

Riconoscimento automatico di nomi scientifici di
specie in testi per le indicizzazioni documentali
FAO

Tesi di laurea magistrale



Autore: John Bianchi (Matricola n° 517601)

Relatore: prof. Gianpaolo Coro

Correlatore: prof.ssa Maria Simi

Dipartimento di Filologia, letteratura e linguistica

Corso di laurea: Informatica Umanistica

Percorso formativo: Tecnologie del linguaggio

Università di Pisa

Anno accademico: 2020-2021

Indice

1	Introduzione	2
1.1	Contestualizzazione del lavoro e motivazione	2
1.2	ASFA: descrizione	4
1.3	Uso delle indicizzazioni documentali di ASFA	5
1.4	Problematiche generali del riconoscimento dei taxa	6
1.5	Idea della soluzione	9
2	Overview	11
2.1	Overview su NER	11
2.1.1	Sistemi NER statistici	14
2.1.1.1	NER ad apprendimento supervisionato	15
2.1.1.2	NER ad apprendimento semi-supervisionato	17
2.1.1.3	NER ad apprendimento non supervisionato	18
2.1.2	Rule-Based Named Entity Recognition	20
2.1.3	Sistemi Ibridi	22
2.2	Overview sulle taxa biologiche	24
2.2.1	Casi di sinonimia e omonimia	28
2.2.2	Specific epithet	29
2.2.3	Indicazione dell'anno di pubblicazione	30
2.3	Overview sui sistemi di taxa identification	31
2.4	Overview su NER per taxa	34
3	Metodo	39
3.1	Descrizione generale dell'approccio usato	39
3.1.1	Dataset utilizzati	43
3.1.2	Codifica e formati e di input e output	46
3.2	Descrizione particolareggiata dell'approccio: algoritmi e workflow	47
3.3	Identificazione dei Thesauri	57
3.4	Descrizione del sistema di cloud computing sul quale gira l'algoritmo	59
3.5	Descrizione del sistema NLP Hub	62
3.6	Open Science, ripetibilità, riproducibilità, riuso	69
4	Sistema di confronto utilizzato	70
4.1	Descrizione generale del sistema di confronto utilizzato	70
4.2	Data Collection	71
4.3	Preprocessing	74

4.4	IOB-Annotation	74
4.5	Creazione del Corpus Gold Standard	76
4.6	Valutazione e trattamento delle annotazioni semi-automatiche . .	76
4.7	Modello basato sull'architettura bi-LSTM-CRF	77
5	Risultati	79
5.1	Metriche di valutazione dei risultati	79
5.2	Approccio alla valutazione dei risultati	81
5.3	Valutazione dei risultati	83
5.3.1	Valutazione delle performance nell'estrazione di TAG . . .	83
5.3.2	Confronto con BotanicalNER	85
5.4	Tempi di esecuzione	89
6	Discussione	92
6.1	Sommario e valutazione qualitativa del sistema e dei risultati nei confronti delle soluzioni attuali.	92
6.2	Potenziati utilizzi	95
6.3	Estensioni future	97
6.3.1	Estensioni previste dal committente del progetto	97
6.3.2	Altre estensioni	99
7	Acronimi	100
	Bibliografia	114

Sommario

La divisione ASFA dell'Organizzazione delle Nazioni Unite per l'alimentazione e l'agricoltura (FAO) ha il compito operativo di gestire, indicizzare, e permettere l'accesso a documenti inerenti alle scienze acquatiche e alla pesca. L'insieme di dati ed informazioni raccolti da ASFA è composto da oltre 3000 elaborati testuali ed è considerato un riferimento autorevole del settore. Tuttavia, le attuali funzionalità di ricerca offerte dai motori di ricerca resi disponibili da ASFA risultano limitate al reperimento di entità indicizzate manualmente, il che influisce negativamente sull'efficacia di tale operazione. Di particolare importanza per ASFA è la necessità di collegare i documenti alle specie acquatiche studiate e menzionate in tali documenti. In questo contesto, un software di Named Entity Recognition (NER) per il riconoscimento dei nomi scientifici di specie nel testo potrebbe essere impiegato per identificare le specie menzionate in un documento ed etichettarlo/indicizzarlo per il suo reperimento tramite i motori di ricerca di ASFA. Tuttavia la realizzazione di un tale sistema dipende fortemente dalla gestione della complessità delle nomenclature tassonomiche. Queste, infatti, variano a seconda delle diverse aree della biologia e le regole generali sono affiancate da molteplici ambiguità ed eccezioni. Tale complessità rende difficile anche ai sistemi di apprendimento automatico di raggiungere prestazioni ottimali nell'identificazione di nomi scientifici di specie nel testo.

Nell'ambito del progetto europeo iMarine (2011-2014), un nucleo di esperti ha identificato un insieme ampio di regole ed eccezioni che consentissero di costruire un sistema a regole per identificare nomi scientifici di specie in un testo. Il presente studio, implementa proprio tale sistema, in maniera computazionalmente efficiente, e lo integra in un NER offerto come Web service ad ASFA per le indicizzazioni documentali automatiche. Il NER sviluppato è stato concepito per essere sufficientemente flessibile da riconoscere anche i termini notevoli usati nei documenti ASFA (thesauri). Esso si basa su una ampia base di conoscenza di nomi scientifici di specie (GBIF), disponibile grazie agli investimenti pregressi della Comunità Europea nei dati Findable, Accessible, Interoperable and Re-usable (FAIR) per il monitoraggio della biodiversità mondiale. Il NER realizzato è stato integrato nell'infrastruttura digitale *D4Science* del CNR e pubblicato mediante un'interfaccia Web (nlp.d4science.org/asfa/) e un sistema di cloud computing ad accesso standardizzato (*DataMiner*), che permette di parallelizzare le ricerche dei termini su più *core* e di gestire molteplici richieste utente contemporanee (mediante l'uso di una rete di macchine virtuali). È opportuno precisare che

l'applicativo in questione è collocabile anche come un sistema di Named Entity Recognition and Linking. Questo lo si evince dalla possibilità di associare le entità reperite ad una specifica entry collocata in una base di dati tassonomica. Tuttavia, ai fini divulgativi di questo elaborato, esso sarà descritto con il termine NER poiché si è voluto uniformare la nomenclatura a quella degli altri algoritmi propri del sistema nel quale è stato integrato.

I risultati sono stati confrontati, in termini di efficienza ed efficacia, con quelli di un modello di Deep Learning sviluppato dall'Istituto di Linguistica Computazionale dell'Università di Zurigo nel dominio della botanica. Questa valutazione ha messo in evidenza il fatto che, grazie alla combinazione di conoscenza esperta e dati FAIR, il nostro sistema riesce ad essere più efficiente e comparabilmente efficace. Inoltre, il NER sviluppato supporta 5 lingue, è facilmente aggiornabile e non richiede la costosa fase di preparazione di un corpus di addestramento dei sistemi supervisionati.

Il software realizzato è stato offerto ad ASFA sia come identificatore di generi ed epiteti di nomi scientifici di specie, che di riconoscitore di termini del thesaurus ASFA. Esso ha superato una valutazione qualitativa da parte del personale FAO ai fini operativi dei loro motori di indicizzazione documentale, soprattutto grazie alla flessibilità delle modalità di aggiornamento, di estensione della base di conoscenza tassonomica e del thesaurus, dell'efficienza computazionale e della estendibilità ad ulteriori lingue.

1 Introduzione

1.1 Contestualizzazione del lavoro e motivazione

Di pari passo al crescente processo di digitalizzazione degli elaborati testuali prodotti all'interno delle varie realtà istituzionali è seguito lo sviluppo di software sempre più efficienti adibiti alla gestione dei documenti. Tali sistemi sono in grado di rendere automatiche molte delle procedure proprie delle mansioni di coordinazione delle molteplici tipologie di contenuti ponendosi come uno strumento indispensabile per una quantità sempre maggiore di figure professionali. In questo contesto si collocano i software di Named Entity Recognition (NER). Essi, generalmente utilizzati nei task di Information Extraction, sono progettati per individuare specifiche tipologie di entità all'interno del contenuto dei documenti. Nel dominio delle scienze biologiche, trovano ampio utilizzo i NER designati al riconoscimento delle specie scientifiche. La realizzazione di questi software è un processo connesso alle criticità proprie della disciplina della tassonomia. Tali problematiche sono imputabili sia all'ambiguità dei costrutti che regolano la generazione dei nomi in questione che alla presenza di errori dovuti a scorretti processi di trasposizione da parte degli studiosi all'interno della letteratura dedicata [64].

Lo sviluppo di questi applicativi è stato originariamente dominato dall'impiego del paradigma rule-based il quale è stato gradualmente soppiantato dai modelli statistici i quali risultano essere dominanti anche negli scenari attuali. Questi ultimi, se da un lato hanno reso possibile il raggiungimento di prestazioni più elevate, presentano però una struttura in cui le regole utilizzate per il riconoscimento non sono rese esplicite e - nella maggior parte dei casi - comportano costi elevati per la creazione dei dati di addestramento.

È sulla base di queste considerazioni e dei relativi investimenti nella realizzazione di dati FAIR da parte della comunità scientifica ¹, ampiamente disponibili nel dominio in questione, che la divisione ASFA dell'Organizzazione delle Nazioni Unite per l'alimentazione e l'agricoltura (FAO) ha commissionato l'applicativo oggetto di questo elaborato. Le principali caratteristiche imposte dal committente per la

¹Insieme dei principi adibiti a garantire le seguenti proprietà sui dati, definiti dall'iniziativa *GO FAIR: "Findability, Accessibility, Interoperability, and Reuse of digital assets"* [55].

prima fase di sviluppo sono inerenti a:

- L'identificazione dei nomi scientifici di specie intesi come formati da generi (Genus) e o epiteti (species).
- L'impiego del dataset GBIF [8] da utilizzare come dizionario di riferimento per le entità identificabili.
- L'indipendenza dalla lingua in analisi.
- L'utilizzo di un approccio applicabile anche per l'identificazione di vocaboli contenuti nei loro thesauri.

Tali requisiti, in concomitanza all'assenza di dati da adoperare per le fasi di training di un modello statistico, hanno indirizzato il progetto in direzione del paradigma rule-based. Questo approccio, data l'ottima qualità dei dati FAIR impiegati e a regole di riconoscimento stabilite sulla base della conoscenza esperta tratta dall'articolo *Taxa Merging Discussion* del progetto europeo *iMarine* [64], ha permesso la realizzazione di un sistema avente prestazioni in linea con gli obiettivi prefissati. Dal confronto con *BotanicalNER* [87], un software costruito sul modello statistico LSTM-CRF che comprende anche la funzionalità di NER per nomi scientifici di specie, il programma sviluppato nel nostro studio ha raggiunto prestazioni comparabili nell'identificazione delle entità di genere e specie.

1.2 ASFA: descrizione

"The Food and Agriculture Organization (FAO) is a specialized agency of the United Nations that leads international efforts to defeat hunger. Our goal is to achieve food security for all and make sure that people have regular access to enough high-quality food to lead active, healthy lives. With over 194 member states, FAO works in over 130 countries worldwide. We believe that everyone can play a part in ending hunger." [3]

L'Organizzazione delle Nazioni Unite per l'alimentazione e l'agricoltura è un'istituzione che si occupa della gestione delle risorse alimentari all'interno dei territori compresi nelle nazioni facenti parte dell'ONU. Per l'adempimento di tale obiettivo, tra le svariate iniziative, trovano maggior rilievo le pubblicazioni dei rapporti contenenti statistiche e proiezioni inerenti alle risorse ed attività collaterali del settore dell'alimentazione e alle capacità nutrizionali di ogni singolo paese. [3, 122, 66]

All'interno di questo contesto trova collocazione la divisione *Aquatic Sciences and Fisheries Abstracts* (ASFA). Essa, in conformità al ruolo di *"International Cooperative Information System"* è adibita all'amministrazione di elaborati testuali utili ai fini operativi della FAO. Concretamente il servizio che offre è relativo all'indicizzazione e all'*abstracting* di documenti di carattere scientifico con l'aggiunta della gestione dei relativi aspetti socio economici e legali degli stessi [4].

L'insieme dei testi gestiti è conservato all'interno del *database ASFA* e costituisce uno dei riferimenti nel campo delle scienze acquatiche e della pesca. Pubblicato inizialmente nel 1971, questo insieme di dati è composto da pubblicazioni, libri, rapporti, traduzioni, atti di convegni e letteratura grigia e viene costantemente arricchito da una rete internazionale di centri di informazione. Formato da oltre 2 500 000 documenti è distribuito digitalmente in tutto il mondo mediante un contratto tra il publisher *ProQuest* [101], la FAO e le Nazioni Unite [2].

1.3 Uso delle indicizzazioni documentali di ASFA

La divisione ASFA rende disponibile il *database ASFA* mediante due differenti motori di ricerca. Il primo si colloca internamente al dominio della FAO [48] e utilizza funzionalità di indicizzazione sviluppate direttamente da quest'ultima. Tale motore di ricerca permette all'utente di ricercare distintamente all'interno delle seguenti categorie di documenti:

- Publications
- Fact Sheets
- Meetings & News
- Legislation

Nella prima di esse (*Publications*) è possibile reperire documenti mediante la definizione di query distinguendo per le categorie di titolo dell'opera o in alternativa dei relativi autori, specificando opzionalmente l'anno di pubblicazione, la lingua e titolo della serie. Le funzionalità offerte da tale operazione non permettono all'utente di reperire documenti tramite l'indicazione di keyword contenute all'interno del testo che compone gli stessi poiché la ricerca, basata sulle informazioni metatestuali associate a ciascun documento, non incorpora questa tipologia di dati.

Un'altra possibilità di ricerca è resa disponibile mediante *AquaDocs* [17]. Esso si configura come un repository congiunto dell'*UNESCO/IOC International Oceanographic Data and Information Exchange* e dell'*International Association of Aquatic and Marine Science Libraries and Information Centers* ed è comprensivo anche del *database ASFA*. Le funzionalità offerte sono analoghe a quelle descritte per ASFA con l'aggiunta della possibilità di distinzione per continente di provenienza [17] ma come il precedente non permette di reperire elementi in base al contenuto.

Questa limitazione, condivisa da entrambi i sistemi, risulta colmabile mediante l'applicazione di un software di NER in grado di identificare ed associare tag inerenti al contenuto dei relativi documenti. Questi, realizzati sulla base delle entità nominali presenti all'interno degli abstract, debbono poi essere inclusi nei metadati di indicizzazione utilizzati dai relativi motori di ricerca.

1.4 Problematiche generali del riconoscimento dei taxa

La definizione di un algoritmo che, mediante regole definite, riesca ad identificare in maniera univoca le entità delle specie scientifiche all'interno del testo è un obiettivo che non è ancora stato raggiunto. Questo è dato dal fatto che le regole vigenti all'interno dei vari protocolli della nomenclatura per la gestione dei nomi scientifici risultano essere frammentarie. Esse variano indipendentemente nelle singole branche della biologia e sono spesso interpretate in maniera scorretta dagli stessi tassonomisti. Inoltre, il vincolo di unicità per i singoli taxon, è garantito solo all'interno di ciascuna di tali branche. La dinamicità dei singoli nomi, intesa come il possibile cambiamento nel corso del tempo in base al dibattito accademico in merito, costituisce un'ulteriore complicazione [64].

