



## Technical Report

TR2.4.1

### TITOLO

**Stato dell' arte ed analisi delle metodologie e delle tecniche per il  
Natural Language Processing**

PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------



## Indice

Abstract.....	3
1.1 Introduzione .....	3
1.2 Definizione del NLP .....	6
1.3 La linguistica computazionale nel paradigma del NLP.....	7
1.4 Trattamento del linguaggio .....	8
1.5 La gerarchia di Chomsky .....	11
1.6 Metodologie e Tecniche del NLP .....	14
1.6.1 Stopwords Removal .....	21
1.6.2 Stemming .....	22
1.6.3 Part-of-Speech Tagging .....	23
1.6.4 Frasi statistiche e composte .....	29
1.6.5 Parser.....	30
1.6.6 Augmented Transition Network .....	35
1.6.7 Word Sense Disambiguation.....	38
2 Bibliografia.....	39

PROGETTO LC3	Revisione n*	0	Del	----
--------------	--------------	---	-----	------

**ABSTRACT**

*In questo technical report si esaminano le principali metodologie e tecniche per il Natural Language Processing.*

*La possibilità di ricercare informazioni su una base documentale o in remoto ha un presupposto fondamentale: l'informazione deve essere disponibile! Documenti, pagine web, file multimediali e tant'altro, devono essere analizzati e compresi dal sistema informatico che si sta utilizzando. Nel corso del seguente report si effettuerà un'analisi delle tecniche del linguaggio naturale, alla base per la comprensione computazionale. Queste risultano essere di notevole importanza per i sistemi di processamento del linguaggio naturale e di estrazione di informazione. Se ne definisce il significato e se ne propongono le diverse problematiche che lo coinvolgono.*

*In particolare si esaminano le principali tecniche per l'analisi lessicale del testo, con particolare riferimento alle procedure di Stopwords Removal, Stemming, di individuazione di Collocazioni, di estrazione di indici e di pesatura dei termini.*

*Si esaminano infine le principali metodologie e tecniche per l'analisi sintattica e semantica, in particolare il Part of Speech (POS), il Parsing e la Word Sense Disambiguation, con particolare riferimento alle metodologie basate su Parsing grammaticale di tipo Context Free e Context Sensitive, ed alle metodologie stocastiche.*

**1.1 INTRODUZIONE**

La capacità di comunicare attraverso un linguaggio è una delle caratteristiche della specie umana. Per questo motivo il trattamento del linguaggio naturale, sin dalle sue origini ha costituito una parte di interesse sia del ramo di Intelligenza Artificiale e sia nell'Informatica in generale. La prima considerazione da fare è che parlare di linguaggio naturale significa riferirsi al linguaggio umano, che si oppone sia al linguaggio artificiale che a quello formale. Da quest'ultimo deriva la prima dipendenza: il linguaggio naturale è di gran lunga più difficile di qualunque linguaggio formale. Anche se per anni, le tecniche per il *Natural Language Processing* (N.L.P.) erano derivate dalla *Teoria del Linguaggio Formale* o dalla *Teoria della Compilazione*, un linguaggio naturale non è un linguaggio di programmazione e per questo motivo è collocato in una propria disciplina. Con l'evolversi della linguistica, si costruivano teorie sintattiche senza tener conto della possibile realizzazione computazionale e gli informatici trattavano tali applicazioni come un problema di ingegneria del software. Attualmente ogni proposta della teoria linguistica trova riscontro in

PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------

uno studio della sua componente computazionale. Il processo di comprensione del linguaggio naturale necessita di un'ingente quantità di conoscenza di base e l'utilizzo di processi intelligenti. La comprensione di un testo in linguaggio naturale avviene mettendo in relazione le parole che lo compongono e confrontando le informazioni così ottenute con le conoscenze possedute dal lettore o che egli può acquisire. Il problema principale quindi è far comprendere ad una macchina ciò che si sta dicendo (oralmente o per iscritto), formando quindi questa caratteristica propriamente naturale.

Il computer, al contrario dell'uomo, non è in grado di attribuire un significato a una sequenza di parole in base a conoscenze già acquisite, anche se è in grado di riconoscere una sequenza di parole e le relazioni che le legano. L'ambito del NLP è estremamente vasto.

In un primo approccio semplicistico potremmo sintetizzarlo in tre passaggi essenziali:

- *Tradurre l'input in un metalinguaggio.*
- *Elaborare l'input su una base di conoscenza pregressa.*
- *Tradurre l'output in linguaggio naturale.*

È chiaro che il N.L.P. abbia di base una grammatica ben determinata che stabilisce: costruzioni sintattiche, domini, sigle, formula, abbreviazioni, nomi propri ed altro. Qualunque input linguistico che venga elaborato è soggetto all'**ambiguità**: anche la frase più semplice è possibile che abbia più interpretazioni possibili a partire dalla stessa costruzione simbolica.

Si potrebbero infatti creare:

- **Ambiguità lessicali:**

- Carlo mangiò una pesca
- Carlo andò a pesca con gli amici.

Che tipo di conoscenza si deve utilizzare per comprendere il termine "pesca" se si riferisce ad un frutto oppure ad un'attività sportiva?

- **Ambiguità sintattiche:**

- Il venditore di giornali del quartiere

PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------

Si vuole indicare che il venditore è del quartiere o i giornali sono locali (trattano notizie del quartiere)?

➤ **Ambiguità semantiche:**

- Pedro diede un pastello ai bambini

Uno per tutti? O uno ad ognuno?

Gli esempi proposti trattano problemi grammaticali, ed appartengono a diversi livelli della descrizione linguistica, e quindi, la loro soluzione richiede conoscenze diverse.

Due sono le grandi aree di applicazione dei sistemi di N.L.P.: le applicazioni basate su dialoghi e quelle basate sul trattamento dell'informazione testuale. Il punto centrale, qualunque sistema usiamo, è chiaramente uno: la conoscenza deve esserci! E quindi, deve essere prima acquisita dal sistema e poi rappresentata internamente in un formato utile ed efficace.

Il modo più semplice di portar a termine il trasferimento di informazione è utilizzare una cascata di trasduttori che analizzano il testo a diversi livelli di descrizione linguistica:

- *analisi lessicale*: scomposizione di un'espressione linguistica in *token* (in questo caso le parole);
- *analisi sintattica*: arrangiamento dei *token* in una struttura sintattica (ad albero: *parse tree*);
- *analisi semantica*: assegnazione di un significato (semantica) alla struttura sintattica e, di conseguenza, all'espressione linguistica.

Alternative sono: usare meccanismi di cooperazione tra i diversi livelli o sviluppare processi di comprensione globale.

Diversi sono i fattori che hanno inciso sullo sviluppo di applicazioni software nel settore del Natural Language Processing, con un costante trasferimento di tecnologia dall'uso di teorie di linguaggi formali nella costruzione di compilatori, al controllo ortografico nei word processor:

PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------



- la rapida crescita tecnologica che ha notevolmente aumentato la velocità dei processori e la capacità di memoria, a cui è corrisposta una drastica diminuzione dei prezzi, e quindi processi computazionalmente costosi caratteristici dell'elaborazione del linguaggio lavorano molto più velocemente;
- la crescente disponibilità su larga scala di risorse linguistiche on-line, come dizionari, thesauri, corpora per diverse lingue annotati con informazioni descrittive;
- la domanda di applicazioni in un mondo in cui i testi elettronici sono cresciuti in volume e dove le comunicazioni elettroniche e la mobilità hanno accresciuto l'importanza della comunicazione multilingua;
- la maturità della tecnologia NLP oggi disponibile, per alcuni compiti specifici.

## 1.2 DEFINIZIONE DEL NLP

Il linguaggio è come *"un sistema per l'espressione dei pensieri, bisogni, ecc., mediante l'uso di suoni parlati o simboli convenzionali"* [1], il linguaggio è un meccanismo di comunicazione il cui tramite è il testo o il discorso.

Il Natural Language Processing è un termine utilizzato in vari modi in differenti contesti, è un ramo dell'informatica che studia *"i sistemi automatici per l'elaborazione del linguaggio naturale, include lo sviluppo di algoritmi per il parsing, la generazione, e l'acquisizione di conoscenza linguistica; l'indagine sulla complessità spaziale e temporale di tali algoritmi; la progettazione di linguaggi formali computazionalmente utili (come grammatiche e formalismi lessicali) per codificare conoscenza linguistica; l'indagine su architetture software appropriate per i vari compiti del NLP; e considerazioni sui tipi di conoscenza non linguistica che vengono a contatto con il NLP. E' un'area di studio discretamente astratta che non mette particolare impegno nello studio della mente umana, e neppure mette particolare impegno nel produrre artefatti utili."*[2].

In definitiva NLP è quella parte dell'informatica che si occupa dei sistemi computerizzati per l'elaborazione del linguaggio.

Il Language Engineering è l'applicazione del NLP alla costruzione di sistemi computerizzati che elaborano il linguaggio per qualche compito come la modellazione del linguaggio stesso, o *"...l'uso strumentale della elaborazione del linguaggio, tipicamente*

PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------



come parte di un sistema più grande con qualche obiettivo pratico, per esempio l'accesso ad un database"[3].

