standard property graph introducing extra entities (nodes) that represent the subgraph made of the items that partecipate in the hyperedge. In our model, these extra nodes actually end up explicitly representing **event structures** (see figure 5.6).

[3] `https://neo4j.com/`

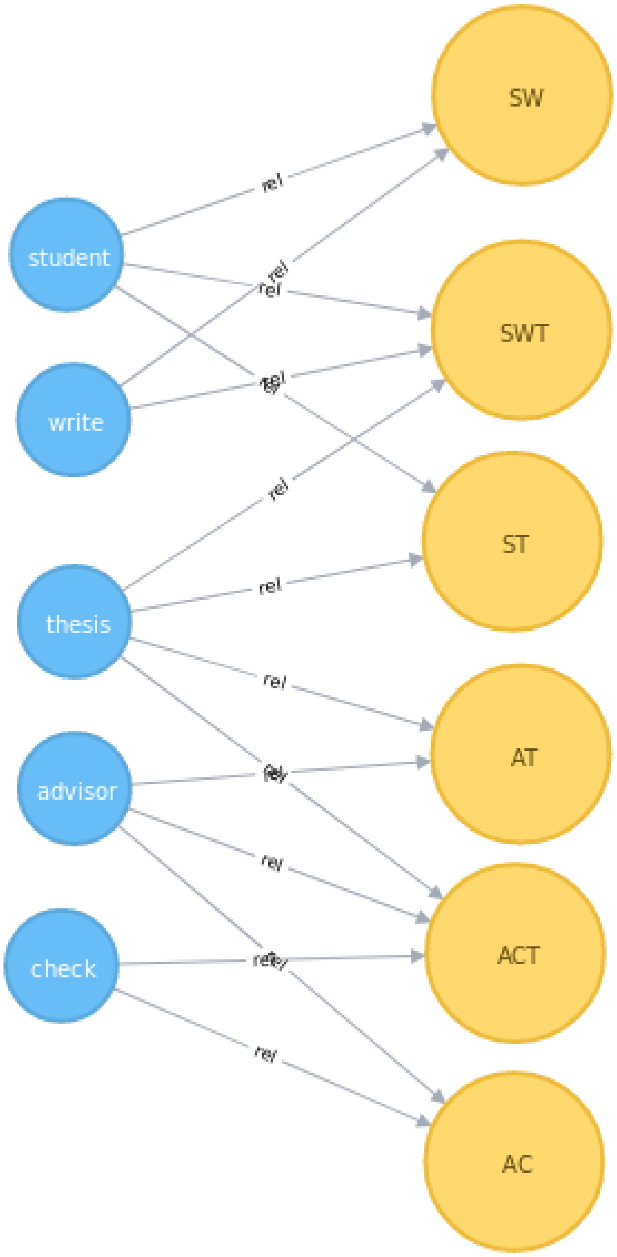

Figure 5.6: Nodes of the left side of the graph are lexical nodes, wich are connected to *event* nodes (on the right side). Event nodes are labeled with the first word of each lexical item involved (e.g., SW stands for *student writes*, SWT stands for *student writes thesis*...).

5.2 Meaning Composition Function

Me model sentence comprehension as the creation of a semantic representation SR, which summarizes two different yet interacting levels of information, that are equally relevant in the overall representation of sentence meaning:

- one component is constituted by the *linear* semantic representation of the sentence, which is a context-independent tier of sentence meaning that accumulates the lexical content of the sentence: we will refer to this component as **lexical meaning component** (LM);

- a second component aims at representing the most probable event, in terms of its participants, which can be reconstructed from DEG from the lexical items of the sentence. It corresponds to the GEK activated by the single lexemes (or by other contextual elements) and integrated into an overall semantically coherent structure representing the event expressed by the sentence and its associated information. We call this structure **Active Context** (henceforth, AC): it is incrementally updated during processing, when a new input is integrated into existing information. AC represents the structured set of information that is available to the agent when processing a sentence.

5.2.1 Lexical Meaning

The LM component is a function of the out-of-context representations (i.e., typically general purpose distributional vectors) attached to the lexemes of the sentence. Therefore, for each lexeme w, its vector $\vec{w}$ is retrived from DEG.

These discrete representations are then composed into a single one, as extensively shown in literature, through algebraic operations such as addition or element-wise multiplication.

$$\begin{tikzcd}
\text{student} \arrow[d] & & \text{read} \arrow[d] & & \text{book} \arrow[d] & \\
\overrightarrow{student} & \oplus & \overrightarrow{read} & \oplus & \overrightarrow{book} \arrow[r] & \overrightarrow{LM}
\end{tikzcd}$$

Figure 5.7: The figure shows how vectors, triggered by lexical items in the sentence, are composed in order to build up the LM component of the semantic representation.

5.2.2 Active Context

The AC component is the *event knowledge component* (EK): each lexical item in the sentence activates a portion of GEK, that is integrated into the already active context through a process of mutual re-weighting that aims at maximizing the overall semantic coherence.

The behavior and the internal architecture of the AC data structure can be specialized through a number of parameters, that will be described in the following sections.
In general, AC represents at each processing step a set of expectations about upcoming linguistic events, and to this end three basic operations are required by the framework:

- Start a new processing phase (`inizialize` operation);
- Process a new piece of input (`retrieve` operation);
- Link new information to existing data (`merge` operation).

Initialization

At the outset of each processing phase, for example at the beginning of a new sentence, a new AC is initialized, with respect to a certain amount of knowledge, as described below.

While DEG contains linguistic knowledge, broadly speaking semantic knowledge about the world, comprehension can be certainly influenced by other, non-linguistic or context-specific factors as well: this might include personal biases, presuppositions, interaction of domain knowledge, but can also be used to keep track of discourse-related expectations, for instance. Since AC aims at modelling linguistic processing, this pre-existing information can be formalized in terms of the active context itself (i.e., expectations about linguistic events), thus allowing us to dynamically weigh new data.

Because this kind of non-linguistic interaction does not strictly pertain to the analysis of the compositional phenomena that we focus on in this thesis and would require a much wider work on the topic, we will not explore the influence of any non-linguistic input.

Retrieval from DEG

The `retrieve` operation represents the interface with the *memory* component (i.e., DEG), and it is concerned with the creation of new meaning blocks representing both lexical meaning and triggered event knowledge.

At each step of the processing, a new pair *(w, synrel)* is encountered, where *w* is a token and *synrel* is the syntactic label attached to it in the dependency parsing, which is taken as an approximation of its semantic role.

The **event knowledge** block (figure 5.8) represents the set of expectations that the lexeme generates about the sentence.

At first, *w* is used to perform a query on the DEG, activating fellow participants to the most likely events in which the *lexeme* participates. The paradigmatic neighbours of *w* (henceforth, *p-neighbours*) can as well be activated when *w* is encountered. Therefore, the distributional neighbourhood (either considering the k top neighbours or setting an activation threshold) of the lexeme could be also added in this phase and participate in the query for event knowledge.

Syntactic labels provide the mapping between expectations and linguistic realization of participants, and are needed in order to perform the query to DEG.

In principle, expectations work also on syntactic structure (e.g., when we encounter a more agentive subject such as *killer* we are more likely to expect a patient in a direct object role, while when we encounter nouns like *reharsal* or *play* we are more likely to expect locations or timings, etc.) and need not to be single labels but can rather be set of roles defining the most likely event structure. Although this represents an interesting area for further research, we did not explicitly address it because of the exploratory nature of our experiments (described in the following chapter). Instead, in order to query the graph, both the syntactic role associated with the current lexeme and the triggered (i.e., expected) ones are used as input. Therefore, in the present case, event knowledge consists of something like ”the typical *direct object*s of *student* as a subject” or ”the typical *locations* of *concert* as a direct object”.

The set of lexemes retrieved from the graph for each target syntactic relation therefore constitutes the weighted set of syntagmatic neighbours (referred to as *s-neighbours*) of the current lexeme.

To sum up, for each processed word pair in the sentence, an hypothesis about the content of the whole sentence is generated, composed of expectations about each semantic role fulfilled in the event, in particular:

- expectations about the role filled by the lexeme itself are represented by its vector (and possibly by its *p-neighbours*);
- expectations about sentence structure and other participants are collected in weighted list of vectors of its *s-neighbours*.

Linking new information

The `merge` operation is concerned with the integration of newly retrieved data into AC. As described in the previous section, when new pairs (w_i, r_i) are encountered, they are incrementally added to AC. Each `add` operation has two main effects (see figure 5.9):

- the event knowledge triggered by the lexeme is weighted according to EK already available in AC;
- the newly retrieved information can be used to reweigh what is already available in AC. This process integrates new and old information by increasing the relevance of the context information that is more semantically coherent with the new items.

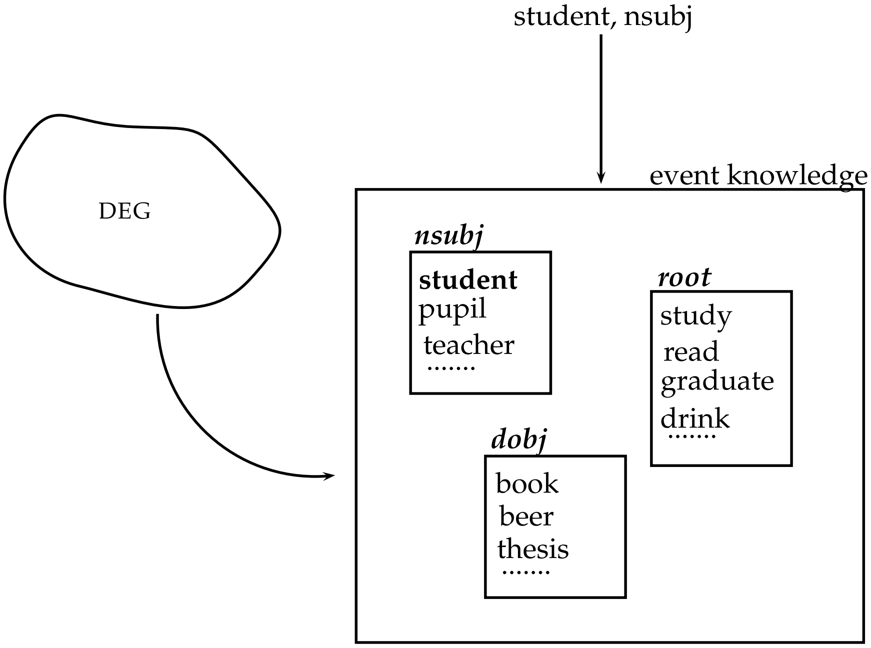

Figure 5.8: The image shows the internal architecture of an EK block, which is created thanks to the `add` operation. The interface with DEG is shown on the left side of the picture, each internal list of *neighbors* are labeled with their expected syntactic label in the sentence. The *nsubj* list is composed by **student**, which is the input lexeme, and the list of its *p-neihgbors*, while the other block are composed of lists of *s-neighbors* of student. All the items are intended as vectors, arrows are not shown for space reasons.

