
UNIVERSITÀ DI PISA

## Corso di Laurea in Informatica Umanistica

**RELAZIONE**

# Esperimenti di adattamento al dominio specialistico di un Part of Speech Tagger stocastico

**Candidato:** *Marco Lacaria*

**Relatore:** *Prof. Alessandro Lenci*

**Anno Accademico 2007-2008**

# Sommario

# Introduzione

La seguente tesi può essere suddivisa in due parti: la prima composta dal capitolo 1, la seconda composta dai capitoli 2, 3, 4, e 5.

Obiettivo di questa tesi è quello di fornire, nella sua prima parte, un quadro generale di quali siano e come funzionino nell'ambito del Natural Language Understanding le principali architetture dei sistemi Part of Speech Taggers sviluppati per l'analisi del linguaggio naturale da parte della macchina. Nella seconda parte verrà prima introdotto e descritto un sistema per il trattamento automatico del linguaggio naturale sviluppato presso l'Istituto di Linguistica Computazionale del CNR di Pisa, dopo di che ne verranno valutate le prestazioni misurandone il grado di adattabilità su diversi domini linguistici, focalizzando l'attenzione sul processo di Tagging.

Nel Capitolo 1 viene descritto a livello introduttivo cosa sia il "Natural Language Processing", il sottocampo della Linguistica Computazionale che si occupa dell'elaborazione del linguaggio naturale sia da un punto di vista di "analisi del linguaggio" (Natural Language Understanding) sia da un punto di vista "produzione autonoma del linguaggio" (Natural Language Generation); si affronta poi il problema del Natural Language Understanding attraverso la descrizione dei processi che compongono l'architettura di un generico sistema per il riconoscimento automatico del linguaggio naturale.

Viene trattato poi il task dell'annotazione linguistica Part of Speech Tagging, tappa fondamentale per qualsiasi sistema di elaborazione automatica del linguaggio naturale, definendo in cosa tale processo consista , quali siano a livello linguistico le principali questioni di cui si occupa e quali siano i campi di applicazione.

Poiché il processo di Tagging è un compito che può essere svolto automaticamente da una macchina a tal proposito vengono descritti, facendone un'analisi ed una comparazione, i tipi di Part of Speech Taggers più utilizzati oggi per il task dell'annotazione automatica, dividendo le categorie in due macro divisioni: Taggers basati su regole manuali e Taggers basati su algoritmi di machine learning.

A fine di questo capitolo vengono tratte le conclusioni su quale sia ad oggi tra le due

tipologie di Taggers quella che da i migliori risultai.

Nel Capitolo 2 viene introdotto SmarText, il sistema per il "Trattamento Automatico del Linguaggio Naturale" sviluppato presso l'istituto di Linguistica Computazionale del CNR di Pisa: dopo aver dato un prima panoramica di insieme della sua struttura si passa ad esaminare in dettaglio i vari livelli, o processi, che lo compongono, illustrando ed analizzando rispettivamente struttura interna, output prodotti e funzionamento del modulo Tokenizzatore deputato alla segmentazione del testo, del modulo Analizzatore Morfologico deputato all'analisi morfologica delle unità componenti il testo ed infine del modulo Part of Speech Tagger deputato all'assegnazione della corretta categoria grammaticale a ciascuna unità componente il testo.

I Capitoli 3, 4 e 5 riguardano la valutazione vera e propria del sistema SmarText.

Per verificare e valutare le prestazioni di SmarText è stato necessario:

- Analizzare le uscite del modulo "Part of Speech Tagger" , al fine di capire se, perché e a quale dei tre livelli il sistema in determinati punti sbagli l'analisi automatica di un testo passatogli come input.
- Correggere i potenziali errori del processo di annotazione automatica, cercando di capire quale sia la modifica più appropriata da apportare tenendo conto della natura dell'errore.
- Effettuare un primo Test di confronto tra la porzione di testo analizzata e corretta, e la medesima porzione di testo presa allo stato precedente il lavoro di analisi e correzione, al fine di avere una stima del numero di errori compiuti dal sistema.
- Analizzare almeno 20.000 unità, così da poter considerare la porzione di testo analizzata un piccolo corpus di training con il quale effettuare un nuovo ciclo di addestramento del modulo PoS Tagger (Tagger Stocastico) al fine di migliorarne le prestazioni.
- Effettuare nuovi test per valutare il sistema dopo il ciclo di Training.

Nel Capitolo 3 vengono categorizzati e passati in rassegna quelli che sono stati considerati gli errori più rilevanti compiuti da ciascun modulo di SmarText, (Tokenizzatore, Analizzatore Morfologico e Tagger) nel processo di elaborazione del testo, motivando le possibili cause dell'errore, le potenziali conseguenze e la correzione più appropriata da

apportare.

Nel Capitolo 4 si descrivono le modalità secondo le quali il sistema è stato valutato, riportando, confrontando e discutendo i risultati ottenuti durante i vari cicli di test effettuati. Viene inoltre fatta una breve analisi linguistica (e sociolinguistica) delle tipologie dei domini linguistici caratterizzanti i testi sui quali il sistema è stato valutato.

Nel Capitolo 5 si traggono le conclusioni per quanto riguarda il lavoro svolto su SmarText.

# 1 Natural Language Processing e Part of Speech Tagging

## 1.1 Natural Language Processing: un'introduzione

Il Natural Language Processing, o Trattamento Automatico del Linguaggio, è un sottocampo della linguistica computazionale e dell'intelligenza artificiale.
Le applicazioni sviluppate per il NLP hanno lo scopo pratico di rendere possibile il "dialogo" uomo-macchina (inteso come interazione).
L'elaborazione del linguaggio naturale si divide in due filoni principali, l'uno l'opposto dell'altro:

1. Natural Language Understanding, ovvero la "comprensione" del linguaggio da parte della macchina, al fine di generare una rappresentazione astratta dell'analisi di una frase presa in input.
2. Natural Language Generation, ovvero la "produzione autonoma" da parte della macchina di frasi di senso compiuto data la grammatica di una lingua naturale[1].

Un sistema per il trattamento automatico del linguaggio sviluppato presso il Dipartimento di Linguistica Computazionale del CNR di Pisa verrà descritto nei prossimi capitoli di questa tesi.

## 1.2 Il Natural Language Understanding

Secondo il "modello del messaggio" quando un parlante A vuole comunicare qualcosa ad un destinatario B, A codifica il messaggio M secondo le regole linguistiche della propria lingua, dopodiché trasmette il messaggio a B (tramite canale vocale o scritto). Non appena il destinatario B inizia a ricevere il messaggio M, si attiverà in lui un processo di decodifica che sequenzialmente identificherà i suoni (se il messaggio è vocale), la parole, le categorie grammaticali di esse, i rispettivi legami sintattici ed il loro significato. Il destinatario B decodifica e comprende il messaggio M, come descritto in FIGURA 1.1.

1 Questo secondo metodo di studio non sarà approfondito poiché non rilevante ai fini della mia tesi.

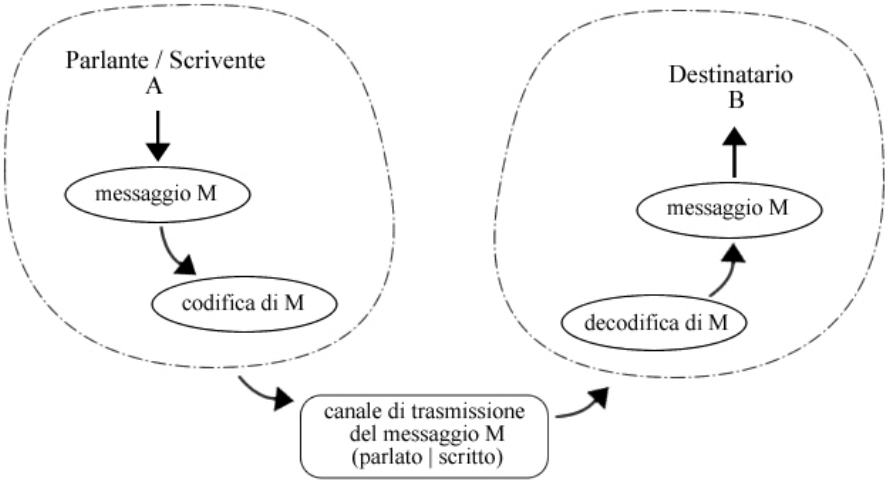


*FIGURA 1.1 – Modello del messaggio nella comunicazione linguistica*

Tra due esseri umani "la comunicazione linguistica avviene facilmente, ma non si spiega altrettanto facilmente" (Akmajian, 1984); per una persona il passaggio dal linguaggio all'informazione è immediato, ma per una macchina che si trovi a dover gestire un input linguistico, come ad esempio una frase codificata in linguaggio naturale, il passaggio dal linguaggio all'informazione è un processo molto complesso, tutt'altro che immediato.

I programmi di Natural Language Understanding (o Natural Language Analysis) devono riuscire a decodificare e rappresentare il contenuto di un input linguistico in maniera che poi altri "agenti intelligenti" possano lavorarci sopra.

Un tipico sistema per l'analisi e la codifica del linguaggio naturale è rappresentato in FIGURA 1.2.

Per arrivare all'informazione il sistema cerca di simulare quelle che sono le operazioni effettuate da un essere umano, attraverso i seguenti passi:

1. Conversione dell'input in formato "machine readable": l'input può presentarsi sia sotto forma di segnale acustico, sia sotto forma di testo non digitalizzato, sia sotto forma di testo già digitalizzato.

   Nel primo caso, input vocale, un apposito software di "Speech to Text" deputato al riconoscimento dei fonemi converte l'input in formato digitale.

   Nel secondo caso, input cartaceo, il testo viene convertito in formato digitale, e

quindi leggibile dalla macchina, da un apposito software “optical character reader” in grado di riconoscere e gestire i grafemi.

2. Tokenizzazione del testo: una volta reso il testo “machine readable”, viene effettuata la segmentazione delle parole in esso contenute.
3. Analisi Lessicale: l'analisi lessicale permette di determinare la struttura morfologica e la categoria grammaticale delle parole. Questo compito si suddivide in due fasi:
   - in una prima fase ciascuna parola viene analizzata morfologicamente e ricollegata al proprio lemma, sulla base di un dizionario.
   - in una seconda fase uno specifico programma chiamato “Part of Speech Tagger” associa a ciascuna parola la rispettiva categoria grammaticale. Tale fase prende il nome di “tagging morfosintattico.”

     Il “tagging morfosintattico” ed i “Part of Speech Taggers” saranno approfonditi nei prossimi paragrafi di questa tesi.
4. Parsing sintattico: sulla base di una grammatica viene assegnata a ciascuna frase la rispettiva analisi sintattica, cercando di identificare le dipendenze sintattiche tra le varie parole.
5. Analisi semantica: sulla base di un lessico computazionale semantico vengono associati alle parole i rispettivi ruoli e sensi semantici.

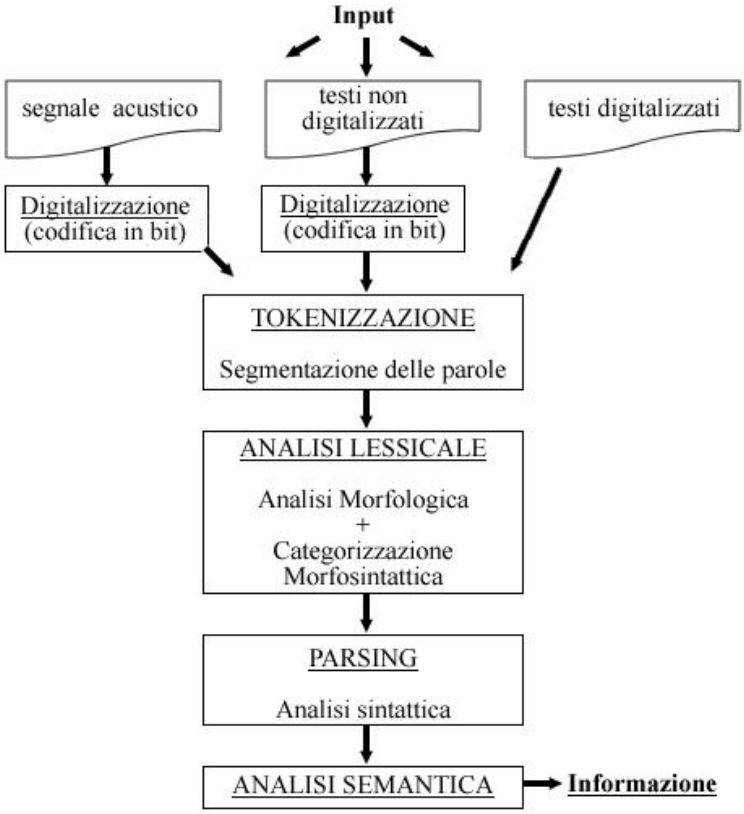


*FIGURA 1.2 – Architettura di un sistema per il riconoscimento di un linguaggio naturale*

## 1.3 L'annotazione Part of Speech Tagging

Nell'ambito del Natural Language Processing il Part of Speech Tagging, o Tagging, è un tipo di annotazione linguistica che assegna a ciascuna parola presente in un testo un **codice** che ne espliciti la rispettiva categoria morfosintattica.

Questo processo di annotazione è una tappa fondamentale per qualsiasi sistema di "Natural Language Processing", poiché molte delle funzionalità di tali sistemi richiedono come base di partenza una classificazione grammaticale delle parole presenti in un testo.

- Nel "parsing" conoscere la corretta categoria grammaticale di una parola è fondamentale per la costruzione della struttura sintattica della frase.
- Nella "analisi semantica" sapere, per esempio, se una parola sia un verbo o un nome permette di determinarne il ruolo semantico.
- Nell' "information extraction" è possibile estrarre l'informazione impostando i

parametri di ricerca su una data classe (ricercare un azione in base alla classe “verbo”, un luogo in base alla classe “nome”).

- Nello “stemming” è possibile generare le forme flesse di una parola conoscendone la categoria grammaticale, quindi effettuare ricerche per forme flesse.

Al giorno d'oggi l'annotazione Part of Speech è un lavoro che può essere svolto automaticamente da un computer con un alto grado di precisione poiché la corretta parte del discorso di una parola è altamente predicibile dal contesto circostante la parola stessa.

## 1.4 Definizione del codice di annotazione: il Tagset

La parte di codice associata a ciascuna parola prende il nome di Tag e l'insieme di tutte le Tags definite costituiscono i Tagset, i quali si dividono principalmente in due categorie:

- Tagset semplici: contengono un numero limitato di Tags. Sono molto generici e la precisione con cui descrivono un testo è piuttosto bassa.

  Generalmente i Tagset semplici si limitano a distinguere solo le principali categorie grammaticali come nome, verbo, articolo, aggettivo, preposizione, avverbio, congiunzione e pronome[2] .

  Questi vengono impiegati per costruire sistemi il cui scopo sia la semplice classificazione di parole.
- Tagset complessi: contengono un numero di Tags molto elevato, sufficiente a descrivere qualsiasi categoria grammaticale presente in un testo.

  I Tagset complessi non si limitano a descrivere le parole di un testo attraverso la semplice categoria grammaticale, ma associano ad esse anche i rispettivi tratti morfologici. Ciò permette di poter fare distinzione tre le categorie grammaticali stesse; per esempio, alla categoria grammaticale “nome” possono essere associati i tratti di numero (singolare, plurale) e di genere (maschile, femminile), così come alla categoria di aggettivo. Alla categoria “verbo” possono essere associati i tratti che ne specifichino la forma (congiuntivo, indicativo [...] gerundio), il tempo (presente [...] imperfetto), la persona (prima, seconda, terza) ecc.. Così per le restanti categorie grammaticali.

2 Ci sono addirittura Tagset che contengono soltanto le Tags “nome”, “verbo”, “altro”. “Nome” e “Verbo” sono le due categorie grammaticali presenti in tutte le lingue esistenti.

Questi Tagset sono generalmente utilizzati per estrarre la maggiore informazione possibile dai contesti.

Più Tags sono state definite, più un Tagset sarà ricco e preciso nel descrivere le differenti categorie grammaticali. Paradossalmente, più è granulare la distinzione tra le varie Tags maggiore sarà la possibilità di incorrere in errore, poiché per un sistema di annotazione automatica sarà più difficile applicarle[3].
Quando si definisce un Tagset, la scelta del codice deve essere esplicita e strutturata in maniera omogenea per renderne la lettura e l'analisi più semplice possibile. Ovviamente la definizione di un Tagset dipende anche dal tipo di lingua che esso andrà a descrivere.
I Tagset per l'italiano sono molto complessi e presentano un gran numero di Tags.

## 1.5 I Part of Speech Taggers

Il compito di Tagging oggi viene svolto automaticamente dai Part of Speech Taggers, o più semplicemente Taggers, sistemi che, utilizzando un Tagset, annotano automaticamente il testo passatogli come input.

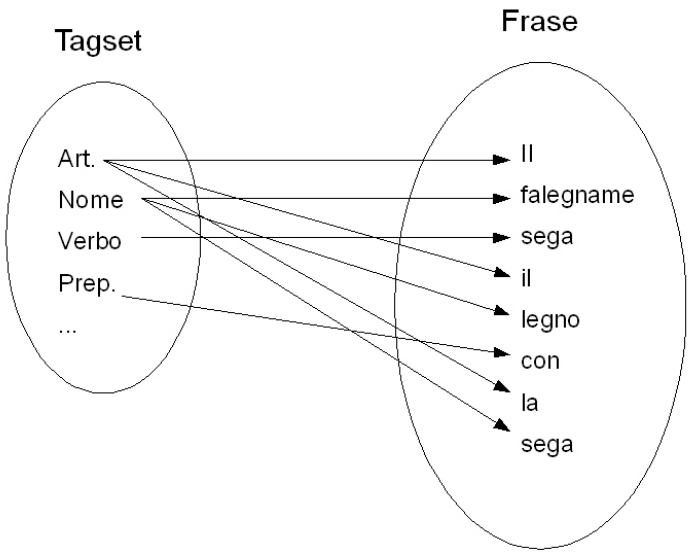


*FIGURA 1.3 – Ambiguità delle parole omografe*

3 E non solo! Un Tagset eccessivamente granulare in alcuni casi può essere difficile da applicare anche per un linguista esperto.

Il problema principale per un Tagger è quello di risolvere l'ambiguità che si crea in presenza di parole aventi forma omografa ma diversa categoria grammaticale dovuta al diverso contesto sintattico che circonda la parola stessa, come nel caso illustrato in FIGURA 1.3 per la parola "*sega* ", la quale può essere compatibile sia con la lettura "verbo", sia con la lettura "nome" (quindi due Tags).

L'output di un PoS Tagger consisterà in una sequenza di parole dove a ciascuna delle quali sarà associata una sola Tag, la migliore possibile.

In una lingua come l'italiano, circa il 55% delle parole sono compatibili con una sola analisi morfosintattica, ma le restanti, se prive di contesto, possono riceverne più di una.

La struttura di un PoS Tagger generico è descritta in FIGURA 1.4.

Per quanto riguarda lo stato dell'arte attuale dei PoS Taggers, per la lingua inglese[4] l'accuratezza raggiunta si aggira intorno al 98%, ovvero, il Tagger nel 98% dei casi assegna ad una parola la stessa categoria grammaticale che avrebbe assegnato un linguista esperto. Il tagging automatico ad oggi è una tecnologia difficilmente migliorabile a causa della forte ambiguità intrinseca al linguaggio naturale. Può capitare infatti che anche linguisti esperti si trovino in disaccordo sull'analisi da attribuire ad una parola in un determinato contesto, o persino ad una intera frase.

"*una vecchia legge la regola*" → quale analisi associare a questa frase?

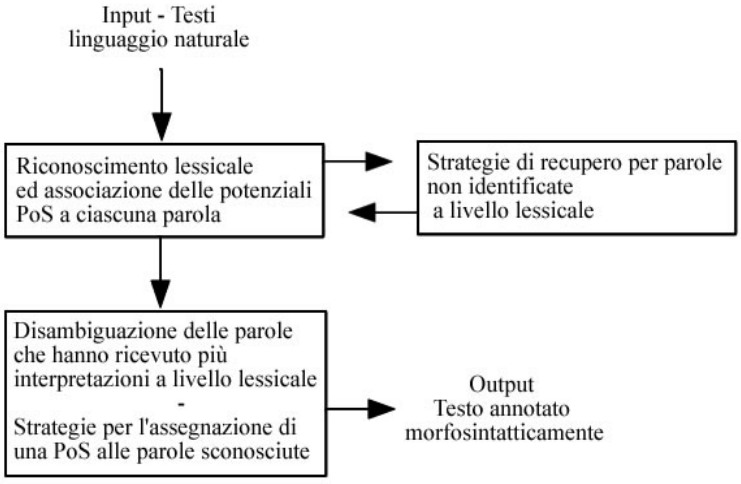


*FIGURA 1.4 – Struttura generica di un Part of Speech Tagger*

4 Risultati simili sono stati raggiunti anche per la lingua italiana nell'ambito dell'iniziativa EVALITA 2007

## 1.6 Principali architetture per il Part of Speech Tagging automatico

I principali algoritmi per il task della disambiguazione implementati dai Taggers si dividono in due classi:

1. Taggers basati su regole manuali.
2. Taggers basati su algoritmi di Machine Learning, i quali si dividono principalmente in due gruppi:
   - Taggers "Stocastici", i quali risolvono l'ambiguità su base statistica
   - Taggers "Transformation Based Learning"

## 1.7 Taggers a regole manuali

Risolvono l'ambiguità sulla base di regole scritte manualmente da linguisti. Esistono vari criteri per scrivere le regole. In linea di massima, per ogni categoria grammaticale si devono prima ipotizzare tutti i contesti sintattici nei quali una parola possa ricevere una determinata part of speech, poi eliminare le situazioni più improbabili e convertire le restanti in regole di disambiguazione da applicare.
Architettura degli algoritmi basati su regole per l'assegnazione della PoS:

1. Al primo stadio di analisi viene utilizzato un dizionario per assegnare ad ogni parola presente nel testo una lista di potenziali parti del discorso.
2. Al secondo stadio di analisi viene utilizzata la lista di regole di disambiguazione manualmente scritte per selezionare una singola parte del discorso per ogni parola presente nel testo.

