From syntax to semantics: taking advantages of 5P

Luísa Coheur(L^{2}F(INESC-ID)/GRIL), Nuno Mamede(L^{2}F(INESC-ID)/IST)

Spoken Language Systems Lab, R. Alves Redol, 9, 1000-029 Lisbon, Portugal
{luisa.coheur,nuno.mamede}@inesc-id.pt

Abstract. This paper states how to profit from 5P's arrows formalism to go from a text with its surface structure to a chosen semantic representation. We emphasize on this formalism flexibility both to connect models and to define sets of conditions (over that connections) that will trigger semantic functions.

1 Introduction

5P ([4], [3]) is a paradigm which goals are much the same than the ones of other linguistics approaches for the study and processing of Natural Language (NL). However, these same goals are reached in particular ways. This paper states how to take advantage of 5P's arrows formalism to go from syntax to semantics. We start by presenting the 5P ingredients. Then, we explain how we use the arrows formalism to connect models and to define sets of conditions (over that connections) to which semantic functions are associated. Finally, we present an example of application having First Order Logic (FOL) and Discourse Representation Structures (DRS) as representation targets.

We would like to make clear that we make our own interpretation/use of this paradigm, which does not necessarily reflect that of 5P.

2 Some 5P ingredients

Nuclear phrases are very close to Abney's chunks [1] (see [6] for a detailed explanation of how to identify nuclear phrases and a comparison with chunks and other similar structures). In the linguistic descriptions made within 5P, nuclear phrases play an important role, as they have (among others) the useful characteristic of not having internal recursivity if coordination is excluded [5].

5P offers a formalism allowing to describe sequences verifying a set of properties ([3]). These sequences – nuclear or non nuclear phrases, etc. – are called models. In order to connect models, 5P offers the arrows properties, that we following present within our own interpretation/extention (see [3] for details over the ori/-ginal formalism).

A basic arrow property expresses a link between two syntactic elements (the source and the target) within a model (the current model). As an example, if we have a nominal nuclear phrase (nn), and an adjectival nuclear phrase (an) inside a nominal phrase (n) and, moreover, an relates with nn, we say that an arrows nn in n and we write an $\to_{n}$ nn, being: a) an the source; b)nn the target; c)n the current model. In order to add more expressiveness to this formalism, it can be extended with restrictions over the:

- current model (ex: an $\to_{n}$ nn [restrictions over n])
- source or target[1](ex: an(restrictions over an)$\to_{n}$nn(restrictions over nn))
- upper model (ex: an $\to_{n}$ nn ↑[restrictions over the model above n])

Each restriction set is divided in 2 sets, separated by a slash (/): a set of restrictions to satisfy and a set of restrictions that can not be verified.
These restrictions can be of one of the following types (adj is for adjective, pn for prepositional nuclear phrase, det for determiner and vn to verbal nuclear phrase):

- existence: x (ex: an(adj/) $\to_{n}$ nn [/pn] (adj exists in an. pn doesn't in n));
- nuclear: °x (ex: det $\to_{nn}$ adj [°adj/] (adj exists in nn and is the nucleus));
- linear: x $<^*$ y if x precedes y (x < y, if x immediately precedes y)(ex: an $\to_{n}$ nn [nn $<^*$ an/] (both nn and an exists in n. nn precedes an));
- arrowing: x → y (ex: an $\to_{n}$ nn [pn→nn/](pn, nn exist in n. pn arrows nn)).

3 Connecting models

Consider the classic example *Saw the man in the house with a telescope*. As Allen says in [2], there are five interpretations arising from the different attachments. By retrieving nuclear phrases, we obtain:

(saw)$_{vn}$ *(the man)*$_{nn}$ *(in the house)*$_{pn}$ *(with a telescope)*$_{pn}$

By using the following arrows properties[2]:

nn $\to_{s}$ vn, pn $\to_{s}$ vn | nn, pn' $\to_{s}$ pn [pn $<_{s}$ pn']

the 5 hypotheses are obtained, without having to duplicate the basic syntactic structures (in all nn $\to_{s}$ vn):

H_1:pn $\to_{s}$ vn, pn' $\to_{s}$ vn, H_2:pn $\to_{s}$ vn, pn' → pn, H_3:pn $\to_{s}$ nn, pn' $\to_{s}$ vn,
H_4:pn $\to_{s}$ nn, pn' $\to_{s}$ nn, H_5:pn $\to_{s}$ nn, pn' $\to_{s}$ pn

However, with these arrows, a sixth interpretation, corresponding to a false ambiguity[3], becomes possible: pn $\to_{s}$ vn, pn $\to_{s}$ nn. By adding restrictions over the arrows, this solution would be avoided. Alternatively, accepting that no arrows crossing is allowed, this sixth hypothesis is no longer possible.