Both *s-neighbours* and *p-neigbours* typically come with an activation weight with respect to the item that activated them (e.g. cosine similarity, mutual information...). We modeled the interaction with pre-existing information in AC as a **bi-directional weighting process**, that allows more salient participants to float to the most prominent positions in AC, while less fitting ones are gradually removed. In practice, each time a weighted list needs to be compared with AC, all the relevant triggered elements in AC are aggregated in a compact representation (i.e., a weighted centroid of the head of the list) and the target list is re-ranked according to the similarity (e.g., cosine similarity between vectors) of each element of the list with the created centroid. The same happens for the lists contained in AC, which are re-weighted according to the centroids of newly retrieved *ek* (the effect is shown in figure 5.10).

It is reasonable to introduce a number of constrains on how this interaction works, as it might not have the same strenght throughout the whole composition process. While this is partially modeled by association scores in the EK component, it can also be explicitly set, thus defining, in a sense, the architecture of the composition process. This is meant to capture two different kinds of phenomena:

- interaction can be propagated up to a fixed distance (in terms of tokens) from the processed one, or with a strength depending on linear distance;
- some semantic roles can show stronger interactions than others (e.g., subjects can pose more constraints on objects than locations).

5.3 Semantic representation and similarity

In order to perform semantic tasks on the basis of the semantic representation $\text{SR} = (\text{LM}, \text{AC})$, a more compact representation of AC is needed.
Different tasks may require different scoring functions, we provide here an outline of our implementation of a similarity measure between semantic representations.

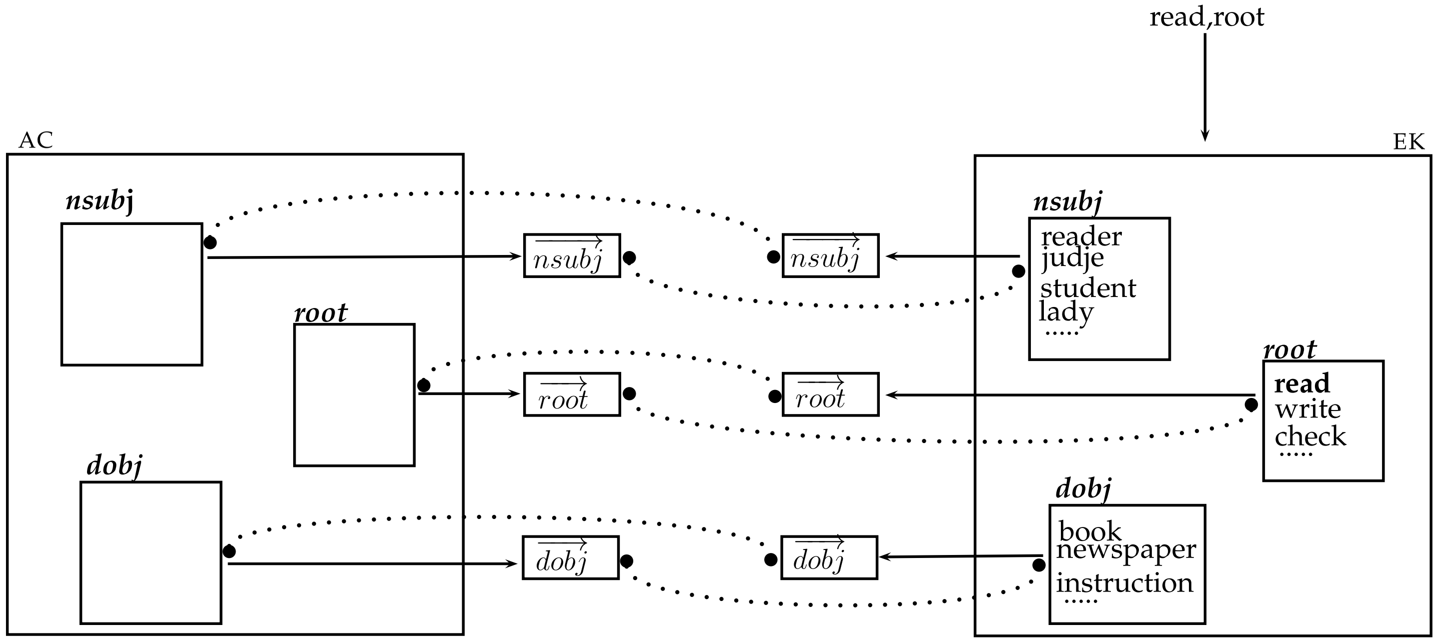

Figure 5.9: The picture shows how the weighing process works. For each syntactic label in AC and in the newly retrieved EK block, the weighted centroid of its top components is created (central vectors in the picture). Information in AC is then scored against newly retrieved participants, and vice versa, in a process of mutual adaptation.

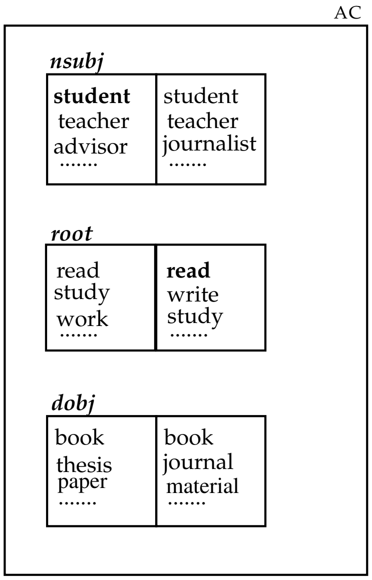

Figure 5.10: The picture shows the new AC, which has integrated information from the new EK block shown in figure 5.9. Lists are reweighted in order to maximize semantic coherence.

While vectors in LM are easily summed up, event participants are represented, for each encountered lexeme, through weighted lists of vectors, one for each role (i.e., syntactic label) in the sentence. Although different aggregating functions are possible, in our implementation we hypothesize that this happens in two steps: at first a weighted centroid is created from the top k vectors triggered for each label, then these centroids are summed up in order to get a single vector (see figure 5.11).

In practice, the semantic representation obtained from the process is composed of two vectors, that represent two different aspects of sentence meaning and that are meant to be compared independently. The two computed scores can, if needed, be later composed.

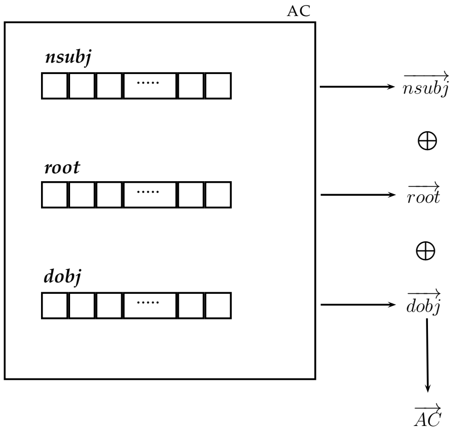

Figure 5.11: The figure shows how vectors derived from AC are composed in order to build up the AC component of the semantic representation.

5.4 Traditional frameworks and AC

Our framework is not intended to be in contrast with all the classic compositional operations, that typically take into account out-of-context distributional objects for each lexeme in the sentence and compose them through algebraic operations. All these models are concerned with making the LM component of our SR more representative of sentence meaning. Our proposal is to enrich such models with event knowledge.

The addition model, for example, is easily implemented in our framework as follows:

$$\text{SR} = (\sum_{w \in S} \vec{w}, \emptyset) \tag{5.5}$$

where S is the set of words in the sentence, and the EK component is empty as no event knowledge is used in the model.

6 Evaluation

6.1 RELPRON

6.1.1 Dataset description

RELPRON (Rimell et al., 2016) consists of 1,087 pairs formed by a *target* noun labeled with a syntactic role (either *subject* or *direct object*) and a *property* expressed as *[head noun] that [verb] [argument]* when the target has the role of a subject, or *[head noun] that [argument] [verb]* when the target is the object of the relative clause. For instance, in table 6.1 the full list of properties for the target noun *treaty* are reported:

OBJ	treaty/N: document/N that country/N sign/V
OBJ	treaty/N: document/N that delegation/N negotiate/V
OBJ	treaty/N: document/N that ruler/N conclude/V
OBJ	treaty/N: document/N that state/N ratify/V
OBJ	treaty/N: document/N that government/N violate/V
SBJ	treaty/N: document/N that end/V war/N
SBJ	treaty/N: document/N that establish/V union/N
SBJ	treaty/N: document/N that require/V ratification/N
SBJ	treaty/N: document/N that grant/V independence/N
SBJ	treaty/N: document/N that cede/V land/N

Table 6.1: Full list of RELPRON properties for the term *treaty*. For the same head noun (*document*), RELPRON contains the following target terms: *account, assignment, ballot, bond, form, inventory, lease, license, specification, treaty*.

The dataset was built by manually selecting candidate terms and head nouns from WordNet[1] (choosing sufficiently non ambiguous nouns that occurred at least 5,000 times in the source corpus), and automatically extracting from corpora (a 2010 dump of Wikipedia and the BNC corpus) triplets in which the terms occurred either in subject or object position, that were later manually filtered in order to keep only good identifying properties (i.e., a property able to distinguish a term from other hyponyms of its head noun).

Overall, the data set includes 565 subject and 522 object properties, with 15 head nouns and 138 terms, divided into a development and a test set as shown in table 6.2.

Test Set			**Development Set**		
Head Noun	**Sbj**	**Obj**	**Head Noun**	**Sbj**	**Obj**
phenomenon	38	42	quality	16	65
activity	46	37	organization	54	45
material	30	52	device	40	36
room	38	42	building	37	32
vehicle	38	41	document	21	48
mammal	41	29	person	59	27
woman	20	17	player	29	9
scientist	58	0			
TOT	309	260	*TOT*	256	262

Table 6.2: Total number of subject and object properties by head noun in RELPRON.

6.1.2 Composition techniques description

Rimell et al. (2016) apply a number of composition techniques, briefly described below, to distributional vectors, testing both count vectors and neural embeddings, obtaining the results summarized in tables 6.3 and 6.4.

[1] Miller et al. (1990)

Lexical baselines - the vector representation of the property is set to the verb or the argument vector, involving no composition.

Arithmetic - vector addition and elementwise vector product of the vectors for the lexical items of the property.

Frobenius Algebra - implementation of the categorial framework, described in equations 6.1-6.2 for the subject and object case respectively ($\odot$ stands for elementwise multiplication):

$$\vec{n} \odot (\vec{V}\vec{o}) \tag{6.1}$$

$$\vec{n} \odot (\vec{V}^T\vec{s}) \tag{6.2}$$

RPTensor - two third order tensors ($\overline{\overline{\overline{R}}}^s$ and $\overline{\overline{\overline{R}}}^o$) are built to represent the relative pronoun in the subject and object case respectively, which combine with the head noun vector and the vector resulting from the verb-argument composition

PLF - implementation of the Practical Lexical Function model: subject and verb matrices combine with their arguments by tensor contraction, and the resulting vectors are added (equations 6.3 - 6.4).

$$\overline{R}^s\vec{n} + \overline{V}^o\vec{a} \tag{6.3}$$

$$\overline{R}^s\vec{a} + \overline{V}^o\vec{n} \tag{6.4}$$

SPLF - Simplified Practical Lexical function model, verb and argument are combined by tensor contraction as in PLF, but the resulting verb-argument vector is combined with the head noun by vector addition.

FLPF - Full Practical Lexical Function model: decouples the interaction of the relative pronoun with the head noun from the interaction with the composed verb-argument phrase. Therefore, a third order tensor is not needed anymore and sparsity problems of RPTensor are mitigated.

