Distributional Semantics

Lecture 11. Compositional Distributional Semantics

Florian Gouret

May 4th, 2019

Presentation of the seminar

Plan:

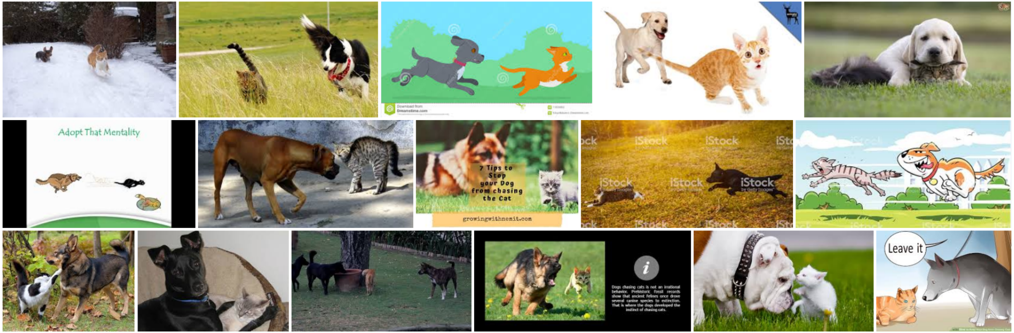
1) Introduction
2) Algebraic semantics
3) Background
4) Element of DisCoCat
5) Extension to DisCoCirc

Introduction: Origines

A cat chasing a dog

Introduction: Origins

A dog chasing a cat



Introduction:

Different queries for similar results...

Order of words within a sentence doesn't matter if everything is placed into a bag.

Introduction:

What about considering the order of words?
How to do it?
What about the grammar structure?

Algebraic Semantics:

If having a vector for each word of the corpora, it does not mean that everything can be done. For instance, the compositionality of the relation adjective-noun or subject-verb-object is far from being accurate all the time with simple product between terms. Moreover, the composition of words can completely change the meaning of sentences, as it is the case with *Fake guns can't kill* where the adjective *fake* changes the main property of its subject *guns*.

Algebraic Semantics: Adjectives as Linear Maps

The main purpose to use adjectives is to modify properties of a noun. Hence, it has to be found a way to encode a kind a function for adjective where the input would be the noun and the output the modified noun. The assumption made within is that adjective in attribute position can be seen as linear functions from R^n to R^n. In other words, adjectives are endomorphic linear maps in the noun space.

$$p_{amod} = B_{adj} v_{noun}$$

where p is the encoded relation "attribute-noun B is the weight matrix representing the adjective and v the vector representation of the noun. In this case, the adjective matrices are evaluated and adjusted one by one.

Algebraic Semantics: Verbs

Verb: In sentences, verbs are the grammatical center. They express an act or a mode of a given object, considered as subject. Hence, the amount of information they carry is rather important and needs a more advance structure than a simple vector to store them.
Intransitive verbs: From an extensional view, the meaning of an intransitive verb (IN) is the set of objects which perform the action denoted by this one. A formal definition is as follows:

$$[verb_{IN}] = \{x | x \in U \land x \ verb_{IN}\}$$

However this representation does not follow the type induced by the Lambek category for intransitive verb. This one should be $N \otimes S$ instead of N as it is now.

$$[verb_{IN}] = \{(x, t) | x \in U, t = \top if x \ verb_{IN} \ and \ \ \bot\}$$

$$\overrightarrow{v_{IN}} = \sum_{(n_i, \overrightarrow{s_i}) \in A} \overrightarrow{n_i} \otimes \overrightarrow{s_i}$$

Algebraic Semantics: Verbs

Transitive verbs: A formal definition of a transitive verb with 1 argument is as follows:

$$[verb_{Tr}] = \{(x, y)|x, y \in U \wedge xverb_{Tr}y\}$$

The category type for a transitive verb of degree 2 is $N \otimes S \otimes N$. Hence, let $(e^{(i)}, e^{(j)})$ be the vectors denoting the pair of objects (n_i, n_j) and $e^{(i)} \otimes e^{(j)}$ be their semantic representation . As for intransitive verbs, the tensor S is composed of 1 excepted where the pair satisfies $n_i \; v_T \; n_j$. The representation is as follows:

$$v_{v_T} = \sum_{((n_i, n_j), S_{i,j,:}) \in U'} e^{(i)} \otimes S_{i,j,:} \otimes e^{(j)}$$

where U' is the set of relation defined by the context of the verb.

