Parsing of raw semantic particles and synthesis of semantic structures from the former

D.Gueorguiev 10/12/2021

The Notion of Semantic Parsing

Semantic Parsing ($\boldsymbol{SP}$) is a process, distinctly different than our understanding of what *Parsing* is in the classical sense of Computer Science. The *Semantic Parsing* uses inference to construct semantic structures and it is closely related to the Inference Process ($\boldsymbol{IP}$) as it will be shown in this section.

Process for transforming a sequence of raw particles into semantic structure

In the beginning we have a sequence of raw (naked) particles containing single property – $text$. We know which raw particle precedes the current one and which one succeeds it. The first part of dressing the raw particle sequence is performing *Semantic Association*.

Semantic Association

The purpose of this process is to associate already existing semantic particles with each raw particle in a set of raw particles. We would like to identify particles with similar signatures and link them through MA-particles and MR-particles using *similarity association* process.
This association process may require splitting and merging raw particles from the set into another set of raw particles. The algorithm by which we transform the original raw particles through splitting and merging into a new set will be discussed in this Section.

Example:
<u>Context:</u>
Apostrophe plus s denotes possession.
I know John who is my friend. John has children.
<u>New thought:</u>
Ivan is John's son.
The we can write in symbolic notation:
$\prec [V_{101}\ A_{101}\ V_{102}\ A_{102}\ V_{103}]\ A_{103}\ V_{104}\ A_{104}\ V_{105} \succ$
$\prec V_1\ A_1\ V_2\ A_2\ V_3\ A_3\ V_4\ A_5\ V_5\ A_6\ [V_6\ A_7\ V_8] \succ$
$\prec V_3\ A_8\ V_9\ A_9\ V_{10} \succ$

Let $V_{apostrophe} = [V_{101}\ A_{101}\ V_{102}\ A_{102}\ V_{103}]$ with
$prop(V_{apostrophe}, `text`, `all\ text\ values`) = list(“Apostrophe\ plus\ s”, “’s”)$

The new thought contains the following naked particles (N-particles):
$\prec N_{11}\ N_{12}\ N_{13}\ N_{14} \succ$

$text(N_{11}) = “Ivan”$
Here we utilized the shortcut $text(N_{11}) = prop(N_{11}, `text`, `text`)$. Similarly:
$text(N_{12}) = ”is”$
$text(N_{13}) = ”John’s”$

$text(N_{14}) = "son"$

The Semantic Association ($\boldsymbol{SA}$) process starts by sending *default-association* particles to attach to each raw (naked) particle on the left and on the right as follows:
$\prec N_{11}\ DA_1\ N_{12}\ DA_2\ N_{13}\ DA_3\ N_{14} \succ$
Then a set of M-particles are created attaching to the *text* property of each naked particle.
Each M-particle which binds to the same *text* property of the naked particle creates slightly different binding pattern based on the textual representation of the naked particle. Each of those M-particles will attempt to attract *similarity-association* particle (SA-particle) bound to already processed V-particles. For the case of N_{13} one of those M-particles will attract SA-particle associated with V_3 and another one of those M-particles will attract SA-particle associated with $V_{apostrophe}$.

Details on the algorithm performing similarity association and dressing of raw particles are discussed in *Supplement-9-5-29 (page1-page10)*.

Partially dressed particle
Raw particle which has been identified by name and its position in the semantic path with respect to the other partially dressed particles.

Property inference in partially dressed particles
Let us assume we have a partially dressed particle N together with other partially dressed particles N_1, N_2, … , N_k which are part of the same semantic path (sequence). We would like to infer as many properties of those as possible.

S – the set of all allowed property values for particle N
$S_1, S_2, \dots, S_k$ are subsets of S
Existing chain C_1: N has the pairs (p_i, v_j) where (i, j) belongs to S_1
Existing chain C_2: N has the pairs (p_i, v_j) where (i, j) belongs to S_2
. . .
Existing chain C_k: N has the pairs (p_i, v_j) where (i, j) belongs to S_k

New chain C_{new}: N has only (p_0, v_0) where p_0 is the *name* property and v_0 is the *name* value.
Goal: find out the set S_{new} containing the indices (i, j) of all pairs (p_i, v_i) which belong to N in the new chain C_{new}.
~~One can obtain S_{new} by constructing Bayesian network from $C_1, C_2, \dots, C_k$ and the sets $S_1, S_2, \dots, S_k$.~~

And here is discussion how the similarity association process works: Let us start with a raw particle N which we would like to associate/match with fully-fledged (adorned with properties) V particles. For the purpose N will attract some number of match-seeking (MA) and match-repelling particles (MR) which will bind to the sole property exposed by N. The

```
                N
                |
MR_1---------text-------MR_l
        /    |       \
       /     |        \
     MA_1  MA_2  ...  MA_k
     /       |          \
   AS_1     AS_2   ...   AS_k
    |        |            |
   MA^1     MA^2   ...   MA^k
    |        |            |
  text_1   text_2  ...  text_k
    |        |            |
   V_1      V_2    ...   V_k
```