Alcuni esperti del settore della tassonomia hanno collaborato nella stesura di un articolo (*Taxa Merging Discussion* [64]) nel contesto del progetto europeo iMarine [7], che illustra vari scenari di ambiguità in tal senso, corredando la trattazione con relativi esempi. L'incorporazione di tale conoscenza all'interno di un sistema mediante la definizione di regole costruite ricalcando ciascuna delle singole casistiche risulta essere un processo legato alle criticità degli scenari descritti nel testo in questione.

Un possibile esempio è illustrato dalla seguente esemplificazione in cui Lamarck è l'autore che ha firmato la pubblicazione mentre Leach è la persona che l'ha effettivamente scritta. Stando alle normative standard, solamente Lamarck dovrebbe essere riportato ma per riconoscere l'impegno intellettuale può essere incluso anche Leach.

Family Semelidae
Genus *Abra* Leach in Lamarck, 1818
Species *Abra alba* (W. Wood, 1802)

esempio tratto da: *Taxa Merging Discussion, iMarine Project Wiki* [64]

La definizione formale di un algoritmo in grado di interpretare correttamente scenari di questo tipo risulta essere un processo arduo a causa della componente aleatoria rappresentata da diverse eventualità, ad esempio:

- La descrizione di ogni specie scientifica può essere pubblicata e descritta da due persone diverse.

- Il nome di tale persona può essere arbitrariamente riportato all'interno della letteratura in base a nessun fattore specifico.

Un altro caso si colloca nei documenti della botanica in cui, eccezionalmente rispetto alle altre scienze biologiche, i sottogeneri sono riportati in maiuscolo all'interno delle parentesi, come mostrato nell'esempio seguente:

1 - *Uca (Paraleptuca)* Bott, 1973
 2 - *Uca (Paraleptuca) lactea* (De Haan, 1835)
 3 - *Uca (Paraleptuca) lactea annulipes* (H. Milne-Edwards, 1837)

esempio tratto da: *Taxa Merging Discussion, iMarine Project Wiki* [64]

Queste casistiche possono essere interpretate erroneamente da parte di un algoritmo il quale può confondere la dicitura in questione con l'autore della pubblicazione andando ad identificare scorrettamente le entry 2 e 3 della medesima esemplificazione.

Un'eventualità simile la si riscontra nell'esempio seguente in cui il nome del firmatario della pubblicazione può essere riportato o meno all'interno delle parentesi a seconda della disciplina di riferimento a cui appartiene l'articolo:

Mactra alba W. Wood, 1802
Abra alba (W. Wood, 1802)

esempio tratto da: *Taxa Merging Discussion, iMarine Project Wiki* [64]

Se ogni taxon fosse seguito dall'indicazione mediante parentesi dell'anno ed autore della pubblicazione allora il compito fin qui descritto risulterebbe di facile definizione poiché sarebbe sufficiente indicare come confine di termine per i Genus e Species il carattere dell'apertura della parentesi tonde "(" . Tuttavia ciò risulta essere parzialmente dedicato alle singole discipline e anche all'interno di esse non viene sempre rispettato. Quindi, le parentesi non rappresentano un elemento sul quale poter costruire regole in tal senso.

Nonostante che tali considerazioni non riguardino direttamente i ranghi di Genus e Species, i quali occorrono all'interno della letteratura dedicata unicamente nelle due forme illustrate dall'esempio seguente,

Genus
Genus species
G. species

nella definizione di un algoritmo in grado di riconoscere questi ultimi nel testo è necessario conoscere queste ed altre casistiche in modo da poter definire con precisione i confini di identificazione delle entità in questione soprattutto in virtù della corretta gestione dei casi in cui i Genus non risultino essere seguiti dall'epiteto della specie.

1.5 Idea della soluzione

Le informazioni contenute all'interno dell'articolo *Taxa Merging Discussion* [64] sono state redatte da esperti nell'ambito della tassonomia che hanno riportato puntualmente tutte le possibili incongruenze verificabili all'interno della letteratura dedicata. Queste informazioni, liberamente consultabili online all'interno della iMarine Project Wiki [7], descrivono tutte le casistiche verificabili in tal senso e sono illustrate con relativi esempi. Tale conoscenza è stata analizzata con attenzione a monte della realizzazione del lavoro svolto ed è stato constatato che ogni esemplificazione fosse di fatto definibile formalmente. Questa consapevolezza ha indirizzato lo sviluppo del programma realizzato nel nostro studio in direzione del paradigma rule-based.

Se i NER costruiti con modelli ad apprendimento supervisionato impongono l'utilizzo di grandi insiemi di dati necessari per la fase di addestramento dello stesso, quelli basati su regole si servono di gazetteers composti unicamente dall'insieme delle entità identificabili dal sistema in questione. A differenza dei corpora di addestramento che comportano costi elevati sia per la loro realizzazione che per le eventuali operazioni di aggiornamento, le liste contenenti i nomi scientifici delle specie risultano essere reperibili gratuitamente all'interno dei domini garantiti dalle varie istituzioni del settore. Negli ultimi anni la comunità scientifica di riferimento ha direttamente finanziato la realizzazione di knowledge base composti da una quantità di elementi sufficienti a rappresentare la totalità delle specie scientifiche esistenti. GBIF [8], ovvero il dataset utilizzato nel nostro studio, si propone come una delle operazioni più riuscite in tal senso essendo il risultato della combinazione di oltre 100 knowledge based. La versione attuale, ossia la medesima utilizzata per l'applicativo sviluppato nel nostro studio, è costituita da oltre 6 milioni di entità per le quali sono disponibili molteplici tipologie di informazioni. Esso, liberamente scaricabile in rete [8], è disponibile nelle modalità standardizzate riguardanti accesso e forma dei dati.

La soluzione proposta nel nostro lavoro è estendibile anche al riconoscimento delle entità dei thesauri. Tale funzionalità, presente nei requisiti iniziali del progetto, è stata raggiunta sfruttando sia la natura delle entità in questione che le potenzialità del sistema a regole il quale è stato modellato anche in funzione di questa specifica necessità espressa da parte del committente di tale progetto. Essa, collocabile come una versione *pre-alpha*, è stata costruita utilizzando il dataset "*ASFA*" [46].

Lo sviluppo del software del nostro studio è stato quindi interamente svolto mediante l'impiego di dati FAIR i quali, in concomitanza all'utilizzo del paradigma rule-based mediante regole definite sulla base della conoscenza di esperti, hanno permesso al nostro sistema di essere impiegato per le indicizzazioni documentali della FAO per i documenti del *database ASFA*.

Inoltre, le fasi di progettazione sono state concepite per rendere l'applicativo conforme all'integrazione con la piattaforma di cloud computing *NLPHub* [34]. Tramite un servizio web dedicato, le funzionalità proprie del nostro sistema vengono integrate con quelle di altri NER mediante l'applicazione di uno specifico algoritmo orchestrator in grado di combinare sapientemente i rispettivi output. Queste ed altre funzionalità sono disponibili all'interno dell'infrastruttura *D4Science* [18] la quale, oltre che a fornire una specifica interfaccia, dispone di risorse computazionali sufficienti per poter eseguire questa tipologia di processi all'interno di un workspace dedicato a ciascun utente garantendo un elevato livello di efficienza nell'esecuzione delle varie operazioni. In aggiunta esse sono marcate mediante l'assegnazione di un identificativo distinto che garantisce riproducibilità e condivisione di ogni azione svolta. I singoli processi, così come l'architettura stessa del sistema, risultano essere in linea con il paradigma dell'Open Science.

2 Overview

2.1 Overview su NER

Il task di individuazione e categorizzazione di nomi all'interno di un testo prende il nome di Named Entity Recognition (NER) [77, 85, 90, 133, 34]. Esso è ampiamente utilizzato nella disciplina del Natural Language Processing in qualità di componente ausiliario di molti task strutturati come Information Extraction, Question Answering e Machine Translation. In quest'ultima, ad esempio, i nomi propri di persona vengono identificati ed esclusi dalla traduzione mediante l'applicazione di un software NER [133, 13].

L'espressione "*Named entity*" è stata utilizzata per la prima volta nel corso della sesta edizione della *Message Understanding Conference* [57] dove venne sottolineata l'importanza dell'estrazione d'informazione da testi non strutturati al fine di distillare unità informative specifiche come nomi (intesi come propri e non), espressioni numeriche e altre tipologie di dati.

"The first goal was to identify, from the component technologies being developed for information extraction, functions which would be of practical use, would be largely domain independent, and could in the near term be performed automatically with high accuracy. To meet this goal the committee developed the "named entity" task, which basically involves identifying the names of all the people, organizations, and geographic locations in a text. The final task specification, which also involved time, currency, and percentage expressions, used SGML markup to identify the names in a text." [57]

Questo task, riconosciuto come una sottobranchia della già sviluppata Information Extraction, prese il nome di "Named Entity Recognition and Classification (NERC)" [90].

Ogni tipologia di Named Entity Recognition realizza l'operazione di riconoscimento mediante due passaggi distinti: il primo consiste nell'identificazione delle parole target nel testo, mentre il secondo, assegna queste ultime a delle categorie

decise a priori e univocamente distinte quali ad esempio: nomi di persona, organizzazioni, luoghi, date ed espressioni temporali [34]. L'importanza dell'univocità delle categorie di assegnazione è una caratteristica fondamentale di ogni software di Named Entity Recognition e fonda le proprie radici concettuali nel pensiero del filosofo del linguaggio Saul Kripke espresse nel libro *Naming and necessity* [70].

Se per un essere umano distinguere e categorizzare gli elementi presenti in un testo è un compito relativamente semplice, al contrario risulta essere un task molto complesso per una macchina. La maggiore difficoltà che si riscontra durante la realizzazione di un Named Entity Recognition è quella di gestire le ambiguità presenti nel linguaggio naturale che devono essere interpretate correttamente sia nella parte d'individuazione che in quella di categorizzazione. È necessario identificare i casi di omonimia fornendo al NER istruzioni per la disambiguazione dei possibili contesti linguistici [90].

I software di questa tipologia possono essere classificati sulla base di funzionamento e realizzazione. Per quanto concerne la prima delle due distinzioni, i NER sono suddivisi in *flat NER* e *nested NER* [78]. Dei primi fanno parte tutte quelle tipologie adibite all'assegnazione di etichette a ciascuna delle parole contenute in un testo. In questi processi, conosciuti anche con il nome di "*sequence labeling task*", non si possono verificare sovrapposizioni nelle categorie assegnate [78]. Un esempio di applicazione si può riscontrare in *Named Entity Recognition with Bidirectional LSTM-CNNs* [31] in cui è stata realizzata un'architettura ibrida e bidirezionale per ovviare al problema della mancanza di dati di addestramento. Nel lavoro di Ma e Hovy (2016) [83] è stato costruito un task di sequence labeling attraverso la combinazione di LSTM, CNN e CRR bidirezionali in sostituzione di specifiche regole per l'identificazione e di dati di addestramento preprocessati.

Al contrario, i Named Entity Recognition ad approccio *nested*, sono in grado di assegnare un numero variabile di categorie ad una o più parole all'interno di un testo come illustrato in *figura 1*. Tale tipologia di NER ha origine nello studio condotto da Kim et al. (2013) [67] in cui questo task è stato realizzato tramite un approccio basato unicamente su regole. Da allora sono state applicate molteplici tecniche, ad esempio nel lavoro di Alex et al. (2007) [14] mediante modelli di tipologia Conditional random fields a più livelli.

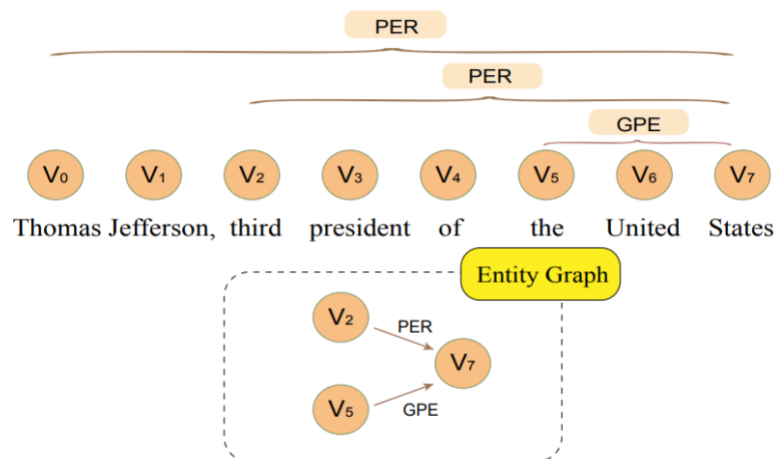


Fig. 1: Un esempio del funzionamento di nested NER. Gli indici iniziali e finali delle entità interne annidate sono collegati dalle linee continue (Ying Luo e Hai Zhao 2020) [82]

Per quanto concerne la componente realizzativa, è possibile suddividere i software di Named Entity Recognition in tre macrocategorie [133]:

1. Sistemi NER statistici
2. Sistemi ibridi
3. Rule-Based NER

Questi sono descritti, con relativi esempi di realizzazione, nelle sezioni successive.

2.1.1 Sistemi NER statistici

I software Named Entity Recognition statistici rappresentano la variante odierna più utilizzata. Con essi è stato possibile raggiungere prestazioni elevate in termini di accuratezza nei domini applicativi più generici. Essi generalmente necessitano dei seguenti componenti [133]:

1. Corpora di dati annotati per l'addestramento e la valutazione del modello statistico
2. Modello statistico per la rappresentazione numerica del corpus di addestramento

Questi modelli sono composti da una serie di valori numerici che indicano la probabilità del manifestarsi di un evento in un determinato contesto linguistico.

Le tre principali tecniche di addestramento, relative non solo all'applicazione nel campo della Named Entity Recognition, sono:

1. NER ad apprendimento supervisionato
2. NER ad apprendimento semi-supervisionato
3. NER ad apprendimento non supervisionato

2.1.1.1 NER ad apprendimento supervisionato

Rappresenta la variante più utilizzata per l'approccio alla creazione di NER statistici. Questa tecnica può essere realizzata mediante diversi algoritmi di apprendimento come ad esempio alberi decisionali (Decision Trees), Support Vector Machines, Maximum Entropy models e altri. Tutti questi, allo stesso modo, vengono addestrati su corpora di dati annotati manualmente in cui sia i confini delle entità che le loro etichette vengono rese esplicite. Ciò rende possibile la comprensione, dal punto di vista dell'apprendimento statistico, delle varie casistiche possibili in base all'individuazione di *Feature* ovvero di caratteristiche significative all'interno del testo di addestramento [90]. Una volta completata questa fase, il software NER viene testato nel dataset di valutazione. Una tecnica utilizzata per stimare con precisione le performance di questa tipologia di algoritmi è quella del conteggio del *vocabulary transfer*. Questo valore è composto dalle entità distinte che concorrono nell'insieme del corpus di addestramento e di valutazione. Il *vocabulary transfer* risulta essere un ottimo indice per stimare l'ammontare della recall (ammontare di entities identificate in rapporto alla totalità delle presenti nel testo in input) dell'algoritmo, ma in altri casi si dimostra un valore eccessivamente permissivo poiché non tiene conto della quantità di assegnazione corretta di entità ripetute nel testo in input [90].