4 From Syntax to Semantics

Associating semantic functions to syntactic rules is a classic procedure in NL processing. It was started by Montague, being [8] a good illustration of this

---

[1] If they are models, and not categories.

[2] The vertical slash denotes "or", and we use pn' to distinguish the pn on the left (pn) from the one on the right (pn').

[3] Someone (in the house) saw a man who had a telescope.

method. Nowadays, it is still used in most prominent theories. However, and as pointed in [4], there are no syntactic rules in 5P, in the sense of conventional grammar rules, *i.e.*, representing part of a tree structure. From this, we can say that 5P does not allow the association of semantic rules with syntactic rules, for the simple reason that it has no syntactic rules. So, the question is, over which structures will semantic functions operate? The answer is very simple: over the arrows properties[4]. As an example, if a arrows b in c defines a syntactic context, a semantic function f can be associated with it[5], and we note it by:

$\mathsf{a} \rightarrow_{\mathsf{c}} \mathsf{b}$: $f_{\mathsf{c}}(\mathsf{b}, \mathsf{a})$

Notice that a syntactic context can be defined over a set of arrows:

$\mathsf{a}_1, \ldots, \mathsf{a}_n \rightarrow_{\mathsf{c}} \mathsf{b}$: $f_{\mathsf{c}}(\mathsf{b}, \mathsf{a}_1, \ldots, \mathsf{a}_n)$

Due to the extended arrows formalism, the conditions that trigger semantic functions can be very precise. As an example in Portuguese, consider the an *algum*. *algum* means *some*, but if this an appears after the nn, it means *none*[6]. We express this with:

$algum \rightarrow_{\mathsf{an}} algum[\mathsf{nn}<\mathsf{an}/]$: $f_{\mathsf{an}}(algum)$[7] and $algum \rightarrow_{\mathsf{an}} algum$: $g_{\mathsf{an}}(algum)$

The next example shows how this formalism allows dispensing with labels. Consider the sentence $(Jones)_{\mathsf{nn}}$ $(owns)_{\mathsf{vn}}$ $(Ulysses)_{\mathsf{nn}}$ from [7]. If we decide that our surface structure only has nuclear phrases, we obtain the two nn at the same level. Nevertheless, the first nn is the subject, and the second the object[8], and we want to be able to distinguish them, as this will have obvious influence in the semantic results. So, if we use the following arrows, we are able to associate with each one the appropriate semantic function:

$\mathsf{nn} \rightarrow_{\mathsf{s}} \mathsf{vn}[\mathsf{nn}<\mathsf{vn}]$: $g_{\mathsf{s}}(\mathsf{vn},\mathsf{nn})$(subject), and $\mathsf{nn} \rightarrow_{\mathsf{s}} \mathsf{vn}[\mathsf{vn}<\mathsf{nn}]$:$h_{\mathsf{s}}(\mathsf{vn},\mathsf{nn})$(object)

That is, without having to add extra labels, the subject and the object are identified by their own syntactic properties.

5 Semantic functions

We now show how to go from a text in which the nuclear phrases were identified, to a semantic representation (either in FOL or DRS).

Continuing with the example from the previous section, consider that the semantic associated with the nn *Jones* is `Jones(z)` and with the vn *owns* is `owns(x, y)`. These will help us to illustrate the following functions:

- θ(A) returns the semantics associated with A (ex: $\theta(Jones)$ = `Jones(z)`);
- $\textsc{abs}_i$(A) returns the ith variable associated with θ(A)(ex: $\textsc{abs}_1(\mathsf{vn}) = x$, $\textsc{abs}_2(\mathsf{vn}) = y$ and $\textsc{abs}_1(\mathsf{nn}) = z$);
- [X/Y]S replaces Y by X in S (ex: $[\textsc{abs}_1(\mathsf{vn})/\textsc{abs}_1(\mathsf{nn})]\theta(\mathsf{nn}) = [x/z]$`Jones(z)` = `Jones(x)`);

---

[4] In 5P, they say that a set of arrows originates a graph, which is the input to semantic functions [3].

[5] We say that the syntactic condition $\mathsf{a} \rightarrow_{\mathsf{c}} \mathsf{b}$ triggers the semantic function $f_{\mathsf{c}}(\mathsf{b}, \mathsf{a})$.

[6] $(algum)_{\mathsf{an}}(rapaz)_{\mathsf{nn}}$ means *some boy*, and $(rapaz)_{\mathsf{nn}}$ $(algum)_{\mathsf{an}}$ means *no boy*.

[7] If $\mathsf{a} \rightarrow_{\mathsf{b}} \mathsf{a}$, instead of $f_{\mathsf{b}}(\mathsf{a}, \mathsf{a})$ we note the associated semantic function as $f_{\mathsf{b}}(\mathsf{a})$.

[8] Note that we are not considering the passive voice.

