

Tecnologie del Linguaggio Naturale

Progetto di laboratorio: Traduzione interlingua

<https://github.com/GabrielePicco/Inter-language-translator>

Gabriele Picco



Anno Accademico 2018/2019

Indice

1. Descrizione concettuale del sistema di traduzione.....	3
2. Grammatica CF con semantica e parsificazione delle frasi.....	4
2.1. Frasi con verbi transitivi.....	4
2.2. Frasi con verbi intransitivi e complementi.....	5
2.3. Frasi con “there is” e “there are”	7
3. Parsificazione, disambiguazione e generazione del sentence plan	7
3.1. Descrizione del processo di traduzione	8
4. Esempi di traduzione	8

Traduzione interlingua

Da un punto di vista concettuale, il processo di traduzione consiste nell'effettuare un'analisi linguistica completa della frase che si vuole tradurre per generare una formalizzazione interlingua mediante un meccanismo di rappresentazione della conoscenza. La formula viene successivamente utilizzata per generare la frase nel linguaggio desiderato.

Nella seguente relazione si presenta la soluzione realizzata per la traduzione dall'inglese all'italiano delle frasi di esempio contenute nel file:

https://github.com/GabrielePicco/Inter-language-translator/blob/master/TranslatorCorePython/test_sentences.txt

1. Descrizione concettuale del sistema di traduzione

Da un punto di vista concettuale è possibile identificare le seguenti componenti all'interno del sistema:

- La grammatica formale libera dal contesto aumentata con la semantica, responsabile dell'analisi sintattica e semantica delle frasi
- Il parser che utilizzando la grammatica parsifica le frasi e ne deriva una formula logica del primo ordine
- Una componente responsabile di disambiguare le rappresentazione interlingua applicando dei semplici meccanismi di ragionamento
- Una componente responsabile della creazione dei sentence plan, derivandoli dalla formalizzazione interlingua
- La componente responsabile delle realizzazione del sentence plan in lingua italiana (SimpleNLG-IT)

Si analizzano di seguito le classi ed i relativi metodi dell'implementazione realizzata, facendo riferimento ed analizzando le relazione delle componenti concettuali descritte.

2. Grammatica CF con semantica e parsificazione delle frasi

La grammatica realizzata, viene presentata in relazione alle frasi di esempio, discutendone le scelte di formalizzazione.

2.1 Frasi con verbi transitivi

Frase come: “Angus imagines a thing”, “you are imagining things”, “Irene is chasing a dog”, sono accomunate dal fatto di contenere un verbo transitivo, di cui è possibile identificare un soggetto ed un oggetto dell’azione specificata dal verbo.

La formalizzazione proposta è costituita da un predicato *clause*(soggetto, verbo, oggetto), oltre ai predicati *verbTense* e *objectRef* che specificano rispettivamente il tempo verbale e se l’oggetto o il soggetto sono entità singole o composte (singolare o plurale).

La formula che si vuole ottenere per la frase “you are imagining things”, corrisponde quindi a:

exists z. (objectRef(z, thing, pl) & clause(you, imagine, z) & verbTense(imagine, progPres))

La regola di produzione S radice della frase è composta da una frase nominale (NP) ed una frase verbale (VP), dove il costituente VP a sua è composto da una frase verbale transitiva TVP che ha come oggetto una frase nominale NP.

$S[SEM = \langle ?subj(?vp) \rangle] \rightarrow NP[-PronEXP, NUM=?n, SEM=?subj] VP[NUM=?n, SEM=?vp]$
 $S[SEM = \langle ?subj(?vp) \rangle] \rightarrow NP[PronEXP=you, SEM=?subj] VP[NUM=pl, SEM=?vp]$

$TVP[NUM=?n, SEM=?g] \rightarrow AUX[NUM=?n] TVP[-NUM, SEM=?g, TNS=progPres]$
 $TVP[SEM=\langle \lambda X x.X(\lambda y.(clause(x, imagine, y) \& verbTense(imagine, progPres))) \rangle] \rightarrow 'imagining'$

Se il tempo verbale della frase è progressivo presente, TVP può essere a sua volta composto da un ausiliare AUX ed un costituente TVP (il -NUM nella regola di produzione evita la ricorsività dell’ausiliare, rendendo esterne alla grammatiche frasi del tipo: “you are are imagining”).

