

# Progetto Linguaggi e Traduttori

di Marco Zanghì e Tommaso Ruscica

## **Specifiche e stato dell'arte**

Si vuole progettare un sistema per il riconoscimento di parole in lingua italiana con l'ausilio di un dizionario contenente i vocaboli e le definizioni della lingua Italiana. Il sistema sarà in grado di riconoscere le parole anche nelle sue forme flesse ovvero le particolari variazioni di un vocabolo in genere numero o coniugazione verbale.

Il riconoscimento lessicale è una componente essenziale del processo di elaborazione automatica mediante calcolatore elettronico dei linguaggi naturali (NLP). Tale processo viene suddiviso in fasi diverse, simili a quelle che si possono incontrare nel processo di elaborazione di un linguaggio formale:

- **analisi lessicale:** estrazione delle singole parole da un'espressione linguistica
- **analisi grammaticale:** associazione delle parti del discorso a ciascuna parola nel testo
- **analisi sintattica:** rilevazione e organizzazione di una struttura sintattica (parse tree)
- **analisi semantica:** assegnazione di un significato (semantica) alla struttura sintattica

L'analisi semantica è una delle più complesse operazioni a causa delle caratteristiche intrinseche di ambiguità del linguaggio umano. (Operazioni di disambiguazione)

## **Struttura del programma**

Si è deciso di suddividere il programma in due fasi:

1. La prima fase consiste nell'estrazione e formattazione di tutte le informazioni utili alla generazione di un dizionario, utilizzando in input un vocabolario in formato testuale.
2. La seconda fase consiste nel verificare che una parola appartenga alla lingua italiana attraverso una ricerca nel dizionario generato in precedenza e tenendo in considerazione delle possibili flessioni della stessa parola.

Fase 1 : ParserDizionario

E' stato analizzato il file vocabolario.txt attraverso uno Scanner il quale ha permesso di ricavare per ogni entry del vocabolario le informazioni di interesse:

- vocabolo
- tipologia: sostantivo, verbo , aggettivo ....
- definizione

Queste informazioni sono utili alla generazione di un istanza della classe Info. Tali istanze verranno organizzate in un HashMap <String,LinkedList<Info>>, ove la chiave di tipo string è rappresentata dalla radice del vocabolo, ottenuta dal metodo estraiRadice() combinando le informazioni del vocabolo e del tipo vocabolo, l'oggetto riferito dalla chiave è una lista delle possibili info estratte associate alla stessa radice. Infatti in fase di aggiornamento del dizionario all'atto dell'inserimento di una nuova Info si deve verificare se la radice è già presente nel dizionario e in tal caso inserire in aggiunta le nuove informazioni estratte.

A causa della presenza di grandi quantità di particolarità si è scelto di limitare il processo di estrazione delle informazioni anche alle particolarità sui vocaboli di tipo sostantivo. A tal proposito si analizza la definizione di un vocabolo per ricercare se vi sono inserite delle particolarità. Ad esempio nella definizione del vocabolo "cane" di tipo sostantivo maschile vi è specificato che il suo femminile è rappresentato dal vocabolo "cagna". E' dunque necessario inserire nel nuovo dizionario che si sta generando anche tale informazione, quindi inserendo nell'hashmap anche la chiave-radice "cagn" e le informazioni del tipo ("Sostantivo femminile").

Per migliorare la leggibilità del codice è stata implementata anche l' Enum Grammatica che serve ad identificare tutte le tipologie dei vocaboli.

```
SOSTANTIVO_MASCHILE("o","i"),  
SOSTANTIVO_MASCHILE_INVARIANTE(),  
SOSTANTIVO_MASCHILE_FEMMINILE("e","i"),  
SOSTANTIVO_FEMMINILE("a","e"),  
SOSTANTIVO_FEMMINILE_INVARIANTE(),
```

```
VERBO_1("are","o","i","a","iamo","ate","ano","avo","avi","ava","avamo","avate","avano","ai","asti","ò","ammo"  
,"aste","arano","erò","erai","erà","eremo","erete","eranno","erei","eresti","erebbe","eremmo","ereste","erebber  
o","i","iamo","iate","ino","assi","asse","assimo","aste","asserò","a","i","iamo","ate","ino","ando","ante","ato"),
```

```
VERBO_2("ere","o","i","e","iamo","ete","ono","evo","evi","eva","evamo","evate","evano","ei","etti","esti","é","et  
te","emmo","este","erono","ettero","erò","erai","erà","eremo","erete","eranno","rò","rai","rà","remo","rete","ran  
no","erei","eresti","erebbe","eremmo","ereste","erebbero","a","iamo","iate","ano","essi","esse","essimo","este"  
,"essero","i","a","iamo","ete","ano","endo","ente","uto"),
```

```
VERBO_2_Plus("rre"),
```

```
VERBO_3("ire","o","i","e","iamo","ite","ono","isco","isci","isce","iscono","ivo","ivi","iva","ivamo","ivate","ivano","
```