Categorial Baselines (Varg, Vhn) - Varg is the verb matrix (subject or object, as appropriate) multiplied by the argument, without considering the head noun. Vhn, conversely, is the verb matrix (subject or object, as appropriate) multiplied by the head noun, without considering the argument.

	Method	**Count**	**Count-SVD**	**Skip-Gram**
Lexical	$\overrightarrow{arg}$	0.272	0.386	0.347
	$\overrightarrow{verb}$	0.138	0.179	0.176
Arithmetic	$\overrightarrow{hn} \odot \overrightarrow{arg} \odot \overrightarrow{verb}$	0.364	0.123	0.181
	$\overrightarrow{hn} + \overrightarrow{arg} + \overrightarrow{verb}$	**0.386**	**0.442**	**0.496**
	$\overrightarrow{arg} + \overrightarrow{verb}$	0.331	0.407	0.401
	$\overrightarrow{hn} + \overrightarrow{arg}$	0.339	0.425	0.450
	$\overrightarrow{hn} + \overrightarrow{verb}$	0.231	0.229	0.264
Categorial	Frobenius	0.277	0.023	0.030

Table 6.3: MAP scores of Lexical, Arithmetic, and Frobenius algebra methods on the RELPRON development set using Count, Count-SVD, and Skip-Gram vectors, and relational verb matrices.

Results are expressed in terms of *Mean Average Precision* (henceforth, MAP), as the evaluation was done on a **ranking task**: the ideal system is supposed to rank, for each term, all properties corresponding to that term above all other properties[2]. MAP is defined as:

$$MAP = \frac{1}{N}\sum_{i=1}^{N} AP(t_i) \tag{6.5}$$

where Average Precision is defined as follows for term t_i:

$$AP(t) = \frac{1}{P_t}\sum_{k=1}^{M} Prec(k) \times rel(k) \tag{6.6}$$

[2] The task is analogous to an Information Retrieval task (Schütze et al., 2008), where documents are required to be ranked given a query.

where P_t is the number of correct properties for term 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 an indicator function which is equal to one if the property at rank k is a correct property for t, and zero otherwise.

Method	**Development**	**Test**
SPLF	**0.496**	**0.497**
Addition	**0.496**	**0.472**
PLF	0.444	0.433
FPLF	0.405	0.387
RPTensor	0.380	0.354
Varg	0.448	0.397
Vhn	0.219	0.204

Table 6.4: MAP scores of composition methods on the RELPRON development and test sets using Skip-Gram vectors.

As shown by the results, vector addition is still among the best performing compositional models, which is able to match the way more complex SPLF.

6.1.3 More on RELPRON plausibility

To the best of our knowledge, RELPRON, although hand-created by the authors, has never been validated with respect to speakers' judgements. The properties were in fact chosen as good descriptors for a certain target, but the authors themselves point out that they could as well be suitable as definitions of other targets. See for example sentences (24)-(30), provided for the target nouns *cinema* (sentences (24)-(25)) and *theater* (sentences (26)-(30)):

(24) building that show movie

(25) building that screen film

(26) building that audience fill

(27) building that audience exit

(28) building that show film

(29) building that host premiere

(30) building that sell popcorn

Evaluating the extension of this phenomenon in the dataset could be crucial to the correct understanding of results, therefore we collected similarity judgements on the developement portion of RELPRON. This allows us to carry out a more in-depth analysis of the pitfalls of both the model and the dataset, and moreover introduces a correlation task which was not present in the original work.

In order to collect similarity judgements for both highly typical paraphrases and atypical paraphrases, we randomly sampled, for each RELPRON target with n associated properties, n additional properties from targets that shared its same head noun. The resulting dataset is therefore composed of 1,036 items, 518 from RELPRON developement set and 518 randomly generated.

Participants were asked to provide a score between 1 and 7 (standard Likert scale) according to how typical a paraphrase was judged as a definition for the given target. Instructions provided to workers are reported below:

> Hi!
>
> In this test you will find questions like the following ones:
>
> ”Would you define a **cow** as a **mammal that a farmer milks**?”
>
> ”Would you define an **astronomer** as a **scientist that uses a telescope**?”
>
> For each question you are asked to **provide a score between 1 (under no circumstances would you choose the provided definition) and 7 (the definition is very typical or common for the questioned word)**.

> Try to **use also intermediate scores** (e.g., when a definition does not include the most typical features of the questioned word, but you would still apply it in some circumstances).
>
> Use your **knowledge of everyday situations and activities** to answer the questions!

We explicitly referred to **knowledge of everyday situations and activities**, as we did not want the workers to interpret *definition* as a sort of a *dicionary definition*, which typically lists a series of necessary and sufficient conditions, but more as a paraphrase that would be suitable in everyday usage to refer to the target item or to further clarify its meaning.

We collected on average 5.33 judgements for each item, with the distributions showed in figure 6.1.

As expected, original RELPRON items get on average higher values than randomly generated ones, no difference is shown in variance distributions on items judgements (figure 6.2) and variances result higher for medium scores and lower for higher/lower scores (figure 6.3).

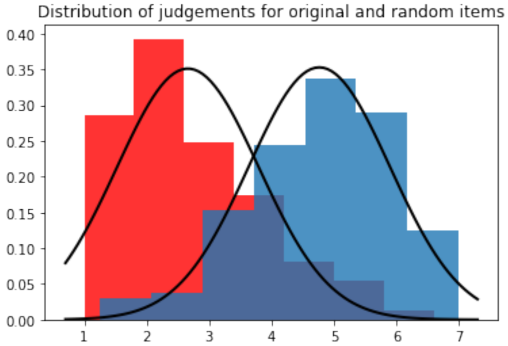

Figure 6.1: The plot shows the distributions of collected judgements for RELPRON items (blue, centred around 4.7) and for the randomly generated ones (red, centred around 2.6).

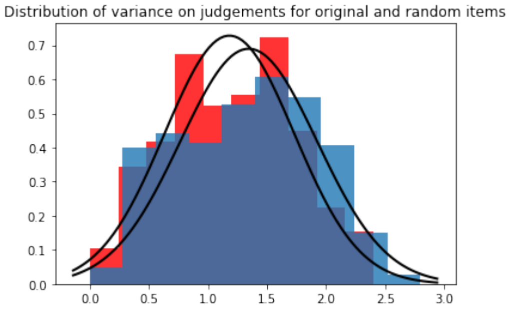

Figure 6.2: The plot shows the distributions of variance on collected judgements for RELPRON items (blue) and for the randomly generated ones (red).

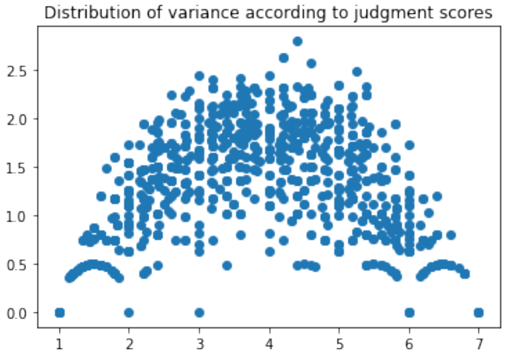

Figure 6.3: The plot shows the distributions of variance dependently on judgements scores. As expected, the pattern seems non-linear.

Moreover, when grouped according to head noun, judgements seem to differ significantly (figure 6.4), showing that some classes are more well defined than others.

The fact that these differences are statistically significant is confirmed by the KRUSKAL-WALLIS test[3], which yielded the results reported in table 6.5.

[3] A non parametric test was run as the *Shapiro* test for normality did not report significant values for all variables.

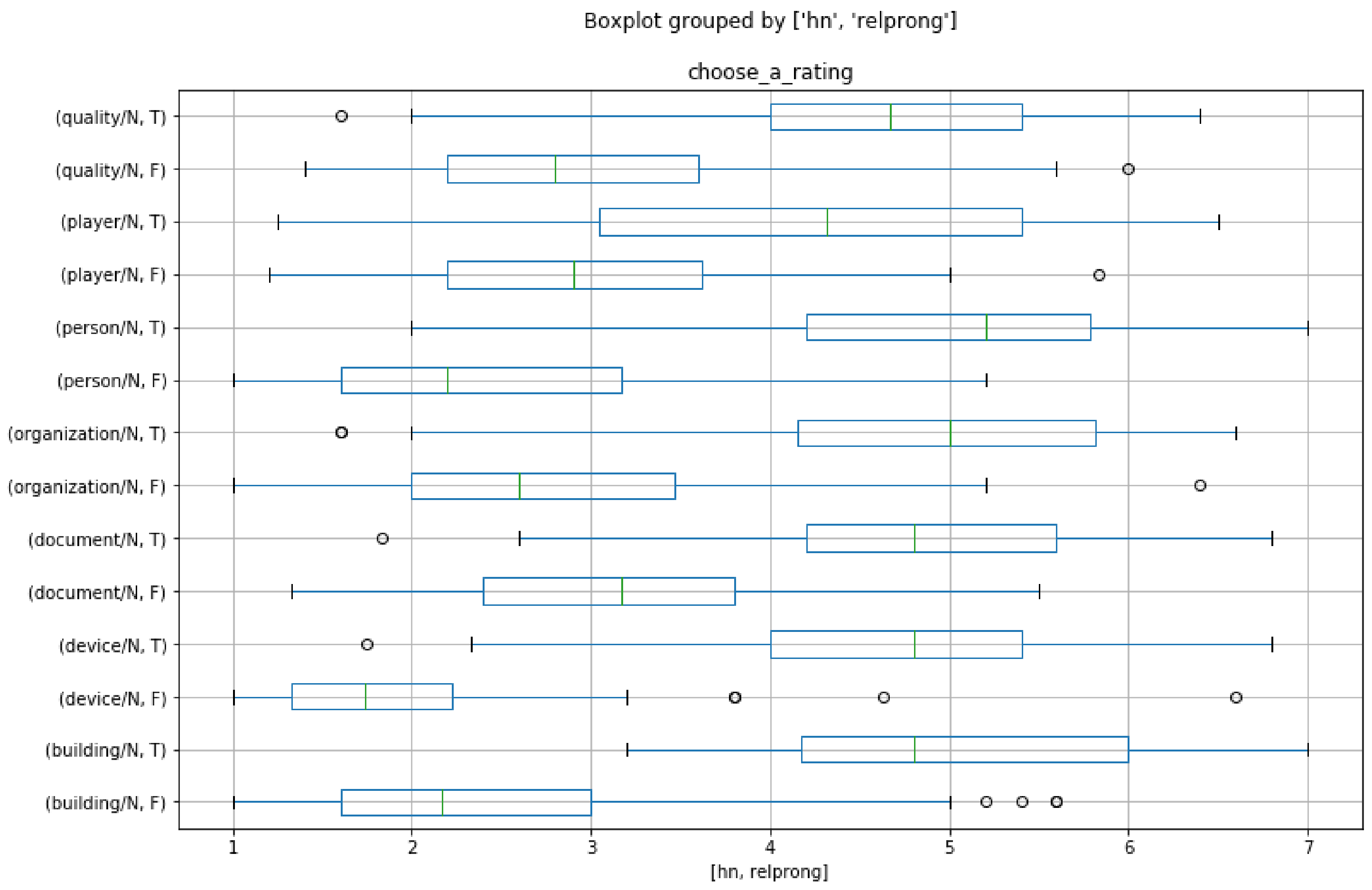

Figure 6.4: The boxplot shows the distribution of judgments according to head noun (*quality, player, person, ...*) and whether the item was a true RELPRON item (therefore expected to have high judgment) or a randomly generated one (expected to have low judgement).

	f-value	p-value
True items	13.12	0.0411
Random (false) items	82.518	1.077604347758283e-15

Table 6.5: Results of KRUSKAL-WALLIS analysis of judgements

The post-hoc analysis (DUNN test, figure 6.5) confirms that *player* is the head noun that behaves most differently from the others, with respect to true items (exhibiting lower judgements on average), while *device* seems the most neatly defined class. Randomly generated items, moreover, behave more differently than true items, thus suggesting that not all RELPRON properties share the same tipicality with respect to the target they are associated to. While all the tested properties seem to be well paired with their target, they could as well be interpreted as good definitions for other targets in the dataset and this needs to be taken into consideration when evaluating any model on the dataset.