$AUX[NUM=pl, SEM=\langle \lambda x.x \rangle] \rightarrow 'are'$

Il motivo della duplicazione della regola di produzione $S \rightarrow NP VP$ consiste nella gestione del caso particolare del pronome “you” come soggetto. Mentre infatti in generale per i pronomi personali ed i nomi propri il numero deve coincidere con il numero della frase verbale VP, il pronome “you” (anche se inteso come singolare “tu”), risulta grammaticalmente scorretto

con frasi verbali che presuppongono il numero singolare nel soggetto. Gestendo “you” come eccezione si escludono frasi scorrette dalla grammatica, come: “you is ...”.

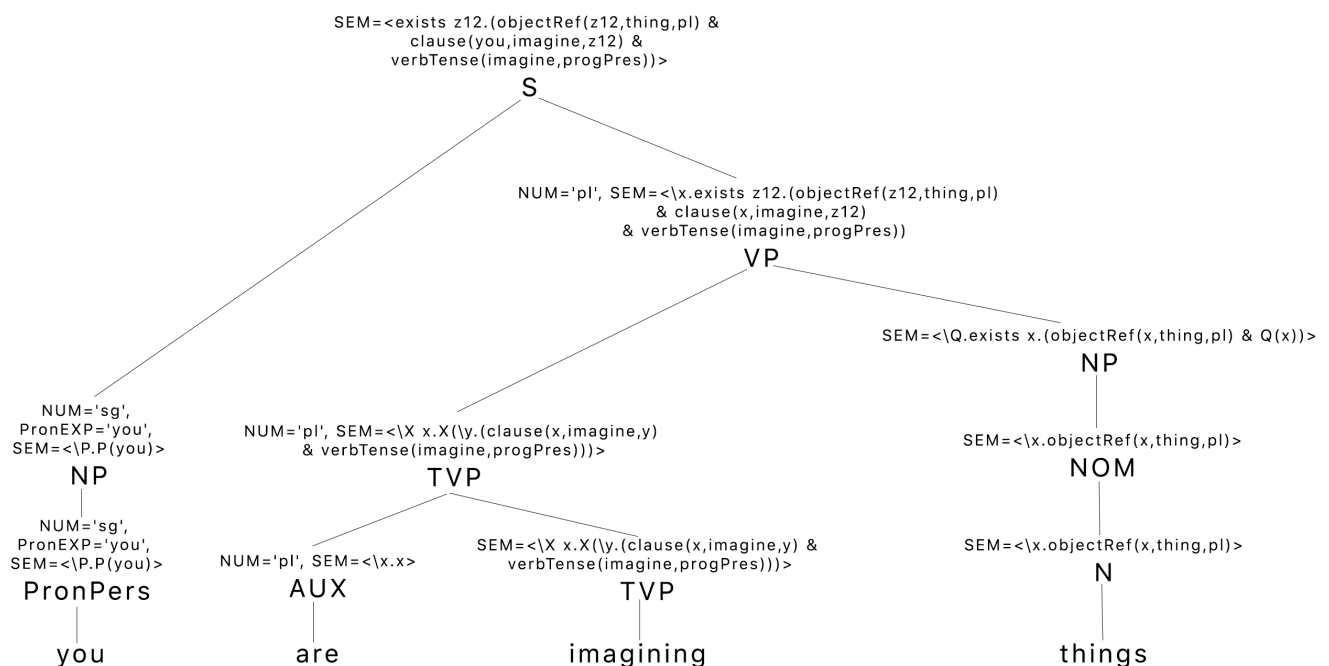
La grammatica definita permette di parsificare diverse possibili variazioni della frase, sarebbe infatti possibile avere un nome singolare o plurale come soggetto.