Tra le prime applicazioni di algoritmi basati su supervised learning per task di NER un caso significativo si colloca nel lavoro di Bikel et al. (1998) [22] in cui, mediante un *Hidden Markov Model*, è stato realizzato un NER di tipologia flat adibito al riconoscimento di otto diverse categorie di entità nominali che ha raggiunto il più alto valore in termini di accuratezza mai segnato prima di allora (90%). Per quanto concerne gli utilizzi più recenti, nel lavoro di Wibawa et al. (2016) [121], è stato costruito un software in grado di assegnare, a testo proveniente da articoli di giornale indonesiani, 15 differenti classi, utilizzando un algoritmo di tipologia *Simple logistic regression* mediante l'utilizzo di features inerenti a: lessico delle singole parole, struttura della frase, contesto linguistico. Mediante questa combinazione è stato raggiunto il valore *F1-score* di 0.528. Nello studio di Guo et al. (2009) si è optato per un approccio basato su *Latent Dirichlet Allocation* per lo sviluppo di un Named Entity Recognition *in Query* adibito, quindi, all'identificazione e categorizzazione di testo caratterizzato da dimensioni ridotte. Questa tipologia di software, molto utilizzata dai motori di ricerca, è stata realizzata in mediante un'architettura basata su *WS-LDA* (*Weakly Supervised Latent Dirichlet Allocation*) che utilizza entità etichettate solo parzialmente [58]. Chieu

et al. (2002) [30] realizzarono per primi un NER statistico basato sull'entropia massima che utilizza esclusivamente informazioni contenute all'interno del documento in analisi ed un solo classificatore. Vennero raggiunte performance paragonabili a quelle degli approcci precedenti in cui, al contrario, veniva impiegato un classificatore secondario adibito alla correzione degli errori commessi da quello principale.

Questa tipologia di algoritmi richiede tipicamente una grande quantità di dati di addestramento i quali, molto spesso, rappresentano la causa di diverse problematiche. Come evidenziato dall'indagine condotta da *Dimensional Research* nel 2019 [40] il 96% delle aziende che utilizzano algoritmi che necessitano di dati di addestramento supervisionati riscontrano complicazioni imputabili a molteplici fattori. Lo scenario più frequente (66%) è quello della presenza di errori, commessi dal personale adibito all'annotazione manuale, nei dati di addestramento seguito, al 51%, dalla ridotta disponibilità di questi ultimi [40]. Inoltre, come sollevato da Andre Ye (2020) [128], gli algoritmi di supervised learning sono capaci unicamente di interpretare scenari che sono già presenti nei dati di training in quanto essi non dispongono delle istruzioni necessarie per eseguire operazioni di estrapolazione per ricavare ulteriore conoscenza da essi. Da questo segue che il dominio applicativo risulti essere strettamente vincolato agli stessi, rendendo ostiche operazioni di aggiornamento e debugging. Inoltre, questi algoritmi, necessitano di tempi di addestramento significativamente lunghi i quali risultano essere costosi in termini di risorse computazionali [39].

2.1.1.2 NER ad apprendimento semi-supervisionato

"It's a special form of classification. Traditional classifiers need labeled data (feature/label pairs) to train. Labeled instances however are often difficult, expensive, or time consuming to obtain, as they require the efforts of experienced human annotators. Meanwhile unlabeled data may be relatively easy to collect, but there has been few ways to use them. Semi-supervised learning addresses this problem by using large amount of unlabeled data, together with the labeled data, to build better classifiers. Because semi-supervised learning requires less human effort and gives higher accuracy, it is of great interest both in theory and in practice." [132]

Il paradigma di apprendimento semi-supervisionato è inerente alla creazione di algoritmi di apprendimento automatico in grado di imparare da dati di addestramento annotati solo parzialmente. Rappresenta una tipologia di algoritmi molto utilizzata a causa delle problematiche inerenti ai dati di addestramento. L'obiettivo di questi software, è quello di riuscire ad interpretare correttamente il rapporto che vi è tra dati annotati e non al fine di trarre vantaggio da questa dicotomia estraendo la conoscenza necessaria per eseguire al meglio la fase di addestramento [131].

I sistemi NER ad approccio semi-supervisionato vengono realizzati tramite svariate tecniche di supervisionamento parziale o automatizzato dei dati di training. Una delle metodologie più utilizzate è quella denominata *"bootstrapping"* in cui solo alcune parti di essi vengono revisionati manualmente. Il sistema infatti richiede all'utente di correggere solo una porzione dei termini etichettati nei dati di training ed è proprio a partire da questi che ne identifica automaticamente altri basandosi su contesti ritenuti analoghi. Questo processo viene poi iterato più volte al fine di ottimizzare al meglio i dati [90]. La tecnica del bootstrapping viene eseguita mediante diversi algoritmi, tra i quali, anche utilizzando le espressioni regolari (*regex*). Queste sono in grado di identificare casistiche linguistiche caratterizzanti sulla base della successione arbitraria e definita di caratteri all'interno del testo [90]. Un concreto esempio dell'applicazione di questa metodologia si riscontra in S. Brin (1998) [27] in cui, una serie di titoli di libri, sono stati assegnati al rispettivo autore. Una variante di questo approccio è denominato *"mutual bootstrapping"*, ed è stato realizzato per la prima volta da Riloff and Jones (1999) [105]. In esso viene generato simultaneamente sia il lessico semantico che le re-

gole per l'estrazione a partire da dati di addestramento e un da numero ristretto di termini per ogni categoria. Il funzionamento è stato orchestrato in due fasi: nella prima, attraverso l'utilizzo dell'algoritmo di *mutual bootstrapping*, viene selezionato il miglior modello di estrazione per ciascuna categoria per poi utilizzare questa conoscenza nel lessico semantico aggiornando le basi per il task successivo. Nella seconda fase si utilizza un altro algoritmo, denominato *meta-bootstrapping*, in supporto al primo, per rendere più robusta l'esecuzione. Questo approccio è stato impiegato anche da M. Pasca et al. (2006) [95] con l'aggiunta della generazione di sinonimi basata sulla proprietà della similarità semantica.

Tuttavia, come dimostrato da Lu, Tyler Tian (2009) [81], gli algoritmi ad apprendimento semi-supervisionato non possono superare le performance degli approcci supervisionati poiché solamente in grado di eguagliare i risultati di questi ultimi. Inoltre condividono con essi anche molte delle limitazioni, tra le quali quella di avere un'efficacia totalmente vincolata ai dati di addestramento.

2.1.1.3 NER ad apprendimento non supervisionato

"Informally, unsupervised learning refers to most attempts to extract information from a distribution that do not require human labor to annotate examples" [56]

In questo approccio non si utilizzano dati di addestramento annotati manualmente ma al contrario, l'algoritmo in questione, impara sulla base delle caratteristiche significative all'interno dei dati stessi riuscendo spesso ad identificare pattern e raggruppamenti di dati non distinguibili dagli esseri umani [56]. Per questo, gli algoritmi basati su unsupervised learning, vengono spesso utilizzati come strumento nelle procedure di data visualization.

Per ottenere questo tipo di conoscenza utile vengono impiegate diverse tecniche; la più nota è quella del *clustering* in cui, in base alla proprietà della similarità semantica, vengono raggruppati insieme di parole alle quali si assegna un nome che descrive tale gruppo. *WordNet* è il database più utilizzato, come supporto a questa particolare tecnica, la quale può essere implementata anche utilizzando come riferimento altre caratteristiche del linguaggio inerenti al lessico [90].

Un esempio di applicazione di questo approccio la si ritrova in Alfonseca et al. (2002) [15] in cui, un NER adibito al riconoscimento delle categorie ontologiche

associate a documenti, è stato progettato sulla base del valore di probabilità delle relative frequenze di co-occorrenza di ciascun gruppo di sinonimi presenti in WordNet. Evans et al. (2003) [45], mediante l'utilizzo dei motori di ricerca online, hanno sviluppato un NER in grado di associare, alle parole riportate con la iniziale maiuscola in un testo, i relativi iperonimi (termine che denota un vocabolo di significato più comprensivo). Shinyama et al. (2004) [113], attraverso l'assunzione che i named entities, a differenza dei nomi propri, compaiono spesso sincronicamente negli articoli di giornale, hanno realizzato un NER, basato sull'apprendimento non supervisionato, in grado di identificare categorie specifiche che raramente vengono gestite in modo corretto dagli altri software di questa tipologia. Nel lavoro di Zhang et al. (2013) [130] è stato realizzato un NER adibito all'estrazione di named entities da testi provenienti dalla disciplina della biomedica. La scelta dell'unsupervised learning è stata dettata dal fatto che si è voluto realizzare un sistema che fosse indipendente da regole e dati di addestramento in modo da rendere immediate le operazioni di estensione ed aggiornamento. Nel più recente studio di Qi et al. (2021) [97] per ovviare al problema della mancanza dei dati di addestramento in un certo ambito, l'approccio non supervisionato, è stato utilizzato per sviluppare un NER in grado di riconoscere named entities in un dominio target a partire dall'analisi di un altro dominio, chiamato "source", che condivide in parte entità presenti nel primo.

Gli algoritmi basati su apprendimento non supervisionato spesso riscontrano problemi nell'accuratezza dei risultati [39] dimostrandosi poco adatti alla gestione elementi di dominio specifico. Inoltre, i pattern individuati da questi algoritmi, risultano di difficile interpretazione e poco utili come supporto per le operazioni decisionali [72]. In aggiunta, utilizzando l'approccio non supervisionato, non esiste una concreta metrica di valutazione poiché per definizione non si può disporre di un'unità di misura dell'accuratezza precisa per la stima dei risultati, non essendo teoricamente disponibile un corpus annotato di riferimento [108].

2.1.2 Rule-Based Named Entity Recognition

I primi software di NER furono realizzati con la metodologia “*Rule-Based*” ovvero senza l’utilizzo di learner statistici ma unicamente definendo regole che dominano il funzionamento dell’esecuzione. Questo approccio è solitamente caratterizzato dall’impiego di tre componenti principali: [133]

1. Un insieme di regole per l’estrazione
2. Una lista di termini detti “*gazetteers*” che costituisce il lessico del NER
3. Una componente in grado di applicare le regole ed il lessico al testo in input

Per quanto riguarda i primi due elementi della lista, ovvero l’insieme di regole e per la lista di gazetteers, questi debbono necessariamente essere realizzati manualmente o in alternativa estrapolati da esempi sempre realizzati da esseri umani [133]. È opportuno chiarire che, sia nella letteratura scientifica che in questo elaborato, con il nome di “*dictionary-based*” si fa riferimento a particolari NER, comprendenti una componente rule-based, che fanno un utilizzo più intensivo dei gazetteers [87, 114].

Questa metodologia risulta preferibile per l’identificazione di elementi appartenenti ad un dominio specifico [133] o in alternativa per l’individuazione di entità complesse con le quali i NER statistici riscontrano più difficoltà [85]. La portabilità è il maggiore problema di questi sistemi che si dimostrano significativamente meno adeguati negli utilizzi più generici. La loro efficacia dipende strettamente dalla lista dei termini che ne costituisce il lessico e dall’insieme di regole definite per l’estrazione [133]. Inoltre, come dimostrato da Sekine e Nobata (2004) [112], lo sviluppo di un sistema NER basato su regole rimane la scelta preferibile, se non obbligata, quando non si dispone di corpora con cui poter addestrare un algoritmo statistico [90]. Solitamente questi sistemi sfruttano le proprietà: sintattiche, morfosintattiche, relative a categorie lessicali oppure anche a caratteristiche ortografiche in combinazione con i dizionari annessi (gazetteers) [85].

Un esempio di applicazione di questa tipologia di NER è quello di Rayner et al. (2014) [16], in cui è stato sviluppato un applicativo in grado di individuare le categorie lessicali proprie dell’analisi logica (Part-of-speech tagging) inerenti ad articoli scritti in lingua malese. Nello studio di Farmakiotou et al. (2000) [47] è stato progettato un NER, basato su regole, adibito al riconoscimento di tre

diverse entità presenti in testi di natura finanziaria; scritti in lingua Greca. Nel lavoro di Riaz (2010) [104] è stato realizzato un NER rule-based, per la lingua Urdu, che ha raggiunto valori superiori, in termini di accuratezza, se paragonato ai precedenti software basati su modello statistico. La scelta dell'approccio a regole, come dichiarato dallo stesso autore dello studio, è stata dettata dalla mancata disponibilità di dati di addestramento. Nello studio condotto da Zaghouani (2012) [129], in cui è stato progettato un NER per il riconoscimento di entità contenute in articoli di giornale scritti in lingua araba, non sono stati raggiunti risultati soddisfacenti sia per l'individuazione di alcune categorie di named entities che, più in generale, nei valori inerenti all'indice di recall. Popovski et al. [99] hanno sviluppato un software di questa tipologia, unicamente adibito all'estrazione di informazioni inerenti all'alimentazione, in testi contenenti dati non strutturati. Quest'ultimo è stato in grado di segnare i seguenti valori nella fase di valutazione: F1 Score: 0.9605, precision: 0.9780 e recall: 0.9437. Nel lavoro di Milanova et al. (2019) [88], mediante un approccio rule-based, è stato sviluppato un NER designato al riconoscimento ed estrazione di entità di tipologia "luogo" in testo scritto in lingua latina. La scelta di tale metodologia realizzativa è stata dettata dalla volontà di concepire un sistema che fosse indipendente da dati di addestramento annotati ma unicamente governato da un insieme di regole costruite mediante specifica conoscenza. Nella fase di valutazione di questo studio, è stato raggiunto il valore, inerente all'indice di precision, di 0.92. Lhioui et al (2017) [76] hanno sviluppato un NER adibito all'estrazione di espressioni temporali, numeriche e di nomi propri per la lingua araba standard moderna. Esso è stato progettato mediante la combinazione di un analizzatore morfologico e di regole definite utilizzando la piattaforma *NooJ Local Grammars*.

2.1.3 Sistemi Ibridi

Questa tipologia di sistemi NER è caratterizzata dall'utilizzo di due o più sistemi che svolgono collaborativamente il task di individuazione ed assegnamento delle named entities. Tipicamente sono costituiti da una componente a regole posta in collaborazione con un approccio statistico. Generalmente le competenze umane, che rappresentano il costruito rule-based, forniscono le indicazioni per la gestione dei casi più specifici, mentre la parte, costituita dal modello statistico, si pone come un supporto per la generalizzazione di tali conoscenze [133].

Un esempio dell'applicazione di questa tecnica è quello di K. Shaalan e Oudah (2013) [92], in cui è stato realizzato un sistema ibrido basato in parte su regole ed in parte su algoritmi statistici in grado di riconoscere 11 tipologie di named entities. La scelta di questo approccio è stata preferita dopo aver testato singolarmente il funzionamento delle singole metodologie. Nel lavoro di Florian et al. (2003) [49] sono stati combinati quattro diversi algoritmi statistici con un approccio rule-based. Quest'ultimo ha un utilizzo costante mentre la componente statistica viene alternata a seconda del contesto linguistico in analisi. Tra le prime applicazioni si colloca il lavoro di Borthwick (1999) [24], in cui è stato sviluppato un NER ibrido in grado di combinare una componente rule-based con un approccio basato su Maximum Entropy, addestrato sia in inglese che in giapponese (quest'ultima per testare la portabilità del sistema). Per quanto concerne la prima delle due lingue, esso è stato valutato secondo gli standard della *Message Understanding Conference (MUC) 7* posizionandosi in quarta posizione, mentre per il giapponese, è stato testato secondo i parametri di *MET-2* che possono essere considerati analoghi a quelli del *MUC-7* segnando valori che certificano l'efficacia della portabilità di questo sistema (F1-score: 83.80, precision: 91, recall: 78). Nello studio condotto da Rocktäschel et al. (2012) [106] è stato realizzato un NER per l'identificazione di menzioni di sostanze chimiche, in testi scritti in linguaggio naturale, espresse in modo informale o standardizzato. La metodologia ibrida, in questo caso, si concretizza mediante la combinazione dell'algoritmo *conditional random field* ad un approccio dictionary-based. Esso ha raggiunto un valore di F1-score di 68,1% superando, di oltre dieci punti percentuali, l'unico altro strumento di NER liberamente disponibile nel medesimo dominio applicativo.

