Quantum Logic Unites Compositional Functional Semantics and Distributional Semantic Models

Anne Preller, Violaine Prince

Quantum logic unites compositional functional semantics and distributional semantic models

Anne Preller, Violaine Prince[*][†]
Informatique
LIRMM/CNRS
Montpellier, France

1   Introduction

When we retrieve information from text by statistical methods, we apply these methods not to random strings of words but to sentences, paragraphs etc. They are ruled by laws of logic inherent to language. Natural language conveys information about individuals (extension) using concepts (intension). The extensional aspect is captured by the familiar logical models, the intensional aspect by distributional semantic models, DSM's. We propose a unique frame for both DSM's and functional logical models and show how compositionality of the latter transfers to compositionality of the former. The frame is the theory of compact closed monoidal categories, materialised by the category of finite-dimensional vector spaces for semantics and the category of proofs of compact bilinear logic for syntax, [Lambek 1993].

An essential difference with previous approaches is that we consider a DSM as a finite-dimensional space over the lattice of real numbers and not over the field of real numbers as do for example [Clark, Coecke, Sadrzadeh]. In this way, we capture both the logical and the numerical content of a DSM. Indeed, the lattice structure of subspaces, with logical operators defined by quantum logic, [van Rijsbergen], on one hand and the partial order of vectors on the other hand are isomorphic. Moreover, the lattice structure is distributive, unlike in Hilbert spaces. Negation, however, remains orthogonality.

Next, we study compositionality in the particular case where the basis vectors of the DSM correspond to strings of key words, the so-called *concept space*. The logic of a concept spaces corresponds to a classification of words by a thesaurus. A particular meaning of a word is given by a set of (present or absent) features. This meaning is represented by the vector standing for the conjunction of the features. In the case where the features are the same as the key words, the conjunction coincides with a tensor product. An ambiguous word

with several meanings gives rise to a disjunction of conjunctions in a concept space that includes the concept spaces of the particular meanings. Concepts defined by strings of words are vectors in a superspace that depends on the concept space of every word, including the words figuring under 'noise' like determiners, prepositions etc.

Syntactical analysis is required to define the meaning vector of a string of words. Pregroup grammars, [Lambek 1999], are particularly appropriate for this task, because the meaning expressions they associate to words can be read as vectors in a finite dimensional space. Moreover, parsing consists in a proof of compact bilinear logic and can be read as a linear map between finite dimensional vector spaces. The vectors belong to tensor products of three basic spaces, one for individuals, one for truth-values and an arbitrary DSM. The meaning of a grammatical string of words is obtained by forming the tensor product of the word vectors and applying the parsing map to it. Every word in a grammatical string contributes to its meaning vector. Words that are distributional 'noise' are decoded as functions used in the computation of the vector of the string. Moreover, a sentence vector is true if and only if the corresponding formula in predicate logic holds if and only if the corresponding inequality of concepts holds. There are two reasons for this. First, the basis vectors of the DSM, i.e. words, carry an inbuilt logic inherited from a concept space. Second, the meaning vectors are expressions in compact closed monoidal categories, which can also be read as predicates and functions in two-sorted first order logic, [Preller, 2005, 2007].

2   A simplified quantum logic

Distributional semantic models represent words by vectors of a finite dimensional real space. The coordinates of the vectors are in general obtained by a frequency count in text-windows and belong to a bounded set, which can be assumed to be the real interval $I = [0, 1]$. The order of real numbers induces a distributive and implication-complemented lattice structure on I, namely $\alpha \vee \beta = \max \{\alpha, \beta\}$, $\alpha \wedge \beta = \min \{\alpha, \beta\}$, $\alpha \rightarrow \beta = \max \{\gamma \in I : \alpha \wedge \gamma \leq \beta\}$ and $\neg \alpha = \alpha \rightarrow 0$. It also defines a semi-ring structure on I

$$\alpha + \beta = \alpha \vee \beta \qquad \alpha \cdot \beta = \alpha \wedge \beta,$$

with 0 the neutral element of addition and 1 the unit of multiplication. Hence, semantic vectors belong to a semi-module over the semi-ring I.

Many of the definitions familiar from vector spaces over the field $\mathbb{R}$ carry over to vector spaces (semi-modules) over the semi-ring I, called I-*spaces* for short. Among them are the inner product, orthogonality, the tensor product, linear maps $f : V \rightarrow W$ with the corresponding matrix $\overline{f} \in V^* \otimes W$, where V^* is the dual of V.

