NER Multilingua

1. Introduzione

La Named Entity Recognition (NER) tagging multilingua è una sfida rilevante nel campo del Natural Language Processing (NLP). La NER è un componente cruciale del NLP poiché consente l'individuazione e l'etichettatura di entità come nomi di persone, luoghi e organizzazioni all'interno di un testo. Questa tecnica è fondamentale in molte applicazioni, tra cui l'estrazione di informazioni e l'analisi del sentiment.  
La NER è una tecnica fondamentale nel NLP in quanto consente l'estrazione di informazioni strutturate da testi non strutturati. Ad esempio, in un articolo di giornale, la NER può identificare i nomi delle persone coinvolte, le organizzazioni menzionate e i luoghi citati, facilitando la creazione di un database di informazioni utilizzabile per varie applicazioni, dall'analisi dei trend alla costruzione di grafi di conoscenza.

### Applicazioni pratiche del NER

Le applicazioni della NER sono diverse e ampie, includono:

* ***Estrazione di informazioni***: Automatizzazione della raccolta di dati da grandi quantità di testo.
* **Analisi del sentiment**: Identificazione di entità per analizzare il sentimento associato a esse.
* **Raccomandazioni personalizzate**: Utilizzo delle entità riconosciute per migliorare i sistemi di raccomandazione.
* **Assistenza virtuale:** Miglioramento della comprensione del contesto nei sistemi di assistenza virtuale.

Descrizione del progetto  
Questo progetto si propone di sviluppare un sistema di NER tagging multilingua utilizzando l'algoritmo di Viterbi. Il sistema sarà addestrato e valutato su tre lingue: inglese, italiano e spagnolo. A tal fine, saranno utilizzati tre dataset distinti, ciascuno contenente testi annotati con le entità di interesse, suddivisi in insiemi di addestramento (train), validazione (validation) e test. adesso verrà esaminato più nel dettaglio l'algoritmo di Viterbi e il suo ruolo nel contesto della NER tagging multilingua.

1. L’algoritmo di Viterbi

L'algoritmo di Viterbi è ampiamente utilizzato in diverse aree, tra cui il riconoscimento di sequenze e l'analisi dei dati sequenziali. Nel contesto del NER tagging, l'algoritmo di Viterbi gioca un ruolo fondamentale nel determinare la sequenza ottimale di etichette di entità per una data sequenza di parole.  
Il funzionamento dell'algoritmo di Viterbi può essere compreso meglio attraverso un esempio. Supponiamo di avere una frase di parole e dobbiamo determinare la sequenza più probabile di etichette di entità per queste parole. Immaginiamo di avere le seguenti parole: "John", "vive", "a", "New", "York". Ogni parola può essere etichettata come una persona (PERSON), una località (LOCATION), o un'altra entità (MISC).

L'algoritmo di Viterbi funziona in due fasi principali:

1. **Forward Pass**: In questa fase, vengono calcolate le probabilità di transizione e le probabilità di emissione. Ad esempio, potremmo calcolare la probabilità di passare da uno stato a un altro (ad esempio, da PERSON a LOCATION) e la probabilità di una parola data un'etichetta (ad esempio, la probabilità di "John" essere etichettata come PERSON). Queste probabilità vengono moltiplicate tra loro per ottenere la probabilità parziale per ogni possibile etichetta in ogni posizione.

2. **Backward Pass**: In questa fase, viene selezionata la sequenza più probabile di etichette di entità utilizzando una strategia di programmazione dinamica. L'algoritmo considera tutte le possibili sequenze di etichette e sceglie quella con la massima probabilità cumulativa, tenendo conto sia delle probabilità di transizione che delle probabilità di emissione calcolate nella fase precedente.

Nel nostro esempio, l'algoritmo di Viterbi considererebbe tutte le possibili combinazioni di etichette per le parole "John", "vive", "a", "New", "York" e selezionerebbe quella con la massima probabilità cumulativa. L'algoritmo di Viterbi è efficiente in termini di tempo di calcolo e spazio richiesto, il che lo rende una scelta preferita per applicazioni in cui è necessario determinare sequenze ottimali in modo rapido ed efficiente.