"Therefore, it can be claimed that hybrid methods contain different single methods and form a method with higher flexibility with a high capability compared with single methods. Hybrid methods have become more popular due to their high potential and capability." [10]

I sistemi ibridi sono generalmente considerati come la scelta migliore tra i vari approcci poiché considerati più robusti ed efficienti [10]. Come illustrato in *figura 2* il loro utilizzo enterprise segna un incremento sempre maggiore di anno in anno.

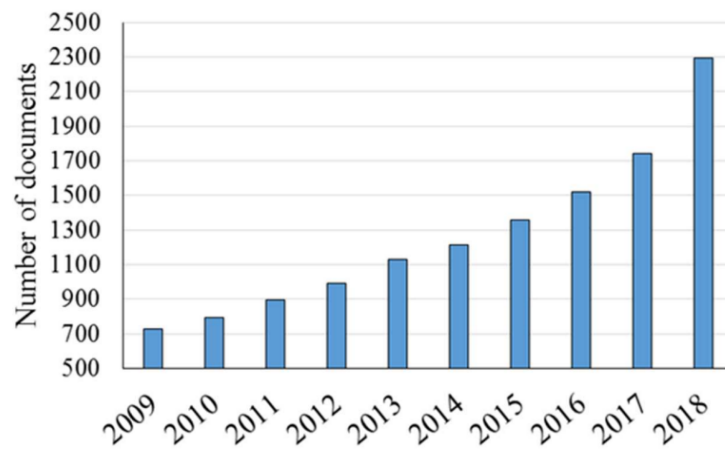


Fig. 2: La crescente tendenza dei metodi machine learning ibridi [10].

Tuttavia essi condividono non solo i punti di forza dei singoli algoritmi, dai quali sono realizzati, ma anche le relative problematiche. Queste sono, in particolar modo, inerenti ai paradigmi dell'apprendimento supervisionato e semi-supervisionato i quali necessitano di dati di addestramento annotati manualmente che, a loro volta, presentano le complicazioni descritte nelle sezioni 2.1.1.1 e 2.1.1.2.

2.2 Overview sulle taxa biologiche

"taxonomy, in a broad sense the science of classification, but more strictly the classification of living and extinct organisms, i.e., biological classification." [28]

Il termine tassonomia è composto dalle parole greche "*taxis*" (τάξις) che significa "ordine" e "*nomos*" (νόμος) che può essere tradotto in "uso". Con questa espressione si fa riferimento sia alla componente meramente pratica che alle indicazioni teoriche e concettuali che vengono utilizzate nelle procedure di categorizzazione e classificazione. L'applicazione di questi costrutti non è da intendersi come vincolata all'insieme degli organismi ma al contrario costituisce una metodologia di classificazione universalmente applicabile [124].

Esistono altre discipline che, utilizzando le tecniche proprie della tassonomia, sono adibite alla classificazione delle varie entità facenti parte del dominio delle scienze naturali. Il sistema di denominazione che regola e gestisce i nomi assegnati alle specie, inerenti alle scienze botaniche, è detta *Nomenclatura* mentre la creazione di macro gruppi, contenenti insiemi di taxa, è un task proprio della materia della *Classificazione*. Al contrario, spesso, non sono chiaramente definiti i confini di competenza tra *Sistematica* e *Tassonomia*. Queste ultime talvolta si riferiscono allo stesso contesto e le loro definizioni di dominio applicativo risultano poco chiare anche se interpretate attraverso le indicazioni degli stessi studiosi del settore. In linea generale la *Sistematica* è intesa più frequentemente come una disciplina di studi di natura più ampia che comprende anche la *Biogeografia*, *Filogenesi* e *Tassonomia* [64, 123].

Stabilire mediante un insieme di regole formali, e quindi attraverso criteri chiaramente definiti, quando due o più gruppi di parole si riferiscono o meno allo stesso taxon risulta essere un'attività connessa alle criticità dei protocolli che regolamentano la gestione delle nomenclature. Questi infatti sono relativi ad ogni specifica sotto-branca della biologia le cui regole, in molti casi, non vengono osservate nemmeno dagli studiosi stessi [64]. Inoltre, i nomi delle specie, sono considerati come dinamici per cui soggetti al cambiamento nel tempo a seconda delle valutazioni condotte degli esperti del settore. Il vincolo di unicità, che regola la creazione di una nuova nomenclatura, è inerente unicamente alle singole branche delle scienze naturali e non tiene conto quindi dell'intero insieme dei nomi scientifici delle specie [64].

Quello che al contrario risulta essere chiaro a tutte queste aree di studio è che per far riferimento a tali elementi si utilizza l'espressione "*Taxa*". Ogni singola specie vivente è quindi indicata con un "*Taxon*" il quale comprende un nome scientifico e un rango. Il ranghi delle specie, solitamente indicati in lingua latina [125], possono essere di più tipologie; quelli standard sono [64]:

- Regnum
- Phylum
- Classis
- Ordo
- Familia
- Genus
- Species

Può anche accadere che venga utilizzata una parola come prefisso, con funzione qualificativa, in aggiunta al nome del rango. Queste solitamente sono parole come "*super*", "*sub*", "*infra*" e discriminano ulteriormente la specie a cui sono anteposte. Esistono anche ranghi denominati "*tribus*" che sono composti da una parola che si interpone tra Genus e Species. Inoltre sono presenti gruppi di ranghi per indicare specie al di sotto delle sottospecie; questi vengono indicati mediante il prefisso "*sub*" e si utilizzano per far riferimento a ranghi definiti come "infra specifici". I nomi dei ranghi inerenti ai generi (*Genus*) sono composti da un solo termine e vengono per questo definiti come "*uninomen*" mentre, per quanto concerne le altre categorie, si ricorre all'utilizzo di più termini. Esistono suffissi che costituiscono degli standard come ad esempio "*idea*" che viene utilizzato per la *Zoologia* mentre "*aceae*" per la *Botanica*; come indicato nella *tabella 1* [64].

	Plants	Algae	Fungi	Animals	Bacteria
phylum	phyta	phycota	mycota		
subphylum	phytina	phycotina	mycotina		
class	opsida	phyceae	mycetes		ia
Subclass	idea	phycidae	mycetidae		idea
Superorder	anae	anae	anae		
Order	ales	ales	ales		ales
Suborder	ineae	ineae	ineae		ineae
Infraorder	aria	aria	aria		
Superfamily	acea	acea	acea	oidea	
Family	aceae	aceae	aceae	idea	aceae
Subfamily	oideae	oideae	oideae	inae	oideae
Tribe	eae	eae	eae	ini	eae
Subtribe	inae	inae	inae	ina	inae

Tabella 1: Suffissi standard per i nomi dei ranghi superiori al Genus.
Esempio tratto da: *Taxa Merging Discussion, iMarine Project Wiki* [64]

I nomi dei Genus e delle categorie superiori vengono riportati con l'iniziale maiuscola. Diversamente, quello di una Species, è composto anche dallo *Species epithet/specific epitheton*. Quest'ultimo, mai con l'iniziale maiuscola, viene reso in corsivo allo stesso modo delle categorie sottostanti, come illustrato in questo esempio:

Family Semelidae
Genus <i>Abra</i>
Species <i>Abra alba</i>

esempio tratto da: *Taxa Merging Discussion, iMarine Project Wiki* [64]

Il nome della persona che ha originariamente descritto la specie, viene solitamente riportato adiacente a quello del rango, seguito dall'anno della pubblicazione di tale descrizione. Può capitare che ad esso venga affiancato anche il Genus ed ancor più raramente dalla *Familia* e superiori, come indicato nell'esempio seguente.

Family Semelidae
Genus <i>Abra</i> Leach in Lamarck, 1818
Species <i>Abra alba</i> (W. Wood, 1802)

esempio tratto da: *Taxa Merging Discussion, iMarine Project Wiki* [64]

Un altro particolare caso è illustrato sempre dal medesimo esempio, in cui, l'autore che ha firmato la pubblicazione è *Lamarck* ma la persona che ha effettivamente scritto descrizione è *Leach*. Sempre in questo esempio, viene riportato anche

Leach. Ciò è stato fatto in funzione di riconoscimento dell'impegno intellettuale di quest'ultimo ma, stando alle normative standard, solamente *Lamarck* dovrebbe essere riportato.

I nomi infraspecifici sono composti da più di due parti. Così una sottospecie in Zoologia risulterebbe scritta come:

<i>Uca lactea annulipes</i> (H. Milne-Edwards, 1837)
--

esempio tratto da: *Taxa Merging Discussion, iMarine Project Wiki* [64]

Al contrario, non esiste una regolamentazione per i nomi di grado inferiore alla sottospecie in Zoologia. A causa di questo, in linea generale, è ritenuto che un *trinomen* sia in tutti i casi una sottospecie escludendo di fatto la possibilità che si tratti di una *varietà* o di una *forma*. Per queste ultime, il rango viene indicato anteponendo le diciture “*var*” o “*f*” davanti alla parte del nome corrispondente:

<i>Balanus amaryllis f. nivea</i> Gruvel
--

esempio tratto da: *Taxa Merging Discussion, iMarine Project Wiki* [64]

Nella Botanica, per indicare una sottospecie, si antepone “*ssp*” al nome corrispondente. Questo perché i ranghi infraspecifici sono coperti da codice. Ad ogni modo, questa prassi, non è realizzata nei maggiori knowledge base del settore ovvero *OBIS* e *WoRMS*.

I sottogeneri vengono riportati all'interno delle parentesi con l'iniziale maiuscola.

<i>Uca (Paraleptuca)</i> Bott, 1973 <i>Uca (Paraleptuca) lactea</i> (De Haan, 1835) <i>Uca (Paraleptuca) lactea annulipes</i> (H. Milne-Edwards, 1837)
--

esempio tratto da: *Taxa Merging Discussion, iMarine Project Wiki* [64]

In quest'ultimo esempio, l'ultimo nome fa riferimento allo stesso taxon di "*Uca (Paraleptuca) lactea annulipes* (H. Milne-Edwards, 1837)", così come l'"*Uca (Paraleptuca) lactea* (De Haan, 1835)" e "*Uca lactea* (De Haan, 1835)" che sono due stringhe alternative che si riferiscono alla stessa specie.

Ogni qualvolta viene descritto un nuovo taxon esso deve essere necessariamente

assegnato ad uno o più *specimen* di "*tipo*" (esemplari di riferimento). Questo fornisce al nome in questione una base concettuale non contestabile. Il "*tipo*" è, sia in Zoologia che in Botanica, un esemplare di riferimento che viene chiamato "*olotipo*" mentre il luogo dove esso è stato ritrovato viene denominato "*località tipo*". Il primo deve essere disponibile presso un luogo adibito alla conservazione e alla consultazione da parte di altri studiosi in modo da garantire la validità della suddetta descrizione [64].

2.2.1 Casi di sinonimia e omonimia

Uno degli obiettivi della scienza della tassonomia è quello di cercare di realizzare ragionevolmente una corrispondenza univoca e distinta tra nome e taxon. Nonostante ciò, sono presenti molti casi di omonimia e sinonimia, in cui nei primi, più taxa fanno riferimento ad un unico nome scientifico mentre, nei secondi, gruppi di nomi scientifici rimandano allo stesso taxon [64, 21]. Molto spesso questi fenomeni sono causati da errori commessi dagli stessi studiosi oppure possono essere il prodotto di un contrasto di opinioni derivante dal naturale dibattito scientifico in merito.

Riguardo i casi di omonimia, questi sono causati solitamente dall'erronea trascrizione del nome degli autori della pubblicazione originale del taxon da parte di studiosi che non hanno svolto tutte le necessarie operazioni di controllo e verifica. Seppur poco frequenti, si sono verificati casi in cui uno stesso taxon venga pubblicato omonimamente dalla medesima persona. Da questo segue implicitamente che il nome dell'autore della pubblicazione sia, nella maggioranza dei casi, un elemento che permette di ottenere una discriminazione in grado di eludere un possibile caso di omonimia.

Nei database di riferimento, la codifica, può essere composta da tutta la catena tassonomica alla quale viene quindi assegnato un codice. Questi ultimi, possono essere standard (indicati come "*ref*") ed identici in tutti i database oppure non standard (indicati come "*code*") ovvero specifici per ognuno di essi poiché ancora non approvati dalla totalità della comunità scientifica. Esistono però diverse istanze di omonima che sono perfettamente valide nello specifico dominio ma con nomi scientifici che si riferiscono ad entità completamente differenti. Questo significa che in questi casi si deve far riferimento o al codice o riportando l'interezza della specifica catena tassonomica [64].

Al contrario, la sinonimia, si manifesta maggiormente quando vengono classificate specie per la seconda volta. In questi casi può accadere che l'autore della trascrizione non sia al corrente dell'esistenza della precedente e venga così generato un nuovo nome per lo stesso taxon. Quando queste casistiche vengono identificate si parla di "*junior synonym*"; termine con il quale si fa riferimento ad un sinonimo che non deve essere utilizzato. Si parla invece di "*subjective synonyms*" quando due nomi sono trascritti con i medesimi termini ma associati a differenti olotipi. Può anche accadere che un taxon, già classificato, venga nuovamente descritto da un nuovo autore e categorizzato sotto un differente Genus. In queste eventualità vi è una corrispondenza in termini di omotipicità ed oggettività.

Il nome dell'autore della pubblicazione di un determinato taxon può essere riportato diversamente a seconda della branca della biologia nella quale lavora. Ad esempio, nella zoologia, questi è inserito all'interno delle parentesi mentre nella botanica viene riportato, se presente, l'autore della pubblicazione originale in aggiunta a quello della successiva; così come indicato nell'esempio seguente: [64]

Uca (Paraleptuca) Bott, 1973
Uca (Paraleptuca) lactea (De Haan, 1835)
Uca (Paraleptuca) lactea annulipes (H. Milne-Edwards, 1837)

esempio tratto da: *Taxa Merging Discussion, iMarine Project Wiki* [64]

2.2.2 Specific epithet

Per descrivere in modo più preciso i nomi delle specie ci si serve dello "*specific epithet*". Questo è solitamente costituito da un avverbio che ha la funzione di rendere più specifica l'indicazione fornita dal Genus. Dato il vincolo imposto dalla suddetta parte del discorso, esso deve essere necessariamente concordante con il genere latino del Genus. Nell'eventualità in cui lo specific epithet risulti essere un nome, secondo le regole proprie della grammatica latina, questi può non essere concordante in termini di genere con il Genus con il quale è associato. Questa caratteristica è stata causa di errori in quanto sono stati utilizzati nomi maschili e femminili insieme nei nomi delle specie rendendo così non chiara l'identificazione del genere vero e proprio inerente al taxon.