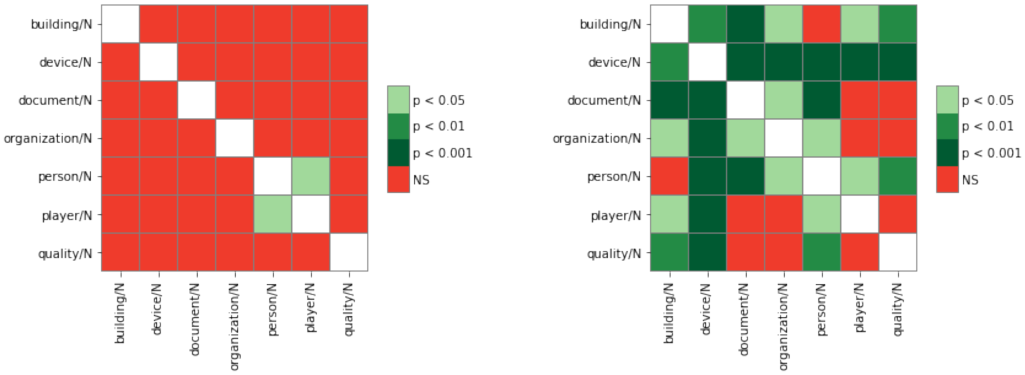

Figure 6.5: The plots show significance levels for pairwise comparisons for each *head noun*, with respect to true items (left plot) and random items (right plot).

6.2 Transitive sentence similarity dataset

6.2.1 Dataset description

The 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 is composed by a triplet *subject verb object*. Here are two pairs with high (31) and low (32) similarity scores respectively:

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

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

6.2.2 Composition techniques description

Milajevs et al. (2014) apply a number of composition techniques, briefly described below, obtaining the results summarized in table 6.6.

In particular, they apply standard vector addition and elementwise product of the vectors for the lexical items of the sentence, and a number of different techniques that mainly differ with respect to how the tensor for the verb is built.

Relational - $\overline{Verb}$ is computed through the formula $\sum_i \overrightarrow{Sbj}_i \odot \overrightarrow{Obj}_i$, where $\overrightarrow{Sbj}_i$ and $\overrightarrow{Obj}_i$ are the subjects and objects of the verb across the corpus. The sentence is then composed as $\overline{Verb} \odot (\overrightarrow{Sbj} \otimes \overrightarrow{Obj})$

Kronecker - $\widetilde{Verb}$ is computed through the formula $\overrightarrow{Verb} \otimes \overrightarrow{Verb}$, where $\overrightarrow{Verb}$ is the distributional vector of the verb. The sentence is then composed as $\widetilde{Verb} \odot (\overrightarrow{Sbj} \otimes \overrightarrow{Obj})$

Frobenius - The expansion of relational verb matrices is obtained by either copying the dimension of the subject into the space provided by the third tensor (*Copy-Sbj*, $\overrightarrow{Sbj} \odot (\overrightarrow{Verb} \times \overrightarrow{Obj})$), or copying the dimension of the object in that space (*Copy-Obj*, $\overrightarrow{Obj} \odot (\overrightarrow{Verb}^T \times \overrightarrow{Sbj})$); furthermore, we can take addition, multiplication, or outer product of these, which are referred to by *Frobenius-Add*, *Frobenius-Mult*, and *Frobenius-Outer*.

Results are expressed in terms of Spearman's rank correlation coefficient (ρ), which assesses how well the relationship between two variables can be described using a monotonic function[4].

Method	**Spearman's ρ**
Verb only	0.561
Addition	**0.689**
Multiplication	0.341
Kronecker	0.561
Relational	0.618
Copy-sbj	0.405
Copy-obj	0.655
Frobenius-Add	0.585
Frobenius-Mult	0.387
Frobenius-Outer	0.622

Table 6.6: Results obtained by Milajevs et al. (2014) on the transitive sentence similarity datasets. The reported results are obtained employing neural embeddings. The first line of the table refers to a baseline obtained computing the cosine similarity between the verbs of the sentences. Milajevs et al. tested the same composition techniques also on count based vectors, and it must be reported that the highest score with vector addition is obtained with one of those vector spaces (0.73).

As for the RELPRON dataset, vector addition is still the best performing model, and some models are not even able to outperform the lexical baseline involving only the verb.

[4] A *monotonic* function between ordered sets is a function that preserves or reverses the given order.

6.3 Models description

6.3.1 Dataset representation

Both RELPRON and the *trantive sentences dataset* are composed of triplets.

Each property in RELPRON was represented 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 verb and the argument of the relative clause, ordered depending on the semantic relation of the target. Each element of the triplet is associated with its syntactic role in the property sentence: in particular, the head noun shares the syntactic role of the target (either *subject* or *direct object*).

Each sentence of the transitive sentences dataset was represented as the triplet $((w_1, nsbj), (w_2, root), (w_3, dobj))$.

6.3.2 Semantic representation

Because the sum model has proven to be the best performing one for both datasets, we aimed at testing the impact of generalized event knowledge within each of the lexical baselines presented in Rimell et al. (2016) and Milajevs et al. (2014).

Given a sentence or a RELPRON item, each model builds therefore a semantic representation $\text{SR} = (\text{LM}, \text{AC})$, where:

- the AC component is empty when no event knowledge is considered, therefore implementing the standard baselines and addition models;
- the LM component is empty for the models employing only event knowledge triggered by the sentence;
- different subsets of the lexical items of the sentence can be selected to take part in the semantic representation, thus generating all the different baselines and their enriched versions.

Since RELPRON properties and sentences from the transitive sentence similarity dataset are both triplets, for both datasets 7 groups can be considered, one for each non empty subset of the elements of the triplet. As far as RELPRON is concerned, however, we do not consider the model involving the head noun alone as the same head noun is shared by many targets in the dataset, thus the ranking task performed on the head noun alone would not be directly comparable to the other results.

Lexical Meaning

The lexical meaning (LM) component of each SR was built by adding the distributional vectors attached to the lexical items in the sentence.

One important parameter of the model is the kind of word embeddings employed to represent words (and the lexical nodes of DEG as well). We reproduced the vector space of Rimell et al. (2016), as described in chapter 5.
We summarize here the results obtained by our vector space on standard datasets (see tables 6.7 and 6.8), as these must be considered in order to properly evaluate the model results.
It may be the case, in fact, that more specialized embeddings would capture different aspects of lexical meaning and therefore sensibly influence the performances of the model.

Tested datasets include:

RG - Rubenstein and Goodenough dataset, Rubenstein and Goodenough (1965)
WS, WS-SIM, WS-REL - WS353 dataset for testing attributional and relatedness similarity, Finkelstein et al. (2001); Agirre et al. (2009)
YP - verb similarity, Yang and Powers (2006)
MTurk - Mturk dataset for attributional similarity, Radinsky et al. (2011)
RW - Rare Words dataset, Luong et al. (2013)
MEN - Bruni et al. (2014)
SimLex - Hill et al. (2015)
SimVerb - verb similarity, Gerz et al. (2016)

	RG	WS	WS-SIM	WS-REL	YP	MTURK	RW	MEN	SimLex	SimVerb
GloVe 100dim	0,676	0,601	0,602	0,491	0,454	0,581	0,366	0,696	0,258	0,179
SGNS 100dim 10w	0,802	0,781	0,782	0,647	0,526	0,639	0,446	0,766	0,341	0,260

Table 6.7: Results of the implemented vector space model (SGNS 100dim 10w) on standard datasets. Performances are comparable to standard results achieved in literature by general purpose models. The table also provides results obtained by 100 dimensions *GloVe* vectors, as reference.

	RG	WS	WS-SIM	WS-REL	YP	MTURK	RW	MEN	SimLex	SimVerb
number of items	65	202	203	252	130	771	2034	3000	999	3500
GloVe 100dim (% of coverage)	100	96,53	96,55	94,44	100	100	87,61	76,17	86,29	99,94
SGNS 100dim 10w (% of coverage)	100	100	100	99,60	100	98,57	58,65	76,07	86,29	100

Table 6.8: The table shows, for each dataset, how many items of each dataset are present in the distributional space.

Active Context

For each item, an AC is initiated empty. Triplets are then processed in linear order, adding elements one by one to the AC. Each time an element is added, it activates some event knowledge with respect to the semantic roles relevant for the dataset. In particular, as far as RELPRON is concerned, event knowledge is triggered only for the syntactic relation of the target, whereas for the transitive sentences dataset all syntactic relations (i.e., *subject*, *root*, *direct object*) are considered.
For each relation r:

- if the lexeme bears the relation r itself, it is added as event knowledge (e.g., in the case of *student-nsubj*, the vector of *student* is added as event knowledge for the *nsubj* relation);

- otherwise, the top 50 *s-neighbours*, along with their LMI as weight, are extracted from the graph and added to AC. This list is reweighted with respect to previous information as follows: for each pre-existing list in AC, labeled with the same relation r, the top 20 elements of its *s-neighbours* list are aggregated in a weighted centroid. All the centroids are then summed up, and the newly retrieved *s-neighbours* list is reweighted with respect to cosine similarity with the vector representing AC.

.

Example

We provide an example of the reweighing process the property *document that store maintains*, whose target is *inventory*:

- 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 against wich GEK is re-weighted. At this point, the cosine similarity between *inventory* and the AC (which contains only the head noun vector) is 0.530. The most similar items to the AC would be general nouns such as *documentation*, *archive*, *dossier...*;

- the noun *store* is encountered, and expectations for objects of *store* as a subject are queried from DEG. These include *product*, *range*, *item*, *technology*, etc. in the top positions, and if the centroid were built from the top of this list, the cosine similarity with the target would be around 0.62;

- *s-neighbours* of *store* are re-weighted according to the fact that AC contains some information about the target already, namely the fact that it is a document. The re-weghting process has the effect of placing on top of the list elements that are more similar to *document*, so now we find *collection*, *copy*, *book*, *item*, *name*, *trading*, *location*, etc., improving already cosine similarity, that goes up to 0.68;

- similarly, *s-neighbours* of *maintain* are extracted from DEG (these are: *standard*, *relationship*, *position*, *record*, *level*, etc.) and reweighted with respect to the current of AC, that is *document*$+$*objects of store as a subject*, thus getting *database*, *page*, *website*, *site*, *register*, *property*, *list*, etc. as top items, and improving cosine similarity with *inventory*, from 0.55 to 0.61.