Algebraic Semantics: Simple sentences

An accurate distributional representation of the general sentence "subject verb object"can be computed with the formula:

$$\overrightarrow{sub\ verb\ obj} = (\overrightarrow{sub} \otimes \overrightarrow{obj}) \odot \overrightarrow{verb}$$

Similarly, the general sentence "subject verb"is computed thanks to the formula:

$$\overrightarrow{sub\ verb} = \overrightarrow{sub} \odot \overrightarrow{verb}.$$

E.Grefenstette and M.Sadrzadeh extend this computation to a more advanced sentence composed of an adjective and an adverb :

$$\overrightarrow{adj\ sub\ verb\ obj\ adv} = (\overrightarrow{adv} \odot \overrightarrow{verb}) \odot ((\overrightarrow{adj} \odot \overrightarrow{sub}) \otimes \overrightarrow{obj})$$

Background:

What do we have :

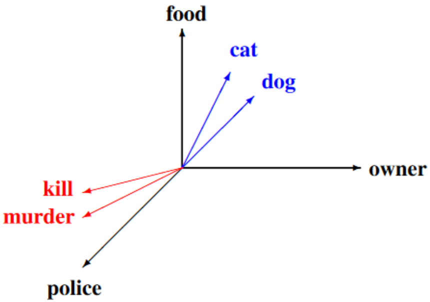
- words
- relations between words
- mathematical definitions for words
- anything about the grammar representation?

Background:

But are these elements connected within each other?

Background: Words - vectors

Words, due to dictionaries have some definitions, however these lasts are made for human. To make it computer based, we need to find something else.
Word2Vec can overtake this issue and can allow words to have a computer meaning.



Background: Pregroup Grammar

Within a sentence, positions of words, suffixes and prefixes indicate to the reader how to read correctly the sequence of words. This is due to the fact our brain links syntactical information to make sens of them.

In , J.Lambek described the structure of a sentence through different types associated to each word. Thanks to these, a sentence can be defined as grammatically correct even if the sens is missing.

$$\begin{array}{cc} \textit{John} & \textit{works} \\ n & n \setminus s \end{array}$$

$$\begin{array}{ccccc} \textit{The} & \textit{dog} & \textit{chases} & \textit{the} & \textit{cat} \\ n \setminus n & n & n \setminus s \setminus n & n \setminus n & n \end{array}$$

Background: Pregroup Grammar

For J. Lambek, grammar can be seen as a mathematical structure. This lead to a possible and effective way to reduce sentence, i.e. to know if they are well formed.

$$G = \left(P, \leq, \cdot, 1, (-)^l, (-)^r\right)$$

$(P, \leq, 1, \cdot)$ is a partially ordered monoid.

A partially ordered monoid is a partially ordered set $(P, \leq)$ with an associative binary operation$\cdot$such as$\cdot$:$P \times P \rightarrow P$ and an unit element 1 which remains the multiplication order preserving.

$$p \cdot p^r \leq 1 \leq p^r \cdot p$$

$$p^l \cdot p \leq 1 \leq p \cdot p^l$$

Background: Pregroup Grammar

Functional word	type
noun	n
transitive verb	$n^r s n^l$
intransitive verb	$n^r s$
adjective	$n n^l$
subject relative pronoun	$n^r n s^l n$
object relative pronoun	$n^r n n^{ll} s^l$

Example: He likes Mary with the type $\pi\left(\pi^r s o^l\right) o$

$$\begin{aligned}
\pi_3\left(\pi^r s o^l\right) o &\leq \left(\pi_3 \pi^r\right) s \left(o^l o\right) \\
&\leq \left(\pi \pi^r\right) s \left(o^l o\right) \\
&\leq 1 s 1 1 \\
&\leq s
\end{aligned} \tag{1}$$

Background: Linking definitions and grammar

As seen previously, we have a way to define words from the computer side and way to obtain the grammar structure of sentences thanks to the pregroup grammar. The question is now:

How to link these two?

Background: Compact closed categories

As the distributional aspect and the syntactic one have been described, thefocus is to explain the structure that is going to link these two : the compact close category.