L'elaborazione automatica del linguaggio naturale ha dunque lo scopo di implementare strumenti informatici per analizzare, comprendere e generare testi che gli uomini possano comprendere in maniera naturale, come se stessero comunicando con un altro interlocutore umano e non un computer. È caratterizzato da due prospettive diverse, che mirano l'una all'analisi del materiale testuale, l'altra alla generazione di testi linguistici:

- Natural Language Analysis (NLA) o Natural Language Understanding (NLU): data una frase ha l'obiettivo di darne una rappresentazione della sua analisi, ossia del processo di comprensione della frase;
- Natural Language Generation (NLG): data una grammatica di una lingua naturale, ha lo scopo di produrne frasi di senso compiuto.

### 1.3 LA LINGUISTICA COMPUTAZIONALE NEL PARADIGMA DEL NLP

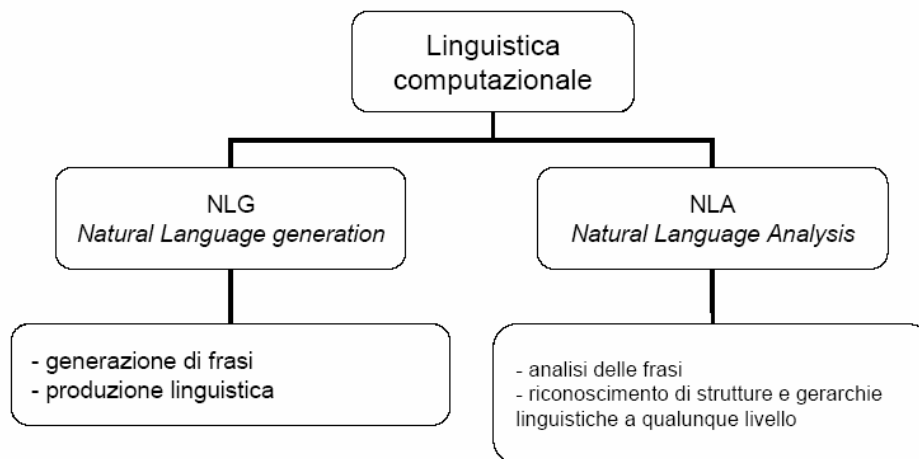
Nella società dell'informazione differenti categorie di utenti (professionisti, amministratori pubblici e comuni cittadini) devono confrontarsi con la necessità quotidiana di accedere a grandi quantità di contenuti digitali *semi-strutturati* o *non strutturati*, all'interno di basi documentali in linguaggio naturale disponibili sul Web o su Intranet locali. La natura non strutturata di tale informazione richiede due passi fondamentali per una sua gestione efficace: ovvero, la selezione dei documenti rilevanti rispetto alle necessità specifiche dell'utente e l'estrazione dell'informazione dai testi, per garantire il suo impiego in altre applicazioni o per compiti specifici. La facilità di tale accesso, la capacità di recuperare l'informazione adeguata in tempi rapidi, la sua gestione e usabilità sono, dunque, parametri chiave per garantire il successo di imprese economiche, lo sviluppo imprenditoriale,

la competitività professionale, così come anche l'integrazione sociale e occupazionale e la formazione permanente.

Gli sviluppi più recenti della *linguistica computazionale* e del *natural language engineering* hanno creato soluzioni tecnologiche dalle enormi potenzialità per migliorare la ricerca e gestione intelligente dell'informazione contenuta nei documenti testuali.

PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------

Le nuove tecnologie della lingua, infatti, permettono ai sistemi informatici di accedere al contenuto digitale attraverso il *Trattamento Automatico della Lingua* (TAL) o *Natural Language Processing* (NLP). Il problema di come acquisire e gestire la conoscenza depositata nei documenti testuali dipende dal suo essere codificata all'interno della rete di strutture e relazioni grammaticali e lessicali che costituiscono la natura stessa della comunicazione linguistica. Sono il lessico e le regole per la combinazione delle parole in strutture sintatticamente complesse che nel linguaggio si fanno veicoli degli aspetti multiformi e creativi dei contenuti semantici.



**Figura 1.1:** La linguistica computazionale nel paradigma del NLP

#### 1.4 TRATTAMENTO DEL LINGUAGGIO

Di particolare interesse e impatto sono le possibilità offerte dalle più recenti tecnologie per *trasformare i documenti testuali in risorse di informazione e conoscenza*. Alla base di questo processo di accesso e analisi del contenuto digitale risiedono tre tipi di tecnologie, fondamentali per ogni sistema[5,6,7]:

**1. strumenti per l'analisi linguistica di testi e l'acquisizione dinamica di conoscenza:** analizzatori morfologici, acquisitori automatici di terminologia e informazione semantica dai testi, parser sintattici; dove *il parsing* è il processo di analisi linguistica attraverso cui viene ricostruita la struttura sintattica di una frase, rappresentata dall'articolazione dei costituenti sintagmatici e dalle relazioni di dipendenza grammaticale (esempio soggetto, complemento oggetto ecc.).

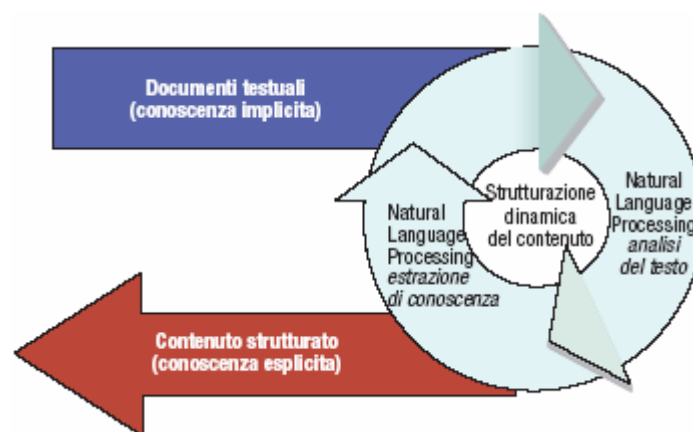
PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------

**2. risorse linguistiche:** lessici computazionali, reti semantico-concettuali multilingui, corpora testuali anche annotati sintatticamente e semanticamente per lo sviluppo e la valutazione di tecnologia del linguaggio;

**3. modelli e standard per la rappresentazione dell'informazione linguistica:** ontologie per il *knowledge sharing* e la codifica lessicale, modelli per la rappresentazione e interscambio di dati linguistici.

Grazie anche alle nuove opportunità offerte dalla tecnologia XML (*eXtensible Markup Language*) è possibile realizzare una maggiore integrazione tra i diversi moduli per l'elaborazione della lingua, e la standardizzazione della rappresentazione dei dati, necessaria per assicurare la loro interscambiabilità e la coerenza del trattamento dell'informazione.

Strumenti di analisi, risorse linguistiche e standard di rappresentazione vengono, dunque, a costituire un'infrastruttura che attraverso l'analisi linguistica automatica dei documenti testuali permette di estrarre la conoscenza implicitamente contenuta in essi, trasformandola in conoscenza esplicita, strutturata e accessibile sia da parte dell'utente umano che da parte di altri agenti computazionali (Figura 1.2).



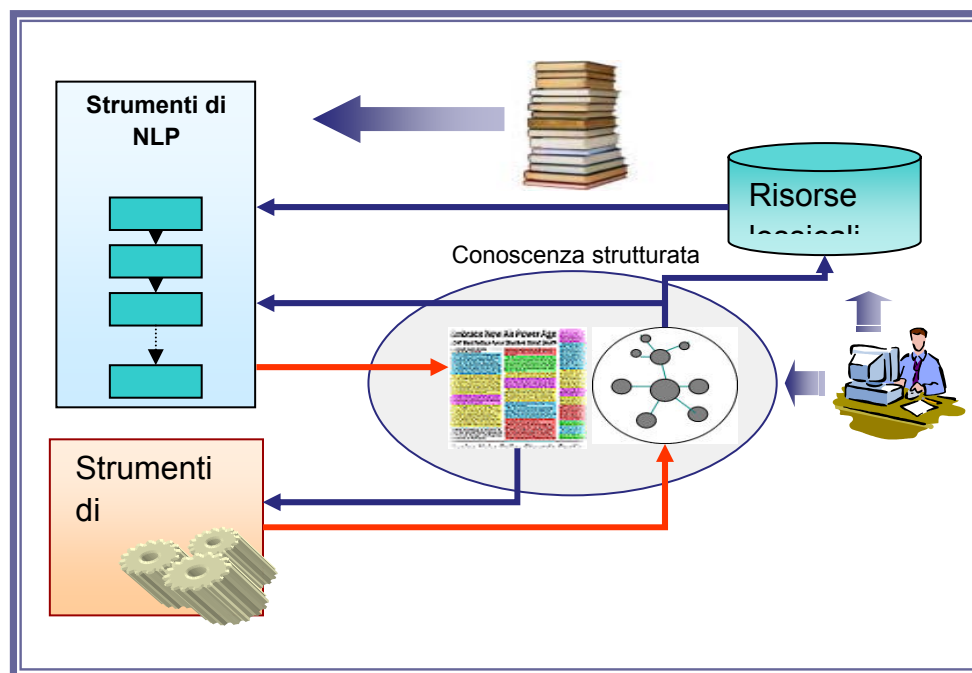
**Figura 1.2:** Dalla conoscenza implicita alla conoscenza esplicita

È importante sottolineare l'aspetto di stretta interdipendenza tra i vari componenti per il TAL, illustrata in maggior dettaglio in Figura 1.3. Gli strumenti di analisi linguistica

PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------

costruiscono una rappresentazione avanzata del contenuto informativo dei documenti attraverso elaborazioni del testo a vari livelli di complessità: analisi morfologica e lemmatizzazione, analisi sintattica, interpretazione e disambiguazione semantica ecc.. I moduli di elaborazione sono solitamente interfacciati con *database* linguistici, che rappresentano e codificano grandi quantità di informazione terminologica e lessicale, morfologica, sintattica e semantica, che ne permettono sofisticate modalità di analisi.