6.3.3 Scoring

Given two semantic representations $SR_1 = (\overrightarrow{LM_1}, \overrightarrow{AC_1})$ and $SR_2 = (\overrightarrow{LM_2}, \overrightarrow{AC_2})$ (these are the property and the target in case of RELPRON, and two sentences in case of the transitive sentence similarity dataset), the final score is computed as follows:

$$s = cos(\overrightarrow{LM_1}, \overrightarrow{LM_2}) + cos(\overrightarrow{AC_1}, \overrightarrow{AC_2}) \tag{6.7}$$

where $cos(\vec{x}, \vec{y})$ is standard cosine similarity.

6.4 Results

Tables 6.9-6.10 summarize the results obtained by all models, on both datasets. The results show that event knowledge alone (i.e., information in AC) obtains scores which are not far from the baseline, while combining of the two parts of the semantic representation sistematically improves the basic scores. In particular, the information provided by the event knowledge in AC is able to outperform the simple vector addition.

	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 6.9: The table shows results in terms of MAP for the RELPRON dataset. Except for the case of verb+arg, the models involving event knowledge in AC always improve the baselines.

	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 6.10: 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 ($p < 0.01$).

We also evaluated correlations on judgments provided by human annotators on the RELPRON dataset. The overall results are shown in table 6.11. Here, event knowledge does not seem to improve the baselines and the standard sum model still shows the best results on average. However, the dataset in this case was composed of true RELPRON items, which were supposed to get high human judgments, along with randomly created items, which were supposed to get low scores. Table 6.12 shows correlations coefficients for the two subsets respectively: while on true items the role of lexicalized information is greater, when it comes to random items event knowledge improves the baselines systematically. This suggests that, while pure lexical information is enough for certain situations, event knowledge seems to be able to provide better disambiguation when needed.

	RELPRON		
	LM	AC	LM+AC
verb	**0.37**	0.32	**0.37**
arg	**0.56**	0.48	0,55
hn+verb	0.24	**0.29**	0.28
hn+arg	0.49	0.49	**0.52**
verb+arg	**0.60**	0.40	0.55
hn+verb+arg	**0.54**	0.44	0.51

Table 6.11: The table shows Spearman's ρ coefficients between the series of scores provided by the model and those provided by human annotators.

	RELPRON items			RANDOM items		
	LM	AC	LM+AC	LM	AC	LM+AC
verb	0,06	**0,08**	0,07	0,26	0,23	**0,27**
arg	**0,22**	0,16	0,20	0,27	**0,32**	0,31
hn+verb	0,01	**0,04**	0,02	0,13	**0,21**	0,18
hn+arg	**0,18**	0,15	**0,18**	0,21	**0,28**	0,26
verb+arg	**0,20**	0,06	0,14	0,31	0,30	**0,33**
hn+verb+arg	**0,16**	0,09	0,14	0,25	0,24	**0,26**

Table 6.12: The table shows Spearman's ρ coefficients between the series of scores provided by the model and those provided by human annotators, for true RELPON items and randomly generated items respectively.

7 | Error Analysis

Rimell et al. (2016) provide some in depth analysis of their result on the RELPRON dataset. They point out four main aspects, namely the role of:

- the grammatical function of the term (*subject* or *object*);
- the head noun;
- the intersective semantics of the relative clause construction (how often is the head noun correctly identified and how well are properties ranked, given the head noun);
- lexical overlap and plausible, yet not annotated descriptors in the dataset.

We follow the same analysis scheme, aggregating the second and the third point as they are better explained when seen together.

7.1 Accuracy by grammatical function

Rimell et al. (2016) find their models to be on average balanced with respect to *subject* and *object* predictions. The most unbalanced model among theirs is the FPLF model, which shows much better perfomances on subjects than on objects. They impute this discrepancy to the different amount of training data available for subject and object relative clauses respectively.

Tables 7.1, 7.2, 7.3 show our results for the lexical baselines, the models involving only AC and the complete models respectively.

LM	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 7.1: The table shows MAP results, for each model involving only the LM 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

Table 7.2: The table shows MAP results, for each model involving only the AC component, for subject and object relations separately.

While models involving only the sum of lexical vectors show balanced results, a value stands out in table 7.2. The composition of event knowledge elicited by verb and argument seems much better at predicting the object than the subject. This may suggest that subjects are more likely to trigger, distributionally speaking, their object rather than the other way round, but it would be in contrast with the fact that the argument alone predicts subjects better than objects.

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 7.4).
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*.

LM+AC	verb	arg	hn+verb	hn+arg	verb+arg	hn+verb+arg
subject	0,23	0,44	0,33	0,58	0,46	0,60
object	0,20	0,37	0,32	0,53	0,45	0,60
Δ	0,03	0,07	0,01	0,04	0,01	0,00

Table 7.3: The table shows MAP results, for each model involving both the LM component and the AC, for subject and object relations separately.

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

Table 7.4: 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.

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 both the fact that verb perform distributionally worse than nouns on standard similarity tasks, and the fact that information derived by arguments has also in general better correlations with human judgements, and seems therefore a better source for event knowledge than the one provided by verbs.

7.2 Accuracy by Head Noun and intersective semantics of relative clauses

Rimell et al. (2016) evaluate MAP by averaging APs obtained for each term, still computed on the ranking of all RELPRON properties, divided by head noun, and show stable results across head nouns for all methods, concluding there are no outliers among head nouns in the ability of the methods to compose relative clauses.

Figures 7.1, 7.2 and 7.3, when compared with data reported in table 7.5, show in our opinion a different perspective: the fact that some targets have on average higher or lower similarities with their head noun plays as a strong bias in the performances of the models.

Looking at the ranked plots (on the right of each pair of plots), in particular, it seems pretty clear that tendencies tend to invert when passing from the first three models (the ones that do not involve any knowledge on the head nouns) to the second three (all of which include the head noun).

Head nouns such as *organization* and *document*, which are on average not very similar to their targets, show a sensible drop of performances in the models where *hn* is involved, while *device*, which in particular shows the lowest scores on the verb only model, increases sensibly its performances in the head noun + verb model, presumably as an effect of the fact that device targets show quite high similarity with their head noun.

Rimell et al. look also at the integration of the semantic contribution brought by the head with the contribution of the verb and the argument, and break down the task into two subtasks that demonstrate performances on the two aspects independently:

- considering the top ten ranked properties for each term from the full development set, and calculating the percentage of them which have the correct head noun;
- looking at the MAP scores when the ranking of properties for each term is restricted to properties with the correct head noun.

The first analysis (percentage of correct head nouns ranked among the top 10 properties, table 7.6) only confirms that general similarity between the target and its head noun plays a role in the performances.

As far as the second point is concerned (MAP scores within each head noun group), table 7.7 shows no clear trends. Results are overall not satisfactory: discrepancies between the ability to rank the right head noun and the ability to discriminate between properties bearing the same head noun have two possible explanations. Either the model is not able to distinguish between two targets belonging to the same family, or the properties are not informative enough.

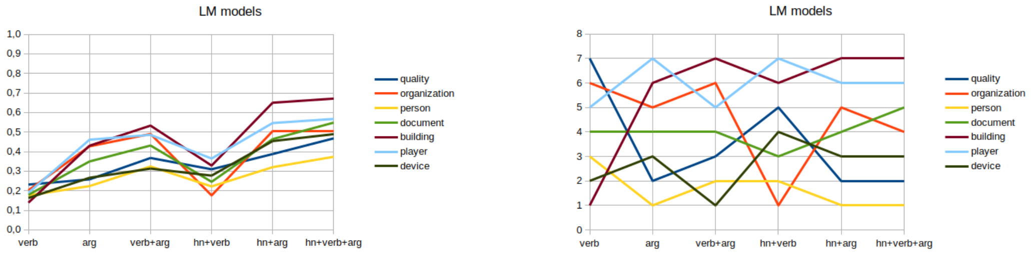

Figure 7.1: The two plots show performances of the LM models, for each class defined by the head nouns. The left plot shows the actual MAP values, while the right plot shows their rank.

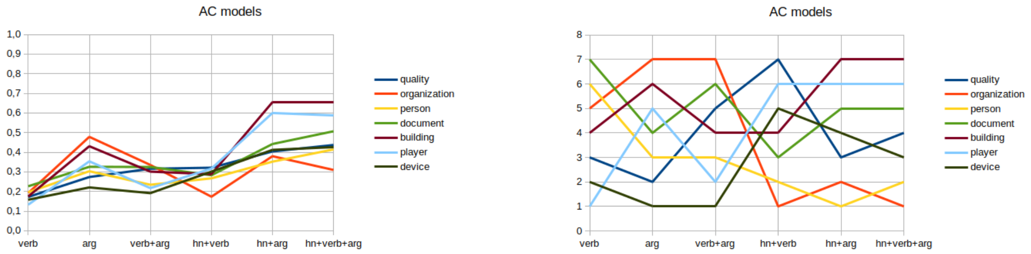

Figure 7.2: The two plots show performances of the AC models, for each class defined by the head nouns. The left plot shows the actual MAP values, while the right plot shows their rank.

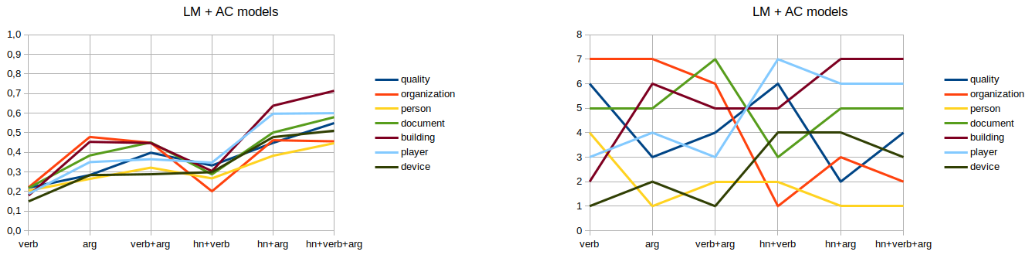

Figure 7.3: The two plots show performances of the LM+AC models, for each class defined by the head nouns. The left plot shows the actual MAP values, while the right plot shows their rank.

	mean cosine similarity	
head noun	**with head noun**	**among targets**
player	0.558	0.545
building	0.528	0.426
device	0.495	0.401
quality	0.463	0.375
document	0.444	0.324
organization	0.378	0.334
person	0.359	0.350

Table 7.5: The table shows, for each head noun, the average cosine similarity between the head noun and each target noun in the class (first column) and the average cosine similarity between target nouns within the same class.

Some insights on this could come by looking at the discrepancies between these results and the scores obtained through human annotation. Looking at boxplot 6.4, we would expect that *device* would get the highest scores as it seems the most easy group to split between true items and random (i.e., false) items. This is not the case probably because the class of *devices* has one clear outlier, which is the target noun *fan* (its AP is only 0.01). It is clearly polysemous, and it is intended in the dataset as *”an electric device with blades that turn quickly, used to move the air around”*, or *”an object made of folded paper or other material that you wave with your hand in order to move the air around”*, but its distributional neighbourhood (*crowd, fanbase, die-hard, booing, favourite, audience, spectator, cheer...*) shows that the meaning of *fan* in the distributional model is caught in the sense of *”someone who admires and supports a person, sport, sports team, etc...”*[1]. By removing *fan* from the picture, scores go up to 0.66 in the LM model and to 0.60 in the LM+AC model.

