Natural Language Processing in Prolog

Integrazione del programma *Talk* con *WordNet*

Michele Di Capua
michele.dicapua@studio.unibo.it

A.A. 2021/2022

Attività progettuale per Fondamenti di Intelligenza Artificiale – Prof.ssa P. Mello

IL CONTESTO

NLP = analisi e rappresentazione automatica del linguaggio naturale tramite calcolatore

APPLICAZIONI

- INFORMATION RETRIEVAL
- INFORMATION EXTRACTION
- QUESTION-ANSWERING
- SUMMARIZATION
- MACHINE TRANSLATION
- DIALOGUE SYSTEMS

MODELLO A LIVELLI

- LIVELLO FONOLOGICO
- LIVELLO MORFOLOGICO
- LIVELLO LESSICALE
- LIVELLO SINTATTICO
- LIVELLO SEMANTICO

APPROCCI PRINCIPALI

- APPROCCIO SIMBOLICO
- APPROCCIO STATISTICO

IL CONTESTO

APPLICAZIONI

- QUESTION-ANSWERING

MODELLO A LIVELLI

- LIVELLO MORFOLOGICO
- LIVELLO LESSICALE
- LIVELLO SINTATTICO

APPROCCI PRINCIPALI

- APPROCCIO SIMBOLICO

Talk

- Prende in input frasi
- affermazioni
- domande
- FOL
- Base di conoscenza
- Risposte

MyTalk

Talk ⇄ WordNet

MODIFICHE A MyTalk – Interazione Talk-WordNet nei verbi

wn_fr.pl

```
fr(synset_ID, f_num, w_num)
```

```
fr(synset_ID, w_num ,f_num)
```

```
fr(200001740,0,2).
fr(200001740,0,8).
fr(200002325,0,2).
fr(200002573,0,2).
fr(200002724,0,2).
```

MyTalk

```
iv( no, [[W, -s]], [[W, -ed]], [[W, -past]],
        [[W, -ed]], [[W, -en]],
        [[W, -past]], [[W, -ing]], FOL, sg):- s(SysID,W_Num,W,v,_,_),
                                              fr(SysID, FR, W_Num),
                                              sen_fol_iv(FR, W, FOL).
```

NotMyTalk

```
iv( no, [[W, -s]], [[W, -ed]], [[W, -past]],
        [[W, -ed]], [[W, -en]],
        [[W, -past]], [[W, -ing]], FOL, sg):- s(SysID,_,W,v,_,_),
                                              fr(SysID, _, FR),
                                              sen_fol_iv(FR, W, FOL).
```

MODIFICHE A MyTalk – L'implicazione logica

```
%%%                                If...Then Sentences

s(S=>S1, GapInfo) -->
                [if],
                np(VP^S, nogap, Num),
                vp(finite, VP, GapInfo, Num),
                [then],
                np(VP1^S1, nogap, Num),
                vp(finite, VP1, GapInfo, Num).
```

Es:

If
students
pay attention
then
professors
teach

da NotMyTalk...

```
Asserted "teach(professor):-pays(student,attention)."
```

MODIFICHE A MyTalk – L'implicazione logica

```
%%%                                If...Then Sentences

s(S=>S1, GapInfo) -->
                [if],
                [something],
                vp(finite, X^S, GapInfo, Num),
                [then],
                [it],
                vp(finite, X^S1, GapInfo, Num).
```

Es:

If
something
is a metal
then
it
conducts electricity

da NotMyTalk...

```
Asserted "conducts(_25354,electricity):-metal(_25354)."
```

https://github.com/ellerre/MyTalk

MyTalk e NotMyTalk a confronto: esempi

NotMyTalk	MyTalk
>> mike is an engineer **Asserted "engineer(mike)."**	>> mike is an engineer Error: "too difficult."
>> every engineer loves maths **Asserted "loves(_21816,math):-engineer(_21816)."**	>> every engineer loves maths Error: "too difficult."
>> terry is an architect **Asserted "architect(terry)."**	>> terry is an architect Error: "too difficult."
>> every architect hates maths **Asserted "hates(_23152,math):-architect(_23152)."**	>> every architect hates maths Error: "too difficult."
>> who hates maths **terry.**	>> who hates maths Error: "too difficult."
>> who loves maths **mike.**	>> who loves maths Error: "too difficult."