Le analisi linguistiche forniscono l'*input* per i moduli di estrazione, acquisizione e strutturazione di conoscenza. La conoscenza estratta costituisce una risorsa per l'utente finale, e permette allo stesso di popolare ed estendere i repertori linguistico-lessicali e terminologici che sono usati in fase di analisi dei documenti. Si realizza, così, un ciclo virtuoso tra strumenti per il TAL e risorse linguistiche.



**Figura 1.3:**Un'architettura per l'estrazione di conoscenza dai testi basata sul TAL

Le risorse linguistiche lessicali e testuali permettono di costruire, ampliare, rendere operativi, valutare modelli, algoritmi, componenti e sistemi per il TAL, sistemi che sono, a loro volta, strumenti necessari per alimentare dinamicamente ed estendere tali risorse.

PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------



I differenti approcci al Natural Language Analysis possono essere raggruppati in due principali classi:

- *Knowledge Engineering*: codifica manuale di grammatiche e lessici da parte di esperti;
- *Machine Learning*: addestramento di modelli statistici su grandi quantità di dati, rappresentati da corpora annotati o meno.

Un modo di vedere questa dicotomia è nella metodologia: il primo approccio tende a

lavorare secondo una modalità top-down, imponendo al testo dei pattern grammaticali e

relazioni semantiche ben noti, mentre il secondo ha un modus operandi bottom-up,

ricercando pattern e associazioni da modellare, alcuni dei quali possono non corrispondere a delle proprie relazioni sintattiche e semantiche. Un altro modo di vedere tale distinzione è sulla base della gestione della complessità delle lingue, in particolare in merito al problema dell'ambiguità. Un approccio puramente simbolico, come il primo, deve risolvere l'ambiguità imponendo delle regole addizionali o fattori contestuali, che possono essere in qualche modo formalizzati. Questa è una metodologie basata sulla conoscenza, da momento che si affida a degli esperti per identificare e descrivere le regolarità del dominio. L'approccio empirico è più quantitativo, siccome tende ad associare delle probabilità alle diverse analisi testuali, e decide tra queste usando dei metodi statistici.

### 1.5 LA GERARCHIA DI CHOMSKY

Un *linguaggio* è definito come un insieme (anche infinito) di stringhe, ognuna costituita da una concatenazione di simboli terminali, chiamati talvolta parole. I *linguaggi formali* hanno definizioni matematiche rigorose, in questo si differenziano dai *linguaggi naturali*, come l'italiano e l'inglese, che non hanno ne una precisa definizione, ma sono caratterizzati da una vasta comunità di

PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------

parlanti. Una *grammatica* è un insieme finito di regole che specificano il linguaggio. I linguaggi formali, per definizione, sono sempre dotati di una chiara grammatica, mentre per quelli naturali non è così. I linguisti, però, si sforzano di scoprire le loro proprietà attraverso un processo di indagine scientifica, per poi codificarne i risultati in una grammatica. Il modo più comune per rappresentare la struttura grammaticale di una frase, ad esempio “Mary loves that person”, è di adoperare un albero, come illustrato in Figura 2. Il nodo S è quello radice dei nodi NP e VP, rispettivamente per la parte nominale e verbale della frase. VP è nodo padre dei nodi V e NP, rispettivamente verbo e nome. Ad ogni nodo foglia è associata una parola della frase da analizzare. Per realizzare l’albero di una frase, è necessario conoscere la struttura del linguaggio, così da servirsi di un insieme di regole per determinare quali strutture ad albero sono consentite. Tali regole, alla destra della Figura 2, determinano che un certo simbolo può essere espanso in un

albero di una sequenza di altri simboli (ad esempio,  $S \rightarrow NP VP$  significa che il nodo S può

generare a sua volta due nodi NP e VP). La struttura grammaticale aiuta a determinare il significato di una frase. Nella teoria dei linguaggi formali, le grammatiche sono rappresentate dalla quadrupla  $G = \langle V, T, P, S \rangle$ , dove V e T sono un insieme finito di simboli, P sono le regole grammaticali di generazione e S è il carattere del nodo radice. Rispettivamente V contiene tutti i simboli non terminali, in Figura 2 sono ad esempio S, NP, VP e simili, mentre T contiene i simboli terminali, ad esempio “Mary”, “loves” e simili.

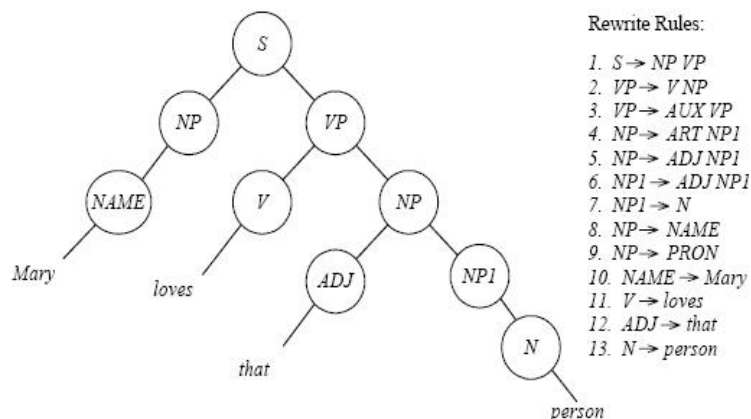


Figura 2 Rappresentazione ad albero di una frase e relativa grammatica

PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------



I formalismi grammaticali possono essere classificati in base alla loro *capacità generativa*, ovvero l'insieme dei linguaggi che possono rappresentare. Il linguista Chomsky descrive quattro classi di formalismi grammaticali, che differiscono solo per il formato delle regole di riscrittura. Le classi possono essere organizzate in una gerarchia, in cui ogni classe può essere utilizzata per descrivere tutti i linguaggi che appartengono ad una classe meno potente e alcuni linguaggi aggiuntivi. Salendo lungo la gerarchia aumenta il potere espressivo delle grammatiche, ma naturalmente gli algoritmi che le gestiscono sono meno efficienti. La gerarchia di Chomsky è composta dai seguenti livelli, vedi in Figura 3:

- Grammatiche di tipo 0 (illimitate o ricorsivamente enumerabili) include tutte le grammatiche dei linguaggi formali, e non hanno alcun tipo di restrizione nell'impostazione delle regole, ad eccezione che il termine a destra non sia nullo;
- Grammatiche di tipo 1 (dipendenti dal contesto) hanno regole della

forma  $\alpha A \beta \rightarrow \alpha \gamma \beta$ , con  $A$  simbolo non terminale e  $\alpha$ ,  $\beta$  e  $\gamma$  stringhe di simboli

terminali e non. Le stringhe  $\alpha$  e  $\beta$  possono essere vuote, ma la  $\gamma$  non deve essere vuota.

- Grammatiche di tipo 2 (libere dal contesto) sono definite da regole nella forma

$A \rightarrow \gamma$ , con  $A$  simbolo non terminale e  $\gamma$  una stringa di simboli terminali e non

terminali.

- Grammatiche di tipo 3 (regolari) restringe le sue regole ad un singolo simbolo non terminale nel lato sinistro della produzione e nel lato destro un singolo simbolo terminale, possibilmente seguito (o preceduto, ma non entrambe le forme nella stessa grammatica) da un singolo simbolo non terminale.

PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------

L'appartenenza di una classe di grammatica di tipo superiore in quella di tipo inferiore è un'inclusione propria, nel senso che esistono, ad esempio, linguaggi sensibili al contesto che sono non liberi dal contesto e linguaggi liberi dal contesto che sono non regolari. Si dimostra che le lingue naturali non sono regolari, e per la maggior parte delle lingue e delle costruzioni sia sufficiente una grammatica libera dal contesto. Tuttavia, esistono rari casi (un caso famoso è contenuto nel tedesco svizzero) che richiedono una grammatica dipendente dal contesto.