Per esempio, considerando la frase "*il bambino la mangia*", il processo di analisi sarà il seguente:

- Primo stadio: analisi morfologica.

  Un analizzatore morfologico sulla base di un dizionario e di determinate regole associa a ciascuna parola le rispettive interpretazioni morfologiche (letture):

| *Parola* | | *Lettura/e* |
|---|---|---|
| Il | → | Art |
| bambino | → | N |
| la | → | Art, Pron |
| mangia | → | V |

- Secondo stadio: fase di disambiguazione.

  Il Tagger basato su regole esamina la sequenza delle parole componenti la frase.

  La situazione ideale è quella in cui a ciascuna parola sia stata associata una sola lettura.

  Nel caso della frase "il bambino la mangia", la stringa "la" ha ricevuto due letture possibili.

  Una potenziale regola di disambiguazione potrebbe essere la seguente:

  Rimuovi la lettura "articolo" se la parola che segue è un "verbo"

  Applicata la regola il risultato sarà quello descritto in TABELLA 1.1

TABELLA 1.1
Output di un Tagger basato su regole di disambiguazioni manualmente scritte

| **PoS** | **Art** | **N** | **Pron** | **V** |
|---|---|---|---|---|
| **Parola** | Il | Bambino | La | mangia |

Nel 1971, presso la statunitense Brown University, con un programma di Part of Speech automatico chiamato Taggit (Greene and Rubin, 1971) fu raggiunta un'accuratezza del 77% nell'annotazione automatica di un corpus delle dimensioni di un milione di parole[5], composto da testi scritti in inglese standard.

Taggit operava con un Tagset composto da 86 diverse Tags per la descrizione delle varie parti del discorso, e circa 3300 regole scritte da linguisti per il task della disambiguazione.

Un altro importante Part of Speech Tagger basato su regole è stato l'ENGCG (Voutilainen, 1995), sviluppato nel 1995 presso il dipartimento di studi linguistici dell'università di

5 Si trattava del "Brown Corpus". L'annotazione fu completata con correzioni e revisioni manuali da parte di linguisti.

Helsinki, il quale, disponendo di oltre 1.100 regole[6] per il task della disambiguazione, raggiunse un accuratezza del 99,7%, anch'esso sull'inglese standard.

## 1.8 Taggers Stocastici

Questa tipologia di taggers risolve l'ambiguità assegnando a ciascuna parola presente in un testo la Tag contestualmente più probabile.
Un Tagger stocastico generalmente si compone di:

- un formario, cioè un dizionario utilizzato per assegnare a ciascuna parola presente nel testo una lista di potenziali parti del discorso.
- un "training corpus", cioè un campione di dati manualmente annotati sul quale gli algoritmi del tagger calcolano, in base ad evidenza statistica, la probabilità che ad una data parola venga associata una specifica Tag dato un particolare contesto.
- Il programma di disambiguazione vero e proprio, il quale, sulla base delle evidenze statistiche calcolate sul "training corpus", sceglierà per ciascuna parola presente nel testo da annotare una sola Tag.

Per questa tipologia di Taggers non c'è bisogno di scrivere manualmente regole di disambiguazione, poiché le evidenze probabilistiche vengono estratte automaticamente dal corpus di addestramento preannotato (non ambiguo).

### 1.8.1 Il corpus di addestramento

I training corpora, o corpora di addestramento, sono corpora annotati con del particolare codice che esplicita l'informazione linguistica associata al dato testuale[7].
Un "training corpus" costruito per addestrare un sistema di Tagging basato su algoritmi auto-apprendenti è una collezione di testi, selezionati ed annotati a mano da linguisti esperti, dove a ciascuna parola presente è stata assegnata la Tag che ne esplicita la corretta categoria morfosintattica (esempio in FIGURA 1.5).
Sulla base di questo fondamentale strumento, durante la fase di apprendimento supervisionato l'algoritmo di "machine learning" del Tagger "capisce" quali sono le

6 Molto più sofisticate rispetto a quelle implementate dai primi algoritmi di tagging basati su regole.
7 I livelli di annotazione più comuni e sono: annotazione morfo-sintattica, sintattica, semantica, pragmatica.

regolarità che determinano l'assegnazione di una Tag ed il rapporto che questa ha con le informazioni contestuali, “capisce” che esistono dei vincoli che determinano sequenze ben precise di Tag, ed induttivamente si costruisce un modello del funzionamento del linguaggio.

Il corpus di addestramento deve essere costruito con molta attenzione e con criteri ben precisi, selezionando i testi che lo compongono in base allo scopo per cui viene addestrato il programma.

Per esempio, se lo scopo del programma è quello di analizzare l'italiano standard, i testi selezionati dovranno essere rappresentativi di quella particolare “variante” di italiano; un corpus di addestramento costruito con testi appartenenti a registri specialistici della lingua italiana produrrebbe pessimi risultati, poiché la macchina basandosi su di esso si creerebbe una modello di linguaggio troppo specifico e non generalizzabile a quello che è l'italiano standard.

Un fattore importante è rappresentato dalla dimensione del corpus: infatti più un corpus è ampio maggiori saranno i casi osservati in fasi di training e di conseguenza più accurato il modello del linguaggio che verrà costruito.

Per il programma c'è quindi una forte dipendenza dai dati di partenza e di conseguenza, migliore sarà il corpus di addestramento, qualitativamente parlando, migliori saranno i risultati ottenuti.

```
Battle-tested/NNP*/JJ industrial/JJ managers/NNS here/RB always/RB
buck/VB*/VBP up/IN*/RP nervous/JJ newcomers/NNS with/IN the/DT tale/NN
of/IN the/DT first/JJ of/IN their/PP$ countrymen/NNS to/TO
visit/VB Mexico/NNP ,/, a/DT boatload/NN of/IN samurai/NNS*/FW
warriors/NNS blown/VBN ashore/RB 375/CD years/NNS ago/RB ./.
"/" From/IN the/DT beginning/NN ,/, it/PRP took/VBD a/DT man/NN
with/IN extraordinary/JJ qualities/NNS to/TO succeed/VB in/IN Mexico/NNP ,/,
"/" says/VBZ Kimihide/NNP Takimura/NNP ,/, president/NN of/IN
Mitsui/NNS*/NNP group/NN 's/POS Kensetsu/NNP Engineering/NNP
Inc./NNP
unit/NN ./.
```

*FIGURA 1.5 – Esempio di testo annotato con il Tagset della Penn Treebank*

### 1.8.2 Taggers basti su Hidden Markov Model

Esistono molti tipi di taggers stocastici, i quali differiscono tra loro a seconda del modo in cui estraggono le statistiche dal corpus di addestramento e di come le riapplicano ai nuovi dati.

I modelli più semplici e diffusi di Tagger stocastici sono quelli basati sulle "Catene Nascoste di Markov" (Hidden Markov Model Taggers - HMM).

In questi modelli, i tokens vengono considerati come una catena di simboli osservabili emessa da una catena di stati sottostanti e non direttamente osservabili, le Tags. (stati del modello).

Le catene nascoste di Tags vengono calcolate dal Tagger sulla base di due matrici di probabilità elaborate sul corpus di addestramento in fase di training:

1. Matrice delle "transizioni", nella quale vengono registrate le probabilità di passaggio da una Tag all'altra (probabilità di una Tag date "*n* - Tags" precedenti).
2. Matrice delle "emissioni", nella quale vengono registrate le probabilità di emissione di una parola data un determinata Tag.

Lo scopo del tagger è quello di calcolare quale sia la sequenza di stati nascosti (Tags) che abbia la più alta probabilità di emettere una data sequenza di simboli (Tokens, "parole"), come esemplificato dalla [1.1].

La probabilità di una determinata sequenza di Tags viene calcolata moltiplicando la probabilità delle transizioni per le probabilità delle emissioni:

[1.1] $$\underbrace{P(\text{word} \mid \text{Tag})}_{\text{emissioni}} * \underbrace{P(\text{Tag} \mid \textit{precedenti}\ \text{n Tag})}_{\text{transizioni}}$$

La [1.1] indica che ogni parola dipende unicamente dalle sue possibili Tag, $P(\text{ w} \mid \text{Tag })$, e che ogni Tag dipende dalle Tags precedenti, $P(\text{ Tag} \mid \textit{precedenti}\ \text{n Tag})$. A livello grafico la [1.1] può essere rappresentata come illustrato in FIGURA 1.6

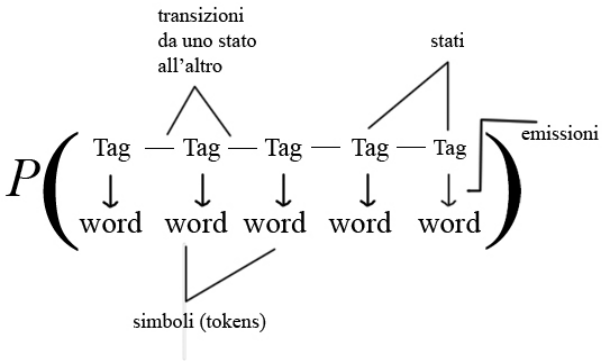


*FIGURA 1.6 – Modello di Tagger basato su HMM*

### 1.8.3 Descrizione di un Tagger HMM a Bigrammi:

I Taggers HMM assegnano ad una data parola una determinata Tag partendo dal presupposto che la Tag da assegnare dipenda esclusivamente dalle Tags che precedono quella parola.

A titolo esemplificativo vediamo come un Tagger HMM assegna la corretta Tag ad una singola parola, attraverso la descrizione del funzionamento di un Tagger Bigramma basato su HMM.

Un Tagger Bigramma sceglie la Tag "$t_i$", da assegnare ad una parola "$w_i$", che risulti più probabile data la Tag precedente "$t_{i-1}$" e la parola corrente "$w_i$", secondo la [1.2]:

$$[1.2] \qquad t_i = \arg\max_j P(t_j \mid t_{j-1})\, P(t_j \mid w_i)$$

Nella [1.2] , l'operatore "arg*max*" serve per calcolare la massima probabilità per ciascuna sequenza "parola - tag" possibile; "*j*" è l'insieme "*j*" delle possibili Tag.

Ipotizzando che il Tagger HMM Bigramma si trovi difronte alla frase "*lo stato italiano*", e che in tale frase l'unica parola ambigua sia "*stato*".

lo / Art stato / ??? italiano / Adj

Il compito del Tagger sarà quello di decidere se assegnare alla parola "stato" la categoria

grammaticale “nome” o “verbo”, calcolandone le rispettive probabilità[8]:

$$P(\text{V} \mid \text{Art})\ P(\text{stato} \mid \text{V}) \qquad vs \qquad P(\text{N} \mid \text{Art})\ P(\text{stato} \mid \text{N})$$

Sulla base di un corpus di addestramento vengono calcolare sia le probabilità di emissione, sia le probabilità di transizione.

Supponiamo che le probabilità di transizione siano le seguenti:

- $P$ (V | Art) = .0012 – probabilità di uno stato[9] “verbo” dato lo stato precedente “articolo”
- $P$ (N | Art) = .02 – probabilità di uno stato “nome” dato lo stato precedente “articolo”

Supponiamo che le probabilità di emissione (probabilità lessicali) siano le seguenti:

- $P$ (stato | V)[10] = .019
- $P$ (stato | N) = .011

Moltiplicando tra loro questi risultati otterremo che la categoria grammaticale più probabile per la parola “*stato*” dato la Tag precedente, “il / Art”, è: “nome” → stato / N

$P$ (V | Art) $P$ (stato | V) = .0000228

$P$ (N | Art) $P$ (stato | N) = .00022

I Taggers HMM generalmente sono più complessi di quello descritto sopra e non si limitano a selezionare una sola Tag per una sola parola, bensì una sequenza di Tag per una sequenza di parole, come descritto nella [1.3]

[1.3] $$T(w_{i\ldots n}) = \underset{t_{1\ldots n}}{\arg max}\ P(t_{1\ldots n} \mid w_{1\ldots n})$$

I modelli più diffusi e che in genere danno buoni risultati soni i Tagger HMM a Trigrammi.

[1.4] $$P(t_n) = P(w_n \mid t_n)\ P(t_n \mid t_{n-2}, t_{n-1})$$

8 La scelta verte tra “nome” e “verbo” poiché si assume che a livello di analisi morfologica, sulla base di un dizionario, siano state ipotizzate solo queste due categorie grammaticali per la parola “*stato*”

9 Stato = “stato finito” della macchina basata su HMM, da non confondersi con il token in esame *stato*

10 Da leggere “se ci aspettassimo un verbo, quanto è probabile che questo verbo sia *stato*?”

Data una determinata sequenza di parole, un Tagger HMM deve sostanzialmente esplorare tutte le possibili sequenze di Tags compatibili con la sequenza di parole in esame e selezionare quella che abbia la maggiore probabilità di emettere quella data sequenza di parole.

Il metodo di calcolo implementato per l'esplorazione delle sequenze di Tags dai Taggers HMM è chiamato "Algoritmo di Viterbi"[11].

L'Algoritmo di Viterbi riduce drasticamente il numero di operazioni svolte nel compito dell'esplorazione partendo dal presupposto che la maggior parte dei potenziali calcoli siano altamente improbabili; anche se la catena di Tags selezionata potrebbe non essere effettivamente la più probabile, poiché la certezza assoluta la avremmo soltanto tramite il calcolo di tutte le possibili catene nascoste di Tags[12], i risultati ottenuti con l'adozione dell'Algoritmo di Viterbi risultano corretti nella maggioranza dei casi.

### 1.8.4 Gestione delle parole sconosciute

Si è detto che le emissioni e le transizioni vengono calcolate sulla base di un corpus di addestramento in fase di training, e che la potenziali categorie grammaticali vengono attribuite a ciascuna parola in una precedente fase di analisi morfologica sulla base di un formario (dizionario).

Per quanto possa essere esteso un corpus di addestramento e per quanto possa essere ampio e ben strutturato un formario, può capitare che in fase di tagging il Tagger HMM si trovi davanti alcuni simboli (tokens, "parole") mai visti prima, sia perché non presenti nel formario, sia perché non riscontrati nel training corpus.

Esistono varie strategie per la gestione di questi casi; quella di maggior successo si basa sull'analisi dei suffissi, la quale, esaminando i suffissi della parola, cerca di indurne la corretta categoria grammaticale.

Nel 1988 il Tagger HMM DeRose[13] (DeRose, 1988) raggiunse sull'inglese standard un'accuratezza di tagging che si aggirava intorno al 95-96%.

---

11 Elaborato nel 1967

12 Visto il numero di calcoli richiesto, l'esplorazione di tutte le potenziali catene nascoste risulterebbe ad oggi impraticabile e nella pratica renderebbe impossibile l'utilizzo dei modelli HMM.

13 DeRose, 1988 – Sviluppato da Steven J. DeRose

## 1.9 Taggers ”Transformation Based Learning”

I Taggers “Transformation Based Learning” risolvono l'ambiguità utilizzando delle regole, come i Taggers “Rules Based” descritti sopra, ma a differenza di quest'ultimi le regole non vengono scritte a mano. Infatti questa tipologia di tagger, basandosi su speciali algoritmi di “Machine Learning”, apprende automaticamente le regole necessarie per risolvere il task della disambiguazione.

Un esempio efficace di Tagger basato su algoritmi di “Machine Learning” è rappresentato dal Brill Tagger (Eric Brill, 1992), il più famoso Tagger ”Transformation Based Learning”.

### 1.9.1 Descrizione del funzionamento del Brill Tagger

L'input del Brill Tagger è un testo tokenizzato e non annotato sul quale vengono eseguiti i seguenti passi:

- A ciascuna parola presente nel testo viene assegnata una sola Tag, sulla base di un lessico.

  Se il lessico prevede più Tag per la parola in esame, il Brill sceglie la prima disponibile.
- Alle parole sconosciute viene assegnata le Tag più probabile attraverso criteri euristici, regole di trasformazioni lessicali e contestuali.

  Prendiamo la frase “*il luogotenente del regno*” ed ipotizziamo che “*luogotenente*” sia una parola sconosciuta.

  *Il* / Art@ *luogotenente* / ?? *del* / Prep@ *regno* / N@

  Un esempio di euristica potrebbe essere il seguente: chiamiamo “X” la parola sconosciuta. Come esemplificato nella [1.5]; se la parola “X” inizia per lettera maiuscola, considerare la parola come “nome proprio”, NP@, altrimenti considerare la parola come “nome comune”, N@.

[1.5]
```
IF (X inizia per lettera maiuscola) {
    assegna NP@ alla parola X
} ELSE {
    assegna NN@ alla parola X
}
```

Un esempio di trasformazione lessicale potrebbe essere il seguente: Brill Tagger cerca di assegnare una Tag esaminando la struttura morfologica di una parola, in questo caso il suffisso, come esemplificato dalla [1.6]. Se le ultime 4 lettere della parola sono "*ente*", allora scrivere la Tag come avverbio, Adv@.

[1.6] *ente* **HASSUF** 4 word → Adv@

Un esempio di trasformazione contestuale potrebbe essere il seguente: Brill Tagger assegna la Tag esaminando il contesto sinistro e destro della parola da disambiguare, come esemplificato dalla [1.7]. La Tag Adv@ assegnata alla parola "X" viene riscritta come nome, N@, se la Tag precedente è un articolo, Art@ e la Tag successiva una preposizione, Prep@.

[1.7] Adv@ → N@ **SURROUNDTAG** Art@ Prep@

Il risultato finale per la frase iniziale sarà: *il* / Art@ *luogotenente* / N@ *del* / Prep@ *regno* / N@

### 1.9.2 Addestramento supervisionato del Brill Tagger

Per la fase di addestramento viene fornito al Brill Tagger un corpus già annotato, $C_a$, sulla base del quale il Tagger individua per ogni parola la Tag più probabile applicando la nozione di "probabilità con frequenza", come descritto dalla [1.8], dove "$F_w$" indica il numero di volte in cui la parola in esame compare in tutto il corpus di addestramento. Per esempio, se nel corpus di addestramento la parola "*ciao*" compare 90 volte come "interiezione" e 15 come "nome", Brill Tagger considera che "interiezione" sia l'annotazione più probabile.

[1.8]
$$P(\text{tag} \mid \text{parola}) = \frac{(\text{parola} / \text{tag})}{F_w}$$

*quindi*

$$P(\text{Int@} \mid \text{ciao}) = 90/105 \quad \textit{vs} \quad P(\text{N@} \mid \text{ciao}) = 15/105$$

Fatto ciò, il Tagger deve "imparare" quali sono i contesti nei quali una Tag debba essere trasformata in un'altra Tag. (regole di trasformazione contestuale).
L'input del Tagger è costituito dai seguenti elementi:

- Una lista ordinata "**T**" di transizioni inizialmente vuota.
- Un lessico "**L**" con la Tag più probabile per ogni parola (costruito in precedenza, come descritto nella [1.8])
- Il corpus di addestramento "**C**" privo di annotazione
- Il corpus di addestramento "$C_a$" (corpus già annotato)
- Gli schemi di trasformazione "**S**"

In un primo passaggio **C** viene annotato mediante l'assegnazione della Tag più probabile a ciascuna parola. Dopodiché **C** viene confrontato con $\mathbf{C}_a$ per redigere una lista degli errori compiuti dal Tagger (inizialmente saranno molti).
Il Tagger istanzia per ogni errore gli schemi di trasformazione, creando una serie di potenziali trasformazioni contestuali, alcune sensate, alte no. Per ogni potenziale trasformazione contestuale il corpus **C** viene riannotato. Tra tutte le trasformazioni potenziali, quelle che hanno determinato la maggiore riduzione degli errori vengono aggiunte alla lista ordinata **T** , le altre vengono scartate.
A questo punto il corpus **C** annotato con le trasformazioni contenute in **T** viene nuovamente confrontato con $\mathbf{C}_a$ , per redigere una nuova lista di errori ed indurre nuove potenziali trasformazioni. Il processo va avanti fino a quando non si raggiunge un tasso di errore molto basso.
Le regole di trasformazione generate dal Tagger sono facilmente leggibili e manualmente modificabili.

## 1.10 Conclusioni: "Regole manuali" vs "Machine Learning"

Come abbiamo visto dalla descrizione dei vari modelli di Taggers, sia le architetture basate su regole manuali, sia le architetture basate su algoritmi di machine learning producono ottimi risultati nel campo dell'annotazione automatica del linguaggio naturale.
Tuttavia i taggers basati su machine learning presentano notevoli vantaggi rispetto ai taggers basati su regole manuali: questi ultimi infatti richiedono molto tempo e lavoro per

la costruzione, soprattutto per la scrittura delle regole di disambiguazione. Inoltre questi tipi di taggers funzionano soltanto per la lingua per la quale sono stati costruiti. Cambiando lingua o dominio linguistico le regole scritte diventano inefficaci e in alcuni casi del tutto inapplicabili.

Viceversa, i taggers basati su algoritmi di machine learning risultano molto più versatili poiché grazie ai corpora di addestramento possono essere riaddestrati e quindi adattati ad ogni tipo di lingua e dominio linguistico che si voglia analizzare. In più, in termini di lavoro, sono meno costosi da sviluppare.