In ultima analisi, nelle circostanze in cui lo specific epithet risulti essere un gen-

itivo, una forma possessiva di un luogo o persona; esso deve essere latinizzato mediante rigide regole grammaticali. Queste, non sempre osservate a pieno, sono state fonte di errori nei casi in cui il nome sia stato assegnato per la prima volta. Ad esempio una specie, che prende il nome da "*Edward*", è possibile che possa diventare "*SomeGenus edwardi*" o "*SomeGenus edwardii*". Queste due forme possono poi essere utilizzate entrambe nella letteratura successiva (esempio fittizio). Le differenze, in questo caso, sono probabilmente solo varianti lessicali che fanno riferimento allo stesso taxon il quale ha subito un erroneo processo di latinizzazione [64].

2.2.3 Indicazione dell'anno di pubblicazione

In Zoologia, come in generale nella letteratura scientifica, l'anno di descrizione di una specie viene posto successivamente a quello dell'autore. Quest'ultimo può essere riportato sia prendendo come riferimento l'anno mostrato sulla copertina dell'opera stampata, sia considerando l'anno in cui la pubblicazione è diventata disponibile alla consultazione. È possibile che queste cifre siano diverse in quanto può intercorrere un ammontare significativo ed arbitrario di tempo tra la stampa della copertina e la sua pubblicazione [64].

2.3 Overview sui sistemi di taxa identification

L'insieme dei software in grado di realizzare task di taxa identification basano il loro funzionamento sull'applicazione del "*lexical matching*". Esso, utilizzato anche per le operazioni di *spell checking* e *word preprocessing*, viene solitamente costruito mediante algoritmi *general purpose* come ad esempio: [21]

- *Distanza Damerau–Levenshtein* (Bard (2007) [19]), basata sulla *Minimum edit distance* di Levenshtein, sviluppata nel 1966
- *N-grams* (Owolabi e McGregor (1988))
- *Soundex* (Odell (1956))

Per quanto concerne le applicazioni nel campo delle scienze naturali, trova collocazione il sistema *Global Names Architecture*. Esso, come dichiarato all'interno del servizio web dedicato:

"The Global Names Architecture (GNA) is a system of web-services which helps people to register, find, index, check and organize biological scientific names and interconnect on-line information about Species."[54]

si propone come un servizio web in grado di assistere l'utente a trovare, indicizzare, registrare, controllare, organizzare i nomi scientifici e rendere interconnesse le informazioni on-line riguardanti le varie specie. Questo sistema nasce grazie al supporto del "*Global Biodiversity Information Facility* (GBIF) [41] e del relativo programma *ECAT* [52], così come da "*Encyclopedia of Life*" [126]. Mediante l'utilizzo di GNA è stato compilato il database, composto da nomi tassonomici e relative varianti, utilizzato dal software *Global Names Index* (GNI) [96] che incorpora circa 20 milioni di entità. Esso include anche funzionalità di ricerca (limitate all'utilizzo mediante wildcard) ed un parser ("*GNI Parser*" [26]) utilizzato per la suddivisione di nomi tassonomici in singole unità.

Nel lavoro di Berghe et al. (2015) [21] è stato realizzato "*BiOnym*", un applicativo adibito al task di taxa name matching strutturato tramite un effettivo isolamento tra le componenti di: lista dei nomi di riferimento, criteri di ricerca e matching

engine. Questi, disponibili in più varianti, sono intercambiabili dall'utente finale al quale è concessa anche la possibilità di poter utilizzare un Taxonomy Authority file (TAF) esterno e di modificare il funzionamento interno del programma tramite una piattaforma appositamente sviluppata. Queste scelte progettuali sono state dettate dal fatto che si è voluto realizzare un software flessibile e trasparente in contrapposizione agli approcci in cui sia i dizionari di riferimento che le regole per il matching non sono accessibili all'utente poiché cablate all'interno dell'engine.

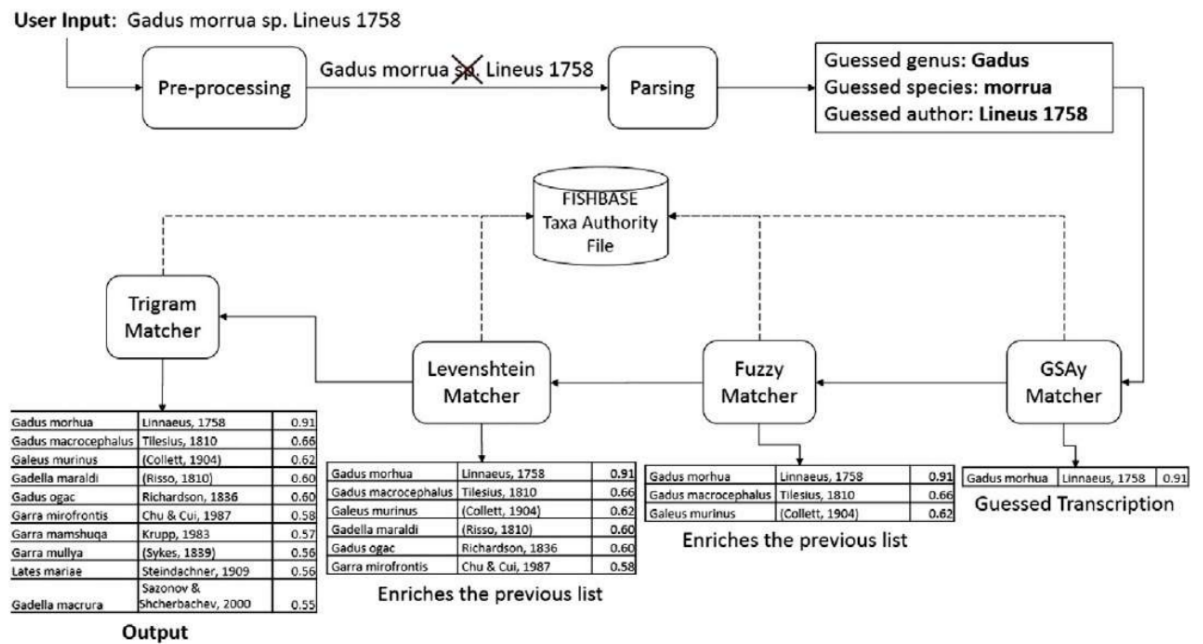


Fig. 3: Schema del workflow di BiOnym.
Esempio tratto da Berghe et al. (2015) [21]

La figura 3 raffigura uno dei possibili processi di matching del software in questione, il quale procede secondo una serie di step impostabili dall'utente. A partire da una stringa inserita dall'utente esso compie un'operazione di preprocessing nella quale la stringa in questione viene resa conforme alle fasi di analisi successive. In seguito, viene utilizzato un parser (nel caso della Figura3 "REGEXP") adibito all'identificazione delle componenti di Genus, Species ed autore all'interno della stringa pre-processata. Il risultato, viene sottoposto all'analisi da parte di più algoritmi di matching facenti riferimento ad uno specifico TAF. Essi costituiscono una catena nella quale la maggiore priorità viene assegnata a quelli che eseguono questa operazione per primi. Ognuno di essi produce una lista di possibili trascrizioni per la determinata stringa di input la quale viene restituita in output all'utente.

Un altro esempio di realizzazione di questa tipologia di software, si colloca in "*TAXAMATCH*" [103]. Sviluppato mediante l'approccio del "*fuzzy name matching*", esso viene utilizzato come base da svariati applicativi di taxon name matching come WoRMS e FishBase. L'implementazione di riferimento di questo software, disponibile liberamente online, permette di modificare parametri inerenti soprattutto alle modalità di utilizzo del TAF.

La variante di *TAXAMATCH* adottata da WoRMS (che prende il nome di *The WoRMS Taxon matcher* [127]) utilizza un TAF specializzato nel riconoscimento di specie marine. Questi si serve del parser "*GNI Parser*" per le operazioni di suddivisione delle entità. Di contro, FishBase utilizza un sistema di name matching articolato in quattro differenti passaggi impiegando in parte l'algoritmo GSay ("*Genus-Species-Authority-year*") e in parte BiOnym. Anche l'applicativo *Taxonomic Name Resolution Service* [26] include un port di *TAXAMATCH* scritto nei linguaggi PHP e MySQL. Esso, disponibile con un interfaccia web ed un servizio REST, è in grado di standardizzare i nomi tassonomici in modo da poter rinvenire la stessa entry in più TAF.

Per quanto riguarda i software che non utilizzano come base il software sviluppato da Rees [103]; Kluyver et al. [68] hanno realizzato "*Taxonome*" un tool di Lexical matching adibito al trattamento dei taxa ed utilizzabile mediante un'interfaccia grafica (GUI) concepita per facilitare l'esperienza utente. Questo programma include funzionalità anche per standardizzare l'insieme di informazioni di distribuzione ad una collezione distinta di regioni; utilizzando anche lo schema geografico definito da TDWG (*International Taxonomic Database Working Group*) come supporto per la registrazione delle distribuzioni di piante. In questo lavoro, così come per *Global Biotic Interactions programme* [98], il task di taxon name matching è strutturato mediante la combinazione dei seguenti applicativi: [21]

- "*Encyclopedia of Life*" [126]
- "*Taxize*" [29]
- "*Taxonomic Nomenclature Checker*" [118]
- "*gnparser*" [89]

2.4 Overview su NER per taxa

Gli applicativi in grado di eseguire task di Named Entity Recognition adibiti al riconoscimento ed estrazione di taxa all'interno di testo, sono realizzati mediante le più svariate metodologie. In seguito, è fornita una panoramica del software esistente comprensivo di questa specifica funzionalità.

Nel lavoro condotto da Meraner (2019) [87] è stato sviluppato un software designato al NER di nomi scientifici (taxa) e vernacolari di piante contenute in testi di vario genere; scritti in lingua inglese o tedesca. Utilizzato un sistema statistico con paradigma semi-supervisionato sono state scelte nove categorie gerarchiche di etichette per la classificazione; fattore che ha garantito la robustezza del software nei vari domini testuali. Per ovviare al problema della creazione manuale di gazetteers, essi sono stati realizzati automaticamente mediante il paradigma del *Dictionary Based Tagging* per poi essere successivamente impiegati per annotare i dati di addestramento. Tuttavia questi ultimi, realizzati mediante il supporto delle espressioni regolari (*regex*), non hanno soddisfatto i requisiti qualitativi minimi, rendendo doverosa la creazione di corpora annotati manualmente i quali sono stati utilizzati in varie combinazioni con i primi nelle operazioni di addestramento e valutazione del software. Questo NER è basato su un modello di deep learning denominato *long-short-term-memory* (LSTM) bidirezionale con uno strato *conditional random field* (CRF). Gli esiti, in termini di valutazione, sono stati 86% di F1-score per la lingua inglese e 94% dello stesso indice per il tedesco. Nel software sviluppato in questa Tesi non è stata presa in considerazione l'architettura in questione perché non poteva soddisfare i requisiti fondamentali (ad esempio riapplicabilità, flessibilità, mancanza di dati di addestramento, supporto multilingua) imposti come requisiti fondamentali per il progetto realizzato (sezione 5).

Mediante un altro approccio, Pafilis et al. (2013) [93] hanno sviluppato un NER adibito al riconoscimento di nomi scientifici di specie ed altre tipologie di taxa nel testo, utilizzando il paradigma del *Dictionary Based Tagging*. Per stimare con la massima precisione le performance è stato costruito, attraverso l'annotazione manuale, uno specifico corpus (*Species-800*) composto da 800 Abstract provenienti da varie riviste; fattore che ha permesso di poter rappresentare al meglio i vari scenari operativi possibili. Questo sistema è stato utilizzato per etichettare i nomi degli organismi presenti nel database *Medline* rendendo l'output di questo processo pubblico, così come il software nella sua interezza. Come dichiarato dagli autori di questo studio, l'applicativo in questione risulta essere più efficiente in

termini di tempi di esecuzione, precision e recall nell'annotazione di Species-800 se paragonato a LINNEAEUS [53]. Quest'ultimo, realizzato da Genrer et al. (2010), è concepito sul paradigma del "*Dictionary Based Tagging*" ed è adibito all'identificazione di nomi scientifici di specie in documenti specifici di ambito biomedico. Esso, in grado anche di disambiguare i vari contesti linguistici, è stato valutato sulla base di corpora gold standard raggiungendo le performance di 94% per recall e 97% per precision. Ciò nonostante, nel software realizzato, è stato scelto di non utilizzare o rielaborare componenti (intesi sia come gazetteers che come insieme di regole adibite al riconoscimento) propri di tali approcci poiché eccessivamente connessi alla specificità del dominio della biomedicina.

Nel lavoro di Koning et al. (2005) [69] mediante un approccio rule-based, strutturato tramite una combinazione di regole contestuali ed un'insieme di termini lessicali, è stato sviluppato un sistema NER per i nomi di natura tassonomica contenuti in testi appartenenti alla letteratura della biologia. Questo software ha segnato prestazioni superiori al 96% per il valore di precision e il 94% per recall. Per quanto concerne l'insieme dei termini lessicali, essi non comprendono taxa ma sono composti unicamente da termini inglesi dal significato generico. Essi sono stati ricavati utilizzando due database semantico-lessicali per la lingua inglese: WordNet e SPECIALIST [86]). Tale studio condivide la problematica legata all'obsolescenza dell'insieme dei named entities identificabili al quale si sommano le difficoltà di portabilità del codice in questione.

Nel lavoro di Naderi et al. (2011) [91] è stato realizzato *OrganismTagger* un software in grado di estrarre menzioni di organismi all'interno di testo appartenente a letteratura scientifica. Esso, sviluppato mediante un'architettura ibrida (rule-based/machine learning), è in grado di generare automaticamente risorse lessicali ed ontologiche a partire dalla consultazione dei dati disponibili nel *NCBI Taxonomy database* facilitando le operazioni di aggiornamento da parte degli utenti finali. Ogni organismo rilevato viene normalizzato ed associato ad un nome canonico mediante la funzionalità di risoluzione di abbreviazioni ed acronimi. Inoltre, viene stabilito un collegamento tra il match e la relativa entità all'interno del suddetto database; comprensiva di codice identificativo. Per quanto concerne la fase di valutazione, *OrganismTagger* è stato testato su un corpus annotato manualmente raggiungendo i valori di 95% in precision, 94% recall e 97.5% in accuracy. Questa operazione è stata svolta anche utilizzando il corpus di valutazione "*Linnaeus-100*", realizzato per lo studio precedentemente menzionato [53], segnando i valori di 99% in precision, 97% recall e 97.4% in accuracy. Nel software sviluppato in questo elaborato si è optato di non utilizzare un approccio

ibrido date le complicazioni descritte nella sezione 2.1.3 .