The lattice structure on I lifts to the vectors in an I-space V, where the lattice operators $(v, v) \mapsto v \vee v = v + v$, are defined coordinate by coordinate.

Hence the I-space V is an implication-complemented distributive lattice with smallest element $\overrightarrow{0}$ and largest elements $\overrightarrow{1}$.

The lifted lattice structure is isomorphic to the lattice structure defined by quantum logic on the projectors (equivalently, subspaces) of the I-space V. Indeed, according to [van Rijsbergen], the logical connectives of quantum logic are defined by

$$\neg E = E^{\perp},\ E \vee F = E \oplus F,\ E \wedge F = E \circ F,\ E \to F = \{u\,:\,F(E(u)) = E(u)\}\ .$$

Indeed, we prove that the map

$$w \mapsto E_w = \{v\,:\,v \leq w\}$$

that sends $w \in V$ to a closed subspace is the required lattice isomorphism.

The vector w such that $E = E_w$ is unique. It is said to *internalises* E. Subspaces generated by a subset of m basis vectors are maximal among the subspaces of dimension m. The maximal subspaces form a Boolean sublattice. They are internalised by vectors with coefficients in $\{0,1\}$. Indeed, the space generated by a set A of basis vectors is internalised by the vector $\sum_{x \in A} x$.

Viewing the basis vectors as the elements of some 'universe of discourse', the lattice operators of the I-space work like union, intersection and complement of subsets. Viewing the basis vectors as concepts, the same operators work like propositional logical connectives. Quantum logic defines both the set-theory of the universe of discourse as well as its logic. In the next section, we consider the case where the lattice operators are logical connectives. In the last section, pregroup grammars unite the extensional (individuals) and the intensional aspect (concepts) of discourse.

3   Concept Spaces

Roughly speaking, concept spaces are distributional semantic models that are based on a classification of concepts by key words. They play an essential role when linking the inbuilt quantum logic of a DSM with the predicate logic introduced on vector spaces via pregroup grammars in the next section. A concept space may be also viewed as an event space, the basic event being the occurrence of a key word.

Indeed, consider a set $P = \{p_1, \ldots, p_d\}$ of *basic concepts* or *key words*. The *concept space defined by* P is the tensor product $C(P) = C_1 \otimes \ldots \otimes C_d$ of two-dimensional I-spaces C_i with basis vectors $p_{i\top},\ p_{i\perp}$. A basis vector $b_f \in C_1 \otimes \ldots \otimes C_d$ is essentially a choice $f \in \prod_{i=1}^{d} \{p_{i\top},\ p_{i\perp}\}$ between 'yes' $= p_{i\top}$ and 'no'$= p_{i\perp}$ for every feature p_i. Therefore, the vector of $C(P)$ corresponding to the key word p_i is

$$\overrightarrow{p_i} = \sum_{f, f(i) = p_{i\top}} b_f \in C(P)$$

Then $C(P)$ includes the free Boolean algebra $B(P)$ generated by P, namely the lattice generated the set $\{\overrightarrow{p_1}, \ldots, \overrightarrow{p_d}\} \cup \{\neg\overrightarrow{p_1}, \ldots, \neg\overrightarrow{p_d}\}$.

Moreover, every individual x defines a basis vector $b_x = q_1 \otimes \ldots \otimes q_d$, where $q_i = p_{i\top}$ if the individual x satisfies the feature p_i and $q_i = p_{i\perp}$ else. For example, the word *bank* is represented by the concept vector

$$\overrightarrow{\texttt{bank}} = \sum_{x \in Bank} b_x \in C(P)\,,$$

where *Bank* is the set of individuals designated by the word *bank*.