# 2 Il sistema SmarText

## 2.1 SmarText: un'introduzione

Presso l'istituto di Linguistica Computazionale del CNR di Pisa è stato sviluppato un sistema per il "Trattamento Automatico del Linguaggio Naturale" per la lingua Italiana chiamato SmarText.

Tale sistema si sviluppa su più moduli, o processi, ed ogni modulo definisce un diverso livello di analisi del testo, come esemplificato dallo schema in FIGURA 2.1:

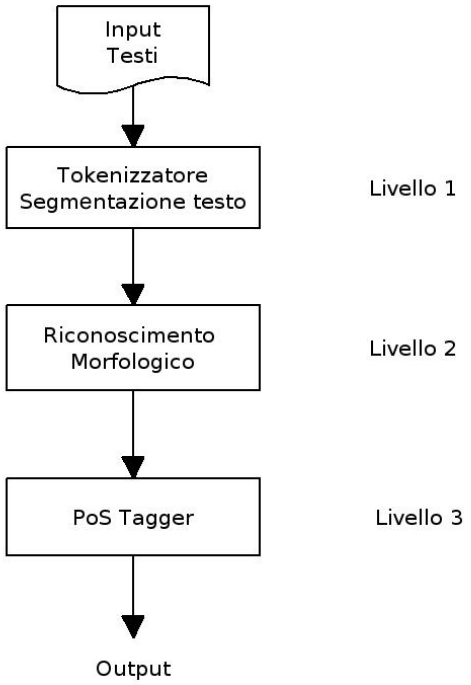


*FIGURA 2.1 – Rappresentazione semplificata del sistema SmarText*

In una fase di pre-tokenizzazione vengono preparate le "Unità Testuali" (UT), porzioni di testo in formato digitale opportunamente strutturate con del codice di Mark-Up che ne espliciti la struttura (meta-informazione), le quali vengono poi passate manualmente al sistema SmarText come files di input. L'output generato da ciascun modulo diventa a sua volta l'input del modulo che si trova al livello immediatamente successivo.

Le Unità Testuali vengono elaborate dal primo modulo di SmarText, il Tokenizzatore, il

quale ha il compito di segmentare il testo in tokens, producendo come output una lista di "Unità Ortografiche" (UO).

Il modulo successivo, l'Analizzatore Morfologico, prende in input le unità ortografiche precedentemente generate dal Tokenizzatore ed ipotizza per ciascuna di esse una o più interpretazioni morfologiche possibili (IM). Alcune delle interpretazioni morfologiche prodotte possono risultare ambigue o sbagliate; ciò è dovuto al fatto che a questi due primi livelli di elaborazione del testo i tokens vengono analizzati e processati singolarmente, prescindendo dal contesto circostante.

Il terzo modulo di SmarText, il "Part of Speech Tagger", ha il compito di:

- Validare l' interpretazione morfologica associata a ciascuna UO, nel caso ve ne sia una soltanto.
- Disambiguare e scegliere una sola interpretazione morfologica per ciascuna unità ortografica alla quale l'Analizzatore Morfologico abbia associato più interpretazioni.
- Assegnare una sola interpretazione morfologica all'unità ortografica in esame, nel caso in cui l'Analizzatore Morfologico non sia riuscito ad associargliene una.

Lo schema raffigurato in FIGURA 2.1 non rappresenta l'intera struttura del sistema oggetto di studio, il quale si compone di due ulteriori livelli:

- Livello 4: modulo per il livello di annotazione sintattica del testo (modulo Chunking).
- Livello 5: modulo per l'individuazione e classificazione semantica delle entità quali nomi, indirizzi, organizzazioni, numeri ecc. (modulo NER, Name Entity Recognition).

La ragione per cui i moduli dei livelli 4 e 5 non saranno trattati in questa tesi e dovuta al fatto che il centro di interesse delle mie analisi è costituito essenzialmente dai primi tre moduli, focalizzandomi in particolare sul terzo livello.

## 2.2 Processi di SmarText: la Tokenizzazione del testo

Tokenizzare un testo significa spezzarlo in singole unità ortografiche dette "tokens", unità minime di analisi necessarie per le successive operazioni computazionali di SmarText, dove per "token" intendiamo "*una sequenza continua di uno o più caratteri contenuta tra*

*due spazi bianchi*"[14].

Un computer che debba processare un testo di qualsiasi genere o lingua, deve prima di tutto essere in grado di "capire" dove siano le parole, dove finisca una parola ed inizi la seguente, cosa sia "parola" e cosa sia "punteggiatura".

La fase di tokenizzazione può essere dunque considerata una fase preliminare rispetto alle successive fasi di analisi linguistica del testo.

Il processo di segmentazione del testo in SmarText si svolge come descritto dal grafico in FIGURA 2.2:

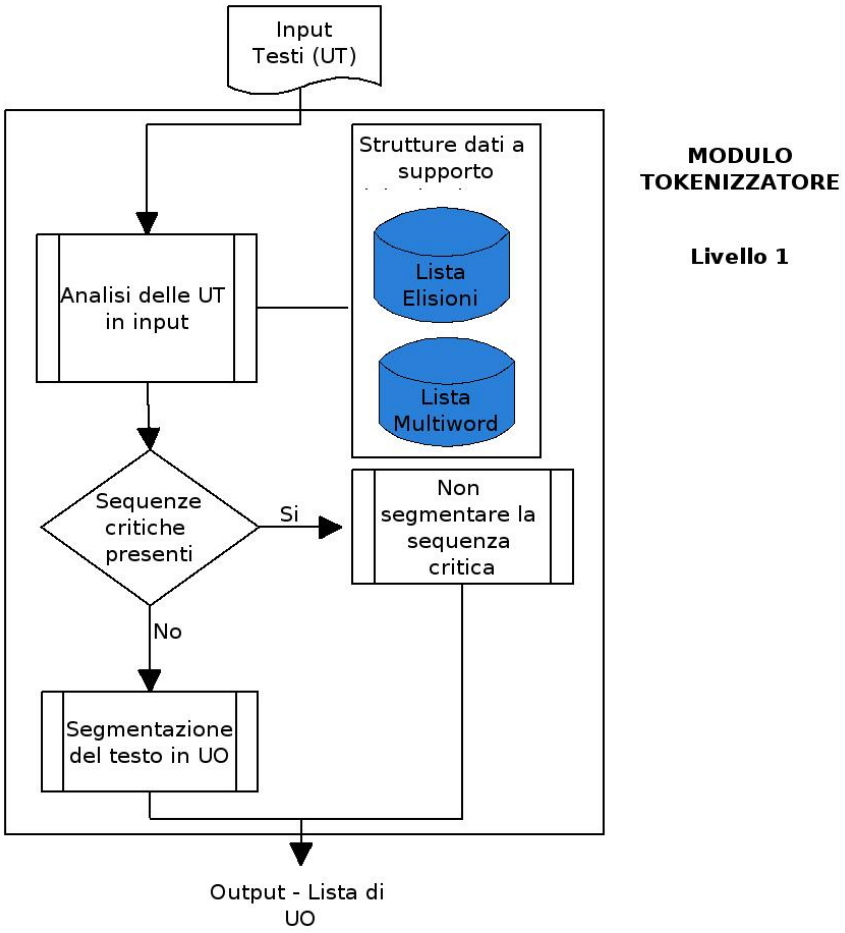


*FIGURA 2.2 – "Livello 1" il modulo tokenizzatore di SmarText*

14 Tale definizione risulta comunque approssimativa, a causa delle numerose eccezioni per natura intrinseche alla lingua. Fanno eccezione le *multiword*, parole contenenti caratteri di spaziatura al loro interno ma considerate come un unico token, ed i caratteri di interpunzione, che, sebbene siano attaccati alla parola che li precede devono essere considerati token a se.

Prendiamo in esame la seguente porzione di testo fornita al modulo Tokenizzatore come input:

| *Non appena vide l'autobus, il sig. Rossi corse alla fermata.* |
|---|

Tutto ciò che dovrà fare il Tokenizzatore sarà cercare di segmentare il testo in unità ortografiche, sequenze di caratteri che corrispondano alla definizione di "token" sopra data.

Il processo di segmentazione del testo avviene secondo regole implementate dal Tokenizzatore tramite l'utilizzo di Espressioni Regolari, strumenti fondamentali per l'elaborazione di stringhe.

Il risultato di tale operazione, in base alla nostra definizione di token, sarà il seguente:

| *Non* | *appena* | *vide* | *l'autobus* | *,* | *il* | *sig.* | *Rossi* | *corse* | *alla* | *fermata.* |
|---|---|---|---|---|---|---|---|---|---|---|

Esaminando l'output ci rendiamo subito conto che non tutte le unità sono state segmentate correttamente, come nel caso di "*l'autobus"*, dove l'articolo in forma elisa non è stato staccato dal sostantivo, e come nel caso di "*fermata.*", dove il punto di fine frase non è stato separato dal sostantivo; ciò è dovuto al fatto che la stringa viene suddivisa prendendo come punto di riferimento per identificare il confine di parola gli spazi bianchi che intercorrono tra ogni sequenza continua di caratteri.

Per gestire casi come il primo, il Tokenizzatore si avvale di due *strutture dati di supporto*, liste aperte contenenti la prima un elenco di "multiword", parole che sebbene contengano spazi al loro interno, devono essere considerate come un unico token (best seller, nulla osta, al di là [...] ecc.), l'altra un elenco di "forme elise della lingua italiana" (l', dell', all', [...] ecc.); grazie a tali strutture di supporto la forma elisa verrà riconosciuta e segmentata in maniera corretta:

| *Non* | *appena* | *vide* | *l'* | *autobus* | *,* | *il* | *sig.* | *Rossi* | *corse* | *alla* | *fermata.* |
|---|---|---|---|---|---|---|---|---|---|---|---|

Due casi particolari sono rappresentati dai caratteri "punto" ed "apice" i quali possono assumere funzioni diverse a seconda del contesto in cui si trovano, tanto che in alcuni casi

questi caratteri risultano essere token indipendenti e vanno quindi segmentati, mentre in altri sono parte integrante della sequenza di caratteri che seguono e dalla quale non devono essere separati.

Nel caso di "f*ermata.*" il "punto" ha funzione di "punto di fine frase", quindi token indipendente, mentre nel caso di "*sig.*" il "punto" assume funzione di "punto di abbreviazione" facendo quindi parte della sequenza "*sig*"; nel caso in cui la sequenza di caratteri "*sig.*" si trovasse in posizione di fine frase, allora il "punto" assumerebbe la duplice funzione di "punto di fine frase" e "punto di abbreviazione". Problema analogo si presenta in presenza del carattere "apice", il quale può avere funzioni sia di "accento" sia di "virgoletta".

Le sequenze di caratteri di questo tipo vengono identificate come "sequenze critiche".

Poiché a questo livello di elaborazione del testo non è possibile determinare se un "punto" o un "apice" facciano parte o meno di una determinata parola, la "sequenza critica" in esame non viene segmentata dal Tokenizzatore, e la sua analisi sarà rimandata ai successivi livelli di elaborazione testuale di SmarText.

## 2.3 Processi di SmarText: Magic e l'analisi morfologica

Magic è il modulo per l'analisi morfologica[15] del testo in SmarText; il suo compito è quello di analizzare morfologicamente il testo tokenizzato, associando ad ogni UO un codice (Tag) che ne espliciti la struttura e le informazioni morfosintattiche.

Il testo pre-elaborato dal Tokenizzatore viene rappresentato come una lista di unità ortografiche corredate da determinati attributi, i quali permettono a Magic l'identificazione di ciascuna UO all'interno della lista.

La parte principale di Magic è costituita dal formario, un repertorio di circa 100.000 entrate lessicali per uso generico, composto da:

- Parole (sottolessico di tipo "word"): componenti lessicali che non flettono e non sono liberamente declinabili (avverbi , [...], preposizioni semplici ).
- Radici (sottolessico di tipo "root"): le radici costituiscono una classe aperta, in continua

15 Morfologia: con tale termine in linguistica si identifica quella disciplina che studia la struttura della parola, dove per "parola" si intende "la minima combinazione di morfemi costruita spesso attorno ad una base lessicale [...] che possa costituire da sola un segno linguistico compiuto" - Berruto, 1997

evoluzione; sono "morfemi[16] lessicali" che fanno riferimento ad un concetto, alla realtà esterna alla lingua.

- Desinenze (sottolessico di tipo "ending"): parti finali di parola che, combinate alle radici, servono a produrre le forme flesse delle parole.
- Suffissi (sottolessico di tipo "suffix"): morfemi che, attaccati alla radice, permettono la formazione di nuove parole. Possono essere suffissi derivazionali, suffissi diminutivi, ecc.

Le desinenze ed i suffissi sono "morfemi grammaticali" e costituiscono un inventario chiuso.

Alle radici ed alle desinenze sono associate determinate "classi di flessione" che permettono a Magic di coniugare una determinata classe di radice con una certa classe di desinenza: per esempio, il verbo "mangiare" verrà identificato dalla classe di flessione con valore "verbo in a" (verbo della prima coniugazione).

L'utilizzo della "classe di flessione" è fondamentale per il riconoscimento delle parole poiché impone al sistema di generare stringhe solo dove la classe di flessione di una radice sia compatibile con la classe di flessione di una determinata desinenza.

La ricchezza del formario di Magic e la qualità delle informazioni in esso contenute sono due fattori estremamente importanti ai fini della corretta elaborazione del linguaggio, poiché in SmarText ogni singolo livello di analisi è strettamente dipendente dal livello che lo precede.

Magic si avvale anche di altre strutture dati di supporto, rappresentate da una lista di abbreviazioni ed una lista di acronimi.

Tutti questi repertori sono liste aperte oggetto di continue modifiche, aggiornamenti e correzioni.

L'input di Magic è il file tokenizzato ed il processo di analisi morfologica delle unità ortografiche avviene come descritto dal grafico in FIGURA 2.3:

16 Morfema: pezzo di parola recante ancora significato proprio. Es: nella parola "*Gatto* ", il morfema finale "*o* " reca il significato di "singolare, maschile ".

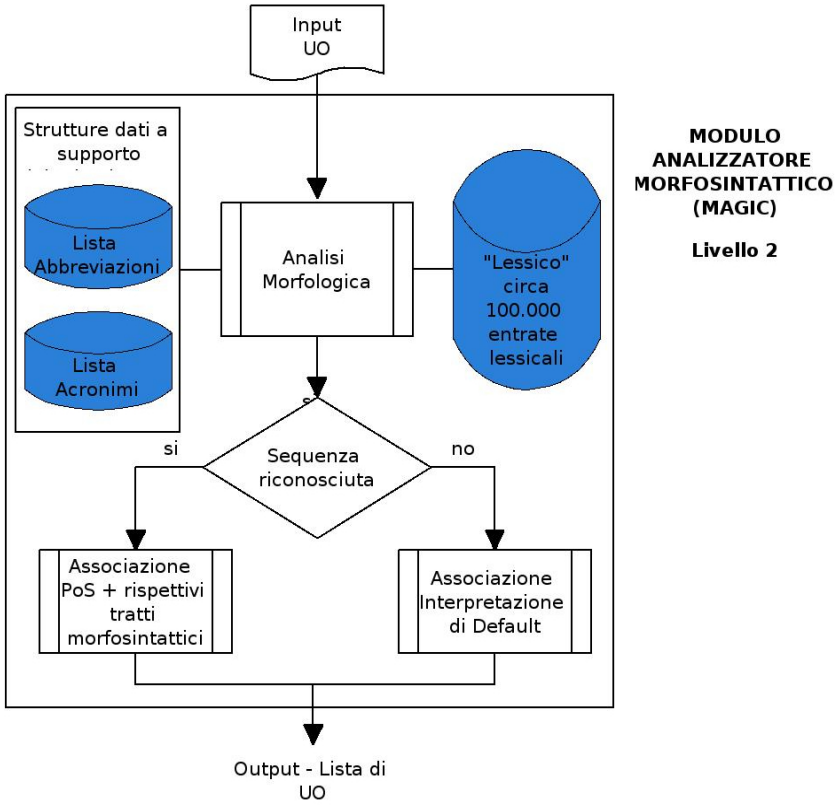


*FIGURA 2.3 - "Livello 2" Magic modulo analizzatore morfosintattico di SmarText*

### 2.3.1 Funzionamento di Magic

Presa in esame una singola unità ortografica UO, Magic effettua un primo test, andando a controllare se tale parola sia presente o meno nel suo sottolessico di tipo "word". In caso di riscontro positivo viene prodotto l'output.

In caso di riscontro negativo, il sistema toglie l'ultimo carattere della stringa e va a vedere se per i due pezzi esiste una corrispondenza rispettivamente nel sottolessico di tipo "radice", per il primo pezzo, e nel sottolessico di tipo "desinenza" per il carattere tolto, come descritto in TABELLA 2.1

Il motore di analisi verifica che radice e desinenza abbiano lo stesso valore di "classe di flessione", ed in caso positivo Magic produce l'output, altrimenti procede togliendo un altro carattere all'unità ortografica in questione. Il ciclo andrà avanti fino a quando

l'analizzatore non riuscirà a trovare un riscontro positivo con le entrate presenti nel proprio formario.

TABELLA 2.1
Unificazione di una radice con una desinenza

| Unità Ortografica | Radice | Desinenza |
|---|---|---|
| case | cas - | - e |

### 2.3.2 Output prodotto da Magic

Come output Magic produce una lista di unità ortografiche dove a ciascuna delle quali viene assegnato:

- Lemma: "categoria lessicale astratta che accomuna tutte le forme flesse ad esso riconducibile" (Lenci et al., 2005). "Al", "alla", sono forme riconducibili all'unico lemma "A".
- Part of Speech: parte del discorso (PoS) è rappresentata dalla categoria grammaticale a cui l' unità ortografica appartiene (Nome, Aggettivo [...] Verbo, Punteggiatura).
- Tratti morfosintattici: ulteriori informazioni associate alla categoria grammaticale cui l'unità ortografica appartiene**.**

Sebbene per circa la metà delle forme dell'italiano sia possibile associare una sola analisi morfologica, come per i casi elencati a titolo esemplificativo[17] in TABELLA 2.2, la restante parte può essere oggetto di più interpretazioni morfosintattiche, a seconda del contesto in cui la parola si trovi, come descritto in TABELLA 2.3. In presenza di tali forme, dette omografe, Magic associa a ciascuna di esse tutte le interpretazioni morfosintattiche compatibili con tale forma e, poiché Magic non effettua alcun tipo di confronto fra parole nel contesto, l'output di Magic verrà passato come "non disambiguato" al successivo modulo di analisi, il Tagger, il quale avrà il compito di scegliere l'analisi più appropriata in base al contesto circostante l'unità ortografica.

17 Il dizionario di riferimento utilizzato è il "DeMauro Paravia", considerato lo standard per la lingua Italiana

TABELLA 2.2
Esempi di forme dell'italiano associabili ad una sola analisi morfologica

| Unità Ortografica | Lemma | Parte del Discorso | Tratti Morfosintattici |
|---|---|---|---|
| veramente | veramente | AVVERBIO | Nessuno |
| trattazione | trattazione | NOME | Singolare - Femminile |
| per | per | PREPOSIZIONE SEMPLICE | Nessuno |
| egli | egli | PRONOME PERSONALE | Maschile – Singolare – Terza persona |

TABELLA 2.3
Esempi di forme dell'italiano associabili a più interpretazioni morfosintattiche

| Unità Ortografica | Lemma | Parte del Discorso | Tratti Morfosintattici |
|---|---|---|---|
| miscela | miscelare | VERBO | Presente Indicativo |
| | miscelare | VERBO | Imperativo |
| | miscela | NOME | Femminile - Singolare |
| ancora | ancora | AVVERBIO | Nessuno |
| | ancora | NOME | Femminile - Singolare |
| | ancorare | VERBO | Presente Indicativo |
| | ancorare | VERBO | Imperativo |

### 2.3.3 Il caso del punto e dell'apice

A questo livello di analisi del testo verranno trattate le sequenze critiche che il modulo tokenizzatore non è riuscito a segmentare, stringhe di caratteri terminanti con un "punto" o con un "apice"; prendiamo in esame una sequenza critica come "*nr.*"

Il motore di analisi confronta la sequenza critica con le entrate presenti nel proprio repertorio, cercando una corrispondenza sia per la forma terminante con il punto, sia per forma senza il punto finale. Osservando la TABELLA 2.4, vediamo che per la sequenza critica "*nr.*" Magic ha trovato corrispondenze per entrambe le forme[18]:

18 Magic associa altre interpretazioni a tale sequenza critica, ma per brevità ne ho elencate solo due.

1. La sequenza è stata riconosciuta e segmentata come un nome più un punto di fine frase.
2. Il punto è stato riconosciuto come parte integrante della sequenza critica, ma viene comunque replicato, poiché esiste la concreta possibilità che un punto di abbreviazione sia anche un punto di fine frase.