Category: A category C is composed of two elements : morphisms andobjects. A morphism is a map between two mathematical structure which preserved properties whereas objects may be different things. To define a mathematical category, the following elements have to be implemented:

- a collection of objects $ob(C)$
- a collection of morphisms $morph(C)$
- an identity element for each object such as $id_A : A \rightarrow A$
- a composition rule:
  - associative: $f : A \rightarrow B, g : B \rightarrow C, h : C \rightarrow D, (h \circ g) \circ f = h \circ (g \circ f)$
  - identities are neutral elements: $f : A \rightarrow B, id_b \circ f = f = f \circ id_A$

Background: Compact closed categories

Functor: A functor is a mapping between two categories which conserves identities and compositions. Let P and Q two categories, a functor $F : P \to Q$:

$$\begin{aligned} F &: Ob(P) \to \mathrm{Ob}(Q) \\ F\,(id_A) &= Id_A \\ F(u \circ v) &= F(u) \circ F(v), (u, v \in P) \end{aligned}$$

Monoidal category: A monoid is a mathematical structure which is associative $x(yz) = (xy)z$ and where an identity element 1 exists. In category theory, it is an object (M, μ, η) in a monoidal category $(\mathrm{C}, \otimes, I)$, which has two morphisms such as:

$$\begin{aligned} \mu &: M \otimes M \to M \\ \eta &: I \to M \end{aligned} \tag{2}$$

Background: Compact closed categories

Compact closed category: A compact closed category C is a category in which for all elements E, there exists an object E^r and an object E^l called right and left adjoint.

$$\begin{array}{cc} \eta^l : I \to A \otimes A^l & \eta^r : I \to A^r \otimes A \\ \epsilon^l : A^l \otimes A \to I & \epsilon^r : A \otimes A^r \to I \end{array}$$

The first pair is called generation and the second pair cancellation.The previous mapping has to satisfy the Yanking equations which ensures that diagrams commute.

$$\begin{array}{cc} (1_A \otimes \epsilon^l_A) \circ (\eta^l_A \otimes 1_A) = 1_A & (\epsilon^r_A \otimes 1_A) \circ (1_A \otimes \eta^r_A) = 1_A \\ (1_{A^r} \otimes \epsilon^r_A) \circ (\eta^r_A \otimes 1_{A^r}) = 1_{A^r} & (\epsilon^r_A \otimes 1_{A^l}) \circ (1_{A^l} \otimes \eta^r_A) = 1_{A^l} \end{array}$$

Background: Frobenius algebra

An extension of the category theoretic framework is made through some elements of the Frobenius algebra. Developed by G.Frobenius a frobenius object $(O, \mu, \zeta, \Delta, \iota)$ in a monoid $(C, \otimes, 1)$ is an object O in C where:

$$\mu : A \otimes A \rightarrow A$$
$$\zeta : 1 \rightarrow A$$
$$\Delta : A \rightarrow A \otimes A$$
$$\iota : A \rightarrow 1$$

Background: Frobenius algebra

It commutes:

$$(\mu \otimes 1_A) \circ (1_A \otimes \Delta) = \Delta \circ \mu = (1_A \otimes \mu) \circ (\Delta \otimes 1_A)$$

Let $V \in Ob(FHilb)$ be a finite dimensional Hilbert space with basis $\vec{v_i}_i$, a Frobenius algebra can be defined as:

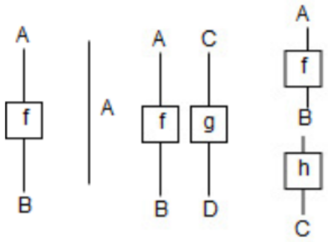
$$\begin{array}{ll} \Delta : V \to V \otimes V & \mu : V \otimes V \to V \\ \overrightarrow{v_i} \longmapsto \overrightarrow{v_i} \otimes \overrightarrow{v_i} & \overrightarrow{v_i} \otimes \overrightarrow{v_j} \longmapsto \delta_{ij} \overrightarrow{v_i} \end{array}$$

$$\begin{array}{ll} \iota : V \to 1 & \zeta : 1 \to V \\ \overrightarrow{v_i} \longmapsto 1 & 1 \longmapsto \sum_i \overrightarrow{v_i} \end{array}$$