MA – match-seeking particle
MR – match-repelling particle – acts as a repellent toward particular association particles. Models constraints imposed on certain V-particles in terms of similarity matching
AS – similarity particle: a special type of link particle (A-particle)
N – naked particle candidate for dressing
$text$ – the text property value of the naked particle N
V_i – V-particle (semantic particle)

Match-seeker particle

Match-seeker particle is denoted with MA aka MA-particle. We do not usually depict this kind of particles in our graph representations.
Attaches to a specific property with specific prop-name and prop-value. The property value is a vector which is a key allowing the match-seeker particle to be attached to this property. The match-seeker particle exposes a pattern serving as an attraction of an association link particle which would recognize the pattern and attach to the match-seeker. Each match-seeker particle has a property `*charge*` (type `*default*`) with a value indicating the strength of the charge.

$$MA(key_pattern, pvalue) - V(pvalue)$$
$$|$$
$$A(key_pattern, key_pattern2) \text{———} MA(key_pattern2, pvalue2) - V_{other}(pvalue2)$$

Match Repelling particle

Repelling particle is denoted with MR aka MR-particle. We do not usually depict this kind of particles in our graph representations.
Attaches to a specific property with prop-value, a vector allowing a repelling particle to be attached to this property. The repelling particle exposes a pattern serving to repel an *association-link* particle which has a property matching the pattern. Each repelling particle has a property `*charge*` (type `*float*`) with a value indicating the strength of the charge.

To each property of object particle V can be attached multiple MA-particles and MR-particles.

Coulomb's law for semantic particles

Let the particle V_{p_1} has "charge" with value q_1 and particle V_{p_2} has property "charge" with value q_2. Then if $sign(q_1) \neq sign(q_2)$ there will be attraction force between the two particles with magnitude F:

$F(V_{p_1}, V_{p_2}) = K \times \frac{|q_1| \cdot |q_2|}{f(r)}$ where $K > 0$ is some proportionality constant and $f(r)$ is some monotonously increasing function of the semantic distance $r(V_{p_1}, V_{p_2})$ between the two particles. If $sign(q_1) = sign(q_2)$ the force would be repelling and will be with the same magnitude F.

<u>*Questions which we need answers for :*</u>

1) When do we split the raw particle N into a set of new raw particles N1, N2, …, Nl ?
2) When can we establish viable association link between a pair of particles from a set?

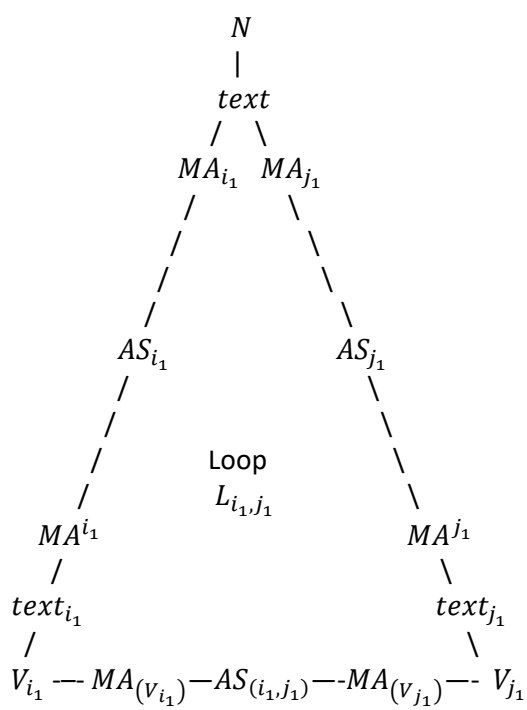
Consider a set of k V-particles semantically close to each other: $V_1, V_2, \dots, V_k$
Let us assume that we have established a viable association between a pair from the set:

$\exists\, i, j \in [1, \dots, k] \quad V_i$ --- MA_i --- AS_i --- MA^j --- V_j

Let us assume we have a set of pairs for which association link can be established:

$(i_1, j_1), (i_2, j_2), \dots, (i_l, j_l),\ l = 1..k^2$

Let us consider the first matched tuple (i_1, j_1)



The loop N-MA_{i_1}-AS_{i_1}-MA^{j_1}-V_{i_1}-$MA_{(V_{i_1})}$-$AS_{(i_1,j_1)}$-$MA_{(V_{j_1})}$-V_{j_1}-MA^{j_1}- AS_{j_1}-MA_{j_1}- N will be denoted with L_{i_1,j_1}.

