**TECNOLOGIE DEL LINGUAGGIO NATURALE**

**RELAZIONE PROGETTO: TRADUTTORE DA ITALIANO A ITALIANO-YODA**

**UNIVERSITÀ DEGLI STUDI DI TORINO A.A 2018/19**

**STUDENTI DOCENTE**

Donadio Gaetano 899187 prof. Mazzei Alessandro

Marrazzo Alessandro 898842

**SISTEMA DI TRADUZIONE ITALIANO -> ITALIANO-YODISH**

**Introduzione**

In tempi non molto lontani la traduzione di una frase da una lingua ad un’altra era affidata alle persone. Esse usavano la propria conoscenza delle due lingue e, aiutate da dizionari, facevano la traduzione a mano.

Con l’avvento dell’informatica, si sono diffusi ampiamente i traduttori automatici, che, data una grammatica e degli algoritmi di traduzione, traducono frasi da una lingua di origine ad un’altra di arrivo.

L’obiettivo di questo progetto è la creazione di un sistema di traduzione automatica dall’italiano all’italiano-yodish. Si sfrutteranno una grammatica context-free, cioè libera dal contesto, da noi creata e degli algoritmi di analisi e traduzione, come ad esempio il CKY.

L'algoritmo Cocke-Younger-Kasami (CKY) è un algoritmo di analisi per grammatiche context-free. Esso impiega un procedimento bottom-up con gestione della memoria tramite programmazione dinamica per generare l’albero sintattico di una frase. La versione standard di CKY funziona solo su grammatiche context-free fornite in Chomsky Normal Form (CNF), cioè con regole grammaticali binarie (del tipo **A -> B C)** e regole lessicali unarie (del tipo **A -> a**). Questo, naturalmente, non lede la generalità dell’algoritmo, poiché qualsiasi grammatica context-free, che non genera la stringa vuota, può essere rappresentata in Chomsky Normal Form, aggiungendo alcune regole o modificando opportunamente le regole della grammatica in uso. L’intuizione, dietro l’algoritmo, è che, se abbiamo una regola **A -> BC** e **A** domina la frase dalla parola *i* alla parola *j*, allora esisterà un punto *k* tale che **B** domina la frase dal punto *i* al punto *k*, e **C** dal punto *k* al punto *j*.

Utilizzeremo una traduzione di tipo transfer, ovvero una traduzione in cui, a seguito di un'analisi morfologica si genera la struttura sintattica della frase da tradurre e, a partire da questa, viene generata la struttura sintattica della frase nella lingua d'arrivo. Infine, mediante un dizionario bilingue, vengono tradotte e riordinate le parole che compongono la frase. Nel sistema implementato, l'analisi prevede la consultazione della grammatica italiana, dove vi sono alcune delle regole grammaticali e lessicali della lingua italiana, l’applicazione dell’algoritmo CKY per la generazione dell’albero sintattico e la traduzione, tramite modifiche dell’albero sintattico, nella lingua di arrivo.

Le due lingue coinvolte nella traduzione sono molto simili; la differenza principale è che in italiano le frasi si generano nella forma soggetto - verbo - oggetto, mentre in italiano-yodish si antepone l’oggetto al soggetto e al verbo, mantenendo i restanti costrutti inalterati.

**Strutture dati utilizzate**

La grammatica italiana viene memorizzata all’interno di un hashmap. È stata scelta questa tipologia di struttura dati per indicizzare al meglio le regole, cosi da avere un accesso diretto ad esse.

Ogni coppia <chiave, valore> dell'hashmap è formata, rispettivamente, da una stringa rappresentante il corpo di una regola, e da una lista di stringhe, ognuna delle quali rappresenta un sintagma generabile dalla chiave.

Ad esempio, se abbiamo una regola **S -> VP ADV**, essa verrà salvata nell’hashmap come una tupla, avente come chiave la stringa “**VP ADV**”e come valore una lista contenente il sintagma rappresentato dalla stringa “**S**”. Allo stesso modo, se è presente una regola simile in corpo ma diversa in testa del tipo: **VP -> VP ADV,** la stringa rappresentante **"VP"** verrà aggiunta alla lista.

In questo modo, durante l’esecuzione dell’algoritmo CKY, si potrà controllare direttamente se esistano uno o più sintagmi che soddisfino la regola.

Ogni sintagma è rappresentato da una struttura dati, denominata TREE, che ha al suo interno una stringa, rappresentante il valore del sintagma e una lista di alberi figli, rappresentanti i sintagmi che lo hanno generato.

Tree implementa anche alcuni metodi che consentono l’accesso, l’aggiunta o la rimozione diretta di un elemento dalla lista dei figli. Tale struttura dati è stata scelta, in quanto è quella che rappresenta al meglio la relazione tra sintagmi.