1. Valutazione delle Performance

Durante lo sviluppo del progetto, la performance del sistema di NER tagging è stata valutata attraverso diverse metriche, ciascuna con un ruolo specifico nell'analisi dei risultati.

Accuratezza Complessiva del Tagging  
L'accuratezza complessiva del tagging misura la percentuale di token etichettati correttamente rispetto al numero totale di token nel testo. Questa metrica fornisce una valutazione generale della precisione del sistema nel riconoscere le entità all'interno del testo.

Precisione  
La precisione rappresenta la percentuale di entità identificate correttamente rispetto al totale delle entità predette dal sistema. In altre parole, misura la proporzione di entità riconosciute correttamente rispetto a tutte le entità etichettate dal sistema come rilevanti. Una precisione elevata indica una bassa probabilità di falsi positivi, cioè di identificare erroneamente un'entità.

### Recall

Il recall, anche chiamato sensitivity o true positive rate, misura la percentuale di entità identificate correttamente rispetto al totale delle entità presenti nel testo. In sostanza, indica la capacità del sistema di riconoscere tutte le entità rilevanti presenti nel testo. Un recall elevato indica una bassa probabilità di falsi negativi, cioè di non identificare un'entità che è effettivamente presente nel testo.

Le metriche di precisione e recall forniscono una valutazione dettagliata delle prestazioni del sistema, consentendo di valutare sia la precisione nella predizione delle entità (precisione) che la capacità di identificare correttamente tutte le entità presenti (recall).   
Queste metriche sono state confrontate con quelle ottenute da altri approcci di tagging, come un tagger di tipo Naive e un tagger di tipo MEMM (Maximum Entropy Markov Model), per valutare l'efficacia del nostro sistema rispetto ai metodi di riferimento.

Il Progetto

Processo di Sviluppo del Sistema NER Tagging Multilingua

Il nostro progetto si è concentrato sulla costruzione di un sistema di NER tagging multilingua robusto, capace di gestire testi in inglese, italiano e spagnolo.   
L'uso di dataset distinti per ogni lingua e la suddivisione in train, validation e test set hanno permesso di creare un modello ben generalizzato e accurato. L'algoritmo di Viterbi è stato scelto, da consegna, per la sua efficacia nel determinare la sequenza ottimale di etichette di entità, essendo in grado di considerare sia le probabilità di transizione che quelle di emissione.

Concludendo, il progetto di NER tagging multilingua ha dimostrato l'importanza e la versatilità di questa tecnica nel NLP, e l'implementazione dell'algoritmo di Viterbi ha fornito una solida base per il riconoscimento accurato delle entità in più lingue. Le metriche di valutazione hanno confermato l'efficacia del sistema sviluppato, evidenziando il suo potenziale per applicazioni pratiche in vari domini.  
Lo sviluppo del sistema di NER tagging multilingua ha seguito una serie di fasi chiave, ognuna delle quali ha contribuito al successo complessivo del progetto. Le principali fasi del lavoro sono:

Acquisizione dei Dati  
La fase iniziale del progetto ha coinvolto l'acquisizione dei dati necessari per addestrare e valutare il sistema di NER tagging. Per ottenere dataset ricchi e diversificati, abbiamo fatto affidamento sui dataset messi a disposizione da Babelscape tramite il loro progetto WikiNeural. Questi dataset, specificamente preparati per l'addestramento di modelli di NER su Wikipedia, sono stati una risorsa preziosa per il nostro lavoro.

Abbiamo utilizzato i seguenti dataset di WikiNeural per le lingue inglese, italiana e spagnola:

* Inglese: [Link al dataset di WikiNeural in lingua inglese](https://github.com/Babelscape/wikineural/tree/master/data/wikineural/en))
* Italiano: [Link al dataset di WikiNeural in lingua italiana](https://github.com/Babelscape/wikineural/tree/master/data/wikineural/it)
* Spagnolo:[Link al dataset di WikiNeural in lingua spagnol](https://github.com/Babelscape/wikineural/tree/master/data/wikineural/es)a

Questi dataset sono stati già suddivisi in insiemi di addestramento, validazione e test, facilitando il processo di preparazione dei dati per il nostro progetto. Inoltre, essendo basati su Wikipedia, offrono una vasta gamma di testi in diverse categorie e stili linguistici, garantendo una rappresentazione ricca e diversificata delle entità linguistiche nelle rispettive lingue.