The binding force F^b of the association loop L_{i_1,j_1} is given with:

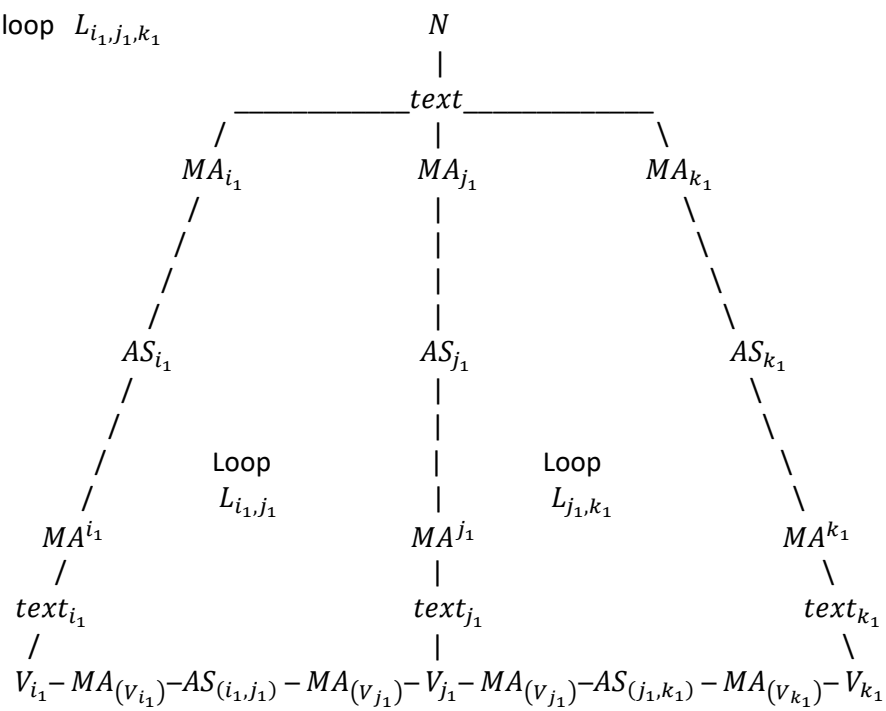
$$F^b(L_{i_1,j_1}) = F(MA_{i_1}, AS_{i_1}) + F(MA^{i_1}, AS_{i_1}) + F\left(MA_{(V_{i_1})}, AS_{(i_1,j_1)}\right) + F\left(MA_{(V_{j_1})}, AS_{(i_1,j_1)}\right) + F(MA_{j_1}, AS_{j_1}) + F(MA^{j_1}, AS_{j_1})$$

Here the attraction force $F(MA_{i_1}, AS_{i_1})$ between the match seeing particle MA and the similarity particle AS is given with the Coulomb's law for semantic particles.

To each pair (i_1, j_1), (i_2, j_2), … , (i_l, j_l) for which an association loop can be established we can compute the binding force $F^b(L_{i_1,j_1})$, $F^b(L_{i_2,j_2})$, …, $F^b(L_{i_l,j_l})$

Note that we are not going to consider higher order loops like the one below. It will become clear later in the discussion on the recursive application of this procedure that those higher order loops are redundant.

Higher order loop L_{i_1,j_1,k_1}



Notice that $L_{i_1,j_1,k_1} = L_{i_1,j_1} \cup L_{j_1,k_1}$
Similarly, we can define higher order loops by merging lower order ones:
$L_{i_1,j_1,k_1,\dots,q_1} = L_{i_1,j_1} \cup L_{j_1,k_1} \cup \dots \cup L_{p_1,q_1}$

Besides loops we can have association chains hanging from N. The binding force for the association chain C_{i_l} of N is expressed as:
$F^b(C_{i_1}) = F(MA_{i_1}, AS_{i_1}) + F(MA^{i_1}, AS_{i_1})$

$$
\begin{array}{c}
N \\
| \\
text \\
| \\
M_{i_k} \\
| \\
AS_{i_k} \\
| \\
M^{i_k} \\
| \\
text \\
| \\
V_{i_k}
\end{array}
$$

On the Figure on the left: association chain C_{i_l} for raw particle N

Let us denote by $\mathfrak{F}^b$ the set of the binding forces for possible association loops and chains:
$\mathfrak{F}^b = \{F^b(L_{i_1,j_1}), F^b(L_{i_2,j_2}), \dots, F^b(L_{i_l,j_l}), F^b(C_{i_1}), F^b(C_{i_2}), \dots, F^b(C_{i_m})\}$

Let $F^b_{max} = max(\mathfrak{F}^b)$. If F^b_{max} is the binding force of an association loop then the raw particle N will be split into two sub-particles N_1 and N_2.