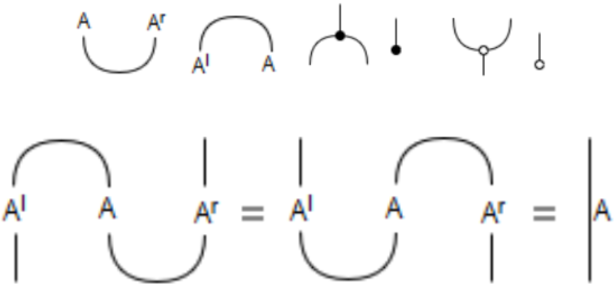
The co-monoid co-multiplication might be referred to by copying and the monoid multiplication by uncopying. The first one aim is to duplicate in-formations contained in one vector whereas the second one aim is to merge information in one vector

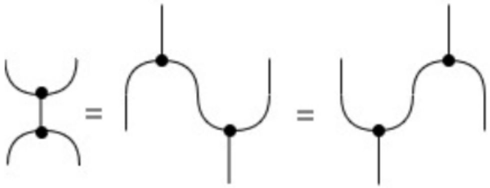
Background: Graphical calculus

As the framework is based on monoid category, it is possible to compute operations through graphic calculus. Objects are represented as labeled wires and morphisms as boxes with input and output.



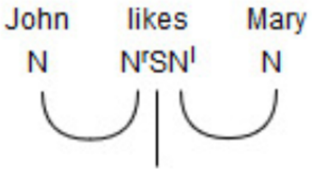
Background: Graphical calculus





Background: Unification

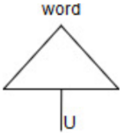
Let $G \in Preg$ be a pregroup which encodes the grammar structure of language:
$G = \{P, \leq, \cdot 1, (-)^l, (-)^r\}$ a compact closed category. Objects of P are grammatical types s, n and morphisms are the grammatical reductions.
The sentence "John likes Mary" has for grammatical type n for the nouns and $n\Delta s\Delta n$ for the verb as this one is transitive. As the pregroup is in $Preg$, a compact closed category, it is reliable to display connections through a graphic.



One the same way as for pregroup, let define a space in which the information encoded is the distributional meaning. Let *FHilb* be a compact close category. Objects are finite vectors and morphisms are linear maps. Let define also a left and right adjoint, $(-)^l$ and $(-)^r$ and η and ϵ two structures preserving maps:

$$\epsilon^l = \epsilon^r : W \otimes W \to \Re : \sum_{ij} c_{ij} \left(\overrightarrow{w_i} \otimes \overrightarrow{w_j}\right) \longmapsto \sum_{ij} c_{ij} \left\langle \overrightarrow{w_i} | \overrightarrow{w_j} \right\rangle$$
$$\eta^l = \eta^r : \Re \to W \otimes W :: 1 \longmapsto \sum_i \overrightarrow{w_i} \otimes \overrightarrow{w_i}$$

Considering the morphism state which transforms an element of $\Re$ to an element of *FHilb* and considering words from a sentence as vectors, it is possible to write that $\overline{word} \in W$. As W is a compact closed category, a graphical representation is given:



Background: Quantizing the grammar

Now that both interpretations of the language are described through the same kind of structure, it is possible to link them together thanks to a strong monoidal functor. This functor links the grammar part *Preg* to the meaning part *FHilb*:

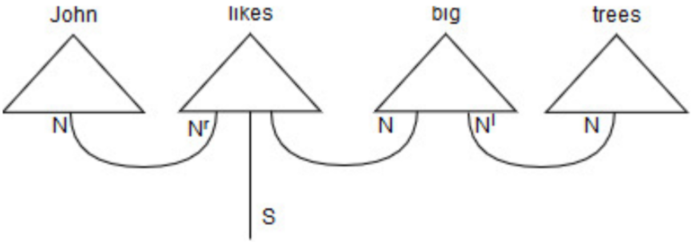
$$F : Ob(\mathrm{Preg}) \to Ob(FHilb)$$

$$F(n \cdot s \cdot n) = F(n) \otimes F(s) \otimes F(n) = N \otimes S \otimes N$$

$$|w_1 \dots w_n > := F(\alpha)\left(|w_1 > \otimes \dots \otimes| \, w_n >\right)$$

Background: Example

Let consider the sentence "John likes big trees" where types are,in the order n, $n^r s n^l$, $n n^l$ and n.



