**Università degli Studi di Torino**

Dipartimento di Informatica

Corsi di Laurea Magistrale in Informatica



**Relazione esercitazioni TLN**

**(prof. Mazzei)**

Salvatore Coluccia

a.a. 2019/2020

# Introduzione e scelta esercitazione

Per il progetto ho scelto l’esercitazione 1 e cioè l’implementazione di un traduttore transfer sintattico dalla lingua italiana alla lingua inglese.

Per la parte relativa al parsing ho utilizzato la libreria TINT per la creazione dell’albero a dipendenze.

Come linguaggio di programmazione ho utilizzato Java 8.

# Struttura ed implementazione

L’implementazione del sistema segue i passi suggeriti dalla consegna:

1. Parsificazione delle frasi in input e creazione dell’albero a dipendenze tramite libreria TINT
2. Trasformazione da albero sintattico a sentence plan per SimpleNLG
3. Generazione traduzione utilizzando SimpleNLG

La classe principale dalla quale si avvia il sistema è Executor.java.  
Il metodo main, dopo aver inizializzato le strutture dati a supporto dell’algoritmo, suddivide l’esecuzione mappando i passi descritti precedentemente:

1. ***constructDependencyTree(sentence):*** parsifica e costruisce l’albero a dipendenze Tint per la frase in input
2. ***translationAndannotationToSPhraseSpec(dependencyTree)***: costruisce il sentence plan per SimpleNLG prendendo in input il dependency tree costruito al passo precedente
3. ***realiser.realiseSentence(simpleNLGInput)***: generazione della traduzione usando SimpleNLG

Per la fase di traduzione lessicale ho utilizzato un dizionario statico che contenesse la traduzione inglese delle parole utilizzate nelle frasi di input: variabile ***itaToEnDict***.

## Sentence Plan

Per costruire il sentence plan per SimpleNLG ho dovuto mappare l’output di TINT con l’oggetto SPhraseSpec di SimpleNLG (che è l’oggetto che rappresenta il sentence plan).   
Per utilizzare l'albero a dipendenze generato da TINT ho estratto le info come json e mappate tutte in oggetti del tipo ***Dependency.java***.  
Ho quindi estratto il "minimo comune multiplo" dei tipi di dipendenze in output da TINT sulle tre frasi da tradurre e fatto il mapping tra TINT e il metodo da chiamare sull'oggetto SPhraseSpec di SimpleNLG.  
L’implementazione di questo mapping è all’interno del metodo ***DependencyMapping.setSimpleNlgPos***.

Il metodo appena citato si occupa anche di definire tutte le regole di transfer necessarie per gestire ad esempio:

* genitivo sassone

*tramite il metodo setFeature(Feature.POSSESSIVE, true)*

* pronomi impliciti

*tramite il metodo sentence.setSubject(…)*

* Posizione aggettivi rispetto a noun

*Tramite il metodo setPreModifier(…)*

## Risultati

Eseguendo la classe Executor questi sono gli output relativi alle traduzioni per le tre frasi in input:

***[È la spada laser di tuo padre]*** *--> [It is your father's laser sword.]*

***[Ha fatto una mossa leale]*** *--> [He did a loyal move.]*

***[Gli ultimi avanzi della vecchia Repubblica sono stati spazzati via]*** *-->* *[The old Republic's last scraps was swept away.]*

La qualità della traduzione mi sembra accettabile, chiaramente è un sistema demo che va bene solo per le tre frasi in input. Per estenderlo bisognerebbe in primis estendere/modificare il dizionario isomorfo e modificare e aggiungere ulteriori regole di transfer.