Suppose that a disambiguation algorithm returns the word *bank* similar to $p_1 = slope$, $p_2 = shore$ and $p_3 = space$ in one context and to $p_4 = store$, $p_5 = treasury$ and $p_6 = volition$ in another context. Hence the vectors $b_1 = \overrightarrow{p_1} \wedge \overrightarrow{p_2} \wedge \overrightarrow{p_3} \in C(P)$ respectively $b_2 = \overrightarrow{p_4} \wedge \overrightarrow{p_5} \wedge \overrightarrow{p_6} \in C(P)$ must correspond to concepts approximating the two meanings conveyed by *bank*. In fact, we prove that every individual (thing) designated by the word *bank* satisfies the property $b_1 + b_2 = b_1 \vee b_2$, i.e. the following inequality holds in $C(P)$

$$\overrightarrow{\texttt{bank}} \leq b_1 + b_2\,.$$

Concept spaces are explicit in classification systems, but they also lurk in the background of an arbitrary distributional semantic model C. For example, if each word occurring in a set of documents is identified with a basis vector of the distributional semantic model C, the basis vectors also live in a space $C(P)$ of a thesaurus, be it the mental thesaurus of the speaker or the Roget's Thesaurus. Otherwise said, the set A of basis vectors of C identifies with the lattice of vectors of $C(P)$ with coordinates in $\{0,1\}$ or, equivalently, the lattice of subspaces of $C(P)$ generated by subsets of basis vectors. For a given set P, we refer to the lattice structure of the basis of C by writing $C = V_{C(P)}$. This explains to a large extent the working of the classification algorithm, [Clark, Pulman], applied to large corpora and based on the French thesaurus [Larousse 1992].

The set of key words P may vary from one set of documents to the next and from one speaker to the next. The logic remains unchanged. Indeed, there are embeddings from $C(P_1)$ and $C(P_2)$ into $C(P_1 \cup P_2)$ preserving the logical connectives. If, as in the example above, P_1 and P_2 are disjoint then $C(P_1 \cup P_2) = C(P_1) \otimes C(P_2)$. Hence, $C(P) = C(\{p_1\}) \otimes \ldots \otimes C(\{p_d\})$, where each $C(\{p_i\})$ is a 2-dimensional I-space. Note that $C(\emptyset)$ is isomorphic to I.

The understanding that the tensor product of constituent vectors allows to represent strings of words can be found already in [Smolensky], [Clark, Pulman] and [Widdows] among others. Our contribution is to reduce the spaces to tensor products of two-dimensional spaces, the linguistic analogue to quantum protocols, [Abramsky, Coecke], where the basic space is the 2-dimensional Hilbert space of 'qubits'. The concept spaces are the linguistic analogue of 'compound systems' of quantum mechanics, which are finite tensor products of two-dimensional Hilbert spaces. Choosing the lattice structure instead of the field structure of the real numbers, we embed classical logic into quantum logic. Natural language uses concepts but talks (also) about individuals. This brings the quantum phenomenon of 'entanglement' to DSM's, illustrated by the vector $\overrightarrow{bank}$ of the example above, because it cannot be decomposed into a tensor

product of vectors belonging to the basic factors. Teleportation also has its linguistic analogue, as we shall see in the next section. It is caused by a word which is distributional 'noise', but is known as a generalised quantifier in logic.

4   Compositional functional semantics for DSM's

Grammatical analysis steps in for a compositional representation of grammatical strings of words in a DSM. Pregroup grammars, [Lambek 1999], are particularly appropriate for the task, because they interpret words and sentences in a compact closed symmetric monoidal category in general vector spaces in particular. Syntactical analysis consists in a proof of compact bilinear logic, [Lambek 1993]. Such a proof is a map in the free compact closed monoidal category and therefore also a linear map of vector spaces.

In fact, the linear maps can be avoided, because they identify with vectors. To be precise, the linear map $f : V \to W$ identifies with a matrix $\overline{f}$, a vector in $W^* \otimes V$ or in the isomorphic space $V \otimes W^*$, whichever is more convenient. Here, W^* denotes the dual space of W. On the other hand, vectors identify with linear maps, namely the vector $v \in V$ identifies with the linear map $v : I \to V$ that assigns to the unique basis vector of the one-dimensional space I the element v of V. In this case, $\overline{v} = v$.

This double representation is exploited by pregroup grammars. The meanings of words w_i are vectors $\overline{\mathrm{m}}_i : I \to W_i$. The *reduction*, (the pregroup equivalent of a parsing tree) of a grammatical string $w_1 \ldots w_n$ defines a linear map $r : W_1 \otimes \ldots \otimes W_n \to V$. The meaning of the string is the vector $r \circ (\overline{\mathrm{m}}_1 \otimes \ldots \otimes \overline{\mathrm{m}}_n) : I \to V$.