This kind of polysemy results in fact in no ambiguity for the human speaker, as the senses are very different, and therefore has no effect on the human judgements which interpret *fan* in the right sense, but needs to be explicitly addressed computationally.

[1] Definitions from the Cambridge online dictionary, `https://dictionary.cambridge.org/`

		quality	organization	person	document	building	player	device	overall
LM	verb	0,52	0,53	0,47	0,41	0,46	0,62	0,47	0,49
	arg	0,52	0,71	0,42	0,49	0,58	0,78	0,49	0,55
	verb+arg	0,64	0,72	0,56	0,51	0,67	0,88	0,55	0,63
	hn+verb	0,77	0,57	0,55	0,63	0,94	1	0,84	0,74
	hn+arg	0,77	0,8	0,47	0,73	0,95	1	0,78	0,77
	hn+verb+arg	0,78	0,79	0,61	0,7	0,93	1	0,75	0,78
AC	verb	0,68	0,64	0,55	0,44	0,62	0,58	0,58	0,58
	arg	0,66	0,68	0,53	0,49	0,65	0,86	0,62	0,62
	verb+arg	0,76	0,64	0,61	0,52	0,66	0,8	0,62	0,65
	hn+verb	0,83	0,53	0,61	0,57	0,94	0,98	0,8	0,73
	hn+arg	0,78	0,69	0,64	0,62	0,94	1	0,78	0,76
	hn+verb+arg	0,82	0,65	0,68	0,62	0,97	1	0,88	0,79
LM + AC	verb	0,67	0,56	0,55	0,47	0,64	0,64	0,51	0,57
	arg	0,65	0,72	0,53	0,56	0,65	0,82	0,55	0,63
	verb+arg	0,77	0,67	0,59	0,54	0,68	0,86	0,57	0,65
	hn+verb	0,86	0,59	0,65	0,64	0,96	1	0,86	0,78
	hn+arg	0,81	0,77	0,6	0,68	0,94	1	0,78	0,78
	hn+verb+arg	0,8	0,74	0,65	0,71	0,97	1	0,8	0,80

Table 7.6: The table shows, for each set of models (LM, AC and LM+AC models), and for each head noun, the percentage of properties sharing the correct head noun within the top 10 ranked properties for each target.

7.3 Common errors in descriptions

Two common sources of errors on RELPRON, as pointed out by the authors, are **lexical overlap** between terms and properties, which was intentionally included in the dataset, , and the fact that there exist properties which are **plausible descriptions** for a term, but were not annotated as gold positives in the first place.

The former refers to the fact that there exist in the dataset properties like *person that religion has*, referred to the target *follower*, while there also is the word *religion* as a target in RELPRON, and the latter to properties such as *organization that recruits soldier*, which is associated to *navy* in the dataset, but could as well be a plausible description for the target noun *army*.

		quality	organization	person	document	building	player	device
LM	verb	0,37	0,35	0,33	0,30	0,27	0,32	0,28
	arg	0,45	0,65	0,50	0,51	0,66	0,52	0,45
	verb+arg	0,56	0,68	0,56	0,58	0,70	0,53	0,53
	hn+verb	0,39	0,35	0,34	0,31	0,35	0,37	0,29
	hn+arg	0,50	0,67	0,49	0,52	0,72	0,55	0,54
	hn+verb+arg	0,61	0,67	0,54	0,61	0,75	0,57	0,58
AC	verb	0,25	0,32	0,33	0,34	0,26	0,29	0,23
	arg	0,36	0,61	0,45	0,52	0,57	0,44	0,36
	verb+arg	0,39	0,46	0,40	0,48	0,44	0,32	0,33
	hn+verb	0,37	0,35	0,39	0,40	0,30	0,34	0,33
	hn+arg	0,48	0,59	0,47	0,58	0,67	0,61	0,45
	hn+verb+arg	0,53	0,55	0,52	0,65	0,66	0,59	0,44
LM + AC	verb	0,34	0,37	0,34	0,34	0,27	0,32	0,26
	arg	0,39	0,66	0,50	0,56	0,65	0,42	0,46
	verb+arg	0,51	0,62	0,51	0,61	0,62	0,42	0,46
	hn+verb	0,37	0,39	0,40	0,37	0,32	0,35	0,32
	hn+arg	0,51	0,66	0,53	0,59	0,70	0,60	0,52
	hn+verb+arg	0,65	0,68	0,57	0,68	0,75	0,60	0,55

Table 7.7: The table shows, for each set of models (LM, AC and LM+AC models), and for each head noun, the MAP obtained in the ranking task where only properties with the correct head noun are considered.

We looked at lexical overlap a bit closer, and selected the possible properties that share a large extent of their meaning with the target. In order to do so, for each target we looked at all properties, excluding its own ones, which had at least one word with cosine similarity greater that 0.8 (i.e., a synonym) with the target. The complete list of found properties is given in table 7.8. The table show that the phenomenon, although intentionally introduced by the authors, is not equally spread throughout the dataset, and for this reason its effect is difficult to evaluate. *Organization* has the greatest number of lexically overlapping properties, but they are equally spread between *inter-class* properties (i.e., properties of targets that share the same head noun) and *intra-classes* properties (i.e., properties of classes that bear a different head noun), while *player*, which is also the head noun with the fewest targets and properties (5 and 38 respectively), has 5 properties that show lexical overlap, all of which are referred to targets in the same class.

As shown by the table, despite the fact that complete models stills struggle with the issue, models involving only event knowledge seem to be not affected at all by lexical overlap. And this comes without much loss of accuracy as the top ranked properties for nearly all the considered items are gold ones (exceptions are *team*, which ranks player properties in the first two positions, *army* wich ranks garrison properties in the first two positions - but *army* and *garrison* are more similar that 0.8 - and *balance* which ranks other qualities in the top 4 positions).

As far as the second issue is concerned, Rimell et al. introduce another ranking task, treating properties ad queries and ranking terms by their similarity to a property.
We opted for using human judgements in this phase, and, from the whole set of 1036 items, we created four subsets on the basis of the expected result (RELPRON, thus *true* item or randomly generated, thus *false* item) and the average collected score (above or below 3.5 out of 7). The items distribute themselves in the four resulting groups as shown by table 7.9 and figure 7.4.

The fact that 112 false items received a score above 3.5 suggests that it would be worth investigating more deeply the complete set of possible co-occurrences between targets and properties. Because they were only randomly selected, however, they do not yield a complete picture of the phenomenon, and for this reason we're not speculating on them.
On the other hand, the 76 true items that received low scores are worth investigating (they are fully reported in tables 7.10): the player class is the one that shows the greatest problems, with one item (*bowler*), that gets completely removed from the dataset.

	hn	**target**	**property**	**true target**	LM	AC	LM+AC
F	building	observatory	device that **observatory** have	telescope	1	15	2
F	device	telescope	building that contain **telescope**	observatory	1	29	2
F	device	fan	building that **fan** pack	arena			
T	device	button	device that **button** replace	dial	1	19	2
F	document	lease	building that brewery **lease**	pub			
F	organization	religion	person that **religion** have	follower	1	6	1
F	organization	family	building that **family** rent	house	2	47	3
F	organization	family	person that lose **family**	survivor	1	2	1
F	organization	family	building that shelter **family**	house	3	28	2
T	organization	garrison	organization that seize **garrison**	army	1	4	2
T	organization	garrison	organization that **troops** defeat	army	7	8	6
T	organization	team	organization that **team** join	division	2	38	3
T	organization	army	organization that **army** install	garrison	7	30	12
T	organization	army	organization that **force** besiege	garrison	1	2	1
F	person	traveler	device that **traveler** set	watch	1	31	2
F	person	philosopher	quality that **scholar** dispute	accuracy	7	25	12
T	player	pitcher	player that **pitcher** strike	batter	1	3	1
T	player	pitcher	player that **pitcher** face	batter	2	8	3
T	player	batter	player that **batter** face	pitcher	3	5	13
T	player	batter	player that strike **batter**	pitcher	1	2	1
T	player	batter	player that walk **batter**	pitcher	2	3	2
F	quality	balance	document that have **balance**	account	1	115	10

Table 7.8: The table shows the complete list of the target-property pairs for which *lexical overlap* has been detected (bold terms are the ones whose cosine similarity with the target is greater than 0.8). The fist column indicated whether or not the given target and the true target of the property share the same head noun. The last three columns show at which position the properties are ranked for the target for which some lexical overlap occurs. Two rows of the table are grayed. In the first case lexical overlap is found for the term *fan*, but the two occurrences refer to two different meanings, and *fan* is an outlier of the dataset. In the second case, the overlap occurs with *lease*, but it is intented as a noun when it is considered as a target, and as a verb in the property, for this reason the pair has a different status from the others in the table.

	RELPRON **item**	RANDOM **item**
$<= 3.5$	76	406
> 3.5	442	112

Table 7.9: The table shows how the set of items for which human judgements were collected distributes itself against the two variables *being a true* RELPRON *item* and *having received a score higher than* 3.5 (i.e., having been rated as a true item).

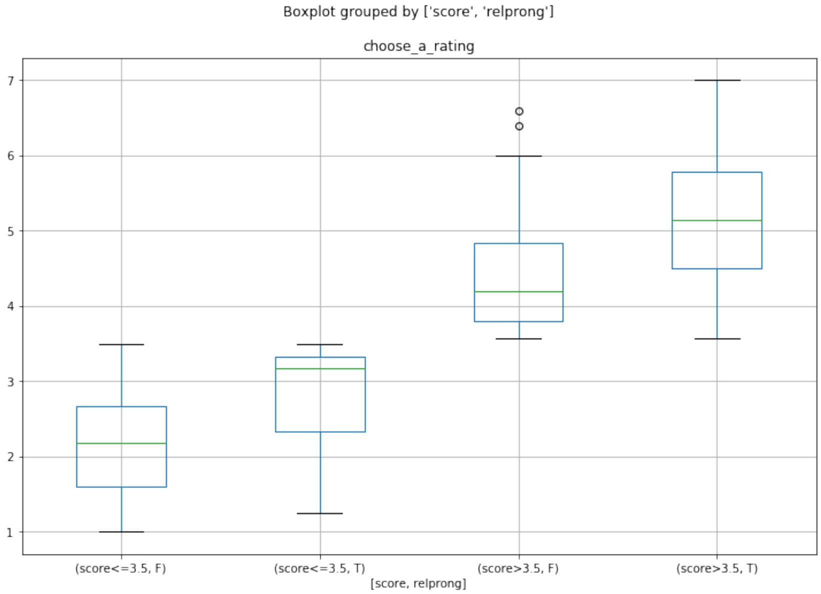

Figure 7.4: The boxplot shows the distribution of scores in the four classed individuated by table 7.9.

7.4 Final remarks

The results above, however partial, express well how faceted could the argumentation be on the performances of a model. We attempted to follow the outline proposed by Rimell et al. (2016), in order to make results as comparable as possible, and tried to introduce new information derived from our experiments.

Our results suggest that RELPRON, despite being hand-crafted and fairly small, shows many variables that could affect the performances in any way. Among the issues, there is the fact that the task has been defined as a ranking task on the whole dataset. This approach has two main issues: the performance on a batch of items depends greatly on which properties are included in the batch, as many variability factors (i.e., head nouns, lexical overlap) are not equally distributed throughout the dataset, moreover it is difficult to replicate the task with human annotators, making it hard to know whether our results reflect the way humans produce compositional meaning.