L’algoritmo CKY sfrutta una tabella per la generazione dell’albero sintattico. Tale tabella è una matrice di celle **n** x **n,** dove n rappresenta il numero di parole nella frase. Ogni cella della tabella può contenere più di un elemento, ed è rappresentata con un oggetto, denominato CELL, al cui interno è presente una lista di alberi, ognuno dei quali rappresenta un elemento presente nella cella.

La classe Cell implementa anche un metodo che permette di aggiungere elementi alla cella e un metodo che consente di accedere direttamente al valore i della lista.

**Descrizione** **del sistema**

Il sistema di traduzione può essere suddiviso in diverse fasi:

* **parsificazione** della grammatica
* **derivazione** dell’albero sintattico della frase in italiano
* **trasformazione** dell’albero generato precedentemente in modo da soddisfare le regole dell'italiano-yodish,
* **formulazione** della frase finale.

Per la realizzazione del sistema, è stata scritta una grammatica context-free per l’Italiano che permette di parsificare alcune frasi d’esempio.

La grammatica è stata costruita in Chomsky Normal Form (CNF), in cui le regole non lessicali sono binarie, mentre le regole lessicali sono unarie.

La fase di parsificazione è effettuata da una classe **Parser** che legge da un apposito file la grammatica e memorizza le regole nel sistema.

La fase di derivazione dell’albero sintattico viene eseguita implementando l’algoritmo CKY. L’algoritmo prende in input una sequenza di parole e inizia a riempire la matrice posizionando i POS tag riferiti ad ognuna di esse lungo la diagonale principale. In seguito, per ogni cella **[j, i]** sopra la diagonale principale, si verifica se nella grammatica sia presente una regola della forma **A -> B C** tale che **B** è rappresentata da un valore nella cella **[j, k-1]** e **C** da un valore nella cella **[k, i],** con k compreso tra **j+1** e **i.** Poichè in ogni cella può essere presente più di un elemento, vengono esaminate tutte le possibili combinazioni tra tali elementi. In caso una o più combinazioni abbiano soddisfatto una regola, vengono aggiunti alla cella **[j, i]** i valori generati dalla regola.

La porzione della matrice al di sotto della diagonale è ininfluente ai fini dell’algoritmo, dunque non viene utilizzata.

Gli indici utilizzati dai for per lo scorrimento e l’accesso alle celle della matrice sono stati modificati in modo tale da non avere righe e colonne inutilizzate nella parte superiore della matrice.

Durante l’esecuzione dell’algoritmo, invece di utilizzare il backtrace per tener traccia degli elementi che costituiscono l'albero sintattico della frase, questo si costruisce parallelamente al popolamento della matrice. Ogni qualvolta una regola grammaticale o lessicale viene soddisfatta, si genera un nuovo albero avente come valore il sintagma generato dalla regola e come figli i sottoalberi presenti nelle celle che hanno contribuito al soddisfacimento di tale regola. Successivamente l'albero viene aggiunto alla cella. Così facendo in ogni cella avremo l’intero sottoalbero avente come radice il sintagma presente nella cella.

Se, al termine dell’algoritmo, nella cella **[0, n-1],** dove **n** è il numero di parole nella frase, è presente l’albero avente valore **S**, il quale rappresenta la frase completa, l’algoritmo ha avuto successo e restituirà l’albero sintattico.

La fase di trasformazione consiste nello spostamento di alcuni rami dell'albero generato precedentemente.

Gli spostamenti vengono effettuati sulla base di considerazioni di tipo linguistico, riguardanti la grammatica italiano-Yodish. Poichè in quest'ultima, le frasi si formano anteponendo il complemento al soggetto e al predicato, si effettua una visita ricorsiva in profondità, in cui si ricerca tale complemento.

Dato un sintagma di tipo Verbal-Phrase (VP) avente come figlio sinistro un verbo V, il complemento è identificato come il sintagma contenuto nel figlio destro. Una volta che si è trovato tale sintagma, esso viene rimosso dalla posizione originaria e aggiunto alla lista dei figli della radice nella posizione iniziale. In alcuni casi (tipicamente quando il verbo essere funge da copula) il complemento può essere sostituito da una parte nominale. Il procedimento, impiegato nella sua ricerca, tuttavia è lo stesso. Nel caso in cui il sintagma trovato sia un avverbio, la sua posizione resta inalterata. Così facendo il complemento (o la parte nominale) viene anteposto al soggetto e al predicato.

La traduzione delle singole parole da una lingua all’altra non viene effettuata poiché essere usano lo stesso insieme di parole, ovvero lo stesso lessico.

Nella fase di formulazione della frase finale, si effettua una visita in profondità sull’albero appena tradotto e viene fornita in output una stringa contenente la frase formata dalle parole contenute nelle foglie dell'albero.