Lo stesso meccanismo vale per le sequenze critiche terminanti con un apice (TABELLA 2.5)

TABELLA 2.4
Una stringa di caratteri terminante con un punto

| Unità Ortografica | Interpretazione Morfologica | Lemma | Parte del Discorso | Tratti Morfosintattici |
|---|---|---|---|---|
| nr. | 1 | nr | NOME | Neutro |
| | | . | PUNTO | Nessuno |
| | 2 | nr. | NOME | Neutro |
| | | . | PUNTO_ABBREVIAZIONE | Nessuno |

TABELLA 2.5
Una stringa di caratteri terminante con un apice

| Unità Ortografica | Interpretazione Morfologica | Lemma | Parte del Discorso | Tratti Morfosintattici |
|---|---|---|---|---|
| senno' | 1 | senno' | CONGIUNZIONE | Nessuno |
| | | ' | ACCENTO | Nessuno |
| | 2 | senno | NOME | Maschile - Singolare |
| | | ' | VIRGOLETTA | Nessuno |

### 2.3.4 Descrizione del Tagset utilizzato da Magic per l'annotazione delle unità ortografiche

Il codice (Tag) con cui Magic annota ciascuna unità ortografica consiste in una serie di attributi organizzati secondo la seguente struttura:

UO + ( *Forma* | *Proprietà Ortografiche* | *Forma Ortografica* | *Lemma* | *PoS @ Tratti Morfosintattici* )

L'attributo "Forma" descrive il tipo di unità ortografica e può assumere i valori di "Simple", per parole semplici, "Numeric" per numeri, "Punct" per la punteggiatura ecc.
L'attributo " Proprietà Ortografiche" descrive la formattazione dell'unità ortografica e può assume i valori "Allcapitalized" nel caso in cui tutti i caratteri dell'unità siano maiuscoli, "Firstcapitalized" se solo il primo carattere è maiuscolo ecc.
L'attributo "Forma ortografica" replica la forma ortografica dell'unità trovata nel testo, assumendola come valore.
L'attributo "Lemma" descrive il lemma al quale la "Forma ortografica" si riconduce, assumendolo come valore.
L'attributo "PoS" descrive la parte del discorso associata alla forma ortografica e può assumere, per ogni interpretazione morfosintattica, uno solo tra i seguenti valori previsti dal Tagset:

- "V_FIN" verbo, forma finita
- "V_NOFIN" verbo, forma non finita (infinito, gerundio, participio)
- "V_PP" verbo, participio presente o passato
- "ADJ" aggettivo
- "ADV" avverbio
- "NN" nome
- "NN_P" nome proprio
- "PRON" pronome
- "PRON_P" pronome personale

- "INT" interiezione
- "PREP" preposizione
- "CONJ" congiunzione
- "CONJ_C" congiunzione coordinativa
- "ART_D" articolo determinativo
- "ART_I" articolo indeterminativo
- "PREP_A" preposizione articolata
- "DET" qualificatore (determinativo)
- "PDET" pre-qualificatore (pre-determinativo)
- "C_NUM" numero cardinale
- "O_NUM" numero ordinale
- "R_NUM" numero romano
- "ND_POS" nessuna analisi ricevuta

I "Tratti Morfosintattici" che si trovano alla destra del carattere "@" arricchiscono la PoS situata alla sinistra di "@" con una serie di attributi ordinati secondo il seguente criterio:

*genere – numero – persona – tempo – modo*[19]

Per ogni singola interpretazione morfosintattica ciascun attributo può assumere uno solo tra i valori previsti dal Tagset elencati in TABELLA 2.6.
Nel caso in cui una determinata categoria grammaticale non preveda uno o più tratti morfosintattici, il valore dell'attributo viene riempito con il carattere "!".

19 La lista prevede anche altri attributi i quali non saranno da me trattati poiché non rilevanti ai fini del mio studio.

TABELLA 2.6
Valori possibili per ciascun Tratto Morfosintattico

| **Genere** | **Numero** | **Persona** | **Tempo** | **Modo** |
|---|---|---|---|---|
| “M” - Maschile | “S” - Singolare | “1” - Prima | “P” - Presente | “I” - Indicativo |
| “F” - Femminile | “P” - Plurale | “2” - Seconda | “R” - Passato | “C” - Congiuntivo |
| “_” - Neutro | “_” - Neutro | “3” - Terza | “F” - Futuro | “D” - condizionale |
| “!” - Non previsto | “!” - Non previsto | “_” - Neutro | “I” - Imperfetto | “M” - Imperativo |
| | | “!” - Non previsto | “!” - Non previsto | “P” - Participio |
| | | | | “G” - Gerundio |
| | | | | “F” - Infinito |
| | | | | “!” - Non previsto |

Un testo analizzato morfologicamente si presenta coma una lista di unità ortografiche strutturate secondo la formattazione sopra descritta.
Prendendo in considerazione la parola “*stato*”, considerandola fuori da un preciso contesto, l'output dell'analisi di Magic avrà la seguente forma:

TABELLA 2.7
Output di Magic per la parola “stato”

| | | | |
|---|---|---|---|
| **stato** *SIMPLE ALLSMALL* | *stato* | *stare* | *V_PP@MS!RP!! N* |
| **stato** *SIMPLE ALLSMALL* | *stato* | *essere* | *V_PP@MS!RP!! N* |
| **stato** *SIMPLE ALLSMALL* | *stato* | *stato* | *NN@MS!!!!P N* |

L'analisi prodotta da Magic per quanto riguarda la parola “stato” è chiaramente ambigua, poiché a tale vocabolo sono state associate tre interpretazioni possibili:

- *stato stare V_PP@MS!RP* : la forma ortografica “stato” è stata ricondotta al lemma “stare”. La categoria sintattica associata è “V_PP”, cioè “Verbo Participio Passato”. I tratti morfosintattici per tale PoS sono “Maschile, Singolare, Persona non rilevante, Passato, Participio”.
- *stato essere V_PP@MS!RP* : la forma ortografica “stato” è stata ricondotta al lemma “essere”. La categoria sintattica associata è “V_PP”, cioè “Verbo Participio Passato”. I tratti morfosintattici per tale PoS sono “Maschile, Singolare, Persona non rilevante, Passato, Participio”.

- *stato stato NN@MS!!!* : la forma ortografica “stato” è stata ricondotta al lemma “stato”. La categoria sintattica associata “NN”, cioè “Nome”. I tratti morfosintattici per tale PoS sono “Maschile, Singolare”. I restanti tratti morfosintattici non sono rilevanti per tale categoria grammaticale.

## 2.4 Processi di SmarText: Part of Speech Tagging – ILC-UniPi Tagger

ILC-UniPi Tagger è il modulo per il Part of Speech Tagging di SmarText.

Il Part of Speech tagging è un processo di annotazione morfosintattica utilizzato per assegnare automaticamente la giusta categoria grammaticale a ciascuna unità ortografica presente in un testo.

A differenza dei precedenti moduli di analisi del testo, Tokenizzatore ed Analizzatore Morfologico, i quali processano rispettivamente unità testuali ed unità ortografiche senza tener conto del contesto in cui esse si trovano, il Tagger sceglie per ogni unità ortografica UO la giusta Tag[20], determinandola in base a:

- il contesto circostante l'unità ortografica in esame
- le varie interpretazioni associate a tale unità ortografica dall'Analizzatore Morfologico

Il processo di tagging in SmarText avviene come descritto in “FIGURA 2.4”:

20 Con il termine *Tag (etichetta)* si identifica l'insieme di informazioni (lemma, categoria grammaticale ed eventuali tratti morfosintattici) relativo ad una singola UO.

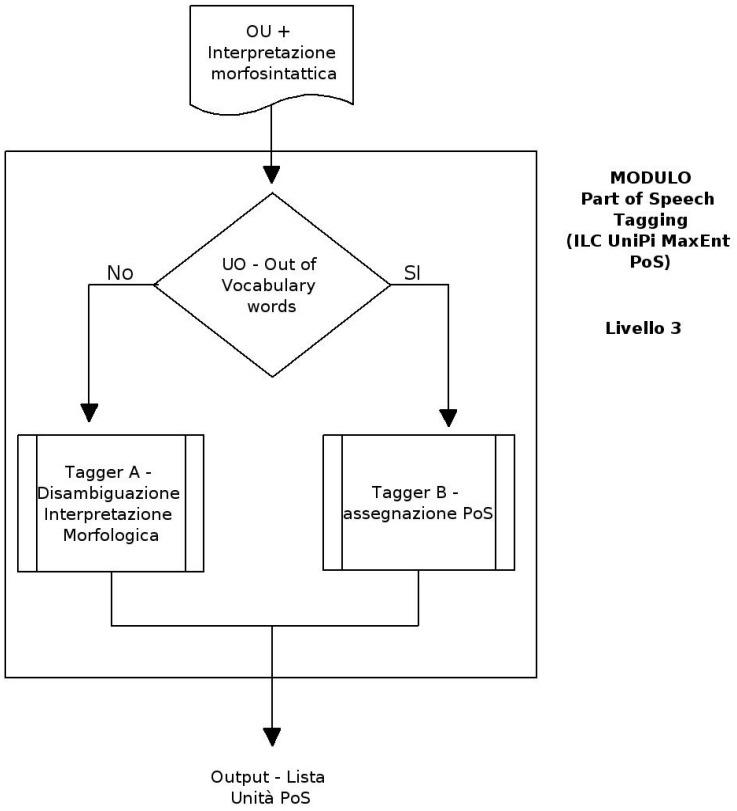


*FIGURA 2.4 - "Livello 3" Tagger: modulo per il PoS Tagging di SmarText*

### 2.4.1 Funzionamento di ILC-UniPi Tagger

ILC-UniPi Tagger è composto da due sottoprocessi, ognuno dei quali svolge compiti diversi:

- Tagger A
- Tagger B

L'input di partenza del Tagger è costituito dall'output del livello di analisi morfologica.
L'output di Magic, come precedentemente detto, è una lista di unità ortografiche dove a ciascuna di esse può essere stata associata nessuna (out of vocabulary words), una o più di una etichetta morfosintattica. La lista risulta pertanto ambigua dal punto di vista morfosintattico.

Chiamiamo l' unità ortografica in esame $W_t$[21]

I casi possibili che il Tagger dovrà gestire sono i seguenti[22]:

1. Magic ha associato all'unità ortografica più interpretazioni morfosintattiche: in questo caso l' unità ortografica viene gestita dal Tagger A, il cui compito sarà quello di disambiguare l'output dell'analizzatore morfologico, scegliendo e validando una sola analisi tra quelle proposte, in base a:
   - PoS + Tratti di $W_{t-1}$, ovvero l'UO precedente $W_t$
   - La forma di $W_t$ (maiuscola, minuscola, prima lettera maiuscola ecc) + PoS e Tratti Morfosintattici delle varie interpretazioni che Magic ha associato a $W_t$
   - PoS + Tratti di $W_{t+1}$, ovvero l'UO che segue $W_t$

2. Magic non è riuscito ad associare nessuna interpretazione a $W_t$: le unità ortografiche alle quali Magic non è riuscito ad associare un'analisi prendono il nome di "out of vocabulary words"[23]. In questo caso l'unità ortografica viene gestita dal Tagger B, il cui compito sarà quello di associarle la categoria grammaticale[24] contestualmente più probabile in base a:
   - PoS + Tratti di $W_{t-2}$
   - PoS + Tratti di $W_{t-1}$
   - La forma di $W_t$ (maiuscola, minuscola, prima lettera maiuscola ecc)
   - PoS + Tratti di $W_{t+1}$

3. Magic ha associato all'unità ortografica una sola interpretazione: il modulo PoS Tagger replicherà tale analisi come "etichetta" definitiva per $W_t$.

L'output del PoS Tagger consisterà in una lista di "UO PoS" non ambigue, dove a ciascuna unità ortografica sarà associata una sola etichetta morfosintattica comprendente il lemma, la categoria grammaticale ed eventuali tratti morfosintattici.

ILC-UniPi Tagger è un tagger statistico basato su algoritmi di apprendimento automatico

21 Target Word

22 I Tagger A e B attribuiscono la Tag corretta in base a vari fattori contestuali, ma per brevità ho elencato solo quelli principali.

23 Forme non presenti nel sottolessico con il quale Magic è equipaggiato

24 Le categorie grammaticali disponibili sono elencate in una lista predefinita.

per il task dell'attribuzione della corretta Tag.
Il meccanismo statistico implementato da ILC-UniPi Tagger per la stima delle distribuzioni probabilistiche e per l'assegnamento della giusta Tag in determinati contesti è quello della Massima Entropia.

### 2.4.2 Il metodo della Massima Entropia

ILC-UniPi Tagger associa a ciascuna unità ortografica UO presente nel testo la categoria morfosintattica "Mtag" più probabile dato un determinato contesto linguistico "$C_{UO}$" ed una serie di tratti rilevanti (definiti nel Tagset) che caratterizzano sia l'unità ortografica sia il contesto linguistico, come descritto dalla [2.1], dove:

- Mtag → è un insieme di tratti morfosintattici inclusi in una interpretazione morfosintattica IM (se ad esempio a livello di analisi morfologica una unità ortografica ha ricevuto tre interpretazioni, l'IM conterrà tre insiemi di tratti tra i quali scegliere).
- $C_{UO}$ → indica una caratterizzazione a base di tratti sia dell'unità ortografica, sia del contesto linguistico immediatamente circostante l'unità ortografica in esame.

[2.1] $$p\left( Mtag_i \subset IM_i \mid C_{UO} \right)$$

Assegnare una $IM_j$ ad una unità ortografica significa calcolare la "$Mtag_i \subset IM_i$" che massimizzi la [2.1], come descritto nella [2.2].

[2.2] $$IM_j = \underset{\text{argmax}\left( Mtag_i \subset IM_i \right)}{p\left( \text{Mtag} \mid C_{UO} \right)}$$

Nel caso in cui l'analizzatore morfologico non abbia prodotto alcuna interpretazione valida per l'unità ortografica in esame, la [2.2] sarà priva dell'insieme di Mtag di partenza, come descritto dalla [2.3], dove "$PoS_i$" è una categoria morfosintattica scelta tra tutte le possibili categorie morfosintattiche definite nel Tagset.

[2.3] $$p\left( PoS_i \mid C_{UO} \right)$$

Il metodo della Massima Entropia prevede la definizione di un set di tratti morfosintattici

sufficienti a categorizzare ciascuna unità ortografica UO ed il rispettivo contesto linguistico.
La probabilità "*p*" di "Mtag" dato un determinato contesto linguistico " $C_{UO}$ " viene dunque calcolata sulla base di un training corpus utilizzando come unici vincoli probabilistici le distribuzioni dei tratti morfosintattici con cui è stata categorizzata ciascuna unità ortografica UO ed il rispettivo contesto " $C_{UO}$ " durante la fase di addestramento del modello.
Considerando per esempio un'unità ortografica UO che a livello di analisi morfologica abbia ricevuto due interpretazioni, sostantivo e verbo, quindi due potenziali set "Mtag" tra i quali dover scegliere ( NN@MS per sostantivo singolare e V_PP@MS per verbo participio maschile singolare); ipotizziamo che i tratti contestuali siano la presenza di un articolo in posizione " – 1 " rispetto all'unità ortografica UO e la presenza di un verbo in posizione " + 1 ".
Il metodo della Massima Entropia assegna ad ogni tratto "*j*" appartenente al contesto linguistico "C" , "$j \in$ C" , un peso relativo " $\boldsymbol{\alpha}_j$ Mtag " a seconda del ruolo che quel determinato tratto ha nell'assegnazione di una determinata "Mtag" all'unità ortografica UO in questione.
Nel nostro caso l'articolo in posizione " - 1 " rispetto all'UO ed il verbo in posizione " + 1 " avranno maggior peso nel categorizzare l'unità ortografica UO come sostantivo piuttosto che come verbo.

### 2.4.3 Descrizione dell'output prodotto dal Tagger

A livello di tagging l'output si presenta come una lista di unità ortografiche alle quali è stata associata a ciascuna di esse una sola tra le Tag proposte dal modulo precedente.
Il codice per l'annotazione adottato è lo stesso utilizzato dall'analizzatore morfologico.
Prendendo in esame la frase "*è **stato** visto a Tarquinia in compagnia*", contenente la parola "*stato*" sulla quale ho sopra discusso (vedi TABELLA 2.7), l'output del Tagger si presenterà come segue:

| | | | |
|---|---|---|---|
| *e'* | *essere* | *ALLSMALL* | *V_FIN@!S3PI* |
| ***stato*** | *essere* | *ALLSMALL* | *V_PP@MS!RP* |
| *visto* | *vedere* | *ALLSMALL* | *V_PP@MS!RP* |
| *a* | *a* | *ALLSMALL* | *PREP@!!!!!* |
| *Tarquinia* | *tarquinia* | *FIRSTCAPITALIZED* | *NN_P@__!!!* |
| *in* | *in* | *ALLSMALL* | *PREP@!!!!!* |
| compagnia | compagnia | ALLSMALL | NN@FS!!! |

Ogni riga corrisponde all'analisi di una singola unità ortografica.
Le informazioni per ogni analisi sono disposte su quattro colonne:

- Prima colonna: viene replicata ciascuna unità ortografica presente nel testo oggetto di studio.
- Seconda colonna: ad ogni unità ortografica viene associato il lemma a cui essa si riconduce.
- Terza colonna: contiene informazione riguardante la forma ortografica con cui l'unità si presenta.
- Quarta colonna: ad ogni unità ortografica viene associata una sola Tag, scelta dal Tagger in base a criteri statistici.

In questo caso, per la parola "*stato*" l'analisi scelta tra le tre proposte dall'analizzatore morfologico è quella di forma "Participio Passato" del verbo "Essere".

### 2.4.4 ILC-UniPi Tagger nell'ambito di EVALITA 2007

Molto interessanti sono stati i risultati raggiunti da ILC-UniPi Tagger nell'ambito di EVALITA 2007 (Tamburini, 2007), un'iniziativa finalizzata alla valutazione dei sistemi per il Part of Speech Tagging automatico della lingua italiana.
Sia il Tagset sia i dati sui quali addestrare e valutare i sistemi di Part of Speech Tagging sono stati messi a disposizione dagli organizzatori, i quali hanno fornito un ambiente di lavoro comune ai vari gruppi di studiosi che hanno preso parte all'iniziativa (undici in totale).
I dati forniti consistono in vari documenti appartenenti principalmente ai generi giornalistici e narrativi, per un totale di circa 151.000 tokens. Questi sono stati suddivisi in

due set:

1. Un set di sviluppo[25] composto da 133.756 tokens utilizzato dai sistemi per la fase di training.
2. Un set di test[26] delle dimensioni di 17.313 tokens utilizzato per la valutazione dei sistemi.

Come metriche di valutazione dei vari tagger sono state prese in considerazione:

- "Accuratezza generale", calcolata in base al numero di Tags giuste assegnate rispetto al numero totale dei tokens presenti nel Test Corpus.
- "Accuratezza nel trattamento delle parole sconosciute", ovvero di quelle parole presenti nel Test Corpus ma non presenti nel Training Corpus (e quindi mai viste dal Tagger prima della fase di test).

I risultati ottenuti sono stati piuttosto elevati e paragonabili a quello che nel campo del Part of Speech Tagging è lo stato dell'arte per la lingua inglese.
ILC-UniPi Tagger si è posizionato in terza posizione raggiungendo un'accuratezza generale di tagging pari al 97,65% ed un'accuratezza nel trattamento delle parole sconosciute pari al 94,12%.
I primi due sistemi classificati hanno raggiunto rispettivamente un'accuratezza generale di tagging di 98,04% e 97,89%, ed un'accuratezza nel trattamento delle parole sconosciute del 95,02% e 94,34%.
Le prestazioni raggiunte dai primi tre sistemi classificati sono comunque vicinissime tra loro.

25 Corpus di addestramento.
26 Corpus di Test

# 3 Analisi degli errori

## 3.1 Caratteristiche dei corpora utilizzati per il training e per la valutazione delle prestazioni di ILC-UniPi Tagger

Il corpus preannotato utilizzato per il primo addestramento di ILC-UniPi Tagger è stato costruito con articoli estratti dal quotidiano "Repubblica", ed ha una dimensione pari a 231.433 tokens.

Tipico dello stile giornalistico, "l'italiano scritto" di Repubblica da una parte cerca di seguire le regole e le norme codificate dalle moderne grammatiche, dall'altra, per fini di chiarezza comunicativa, tende ad evitare costrutti sintattici complessi e di non immediata chiarezza. I principali tratti che lo caratterizzano sono:

- Frasi sintatticamente compiute
- Sintassi chiara e sciolta.
- Assenza di elementi morfosintattici marcatamente regionali.

Finita la fase di addestramento sono stati forniti al tagger nuovi dati da annotare automaticamente.

E' buona norma verificare la prestazioni del tagger su porzioni di testi che presentino caratteristiche linguistiche diverse da quelle dei testi utilizzati per la fase di training.

Ciò che più conta per un tagger è la capacità di "generalizzare", l'essere il più possibile adattabile a qualsiasi dominio linguistico.

Se fornissimo al tagger dati "troppo adatti", per esempio testi estratti dal quotidiano "Il Corriere della Sera", probabilmente il tagger produrrebbe risultati eccellenti, con una soglia di errore minima, ma sarebbero risultati non realistici, poiché i dati forniti avrebbero caratteristiche linguistiche molto simili ai testi usati per l'addestramento.

Il corpus privo di annotazione morfosintattica utilizzato per verificare le prestazioni di ILC-UniPi Tagger prende il nome di "Informative" ed è composto in parte da verbali redatti dalle forze dell'ordine, in parte da trascrizioni di dialoghi spontanei avvenuti tra determinati interlocutori.