The space V depends on the grammatical analysis of the string. For a sentence, it is a two-dimensional space $S = V_{C(\emptyset)}$ with basis vectors $\top$ and $\bot$, called *truth-values*. For a noun phrase, it is a space E, the basis vectors of which stand for *individuals (entities)*. For a predicative adjective phrase, it is a space of 'properties' C, the distributional semantic model $C = V_{C(P)}$. The basis vectors of C bring with them the logical operators of $C(P)$. For example, the vector $\overrightarrow{\mathtt{safe}} \in C(P)$, which is not a basis vector in $C(P)$, is considered a basis vector, denoted $\overline{\mathtt{safe}}$, in C. Hence, the logical connectives of $C(P)$ define maps from and to basis vectors of C. They extend uniquely to linear maps defined for all vectors of C. Note that they are different from the lattice operators of the I-space C.

For example, the negation $\neg : C(\emptyset) \to C(\emptyset)$ defines a linear map $\mathtt{not} : S \to S$ satisfying

$$\mathtt{not}(\top) = \bot,\ \mathtt{not}(\bot) = \top,\ \mathtt{not}(\overrightarrow{0}) = \overrightarrow{0},\ \mathtt{not}(\top + \bot) = \top + \bot.$$

Hence, the linear map $\mathtt{not}$ maps basis vectors to orthogonal basis vectors, but this does not hold for all vectors of S. The linearity of the logical connectives for sentences intervenes decisively when interpreting plurals and quantifiers.

Like all categorial grammars, a pregroup grammar has a lexicon. It consists of triples $w : T :: \overline{\mathrm{m}}$, where w is a word, T a type and $\overline{\mathrm{m}}$ a vector. For example, a

pregroup grammar that parses the sentence *banks are safe* includes the entries

$$
\begin{aligned}
banks &: \boldsymbol{n}_2 && :: I \xrightarrow{\overline{\text{bank}}} E \\
are &: \boldsymbol{n}_2{}^r \boldsymbol{s}\boldsymbol{a}^\ell && :: I \xrightarrow{\overline{\text{are}}} E^* \otimes S \otimes C^* \\
safe &: \boldsymbol{a} && :: I \xrightarrow{\overline{\text{safe}}} C
\end{aligned}
$$

The basic types $\boldsymbol{n}_2, \boldsymbol{s}, \boldsymbol{a}$ stand for 'plural noun phrase', 'sentence', 'predicative adjective' and are interpreted in the I-spaces E, S and C, in that order. Intuitively, a basic type with a superscript r or ℓ denotes a slot that has to be filled. If the superscript is r, the filler moves to the right. If it is ℓ, the filler moves leftward. A type with a superscript is interpreted by the dual of the space corresponding to the basic type.

Both the reduction to the sentence type and its corresponding linear map (on the right below) have a common graphical representation

The underlinks of the reduction become the inner product when computing the values of the corresponding linear map. Parsing with pregroup grammars is cubic polynomial in the worst case and linear in many practical cases. The extra-work needed for computing the meaning vector of the string is proportional to the number of words.

The meaning vectors of the words have a graphical representation as well, for example

Here, bank is the element in E identified with $\overline{\text{bank}}$ and $\text{are} : E \times C \to S$ is the bilinear map the matrix of which is $\overline{\text{are}} \in E^* \otimes S \otimes C^*$.

To compute the meaning of the sentence, form the tensor product of the word vectors and compose it with the linear map, which for the graphs means to connect them at the common interface

The easiest way to compute the equality between the lefthand and righthand expressions is to walk along the paths starting at I and picking up the labels as you encounter them. The plain arrows stand for channels through which the 'resources' are transmitted. The dotted arrows indicate 'entanglement'. The values of the arrows above the interface are provided by the semantic model. The values of the arrows below the interface are provided by syntactic analysis.

Pregroup grammars interpret words in several spaces and the outcome of a grammatical string may be in yet another space. The same quantum logic, however, operates on all spaces and therefore the representation $\overrightarrow{bank} \leq \overrightarrow{safe}$ of the statement *banks are safe* by the vector model C is equivalent to its representation $\mathtt{are}(\mathtt{bank}, \mathtt{safe}) = \top$ by the pregroup grammar. In fact, both are equivalent to the familiar representation $\forall x (x \in Bank \Rightarrow safe(x))$ of the sentence in predicate calculus.