- $\textsc{add}(\{x_1, ..., x_m\}, j, S)$ adds $\theta(x_1), ..., \theta(x_m)$ to S (which can be a formula, a DRS, etc.), in position j (undetermined (_) or with a precise meaning) (ex: being given the drs = ({x, y}, {})[9], then $\textsc{add}(\{\theta(\mathsf{vn})\}, 2, drs)$ = ({x, y}, {`owns(x, y)`})).

Consider the following three syntactic contexts/semantic functions pairs:

$\mathsf{nn}\rightarrow_{\mathsf{S}}\mathsf{vn}$ [$\mathsf{nn}<\mathsf{vn}$]: $g_{\mathsf{S}}(\mathsf{vn}, \mathsf{nn})$, $\mathsf{nn}\rightarrow_{\mathsf{S}}\mathsf{vn}$ [$\mathsf{vn}<\mathsf{nn}$]: $h_{\mathsf{S}}(\mathsf{vn}, \mathsf{nn})$, $\mathsf{vn}\rightarrow_{\mathsf{S}}\mathsf{vn}$: $f_{\mathsf{S}}(\mathsf{vn})$.

Take FOL as the representation language. If F = ∅, by defining the functions as:

$(1) f_{\mathsf{S}}(\mathsf{vn}) = \textsc{add}(\{\theta(\mathsf{vn})\}, \_, F)$

$(2) g_{\mathsf{S}}(\mathsf{vn}, \mathsf{nn}) = \textsc{add}(\{[\textsc{abs}_1(\mathsf{vn})/\textsc{abs}_1(\mathsf{nn})]\theta(\mathsf{nn})\}, \_, F)$

$(3) h_{\mathsf{S}}(\mathsf{vn}, \mathsf{nn}) = \textsc{add}([\textsc{abs}_2(\mathsf{vn})/\textsc{abs}_1(\mathsf{nn})]\theta(\mathsf{nn}), \_, F)$

we obtain: F $=_{(1)}$ {`owns(x, y)`} $=_{(2)}$ {`Jones(x)`, `owns(x, y)`} $=_{(3)}$ {`Jones(x)`, `owns(x, y)`, `Ulysses(y)`}[10].

Take now DRS's. If drs = ({}, {}), 1 denotes the set of reference markers, 2 the conditions set, and the functions are defined as:

$(1) f_{\mathsf{S}}(\mathsf{vn}) = \textsc{add}(\{\textsc{abs}_1(\mathsf{vn}), \textsc{abs}_2(\mathsf{vn})\}, 1, \textsc{add}(\theta(\mathsf{vn}), 2, drs))$

$(2) g_{\mathsf{S}}(\mathsf{vn}, \mathsf{nn}) = \textsc{add}([\textsc{abs}_1(\mathsf{vn})/\textsc{abs}_1(\mathsf{nn})]\theta(\mathsf{nn}), 2, drs)$

$(3) h_{\mathsf{S}}(\mathsf{vn}, \mathsf{nn}) = \textsc{add}([\textsc{abs}_2(\mathsf{vn})/\textsc{abs}_1(\mathsf{nn})]\theta(\mathsf{nn}), 2, drs)$

we obtain: drs $=_{(1)}$ ({x, y}, {`owns(x, y)`}) $=_{(2)}$ ({x, y}, {`Jones(x)`, `owns(x, y)`})$=_{(3)}$ ({x, y}, {`Jones(x)`, `owns(x, y)`, `Ulysses(y)`})

6 Conclusions and further work

We showed how to profit from 5P arrows formalism, which, enriched with restrictions, is used both to connect models and to precise syntactic triggers to the semantic analysis. As future work, we will continue to explore this formalism's potentialities.

References

1. Steven Abney. *Parsing by Chunks.* Kluwer Academic Publishers, 1991.
2. James Allen. *Natural Language Understanding (second edition).* The Benjamin Cummings Publishing Company, Inc, 1995.
3. G. Bès and C. Hagège. Properties in 5p (soon in the GRIL pages). November, 2001.
4. Gabriel G. Bès. La phrase verbal noyau en français. In *in Recherches sur le français parlé, 15.* Université de Provence, France, 1999.
5. C. Hagège and G. Bès. Delimitação das construções relativas e completivas na análise de superfície de textos. In *PROPOR'99*, pages 93–104, Portugal, 1999.
6. Caroline Hagège. *Analyse Syntatic Automatique du Portugais.* PhD thesis, Université Blaise Pascal, Clermont-Ferrand, France, 2000.
7. H. Kamp and U. Reyle. *From Discourse to logic.* Kluwer Academic Publishers, 93.
8. Richard Montague. The proper treatment of quantification in ordinary english. In R.H. Thomason, editor, *Formal Philosophy. Selected papers of Richard Montague*, pages 247–270. Yale University Press, New Haven, 1974.

---

[9] In the canonical set-theoretical notation.

[10] *i.e.*, `Jones(x)` ∧ `owns(x, y)` ∧ `Ulysses(y)`.