Il linguaggio con cui sono redatti i verbali si articola in strutture frasali complesse e periodi molto lunghi, qualche volta poco chiari. Viene spesso fatto ricorso a deissi testuali (*Per*

*quanto sopra...*), costruzioni sintattiche impersonali (*...si riportano*) e all'uso del participio presente. In rari casi capita che la punteggiatura sia stata disposta in maniera non corretta, stravolgendo il senso della frase. Riporto un frammento di testo a titolo di esempio:

> *"In data odierna, giusti ordini superiori, i militari in epigrafe indicati hanno effettuato in Roma e Fiumicino sopralluoghi nei confronti di sedi di società nonché di residenze e/o domicili di personaggi presumibilmente coinvolti nel procedimento indicato."*

Per quanto riguarda la parte delle trascrizioni del parlato, si tratta di dialoghi informali e spontanei.
Le varie battute si articolano in alcuni casi in periodi brevi, a volte ridotti a monosillabi, in altri in periodi lunghi e spesso mal organizzati. Gli interlocutori fanno abbondante uso di forme dialettali.
Riporto un frammento di dialogo tra l'interlocutore A e l'interlocutore B a titolo di esempio:

> A*: la legge che stanno a fà adesso..... non c'ha mica la coscienza a posto...eh.....*
> B*: ..... meno fortuna....*
> A*: ....ma come fai a cambiare.... ma se te domani diventi giudice...diventi giudice..... le tue idee politiche che uno c'avrà.... come fai ad accantonarle...se domani sai che quel tizio che sta sotto procedimento ha fatto male, ha danneggiato, ha parlato male della tua idea politica poi automaticamente sei ...inc/le... come cazzo fai... è una legge de natura....*

## 3.2 Analisi degli errori commessi da ILC-UniPi Tagger

Per valutare la capacità di generalizzare del tagger addestrato sul corpus di "Repubblica"[27], per capire dove e in che modo esso sbagli, è stato utilizzato il corpus delle "Informative" sopra descritto, delle dimensioni di 450.000 tokens circa.
La porzione del corpus "Informative" da me revisionata ha una dimensione pari a 28.233 tokens[28].

27 Delle dimensioni di circa 230.000 tokens
28 32.321 tokens considerando anche le Tags del NER

Gli errori che ho riscontrato analizzando le uscite del tagger sono riconducibili ai tre livelli di analisi da me descritti: Tokenizzazione, Analisi Morfologica, Tagging.
Tutti i casi di errori riscontrati nelle fasi di analisi precedenti quella di tagging, vanno inevitabilmente ad influenzare le prestazioni del Tagger.

## 3.3 Errori riconducibili al livello di segmentazione del testo (Tokenizzazione)

Riporto alcuni degli errori riconducibili al livello di tokenizzazione del testo.
Gli errori sono dovuti al fatto che le regole scritte per la segmentazione di un testo in token non sempre riescono a gestire tutte le casistiche che si presentano nei testi, poiché potenzialmente infinite e non predicibili a priori.

- Stringa di caratteri: =SEDE=

  Il carattere " = " anteposto e post posto a ridosso della parola "SEDE" ha solo la funzione di orpello grafico utilizzato, da chi ha redatto il documento, solo per fini estetici.

  La stringa di caratteri " =SEDE= " è stata segmentata dal tokenizzatore in due distinte unità testuali:

  1 - =SEDE

  2 - =

  Tale segmentazione impedisce ai livelli di analisi successiva di processare correttamente la stringa "SEDE", poiché il carattere " = " precedente la "S" ne ostacola il riconoscimento.

  L'errore di tokenizzazione è sistematico e la segmentazione del testo viene sbagliata ogni qual volta si presenti tale fenomeno.

  Il tokenizzatore è stato aggiornato in modo tale da separare il carattere " = " dalla stringa di caratteri cui è attaccato.

  Dopo l'aggiornamento il tokenizzatore riesce a segmentare la stringa " =SEDE " in tre unità testuali distinte:

  " = " "SEDE" " = "

- Stringa di caratteri: =R O M A

  In questa sequenza di caratteri, oltre al caso del carattere " = " di cui ho precedentemente discusso si presenta un altro problema: la parola "Roma" è stata scritta con un carattere di spaziatura tra lettera e lettera, per fini estetici stabiliti da chi ha redatto il testo. Dopo l'aggiornamento del tokenizzatore il problema del carattere " = " viene risolto, ma non è possibile in alcun modo ricostruire l'unità testuale "ROMA", la quale viene segmentata lettera per lettera ed analizzata come tale dai successivi livelli di elaborazione del testo.

- Stringa di caratteri: V°

  Fuori da qualsiasi contesto, questa stringa formata da due soli caratteri viene spesso utilizzata nelle "Informative". Anche questi caratteri hanno solo funzione di orpelli grafici utilizzati da chi ha redatto il documento solo per fini estetici. Il tokenizzatore segmenta i due caratteri in due unità testuali distinte:

  1 - V

  2 - °

  Abbiamo deciso di aggiornare il tokenizzatore in modo che i due caratteri non vengano separati, poiché se il primo da solo potrebbe avere senso, il secondo, " ° ", da solo non ne avrebbe. Lasciando il carattere " ° " unito al carattere che lo precede, la stringa "V°" potrebbe essere interpretata come una variante grafica del numero romano o ordinale "Quinto".

  Il carattere " ° " viene utilizzato per costruire i numeri ordinali 1°, 2°, 3°.... (primo, secondo, terzo). In questi casi il tokenizzatore non separa mai " ° " dal numero che lo precede.

- Stringa di caratteri: 5.codice

  Questa stringa avrebbe dovuto avere un carattere di spaziatura tra il punto ed il primo carattere della stringa "codice". In questo caso siamo difronte ad un errore di battitura involontario, dimostrato dal fatto che i restanti punti dell'elenco numerato sono stati scritti tutti in maniera corretta:

  4. intestazione trattazione

**5.codice**

6. classifica

7. OGGETTO

Il tokenizzatore non riesce a separare correttamente “codice” dal punto, di conseguenza l'unità ortografica non viene analizzata correttamente da nessuno dei successivi livelli di analisi del testo. Diversamente, tutti gli altri casi che compongono l'elenco numerato ricevono la corretta analisi in tutti i successivi livelli di analisi linguistica del testo.

La soluzione di separare dal punto i caratteri che si trovano alla sua destra e alla sua sinistra non è ipotizzabile, poiché in molti casi le parole contenenti punti al loro interno risultano essere acronimi (U.S.A, S.p.a ...)

Altri casi simili da me riscontrati sono[29]:

10.firma
49'NORD
28'est
All.nr (Allegato nr)

Questi casi, dovuti alcune volte ad errori di battitura, altre volte a convenzioni di scrittura adottate a discrezione di chi redige il testo, sono potenzialmente illimitati poiché non seguono regole ben precise, ed a livello di tokenizzazione non è possibile formulare regole che riescano a descriverli tutti. Ho corretto manualmente i casi riscontrati.

- Stringa di caratteri terminati con più punti consecutivi.

  La questione del punto attaccato ad una sequenza di caratteri non viene risolta dal tokenizzatore, poiché a questo livello di analisi del testo non è possibile stabilire se un punto faccia parte o meno della stringa di caratteri che segue.

  Nel caso di una stringa come “adesso.....” il tokenizzatore replica la sequenza invariata al successivo livello di analisi, il quale interpreterà “adesso.....” come

  1 - adesso.

29 Per brevità ne cito solo alcuni a titolo esemplificativo

2 - ....

L'analizzatore morfologico stacca tutti i punti tranne quello direttamente attaccato alla parola.

Di conseguenza i punti isolati vengono poi riconosciuti come punteggiatura, ma la prima parte della stringa non viene riconosciuta a causa del punto finale.

In questi casi, ritengo opportuno che l'intera stringa di punti venga separata dal resto della sequenza di caratteri fin dalla fase di tokenizzatore, così da permettere ai livelli successivi di analisi di associare alla stinga priva di punti una corretta analisi.

- Stringa di caratteri: /

  Nel corpus "Informative" ho spesso riscontrato la presenza di stringhe di caratteri contenenti al loro interno il carattere " / ", come per esempio "hotel/albergo", "e/o". Difronte a queste sequenze di caratteri il tokenizzatore non opera nessuna segmentazione ma replica l'unità ortografica al modulo di analisi morfologica, il quale non riesce a fornire nessuna interpretazione per la stringa in esame. L'ipotesi di dividere a livello di tokenizzazione le stringhe congiunte tramite il carattere " / " non è attuabile, poiché tale carattere viene spesso utilizzato per la costruzione di sigle indivisibili, quali date (12/02/2008), numeri civici (12/B), codici identificativi (decreto 2353/01-21) ed altre sigle alfa-numeriche recanti significati referenziali utili per il riconoscimento delle entità nel testo (NER).

## 3.4 Errori riconducibili al livello di analisi morfologica del testo (Magic)

Riporto alcuni degli errori riconducibili al livello di analisi morfologica del testo.

Gli errori da me riscontrati e segnalati dipendono dal fatto che, per varie ragioni, non sempre il formario dispone delle informazioni necessarie per catalogare tutte le unità ortografiche che si trovano in un testo. Revisionare ed aggiornare il formario è molto importante perché:

- Se un'unità ortografica riceve un'interpretazione morfologica inesatta, l'annotazione al livello di tagging risulterà sbagliata a prescindere, poiché il Tagger considera solo le analisi prodotte da Magic.

- Se un'unità ortografica riceve un'interpretazione morfologica incompleta o sovrabbondante, ciò comporterà per il Tagger un incremento della probabilità di sbagliare poiché per il calcolo della probabilità della Tag corretta, il Tagger considera tutte le analisi proposte da Magic, giuste o sbagliate che siano.
- Se un'unità ortografica non riceve nessuna interpretazione morfologica, il Tagger dovrà comunque cercare di attribuire la Tag corretta pur non avendo la base di partenza fornita da Magic.

### 3.4.1 Il problema delle abbreviazioni, degli acronimi e dei nomi propri

Le abbreviazioni, gli acronimi ed i nomi propri non presenti nel formario di Magic non vengono riconosciuti e come PoS a livello di analisi morfologica ricevono il valore "ND_POS@". A livello di tagging, risultando unità ortografiche sconosciute, riceveranno la PoS più probabile a seconda del contesto circostante, quindi non sempre la stessa.
Queste forme non rispettano le tradizionali regole morfologiche che strutturano il lessico di una lingua ed è quindi impossibile formulare generalità in grado di descriverle tutte.
L'unica soluzione è quella di arginare il più possibile il problema aggiungendo queste forme al formario di Magic.

### 3.4.2 Errori riconducibili alla parte dei verbali contenuti nelle informative: errori di tipo ortografico

Sono quei tipi di errori compiuti da chi ha redatto in testo, i quali, essendo imprevedibili ed inclassificabili a priori, non possono essere gestiti dall'analizzatore morfologico.

- Stringa di caratteri: ne

  L'uscita di Magic per tale unità ortografica è la seguente

  *ne SIMPLE ALLSMALL ne ne PRON_P@__3!!A! N ne ne ADV@!!!!!!P N ne ne PREP@!!!!!!! N*

  L'uscita non prevede l'analisi di "ne" come congiunzione "CONJ@".

  Questo è dovuto al fatto che "ne" intesa come congiunzione andrebbe scritta in forma accentata, ovvero "né". Però anche "ne" preposizione deve essere scritta nella forma accentata "né"; Magic accetta "ne" preposizione anche senza la "e" accentata.

Secondo lo standard ortografico del DeMauro Paravia i quattro casi si risolvono in:

ne – avverbio (*«sei andato all'università?» «Sì, ne torno proprio adesso»)*

ne – pronome personale (*è molto legata al padre, ne parla sempre)*

né – preposizione (*l'intreccio ne "I Promessi Sposi")*

né – congiunzione (n*on ha risposto né sì né no)*

Il problema dell'interpretazione di "ne" come congiunzione è rilevante, poiché molto spesso chi scrive trascura il carattere " é " accentato, sia che si tratti di avverbio, congiunzione, pronome o preposizione. Questo errore è quindi dovuto ad una mancata conoscenza di quelle che sono le norme ortografiche.

Ritengo che sarebbe opportuno aggiornare il formario in modo che per tale unità ortografica venga ipotizzata anche l'interpretazione di congiunzione CONJ@.

- Stringa di caratteri: 2^

L'analizzatore morfologico interpreta tale stringa di caratteri come numero cardinale "due"

*2^ NUMERIC ND 2^ 2^ C_NUM@!!!!!!! N*

Chi scrive vuole invece rappresentare il numero ordinale "secondo", non il numero cardinale "due". L'interpretazione più probabile è la seconda, poiché non avrebbe molto senso scrivere un numero cardinale seguito dal carattere " ^ ". La formattazione adottata dallo scrivente è plausibile, di conseguenza il formario di Magic è stato aggiornato.

Adesso l'uscita di Magic per l'unità ortografica " 2^ " è la seguente

*2^ NUMERIC ND 2^ 2^ O_NUM@!!!!!!! N*

### 3.4.3 Errori riconducibili alla parte dei verbali contenuti nelle informative*:* uso non comune del linguaggio

Essendo quello di Magic un lessico per scopo generico, può capitare che alcune parole vengano classificate e riconosciute solo nelle loro accezioni più comuni, e non nelle accezioni tecniche-specialistiche che possono assumere nel lessico italiano.

- Stringa di caratteri: li

  L'uscita di Magic per tale unità ortografica è la seguente

  *SIMPLE ALLSMALL li li PRON_P@MP3!!A! N*

  Magic classifica "li" solo come PRON_P@, e non come articolo determinativo ART_D@, poiché "li" in funzione di articolo è tipico del linguaggio specialistico-tecnico burocratese.

  Davanti ad una data "li" ha funzione di articolo determinativo, e non di pronome. Come nel caso di "*li 20 giugno 1952"*.

  Ritengo pertanto opportuno aggiornare il formario in modo che per tale unità ortografica venga ipotizzata anche l'interpretazione di "ART_D@", oltre a quella di "PRON_P"@[30].

- Stringa di caratteri: specifico

  L'uscita di Magic per tale unità ortografica è la seguente

  *specifico SIMPLE ALLSMALL specifico specificare V_FIN@!S1PI!! N specifico specifico ADJ@MS!!!!P N*

  Magic non prevede l'interpretazione di sostantivo NN@ per il termine "specifico", il quale viene utilizzato in funzione di sostantivo principalmente nei sottocodici[31] tecnico-specialistici dell'arte e della medicina.

- Stringa di caratteri: operanti

  L'uscita di Magic per tale unità ortografica è la seguente

  *operanti SIMPLE ALLSMALL operanti operante ADJ@_P!!!!P N operanti operare V_PP@_P!PP!! N*

  Magic non ipotizza per "operanti" l'interpretazione di sostantivo NN@, anche se nel contesto "nel rilasciare gli operanti" ha funzione di nome.

  Nel linguaggio comune, l'uso di "operanti" in funzione di nome è ormai diventato

30 Corretto in base alla definizione del dizionario DeMauro Paravia – Voce reperibile al http://demauroparavia.it/63380

31 Sottocodice, o lingua speciale: varietà particolare dell'italiano

obsoleto, ed è ad oggi confinato al codice burocratico o all'uso letterario.

### 3.4.4 Errori riconducibili alla parte dei verbali contenuti nelle informative: accezioni di uso comune non presenti nel formario di Magic

I seguenti errori sono la conseguenza del fatto che per alcune parole Magic non dispone di tutta l'informazione necessaria per una annotazione completa. Ritengo opportuno aggiornare il formario così da permettere una corretta analisi dei seguenti casi.

- Stringa di caratteri: civico

  L'uscita di Magic per tale unità ortografica è la seguente

  *civico SIMPLE ALLSMALL civico civico ADJ@MS!!!!P N*

  Tale unità ortografica riceve solo l'interpretazione di aggettivo, sebbene in moltissime situazioni "civico" venga utilizzato in funzione di aggettivo sostantivato, come nelle frasi

  - "presso tale civico"
  - "i militari rilevavano al suddetto civico"

  In questi casi "civico" dovrebbe ricevere la categoria grammaticale di nome NN@, e non quella di aggettivo ADJ@.

  Nella lingua italiana l'utilizzo di aggettivi in luogo di sostantivi è molto comune.

- Stringa di caratteri: emittente

  L'uscita di Magic per tale unità ortografica è la seguente

  *emittente SIMPLE ALLSMALL emittente emittente NN@FS!!!!P N*

  L'unità ortografica "emittente" viene interpretata solo come sostantivo e non come aggettivo ADJ@.

  In casi come "banca emittente", "ente emittente", "emittente" ha funzione di aggettivo ADJ@ e non di sostantivo.

- Stringa di caratteri: aver

  L'uscita di Magic per tale unità ortografica è la seguente

*aver SIMPLE ALLSMALL aver aver ND_POS@!!!!!!! N*

L'unità ortografica “aver” come forma tronca del verbo “Avere” non viene riconosciuta dall'analizzatore morfologico, così come non vengono riconosciute tutte le forme tronche degli altri verbi presenti nelle “Informative”. Nell'italiano comune, soprattutto nello scritto, i casi in cui i verbi vengono utilizzati nella loro forma tronca priva della vocale finale sono frequentissimi, soprattutto nell'italiano scritto.

- Stringa di caratteri: pertanto

  L'uscita di Magic per tale unità ortografica è la seguente

  *pertanto SIMPLE ALLSMALL pertanto pertanto ADV@!!!!!!P N*

  Tale unità ortografica viene considerata solo avverbio sebbene sia una congiunzione, come attestano i dizionari “DeMauro Paravia”, “Garzanti Linguistica” e il “Sabatini Coletti - Dizionario della Lingua Italiana”.

- Stringa di caratteri: grazie

  L'uscita di Magic per tale unità ortografica è la seguente

  *grazie SIMPLE ALLSMALL grazie grazia NN@FP!!!!P N grazie grazie INT@!!!!!!! N*

  “Grazie” nel linguaggio, sia parlato che scritto, è utilizzato spessissimo per la costruzione di “locuzioni preposizionali” quali “grazie a te”, “grazie al cielo”. In tali contesti “grazie” ha funzione di preposizione PREP@, non di nome o interiezione.

### 3.4.5 Errori riconducibili alla parte dei dialoghi contenuti nelle informative

Questi errori sono dovuti principalmente al fatto che i parlanti fanno abbondante utilizzo di termini dialettali regionali, termini gergali e neologismi che per ovvi motivi non sono presenti in un lessico di tipo generico come quello di Magic.

- Stringa di caratteri: mo, mò

L'uscita di Magic per tale unità ortografica è la seguente[32]

*mo SIMPLE ALLSMALL mo mo ND_POS@!!!!!!! N*
*mò SIMPLE ALLSMALL mò mò NN@M_!!!!P N*

In entrambi i casi si tratta di una forma dialettale regionale centro-meridionale, sinonimo di "adesso". Questa forma non viene riconosciuta come avverbio ADV, poiché tratti così marcati non sono contenuti nel formario di Magic.
Altri casi non facilmente identificabili perché marcati dal dialetto regionale di chi sta parlando sono[33]

oh *oh SIMPLE ALLSMALL oh oh INT@!!!!!!! N*

Utilizzato "oh" in luogo di "lo", nella frase "oh spero"
Ovviamente "oh" non può essere identificato e ricondotto al pronome "lo".

bancarli *bancarli SIMPLE ALLSMALL bancarli bancarli ND_POS@!!!!!!! N*

Nella frase "bisogna bancarli", riferito a dei soldi, nel senso di "bisogna riscuoterli in banca".
Dalla base del sostantivo "banca" il parlante ha coniato il verbo "bancare", inesistente nella lingua italiana.

Ed ancora:
mioni *mioni SIMPLE ALLSMALL mioni mioni ND_POS@!!!!!!! N*

"mioni" sta per "milioni.
buttà *buttà SIMPLE ALLSMALL buttà buttà ND_POS@!!!!!!! N*
"buttà" sta per "buttare".

## 3.5 Part of Speech Tagging: cause degli errori di annotazione

I seguenti errori sono principalmente dovuti al fatto che i pesi probabilistici stimati dal Tagger sul dominio linguistico di "Repubblica" non sempre sono generalizzabili ad un

32 La scrittura corretta è quella con la "ò" accentata.
33 Ne cito solo qualcuno a titolo esemplificativo, la lista sarebbe lunghissima.

dominio così diverso come quello delle “Informative”, soprattutto in presenza dei formali costrutti sintattici tipici del linguaggio burocratese e di quelli informali, discontinui ed irregolari tipici dei dialoghi delle informative.

### 3.5.1 Errori riconducibili al Tagger A

- Stringa di caratteri: Tributaria Contesto: “Polizia Tributaria”
  Analisi Morfologica *TRIBUTARIA SIMPLE ALLCAPITALIZED TRIBUTARIA tributaria NN_P@__!!!!! N tributaria tributario ADJ@FS!!!!P N*

  L'unità ortografica ortografica “Tributaria” ha generalmente funzione di aggettivo, ma in sintagmi come “Polizia Tributaria” in cui identifica un'istituzione, deve essere considerata nome proprio NN_P@__!!!
  Il Tagger ha classificato tutte le unità ortografiche “Tributaria” come aggettivo ADJ@, anche quando avevano funzione di nome proprio NN_P@.

- Stringa di caratteri: sita Contesto: “società sita”
  Analisi Morfologica *sita SIMPLE ALLSMALL sita sito ADJ@FS!!!!P N sita sitare V_FIN@!S3PI!! N sita sitare V_FIN@!S2PM!! N*

  Il Tagger classifica sempre l'unità ortografica “sita” come voce del verbo “sitare” (sinonimo di puzzare) e mai come aggettivo ADJ@ femminile di “sito” (nel senso di situato, ubicato).
  L'utilizzo di tale participio in funzione di aggettivo è tipico del linguaggio burocratico, notarile specialistico. I pesi calcolati su “Repubblica” associano un probabilità bassissima alla Tag ADJ@ per tale unità ortografica.

- Stringa di caratteri: che Contesto: “i militari riscontravano che”
  Analisi Morfologica *che SIMPLE ALLSMALL che che DET@__!!!!P N che che PRON@__!!!! P N che che CONJ_C@!!!!!!! N che che CONJ@!!!!!!! N*

  In tale contesto il Tagger classifica sempre il “che” come pronome PRON@ e mai come congiunzione CONJ@.