Inoltre il costituente NP può essere accompagnato da un determinante (NP -> Det Nom), come “a”, “an”, “all” o “every”.

La frase “all dogs are chasing Angus” viene formalizzata nella seguente formula:

$all\ x.(objectRef(x,dog,pl) \rightarrow (clause(x,chase,angus) \ \&\ \ verb\ Tense(chase,progPres)))$

Si riporta di seguito l'albero di parsificazione di una delle frasi di esempio e la relativa semantica:



2.2 Frasi con verbi intransitivi e complementi

Con un approccio simile è possibile includere nella grammatica frasi composte da un soggetto ed un verbo intransitivo. In questo caso la formalizzazione logica consiste in un predicato $clause(soggetto, verbo)$, oltre ai predicati $objectRef$ e $verbTense$ che come nel caso precedente specificano le caratteristiche del soggetto ed il tempo verbale.

Una frase del tipo “an opportunity is flying” viene perciò formalizzata nella seguente formula:

2.3 Frasi con “there is” e “there are”

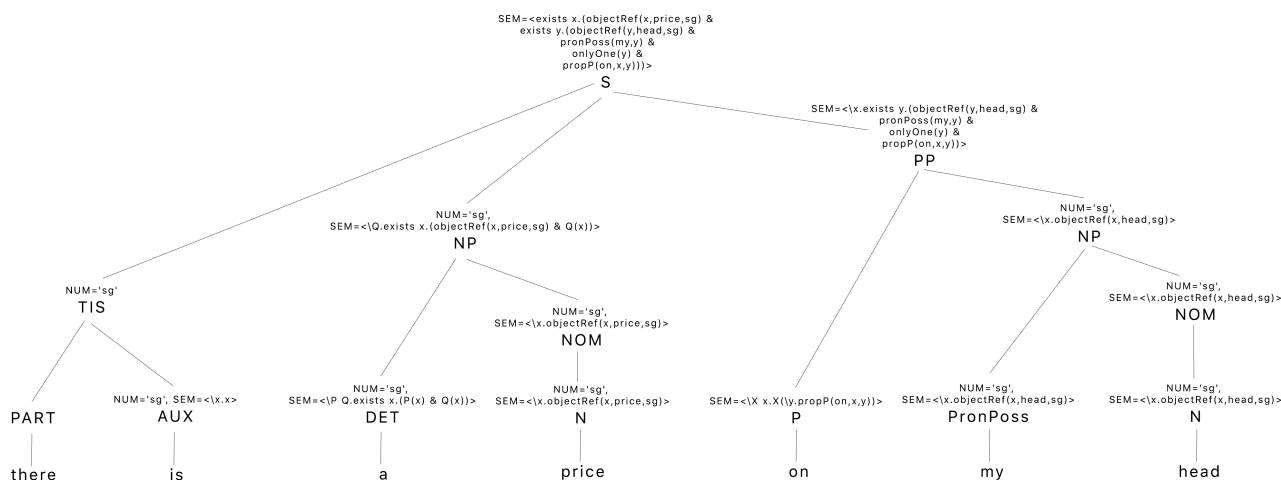
La grammatica può essere estesa per includere frasi della forma “there is..” e “there are...”.

La formalizzazione logica proposta consiste nel far assumere al quantificatore esistenziale il ruolo principale nella frase. Una frase come “there is a dog” viene formalizzata nella seguente formula:

$exists\ x.(objectRef(x,dog,sg))$

Inoltre si vuole estendere la grammatica per permettere frasi in cui è specificato il complemento di stato in luogo, come la frase di esempio “there is a price on my head”. L'estensione aggiunge una produzione dalla costituente radice nel composto “there is” o “there are”, seguito una frase nominale NP ed una frase preposizionale PP. Il costituente PP è a sua volta composto da una preposizione P ed una frase nominale NP.

Si riporta di seguito l'albero di parsificazione della frase di esempio “there is a price on my head”:



3. Parsificazione, disambiguazione e generazione del sentence plan

L'architettura del sistema è composta da un GatewayServer che permette di utilizzare la libreria SimpleNLG (mediante socket) dalle componenti implementate nel linguaggio Python.