Nel lavoro di Sautter et al. (2006) [110] è stato sviluppato un applicativo progettato al riconoscimento di taxa all'interno del testo utilizzando come base di partenza il software "*TaxonGrab*" realizzato precedentemente da Koning et al. (2005) [69] impiegato in combinazione con altri algoritmi per concretizzare un approccio che combina più passaggi definiti a regole con un classificatore preventivamente addestrato; come illustrato dalla *figura 4*. I risultati della valutazione

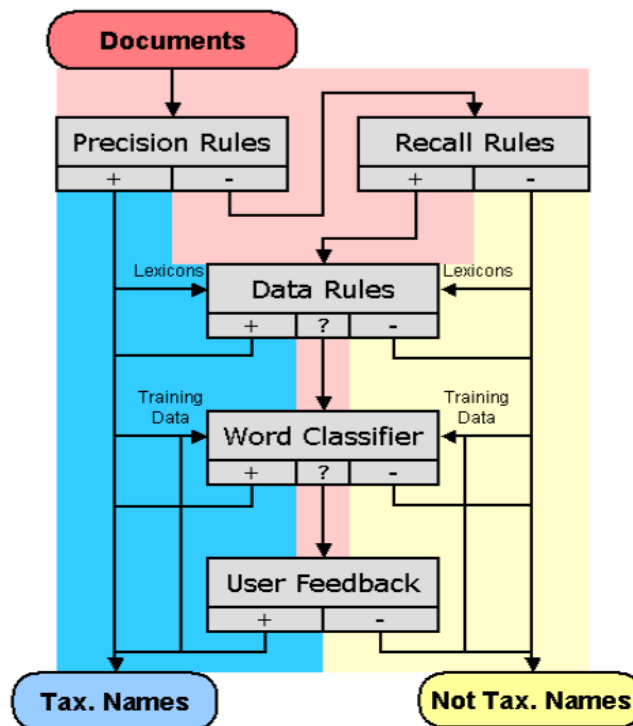


Fig. 4: Processo di classificazione del software *FAT*.
Immagine tratta da Sautter et al. (2006) [110].

dimostrano che, nel medesimo dominio applicativo, questo strumento ha incrementato le performance, inerenti all'indice della precision, dell' 1.7% raggiungendo il valore di 99.7% in favore di TaxonGrab (96%). La mancata disponibilità di corpus/corpora annotati in aggiunta alle problematiche degli approcci ibridi ha conseguito la scelta di non sviluppare un applicativo con approccio analogo nel lavoro svolto in tale elaborato.

Habibi et al (2017) [59] hanno sviluppato un NER basato sul metodo *LSTM-CRF* (*Long short-term memory - Conditional random field*) progettato da Lample et al. (2016) [71]. Questo è totalmente indipendente dal dominio poiché non utilizza alcun tipo di conoscenza di base. L'architettura di LSTM-CRF è composta da tre strati principali: il primo, a partire dal testo non processato in input, genera

una rappresentazione vettoriale dello stesso chiamata "*embedding*". Questa va ad alimentare il secondo strato LSTM che è incaricato delle operazioni di "*refining*". La parte finale consiste nell'utilizzo dello strato "*CRF*" che utilizza l'algoritmo di *Viterbi* per identificare la migliore sequenza di stati all'interno del vettore. Il software è pensato per il riconoscimento di cinque entità nominali ovvero: sostanze chimiche, malattie, Species, geni/proteine e linee cellulari contenute nei testi di letteratura biomedica. Per la fase di valutazione sono stati effettuati 33 passaggi basati su 24 differenti corpora gold standard e confrontati con i punteggi dei migliori software NER specifici del rispettivo dominio. I risultati hanno evidenziato che LSTM-CRF è migliore in 28 dei 33 casi con uno scarto in media di 5 punti percentuali. Tuttavia, come nel lavoro condotto da Meraner (2019) [87], la non conformità di questo applicativo ai requisiti di riapplicabilità e flessibilità in aggiunta all'assenza sia di dati di addestramento validi che alla funzionalità di supporto multilingua, ha reso non adoperabile questo approccio nella soluzione proposta in questa Tesi.

Guillarme e Thuiller (2021) [73], hanno recentemente sviluppato un NER in grado di riconoscere taxa in documenti di argomento ambientale chiamato *TaxoNERD* (*Taxonomic Named Entity Recognition using DeepModels*). Esso è composto da due modelli deep neural network (DNN). Data la scarsa disponibilità di dati di addestramento annotati manualmente in questo specifico dominio, si è deciso di sfruttare la conoscenza di altri modelli DNN presenti (addestrati in testi biomedici) mediante la tecnica del transfer learning. Il risultato di questo processo ha dimostrato che, per quanto riguarda i testi di natura ecologica, l'applicativo sviluppato in questo studio ha raggiunto risultati più elevati se paragonato agli altri software NER. Inoltre, come sottolineato dagli stessi autori dello studio, un significativo margine di miglioramento è costituito dalla creazione di un algoritmo in grado di imparare da corpora, composti da testi ecologici, non annotati. Come sottolineato da Pan et al. (2009) [94], il paradigma del Transfer Learning, risulta avere successo quando si utilizza come partenza un modello avente un dominio applicativo che comprende il target al quale si è interessati; fattore che non permette quindi l'impiego nel riconoscimento della totalità dei taxa i quali non fanno parte di uno specifico sotto insieme ma che al contrario costituiscono un macro gruppo di entità.

Kaewphan et al. (2018) [65] hanno presentato un sistema per l'identificazione automatica di entità nominali presenti in testi di letteratura biomedica. Questo, basato sul modello "*conditional random field*" (*CRF*) è in grado anche di collegare le entities scoperte alla corrispettiva entry all'interno di un database realizzato

ad hoc. Tale funzionalità è realizzata mediante l'algoritmo di *fuzzy character n-gram matching*. La classificazione, che comprende 9 differenti entità, realizza il processo di disambiguazione sfruttando sia la struttura ontologica delle entità nominali che il contesto linguistico circostante. La specificità del dominio biomedico, in aggiunta alle complicazioni derivanti dall'utilizzo dell'architettura CRF, rendono non usufruibile l'applicativo sviluppato in tale studio per il software sviluppato in questo elaborato.

Nel lavoro di Leary (2014) [74] è stato implementato un servizio online adibito al riconoscimento di tutte le tipologie di taxa contenute in un testo. Esso, disponibile anche mediante opportune API, utilizza un approccio dictionary-based. Il riconoscimento (che non comprende i nomi vernacolari) non si limita solamente al Genus e Species ma si realizza nell'interezza del relativo taxon includendo *Kingdom, Phylum, Class, Order, Family, Genus, Species, SubSpecies*. Non sono però rese note le performance poiché esso risulta essere sprovvisto di una documentazione ufficiale, fattore che ha scoraggiato la possibilità di riutilizzo e/o rielaborazione dei costrutti propri di tale applicativo nel software sviluppato.

3 Metodo

3.1 Descrizione generale dell'approccio usato

Come precisato nell'introduzione di questo elaborato, l'applicativo sviluppato ha come scopo la creazione di un NER per il riconoscimento di nomi scientifici di specie all'interno di testi di dominio biologico; nello specifico documenti inerenti alle scienze acquatiche e della pesca. Anche se, propriamente, il sistema implementato è un sistema di Named Entity Recognition and Linking (perché possiamo associare le entità trovate in precise posizioni nel testo ad una base di dati tassonomici) in questa Tesi userò l'acronimo NER anche per l'algoritmo implementato, per uniformare la nomenclatura a quella degli algoritmi già presenti nel sistema NLP Hub che hanno la stessa struttura dell'output.

L'applicativo sviluppato, concepito per essere indipendente dalla lingua in input, è stato progettato mediante il paradigma rule-based e si serve di dizionari di supporto per il task di identificazione. Questi ultimi, descritti in dettaglio nella sezione 3.1.1, costituiscono non solo l'insieme di termini identificabili dal sistema ma comprendono anche una lista di vocaboli inglesi necessari per incrementare le performance del programma.

Il software in questione è stato concepito per essere conforme con i seguenti requisiti:

1. Impiego del paradigma non supervisionato; da imputare alla mancanza di dati di addestramento nel relativo dominio applicativo.
2. Soluzione adattabile anche al dominio applicativo dei tesauri.
3. Soluzione adattabile a sottodomini biologici o a i knowledge base tassonomici.
4. Basso costo di manutenzione e aggiornamento.
5. Capitalizzare sulla conoscenza prodotta dal progetto europeo i-Marine, che si è occupato anche di riconoscimento tassonomico per grandi knowledge base.

6. Essere multilingua ovvero indipendente dalla lingua in input.

Per quanto concerne il dominio applicativo del progetto, si è deciso di includere unicamente le entità di *Genus* e *Species* e la loro combinazione (nome scientifico). Questo è dovuto alla necessità di realizzare un sistema avente una struttura valida anche per l'identificazione dei Thesauri (sezione 3.3), funzionalità che è stata concretamente sviluppata ma ancora priva di test di valutazione dedicati.

Il software è stato scritto mediante il linguaggio di programmazione *Java* (versione 8) avvalendosi unicamente dei package integrati, ovvero provenienti dalla Java API (*java.io*, *java.nio* e *java.util*). Tali scelte sono state imposte dalla necessità di integrare il software nell'infrastruttura D4Science del CNR. Composto da tre classi, l'applicativo in questione, realizza il processo NER attraverso una moltitudine di step distinti. La *figura 5* rappresenta approssimativamente sia l'architettura che il funzionamento del programma. A partire da un file di input in formato *"TXT"*, la classe *"Orchestrator"* gestisce ad alto livello tutte le fasi del NER. Queste, nell'atto pratico, sono realizzate dalle classi *"ResearchObjectSpecies"* ed *"EfficientSearchInText"*. Come illustrato in *figura 5*, il file di testo in ingresso, selezionato dall'utente, viene trasformato in array mediante il metodo *"Capture"* così da essere conforme alla ricerca delle rispettive entità all'interno del primo dei due dizionari contenenti i nomi dei taxa mediante il metodo *"searchParallel"* incluso nella classe *"EfficientSearchInText"*. Al termine di questo processo viene generata una nuova lista, analoga alla precedente, ma con la differenza che le entità che sono state trovate all'interno del dataset in questione, risultano "marcate". Successivamente, mediante il metodo *"enrich"* quest'ultima viene ulteriormente controllata tramite una serie di operazioni che mirano a verificare l'effettiva veridicità dei match. Ciò viene fatto mediante il check delle entità all'interno della lista contenente i termini inglesi più comuni seguita dal riutilizzo del metodo *searchParallel* applicato utilizzando il secondo dei due dizionari annessi. Una volta completata questa fase, sempre tramite il metodo *enrich*, vengono generate due liste contenenti le annotazioni definitive conformi al formato *TXT* e *JSON* che sono poi incorporate nei rispettivi file di output mediante i metodi *"materializeText"* e *"materializeJSON"*.

La sezione 3.2 è adibita alla descrizione approfondita dell'applicativo fin qui descritto; corredata da relativo pseudo codice in supporto alla comprensione dei vari algoritmi realizzati; la sezione 3.1.2 è inerente alla descrizione dei formati di input ed output utilizzati. In aggiunta, l'applicativo in questione è stato integrato all'interno di *"NLPHub"* [34]. La sezione 3.5 è inerente all'illustrazione in

linea generale di quest'ultimo e alle metodologie impiegate per l'integrazione del software sviluppato.

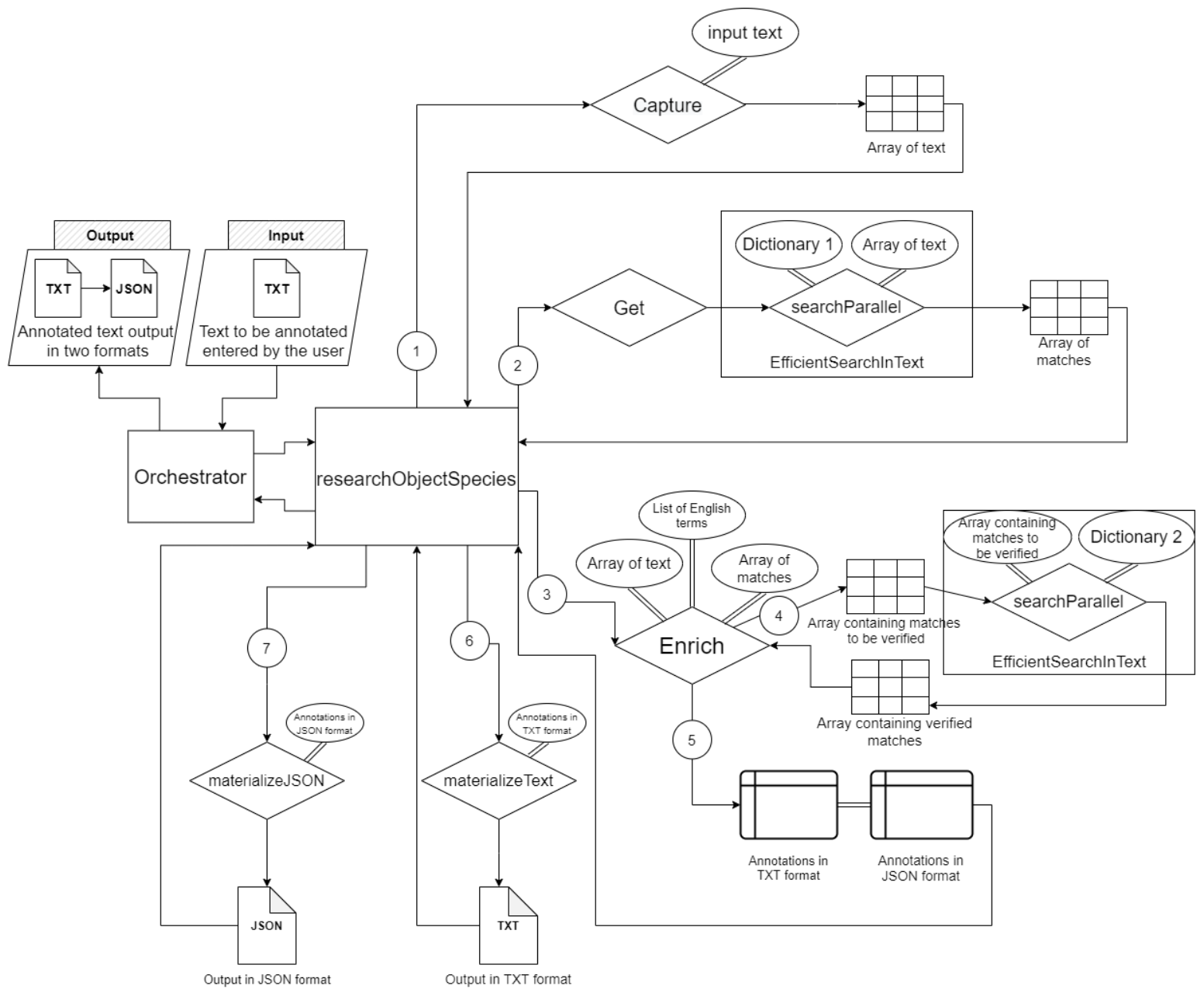


Fig. 5: Diagramma del workflow inerente al software realizzato

3.1.1 Dataset utilizzati

"The GBIF Backbone Taxonomy is a single, synthetic management classification with the goal of covering all names GBIF is dealing with. It's the taxonomic backbone that allows GBIF to integrate name based information from different resources, no matter if these are occurrence datasets, Species pages, names from nomenclators or external sources like EOL, Genbank or IUCN. This backbone allows taxonomic search, browse and reporting operations across all those resources in a consistent way and to provide means to crosswalk names from one source to another." [8]

L'insieme dei gazetteers utilizzati nel software sviluppato è costituito unicamente dal database realizzato da Global Biodiversity Information Facility (GBIF) disponibile presso il relativo sito web [8]. Utilizzando come base di partenza il database *Catalogue of Life* [115] è stato costruito mediante la combinazione di 100 differenti knowledge base del settore. Attualmente è formato da oltre 6 milioni e mezzo di entità, ciascuno inerente ad uno specifico taxon per il quale sono disponibili le informazioni di:

- taxonID
- datasetID
- parentNameUsageID
- acceptedNameUsageID
- originalNameUsageID
- scientificName
- scientificNameAuthorship
- canonicalName
- genericName
- specificEpithet
- infraspecificEpithet

- taxonRank
- nameAccordingTo
- namePublishedIn
- taxonomicStatus
- nomenclaturalStatus
- taxonRemarks
- kingdom
- phylum
- class
- order
- family
- Genus

Questo insieme di dati è stato utilizzato per la creazione dei due differenti dataset impiegati all'interno dell'applicativo sviluppato. Il primo dei due (indicato in questo elaborato come "*dataset 1*") contiene unicamente i valori compresi nelle colonne di *Genus* e *specificEpithet* riportati in un'unica lista (senza ripetizioni). Questa, nello specifico, è stata costruita mediante i record il cui valore di *taxonRank* è indicato come *Genus* e *scientificNameAuthorship* come valore non nullo o *taxonomicStatus* equivalente a *doubtful*. Un estratto di essa è rappresentato nell'esempio successivo:

Madeconeuria
Oxycomanthus
Gibbochonetes
Enigmocoma
deceptor
basalis
psychrasema
sagittigera
dichroa

Esempio delle entità presenti nel dataset 1 utilizzato nella fase di matching.