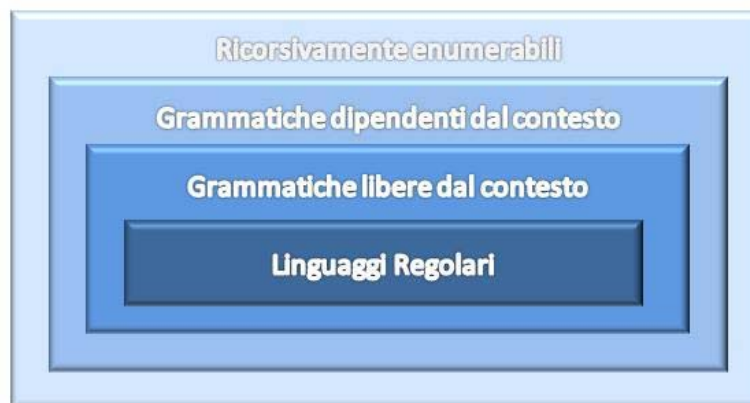


Figura 3 Diagramma di Venn dei linguaggi sulla base della gerarchia di Chomsky

### 1.6 METODOLOGIE E TECNICHE DEL NLP

Il Natural Language Processing (NLP) consiste in un range di metodi e tecniche computazionali, con fondamenti teorici, per analizzare e rappresentare testi in linguaggio naturale ad uno o più livelli di analisi linguistica allo scopo di ottenere elaborazioni di linguaggio human-like per un range di task ed applicazioni.

Il metodo più esplicativo per presentare ciò che in effetti avviene in un sistema di elaborazione di linguaggio naturale è tramite l'approccio dei "livelli di linguaggio", detto anche modello sincronico e che si distingue dal modello sequenziale il quale ipotizza che i modelli di elaborazione seguono l'un l'altro in modalità strettamente sequenziale; tale modello, invece, risulta più dinamico poiché i livelli interagiscono tra di loro in vario ordine. Di frequente si utilizzano le informazioni, che si ricavano da ciò che è tipicamente inteso

PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------



come livello superiore di elaborazione, per supporto nell'analisi di livello più basso. Per esempio, la conoscenza pragmatica che il documento che si sta leggendo riguarda la biologia sarà utilizzata quando si riscontra una specifica parola che ha più significati possibili, e la parola sarà interpretata come avente un significato che rientra nell'ambito della biologia.

I livelli del Natural Language Processing sono:

- *Fonologico*: questo livello ha a che fare con l'interpretazione del suono della pronuncia delle parole e tra le parole. Ci sono, infatti, tre tipi di regole utilizzate nell'analisi fonologica: 1) regole di fonetica: per il suono delle parole; 2) regole fonemiche: per le variazioni di pronuncia quando le parole sono pronunciate assieme; 3) regole prosodiche: per l'oscillazione dell'accento tonico e dell'intonazione in una sentenza
- *Morfologico*: questo livello ha a che fare con il fatto che le parole sono composte di morfemi, le più piccole unità di significato. Per esempio, la parola 'preregistrazione' può essere morfologicamente analizzata in tre morfemi distinti: il prefisso 'pre', la radice 'registra' ed il suffisso 'zione'. Poiché il significato di ogni morfema rimane invariato nelle parole, una persona può suddividere la parola nei suoi morfemi costituenti al fine di comprendere il suo significato. Allo stesso modo, un sistema NLP può riconoscere il significato da ogni morfema in modo da ottenerne e rappresentarne il significato; per esempio, in inglese, aggiungendo il suffisso 'ed' ad un verbo, si comunica che l'azione del verbo avviene nel passato. Questo è un pezzo chiave nel significato, ed infatti, di frequente si evince dal testo ciò dall'utilizzo del morfema 'ed'.
- *Lessicale*: a questo livello, le persone, così come i sistemi NLP, interpretano il significato di parole singole. Vari tipi di elaborazione contribuiscono alla comprensione a livello della parola, la prima di queste è l'assegnamento di un singolo tag di parte del discorso ad ogni parola; in tale elaborazione, alle parole che

PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------

possono fungere da più di una parte del discorso è assegnato il tag di parte del discorso più probabile in base al contesto in cui si trova la parola.

- **Sintattico:** questo livello si focalizza sull'analisi delle parole in una sentenza in modo da rivelare la struttura grammaticale della sentenza. Ciò richiede sia una grammatica sia un parser. L'output di questo livello di elaborazione è una rappresentazione della sentenza che evidenzia le relazioni di dipendenza strutturale tra le parole. La sintassi porta al significato in molti linguaggi poiché l'ordine e la dipendenza contribuiscono al significato.
- **Semantico:** questo è un livello al quale molte persone credono sia determinato il significato; comunque, come possiamo vedere nella definizione precedente di livelli, sono tutti i livelli a contribuire al significato. L'elaborazione semantica determina il possibile significato di una sentenza, concentrandosi sulle interazioni tra i significati delle parole nella sentenza. Questo livello di elaborazione può includere la disambiguazione semantica di parole con multipli significati; in modo analogo a come è realizzata la disambiguazione di parole che possono fungere da più parti del discorso a livello sintattico. Per esempio, tra i vari significati, 'file' come nome può intendere sia una cartella per memorizzare documenti o una linea di individui in una coda. Se è richiesta l'informazione dal resto della sentenza per la disambiguazione, il livello semantico, non lessicale, effettuerà la disambiguazione. Vari metodi possono essere implementati per realizzare la disambiguazione, alcuni dei quali richiedono informazioni sulla frequenza con la quale ogni senso si verifica in un particolare corpus di interesse, o alcuni dei quali richiedono la considerazione del contesto locale ed altri utilizzano la conoscenza pragmatica del dominio del documento
- **Discorso:** mentre la sintassi e la semantica lavorano con unità di lunghezza pari ad una sentenza, tale livello lavora invece su testi più lunghi, ovvero non interpreta più sentenze di testi semplicemente come sentenze concatenate, ognuna delle quali può essere interpretata singolarmente; piuttosto, si concentra sulle proprietà del

PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------

testo per intero che porta al significato facendo connessioni tra sentenze componenti. Vari tipi di elaborazione possono avvenire a questo livello, due dei più comuni sono la risoluzione di anafore ed il riconoscimento di struttura del testo/discorso. La risoluzione di anafore consiste nel sostituire le parole come i pronomi, che sono semanticamente poco rilevanti, con l'entità appropriata cui si riferiscono. Il riconoscimento della struttura del testo/discorso determina le funzioni delle sentenze nel testo, che, a turno, si aggiungono alla rappresentazione significativa del testo.

- *Pragmatico*: questo livello ha a che fare con l'utilizzo determinato del linguaggio in situazioni ed utilizza il contesto per la comprensione. Per esempio, le seguenti due sentenze richiedono la risoluzione del termine anaforico 'they' ma tale risoluzione richiede la pragmatica:

*"The city councilors refused the demonstrators a permit because they feared violence."*

*"The city councilors refused the demonstrators a permit because they advocated revolution."*

I sistemi NLP attuali tendono ad implementare moduli per realizzare principalmente i livelli più bassi di elaborazione, per vari motivi: prima di tutto l'applicazione può non richiedere l'interpretazione a livelli più alti; in secondo luogo, i livelli più bassi sono stati studiati ed implementati più accuratamente; in fine, i livelli più bassi hanno a che fare con unità di analisi più piccole, come morfemi, parole e sentenze, che sono governate da regole, a differenza dei livelli più alti di elaborazione del linguaggio, che invece hanno a che fare con testi e che sono solo governati da regolarità.

Qualsiasi applicazione che utilizza testo è un candidato per essere un'applicazione di NLP; le applicazioni più frequenti che utilizzano NLP includono le seguenti:

- *Information Retrieval*: è l'insieme delle tecniche utilizzate per il recupero mirato dell'informazione in formato elettronico. Per "informazione" si intendono tutti i documenti, i metadati, i file presenti all'interno di banche dati o nel world wide web;

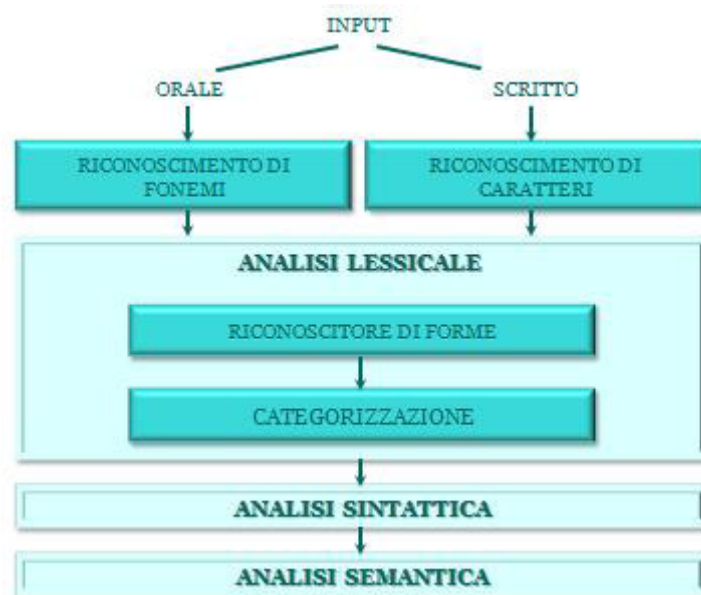
PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------



- *Information Extraction* : si focalizza sul riconoscimento, tagging ed estrazione in una rappresentazione strutturata di certi elementi di informazione chiave, come persone, locazioni, organizzazioni, da ampie collezioni di testo;
- *Question-Answering*: a differenza dell'Information Retrieval, che fornisce una lista di documenti potenzialmente rilevanti in risposta ad una query dell'utente, question-answering fornisce all'utente solo il testo della risposta o i passaggi che forniscono la risposta;
- *Sintesi*: i livelli più alti di NLP possono consentire un'implementazione che reduce un testo lungo in uno più breve, una rappresentazione narrativa abbreviata e significativo del documento originale;
- *Dialogue Systems*: attualmente utilizzano i livelli di linguaggio lessicale e fonetico, ma si ritiene che l'impiego di tutti i livelli sopra riportati offrano il potenziale per dialogue systems migliori.