The type reduction α is given by :$n \cdot n^r s n^l \cdot n n^l \cdot n \xrightarrow{\epsilon_n^r \otimes 1_s \otimes 1_n \otimes \epsilon_n^l} s n^l \cdot n \xrightarrow{1_s \otimes \epsilon_n^l} s$

$\alpha = \left(1_s \otimes \epsilon_n^l\right) \circ \left(\epsilon_n^r \otimes 1_s \otimes 1_n \otimes \epsilon_n^l\right)$

Background: Example

Words in the sentence have the following representation according to their type :

$$\overrightarrow{John} = \sum_i a_i^{john} \overrightarrow{n_i} \qquad \overrightarrow{tree} = \sum_n a_n^{tree} \overrightarrow{n_n}$$

$$\overline{big} = \sum_j a_j^{big} \overrightarrow{n_j} \otimes \overrightarrow{n_j} \qquad \overline{likes} = \sum_{klm} a_{klm}^{like} \overrightarrow{n_k} \otimes s_l \otimes \overrightarrow{n_m}$$

Background: Example

Words in the sentence have the following representation according to their type :

$$(1_s \otimes \epsilon_n^l) \circ (\epsilon_n^r \otimes 1_s \otimes 1_n \otimes \epsilon_n^l)(\overrightarrow{John} \otimes \overrightarrow{likes} \otimes \overrightarrow{big} \otimes \overrightarrow{trees}) =$$

$$\alpha(\sum_i a_i^{john}\overrightarrow{n_i} \sum_{klm} a_{klm}^{like}\overrightarrow{n_k} \otimes s_l \otimes \overrightarrow{n_m} \sum_j a_j^{big}\overrightarrow{n_j} \otimes \overrightarrow{n_j} \sum_n a_n^{tree}\overrightarrow{n_n})$$

$$= (1_s \otimes \epsilon_n^l)(\sum_{ijklmn} a_{klm}^{like} a_i^{john} a_n^{tree} a_j^{big} \langle \overrightarrow{n_i} \mid \overrightarrow{n_k} \rangle \otimes s_l \otimes \overrightarrow{n_m} \otimes \overrightarrow{n_j} \langle \overrightarrow{n_j} \mid \overrightarrow{n_n} \rangle)$$

$$= (1_s \otimes \epsilon_n^l)(\sum_{ijlm} a_{ilm}^{like} a_i^{john} a_j^{tree} a_j^{big} s_l \otimes \overrightarrow{n_m} \otimes \overrightarrow{n_j})$$

$$= \sum_{ijlm} a_{ilm}^{like} a_i^{john} a_j^{tree} a_j^{big} \langle \overrightarrow{n_m} \mid \overrightarrow{n_j} \rangle \otimes s_l$$

$$= \sum_{ijl} a_{ilj}^{like} a_i^{john} a_j^{tree} a_j^{big} s_l$$

Background: Modeling concepts

Example: Let consider the general sentence "subject verb object":

$$\overrightarrow{subverbobj} = \sum_{ijk} \left\langle \overrightarrow{sub} | \vec{n}_i \right\rangle c_{ijk} \vec{s_k} \left\langle \overrightarrow{obj} | \vec{n}_k \right\rangle$$

$$= \sum_{ik} \langle \text{sub} \, | \vec{n}_i \rangle \, c_{ijk} \left(\vec{n_i} \otimes \vec{n}_k \right) \left\langle \overrightarrow{obj} | \vec{n}_k \right\rangle$$

$$= \sum_{ik} c_i^{sub} c_k^{obj} c_{ik} \left(\vec{n_i} \otimes \vec{n_k} \right)$$

$$\overline{subverbobj} = (\overrightarrow{sub} \otimes \overrightarrow{obj}) \odot \overrightarrow{verb} \tag{3}$$