NotMyTalk	MyTalk
>> he likes cooking **Asserted "likes(he,cooking)."**	>> he likes cooking Error: "too difficult."
>> pizza is a food **Asserted "food(pizza)."** **>> dogs eat every food** **Asserted "eats(dog,_2956):-food(_2956)."** **>> who eats pizza** **dog.**	>> pizza is a food Asserted "food(pizza)." >> dogs eat every food Error: "too difficult." >> who eats pizza Error: "too difficult."
>> if someone is a human then he is a mortal **Asserted "mortal(_880):-human(_880)."** **>> Socrates is a human** **Asserted "human(Socrates)."** **>> is Socrates a mortal** **yes.**	>> if someone is a human then he is a mortal Error: "too difficult." >> Socrates is a human Asserted "human(Socrates)." >> is Socrates a mortal Error: "too difficult."

https://proofwriter.apps.allenai.org/

NotMyTalk e ProofWriter a confronto: esempi

NotMyTalk

```
>> every metal conducts electricity
Asserted "conducts(_22894,electricity):-
metal(_22894)."

>> if something is an iron then it is a metal
Asserted "metal(_23846):-iron(_23846)."

>> nail is an iron
Asserted "iron(nail)."

>> does nail conduct electricity
yes.
```

ProofWriter

Facts and rules (you can provide your own):

Metals conduct electricity.
Insulators do not conduct electricity.
If something is made of iron then it is metal.
Nails are made of iron.

Is it true?

Nails conduct electricity.

ProofWriter answer: True

Proof (raw): # sent1@int1 # sent3@int2 sent4 ; with int1: Nails conduct electricity. ; int2: Nails are metal.

Proof (details):

sent4: Nails are made of iron.

sent3: If something is made of iron then it is metal.

===> int2: Nails are metal.

---

int2: Nails are metal.

sent1: Metals conduct electricity.

===> int1: Nails conduct electricity.

https://proofwriter.apps.allenai.org/

NotMyTalk e ProofWriter a confronto: esempi

NotMyTalk

```
>> every metal conducts electricity
Asserted "conducts(_832,electricity):-metal(_832)."

>> if something is a iron then it is a metal
Asserted "metal(_1788):-iron(_1788)."

>> nail is a plastic
Asserted "plastic(nail)."

>> every plastic is a metal
Asserted "metal(_2736):-plastic(_2736)."

>> does nail conduct electricity
yes.
```

ProofWriter

Facts and rules (you can provide your own):

Metals conduct electricity.
Insulators do not conduct electricity.
If something is made of iron then it is a metal.
Nails are made of plastic.
Plastic is a metal.

Is it true?

Nails conduct electricity.

ProofWriter answer: True

Proof (raw): # sent1@int1 # sent5@int2 sent4 ; with int1: Nails conduct electricity. ; int2: Nails are metal.

Proof (details):

sent4: Nails are made of plastic.

sent5: Plastic is a metal.

===> int2: Nails are metal.

---

int2: Nails are metal.

sent1: Metals conduct electricity.

===> int1: Nails conduct electricity.

https://proofwriter.apps.allenai.org/

NotMyTalk e ProofWriter a confronto: esempi

NotMyTalk

```
>> Paul is a Sicilian
Asserted "Sicilian(Paul)."

>> every Sicilian is an Italian
Asserted "Italian(_23440):-Sicilian(_23440)."

>> if someone is an Italian then he loves pizza
Asserted "loves(_24076,pizza):-Italian(_24076)."

>> if someone loves pizza then he hates pineapple
Asserted "hates(_25016,pineapple):-
loves(_25016,pizza)."

>> if someone loves pineapple then he eats
pineapple
Asserted "eats(_37238,pineapple):-
loves(_37238,pineapple)."

>> Luke loves pineapple
Asserted "loves(Luke,pineapple)."

>> if Luke eats pineapple then Paul hates Luke
Asserted "hates(Paul,Luke):-eats(Luke,pineapple)."

>> does Paul hate pineapple
yes.

>> does Paul hate Luke
yes.

>> does Luke hate Paul
no.
```

ProofWriter

Facts and rules (you can provide your own):

Paul is a SIcilian.
Every Sicilian is an Italian.
If someone is an Italian then he loves pizza.
If someone loves pizza then he hates pineapple.
If someone loves pineapple then he eats pineapple.
Luke loves pineapple.
If Luke eats pineapple then Paul hates Luke.