L'utilizzo di dataset provenienti da WikiNeural ha permesso di avviare il progetto con una solida base di dati annotati, accelerando il processo di sviluppo del sistema di NER tagging multilingua.

Preprocessing dei Dati  
Dopo aver ottenuto i dataset da WikiNeural, è stato necessario prepararli per l'addestramento del modello di NER tagging. Tuttavia, poiché i dataset forniti da WikiNeural erano già annotati e strutturati in modo appropriato, il preprocessing si è concentrato principalmente sulla tokenizzazione e sulla preparazione dei dati per l'addestramento.

Il preprocessing dei dati ha coinvolto principalmente le seguenti operazioni:

* **Tokenizzazione**: La divisione dei testi in parole o token, necessaria per analizzare il testo a livello di singola parola. La tokenizzazione è stata eseguita utilizzando le librerie di tokenizzazione disponibili in Python, assicurando che ciascuna parola fosse identificata correttamente.
* **Conteggio delle parole e delle entità**: Per calcolare le probabilità di emissione e di transizione necessarie per l'algoritmo di Viterbi, sono stati conteggiati il numero di occorrenze di ciascuna parola e il numero di occorrenze di ciascuna coppia di parole-etichetta nei dataset di addestramento.

Queste operazioni di preprocessing sono state cruciali per preparare i dati in modo appropriato per l'addestramento del modello di NER tagging. Dopo il preprocessing, i dati sono stati pronti per essere utilizzati per addestrare il modello e valutarne le performance.

Tecniche di Smoothing per Parole Sconosciute

Una delle sfide principali del NER tagging è come trattare le parole sconosciute, cioè le parole assenti nel dataset di addestramento. Per risolvere questo problema, abbiamo implementato diverse strategie di smoothing. Le ipotesi di smoothing utilizzate sono le seguenti:

* **Sempre O**Si presume che le parole sconosciute appartengano sempre alla classe "O" (Outside), cioè non siano entità rilevanti. La probabilità assegnata è:

***P(unk∣O)=1***

* **Sempre O o MISC**Si presume che le parole sconosciute possano appartenere alla classe "O" o alla classe "B-MISC" (Miscellaneo). Le probabilità sono distribuite uniformemente tra queste due classi:

***P(unk∣O)=P(unk∣B-MISC)=0.5***

* **Uniforme**Le parole sconosciute vengono distribuite uniformemente tra tutte le possibili etichette di entità (NER\_TAGs). La probabilità assegnata a ciascuna etichetta è:

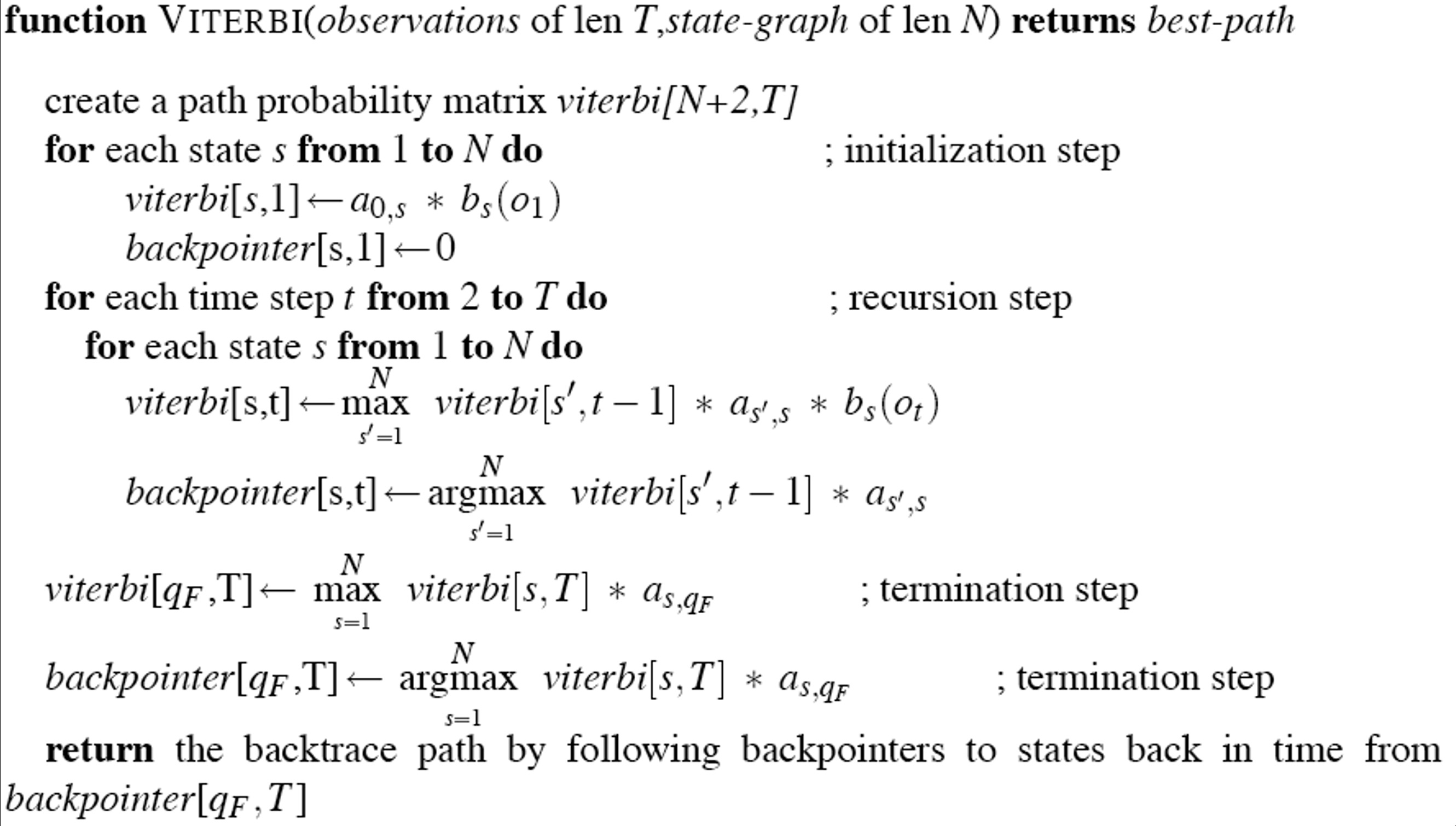
***P(unk|ti) = 1/#(NER\_TAGs)***

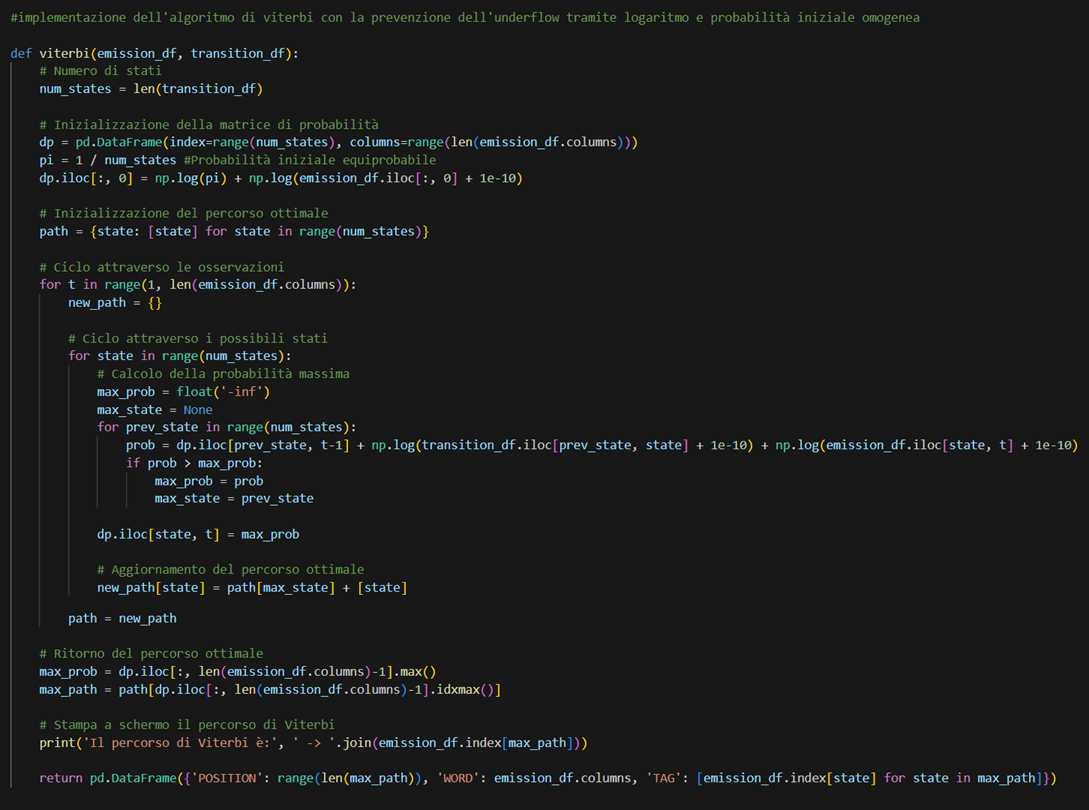
dove **#(NER\_TAGs)** è il numero totale di etichette di entità.

* **Statistica TAG sul Development Set**Questa strategia utilizza la distribuzione delle etichette nel development set per stimare le probabilità delle parole sconosciute. In particolare, si considerano le parole che compaiono una sola volta nel development set per stimare le probabilità di ciascuna etichetta. Questo approccio sfrutta la distribuzione empirica delle etichette nel development set per migliorare l'accuratezza delle previsioni per le parole sconosciute.

### Implementazione dell'Algoritmo di Viterbi

L'algoritmo di Viterbi è stato implementato in Python per determinare la sequenza ottimale di etichette di entità per una data sequenza di parole. La sua implementazione è stata realizzata traducendo direttamente lo pseudo codice fornito nelle slide del docente. Poiché durante il calcolo delle probabilità di transizione e di emissione si possono verificare valori molto piccoli, si è reso necessario utilizzare la scala logaritmica per evitare l'underflow, un problema comune quando si lavora con probabilità molto basse.