Molte tecniche di Natural Language Processing, incluse lo stemming, il part-of-speech tagging, il riconoscimento di parole composte, decomposizione, word sense disambiguation e altre, sono utilizzate nell'Information Retrieval (IR). Molti altri task di IR utilizzano tecniche molto simili, come il clustering di documenti, il filtering, il rilevamento di link, ed esse possono essere combinate con NLP in maniera simile al document retrieval.

PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------



**Figura 1.4:** Schema a blocchi di un sistema per l'analisi del linguaggio naturale

Un sistema per l'analisi di input linguistici ha un'architettura, rappresentata secondo uno schema a blocchi in Figura 1.4, e si compone dei seguenti elementi:

- *Due sistemi di riconoscimento*, l'input può essere sia una produzione scritta sia orale, ma i sistemi che operano l'analisi possono lavorare indistintamente su ognuno di essi, a condizione che siano in una rappresentazione macchina interna che il calcolatore è in grado di manipolare. Il sistema che opera il riconoscimento dei fonemi prende il nome di Speech-to-Text System, e sarà oggetto di un

paragrafo nel capitolo delle tecnologie del parlato. Il sistema per la conversione dei grafemi in una rappresentazione macchina interna realizza uno scanning del documento cartaceo generando un file. Tale sistema è detto Optical Character

PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------

Recognitioner (OCR). L'OCR può basare la sua azione su una base di conoscenza che contiene tutti i possibili elementi tipografici per ogni simbolo della lingua naturale. Tale approccio diventa impraticabile nel caso del riconoscimento della grafia, in tal caso si passa ad un particolare metodologia di Pattern Recognition, detta a riconoscimento strutturale. Si considerare una prospettiva gerarchica, dove gli elementi da riconoscere vengono visti come composti da componenti più semplici, detti primitivi. Il riconoscimento di un campione è dato da tipo di primitivi che lo costituiscono e dalla relazione di composizione intercorrente.

- *Analisi lessicale*: ha il compito di riconoscere gli elementi lessicali, e assegnarvi informazioni in merito alla loro categoria grammaticale, risolvendo le ambiguità. Si compone di due sottosistemi:
  - *Riconoscitore di forme*: ha il compito di riconoscere le forme atomiche oggetto delle future elaborazioni. Si compone di un Tokenizer che compone la successione di caratteri in ingresso in unità linguistiche, ad esempio parole; e di uno Stemmer, che riconosce le possibili forme flesse di una unità linguistica e ne associa la forma radicale e le meta-informazioni di flessione;
  - *Categorizzazione, o Tagger*: associa ad ogni unità linguistica una delle possibili classi morfologico-sintattiche.

Gli ostacoli che si possono riscontrare in un'analisi lessicale sono vari. Nella Tokenizzazione, il problema è dato dalla non determinatezza dei delimitatori: essi dipendono fortemente dalla lingua adoperata nel testo e sono presenti irregolarità (unità atomiche composte da un insieme di parole, i.e Polirematiche). Inoltre, è possibile che un carattere di delimitazione non sia adoperato per delimitare parole (ad esempio il punto nelle sigle). Per lo stemmer è possibile che una forma flessa possa appartenere a varie possibili forme radicali. Nella classificazione, non è

PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------

univoca l'appartenenza di una unità linguistica ad una classe morfologico-sintattica.

I due sottosistemi sono rappresentati in figura collegati in serie, ma spesso è necessario un loro lavoro sinergico, dal momento che l'uno può aiutare a risolvere le ambiguità che ostacolano il lavoro dell'altro. Ad esempio l'ambiguità nell'appartenenza ad una forma flessa è risolvibile conoscendo la classificazione morfologica dell'unità. Ad esempio "porta" può essere sia la flessione del lemma sostantivo "porta", che di quello verbale "portare". Senza ulteriori informazioni il processo di disambiguazione sarebbe impossibile, ma con la conoscenza dell'appartenenza dell'unità al predicato verbale, è semplice operare l'associazione al lemma "portare".

- *Analisi sintattica*, o Parser: ha il compito di assegnare una caratterizzazione sintattica alla frase. Dato in ingresso una frase ed una grammatica, il compito del parser è determinare se la frase può essere generata dalla grammatica e, in caso affermativo, assegnare alla frase un'adeguata rappresentazione, detto albero di parsing. Un albero di parsing è un grafo aciclico etichettato, caratterizzato da: un nodo radice, detto Sentence (S), dei nodi foglia con le parole della frase e dei nodi intermedi, che rappresentano la struttura sintattica assegnata alla frase.
- *Analisi semantica*: ha il compito di eseguire un'analisi semantica del testo in ingresso, generando meaning representations. Si assegna a pezzi di struttura pezzi di significato. La struttura è composta da simboli e relazioni tra simboli che rappresentano stati del mondo.

### 1.6.1 STOPWORDS REMOVAL

E' un processo di trasformazione che elimina dai dati e dall'interrogazione le parole che non hanno un contenuto semantico utile per l'operazione di ricerca. Solitamente sono eliminate le congiunzioni, gli articoli e le parole molto frequenti:

- Le parole da eliminare sono scelte a priori da un esperto;

PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------



- Le parole da eliminare dipendono dalla lingua utilizzata.

L'eliminazione delle stopwords serve ad evitare che due frasi o documenti risultino simili perché contengono le stesse congiunzioni e gli stessi articoli. Eliminando le stopwords si dà maggior peso alle altre parole che, solitamente, hanno un maggiore significato semantico.

Ci sono però molti contro-esempi che mostrano come l'eliminazione delle stop word sia inefficace e controproducente, ad esempio:

1. *To be or not to be*
2. *New Year celebrations*
3. *Will and Grace*
4. *On the road again*

(Le parole in corsivo sono considerate stopwords)

Adattare la lista delle stopwords al dominio in esame può portare ad un miglioramento significativo dei risultati.

### 1.6.2 STEMMING

Gli stemmer sono analizzatori morfologici, che associano le forme flesse di un termine la sua forma radicale. La forma radicale può essere pensata come il lemma che si trova normalmente sui dizionari. I due metodi principali sono:

- 1) il linguistic/dictionary-based stemming;
- 2) il Porter-style stemming.

1) ha un'accuratezza di stemming più elevate, ma anche costi di implementazione e di elaborazione più elevate ed una copertura più bassa. 2) ha un'accuratezza minore, ma costi di implementazione e di elaborazione più bassi ed è solitamente sufficiente per l'IR.

Lo stemming mappa vari termini ad una forma base, la quale è poi utilizzata come termine nel modello a spazio vettoriale. Ciò vuol dire che, nella media, aumenta le similarità tra

PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------



documenti o documenti e query poiché avranno più termini comuni in seguito allo stemming, ma non prima.

Lo stemming ha un costo di elaborazione relativamente basso, specialmente quando si utilizza il Porter-style stemming, il quale riduce la dimensione dell'indice e solitamente migliora di poco i risultati, secondo [9]: 0.328 la precisione media senza stemming, 0.356 con lo stemming. Ciò lo rende molto appetito per l'utilizzo nell'IR.

Il vantaggio dello stemming che si è ritrovato tramite molte ricerche risulta in una sovrapposizione di casi positivi e negativi. L'inflexional stemming è per lo più vantaggioso, anche se vi sono casi ambigui nei quali lo stemming risulta contestabile. Per esempio, un utente probabilmente non sta cercando la parola 'window' quando il termine della sua query è 'Windows' (parte della casa vs. il sistema operativo). Il derivational stemming ha effetti misti: è più adatto a mappare la parola 'resignation' con la parola 'resign', e 'assassination' to 'assassin'. Ma molti mapping generate da un semplice stemmer sono errati o introducono ambiguità: 'expedition' è diverso da 'expedite'; 'importance' è diverso da 'import'; etc.