Theorem 1. *The following are equivalent*

$$\overrightarrow{bank} \leq \overrightarrow{safe}$$
$$\mathtt{are}(\boldsymbol{bank}, \boldsymbol{safe}) = \top$$
$$\forall x (x \in Bank \Rightarrow safe(x))$$

Proof. First we remark that a predicate is either true or false for a given individual. Therefore the map $x \mapsto \mathtt{are}(x, \mathtt{safe})$ takes a basis vector x of E either to the basis vector $\top$ of S or to the basis vector $\bot$ of S.

Next, let $Safe = \{x : safe(x)\} = \{x : \mathtt{are}(x, \mathtt{safe}) = \top\}$ be the set of individuals for which the predicate *safe* holds. Then $\overrightarrow{\mathtt{safe}} = \sum_{x \in Safe} b_x \in C(P)$ is the corresponding concept vector. The definition of the basis vector $b_x \in C(P)$ associated to the individual x then implies that $\mathtt{are}(x, \mathtt{safe}) = \top$ if and only if $b_x \leq \overrightarrow{\mathtt{safe}}$.

On the other hand, the vector $\mathtt{bank} = \sum_{x \in Bank} x \in E$ internalises the subspace generated by the individuals designated by the word *bank*. By bilinearity,

$$\mathtt{are}(\mathtt{bank}, \mathtt{safe}) = \mathtt{are}(\sum_{x \in Bank} x, \mathtt{safe}) = \sum_{x \in Bank} \mathtt{are}(x, \mathtt{safe}).$$

By the initial remark, the equality $\sum_{x \in Bank} \mathtt{are}(x, \mathtt{safe}) = \top$ holds exactly when $\mathtt{are}(x, \mathtt{safe}) = \top$ holds for every $x \in Bank$. Hence, definition of b_x implies $\overrightarrow{bank} = \sum_{x \in Bank} b_x \leq \overrightarrow{safe}$ if and only if $\mathtt{are}(\mathtt{bank}, \mathtt{safe}) = \top$. □

A similar argument shows these equivalences for other sentences. Here is an example that illustrates how *no* provokes 'teleportation' in

$$no\ banks\ are\ safe$$
$$\overrightarrow{bank} \leq \neg\overrightarrow{safe}$$
$$\mathtt{not}(\mathtt{are}(\mathtt{bank}, \mathtt{safe})) = \top$$
$$\forall x (x \in Bank \Rightarrow \neg safe(x)).$$

The relevant lexical entry of the pregroup grammar is

$$no : \boldsymbol{ss^\ell n_2 n_2^\ell} :: I \xrightarrow{\overline{\mathtt{no}}} S \otimes S^* \otimes E \otimes E^*$$

where the vector $\overline{\overline{\text{no}}}$ is the matrix of the linear map $\text{not} \otimes id_E : S \otimes E \to S \otimes E$.

The reduction of the sentence, its corresponding linear map, and the graph representing the vector $\overline{\overline{\text{no}}}$ are

$$
\begin{array}{c}
\textit{No} \quad \textit{banks} \quad \textit{are} \quad \textit{safe} \\
(\boldsymbol{s}\,\boldsymbol{s}^{\ell}\boldsymbol{n}_2\boldsymbol{c}_2{}^{\ell})(\boldsymbol{c}_2)(\boldsymbol{n}_2{}^{r}\boldsymbol{s}\,\boldsymbol{a}^{\ell})(\boldsymbol{a})
\end{array}
$$

$$
r' \;=\; \begin{array}{c} (S \otimes S^* \otimes E \otimes E^*) \otimes (E) \otimes (E^* \otimes S \otimes C^*) \otimes (C) \\[2mm] \downarrow \\ S \end{array}
$$

$$
\overline{\overline{\text{no}}} : I \to S \otimes S^* \otimes E \otimes E^* \;\;=\;\; \begin{array}{c} I \\ \text{not} \\ S \otimes S^* \otimes E \otimes E^* \end{array}\;.
$$