Il secondo dei due dataset realizzati (indicato in questo elaborato come "dataset 2") è costruito sulla base dei medesimi criteri di scelta dei record ma riportando le entry di *Genus* e *specificEpithet* sia in forma estesa che puntata; come illustrato nell'esempio seguente:

<i>E. deceptor</i>
<i>Eopenthes deceptor</i>
<i>E. basalis</i>
<i>Eopenthes basalis</i>
<i>C. psychrasema</i>
<i>Cochylis psychrasema</i>
<i>C. sagittigera</i>
<i>Cochylis sagittigera</i>
<i>B. dichroa</i>
<i>Baputa dichroa</i>

Esempio delle entità presenti nel dataset 2 utilizzato come supporto alla fase di matching.

In aggiunta a questi due dataset, l'applicativo sviluppato si serve di un ulteriore insieme di dati. Esso, indicato in questo elaborato come "lista dei termini inglesi", è costituito da un elenco che contiene i 3000 termini più utilizzati, nella suddetta lingua, secondo la società internazionale *EF Education First (EF)* [42]. A questi si sommano anche i nomi dei *Country regions* e delle rispettive capitali; basati su quelli indicati presso la relativa pagina web dell'Università *Johns Hopkins* [119].

È opportuno chiarire che i dizionari contenenti i termini più utilizzati nelle altre lingue obiettivo di ASFA saranno realizzati nei successivi stadi di sviluppo dell'applicativo.

La descrizione inerente alle metodologie di utilizzo di questi tre distinti insiemi di dati, si colloca nella sezione 3.2.

3.1.2 Codifica e formati e di input e output

Al fine di rendere l'applicativo sviluppato conforme agli standard inerenti a codifiche e formati di input ed output utilizzati all'interno dell'infrastruttura NLP Hub [34], è stato scelto di utilizzare la codifica "*Unicode Transformation Format, 8 bit (UTF-8)*". Questa è gestita dalla classe "*java.nio.charset.StandardCharsets*" mentre per quanto concerne i formati dei file in ingresso ed uscita, il software in questione, richiede in input un documento in formato *TXT* e restituisce in output i due file, contenenti il testo corredato dai match, nei formati *TXT* e *JSON*. Questi incorporano le medesime annotazioni espresse mediante differenti metodologie strutturali le quali sono illustrate, con esempi annessi, nel *punto 10* della sezione 3.2.

3.2 Descrizione particolareggiata dell'approccio: algoritmi e workflow

Questa sezione è adibita alla descrizione puntuale degli algoritmi e workflow dell'applicativo realizzato. L'esposizione, suddivisa in dieci punti, non corrisponde esattamente all'operatività interna del software realizzato ma risulta essere prettamente inerente ai fini illustrativi di questo elaborato. Inoltre, essa, non è comprensiva della trattazione inerente all'integrazione con *NLPHub*; la cui argomentazione è esplicita nelle sezioni 3.4 e 3.5.

1. All'avvio dell'esecuzione da parte dell'utente, la classe "*OrchestratorSpecies*" contenente il metodo "*main*", controlla se all'interno dell'array di Stringhe "*args*" è presente la directory di un file di testo(in formato "*TXT*") il quale verrà utilizzato nei passaggi successivi del programma. In caso contrario viene utilizzato un file di default.
2. Il metodo "*main*" crea l'oggetto "*researchObjectSpecies*" proprio della classe "*ASFAResearchObjectSpecies*". Esso viene inizializzato mediante il metodo "*capture*" inserendo come parametro attuale la directory indicata dall'argomento di input.
3. Il metodo *capture*, a partire dal testo del file in input, genera un array in cui gli specifici elementi sono composti dalle singole parole del testo. Questo viene ottenuto mediante l'ausilio delle espressioni regolari (*regex*), con le quali vengono rimossi anche tutti i caratteri non alfanumerici e gli spazi multipli. Un possibile esempio di questo processo è illustrato dall'esempio seguente:

Testo originale in input:

"We have got *Canis lupus familiaris* in Mozambique."

Array derivante dall'applicazione del metodo *capture*:

[We, have, got, *Canis*, *lupus*, *familiaris*, in, Mozambique]

Esempio della trasformazione del testo originale in input in una lista mediante il metodo "capture".

4. L'oggetto *researchObjectSpecies* viene nuovamente inizializzato mediante il metodo *get*. Questi, a sua volta, si serve del metodo *EfficientSearch-InText* contenuto nella classe *searchParallel*. Tale classe, a partire da dall'array precedentemente realizzato dal metodo *capture* e dal dataset in formato *csv* contenente l'insieme dei taxa in cui sono riportati i Genus seguiti dalle Species non puntate (descritto in dettaglio nella sezione 3.1.1 e successivamente indicato come *dataset 1*) esegue una ricerca mirata all'individuazione delle occorrenze che sono contenute in entrambi i parametri attuali del metodo. Questo processo è progettato per utilizzare a pieno le capacità computazionali della macchina ospitante al fine di massimizzare le performance in termini tempi di esecuzione. Esso, le cui prestazioni sono riportate nella sezione 5, suddivide il task di ricerca in otto processi i quali vengono eseguiti simultaneamente mediante l'utilizzo di tutti i thread disponibili nel dispositivo incaricato all'esecuzione. Al termine di questo processo viene generata in output una lista composta da booleani. La dimensione di quest'ultima è pari a quella dell'array in input e contiene i valori di *True* nelle rispettive posizioni delle entità che sono state ritrovate all'interno del *dataset 1*, *False* altrimenti. Un esempio di possibile input ed output di questo algoritmo è illustrato nell'esempio seguente:

Input:	We	have	got	Canis	lupus	familiaris	in	Mozambique
Output:	false	false	false	true	true	true	false	false

Esempio di array di input ed output nel processo descritto dal punto 4.

5. Mediante il metodo "*enrich*" l'oggetto *researchObjectSpecies* viene ulteriormente raffinato. La prima fase di questo procedimento consiste nello scorrere, attraverso un ciclo *for*, l'array contenente le parole della stringa di input sincronicamente a quello dei match. In quest'ultimo ogni qualvolta si verifica la presenza di un valore corrispondente a *True* vengono svolti i seguenti controlli:

- (a) Viene controllato se l'entità in questione è preceduta da un'altra (un ulteriore match positivo) e se essa risulta essere formata da una lettera maiuscola puntata. Questo è necessario per verificare e catturare l'eventuale presenza di entità espresse in forma puntata, come ad esempio "*L. chalumnae*"
- (b) Viene verificato se l'entità in questione è preceduta da un'altra (un ulteriore match positivo) e se essa risulta essere una parola avente iniziale maiuscola. Questo per verificare e catturare l'eventuale presenza di entità espresse in forma estesa, come ad esempio "*Latimeria chalumnae*"

Qualora si verifichi una di queste due eventualità, entrambe le entità in questione vengono aggiunte ad una stringa chiamata "*annotationseq*" mentre, in caso contrario viene aggiunta (sempre alla medesima lista) l'entità singola corrispondente al match. Questo perché, data la metodologia di ricerca e la struttura della lista di supporto contenente i taxa, già utilizzata nel punto 4, si presuppone che l'entità in questione sia composta da una singola occorrenza di un *Genus*. Al termine di ogni step del ciclo, la stringa *annotationseq* viene inserita all'interno della lista "*allAnnotationsequences*" e inizializzata nuovamente. Questa procedura è illustrata, mediante l'utilizzo di pseudo-codice, nell' algoritmo 1.

```

for elementi nella lista contenente il testo in input do
  if valore "True" nella lista dei match con il "dizionario 1" then
    if elemento che precede è una lettera maiuscola puntata then
      | Aggiungere entrambi alla stringa annotationseq
    end
    if elemento che precede è una parola avente lettera maiuscola
      then
      | aggiungere entrambi alla stringa annotationseq
    end
    else
      | aggiungere il match corrente alla stringa annotationseq
    end
  end
  aggiungere la stringa annotationseq alla lista
  allAnnotationsequences e inizializzare nuovamente
  annotationseq
end

```

Algoritmo 1: Pseudo codice dell'algoritmo illustrato nel punto 5

6. Mediante un nuovo ciclo vengono controllati singolarmente tutti gli elementi della lista *allAnnotationsequences* al fine di verificare le seguenti condizioni:
 - (a) Se è composto da almeno due parole aventi, la prima, iniziale maiuscola e la seconda minuscola, esso viene salvato in una lista denominata *ArrayListGenusEpithet* la quale verrà nuovamente utilizzata nelle operazioni successive (descritte nel punto 7).
 - (b) Se al contrario è composto da almeno due parole, aventi entrambe l'iniziale maiuscola, esso viene scartato.
 - (c) Se l'elemento in questione è composto da una sola parola, avente iniziale maiuscola, questi viene ricercato all'interno della lista contenente i termini inglesi più utilizzati. Se il riscontro è positivo l'elemento viene scartato, altrimenti inserito nella lista *checkedallAnnotationsequences*.
 - (d) Nell'eventualità che l'elemento in questione sia formato da una sola parola avente iniziale minuscola; esso viene scartato.

Le operazioni appena descritte sono illustrate mediante l'utilizzo dello pseudo-codice nella seguente rappresentazione (*Algoritmo 2*).

```
for elementi in allAnnotationsequences do
  if elemento composto da almeno due parole then
    if elemento composto da una parola con iniziale maiuscola e la
      successiva con iniziale minuscola then
      | aggiungere alla lista "ArrayListGenusEpithet"
    end
    if Le prime due parole aventi entrambe iniziale maiuscola then
    | eliminare elemento
    end
  end
  if elemento composto da una singola parola con iniziale minuscola
  then
  | eliminare elemento
  end
  if elemento composto da una singola parola avente iniziale maiuscola
  then
  | if Se presente all'interno dell'insieme delle parole inglesi then
  | | eliminare elemento
  | end
  | else
  | | aggiungere alla lista "checkedallAnnotationsequences"
  | end
  end
end
```

Algoritmo 2: Pseudo codice dell'algoritmo illustrato nel punto 6

7. Le occorrenze della lista *ArrayListGenusEpithet*, vengono singolarmente ricercate mediante l'utilizzo della funzione *searchParallel* presente nella classe *EfficientSearchInText* all'interno del *dataset 2*. Per ogni riscontro positivo l'entità in questione viene aggiunta alla lista *checkedallAnnotationsequences*. In caso contrario ciascuna delle relative entità scartate, viene sezionata conservando unicamente la prima delle parole dalle quali è composta e per ognuna di esse vengono verificate le seguenti condizioni:

- (a) Che abbia l'iniziale maiuscola e che sia formata da più di due lettere.
- (b) Che unicamente la prima lettera sia maiuscola

- (c) Che non sia contenuta all'interno della lista contenente le parole inglesi più comuni.

Nel caso in cui l'entità in questione sia conforme a tali condizioni essa viene aggiunta alla lista *checkedallAnnotationsequences*.

Il termine di queste operazioni coincide con la fase finale del processo di *name matching*. L'algoritmo 3 è esplicativo dei costrutti appena trattati.

```
for elemento in ArrayGenusEpithet do  
  if elemento presente anche nel dataset 2 then  
    | aggiungere alla lista checkedallAnnotationsequences  
  end  
  else  
    | rimuovere tutte le parole successive alla prima  
    if parola composta da più di 2 caratteri then  
      | if parola avente iniziale maiuscola then  
        | | if altri caratteri in minuscolo then  
          | | | if non presente nella lista di termini inglesi then  
            | | | | aggiungere alla lista checkedallAnnotationsequences.  
            | | | end  
          | | end  
        | end  
      | end  
    | end  
  end  
end
```

Algoritmo 3: codice illustrato nel punto 7

8. Una volta ultimata la fase di matching le operazioni che seguono sono inerenti alla generazione del file di output contenenti il testo corredato dalle annotazioni fin qui estrapolate secondo le metodologie descritte nella relativa sezione (3.1.2).

Il file in input viene inserito all'interno di una variabile di tipo *String* denominata "*testooriginale*". Da questa vengono eliminati i "*linebreaks*" ovvero i tag che tipicamente indicano agli editor di testo di riportare ciò che si pospone a ciascuno di essi in una nuova linea. Inoltre, data la scelta

dell'utilizzo delle parentesi quadre come delimitatore delle named entities (per quanto concerne uno dei due formati di uscita), queste debbono essere necessariamente rimosse al fine di non generare ambiguità. A partire dalla lista, *checkedallAnnotationsequences*, ne viene generata una nuova, denominata "*annot*", contenente le medesime entità ma ordinate secondo la lunghezza in caratteri in modo ascendente. In seguito, mediante un ciclo for, per ogni entità presente in tale lista vengono svolte le seguenti operazioni:

- (a) Attraverso l'utilizzo delle espressioni regolari (*regex*) viene definito un pattern personalizzato che mira alla ricerca di ognuna di esse all'interno della variabile *testooriginale*.
- (b) Per ogni riscontro positivo, mediante un insieme di condizioni comprensive dell'utilizzo di altre espressioni regolari, i match in questione vengono corredati da parentesi quadre.

Terminata tale fase, il testo contenuto all'interno della variabile *testooriginale* presenta esplicitamente le entità identificate durante la fase di matching mediante l'utilizzo del carattere delle parentesi quadre di apertura e chiusura agli estremi di ogni match, come illustrato dall'esempio seguente:

They are generally known as niggerheads, bottle washers or pappus grass. [Nardus] is a Genus of plants belonging to the grass family, containing the single Species [[Nardus] stricta], known as matgrass.

Esempio del testo in output delle operazioni descritte fin qui nel punto 8.

È possibile che si possano verificare casi in cui le parentesi utilizzate si manifestino in forma annidata, come nel caso descritto dall'esempio precedente per l'entità "*Nardus stricta*". Questo si verifica perché durante la fase di identificazione (descritta nei punti 4, 5, 6 e 7) il match è risultato positivo sia per l'entità di "*Nardus*" che per quella di "*Nardus stricta*".