Come alternativa non-NLP allo stemming potrebbe essere usato il character n-grams, che consente un'elaborazione di documenti più semplice e indipendente dal linguaggio, ma a costo di una maggiore dimensione dell'indice. Complessivamente, comunque, i risultati dello stemming risultano confrontabili con quelli ottenuti utilizzando il character n-grams, e lo stemming risulta quindi preferito in quanto richiede meno memoria.

### 1.6.3 PART-OF-SPEECH TAGGING


Il Part-of-Speech Tagging ha il compito di assegnare una categoria sintattica ad ogni parola in un testo, risolvendo per cui alcune ambiguità, come visto prima. Le parole appartenenti ad una lingua naturale possono essere classificati in base ad un insieme di classi morfologiche, che costituiscono un insieme che prende il nome di *Tagset di Tagging*. Le parti del discorso possono essere categorizzate come classi chiuse, ovvero quelle in cui la condizione di appartenenza è relativamente fissa, ad esempio le proposizioni, e classi aperte, in cui è possibile di volta in volta trovare nuovi elementi, dovute a parole di recente conio.

Nei sistemi di elaborazione del linguaggio naturale esiste una fase chiamata Part-Of-Speech tagging o POS-tagging (in italiano la traduzione sarebbe "etichettatura delle parti

PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------

del discorso”). In questa fase si etichettano le parole individuate nella fase di analisi lessicale con il POS corrispondente, senza eseguire una vera e propria analisi sintattica, ma ricorrendo in genere ad informazioni statistiche (ad esempio il TnT tagger, basato su trigrammi) o a regole. Le etichette, o tags, sono reperite attraverso tagset, mappe da categorie lessicali in tags, che oltre a contenere i tag per le otto categorie lessicali di base, includono delle specializzazioni o raffinamenti, che distinguono delle sottocategorie, spesso in base al significato della parola. Nella tabella sottostante è riportato un estratto dal tagset di esempio, che rappresenta il primo corpus etichettato sintatticamente. Un esempio di POS-tagging è il seguente: data la frase book that flight (prenota quel volo), le etichettature possibili, considerando il tagset del Penn Treebank corpus, sono: book:NN that:WDT ight:NN oppure book:VB that:WDT ight:NN, per via dell' ambiguità della parola book. In questo caso l'etichettatura corretta è quella che assegna a book il tag VB.

Tag	Descrizione	Esempio	Tag	Descrizione	Esempio
CC	congiunzione	<i>and, or</i>	PP	pronome personale	<i>I, you</i>
CD	numero	<i>one</i>	PP\$	pronome possessivo	<i>my, your</i>
DT	articolo	<i>the, a</i>	RB	avverbio	<i>fully</i>
FW	parole straniere	<i>coup</i>	SYM	simbolo	<i>+, &amp;</i>
IN	preposizione	<i>of, by</i>	TO	to finale	<i>to</i>
JJ	aggettivo	<i>white, big</i>	UH	interiezione	<i>oops!</i>
JJR	aggettivo comparativo	<i>bigger</i>	VB	verbo, forma base	<i>eat</i>
JJS	superlativo	<i>biggest</i>	VBD	verbo, passato	<i>ate</i>
MD	verbo modale	<i>can</i>	VBG	verbo, -ing form	<i>eating</i>
NN	nome	<i>cat</i>	VBZ	verbo, 3ª persona pres.	<i>eats</i>
NNS	nome, plurale	<i>cats</i>	WDT	pronome determinativo	<i>which, that</i>
NNP	nome proprio	<i>George</i>	WP	pronome Wh-	<i>what, who</i>
POS	genitivo sassone	<i>'s</i>	WRB	avverbio Wh-	<i>how, where</i>

	<b>PARTNER: SECONDA UNIVERSITÀ DI NAPOLI</b>
	<b>RESPONSABILE PROF. BENIAMINO DI MARTINO</b> Technical Report: 2.4.1
<i>LC3 – Laboratorio pubblico-privato di ricerca      sul tema della Comunicazione delle Conoscenze Culturali</i>	<b>PAG 25 DI 39</b>

### Figura 1.5: Penn Treebank Part-of-Speech Tags

Gli algoritmi di tagging ricadono in tre gruppi differenti:

- *Rule-based tagger*, generalmente posseggono un grande database di regole di

determinazione della parte del discorso di una unità linguistica, ad esempio una unità che segue un articolo è un nome.

Un esempio è il tagger ENGTWOL, un analizzatore morfologico a due livelli:

1. Per primo viene consultato un dizionario dei termini, con la parte radicale delle unità linguistiche, il POS tag e alcune informazioni aggiuntive, e a tutte le unità della frase da analizzare vengono associate una o più etichette sulla base delle entry del dizionario;

<i>PROGETTO LC3</i>	<i>Revisione n*</i>	<i>0</i>	<i>Del</i>	<i>-----</i>
---------------------	---------------------	----------	------------	--------------

Word	POS	Additional POS features
smaller	ADJ	COMPARATIVE
entire	ADJ	ABSOLUTE ATTRIBUTIVE
fast	ADV	SUPERLATIVE
that	DET	CENTRAL DEMONSTRATIVE SG
all	DET	PREDETERMINER SG/PL QUANTIFIER
dog's	N	GENITIVE SG
furniture	N	NOMINATIVE SG NOINDEFDETERMINER
one-third	NUM	SG
she	PRON	PERSONAL FEMININE NOMINATIVE SG3
show	V	IMPERATIVE VFIN
show	V	PRESENT -SG3 VFIN
show	N	NOMINATIVE SG
shown	PCP2	SVOO SVO SV
occurred	PCP2	SV
occurred	V	PAST VFIN SV

**Figura 1.6:** Esempio di dizionario lessicale di ENGTWOL

- Un insieme di regole sono applicate per risolvere le ambiguità morfologiche, ovvero unità che presentano più di una etichetta.

```

Adverbial-that rule
Given input "that"
if
    (+1 A/ADV/QUANT); /* if next word is one of these */
    (+2 SENT-LIM); /* and following is a sentence boundary */
    (NOT -1 SVO/A); /* and previous word is not a verb like */
        /* consider (object complements) */
        /* "I consider that odd." */
then eliminate non-ADV tags
else eliminate ADV tag
    
```

**Figura 1.7:** Esempio di una regola di vincolo per ENGTWOL

- Stochastic tagger*, adoperano un corpus per determinare la probabilità che una data unità linguistica abbia un preciso tag morfologico in un preciso contesto: minimizza  $\{P(\text{unità} | \text{tag}) * P(\text{tag} | \text{precedenti } n \text{ tag})\}$ .

PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------

Un esempio di POS-tagger statistico è il tagger basato su bigrammi: data una parola ambigua  $w_i$  con un insieme  $J$  di possibili tag, il tagger seleziona come tag  $t_i$  quello più probabile considerando il tag precedente  $t_{i-1}$

$$t_i = \arg \max_{j \in J} P(t_j | t_{i-1}, w_i)$$

La precisione dei migliori pos-tagger statistici si aggira intorno al 95%, cosa che ha permesso, in alcune applicazioni, lo snellimento della fase di analisi sintattica, se non addirittura l'eliminazione della fase stessa (ad esempio, per determinare se un documento parla di sport o di economia, può essere sufficiente considerare solamente i nomi che compaiono nel documento, senza bisogno di analizzare sintatticamente le frasi che lo compongono).

Un particolare tipo di tagger stocastico è quello basato sul Hidden Markov Model (HMM), ed è sostanzialmente formato da tre componenti:

- Un lessico che elenca tutti i termini (sia come lemmi che come forme flesse) e tutte le possibili parti del discorso che un dato termine può assumere (es. “amo” può essere sia sostantivo che prima persona singolare del presente indicativo del verbo amare);
- Un corpus già annotato, detto “di apprendimento”, utilizzato per ricavare tutte le informazioni, sotto forma di frequenze delle transizioni tra i tag (uni/bi/trigrammi) e frequenze delle coppie parola-tag, necessarie al tagger per risolvere tutti i casi di ambiguità grammaticale che si presentano nella fase di annotazione dei testi;
- Il programma di annotazione vero e proprio, che implementa il modello di Markov e l'algoritmo di Viterbi, con l'ausilio delle opportune tecniche di ottimizzazione.

Il metodo di annotazione, sulla base di queste tre componenti, compie le seguenti operazioni:

- Suddivide il corpus da annotare in frasi; generalmente nessuna informazione relativa all'annotazione grammaticale viene trasferita da una frase all'altra,

PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------

quindi il problema dell'annotazione di testi si può ridurre al problema dell'annotazione delle frasi che li compongono. I segni di interpunzione “.”, “!” e “?” fungono da separatori;

- La frase viene suddivisa in token o parole, che rappresenteranno l'unità alla quale verrà assegnata la parte del discorso;
- Utilizzando il lessico associa a ogni token tutti i tag possibili (es. al token “amo” si associano i tag “verbo” e “nome”);
- Applica l'algoritmo di Viterbi, utilizzando le probabilità ricavate dal *corpus* di apprendimento, per risolvere le ambiguità e assegnare a ogni *token* il *tag* che risulta più probabile.