Is it true?

Paul hate pineapple

Submit

ProofWriter answer: True
Proof (raw): # sent4@int1 # sent3@int2 # sent2@int3 sent1 ; with int1: Paul hates pineapple. ; int2: Paul likes pizza. ; int3: Paul is an Italian.
Proof (details):
sent1: Paul is a SIcilian.
sent2: Every Sicilian is an Italian.
===> int3: Paul is an Italian.
---
int3: Paul is an Italian.
sent3: If someone is an Italian then he loves pizza.
===> int2: Paul likes pizza.
---
int2: Paul likes pizza.
sent4: If someone loves pizza then he hates pineapple.
===> int1: Paul hates pineapple.

https://proofwriter.apps.allenai.org/

NotMyTalk e ProofWriter a confronto: esempi

NotMyTalk

```
>> Paul is a Sicilian
Asserted "Sicilian(Paul)."

>> every Sicilian is an Italian
Asserted "Italian(_23440):-Sicilian(_23440)."

>> if someone is an Italian then he loves pizza
Asserted "loves(_24076,pizza):-Italian(_24076)."

>> if someone loves pizza then he hates pineapple
Asserted "hates(_25016,pineapple):-
loves(_25016,pizza)."

>> if someone loves pineapple then he eats
pineapple
Asserted "eats(_37238,pineapple):-
loves(_37238,pineapple)."

>> Luke loves pineapple
Asserted "loves(Luke,pineapple)."

>> if Luke eats pineapple then Paul hates Luke
Asserted "hates(Paul,Luke):-eats(Luke,pineapple)."

>> does Paul hate pineapple
yes.

>> does Paul hate Luke
yes.

>> does Luke hate Paul
no.
```

ProofWriter

Facts and rules (you can provide your own):

Paul is a SIcilian.
Every Sicilian is an Italian.
If someone is an Italian then he loves pizza.
If someone loves pizza then he hates pineapple.
If someone loves pineapple then he eats pineapple.
Luke loves pineapple.
If Luke eats pineapple then Paul hates Luke.

Is it true?

Paul hates Luke

Submit

ProofWriter answer: True
Proof (raw): # sent7@int1 # sent5@int2 sent6 ; with int1: Paul hates Luke. ; int2: Luke eats pineapple.
Proof (details):
sent6: Luke loves pineapple.
sent5: If someone loves pineapple then he eats pineapple.
===> int2: Luke eats pineapple.
---
int2: Luke eats pineapple.
sent7: If Luke eats pineapple then Paul hates Luke.
===> int1: Paul hates Luke.

https://proofwriter.apps.allenai.org/

NotMyTalk e ProofWriter a confronto: esempi

NotMyTalk

```
>> Paul is a Sicilian
Asserted "Sicilian(Paul)."

>> every Sicilian is an Italian
Asserted "Italian(_23440):-Sicilian(_23440)."

>> if someone is an Italian then he loves pizza
Asserted "loves(_24076,pizza):-Italian(_24076)."

>> if someone loves pizza then he hates pineapple
Asserted "hates(_25016,pineapple):-
loves(_25016,pizza)."

>> if someone loves pineapple then he eats
pineapple
Asserted "eats(_37238,pineapple):-
loves(_37238,pineapple)."

>> Luke loves pineapple
Asserted "loves(Luke,pineapple)."

>> if Luke eats pineapple then Paul hates Luke
Asserted "hates(Paul,Luke):-eats(Luke,pineapple)."

>> does Paul hate pineapple
yes.

>> does Paul hate Luke
yes.

>> does Luke hate Paul
no.
```

ProofWriter

Facts and rules (you can provide your own):

Paul is a SIcilian.
Every Sicilian is an Italian.
If someone is an Italian then he loves pizza.
If someone loves pizza then he hates pineapple.
If someone loves pineapple then he eats pineapple.
Luke loves pineapple.
If Luke eats pineapple then Paul hates Luke.