Sono state sviluppate due versioni dell'algoritmo. La prima esegue il tagging dell'intero dataset fornito senza suddividerlo in frasi, mentre la seconda sfrutta la suddivisione in frasi per calcolare i tag frase per frase. Quest'ultima versione è stata preferita considerando che le frasi non sono logicamente consequenziali all'interno del dataset, e l'approccio di suddivisione permette un maggiore controllo e una migliore gestione delle singole frasi.

### Struttura delle Baseline

Il modello Naive Bayes per il Named Entity Recognition (NER) tagging è uno dei modelli di base utilizzati per confrontare le performance del nostro sistema. Questo modello si basa sull'assunzione "naive" che le features (nel nostro caso, le parole) siano indipendenti tra loro, una volta che l'etichetta di entità è nota.

Nel modello Naive Bayes, per ogni parola all'interno di una frase, viene valutata la probabilità condizionata di assegnare un'etichetta di entità data l'osservazione della parola stessa. Se una parola è conosciuta (ovvero presente nel dataset di addestramento), il modello assegna all'etichetta di entità della parola la classe con la maggiore probabilità di emissione. D'altra parte, se una parola è sconosciuta (non presente nel dataset di addestramento), il modello adotta un'approccio più semplice, assegnando all'etichetta di entità della parola la classe "B-MISC" (Miscellaneo) con una probabilità uniforme. Questo significa che le parole sconosciute vengono trattate allo stesso modo e non vengono considerate per le probabilità di emissione calcolate dal modello.