il", "isti", "i", "immo", "iste", "irono", "irò", "irai", "irà", "iremo", "irete", "iranno", "irei", "iresti", "irebbe", "iremmo", "ireste", "i  
rebbero", "a", "isca", "iamo", "iate", "ano", "iscano", "issi", "isse", "issimo", "iste", "issero", "i", "isci", "a", "isca", "iamo", "it  
e", "ano", "iscano", "endo", "ente", "iente", "ito"),

LOCUZIONE(),  
LOCUZIONE\_AVVERBIALE(),  
AGGETTIVO("o", "i", "a", "e"),  
PREPOSIZIONE(),  
PRONOME();

Come è possibile notare dal codice sovrastante ogni definizione del tipo vocabolo contiene al suo interno le possibili flessioni che possono essere accostate ad una radice di quel particolare tipo.

## *Fase 2 : Ricerca e Matching*

Questa fase permette all'utente di inserire una parola da ricercare nel vocabolario creato nella fase precedente. Inserita la parola (es. abbandonare) si farà un match con tutte le radici del vocabolario. Nel caso in cui la radice non è stata trovata, dalla parola iniziale si elimina un carattere e si ripete la ricerca (es. abbandonar...abbandona) fin quando non si ha uno o più match con le radici del vocabolario (es. abandon). Trovata la parola nel vocabolario si prendono le possibili flessioni per quel tipo\quei tipi cui appartiene la radice (es.verbo e sostantivo) e si concatenano con la radice confrontando il risultato con la parola inizialmente ricercata. Se si ha un match per una determinata flessione, si stampa su schermo il risultato ottenuto e quindi il tipo della parola ricercata.

Esempio:

Inizializzazione...

Inserisci parola:

cane

Parola non trovata,elimino l'ultimo carattere e riprovo.. [can]

PAROLA TROVATA, RADICE: can del vocabolo: cane PAROLA CERCATA:cane TIPO:  
SOSTANTIVO\_MASCHILE\_FEMMINILE

Parola non trovata,elimino l'ultimo carattere e riprovo.. [ca]

Parola non trovata,elimino l'ultimo carattere e riprovo.. [c]

Parola non trovata,elimino l'ultimo carattere e riprovo.. []

Parola non trovata!

Inserisci parola:

cana

Parola non trovata,elimino l'ultimo carattere e riprovo.. [can]

Parola non trovata,elimino l'ultimo carattere e riprovo.. [ca]

Parola non trovata,elimino l'ultimo carattere e riprovo.. [c]

Parola non trovata,elimino l'ultimo carattere e riprovo.. []

Parola non trovata!

Inserisci parola:

cani

Parola non trovata,elimino l'ultimo carattere e riprovo.. [can]

PAROLA TROVATA, RADICE: can del vocabolo: cane PAROLA CERCATA:cani TIPO:  
SOSTANTIVO\_MASCHILE\_FEMMINILE

Parola non trovata,elimino l'ultimo carattere e riprovo.. [ca]

Parola non trovata,elimino l'ultimo carattere e riprovo.. [c]

Parola non trovata,elimino l'ultimo carattere e riprovo.. []

Parola non trovata!

Inserisci parola:

Nell'esempio precedente viene mostrato un caso particolare: in fase di parsing del vocabolario nella definizione del vocabolo "cane" è stata estrapolata una particolarità che identifica il femminile del vocabolo "cane" in "cagna" . Il sistema dunque è in grado di verificare che il termine "cana" non è una parola ammissibile. Quest'ultima parola infatti non è stata trovata al contrario della parola "cagna" come mostrato successivamente.

Inserisci parola:

cagna

Parola non trovata,elimino l'ultimo carattere e riprovo.. [cagn]

PAROLA TROVATA, RADICE: cagn del vocabolo: cagna PAROLA CERCATA:cagna TIPO:  
SOSTANTIVO\_FEMMINILE

Parola non trovata,elimino l'ultimo carattere e riprovo.. [cag]

Parola non trovata,elimino l'ultimo carattere e riprovo.. [ca]

Parola non trovata,elimino l'ultimo carattere e riprovo.. [c]

Parola non trovata,elimino l'ultimo carattere e riprovo.. []

Parola non trovata!

Inserisci parola:

Per l'esecuzione del programma leggere attentamente il file readme.txt in allegato.