Again, the meaning vector of the sentence is the tensor product of the word vectors composed with the reduction to the sentence type.
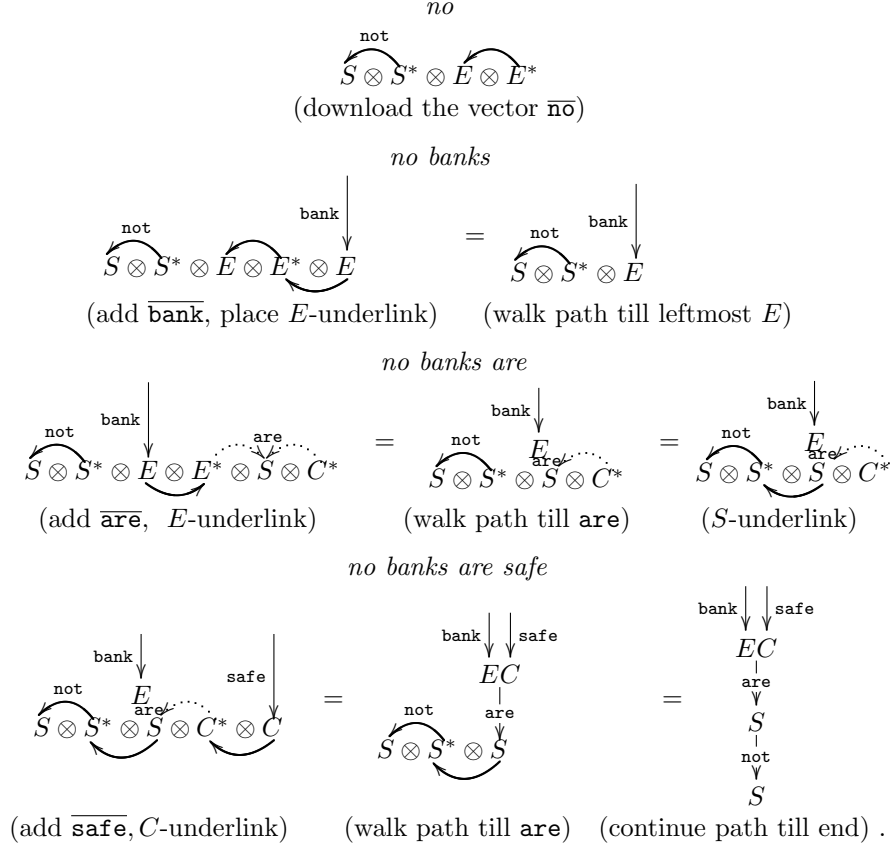
$$
r' \circ \big(\overline{\overline{\text{no}}} \otimes \overline{\text{bank}} \otimes \overline{\text{are}} \otimes \overline{\text{safe}}\big) =
$$

$$
(S \otimes S^* \otimes E \otimes E^*) \otimes (E) \otimes (E^* \otimes S \otimes C^*) \otimes (C)
$$

$$
= \text{not}(\text{are}(\text{bank}, \text{safe}))
$$

Again, one computes the result by walking the paths from I, picking up the labels encountered on each path.

The vector $\overline{\overline{\text{no}}}$ has two pieces: the overlink labelled not standing for the negation in the determiner *no* and the unlabelled overlink for the universal quantification contained in *no*. The negative quantum is 'teleported' from the noun to the verb.

Making or hearing a statement is the linguistic analogue of an experiment in quantum mechanics. Lining up the semantic vectors as you hear the words corresponds to 'preparation' or putting into focus. Preparation concerns the arrows above the interface in the graph. Finding the meaning is 'observation'. The experiment requires time proportional to the number of words. Indeed, the algorithm processes the string of words from left to right. For every word it downloads the syntactic type and the meaning vector. It constructs the reduction by placing an underlink as soon as possible. If a path through several arrows comes into existence it is 'walked' at once. The intermediary nodes along

the path are erased. We illustrate the procedure with the sentence *no banks are safe*.

$$no$$
$$\overset{\text{not}}{\overparen{S \otimes S^* \otimes E \otimes E^*}}$$
(download the vector $\overline{\text{no}}$)

$$no\ banks$$

$$\overset{\text{not}}{\overparen{S \otimes S^*}} \otimes \overset{\text{bank}}{E} \otimes \overparen{E^* \otimes E} \qquad = \qquad \overset{\text{not}}{\overparen{S \otimes S^*}} \otimes \overset{\text{bank}}{E}$$
(add $\overline{\text{bank}}$, place E-underlink) (walk path till leftmost E)