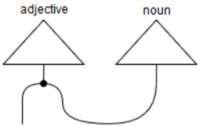
Background: Modeling concepts

Adjective: In English language, adjectives are linked to nouns in order to modify oadd properties to their statement. According to it type nn^l, it is usually followed by a nominal group and can not be used alone. From this idea, the adjective can be seen as an intransitive verb, the noun taking the place of the subject

$$\overrightarrow{adjective} = \sum_i \overrightarrow{noun_i}$$

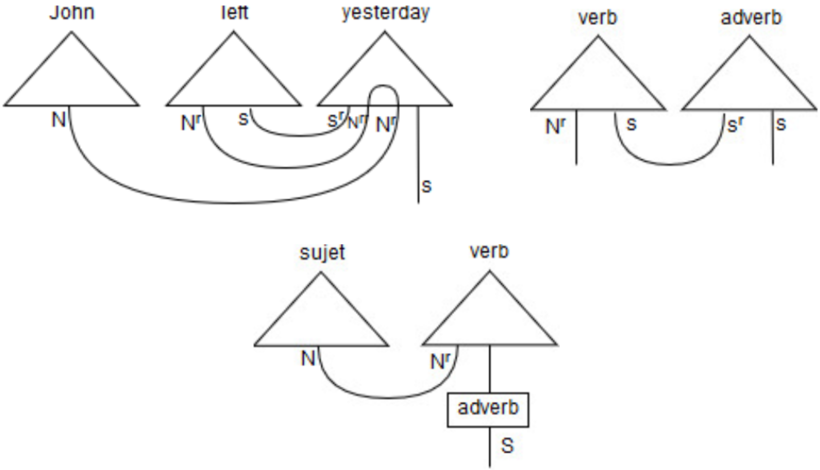
The iteration is made over all contexts where the adjective is contained.As output, there is a matrix containing all information about the adjective computed from a distributional point of view.

$$\overrightarrow{adjective\ noun} = \mu(\overrightarrow{adjective} \otimes \overrightarrow{noun}) = \overrightarrow{adjective} \odot \overrightarrow{noun}$$



Background: Modeling concepts

Adverb:The adverbs can be seen as adjectives or intransitive verbs in a way they are linked with an object. This one can be a noun, an adjective, a verb or a preposition.



Background: Quantum part

In quantum physics, it is rather often to deal with "unknown states"To be able to still perform calculus, some statistical methods are used leading to consider pure and mixed states.
An analogy can be made with words having multiple definitions corresponding to some kind of mixed states and words with one definition corresponding to pure state.

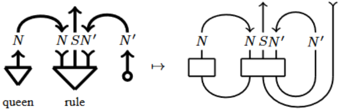
Let consider a distributional model given in the form of a Hilbert space. In this one, w_i corresponds to a words with a p_i probability. The distributional meaning is defined as:

$$\rho(w_t) = \sum p_i \mid w_t^i >< w_t^i \mid$$

Let consider the following example:

$$|rule\rangle = |band\rangle \otimes |true\rangle \otimes |\epsilon\rangle + |chess\rangle \otimes |false\rangle \otimes |\epsilon\rangle$$
$$+ |elisabeth\rangle \otimes |true\rangle \otimes |\text{ england }\rangle$$

$$\rho(queen) = |elisabeth >< elisabeth| + |band >< band + |chess >< chess|$$



DisCoCirc: Introduction

The DisCoCat model is used to understand and compute in an accurate way words of sentences, it doesn't do anything with sentences as elements of a text. However, it seems that some structures developed within this model can be used to understand texts.
It is for this purpose that the DisCoCirc model was introduced. Still a theoretical model, basic knowledge about its way of working were described in a recent paper from B.Coecke.

DisCoCirc: Introduction

In this model, three entities are used:

- word meanings: types
- sentence meanings: I/O proceesses
- text meanings: circuits

If the two first meanings were already described in DisCoCat model, they have been adapted to fit with the aim of DisCoCirc.
The mathematical background mandatory for DisCoCirc remains partially the same, also, only new elements are going to be described.

DisCoCirc: Sentence as I/O processes

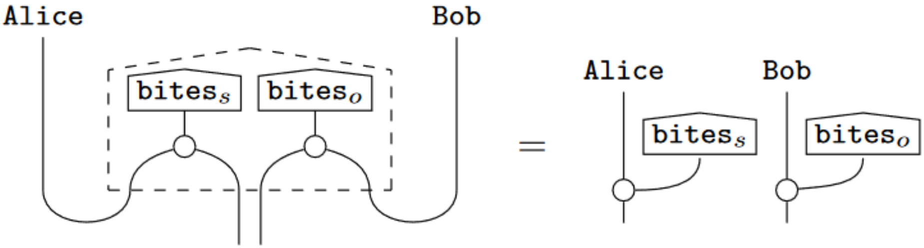
If for DisCoCat model, sentences were considered as fact, leading to the possible vectorization, it is no longer the case with DisCoCirc. Nouns, which were for all of them considered as static may be now dynamic.