In order to overcome some of the issues, the complete *term-properties* matrix could be annotated with similarity judgements provided by native speakers: this would naturally let a batch of stable items emerge, and could give some insights on a hypothetical human performance on the complete ranking task as well.
Nonetheless, a broader and more naturalistic dataset is needed to properly evaluate compositionality models. As far as this is concerned, the structure of RELPRON is well suited to the task, as it makes it possible to isolate singular aspects of sentence meaning and evaluate them directly, while many among the bigger datasets do not allow for this kind of fine-grained analysis. For these reasons, we hope that RELPRON will be further expanded and validated.

rating	head noun	target	property
1,83	SBJ	bond	document that pay coupon
2,6	OBJ	form	document that council return
3	OBJ	lease	document that council terminate
3,17	SBJ	license	document that allow use
3,17	OBJ	specification	document that design meet
3,25	OBJ	form	document that parent sign
3,33	OBJ	bond	document that issuer redeem
3,4	SBJ	specification	document that provide functionality
3,5	OBJ	bond	document that government float

rating	head noun	target	property
1,6	SBJ	weight	quality that cause subsidence
2	OBJ	weight	quality that opinion carry
2,2	SBJ	weight	quality that damage skeleton
2,4	OBJ	mobility	quality that fracture impair
2,67	OBJ	rhythm	quality that defibrillation restore
3	OBJ	accuracy	quality that interference reduce
3	OBJ	accuracy	quality that scholar dispute
3,2	OBJ	wisdom	quality that hermit dispense
3,2	OBJ	morality	quality that ministry emphasize
3,2	OBJ	accuracy	quality that speed affect
3,4	SBJ	rhythm	quality that determine timing
3,4	OBJ	likelihood	quality that distribution determine
3,5	OBJ	accuracy	quality that uncertainty limit

rating	head noun	target	property
1,25	SBJ	bowler	player that dominate batsman
1,4	SBJ	bowler	player that finish spell
1,6	SBJ	bowler	player that use yorker
1,8	OBJ	bowler	player that humidity assist
2	SBJ	bowler	player that dismiss batsman
2,2	OBJ	bowler	player that batsman face
2,25	SBJ	golfer	player that hit wedge
2,57	SBJ	bowler	player that concede run
2,8	OBJ	bowler	player that batsman dominate
3	SBJ	golfer	player that have handicap
3,2	SBJ	golfer	player that use iron
3,29	SBJ	pitcher	player that walk batter
3,4	SBJ	bowler	player that take wicket
3,43	SBJ	batter	player that reach base
3,5	SBJ	pitcher	player that snap wrist

rating	head noun	target	property
1,75	SBJ	dial	device that make revolution
2,33	OBJ	button	device that jeans feature
2,8	OBJ	pipe	device that insulation cover
3	SBJ	telescope	device that collect light
3,17	OBJ	pipe	device that shepherd play
3,2	OBJ	dial	device that timer use
3,2	SBJ	saw	device that make plank
3,2	SBJ	telescope	device that have mirror
3,4	SBJ	dial	device that show time

rating	head noun	target	property
1,6	OBJ	division	organization that company sell
1,6	SBJ	mission	organization that monitor election
2	OBJ	family	organization that mobster represent
2,2	SBJ	mission	organization that convert population
2,33	SBJ	railway	organization that serve quarry
2,6	OBJ	family	organization that father abandon
2,6	SBJ	garrison	organization that hold city
3	SBJ	railway	organization that build station
3,17	SBJ	railway	organization that carry slate
3,2	SBJ	navy	organization that blockade port
3,2	SBJ	navy	organization that establish blockade
3,2	SBJ	division	organization that undergo merger
3,4	OBJ	division	organization that corps include

rating	head noun	target	property
2	OBJ	survivor	person that seaplane spot
2,25	OBJ	killer	person that soldier be
2,4	OBJ	traveler	person that consulate help
3	OBJ	killer	person that profiler find
3	SBJ	bomber	person that target marketplace
3,17	SBJ	philosopher	person that analyze ontology
3,2	OBJ	expert	person that panel include
3,4	SBJ	expert	person that author monograph
3,4	OBJ	expert	person that novice become
3,4	SBJ	survivor	person that suffer flashback
3,5	SBJ	expert	person that describe model

rating	head noun	target	property
3,2	OBJ	arena	building that rider enter
3,33	OBJ	abbey	building that order establish
3,33	OBJ	pub	building that brewer sell
3,33	SBJ	temple	building that hold festival
3,4	SBJ	house	building that line street
3,4	OBJ	house	building that hamlet have

Table 7.10: The tables show the complete set of pairs target - property that were actual RELPRON items, but were rated as false by human annotators. The pairs are split by head noun.

8 | Conclusion

Our work stemmed from the fact that it is now well established even in linguistic literature that event knowledge plays a significant role during semantic comprehension.

We provided a basic implementation of a model for meaning composition, which aimed 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 *generalized event knowledge* plays an important role in building semantic representations during online sentence processing.
This is also suggested by the fact that results obtained on the whole RELPRON dataset (*training set* + *test set*), which are reported in table 8.1, still show some beneficial effect of event knowledge. Because of the analysis discussed in chapter 7, these results need not be interpreted as a verdict on the usefulness of our kind of approach, and is just meant as an initial work towards a linguistically motivated and cognitively inspired model of sentence meaning composition.

The introduction of event knowledge has proven to overcome some of the issues, such as extreme sensibility to lexical overlap, that standard vector addition shows, especially on the considered datasets. Although this represents a promising result, many aspects need further investigations.

To start with, just a few among the many parameters that make it possible to specialize the framework have been explored. In particular, DEG has been created taking into account only pairs of lexical items, while it would be interesting to

	LM	AC	LM + AC
Development Set	0.51	0.47	0.55
Test Set	0.47	0.37	0.46
Full RELPRON	0.42	0.36	0.44

Table 8.1: The table shows the results obtained by the best performing model on RELPRON, as shown by table 6.9, on the test set and the full dataset. Results on each row are not comparable in terms of intensity, as the ranking task is performed on sets of different sizes.

build it with more structured sets of participants. This would allow us to explore a number of untouched areas concerning event knowledge.
In our setting, expectations work independently for each lexical item and for each target syntactic role of the sentence: when considering the case of *advisor* as a subject, for example, we retrieve *check*, *write*, *teach* as *roots*, and *thesis*, *book*, *course* as *direct objects*. However, it is clear that *advisor writes thesis* is less salient than *advisor checks thesis* as an event, and this does not happen because of the selectional preferences of the main verb. If participants could be retrieved jointly rather than independently, we would also have more structured predictions about upcoming items. Moreover, this would allow to generate expectations on sentence structure, while we are only querying DEG to get content (i.e., words).
Also related to this topic is the fact that, with the complete graph, events could be compared with respect to their neighbourhood on the graph, thus allowing similarity queries on the whole structure, rather than on the single participants.

With respect to pure vectorial models, the interface with the graph has two main drawbacks:

- the storage requires much more space than vectors;
- much more computational overload is placed at the processing phase.

We believe that both these drawbacks can be mitigated with appropriate algorithmic techniques, that allow for data compression and approximation with bounded error, which is enough for our purposes.

As far as the meaning composition function is concerned, we did not explore the use of *p-neighbours*, nor did we explore different implementations of the event knowledge blocks, the ones that collect expectations about upcoming fillers: one possibility for this would be to provide clusters of words, rather than lists, thus allowing for a more refined view of possible different areas of meaning for the retrieved event. Aside from this, many more options are available and worth exploring.

We only validated our framework on two, pretty small and non naturalistic datasets. The next step would be to try the same approach on a bigger dataset, that includes more syntactic structures than standard S-V-O sentences. Our analysis was meant as an exploratory one, and we needed to rule out of the picture any possible noisy variable in order to evaluate as more genuinely as possible the impact of event knowledge. Longer or more complex sentences, in fact, typically include a whole range of different arguments, and are more prone to parsing errors: these factors would have expanded too much the space of possibilities.

Moreover, many issues on RELPRON are far from being set. A more general statistical analysis of the results could still shed light on interesting aspects and could be helpful in identifying more precisely which aspects of compositional meaning are better modeled through the introduction of generalized event knowledge.

Last but not least, an interesting path would be to integrate non linguistic information into the model. There is an increasing interest among linguists toward how multimodal information interacts with the linguistic processing. While we only provided the outline for a language model, the same approach could in principle be extended through the distributional analysis of other sources of perceptual input, and could similarly be accessed via linguistic as well as non linguistic cues, providing a more holistic and grounded model of comprehension, as suggested by psycholinguistic and cognitive results.

Bibliography

Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Paşca, M., and Soroa, A. (2009). A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27. Association for Computational Linguistics.

Agirre, E., Cer, D., Diab, M., Gonzalez-Agirre, A., and Guo, W. (2013). * sem 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, volume 1, pages 32–43.

Agirre, E., Diab, M., Cer, D., and Gonzalez-Agirre, A. (2012). Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 385–393. Association for Computational Linguistics.

Asher, N. (2011). *Lexical meaning in context: A web of words*. Cambridge University Press.

Asher, N., Van de Cruys, T., Bride, A., and Abrusán, M. (2017). Integrating type theory and distributional semantics: A case study on adjective–noun compositions. *Computational Linguistics*.

Baggio, G., Van Lambalgen, M., and Hagoort, P. (2012). The processing consequences of compositionality. In *The Oxford handbook of compositionality*, pages 655–672. Oxford University Press.

Baroni, M. and Lenci, A. (2010). Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721.

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

Bentivogli, L., Bernardi, R., Marelli, M., Menini, S., Baroni, M., and Zamparelli, R. (2016). Sick through the semeval glasses. lesson learned from the evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. *Language Resources and Evaluation*, 50(1):95–124.

Bernardi, R., Dinu, G., Marelli, M., and Baroni, M. (2013). A relatedness benchmark to test the role of determiners in compositional distributional semantics. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 53–57.

Bicknell, K., Elman, J. L., Hare, M., McRae, K., and Kutas, M. (2010). Effects of event knowledge in processing verbal arguments. *Journal of memory and language*, 63(4):489–505.

Biemann, C. (2016). Vectors or graphs? on differences of representations for distributional semantic models. In *Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon (CogALex-V)*, pages 1–7.

Biemann, C., Coppola, B., Glass, M. R., Gliozzo, A., Hatem, M., and Riedl, M. (2013). Jobimtext visualizer: a graph-based approach to contextualizing distributional similarity. *Proceedings of TextGraphs-8 Graph-based Methods for Natural Language Processing*, pages 6–10.

Biemann, C. and Giesbrecht, E. (2011). Distributional semantics and compositionality 2011: Shared task description and results. In *Proceedings of the workshop on distributional semantics and compositionality*, pages 21–28. Association for Computational Linguistics.

Blacoe, W. and Lapata, M. (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.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association of Computational Linguistics*, 5(1):135–146.

Boleda, G., Baroni, M., McNally, L., et al. (2013). Intensionality was only alleged: On adjective-noun composition in distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)–Long Papers*, pages 35–46.