- Stringa di caratteri: marca Contesto: “autovettura marca Renault”
  Analisi Morfologica *marca SIMPLE ALLSMALL marca marcare V_FIN@!S3PI!! N marca marcare V_FIN@!S2PM!! N marca marca NN@FS!!!!P N*

  Il Tagger classifica correttamente l'unità ortografica “marca” sia come nome che come verbo in tutti i contesti, tranne in quelli in cui l'unità ortografica “marca” segua l'unità ortografica “autovettura” o “miscela”. In questi contesti il tagger sbaglia regolarmente attribuendo a “marca” la PoS di voce del verbo “marcare” anzi che quella di nome NN@.

- Stringa di caratteri: quale Contesto: - anteposto ad un nome -
  Analisi Morfologica *quale SIMPLE ALLSMALL quale quale ADV@!!!!!!P N quale quale ADJ@_S!!!!P N quale quale PRON@_S!!!!P N*

  “Quale/i” quando precede un nome ed ha funzione di aggettivo viene sistematicamente marcato come pronome, nonostante l'uscita del morfologico preveda ogni categoria grammaticale possibile per tale UO.
  Alcune uscite dove il tagger sbaglia:
  - con quale veicolo --> “PREP@ + *PRON@* + NN@” anzi che “PREP@ + *ADJ@* + NN@”
  - sotto quale sigla --> “PREP@ + *PRON@* + NN@” anzi che “PREP@ + *ADJ@* + NN@”
  - con quale nome --> “PREP@ + *PRON@* + NN@” anzi che “PREP@ + *ADJ@* + NN@”

- Stringa di caratteri: pochi Contesto: - anteposto ad un nome -
  Analisi Morfologica *pochi SIMPLE ALLSMALL pochi poco ADJ@MP!!!!P N pochi poco* PRON@MP!!!!

  Errore sistematico del Tagger il quale marca tale unità ortografica con l'etichetta PRON@, e mai come ADJ@, come dovrebbe nei seguenti casi:

  Dopo dopo PREP@!!!!!
  *pochi poco PRON@MP!!! anzi che ADJ@*
  minuti minuto NN@MP!!!

  addirittura addirittura ADV@!!!!!

*pochi poco PRON@MP!!! anzi che ADJ@*
giorni giorno NN@MP!!!

- Stringa di caratteri: stessa/e/i Contesto: "riferito dalla stessa"
"alle stesse condizioni"
"agli stessi riconducibili"

Analisi Morfologica ***stessa*** *SIMPLE ALLSMALL stessa stessere V_FIN@!S_PC!! N stessa stesso ADJ@FS!!!!P N stessa stesso PRON@FS!!!!P N*

***stesse*** *SIMPLE ALLSMALL stesse stessere V_FIN@!S3PI!! N stesse stesso ADJ@FP!!!!P N stesse stesso PRON@FP!!!!P N stesse stare V_FIN@!S3IC!! N*

***stessi*** *SIMPLE ALLSMALL stessi stessere V_FIN@!S2PI!! N stessi stessere V_FIN@!S2PM!! N stessi stesso ADJ@MP!!!!P N stessi stesso PRON@MP!!!!P N stessi stare V_FIN@!S1IC!! N stessi stare V_FIN@S2IC!! N*

Per l'unità ortografica "stessa" il tagger generalmente riesce ad attribuire la Tag corretta, tranne in alcuni contesti in cui viene attribuita la PoS aggettivo ADJ@ anzi che quella di pronome PRON@. In rari casi invece (tre casi su un corpus di circa 450.000 tokens) il Tagger identifica l'unità ortografica "stessa" come voce del verbo "Stessere" (disfare). In questi casi l'unità ortografica "stessa" è sempre preceduta da un nome o da un pronome.

Per l'unità ortografica "stesse" la percentuale di errore è, al contrario di "stessa", altissima.

Infatti il Tagger attribuisce a "stesse" la Tag aggettivo ADJ@ nella stragrande maggioranza dei casi e dei contesti. Nel corpus delle "Informative" l'unità ortografica "stesse" ha in molti casi funzione di pronome PRON@, ma solo due volte riceve tale etichetta.

Anche l'unità ortografica "stessi" riceve spessissimo la PoS di aggettivo ADJ@ anzi che quella di pronome PRON@.

- Stringa di caratteri: Quest' Contesto: “quest'ultimo”
  Analisi Morfologica *quest' ELISION ALLSMALL quest' questo DET@_S!!!!P E quest' questo PRON@_S!!!!P E*

  “Questo”, nella sua forma elisa “ quest' “ viene marcato dal Tagger sempre come aggettivo dimostrativo DET@, e mai come pronome dimostrativo PRON@. Nel contesto “quest'ultimo”,
  il tagger applica la seguente analisi:

  | | | |
  |---|---|---|
  | *Quest'* | *ques'* | *DET@* |
  | *ultimo* | *ultimo* | *NN@* |

  Anche l'unità ortografica successiva di conseguenza riceve una Tag non corretta, poiché la scelta del Tagger viene influenzata dall'unità che precede quest'ultima.

- Stringa di caratteri: certificato Contesto: preceduta da “altro”
  Analisi Morfologica *certificato SIMPLE ALLSMALL certificato certificare V_PP@MS!RP!! N certificato certificato NN@MS!!!!P N*

  Quando tale unità ortografica ha funzione di sostantivo, il Tagger la marca sempre correttamente tranne nei casi in cui essa venga preceduta dall'unità ortografica “altro”.
  “Altro” per errore riceve sempre la PoS di pronome PRON@ e mai quella di aggettivo ADJ@.
  In tale contesto la scelta della corretta Tag per l'unità ortografica “certificato” viene sbagliata perché influenzata dalla Tag attribuita ad “altro”.

| **Tag sbagliate** | | | **Tag corrette** | | |
|---|---|---|---|---|---|
| altro | altro | *PRON@MS!!!* | *altro* | *altro* | *ADJ@MS!!!* |
| *certificato* | *certificare* | *V_PP@MS!RP* | *certificato* | *certificato* | *NN@MS!!!* |

- Stringa di caratteri: sia Contesto: “sia nei meccanismi di associazione [...] sia in quelli [...]”

Analisi Morfologica *sia SIMPLE ALLSMALL sia sia CONJ_C@!!!!!!! N sia essere V_FIN@! S_PC!! N*

Tale unità ortografica viene spesso marcata con una Tag inadatta al contesto in cui essa si trova.

Nel contesto sopra riportato viene identificata una volta come congiunzione CONJ_C@ e una volta come verbo V_FIN@.

Non ho riscontrato una regolarità precisa dietro all'errore, il Tagger sbaglia poiché i pesi probabilistici calcolati sul corpus di "Repubblica", per tale unità ortografica non sono generalizzabili, al corpus delle "Informative".

- Stringa di caratteri: spagnola Contesto: "la società sia spagnola che italiana"

Analisi Morfologica *spagnola SIMPLE ALLSMALL spagnola spagnolo ADJ@FS!!!!P N spagnola spagnola NN@FS!!!!P N*

Ho riscontrato una regolarità dietro l'assegnazione della PoS che il Tagger attribuisce all'unità ortografica "spagnola".

Se essa è preceduta da un sostantivo NN@, allora il Tagger attribuisce a "spagnola" la PoS corretta, cioè quella di aggettivo ADJ@.

"società spagnola"

Se invece "spagnola" è preceduta da un aggettivo ADJ@, il Tagger attribuisce la PoS sostantivo NN@, anche se non si tratta di sostantivo.

"utenza telefonica spagnola"

In due casi in cui "spagnola" è preceduta da un verbo, riceve la PoS di sostantivo NN@.

- Stringa di caratteri: "sia spagnola che italiana" Contesto: "sia spagnola che italiana"

Riporto una serie unità ortografiche alle quali il Tagger ha assegnato Tags non corrette

| | | |
|---|---|---|
| *sia* | *essere* | *V_FIN@!S_PC* |
| *spagnola* | *spagnola* | *NN@FS!!!* |

| | | |
|---|---|---|
| *che* | *che* | *PRON@__!!!* |
| *italiana* | *italianare* | *V_FIN@!S3PI* |

L'errore parte dall'unità ortografica “sia”, sulla quale ho sopra discusso. La Tag attribuita a “sia” influenza la scelta della Tag per “spagnola”, l'unità ortografica che segue “sia”. Come sopra detto, quando “spagnola” viene preceduta da un verbo riceve la PoS di sostantivo NN@.

Il “che” quando segue un sostantivo ha una grandissima probabilità di ricevere la PoS di pronome PRON@ ( nell'intero corpus delle “Informative” la PoS ricevuta è quasi sempre quella di pronome, e quasi mai quella di congiunzione CONJ@ ).

L'unità ortografica “italiana” viene quasi sempre marcata correttamente dal Tagger, ma questo caso è singolare, poiché in tutto il corpus delle “Informative” questa è l'unica occasione in cui essa venga preceduta da un pronome. Inoltre il verbo “italianare” è una variante molto rara e di basso uso del verbo “italianizzare”.

- Stringa di caratteri: precedenti Contesto: - preceduto da preposizione -
  Analisi Morfologica *precedenti SIMPLE ALLSMALL precedenti precedere V_PP@_P!PP!! N precedenti precedente ADJ@_P!!!!P N precedenti precedente NN@MP!!!!*

Il Tagger assegna a questa unità ortografica sempre la Tag corretta tranne nei casi in cui essa sia preceduta da una preposizione. In tali contesti, anche se “precedenti” ha funzione di sostantivo NN@, riceve quasi sempre PoS di aggettivo ADJ@. Poiché ho riscontrato una manciata di casi, se ben pochissimi rispetto a quelli in cui il Tagger sbaglia, nei quali “precedenti” riceve la corretta PoS anche se preceduto da una preposizione, ritengo che l'errore sia dovuto al fatto che i persi probabilistici calcolati su “Repubblica”, per questa unità ortografica siano fortemente sbilanciati a favore dell'interpretazione di “precedenti” come “aggettivo” nei contesti sopra descritti.

- Stringa di caratteri: grazie Contesto: - locuzioni preposizionali -
  Analisi Morfologica *grazie SIMPLE ALLSMALL grazie grazia NN@FP!!!!P N grazie grazie INT@!!!!!!!*

Poiché il “grazie” non riceve a livello morfosintattico la PoS di preposizione PREP@, nelle costruzioni di locuzioni preposizionali quali “grazie a voi”, “grazie al cielo”, dove “grazie” è preposizione, il Tagger sbaglia a priori nell'assegnare la Tag. Per coerenza in tali contesti ho sempre etichettato il “grazie” con la PoS interiezione INT@.

- Stringa di caratteri: addì Contesto: PUNCT@ + addì + C_NUM@
  Analisi Morfologica *addì SIMPLE ALLSMALL addì addì ADV@!!!!!!P N addì addire V_FIN@!S3RI!! N*

  In tutto il corpus delle “Informative” questa unità ortografica si presenta 47 volte sempre preceduta da un carattere di punteggiatura e seguita da un numero.
  Di queste 47 volte, il Tagger la classifica come avverbio soltanto 7 volte, mentre negli altri 40 casi viene classificata in maniera non corretta, e cioè come voce del verbo “addire”.
  “Addì” è una forma avverbiale tipica del linguaggio burocratese di raro uso in un linguaggio comune come quello di “Repubblica”; ritengo che l'errore sia dovuto al fatto che i persi probabilistici calcolati su “Repubblica”, per questa unità ortografica siano fortemente sbilanciati a favore dell'interpretazione di “addì” come voce verbale piuttosto che come avverbio.

### 3.5.2 Errori del Tagger A – Aggettivo vs Verbo Participio

Il tagger sbaglia spesso l'attribuzione della PoS nei casi in cui l'unità ortografica in esame possa assumere sia il valore di Aggettivo che di Verbo Participio.
Ciò è dovuto al fatto che la distinzione grammaticale tra participio-aggettivo e participio-verbo in alcuni casi può non essere così immediata.
Esistono contesti nei quali non è facile stabilire se ad una parola vada associata la PoS di aggettivo ADJ@ o quella di verbo participio V_PP@ (esclusi i casi in cui un participio venga utilizzato per la costruzione di tempi verbali composti). Non ho riscontrato regolarità dietro a questi tipi di errori, i quali non ricorrono con sistematicità.

- Stringa di caratteri: avente Contesto: “agenzia immobiliare avente come rappresentante”

  Analisi Morfologica *avente SIMPLE ALLSMALL avente avente ADJ@_S!!!!P N avente avere V_PP@_S!PP!! N*

  L'unità ortografica “avente” viene etichettata come ADJ@ anzi che come verbo V_PP@.

  Costruzioni come quella della frase sopra citata sono tipiche del linguaggio burocratese.

- Stringa di caratteri: metallizzato Contesto - funzione aggettivale -

  Analisi Morfologica metallizzato SIMPLE ALLSMALL metallizzato metallizzare V_PP@MS! RP!! N metallizzato metallizzato ADJ@MS!!!!P N

  L'unità ortografica viene sempre classificata come verbo V_PP@, ricondotta al lemma “metallizzare” in contesti quali “argento metallizzato”, “grigio metallizzato”. La PoS corretta è invece quella di aggettivo ADJ@.

  Essendo spesso preceduto da una PoS aggettivo ADJ@, il Tagger calcola che la PoS più probabile per “metallizzato” sia quella di verbo participio V_PP@.

### 3.5.3 Errori del Tagger A – Nome vs Verbo Participio

Anche nella distinzione tra verbo participio V_PP@ e nome NN@, il Tagger sbaglia spesso.

Nei casi da me riscontrati il tagger risulta più incline ad attribuire la PoS di verbo participio V_PP@ a discapito di quella di nome NN@. Non ho comunque riscontrato regolarità che giustifichino una sistematicità dell'errore.

- Stringa di caratteri: giunta Contesto “ ivi giunta il”

  Analisi Morfologica *giunta SIMPLE ALLSMALL giunta giungere V_PP@FS!RP!! N giunta giuntare V_FIN@!S3PI!! N giunta giuntare V_FIN@!S2PM!! N giunta giunta NN@FS!!!!P N*

  Il Tagger etichetta “giunta” come nome NN@ anzi che come verbo participio V_PP@.

- Stringa di caratteri: data Contesto " data inizio data fine"
  Analisi Morfologica *data SIMPLE ALLSMALL data dare V_PP@FS!RP!! N data datare V_FIN@!S3PI!! N data datare V_FIN@!S2PM!! N data dato ADJ@FS!!!!P N data data NN@FS!!!!P*

  Nella frase "codice fiscale data inizio data fine", l'unità ortografica "data" riceve la Tag di verbo participio V_PP@ anzi che di nome NN@.
  La costruzione sintattica della frase è tipica del linguaggio burocratico.

- Stringa di caratteri: condotta Contesto " la micra rossa condotta"
  Analisi Morfologica *condotta SIMPLE ALLSMALL condotta condurre V_PP@FS!RP!! N condotta condotta NN@FS!!!!P N condotta condottare V_FIN@!S3PI!! N condotta condottare V_FIN@!S2PM!! N*

  "Condotta" viene classificato come nome NN@ anzi che come verbo participio V_PP@.

- Stringa di caratteri: uscita Contesto " all'uscita del supermercato"
  Analisi Morfologica *uscita SIMPLE ALLSMALL uscita uscita NN@FS!!!!P N uscita uscire V_PP@FS! RP!! N*

  l Tagger assegna alla parola "uscita" l'etichetta V_PP@, verbo participio passato, anzi che NN@, "nome". Anche nel contesto " Traffico telefonico in entrata ed uscita" il Tagger assegna molto spesso la PoS di verbo participio V_PP@.
  Per questa unità ortografica il Tagger predilige l'interpretazione V_PP@ a discapito di NN@.

## 3.6 Errori riconducibili al Tagger B: annotazioni di unità ortografiche non riconosciute da Magic

- Stringa di caratteri: Nissan, Micra Contesto: "Nissan Micra"
  Analisi Morfologica *NISSAN SIMPLE ALLCAPITALIZED nissan nissan ND_POS@!!!!!!! N MICRA SIMPLE ALLCAPITALIZED nissan nissan ND_POS@!!!!!!! N*

Nei contesti in cui l'UO “Micra” viene preceduta da l'UO “Nissan”, etichettata come NN_P@, il tagger classifica come NN_P@ anche “Micra”, poiché calcola che dopo un NN_P@, “Micra” abbia forte probabilità di essere NN_P@ anche lei stessa. In altri contesti invece la PoS associata a “Micra” può variare. Per esempio nel contesto

| | | |
|---|---|---|
| la | lo | *ART_D@FS!!!* |
| MICRA | *NAN* | *NN@!!!!!* |
| si | si | *PRON_P@__3!!* |

“Micra” viene etichettata come nome NN@, poiché il tagger calcola che un'unità ortografica sconosciuta che si trovi tra un articolo determinativo ed un pronome personale, abbia una forte probabilità di essere un nome comune.

- Stringa di caratteri: te. Contesto: - in fine frase -

  Capita spesso nel testo di trovare l'unità ortografica “te” in fine frase, quindi seguita dal punto “te.”

  Analisi Morfologica *te. DOTEND ALLSMALL te te PRON_P@_S2!!Y! N . . PUNCT@!!!!!!! N te. te. ND_POS@!!!!!!! B . . PUNCT@!!!!!!! N*

  L'analisi giusta è ovviamente la prima, ma l'uscita presa in considerazione risulta essere la seconda[34], quella che ipotizza che la stringa “te.” possa esistere, anche se non viene riconosciuta, e che sia gestita quindi dal Tagger B.

  Il Tagger B in tutti i casi da me riscontrati, riesce sempre ad associare a tale UO la corretta PoS, cioè quella di pronome PRON@.

### 3.6.1 Forme tronche dell'infinito verbale

Poiché tutte le forme tronche dell'infinito dei verbi, presenti nell'intero corpus delle informative, non sono state identificate a livello di analisi morfologica, è possibile raggruppare questi errori in un'unica “classe”.

- Stringa di caratteri: aver Contesto: “dopo aver riferito”

  Analisi Morfologica *aver SIMPLE ALLSMALL aver aver ND_POS@!!!!!!! N*

34 Credo che tale scelta sia da imputare ad un bug di programmazione.

In questa forma del verbo “avere” la vocale finale viene elisa.

Il lemma al quale il Tagger riconduce “aver” è NAN, poiché a livello di analisi morfologia la forma non viene riconosciuta. Tale unità ortografica in tutto il corpus delle informative si presenta ben 211 volte, e nella maggior parte dei casi il Tagger riesce a marcarla con successo con la PoS di verbo forma non finita, V_NOFIN@.

L'accuratezza del Tagger B nel riconoscimento dell'unità ortografica “aver” è altissima, come descritto in TABELLA 3.1

TABELLA 3.1
Accuratezza nel riconoscimento dell'unità ortografica “aver”

| **UO** | **PoS** | **Casi attestati** | **Percentuale casi** | **Interpretazione** |
|---|---|---|---|---|
| aver | V_NOFIN@ | 204/211 | 96,68% | corretta |
| aver | V_FIN@ | 3/211 | 1,42% | errata |
| aver | ADJ@ | 3/211 | 1,42% | errata |
| aver | ADV@ | 1/211 | 0,47% | errata |

Altri casi simili:

Stringa di caratteri voler Contesto: “non voler parlare”

Analisi Morfologica *voler SIMPLE ALLSMALL voler voler ND_POS@!!!!!!! N*

TABELLA 3.2
Accuratezza nel riconoscimento dell'unità ortografica “voler”

| **UO** | **PoS** | **Casi attestati** | **Percentuale casi** | **Interpretazione** |
|---|---|---|---|---|
| voler | V_NOFIN@ | 8/9 | 88,88% | corretta |
| voler | NN@ | 1/9 | 11,11% | errata |

Stringa di caratteri poter Contesto: “per poter continuare”

Analisi Morfologica *poter SIMPLE ALLSMALL poter poter ND_POS@!!!!!!! N*

| TABELLA 3.3<br>Accuratezza nel riconoscimento dell'unità ortografica “poter” | | | | |
|---|---|---|---|---|
| **UO** | **PoS** | **Casi attestati** | **Percentuale casi** | **Interpretazione** |
| poter | V_NOFIN@ | 28/33 | 84,84% | corretta |
| poter | V_FIN@ | 5/33 | 15,15% | errata |

Stringa di caratteri far Contesto: “da far pensare”

Analisi Morfologica *far SIMPLE ALLSMALL far far ND_POS@!!!!!!! N*

| TABELLA 4.4<br>Accuratezza nel riconoscimento dell'unità ortografica “far” | | | | |
|---|---|---|---|---|
| **UO** | **PoS** | **Casi attestati** | **Percentuale casi** | **Interpretazione** |
| far | V_NOFIN@ | 92/97 | 94,84 | corretta |
| far | V_FIN@ | 1/97 | 3,09 | errata |
| far | PRON@ | 1/97 | 1,03 | errata |
| far | PREP@ | 3/97 | 1,03 | errata |

### 3.6.2 Trascrizione del parlato

Buona parte del testo che compone le informative è composto da trascrizioni di parlato spontaneo, il quale, oltre ad avere un struttura sintattica spesso non regolare, da un punto di vista lessicale sfocia spesso nel dialettale e nel gergale.

Prendiamo in esame la seguente frase di trascrizione di parlato estratta dal corpus “Informative”:

*“li potemo pià sti sordi poi non li posso bancà io”*

- Stringa di caratteri potemo Contesto - frase sopra citata -

  Analisi Morfologica: *potemo SIMPLE ALLSMALL potemo potemo ND_POS@!!!!!!! N*

  “potemo” è una forma dialettale che sta per “potremo”.