Le principali classi Python definite sono:

- WordTranslator: responsabile delle traduzioni delle singole parole
- Reasoner: applica delle semplici inferenze per la disambiguazione

- **EnglishTranslator**: responsabile del processo di traduzione, dalla parsificazione della formula, alla generazione e realizzazione del sentence plan. Utilizza le classi precedenti come componenti nel processo di traduzione.

3.1 Descrizione del processo di traduzione

Il metodo *translate_sentence_to_formula(sentence)* della classe **EnglishTranslator** prende in input una frase, di cui effettua la formalizzazione mediante la grammatica precedentemente descritta ed il parser della libreria **NLTK**.

Alla formula vengono applicate le regole di inferenza per la disambiguazione appositamente definite per gli esempi considerati. Il processo di disambiguazione potrebbe essere esteso utilizzando ontologie come **SUMO** a cui corrisponde una rappresentazione nella logica del primo ordine.

La classe **EnglishTranslator** si occupa inoltre della generazione del sentence plan. Mediante un meccanismo di pattern matching basato sulle espressioni regolari, le variabili e le costanti vengono estratte dalla formula. L'algoritmo individua le clausole all'interno della frase, (o eventualmente il quantificatore esistenziale se nessun clausola è presente), da cui estrae ricorsivamente tutti i componenti della frase. La ricorsività implicita nel processo di generazione del sentence plan rispecchia la ricorsività dei linguaggi naturali.

Le singole parole vengono tradotte utilizzando il metodo *translate_word(word)* della classe **WordTranslator**.

La libreria **SimpleNLG** viene infine utilizzata per la effettiva realizzazione della frase in lingua italiana.

4. Esempi di traduzione

Frase Input	Angus imagines a thing
Formula	exists z.(objectRef(z,thing,sg) & clause(angus,imagine,z) & verbTense(imagine,pres))
Traduzione	Angus immagina una cosa

Frase Input	Dogs are chasing Irene
Formula	exists x.(objectRef(x,dog,pl) & clause(x,chase,irene) & verbTense(chase,progPres))
Traduzione	Dei cani stanno inseguendo Irene.

Frase Input	you are imagining things
Formula	exists z.(objectRef(z,thing,pl) & clause(you,imagine,z) & verbTense(imagine,progPres))
Traduzione	Tu stai immaginando delle cose.

Frase Input	there is a price on my apple
Formula	exists x.(objectRef(x,price,sg) & exists y.(objectRef(y,apple,sg) & pronPoss(my,y) & onlyOne(y) & propP(on,x,y)))
Traduzione	C'è un prezzo sopra la mia mela.

Frase Input	there is a price on my head
Formula	exists x.(objectRef(x,price,sg) & exists y.(objectRef(y,head,sg) & pronPoss(my,y) & onlyOne(y) & propP(on,x,y)))
Traduzione	C'è una taglia sopra la mia testa.

In questa traduzione è evidente il ruolo della disambiguazione, che applicando la regola:
 $\text{propP}(\text{on},x,y) \ \& \ \text{objectRef}(x,\text{price}) \ \& \ \text{objectRef}(y,\text{head}) \Rightarrow \text{objectRef}(x,\text{reward})$
 permette di tradurre *price* in taglia, invece cdi prezzo come nel precedente esempio.

Frase Input	your big opportunity fly out of here
Formula	exists y.(objectRef(y,opportunity,sg) & adj(big,y) & pronPoss(your,y) & onlyOne(y) & clause(y,fly) & verbTense(fly,pres) & exists c.(verbCompl(fly,c) & compl(c,out,of,here)))
Traduzione	La tua grande opportunità vola via.

Frase Input	your big opportunity is flying out of here
Formula	exists y.(objectRef(y,opportunity,sg) & adj(big,y) & pronPoss(your,y) & onlyOne(y) & clause(y,fly) & verbTense(fly,progPres) & exists c.(verbCompl(fly,c) & compl(c,out,of,here)))
Traduzione	La tua grande opportunità sta volando via.