Boleda, G., Vecchi, E. M., Cornudella, M., and McNally, L. (2012). First-order vs. higher-order modification in distributional semantics. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1223–1233. Association for Computational Linguistics.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Bride, A. (2017). *Mathematical methods for Compositionality in Distributional Semantics for Natural Language Processing*. Phd thesis, Université Paul Sabatier, Toulouse, France.

Bride, A., Van de Cruys, T., and Asher, N. (2015). A generalisation of lexical functions for composition in distributional semantics. In *ACL (1)*, pages 281–291.

Bruni, E., Boleda, G., Baroni, M., and Tran, N.-K. (2012). Distributional semantics in technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 136–145. Association for Computational Linguistics.

Bruni, E., Tran, N.-K., and Baroni, M. (2014). Multimodal distributional semantics. *J. Artif. Intell. Res.(JAIR)*, 49(2014):1–47.

Callison-Burch, C., Fordyce, C., Koehn, P., Monz, C., and Schroeder, J. (2007). (meta-) evaluation of machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 136–158. Association for Computational Linguistics.

Callison-Burch, C., Fordyce, C., Koehn, P., Monz, C., and Schroeder, J. (2008). Further meta-evaluation of machine translation. In *Proceedings of the third workshop on statistical machine translation*, pages 70–106. Association for Computational Linguistics.

Chen, D. L. and Dolan, W. B. (2011). Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 190–200. Association for Computational Linguistics.

Chersoni, E. (2018). *Explaining complexity in Human Language Processing: a Distributional Semantic Model*. PhD thesis.

Chersoni, E., Blache, P., and Lenci, A. (2016). Towards a distributional model of semantic complexity. In *Proceedings of the Workshop on Computational Linguistics for Linguistic Complexity (CL4LC)*, pages 12–22.

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

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

Cheung, J. C. and Penn, G. (2012). Evaluating distributional models of semantics for syntactically invariant inference. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 33–43. Association for Computational Linguistics.

Clark, S., Coecke, B., and Sadrzadeh, M. (2008). A compositional distributional model of meaning. In *Proceedings of the Second Quantum Interaction Symposium (QI-2008)*, pages 133–140.

Clark, S. and Pulman, S. (2007). Combining symbolic and distributional models of meaning. In *Proceedings of the AAAI Spring Symposium on Quantum Interaction Stanford CA 2007*.

Coecke, B., Clark, S., and Sadrzadeh, M. (2010). Mathematical foundations for a compositional distributional model of meaning. Technical report.

Culicover, P. W. and Jackendoff, R. (2006). The simpler syntax hypothesis. *Trends in cognitive sciences*, 10(9):413–418.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

Deese, J. (1966). *The structure of associations in language and thought*. Johns Hopkins University Press.

Dolan, B., Quirk, C., and Brockett, C. (2004). Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th international conference on Computational Linguistics*, page 350. Association for Computational Linguistics.

Elman, J. L. (2009). On the meaning of words and dinosaur bones: Lexical knowledge without a lexicon. *Cognitive science*, 33(4):547–582.

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

Elman, J. L. (2014). 5 systematicity in the lexicon: On having your cake and eating it too. *The Architecture of Cognition: Rethinking Fodor and Pylyshyn's Systematicity Challenge*, page 115.

Erk, K. (2016). What do you know about an alligator when you know the company it keeps? *Semantics and Pragmatics*, 9:17–1.

Erk, K. and Padó, S. (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.

Erk, K. and Padó, S. (2010). Exemplar-based models for word meaning in context. In *Proceedings of the ACL 2010 conference short papers*, pages 92–97. Association for Computational Linguistics.

Erk, K., Padó, S., and Padó, U. (2010). A flexible, corpus-driven model of regular and inverse selectional preferences. *Computational Linguistics*, 36(4):723–763.

Feng, Y. and Lapata, M. (2010). Visual information in semantic representation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 91–99. Association for Computational Linguistics.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., and Ruppin, E. (2001). Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414. ACM.

Firth, J. R. (1957). A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*.

Gerz, D., Vulić, I., Hill, F., Reichart, R., and Korhonen, A. (2016). SimVerb-3500: A Large-Scale Evaluation Set of Verb Similarity. In *EMNLP*.

Giesbrecht, E. (2010). Towards a matrix-based distributional model of meaning. In *Proceedings of the NAACL HLT 2010 Student Research Workshop*, pages 23–28. Association for Computational Linguistics.

Goldberg, A. E. (2003). Constructions: A new theoretical approach to language. *Trends in cognitive sciences*, 7(5):219–224.

Grefenstette, E. and Sadrzadeh, M. (2011). Experimental support for a categorical compositional distributional model of meaning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1394–1404. Association for Computational Linguistics.

Gupta, A., Utt, J., and Padó, S. (2015). Dissecting the practical lexical function model for compositional distributional semantics. In * *SEM@ NAACL-HLT*, pages 153–158.

Hagoort, P. (2013). Muc (memory, unification, control) and beyond. *Frontiers in psychology*, 4:416.

Hagoort, P. (2015). Muc (memory, unification, control): A model on the neurobiology of language beyond single word processing. In *Neurobiology of language*, pages 339–347. Elsevier.

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

Hare, M., Jones, M., Thomson, C., Kelly, S., and McRae, K. (2009). Activating event knowledge. *Cognition*, 111(2):151–167.

Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3):146–162.

Hill, F., Reichart, R., and Korhonen, A. (2015). Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Jones, M. N. and Mewhort, D. J. (2007). Representing word meaning and order information in a composite holographic lexicon. *Psychological review*, 114(1):1.

Kanerva, P., Kristoferson, J., and Holst, A. (2000). Random indexing of text samples for latent semantic analysis. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 22.

Kartsaklis, D. and Sadrzadeh, M. (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.

Kintsch, W. (2001). Predication. *Cognitive science*, 25(2):173–202.

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

Lambek, J. (1997). Type grammar revisited. In *International Conference on Logical Aspects of Computational Linguistics*, pages 1–27. Springer.

Lascarides, A. and Copestake, A. (1998). Pragmatics and word meaning. *Journal of linguistics*, 34(2):387–414.

Lebani, G. E. and Lenci, A. (2018). 6 a distributional model of verb-specific semantic roles inferences. *Language, Cognition, and Computational Models*, page 118.

Lenci, A. (2008). Distributional semantics in linguistic and cognitive research. *Italian journal of linguistics*, 20(1):1–31.

Lenci, A. (2018). Distributional models of word meaning. *Annual review of Linguistics*, 4:151–171.

Levy, O. and Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, pages 2177–2185.

Levy, O., Goldberg, Y., and Dagan, I. (2015). Improving distributional similarity with lessons learned from word embeddings. *TACL*, 3:211–225.

Lin, D. and Pantel, P. (2001). Discovery of inference rules for question-answering. *Natural Language Engineering*, 7(4):343–360.

Luong, T., Socher, R., and Manning, C. (2013). Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 104–113.

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., and McClosky, D. (2014). The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Marelli, M., Bentivogli, L., Baroni, M., Bernardi, R., Menini, S., and Zamparelli, R. (2014). Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pages 1–8.

McRae, K., Hare, M., Elman, J. L., and Ferretti, T. (2005). A basis for generating expectancies for verbs from nouns. *Memory & Cognition*, 33(7):1174–1184.

McRae, K. and Matsuki, K. (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.

McRae, K., Spivey-Knowlton, M. J., and Tanenhaus, M. K. (1998). Modeling the influence of thematic fit (and other constraints) in on-line sentence comprehension. *Journal of Memory and Language*, 38(3):283–312.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013a). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Mikolov, T., Yih, W.-t., and Zweig, G. (2013b). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751.

Milajevs, D., Kartsaklis, D., Sadrzadeh, M., and Purver, M. (2014). Evaluating neural word representations in tensor- based compositional settings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. J. (1990). Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244.

Miller, G. A. and Charles, W. G. (1991). Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28.

Mitchell, J. and Lapata, M. (2008). Vector-based models of semantic composition.

Mitchell, J. and Lapata, M. (2010). Composition in distributional models of semantics. *Cognitive science*, 34(8):1388–1429.

Paperno, D., Baroni, M., et al. (2014). A practical and linguistically-motivated approach to compositional distributional semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 90–99.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Pham, N., Bernardi, R., Zhang, Y. Z., and Baroni, M. (2013). Sentence paraphrase detection: When determiners and word order make the difference. In *Proceedings of the IWCS 2013 Workshop Towards a Formal Distributional Semantics*, pages 21–29.

Pustejovsky, J. (1991). The generative lexicon. *Computational linguistics*, 17(4):409–441.

Radinsky, K., Agichtein, E., Gabrilovich, E., and Markovitch, S. (2011). A word at a time: computing word relatedness using temporal semantic analysis. In *Proceedings of the 20th international conference on World wide web*, pages 337–346. ACM.

Recchia, G., Jones, M., Sahlgren, M., and Kanerva, P. (2010). Encoding sequential information in vector space models of semantics: Comparing holographic reduced representation and random permutation. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 32.

Reddy, S., McCarthy, D., and Manandhar, S. (2011). An empirical study on compositionality in compound nouns. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 210–218.

Rimell, L., Maillard, J., Polajnar, T., and Clark, S. (2016). Relpron: A relative clause evaluation data set for compositional distributional semantics. *Computational Linguistics*, 42(4):661–701.

Robinson, I., Webber, J., and Eifrem, E. (2013). *Graph databases*. " O'Reilly Media, Inc.".

Rubenstein, H. and Goodenough, J. B. (1965). Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633.

Rudolph, S. and Giesbrecht, E. (2010). Compositional matrix-space models of language. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 907–916. Association for Computational Linguistics.

Rumelhart, D. E. (1979). Some problems with the notion of literal meanings. *Metaphor and thought*, 2:71–82.

Sahlgren, M. (2006). *The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces*. PhD thesis.

Sahlgren, M., Holst, A., and Kanerva, P. (2008). Permutations as a means to encode order in word space. In *The 30th Annual Meeting of the Cognitive Science Society (CogSci'08), 23-26 July 2008, Washington DC, USA*.

Sahlgren, M. and Lenci, A. (2016). The effects of data size and frequency range on distributional semantic models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 975–980.

Salton, G., Wong, A., and Yang, C.-S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

Schütze, H., Manning, C. D., and Raghavan, P. (2008). *Introduction to information retrieval*, volume 39. Cambridge University Press.

Smolensky, P. and Legendre, G. (2006). *The harmonic mind: From neural computation to optimality-theoretic grammar (Cognitive architecture), Vol. 1.* MIT press.

Socher, R., Huval, B., Manning, C. D., and Ng, A. Y. (2012). Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 1201–1211. Association for Computational Linguistics.

Socher, R., Pennington, J., Huang, E. H., Ng, A. Y., and Manning, C. D. (2011). 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.

Wittgenstein, L. (1953). Philosophical investigations (gem anscombe, trans.).

Yang, D. and Powers, D. M. (2006). Verb similarity on the taxonomy of wordnet. *Proceedings of GWC-06*, pages 121–128.

Zanzotto, F. M., Korkontzelos, I., Fallucchi, F., and Manandhar, S. (2010). Estimating linear models for compositional distributional semantics. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1263–1271. Association for Computational Linguistics.

Zarcone, A., Padó, S., and Lenci, A. (2014). Logical metonymy resolution in a words-as-cues framework: Evidence from self-paced reading and probe recognition. *Cognitive science*, 38(5):973–996.