In the sentence "Bob is a human the consequence is that Bob belongs to the human category. Hence, other attributes can be add due to this information.

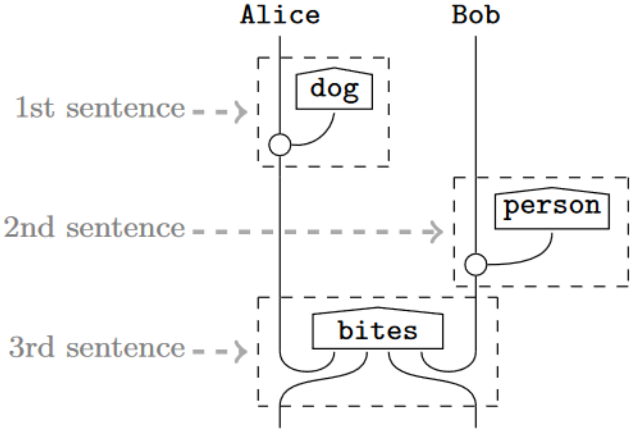
If a following sentence is "Bob is blue the new information is added to the Bob's information, things that was possible before.

Therefore, dynamic nouns can be seen as some kind of ontological objects with possible information liked.

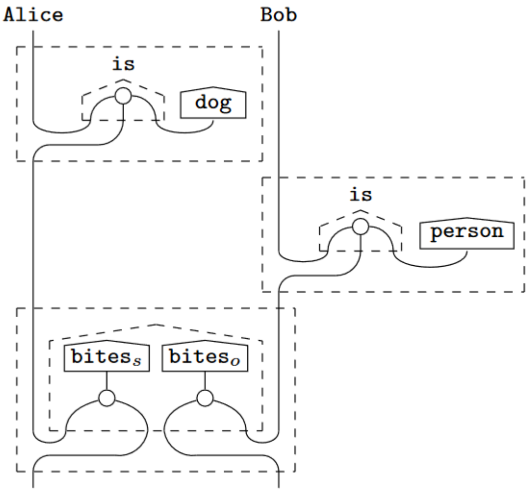
DisCoCirc: Sentence as I/O processes



DisCoCirc: Sentence as I/O processes



DisCoCirc: Sentence as I/O processes

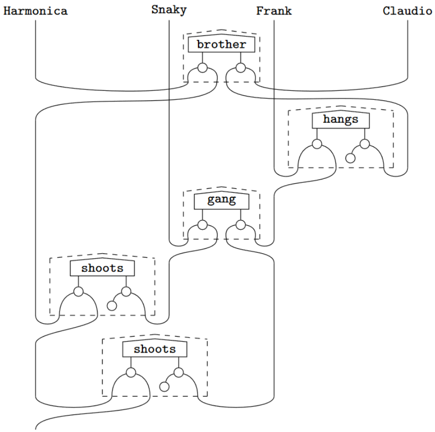


DisCoCirc: Example

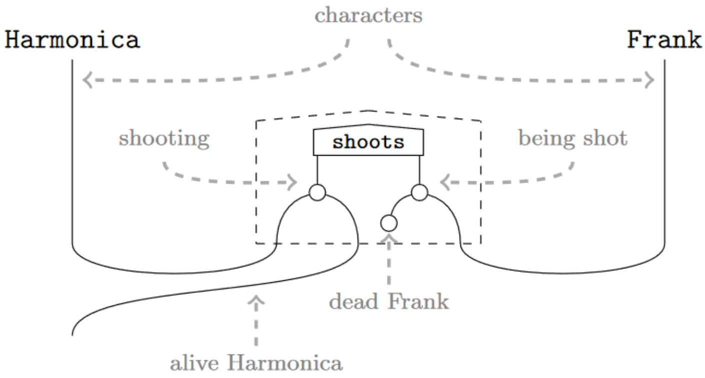
This example comes from the paper presenting the DisCoCirc. It shows how dynamic nouns are used and modified.

Harmonica (is the brother of) Claudio.
Frank hangs Claudio.
Snaky (is in the gang of) Frank.
Harmonica shoots Snaky.
Harmonica shoots Frank.

DisCoCirc: Sentence as I/O processes



DisCoCirc: Sentence as I/O processes



Questions?