  L'unità ortografica viene gestita dal Tagger B, poiché non riconosciuta, ma ogni volta

etichettata con successo a livello di Tagging con la PoS verbo V_FIN@. (due occorrenze in tutto il corpus)

- Stringa di caratteri pià Contesto - frase sopra citata -
  Analisi Morfologica *pià SIMPLE ALLSMALL pià pià ND_POS@!!!!!!!*

  "pià" è una forma dialettale che sta per "pigliare, prendere".
  Si presenta una sola volta in tutto il corpus delle informative in mio possesso, e viene marcato erroneamente dal Tagger B con la PoS NN@.

- Stringa di caratteri sti Contesto - frase sopra citata -
  Analisi Morfologica *sti SIMPLE ALLSMALL sti sto ADJ@MP!!!!P N*

  "sti", in luogo di "questi" viene marcato correttamente, poiché al livello di analisi morfologica riceve la sola analisi di aggettivo ADJ@.

- Stringa di caratteri sordi Contesto - frase sopra citata -
  Analisi Morfologia *sordi SIMPLE ALLSMALL sordi sordo ADJ@MP!!!!P N sordi sordo NN@MP!!!!P N*

  Il tagger sceglie la Tag giusta, e cioè quella di nome NN@, anche se l'intero sistema ha in realtà associato il significante "sordi" al significato di "persona che non sente", e non al significato di "moneta"

- Stringa di caratteri bancà Contesto - frase sopra citata -
  Analisi Morfologica *bancà SIMPLE ALLSMALL bancà bancà ND_POS@!!!!!!!*

  "bancà" sta per "bancare" (non conosco il significato di questo neologismo dialettale)
  La forma non riconosciuta viene gestita dal Tagger B, il quale attribuisce la giusta PoS, ovvero quella di verbo di modo infinito V_NOFIN@.

  Altre forme tipiche del parlato che ho trovato nel testo sono:
  - "di" per "dire", il quale riceve la sola interpretazione di preposizione PREP@ a

livello di analisi morfologica.

- "vede" per "vedere" il quale viene etichettato come forma presente del verbo "vedere" anzi che come forma infinita del verbo vedere.

### 3.6.3 Unità ortografiche terminanti con un punto

Nel caso in cui un'unità ortografica termini con un punto di fine frase, questo deve essere separato da tale unità ortografica tranne nei casi in cui l'unità in esame costituisca un'abbreviazione o un acronimo, di cui il punto fa parte.
Una volta individuata l'unità ortografica come abbreviazione o come acronimo, ci sono due possibili contesti da considerare.

1. L'unità ortografica non si trova in fine frase, quindi il punto finale risulta essere solo una parte dell'unità in esame (sia che si tratti di abbreviazione, sia che si tratti di acronimo).

   In casi come questi, il tagger non deve staccare il punto dall'unità ortografica. Nell'esempio

   *.Company.*
   *I.S.E i.s.e NN_P@__!!!*
   *. . PUNCT@!!!!!*
   *S.R.L s.r.l NN_P@__!!!*
   *./Company.*

   Il punto è stato erroneamente separato dall'unità ortografica "I.S.E", poiché il tagger lo ha considerato un "punto di fine frase". Il fatto che l'unità ortografica seguente iniziasse con una lettera maiuscola ha ulteriormente influenzato il Tagger nel considerare il punto come di fine frase.

   La correzione giusta da apportare sarà la seguente:

   *.Company.*
   *I.S.E. i.s.e. NN_P@__!!!*
   *S.R.L s.r.l NN_P@__!!!*
   *./Company.*

2. L'unità ortografica si trova in fine frase, quindi il punto finale oltre ad essere parte dell'unità in esame ha anche funzione di punto di fine frase. In questo caso un'analisi che vede solo un punto di abbreviazione come:

   *MO.RE.NA.* *mo.re.na.* *NN_P@__!!!*

   Andrà corretta replicando un punto in funzione di "punto di fine frase"

   *MO.RE.NA.* *mo.re.na.* *NN_P@__!!!*

   . . *PUNCT@!!!!!*

## 3.7 Dizionari di riferimento[35]

Dizionari di riferimento sui quali mi sono basato per la determinazione di alcune PoS ambigue:

- De Mauro Paravia – *Dizionario della lingua italiana*.
  Reperibile all'indirizzo http://www.demauroparavia.it/
- Garzanti Linguistica
  Reperibile all'indirizzo http://www.garzantilinguistica.it/
- Sabatini Coletti - *Dizionario della Lingua Italiana*
  Reperibile all'indirizzo http://dizionari.corriere.it/dizionario_italiano.shtml

35 Edizioni online

# 4 Valutazione di ILC-UniPi Tagger

## 4.1 Test effettuati per la valutazione di ILC-UniPi Tagger – Primo ciclo

Lo scopo di questo primo ciclo di test è principalmente quello di verificare quanto il tagger addestrato sul dominio linguistico di Repubblica sia adattabile al dominio linguistico delle Informative.

Ai fini della valutazione, i risultati ottenuti dal Tagger addestrato sul dominio di Repubblica sono stati confrontati con quelli ottenuti dal Tagger addestrato sul dominio delle Informative.

Il corpus adottato per il test è stato il "Test Corpus Informative" delle dimensioni di 4280 tokens.

I pesi probabilistici utilizzati per l'annotazione automatica del "Test Corpus Informative" sono stati calcolati effettuando tre distinti cicli di addestramento su tre diversi corpora:

1. Il Tagger è stato addestrato sul corpus di "Repubblica" delle dimensioni di 231.433 tokens, e testato sul "Test Corpus Informative" delle dimensioni di 4280 tokens.
2. Il Tagger è stato addestrato sul corpus "Informative" delle dimensioni di 23.953 tokens, e testato sul "Test Corpus Informative" delle dimensioni di 4280 tokens.
3. Il Tagger è stato addestrato sulla somma dei due corpora "Repubblica" + "Informative", per una dimensione totale pari a 255.386 tokens, e testato sul "Test Corpus Informative" delle dimensioni di 4280 tokens.

Il ciclo si suddivide in due fasi:

- Test n°1 – A
- Test n°1 – B

I dati utilizzati e le loro caratteristiche sono riassunti in TABELLA 4.1

TABELLA 4.1
Dati utilizzati per il primo ciclo di test di ILC-UniPi Tagger

| | **Nome** | **Dimensione** | **Dominio linguistico** |
|---|---|---|---|
| **Corpus di Training** | Repubblica | 231.433 tokens | Repubblica |
| | Informative | 23.953 tokens | Informative |
| | Repubblica + Informative | 255.386 tokens | Informative + Repubblica |
| **Corpus di Test** | Test Corpus Informative | 4280 tokens | Informative |

## 4.2 Test n°1 – A

Nel calcolo delle statistiche le Tags recanti PoS corrette ma Tratti Morfosintattici sbagliati, sono state considerate errori.

### 4.2.1 Risultati ottenuti da ILC-UniPi Tagger, Tagger A + Tagger B

I risultati ottenuti sono stati calcolati considerando le prestazioni generali di ILC-UniPi Tagger (Tagger A + Tagger B):

Il Tagger addestrato su “Repubblica” e valutato sul “Test Corpus Informative” ha raggiunto un'accuratezza dell' 89,67%, riuscendo ad annotare correttamente 3838 unità ortografiche su 4280. La percentuale di errore è stata del 10,33%, avendo attribuito 442 Tag sbagliate su 4280.

Il Tagger addestrato su “Informative” e testato sul “Test Corpus Informative” ha annotato correttamente 3876 unità ortografiche su 4280 raggiungendo un'accuratezza del 90,56%.

Confrontando questi due primi risultati si evince che, sebbene il corpus di addestramento di “Repubblica” sia dieci volte più grande di quello delle “Informative”, le prestazioni migliori sono state ottenute dal Tagger addestrato sul corpus più piccolo ovvero la dove corpus di addestramento e corpus di test appartenevano allo stesso dominio linguistico.
Teoricamente, maggiori sono le dimensioni di un corpus di addestramento, maggiori saranno i casi osservati dal Tagger in fase di training e di conseguenza migliore sarà la sua

capacità di generalizzare.

Guardando infatti il terzo risultato, vediamo che il Tagger addestrato sulla somma dei due corpora, “Informative” + “Repubblica”, e valutato sul “Test Corpus Informative”, raggiunge un'accuratezza del 91,52%, quindi, se pur di poco, superiore ai due casi precedenti.

I dati sopra descritti sono riassunti in TABELLA 4.2

TABELLA 4.2
Risultati ILC-UniPi Tagger – Prestazioni Tagger A + Tagger B valutato sul “Test Corpus Informative”

| | **Tagger addestrato su Repubblica** | **Tagger addestrato su Informative** | **Tagger addestrato su Informative + Repubblica** |
|---|---|---|---|
| Tags corrette assegnate | 3838/ 4280 (89,67%) | 3876/4280 (90,56%) | 3917/4280 (91,52%) |
| Tags Sbagliate assegnate | 442/ 4280 (10,33 %) | 404/ 4280 (9,44%) | 363/ 4280 (8,48%) |

### 4.2.2 Risultati ottenuti da ILC-UniPi Tagger, solo Tagger B

I risultati ottenuti sono stati calcolati considerando le prestazioni del solo Tagger B, deputato a gestire le parole sconosciute, cioè quelle unità ortografiche che a livello di analisi morfologica non hanno ricevuto alcuna interpretazione.

Nel “Test Corpus Informative” su un totale di 4280 unità ortografiche 269 sono parole sconosciute non presenti nel lessico di Magic:

Il Tagger addestrato su “Repubblica” ha raggiunto un'accuratezza del 65,05%, attribuendo la Tag corretta a 175 parole sconosciute su 269.

Il Tagger addestrato su “Informative” ha assegnato 202 Tag giuste su 269, raggiungendo un'accuratezza del 75,1%.

Il Tagger addestrato sulla somma dei due corpora “Repubblica” + “Informative” ha raggiunto un'accuratezza del 76,57%, avendo classificato correttamente 206 parole sconosciute su 269.

Il miglior risultato è stato ottenuto dal Tagger addestrato sulla somma dei due corpora, ma

il dato più rilevante è senz'altro il divario che separa i primi due risultati. Anche in questo caso il Tagger addestrato su un corpus delle dimensioni di circa 230.000 tokens come quello di “Repubblica” non risulta così adattabile al dominio delle informative quanto invece lo risulta il Tagger addestrato sul piccolo corpus delle “Informative” (delle dimensioni di appena 23.953 tokens), il quale ha ottenuto un incremento delle prestazioni di oltre il 10% in più rispetto al Tagger addestrato su “Repubblica”.
I dati sopra descritti sono riassunti in TABELLA 4.3

TABELLA 4.3
Risultati ILC-UniPi Tagger – Prestazioni solo Tagger B valutato sul “Test Corpus Informative”

| | **Tagger addestrato su Repubblica** | **Tagger addestrato su Informative** | **Tagger addestrato su Informative + Repubblica** |
|---|---|---|---|
| Tags corrette assegnate | 175/269 (65,05%) | 202/269 (75,10%) | 206/269 (76,57%) |
| Tags Sbagliate assegnate | 94/269 (34,95%) | 67/269 (24,90%) | 63/269 (23,43%) |

## 4.3 Test n°1 – B

La prestazioni del Tagger sono state valutate prendendo in considerazione solo le Part of Speech delle Tag assegnate, ed i rispettivi Tratti Morfosintattici, giusti o sbagliati che fossero, non hanno influenzato le statistiche dei risultati.

### 4.3.1 Risultati ottenuti da ILC-UniPi Tagger, Tagger A + Tagger B (Solo PoS)

I risultati ottenuti sono stati calcolati considerando le prestazioni generali di ILC-UniPi Tagger (Tagger A + Tagger B). Il test corpus utilizzato è sempre il medesimo “Test Corpus Informative”:

Il Tagger addestrato su “Repubblica” ha raggiunto in questa seconda fase del primo ciclo di test un'accuratezza dell 90,21%, annotando correttamente 3861 unità ortografiche su 4280.

Il Tagger addestrato su “Informative” ha raggiunto un'accuratezza del 91,26%

annotando correttamente 3906 unità ortografiche su 4280.

Il Tagger addestrato sulla somma dei corpora “Informative” + “Repubblica” ha assegnato 3946 Tag giuste su 4280, raggiungendo un'accuratezza del 92,19%.

Esaminando questi dati, riassunti in TABELLA 4.4, possiamo constatare che anche questa volta il Tagger addestrato sul corpus di “Repubblica” ha raggiunto raggiunto prestazioni inferiori rispetto a quelle raggiunte dal Tagger addestrato sul piccolo corpus delle “Informative”.

L'incremento delle prestazioni non è stato comunque elevatissimo in nessuno dei tre casi, se paragonato ai risultati descritti in TABELLA 4.2: le prestazioni del Tagger addestrato su “Repubblica” sono migliorate dello 0,54%, quelle del Tagger addestrato su “Informative” dello 0,7% ed infine quelle del Tagger addestrato su “Informative” + “Repubblica” dello 0,67%. Questi valori sono indice del fatto che il grado di accuratezza del Tagger nell'associare i tratti morfosintattici alle rispettive PoS è molto elevato.

TABELLA 4.4
Risultati ILC-UniPi Tagger solo PoS – Prestazioni Tagger A + Tagger B valutato sul “Test Corpus Informative”

| | **Tagger addestrato su Repubblica** | **Tagger addestrato su Informative** | **Tagger addestrato su Repubblica + Informative** |
|---|---|---|---|
| Tags corrette assegnate | 3861/ 4280 (90,21%) | 3906/4280 (91,26%) | 3946/4280 (92,19%) |
| Tags Sbagliate assegnate | 419/ 4280 (9,79 %) | 374/ 4280 (8,74%) | 334/ 4280 (7,81%) |

### 4.3.2 Risultati ottenuti da ILC-UniPi Tagger, solo Tagger B (solo PoS)

I risultati ottenuti sono stati calcolati considerando le prestazioni del Tagger B nell'assegnamento della sola PoS. Il test corpus utilizzato è sempre il medesimo “Test Corpus Informative”:

Il Tagger addestrato su “Repubblica” ha raggiunto un'accuratezza del 68,03%, annotando correttamente 183 parole sconosciute su 269.

Il Tagger addestrato su “Informative” ha raggiunto un'accuratezza del 77,69%,

annotando correttamente 209 parole sconosciute su 269.

Infine, il Tagger addestrato sui due corpora ha annotato correttamente 214 parole sconosciute su 269, raggiungendo un'accuratezza del 79,55%.

Paragonate alle prestazioni ottenute dal Tagger addestrato su “Informative”, le prestazioni raggiunte dal Tagger addestrato su “Repubblica” mostrano le difficoltà di adattamento di quest'ultimo al dominio linguistico delle informative. Il divario che separa i primi due risultati si aggira sempre intorno ai 10 punti percentuali.

Sa andiamo a confrontare questi risultati con quelli descritti in TABELLA 4.3 possiamo osservare che le prestazioni del Tagger B sono sensibilmente migliorate.

L'incremento è stato del 2,98% per il Tagger addestrato su “Repubblica”, del 2,55% per il Tagger addestrato su “Informative” e del 2,98% per il Tagger addestrato su “Informative” + “Repubblica”. Questi valori sono significativi e mettono in evidenza il fatto che per il Tagger B la percentuale di errore nell'associare i tratti morfosintattici alle rispettive PoS può essere considerevole[36].

I valori sopra discussi sono riassunti in TABELLA 4.5.

TABELLA 4.5
Risultati ILC-UniPi Tagger solo PoS – Prestazioni solo Tagger B valutato sul “Test Corpus Informative”

| | **Tagger addestrato su Repubblica** | **Tagger addestrato su Informative** | **Tagger addestrato su Repubblica + Informative** |
|---|---|---|---|
| Tags corrette assegnate | 183/269 (68,03%) | 209/269 (77,69%) | 214/269 (79,55%) |
| Tags Sbagliate assegnate | 86/269 (31,97%) | 60/269 (22,31%) | 55/269 (20,45%) |

### 4.3.3 Considerazioni sui risultati ottenuti nel primo ciclo di valutazione

I risultati raggiunti in questo primo ciclo di valutazione mostrano un buon grado di adattamento di ILC-UniPi Tagger ad un dominio linguistico diverso da quello sul quale è stato addestrato.

36 Sebbene si parli di percentuali di errore comunque molto basse

Le performance ottenute dal Tagger addestrato sul dominio di Repubblica, sebbene inferiori, sono risultate molto vicine a quelle ottenute dal Tagger addestrato sul dominio delle Informative., come descritto in TABELLA 4.2 e TABELLA 4.4.
Per quanto riguarda la valutazione del solo Tagger B, quest'ultimo addestrato sul dominio di "Repubblica" non è risultato molto adattabile al dominio delle "Informative", non avendo superato una soglia di accuratezza del 65-68%, contro il 75-78% dell'accuratezza raggiunta dal Tagger B addestrato sul dominio delle "Informative".

## 4.4 Test effettuati per la valutazione di ILC-UniPi Tagger – Secondo ciclo

Nel primo ciclo di test, per addestrare il Tagger sul dominio di Repubblica era stato utilizzato il corpus "Repubblica" delle dimensioni di oltre 230.000 tokens, mentre il training corpus utilizzato per addestrare il Tagger sul dominio delle Informative , corpus "Informative", era di soli 23.953 tokens (dieci volte più piccolo).
Abbiamo pertanto deciso di effettuare un nuovo ciclo di test addestrando il Tagger su una porzione di Treebank[37] di "Repubblica" a disposizione presso l'ILC e paragonabile, come dimensioni, al corpus di training delle "Informative".
Da ora in avanti mi riferirò a questo piccolo corpus di training con il nome di "Repubblica_Ridotto".
Per la valutazione del Tagger addestrato sul dominio di Repubblica con il training corpus "Repubblica_Ridotto" sono stati utilizzati due test corpus:

- Il "Test Corpus Informative" già utilizzato nei precedenti test, delle dimensioni di 4280 tokens.
- Il "Test Corpus Repubblica", un piccolo corpus di 4283 tokens messo a disposizione dall'ILC e paragonabile per dimensioni al "Test Corpus Informative".

Anche il secondo ciclo di test si suddivide in due fasi:

- Test n°2 – A
- Test n°2 – B

37 Database sintattico dove ad ogni frase viene assegnata la struttura sintattica corretta.

I dati utilizzati e le loro caratteristiche sono riassunti in TABELLA 4.6

TABELLA 4.6
Dati utilizzati per il secondo ciclo di test di ILC-UniPi Tagger

| | **Nome** | **Dimensione** | **Dominio linguistico** |
|---|---|---|---|
| **Corpus di Training** | Repubblica_Ridotto | 24.010 tokens | Repubblica |
| **Corpus di Test** | Test Corpus Repubblica | 4283 tokens | Repubblica |
| | Test Corpus Informative | 4280 tokens | Informative |

## 4.5 Test n°2 – A

Nel calcolo delle statistiche le Tags recanti PoS corrette ma Tratti Morfosintattici sbagliati, sono state considerate errori.

### 4.5.1 Risultati ottenuti da ILC-UniPi Tagger, Tagger A + Tagger B

I risultati sono stati calcolati considerando le prestazioni generali di ILC-UniPi Tagger (Tagger A + Tagger B) addestrato su “Repubblica_Ridotto”:

Sul “Test Corpus Informative” il tagger ha raggiunto un'accuratezza dell' 87,26%, annotando correttamente 3735 unità ortografiche su 4280.

Sul “Test Corpus Repubblica” il Tagger ha raggiunto un'accuratezza del 94,49%, avendo annotato correttamente 4047 unità ortografiche su 4282.

L'accuratezza raggiunta dal Tagger valutato sul dominio delle “Informative” è stata inferiore di oltre 7 punti percentuali rispetto all'accuratezza raggiunta del Tagger addestrato e testato su corpora appartenenti al medesimo dominio linguistico (quello di “Repubblica”), come descritto dai dati riassunti in TABBELLA 4.7.

Il grado di adattamento del Tagger, addestrato su “Repubblica_Ridotto”, al dominio delle Informative è comunque buono.

Confrontando i risultati di TABELLA 4.7 con i risultati descritti in TABELLA 4.2

possiamo notare che il Tagger addestrato sul dominio di repubblica ha dimostrato un grado di adattamento maggiore al dominio delle Informative la dove il training corpus era più grande: un'accuratezza dell'87,26% raggiunta dal Tagger addestrato su "Repubblica_Ridotto" contro un'accuratezza dell'89,67% raggiunta del Tagger addestrato su "Repubblica".

L'incremento delle prestazioni a favore del Tagger addestrato sul training corpus più ampio è stato del 2,39%. Questo dato conferma quanto detto in precedenza, ovvero che più un training corpus è grande, più accurati risulteranno i pesi probabilistici stimati per ciascuna Tag e quindi migliore sarà il grado di generalizzare del Tagger stesso.

Possiamo quindi ipotizzare che se il corpus di addestramento di "Repubblica" avesse avuto dimensioni di qualche milione di token, i risultati ottenuti sul "Test Corpus Informative" con molta probabilità sarebbero migliorati ulteriormente, nonostante la diversità del dominio linguistico di test.[38]

TABELLA 4.7
Risultati ILC-UniPi Tagger – Prestazioni Tagger A + Tagger B addestrato su "Repubblica_Ridotto" e valutato sui corpora "Test Corpus Informative" e "Test Corpus Repubblica"

| | **Tagger testato sul "Test Corpus Informative" (4280 tokens)** | **Tagger testato sul "Test Corpus Repubblica" (4283 tokens)** |
|---|---|---|
| Tags corrette assegnate | 3735/4280 (87,26%) | 4047/4283 (94,49%) |
| Tags Sbagliate assegnate | 545/4280 (12,74%) | 236/4283 (5,51%) |

### 4.5.2 Risultati ottenuti da ILC-UniPi Tagger, solo Tagger B

I risultati ottenuti sono stati calcolati considerando le prestazioni del solo Tagger B, deputato a gestire le parole sconosciute, addestrato su "Repubblica_Ridotto":

Sul "Test Corpus Informative" erano presenti 269 parole sconosciute. Il tagger ha raggiunto un'accuratezza del 60,97%, annotandone correttamente 164.