$$no\ banks\ are$$

$$\overset{\text{not}}{\overparen{S \otimes S^*}} \otimes \overset{\text{bank}}{E} \otimes E^* \otimes \overset{\text{are}}{S} \otimes C^* = \overset{\text{not}}{\overparen{S \otimes S^*}} \otimes \overset{\text{bank}}{\underset{\text{are}}{E}} S \otimes C^* = \overset{\text{not}}{\overparen{S \otimes S^*}} \otimes \overset{\text{bank}}{\underset{\text{are}}{E}} S \otimes C^*$$
(add $\overline{\text{are}}$, E-underlink) (walk path till `are`) (S-underlink)

$$no\ banks\ are\ safe$$

$$\overset{\text{not}}{\overparen{S \otimes S^*}} \otimes \overset{\text{bank}}{\underset{\text{are}}{E}} S \otimes C^* \otimes \overset{\text{safe}}{C} = \overset{\text{not}}{\overparen{S \otimes S^*}} \otimes \overset{\overset{\text{bank}\ \text{safe}}{EC}}{\underset{\text{are}}{S}} = \begin{array}{c} \overset{\text{bank}\ \text{safe}}{EC} \\ | \\ \text{are} \\ \downarrow \\ S \\ | \\ \text{not} \\ \downarrow \\ S \end{array}$$
(add $\overline{\text{safe}}$, C-underlink) (walk path till `are`) (continue path till end) .

5   Conclusion

Behind the numerical coordinates of a vector in a distributional semantic model there is a logical structure. A grammatical string also has a logical structure brought into focus by syntactical analysis, taking into account the meaning of each word in the string. The coordinates of word(-vector)s with logical content like $\overline{\text{no}}$ are given by logic and therefore are the same in all models. Prepositions also carry meanings independent of the distribution, e.g. *with* and *without* correspond to *presence* respectively *absence*. They are irrelevant for the distribution of single words, but relevant when it comes to analyse strings of words and construct the corresponding concepts. The interaction of extensional and intensional representation relies on the fact that certain words define both an extensional vector and a concept vector, linked by an equivalence. The coordinates of concept vectors like `bank`, `safe` are provided by the 'user', for example frequencies in a set of documents. As already mentioned, a concept space may be viewed as an event space. A direction for future work is the comparison of

the properties of the probability distribution with the logical and grammatical properties of strings of words.

References

[Abramsky, Coecke] Samson Abramsky, Bob Coecke (2004), A categorical semantics of quantum protocols, in: *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science*, pp. 415 - 425

[Chauché, Prince] Jacques Chauché, Violaine Prince (2007), Classifying texts through natural language parsing and semantic filtering, Proceedings of the 3rd International Language a. Technology Conference, Poznan

[Clark, Coecke, Sadrzadeh] Stephen Clark, Bob Coecke, Mehrnoosh Sadrzadeh (2008), A Compositional Distributional Model of Meaning, *Proceedings of the Conference on Quantum Interactions*, U. of Oxford

[Clark, Pulman] Stephen Clark, S. Pulman (2007), Combining symbolic and distributional models of meaning, AAAI Press, Proceedings of the AAAI Spring Symposium on Quantum Interaction

[Lambek 1993] Joachim Lambek (1993), From categorial grammar to bilinear logic, in: K. Došen, P. Schroeder-Heister, (Eds.), *Substructural Logics*, Oxford University Press, pp. 207-237

[Lambek 1999] Joachim Lambek (1999), Type grammar revisited, in: A. Lecomte et al., (Eds.), *Logical Aspects of Computational Linguistics*, LNAI 1582, pp. 1-27

[Larousse 1992] Thésaurus Larousse - des idées aux mots, des mots aux idées, (1992), Ed. Larousse, Paris

[Preller 2005] Anne Preller (2005), Category Theoretical Semantics for Pregroup Grammars, in: Philippe Blache and Edward Stabler (Eds.), *Logical Aspects of Computational Linguistics*, LNAI 3492, pp. 254-270

[Preller 2007] Anne Preller (2007), Toward Discourse Representation Via Pregroup Grammars, JoLLI, Vol.16, pp. 173-194

[van Rijsbergen] C. van Rijsbergen (2004), The Geometry of Information Retrieval, Cambridge University Press

[Smolensky] Paul Smolensky (1988), Connectionism, Constituency and Language of Thought, in Robert Cummins, Denise Dellarosa Cummins, (Eds.), *Minds, Brains, and Computers*, pp.284-308

[Widdows] Dominic Widdows (2008), Semantic Vector Products: Some Initial Investigations, *Second AAAI Symposium on Quantum Interaction*, Oxford