- *Transformation-based tagger*, è un approccio ibrido, e come gli algoritmi rule-based

ha un insieme di regole per l'assegnazione dei tag alle unità linguistiche, ma ha anche una componente statistica: le regole non sono inserite da un esperto, ma computate a partire da un corpus appositamente annotato. Un esempio è il *Brill tagger*, che opera seguendo questi passi:

1. Ad ogni unità linguistica si applica il tag più probabile, con tale probabilità costruita a partire da un corpus di training annotato considerando solo le singole unità linguistiche;
2. Successivamente si applicano delle regole di trasformazione, apprese dall'osservazione del training e considerando il contesto all'interno di una frase, per correggere i tag erroneamente assegnati alle unità.

PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------

Per poter effettuare la lemmatizzazione e la lemmatizzazione semantica è necessario fare ricorso ad alcune risorse utilizzate nel campo dell'elaborazione del linguaggio naturale. In particolare, il Part-Of-Speech tagger permette di stabilire la categoria delle parole in modo da rendere possibile la ricerca della parola in un apposito thesauro, per estrarre i sinonimi, o in una ontologia, per estrarre una rappresentazione semantica della parola e le sue relazioni con altri concetti. Queste relazioni, infine, vengono utilizzate sia per effettuare alcuni tipi di elaborazioni sul testo come l'espansione delle query, sia nell'ambito di sistemi di disambiguazione non supervisionati, come il disambiguatore semantico basato su densità concettuale.

#### 1.6.4 FRASI STATISTICHE E COMPOSTE

Tale tecnica effettua un'indicizzazione di unità composte da più token piuttosto che singoli token. Consiste nell'accoppiare parole adiacenti, che non rientrino nella categoria stopwords, e quindi utilizzare le coppie con una frequenza superiore ad una certa soglia.

In pratica, si utilizza un misto tra unità a singolo token ed unità a più token: i singoli token, da soli, comportano il match in documenti che non dovrebbero (per esempio, rilevano il match tra 'New' e 'New York'); mentre utilizzare solo unità a più token comporta un'elevata penalità per piccole variazioni (per esempio, in documenti contenenti 'James T. Kirk' non vi sarà il match se la query richiede 'James Kirk'). Aggiungere, invece, al vettore sia unità a singolo token sia unità a più token, allevia tale problema.

Una possibile soluzione potrebbe essere nel verificare se la query è:

per token singoli, non un composto ('York' non dovrebbe ritornare 'New York')

per token singoli, da soli e come composto ('Nobel' dovrebbe ritornare 'Nobel Prize')

per un composto, dove parti del composto possono essere trovate separatamente ('natural language' può tornare 'language', no 'natural'; 'wine stores' può tornare 'wine', no 'stores')

per un composto, non singoli token ('New York' non dovrebbe tornare 'New' o 'York')

Se possono essere distinti i casi riportati sopra, allora vi è un netto vantaggio nell'utilizzo dei composti.

PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------



### 1.6.5 PARSER

Un processo di parsing può essere visto con un algoritmo di ricerca del corretto albero sintattico per una data frase, all'interno dello spazio di tutti i possibili alberi sintattici generabili a partire dalle regole di una grammatica. I parametri che vanno dati al processo di definizione dell'albero sono:

1. le regole grammaticali, che predicano come da un nodo radice S ci siano solo alcune vie di scomposizione possibili per ottenere i nodi terminali;
2. le parole della frase, che ricordano come la (s)composizione di S debba terminare.

I due principali approcci al parsing sono:

- *Top-down o goal-driven approach*, cerca il corretto albero applicando le regole

grammaticali a partire dal nodo radice S, provando a raggiungere i nodi foglia;

- *Bottom-up o data-driven approach*, si inizia con le parole che compongono la frase

di input, da cui si inizia ad applicare le regole grammaticali fino a poter arrivare al nodo radice S.

PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------



La strategia top-down non perde tempo esplorando alberi che non portino a S come nodo radice, cosa che invece si verifica con la strategia bottom.up. Il top-down, però, genera un grande indieme di alberi S-rooted che sono inconsistenti con l'ingresso fornito, dal momento che gli alberi sono generati senza esaminare l'input linguistico. Bottom-up non produce mai alberi inconsistenti con l'input linguistico.

Quando in un nodo dell'albero sintattico si applicano delle regole grammaticali, si possono generare un insieme di percorsi alternativi verso uno o più nodi. Tale ramificazione non è espandibile in parallelo, ma va considerato un percorso per volta. Per questo l'esplorazione è fatta secondo due distinte strategie:

- *Depth-first*, la ricerca procede espandendo sempre il primo nodo generato, e

operando un backtracking nel caso il percorso non fosse giusto;

PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------



- *Breadth-first*, la ricerca procede espandendo prima tutti nodi di un livello, per poi

scendere al livello successivo.

Ci sono molti modi di combinare previsioni top-down con dati bottom-up per ottenere

ricerche più efficienti. La maggior parte usano un tipo come meccanismo di controllo per la generazione degli alberi, e l'altro come filtro per scartare a priori alberi che certamente non sono corretti, un esempio è l'*algoritmo del Left Corner*. L'idea alla base di quest'algoritmo

è di combinare una strategia di generazione degli alberi di tipo Top-down, con il filtraggio

con considerazioni di natura Bottom-up. L'algoritmo si memorizza la prima parola dell'input

(left corner), e non si devono considerare le regole grammaticali in cui sul ramo sinistro dell

Questi approcci di parsing,, soprattutto se si considera una strategia del tipo Depth -first,

possono incorrere in situazioni di stallo, senza mai giungere ad un risultato. Ciò si verifica

PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------

quando la grammatica in ingresso al parser è di tipo left-recursive. Una grammatica si

dice *ricorsiva a sinistra*, se ammette una regola del tipo  $A \rightarrow^* A\alpha$  (es.  $NP \rightarrow NP PP$ ), ovvero se contiene un simbolo non terminale che è sia parte della condizione di innesco che prodotto di una regola grammaticale. Esistono due modi per poter ovviare a questo inconveniente:

- riformulare le regole che presentano ricorsione a sinistra, ottenendo così quella che prende il nome di *weakly equivalent grammar*;
- Gestire esplicitamente il processo di esplorazione, evitando situazioni di stallo.

Nell'ottica del secondo approccio si inserisce la *programmazione dinamica*, una metodologia di parsing in cui si memorizzano i risultati intermedi con l'intento di non ripetere il lavoro già fatto che si può evitare e non cadere nella ricorsione a sinistra. I risultati intermedi vengono memorizzati in una struttura che prende il nome di *chart*. Un chart è un grafo aciclico etichettato, dove un arco contiene l'indicazione dei nodi iniziali e terminali e della regola che deve venir applicata.

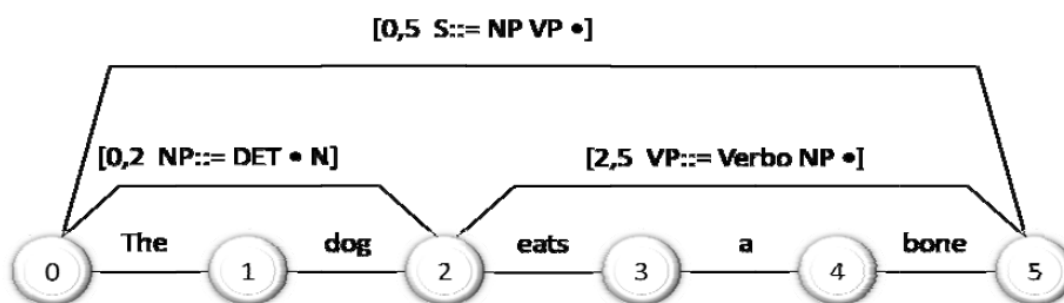


Figura 1.8: Esempio di chart

L'*algoritmo di Earley* è un esempio di programmazione dinamica che opera su un *chart* per realizzare un task di parsing. L'algoritmo inizia con una fase di inizializzazione applicando

PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------



la regola  $\gamma \rightarrow S$ . Ad ogni passo dell'algoritmo, uno dei seguenti tre operatori viene applicato ad ogni nodo del chart, in funzione del suo stato:

- *Previsione* (Predictor): crea nuovi stati nell'entrata corrente del chart,

rappresentando le aspettative top-down della grammatica; verrà quindi creato un

numero di stati uguale alle possibilità di espansione di ogni nodo non terminalee nella grammatica;

- *Scansione* (Scanner): verifica se nell'input esiste, nella posizione adeguata, una parola la cui categoria combacia con quella prevista dallo stato a cui la regola si trova. Se il confronto è positivo, la scansione produce un nuovo stato in cui l'indice di posizione viene spostato dopo la parola riconosciuta. Tale stato verrà aggiunto all'entrata successiva del chart;
- *Completamento* (Completer): quando l'indicatore di posizione raggiunge l'estrema destra della regola, questa procedura riconosce che un sintagma significativo è stato riconosciuto e verifica se l'avvenuto riconoscimento è utile per completare qualche altra regola rimasta in attesa di quella categoria.