Sul "Test Corpus Repubblica" erano presenti 142 parole sconosciute. Il tagger ha

38 Questa è solo una mia ipotesi poiché certe forme e costrutti sintattici tipici del dominio linguistico delle "Informative" potrei non trovarle neppure in un training corpus del dominio di "Repubblica" grande qualche miliardo di tokens.

raggiunto un'accuratezza del 79,58%, annotandone correttamente 113.

I risultati sopra detti sono riassunti in TABELLA 4.8.

L'accuratezza raggiunta dal Tagger B sul dominio delle “Informative” risulta inferiore di quasi 20 punti punti percentuali rispetto a quella raggiunta dal Tagger B addestrato e valutato su corpora appartenenti al medesimo dominio linguistico (quello di “Repubblica”). Il grado di adattamento al dominio delle “Informative” è stato piuttosto basso.

Anche in questo caso, confrontando questi risultati con quelli ottenuti dal Tagger B descritti in Tabella 4.3, emerge che le prestazioni migliori sono state ottenute la dove il Tagger è stato addestrato con un training corpus più ampio: un'accuratezza del 65,05% raggiunta dal Tagger addestrato su “Repubblica” contro un'accuratezza del 60,97% raggiunta del Tagger addestrato su “Repubblica_Ridotto”.

TABELLA 4.8
Risultati ILC-UniPi Tagger –solo Tagger B addestrato su “Repubblica_Ridotto” e valutato sui corpora “Test Corpus Informative” e “Test Corpus Repubblica”

| | **Tagger testato sul “Test Corpus Informative” (4280 tokens di cui 269 buchi)** | **Tagger testato sul “Test Corpus Repubblica” (4283 tokens di cui 142 buchi)** |
|---|---|---|
| Tags corrette assegnate | 164/269 (60,97%) | 113/142 (79,58%) |
| Tags Sbagliate assegnate | 105/269 (39,03%) | 29/142 (20,42%) |

## 4.6 Test n°2 – B

La prestazioni del Tagger sono state valutate prendendo in considerazione solo le Part of Speech delle Tag assegnate, ed rispettivi Tratti Morfosintattici, giusti o sbagliati che fossero, non hanno influenzato le statistiche dei risultati.

### 4.6.1 Risultati ottenuti da ILC-UniPi Tagger, Tagger A + Tagger B (Solo PoS)

I risultati ottenuti sono stati calcolati considerando le prestazioni generali di ILC-UniPi Tagger (Tagger A + Tagger B) addestrato su “Repubblica_Ridotto”:

Sul “Test Corpus Informative” il tagger ha raggiunto un'accuratezza del 87,83%,

annotando correttamente 3759 unità ortografiche su 4280.

Sul “Test Corpus Repubblica” il Tagger ha raggiunto un'accuratezza del 94,67%, avendo annotato correttamente 4055 unità ortografiche su 4282.

Anche in questo caso i risultati riassunti in TABELLA 4.9 mostrano che l'accuratezza del Tagger B testato sul dominio delle Informative è inferiore rispetto a quella raggiunta dal Tagger B sul dominio di Repubblica. Rispetto ai risultati ottenuti dal Tagger B descritti in TABELLA 4.7 possiamo vedere che su entrambi i corpus di test le prestazioni del Tagger sono leggermente migliorate, ottenendo un incremento delle prestazioni dello 0,57% per il test effettuato sul dominio delle Informative, e dello 0,18% per il test effettuato sul dominio di Repubblica. Questi valori molto bassi sono indice del fatto che una volta associata la PoS corretta, il grado di accuratezza del Tagger nell'assegnamento dei rispettivi tratti morfosintattici e molto elevato.

TABELLA 4.9
Risultati ILC-UniPi Tagger solo PoS – Prestazioni Tagger A + Tagger B addestrato su “Repubblica_Ridotto” e valutato sui corpora “Test Corpus Informative” e “Test Corpus Repubblica”

| | **Tagger testato sul “Test Corpus Informative” (4280 tokens)** | **Tagger testato sul “Test Corpus Repubblica” (4283 tokens)** |
|---|---|---|
| Tags corrette assegnate | 3759/4280 (87,83%) | 4055/4283 (94,67%) |
| Tags Sbagliate assegnate | 521/4280 (12,17%) | 228/4283 (5,33%) |

### 4.6.2 Risultati ottenuti da ILC-UniPi Tagger, solo Tagger B (solo PoS)

I risultati ottenuti sono stati calcolati considerando le prestazioni del solo Tagger B nell'assegnamento della sola PoS.

Sul “Test Corpus Informative” erano presenti 269 parole sconosciute. Il tagger ha raggiunto un'accuratezza del 63,57%, annotandone correttamente 171.

Sul “Test Corpus Repubblica” erano presenti 142 parole sconosciute. Il tagger ha raggiunto un'accuratezza del 83,10%, annotandone correttamente 118.

Questi risultati, riassunti in TABELLA 4.10, se confrontati con quelli descritti in

TABELLA 4.8 ci mostrano che le prestazioni del Tagger sono sensibilmente migliorate, avendo il Tagger B incrementato l'accuratezza di circa il 3%, su entrambi i domini linguistici. Tali valori evidenziano il fatto che per il Tagger B la percentuale di errore nell'assegnamento dei tratti morfosintattici alle rispettive PoS in alcuni casi può essere considerevole.

TABELLA 4.10
Risultati ILC-UniPi Tagger solo PoS – solo Tagger B addestrato su "Repubblica Ridotto" e valutato sui corpora "Test Corpus Informative" e "Test Corpus Repubblica"

| | **Tagger testato sul "Test Corpus Informative" (4280 tokens di cui 269 parole sconosciute)** | **Tagger testato sul "Test Corpus Repubblica" (4283 tokens di cui 142 parole sconosciute)** |
|---|---|---|
| Tags corrette assegnate | 171/269 (63,57%) | 118/142 (83,10%) |
| Tags Sbagliate assegnate | 98/269 (36,43%) | 24/142 (16,90%) |

### 4.6.3 Considerazioni sui risultati ottenuti nel secondo ciclo di valutazione

I risultati raggiunti nel secondo ciclo di valutazione mostrano che il grado di adattamento del Tagger ad un dominio linguistico diverso da quello del corpus di addestramento è strettamente dipendente dall'ampiezza del corpus utilizzato per la fase di training. Infatti, sebbene i risultati ottenuti dal Tagger addestrato utilizzando il training corpus di "Repubblica_Ridotto" e valutato sul dominio delle informative siano stati buoni, sono comunque risultati inferiori ai risultati ottenuti dal Tagger addestrato sul corpus di "Repubblica" (dieci volte più ampio di "Repubblica_Ridotto").

Per quanto riguarda la gestione delle parole sconosciute, le prestazioni raggiunte dal Tagger B addestrato su "Repubblica_Ridotto" e valutato sulle informative sono state piuttosto basse, a causa delle dimensioni del training corpus, evidentemente troppo piccolo. Infatti il Tagger B addestrato su "Repubblica_Ridotto" e valutato sul dominio delle informative ha avuto in tutti i test un decremento delle prestazioni di circa il 5% rispetto al Tagger B addestrato su "Repubblica".

## 4.7 Collocazione dei domini linguistici di Repubblica ed Informative nel panorama della lingua italiana contemporanea

Per classificare ed inquadrare meglio le diversità linguistiche ed il "tipo" di italiano che caratterizza il corpus di "Repubblica" da quello delle "Informative", possiamo adottare lo schema grafico proposto da Gaetano Berruto (Berruto, 1987)

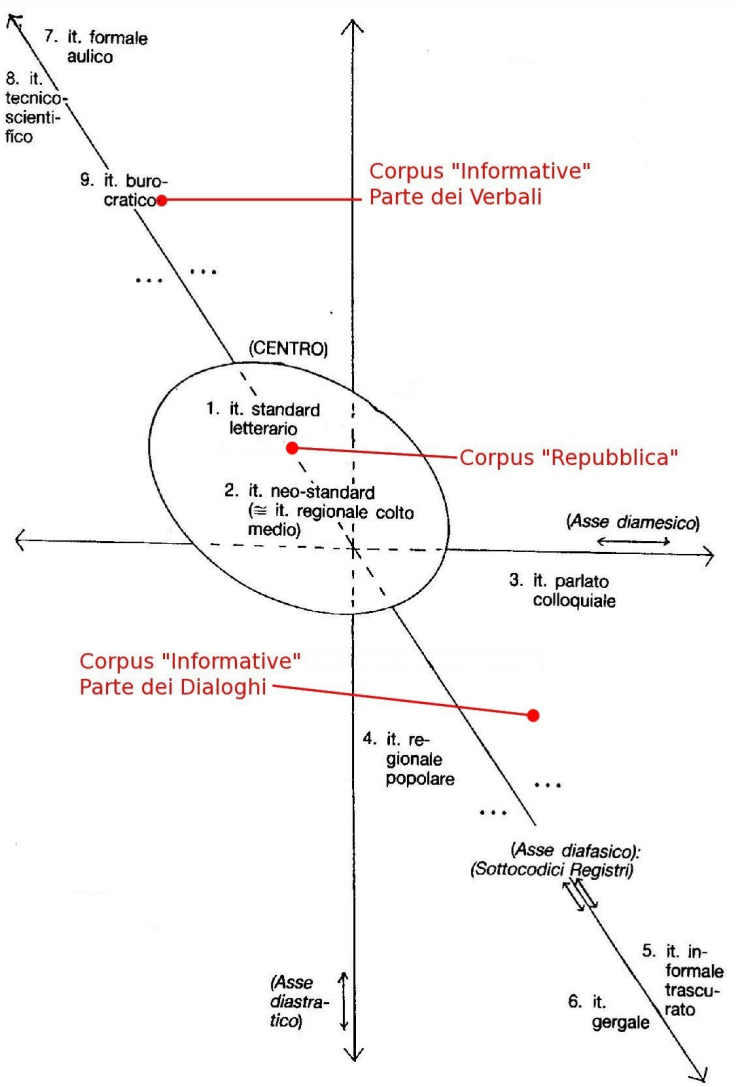


*FIGURA 4.1 – Architettura delle varietà dell'italiano contemporaneo*

Il grafico elaborato da Berruto nel 1987 rappresenta ad oggi una visualizzazione classica di quella che è stata definita "l'architettura dell'italiano contemporaneo".
Lo schema è composto da tre assi, dove ognuno dei quali rappresenta una dimensione di variazione dell'italiano, ovvero una "coordinata fondamentale" per classificare le varietà di italiano che costituiscono la nostra lingua[39].

L'asse orizzontale corrisponde alla "dimensione diamesica", sulla quale vengono identificati i mezzi attraverso i quali la lingua può essere veicolata. All'estrema sinistra dell'asse si colloca la modalità della lingua scritta, mentre alla polarità opposta la modalità della lingua parlata.

Sull'asse verticale è posta la "dimensione diastratica", la quale rappresenta la stratificazione sociale dell'individuo parlante/scrivente. In basso troviamo i ceti bassi, spesso capaci di padroneggiare il solo dialetto, mentre all'estremità opposta si collocano i ceti alti, generalmente capaci di padroneggiare un italiano colto e molto formale.

Sull'ultimo asse, quello obliquo, si colloca la "dimensione diafasica", la quale identifica la scelta di una precisa varietà della lingua in base al contesto ed alla circostanza: dalle situazioni comunicative più informali, in basso a destra, a quelle molto formali, in alto a sinistra.

Le tre dimensioni si incrociano e non sono indipendenti l'una dall'altra.

Attraverso questi parametri di classificazione, sul grafico vengono individuate nove varietà di italiano.

Sulla base dei tratti morfologici e sintattici che caratterizzano l'italiano dei corpora sui quali il Tagger è stato addestrato e valutato, possiamo classificare l'italiano del corpus di "Repubblica" come "Italiano standard", tendente, in alcuni casi, ad un "italiano dell'uso medio", o "neo-standard". La sua collocazione si pone, approssimativamente, appena sotto l'italiano standard.
Il corpus delle "Informative" non è invece collocabile in una sola zona del grafico, poiché internamente è composto da testi che presentano caratteristiche morfologiche e sintattiche molto diverse tra di loro.

39 Lo schema è da considerarsi molto generico, poiché rappresentare ogni varietà dell'italiano su un grafico piatto come questo è pressoché impossibile.

Per quanto riguarda la parte dei verbali, questi possono essere classificati con l'etichetta di “Italiano Burocratico”, la varietà n°8; la parte della trascrizione dei dialoghi invece può essere collocata tra la varietà n°3, “Italiano colloquiale”, e la varietà n°4, “Italiano regionale popolare”, nel quadrante in basso a destra del grafico.

Osservando sul grafico la collocazione delle varietà di italiano con cui il tagger addestrato su “Repubblica” e “Repubblica_Ridotto” è stato valutato, notiamo subito che sia la parte dei verbali, sia la parte dei dialoghi, si collocano ben distanti dalla varietà di italiano sulla quale sono stati calcolati i pesi probabilistici e pertanto il dominio delle “Informative”, linguisticamente parlando risulta molto diverso da quello di “Repubblica”.

# 5 Conclusioni

Osservando i risultati descritti in TABELLA 4.6, si può notare la differenza di prestazioni che il Tagger addestrato su “Repubblica_Ridotto” (24.010 tokens) raggiunge a seconda del test corpus utilizzato: un'accuratezza del 94,67% sul “Test Corpus Repubblica” contro un'accuratezza del 87,83% sul “Test Corpus Informative”.

Il divario dei circa 7 punti percentuali che separano i due risultati è dovuto, oltre alla dimensione del corpus di addestramento utilizzato, al fatto che i testi che compongono le “Informative” presentano caratteristiche linguistiche troppo diverse dai testi che compongono “Repubblica”.

Come abbiamo visto, valutando il Tagger su un corpus di test appartenente allo stesso dominio linguistico del corpus di training, le prestazioni risultano decisamente superiori a quelle raggiunte dal Tagger valutato su un dominio linguistico molto diverso da quello di training. Infatti sul test corpus appartenente al dominio delle “Informative”, i risultati migliori sono stati ottenuti dal Tagger addestrato sul dominio delle “Informative”, come descritto in TABELLA 4.2[40].

Per quanto riguarda i risultati raggiunti dal Tagger addestrato su “Informative + Repubblica” e valutato sul “Test corpus Repubblica”, l'incremento delle prestazioni rispetto al Tagger addestrato solo su “Informative” è stato minimo, e non ha raggiunto neppure un punto percentuale. Il corpus di “Repubblica” non è risultato di molto aiuto a causa della diversità che separa i due domini linguistici di “Repubblica” e delle “Informative”.

Il grado di adattamento di ILC-UniPi Tagger, addestrato sul dominio di Repubblica, al dominio delle informative è stato a mio avviso più che buono, considerando tutte le diversità linguistiche che separano la varietà dell'italiano di “Repubblica” da quello delle “Informative”.

Per quanto invece riguarda il riaddestramento del Tagger effettuato sul dominio delle Informative, sebbene abbia permesso di raggiungere prestazioni migliori sul “Test Corpus Informative” di quelle non ottenute utilizzando i pesi probabilistici calcolati sul training

40 Non considerando la fase di training fatta utilizzando la somma dei corpora “Informative” + “Repubblica”.

corpus di “Repubblica”, tenendo presente il fatto che il Tagger è stato addestrato sullo stesso dominio linguistico sul quale è poi stato valutato, i miglioramenti non sono stati così consistenti come mi aspettato.

Infatti, mentre il Tagger addestrato su “Repubblica_Ridotto”[41] e valutato sul “Test corpus Repubblica” ha raggiunto un'accuratezza del 94,49%, parallelamente il Tagger addestrato sulle “Informative”[42] e valutato sul “Test corpus Informative” ha raggiunto un'accuratezza del 90,56%. I dati sono riassunti in e messi a confronto in TABELLA 5.1.

Poiché in entrambi i casi il dominio linguistico del corpus di test utilizzato è stato lo stesso del corpus di addestramento, a parità di fattori mi sarei aspettato dei risultati molto simili tra loro in termini numerici, e non separati da una distanza di quasi 4 punti percentuali.

TABELLA 5.1
Risultati ottenuti da ILC-UniPi Tagger in due diverse situazioni

| | | | |
|---|---|---|---|
| **Dati utilizzati per il ciclo di training e di test** | **Corpus Training**: Repubblica Ridotto (24.000 tokens circa)<br><br>**Corpus Test**: Repubblica<br><br>**Dominio linguistico**: repubblica | | **Corpus Training**: Informative (24.000 tokens circa)<br><br>**Corpus Test**: Informative<br><br>**Dominio linguistico**: informative |
| **Accuratezza raggiunta da ILC-UniPi Tagger** | 94,49% | | 90,56% |

## 5.1 Qualità dei dati utilizzati per le fasi di training e di test di ILC-UniPi Tagger

Un elemento che ha sicuramente influenzato in negativo le prestazioni del Tagger addestrato sulle “Informative”, e che può in parte motivare i risultati ottenuti[43], è stata la qualità stessa dei dati di partenza sui quali sono stati costruiti il “Training Corpus Informative” ed il “Test Corpus Informative”.

I testi costituenti le Informative in molte parti contengono stringhe di caratteri non segmentabili a livello di tokenizzazione e non identificabili a livello morfo-sintattico, come

41 Dimensione di “Repubblica Ridotto” 24.010 tokens.
42 Dimensione delle “Informative” 23953 tokens.
43 Risultati scarsi rispetto alle mie aspettative, ma comunque molto buoni.

ad esempio errori di battitura, parole non separate da spazi, caratteri speciali inseriti nel testo senza un criterio ben preciso, più tutti quei fenomeni linguistici legati al dominio del linguaggio regionale/gergale (abbondante nella parte dei dialoghi).

Tutti questi elementi rappresentano soltanto "rumore", "elementi di disturbo" che vanno inevitabilmente a compromettere la qualità dei corpora sui quali il Tagger viene addestrato e valutato.

Prendendo come esempio il "Corpus Test Repubblica", della dimensione di 4283 tokens, ed il "Corpus Test Informative", della dimensione di 4280 tokens, i dati mostrano che le stringhe di caratteri che non hanno ricevuto nessuna interpretazione a livello di analisi morfologica sono 142 nel primo corpus (tutti nomi propri, per forza di cose non presenti a priori nel sottolessico di Magic) e 269 nel secondo (parte nomi propri, parte "rumore"). Quasi il doppio.

Come si può vedere dal grafico rappresentato in Figura 4.1, il linguaggio del dominio di "Repubblica" si concentra in un unico punto e ciò significa che da un punto di vista strutturale (sintassi, lessico) è sostanzialmente omogeneo, contrariamente al linguaggio componente le Informative.

Inoltre il linguaggio di Repubblica non presenta quelli "elementi di disturbo" che sono invece presenti in abbondanza nei testi delle informative, e che ne compromettono la qualità.

Detto e considerato ciò, tornando ai valori descritti in TABELLA 5.1, per fare un confronto alla pari tra il Tagger addestrato e valutato sul dominio delle "Informative" ed il Tagger addestrato e valutato sul dominio di "Repubblica", credo che il corpus di addestramento per il dominio delle Informative avrebbe dovuto avere dimensioni almeno doppie di quello utilizzato per il dominio di Repubblica.

Suppongo che in termini di prestazioni i  risultati sarebbero stati decisamente migliori di quelli ottenuti.

# 6 Bibliografia


[1] Abney S. (1996), *Part-of-Speech Tagging and Partial Parsing*, <http://www.vinartus.net/spa/95a.pdf>

[2] Baroni M. (2004), *Part-of-Speech Tagging* , <sslmit.unibo.it/~baroni/compling04f/pos.pdf >

[3] Battista M. & Pirrelli V. (1999) *Una piattaforma di morfologia computazionale per l'analisi e la generazione delle parole italiane,* ILC-CNR technical report.

[4] Berruto G., (1987), *Sociolinguistica dell'italiano contemporaneo*, Roma, La Nuova Italia Scientifica

[5] Berruto G., (1997), *Corso elementare di linguistica generale*, Torino, UTET Libreria|UTET Università.

[6] Cotroneo D., <www.mobilab.unina.it/Resources/Master%20MSTD-Mazzeo/Dispensa%20TACALN.pdf >

[7] DylanLab Team, (2007), *Progetto SmarText – Documenti di progettazione*, Istituto di Linguistica Computazionale CNR di Pisa, Pisa

[8] Jurafsky D. & Martin J.H. (2000), *Speech and Language Processing*, Prentice Hall

[9] Lenci A. & Montemagni S. & Pirrelli V., (2005), *Testo e Computer – elementi di linguistica computazionale*, Roma, Carrocci Editore.

[10] Lenci A. et al, (2007), "Maximum Entropy for Italian POS Tagging", *IA – Contributi scientifici*, N° 2, Giugno 2007, pp. 10 – 11

[11] McEnery T., Wilson A., (2001), "*Corpus Linguistics – An introduction*", second edition, Edinburgh University Press, Edinburgh.

[12] Tamburini F., (2000). *Annotazione grammaticale e lemmatizzazione di corpora in italiano.* <corpora.dslo.unibo.it/People/Tamburini/Pubs/AICLU-CILTA2000.pdf >

[13] Tamburini F., (2007), "Evalita 2007: The Part-of-Speech Tagging ask", *IA – Contributi scientifici*, N° 2, Giugno 2007, pp. 4 – 7