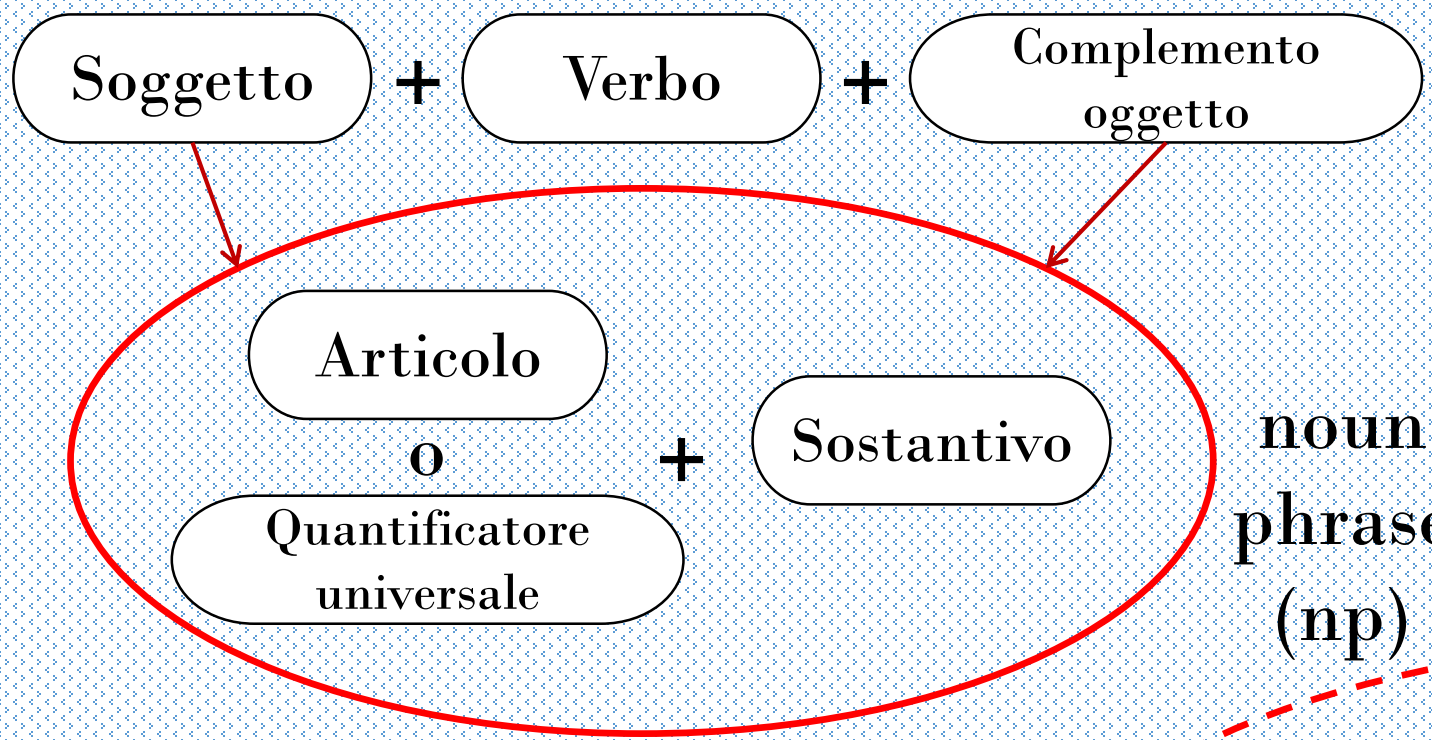
Is it true?

Luke hates Paul

Submit

ProofWriter answer: True
Proof (raw): # sent7@int1 # sent5@int2 sent6 ; with int1: Paul hates Luke. ; int2: Luke eats pineapple.
Proof (details):
sent6: Luke loves pineapple.
sent5: If someone loves pineapple then he eats pineapple.
===> int2: Luke eats pineapple.
---
int2: Luke eats pineapple.
sent7: If Luke eats pineapple then Paul hates Luke.
===> int1: Paul hates Luke.

CONCLUSIONI E SVILUPPI FUTURI



noun phrase (np)

«Professors teach» teach(professor)	*«Great professors teach»* teach(great professor)
«Every professor teaches» teach(X):-professor(X)	*«Every great professor teaches»* teach(X):-professor(X), great(X)
«X is a professor» professor(X)	*«X is a great professor»* professor(X). great(X).