Una volta costruito il chart, è possibile ottenere un albero di parsing, estraendo l'arco o l'insieme di archi che dal primo nodo portano all'ultimo.

Anche per il parsing esistono degli *approcci statistici*, dove si scelgono le regole da espandere in base a probabilità calcolate a partire da un corpus, per arrivare il prima possibile ad un'analisi e restituirla come "più probabile". Dato un insieme di regole, fornite sia da esperti che definite a partire da un'analisi empirica di un insieme di testi, si definisce la probabilità di applicazione della regola come:

PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------

$$P(A \rightarrow B | B) = \frac{\text{Count}(A \rightarrow B)}{\text{Count}(A)}$$

vale a dire, la probabilità nota la parte di innesto è pari al numero di volte in cui la regola è applicata nel corpus, diviso il numero di occorrenze della parte di innesco.

### 1.6.6 AUGMENTED TRANSITION NETWORK

Una Augmented Transition Network (ATN) è una tipologia di grafo usato per la definizione operativa dei [linguaggi formali](#), specialmente per quanto riguarda il parsing di [linguaggi naturali](#) relativamente complessi, ha ampia applicazione in intelligenza artificiale. Una ATN può, in teoria, analizzare la struttura di qualunque frase, anche se complicata.

Le ATN sono costruite sull'idea di utilizzare [macchine a stati finiti](#) per effettuare il parsing di parole. "Transition Network Grammars for Natural Language Analysis" indica che aggiungendo il meccanismo di ricorsione ad un modello a stati finiti, è possibile eseguire il parsing in maniera più efficiente. E' realizzato un insieme di grafi di transizione invece di costruire un automa per una particolare frase. Una frase sintatticamente corretta è parsificata raggiungendo uno stato finale per ogni grafo. Le transizioni fra questi grafi sono semplici chiamate a funzioni da uno stato a uno stato iniziale di qualunque grafo nella rete. Si stabilisce che una frase è sintatticamente corretta se uno stato finale è raggiunto dall'ultima parola della frase.

Questo modello raggiunge molti obiettivi della natura del linguaggio in quanto cattura le regolarità del linguaggio. Ovvero, se è presente un processo che opera su diversi ambienti, la grammatica può incapsulare il processo in una singola struttura. Tale incapsulazione non semplifica solo la grammatica, ma ha il valore aggiunto dell'efficienza dell'operazione. Ulteriore vantaggio di tale modello si ha nell'abilità di rinviare la presa di decisioni. Molte grammatiche effettuano delle ipotesi in presenza di ambiguità; ciò significa che non si conosce al momento abbastanza sulla frase. Attraverso l'uso della ricorsione, le ATN risolvono tale inefficienza rinviando le decisioni sino a quando non si conosce abbastanza sulla frase.

Una Grammatica molto semplice, quindi, può essere rappresentata, graficamente, con un Diagramma di Transizione fra gli Stati (TN) che rappresenta un metodo riconoscitivo per la definizione dei linguaggi ossia permettono di definire come riconoscere tutte e sole le stringhe che appartengono ad un linguaggio.

Le TN semplicemente sono automi a stati finiti che permettono di fare analisi lessicali o di riconoscere frasi semplici, ma non di riconoscere linguaggi più complessi.

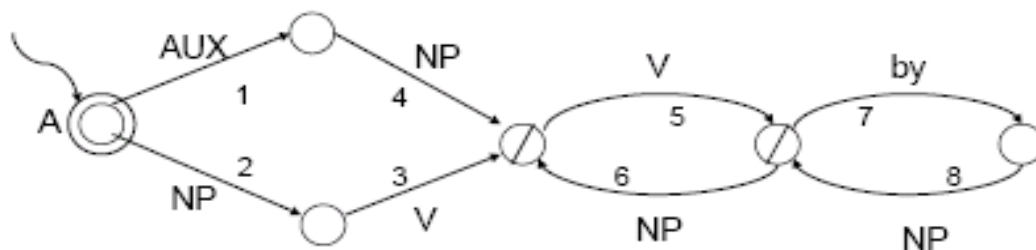
PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------

Una importante categoria delle Transition Network è rappresentata dalle Augmented Transition Network(o ATN). Una ATN è un diagramma di transizione fra gli stati, ai cui archi sono associati una *label* e una *condizione*:

- Le *label* possono denotare categorie di parole oppure altre ATN: si tratta quindi di diagrammi ricorsivi.
- Le *condizioni* devono essere soddisfatte perché l'arco possa essere attraversato.

Un insieme di *registri* permette di memorizzare risultati intermedi oppure lo stato dell'esplorazione dell'ATN.

Un esempio per una più chiara comprensione degli ATN, può essere quello di riconoscere la frase "The rice was eaten by the cat"



### Tabella condizioni-azioni

PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------

	test	action
1	T	SETR V*
		SETR TYPE'QUESTION
2	T	SETR SUBJ*
		SETR TYPE'DECLARATIVE
3	agrees SUBJ*	SETR V*
4	agrees* V	SETR SUBJ*
5	AND(GETF PPRT)(=V'BE)	SETR OBJ SUBJ
		SETR V*
		SETR AGFLAG TRUE
6	TRANS V	SETR SUBJ'SOMEONE
7	AGFLAG	SETR OBJ*
		SETR AGFLAG FALSE
8	T	SETR SUBJ*

- **Iniziamo dallo start node A.**

Ci sono due cammini possibili; ma poiché la prima parte della frase ('the rice') è un NP, si sceglie l'arco 2. Prima di attraversarlo, però, si controlla se ci sono altre condizioni o azioni da soddisfare (vedi tabella).

'T' significa che non ci sono altri test.

Si esegue l'azione che consiste nel memorizzare 'rice' nel registro

SUBJECT e TYPE come 'declarative'.

- **Si passa all'esame di 'was'**

E' un verbo: quindi si attraversa l'arco 3. Il test indica di controllare che questo verbo si accordi con il soggetto. Se questa condizione è verificata, si memorizza 'was' nel registro VERB.

- **Siamo ora al nodo D.**

'Eaten' è un verbo; il test sull'arco 5 indica di controllare che questo verbo si accordi con l'ausiliario.

Si compiono ora 4 azioni: il contenuto del registro SUBJ ('rice') viene trasferito nel registro OBJ. Il contenuto del registro VERB viene sostituito con 'was eaten'. Un flag indica che la frase è in forma passiva. Il registro SUBJ viene settato al valore 'someone'.

- **By:** si sceglie l'arco 7.

PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------



- **'The cat'**: un'altra 'noun phrase': si traversa l'arco 8 e si memorizza il frammento di frase in SUBJECT.
- **Il processo termina.**

### 1.6.7 WORD SENSE DISAMBIGUATION

La Word Sense Disambiguation ha il compito di distinguere il corretto senso di una parola in un contesto. Quando è utilizzata nell'IR, i termini sono rimpiazzati da i loro significati nel vettore del documento.

Un fattore negativo nell'utilizzo di un'ontologia general-purpose è che la word sense disambiguation per query brevi risulta difficile a causa della mancanza di contesto, mentre risulta non necessaria per query lunghe in quanto gli altri termini contribuiscono comunque a restringere la ricerca.

PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------



## 2 BIBLIOGRAFIA

- [1] M. Makins, editor. Collins English Dictionary, 3rd Edition. Harper Collins, 1991.
- [2] G. Gazdar. Paradigm merger in natural language processing. In R. Milner and I. Wand, editors, Computing Tomorrow: Future Research Directions in Computer Science, pages 88--109. Cambridge University Press, 1996.
- [3] H. Thompson. Natural language processing: a critical analysis of the structure of the field, with some implications for parsing. In K. Sparck-Jones and Y. Wilks, editors, Automatic Natural Language Parsing. Ellis Horwood, 1985.
- [5] Bartolini R., Lenci A., Montemagni S., Pirrelli, V., Soria C.: *Semantic Mark-up of Italian Legal Texts through NLP-based Techniques*. Proceedings of LREC 2004, Lisbona, Portugal, 2004.
- [6] Basili R., Catizone R., Pazienza M-T., Stevenson M., Velardi P., Vindigni M., Wilks Y.: *An Empirical Approach to Lexical Tuning*. Proceedings of the LREC1998 Workshop on Adapting Lexical and Corpus Resources to Sublanguages and Applications, Granada, Spain, 1998.
- [7] Busa F., Calzolari N., Lenci A., Pustejovsky J.: *Building a Semantic Lexicon: Structuring and Generating Concepts*. In Bunt H., Muskens R., Thijsse E. (eds.): Computing Meaning Vol. II. Kluwer, Dordrecht, 2001.
- [9] Tomek Strzalkowski, Barbara Vauthey: [Information Retrieval Using Robust Natural Language Processing](#) Meeting of the Association for Computational Linguistics

PROGETTO LC3	Revisione n*	0	Del	-----
--------------	--------------	---	-----	-------