D'altra parte, il modello MEMM (Maximum Entropy Markov Model) è un'altra baseline utilizzata per il confronto delle performance del nostro sistema. A differenza del modello Naive Bayes, il MEMM tiene conto delle dipendenze sequenziali tra le etichette di entità e modella le probabilità condizionate utilizzando una catena di Markov di ordine 1.

Nel MEMM, per ogni parola in una frase, il modello valuta la probabilità condizionata di assegnare un'etichetta di entità, tenendo conto delle etichette delle parole precedenti nella sequenza. Questo modello è in grado di catturare le dipendenze contestuali tra le etichette di entità e può quindi produrre etichette più accurate rispetto al Naive Bayes. Tuttavia, l'addestramento e l'uso di un MEMM possono essere computazionalmente più intensivi rispetto al Naive Bayes, poiché richiede la stima e l'ottimizzazione di un gran numero di parametri.

Risultati

Nel corso dell'analisi dei risultati, sono state calcolate diverse metriche per valutare le performance dei modelli di Viterbi e del Naive Bayes nel contesto del Named Entity Recognition (NER) tagging. Le metriche principali considerate sono state l'accuratezza complessiva, la precisione e il recall.

L'accuratezza complessiva misura la proporzione di etichette di entità correttamente predette rispetto al totale delle etichette nel dataset di test. La precisione indica la proporzione di etichette predette correttamente rispetto al totale delle etichette predette come appartenenti a una specifica classe di entità. Il recall, invece, rappresenta la proporzione di etichette di entità correttamente predette rispetto al totale delle etichette di entità presenti nel dataset di test.

I risultati di queste metriche sono stati salvati nel file "evaluation\_results.txt", insieme alla tipologia di smoothing applicata (se del caso) e alla dimensione del dataset di test utilizzato. Questi risultati forniscono una panoramica delle performance dei modelli nei diversi scenari considerati, consentendo un confronto diretto tra le varie configurazioni di smoothing e le dimensioni dei dataset di test.

Conclusioni

I risultati mostrano che il tagger Naive Bayes ha ottenuto fino a un massimo di 1 punto percentuale in più rispetto al Viterbi tradizionale, mentre il modello Viterbi diviso in frasi risulta essere significativamente meno accurato. Questi risultati sono stati supportati dall'analisi delle metriche di precisione e recall, che hanno evidenziato una maggiore precisione complessiva del Naive Bayes rispetto ai tagger Viterbi.

Tuttavia, è importante considerare il contesto del tagging di tipo BIO-NER, in cui gli unici tag disponibili sono PER, ORG, LOC e MISC. Questa limitata varietà di categorie potrebbe favorire il Naive Bayes, consentendogli di gestire efficacemente le probabilità di emissione per ciascuna parola e catturare le relazioni tra le parole e le categorie in modo efficiente.

L'introduzione di un numero maggiore di categorie potrebbe cambiare la dinamica. Con più tag, il Naive Bayes potrebbe risentire di una perdita di indipendenza tra le features e di una maggiore complessità nel modello, mentre il Viterbi potrebbe trarre vantaggio dalle sue capacità di catturare le dipendenze sequenziali tra le categorie.

Inoltre, l'analisi delle performance del Viterbi tradizionale e del Viterbi diviso in frasi indica che quest'ultimo potrebbe non essere ottimale per il nostro contesto specifico, poiché le frasi nel dataset non sono logicamente consequenziali.

In conclusione, sebbene il Naive Bayes abbia dimostrato una maggiore precisione complessiva in un contesto di tagging con un numero limitato di categorie, le implicazioni di introdurre un numero maggiore di tag devono essere prese in considerazione per comprendere appieno le dinamiche del modello. Il Viterbi, con la sua capacità di catturare dipendenze sequenziali tra le categorie, potrebbe adattarsi meglio a contesti più complessi con un numero maggiore di categorie.