Per ovviare a ciò è stato realizzato uno specifico metodo denominato "*cleanup-Nested*" anch'esso presente all'interno della classe *ASFAResearchObject*

Species. Questi, mediante il conteggio delle occorrenze dei caratteri corrispondenti alle parentesi quadre, produce in output una stringa corrispondente non comprensiva degli annidamenti di parentesi. Un esempio di tale applicazione è illustrato nell'esempio seguente:

Input: *"They are generally known as niggerheads, bottle washers or pappus grass. [Nardus] is a Genus of plants belonging to the grass family, containing the single Species [[[Nardus] stricta], known as matgrass."*

Output: *"They are generally known as niggerheads, bottle washers or pappus grass. [Nardus] is a Genus of plants belonging to the grass family, containing the single Species [Nardus stricta], known as matgrass."*

Esempio dell'applicazione del metodo "cleanupNested"

9. Terminate le operazioni descritte nel punto 8, la stringa *testooriginale* viene analizzata nuovamente carattere per carattere mediante un ciclo *while*. Nel corso di questo processo vengono salvate le posizioni di tutte le parentesi di apertura e chiusura all'interno di una variabile di tipologia *StringBuilder* denominata *"jsonIndex"*. Quest'ultima, utilizzata successivamente (punto 10) per la creazione del file in output in formato *JSON*, è costruita mediante la dicitura espressa nell'esempio seguente. Questo, già parzialmente discusso nella sezione 3.1.2, è stato scelto sulla base degli standard di annotazione precedentemente definiti all'interno dei sistemi di NLP Hub, descritti in sezione 3.5).

"Taxon": [{"indices": [21,40], "indices": [99,120]}

Esempio del formato di annotazione delle entità all'interno della stringa "jsonIndex".

10. Le annotazioni contenute nella stringa *jsonIndex* vengono immagazzinate all'interno di una variabile di tipo *LinkedHashMap* appositamente creata ovvero "*annotationsjson*". Quest'ultima, costituita da due elementi di tipo stringa, incorpora nel primo dei due il testo privo delle annotazioni e nel secondo gli indici corrispondenti ai match effettuati mentre, in un'altra variabile di tipologia *LinkedHashMap* denominata "*annotationstext*", viene salvato il testo comprensivo delle parentesi quadre contenuto nella variabile *testooriginale*.

Una volta ultimate queste operazioni l'oggetto *researchObjectSpecies* viene inizializzato mediante il metodo "*materializeText*" il quale, mediante la classe *PrintWriter* permette di restituire in output un file in formato *TXT* contenente il testo in all'interno della variabile *annotationstext*. Una dimostrazione del possibile contenuto di tale file di output è indicato nell'esempio immediatamente successivo:

Observations include [Latimeria chalumnae] in deep waters of the coast of south eastern Africa while [Latimeria menadoensis] is known from similar habitats in Indonesian waters.

Esempio dell'output del programma in formato TXT.

Tramite il metodo "*materializeJSON*" l'oggetto *researchObjectSpecies* viene inizializzato nuovamente. Viene così generato il file in output in formato *JSON* nel quale vengono salvate le annotazioni presenti nella variabile *annotationsjson*, così come indicato nell'esempio seguente:

```
{"text": "Observations include Latimeria chalumnae in deep waters of the coast of south eastern Africa while Latimeria menadoensis is known from similar habitats in Indonesian waters."}
```

```
"entities": { "Taxon": [{"indices": [21,40]},{ "indices": [99,120]}] }
```

Esempio dell'output del programma in formato JSON.

Questa operazione coincide con il termine dell'esecuzione del programma realizzato.

3.3 Identificazione dei Thesauri

Come già accennato nella sezione 3.1, l'applicativo sviluppato è stato concepito anche per l'identificazione dei Thesauri. Per questo, durante le fasi di progettazione dello stesso, è stato deciso di realizzare una architettura valida per l'identificazione di entrambi gli insiemi di named entities. Concretamente, durante le prime fasi di sviluppo la classe *ASFAResearchObjectSpecies*, chiamata semplicemente "*ASFAResearchObject*", incorporava anche la funzionalità di identificazione delle entità dei Thesauri, azionabile mediante una specifica invocazione del metodo *enrich* attraverso *Orchestrator*. Tuttavia, con il progredire dello sviluppo del software, si è deciso di suddividere la classe *ASFAResearchObject* in due per poter personalizzare al meglio le operazioni di identificazione. Sono così state concepite le classi di *ASFAResearchObjectSpecies* e "*ASFAResearchObjectThesaurus*"; quest'ultima adibita al task di NER delle entità dei Thesauri. Tuttavia, data la totale mancanza di dati necessari per la fase di testing e valutazione, l'applicativo in questione è da considerarsi come una versione *pre-alpha*.

Il funzionamento interno risulta essere in parte analogo a quanto già affermato per la componente adibita al riconoscimento dei taxa mentre gli elementi di divergenza sono rappresentati, in parte minore, dai metodi *get* ed *enrich* che risultano modificati solo in funzione dell'applicazione dei dizionari di supporto i quali costituiscono, al contrario, la differenza più significativa. Questi sono stati ricavati dal dataset "*ASFA*" [46] (2021) realizzato dalla *Organizzazione delle Nazioni Unite per l'alimentazione e l'agricoltura* (FAO) che è stato reso disponibile tra le annotazioni selezionabili dall'interfaccia Web². Sulla base di questo dataset ne sono stati realizzati altri due: il primo, denominato "*thesaurus.csv*", contiene tutte le entry presenti in lingua inglese senza ripetizioni; così come indicato dall'esempio sottostante:

²Disponibile all'indirizzo <http://nlp.d4science.org/asfa/>

methods ways of doing
hybridization
hybridizing
inbreeding
selection
selective breeding
natural selection
analytical methods
analytical techniques

Mentre il secondo dei due, qualificato con il nome di "*thesaurus_monogram.csv*" incorpora i medesimi dati con la differenza che ciascuna entry è composta da una sola parola e le parole composte sono distribuite su più righe in modo da realizzare un insieme di dati in cui è presente una singola parola per ogni record. Anche in questo caso sono state rimosse le ripetizioni. Una porzione di questo dataset è riportata nell'esempio seguente:

methods
ways
of
doing
hybridization
hybridizing
inbreeding
selection
selective
breeding

A differenza dell'applicativo adibito al riconoscimento dei taxa, questo NER è quindi progettato per l'individuazione dei Thesauri. L'estrazione dell'informazione avviene mediante l'opportuna combinazione e l'uso degli insiemi di sopra mediante i metodi *get* ed *enrich* secondo gli algoritmi usati per i taxa. È interessante notare che l'algoritmo prescinde dal dominio del Thesaurus e può essere quindi applicato a qualunque lista di parole.

3.4 Descrizione del sistema di cloud computing sul quale gira l'algoritmo

Come già menzionato nella sezione 3.1, l'applicativo realizzato è stato integrato all'interno di un sistema distribuito. Quest'ultimo, descritto in dettaglio nella sezione successiva 3.5, si serve di una specifica *e-Infrastructure*¹ denominata *D4Science* la quale, a sua volta, include *DataMiner* [32, 35] una piattaforma basata sul paradigma del *cloud computing*. Essa è in grado di supportare l'esecuzione in parallelo di circa 400 processi lanciati mediante lo standard *Wi-Fi Protected Setup* (WPS) il quale, come illustrato in *Figura 6*, permette l'interazione da parte di un ammontare variabile di client utilizzati da software di terze parti. Inoltre, è in grado di eseguire in parallelo i processi che ospita grazie all'utilizzo di risorse computazionali organizzate a cluster e alla possibilità di sfruttare core virtuali. Nello specifico l'architettura in questione (che si colloca presso il CNR e "*Italian Academic and Research Network*" (GARR)) è composta da 15 unità computazionali le quali utilizzano il sistema operativo *Ubuntu* in versione 16.04.4 LTS x86 64; ciascuno comprendente le seguenti risorse hardware:

- 16 core virtuali.
- 32GB di memoria RAM.
- 100GB di spazio di archiviazione.

Le richieste di calcolo, da parte dei client, vengono gestite mediante un software di bilanciamento. Esso impone un massimo di quattro esecuzioni in contemporanea per dispositivo ed assegna, per ognuno di essi, un documento contenente la lista delle istanze prese in carico con annessa lista di attesa per le successive. Ciò permette a *DataMiner* di poter gestire una grande quantità di dati monitorando ogni operazione eseguita. Mediante lo standard ontologico Prov-O XML vengono salvate le informazioni, in formato meta-testuale, riguardanti i dati in input ed output e inerenti ai parametri utilizzati. Per quanto concerne la *user experience*, per ogni processo è disponibile una specifica interfaccia web. Queste, generate automaticamente da *DataMiner* mediante l'interpretazione delle descrizioni WPS,

¹Con il termine "*e-Infrastructure*" si fa riferimento ad un sistema informatico distribuito composto da risorse hardware e software. Esso è adibito al supporto di un certo processo scientifico che pone in collaborazione più studiosi mediante strumenti appositamente sviluppati. [34] [63]

permettono all'utente di svolgere tutte le operazioni di elaborazione dati e condivisione del lavoro con altri studiosi. Inoltre è inclusa la possibilità di realizzare, condividere, ed utilizzare nuovi algoritmi sviluppati mediante diversi linguaggi di programmazione e inerenti ai più disparati domini applicativi, come ad esempio:

- Biologia computazionale [36].
- Realtà virtuale [33].
- Ricerche in grandi database [21].

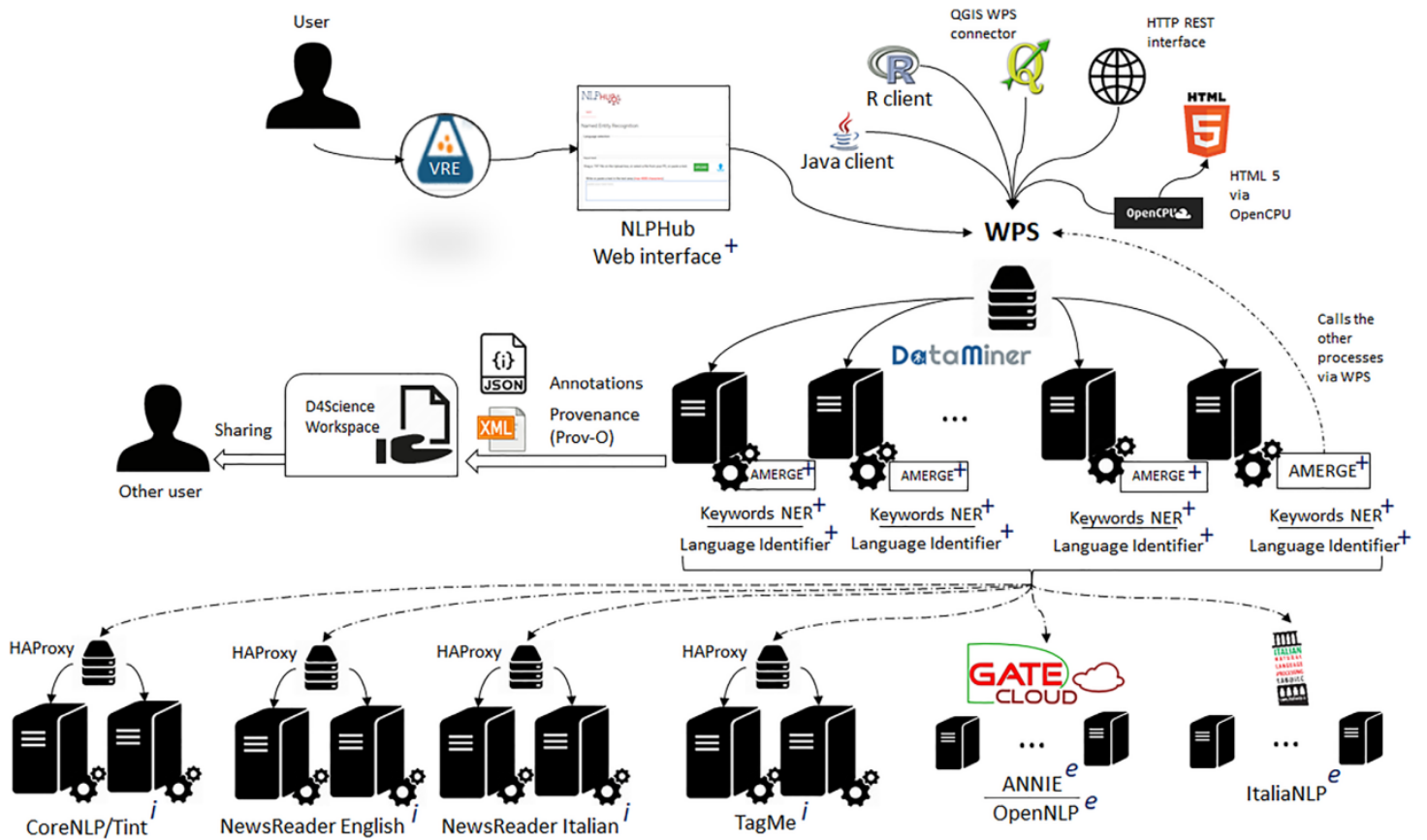


Fig. 6: Diagramma dell'architettura di NLP Hub. Immagine tratta da [34].

3.5 Descrizione del sistema NLPHub

"NLPHub, a distributed system that orchestrates and combines several state-of-the-art text mining services that recognize spatiotemporal events, keywords, and a large set of named entities. NLPHub adopts an Open Science approach, which fosters the reproducibility, repeatability, and reusability of methods and results, by using an e-Infrastructure supporting data-intensive Science." [34]

NLPHub è un sistema adibito a operazioni di text mining basato su una "e-Infrastructure". Questa, già menzionata nella sezione precedente (3.4) prende il nome di *D4Science* che include tra le sue funzionalità e servizi la piattaforma di cloud computing *DataMiner*. Questi ultimi due applicativi rendono conformi i vari utilizzi di NLPHub al paradigma *Open Science*² (approfondito nella sezione successiva 3.6).

NLPHub è quindi un applicativo adibito alla coordinazione di vari servizi di text mining all'interno di un unico *workspace* garantendo le funzionalità di interconnessione tra gli stessi. Nello specifico, questi sono adibiti al NER all'interno di porzioni di testo di qualsiasi dominio, scritti nelle lingue di:

- italiano
- spagnolo,
- tedesco,
- francese,
- inglese.

Le classi ontologiche utilizzate all'interno di NLPHub, adibite al riconoscimento delle named entities, sono per buona parte quelle supportate dal software *Stanford CoreNLP*. Esse riportate in *Figura 7*, risultano di immediata comprensione ad eccezione di quelle elencate in seguito le quali necessitano delle relative delucidazioni:

²Paradigma che impone scelte metodologie basate sulla regola delle tre "R" . Queste ultime, ovvero: riproducibilità, ripetibilità e riusabilità; sono inerenti all'applicazione del metodo scientifico [34, 60, 18].

- *Geopolitical entity*:: Una struttura politica associata ad un'area geografica.
- *Misc*: un concetto di vario tipo che non risulta associabile alle altre classi.
- *Ordinal*: entità che fa riferimento ad una posizione all'interno di una lista ordinata.
- *Token*: Una componente costituita da una sequenza di caratteri portatrice di significato.
- *Sentence*: Un periodo completo all'interno del discorso composto da una serie di unità (*Token*).
- *Event*: Il verificarsi di un certo fenomeno in un determinato luogo e tempo rappresentato da sostantivi, verbi o frasi.
- *Keyword*: Un elemento di primaria importanza per la comprensione di un enunciato.

Language	Service	Annotations																			
		Person	Location	Geopolitical	Organization	Date	Money	Percentage	Address	Misc	Keyword	Event	Number	Ordinal	Time	Duration	URL	Emoticon	Hashtag	Token	Sentence
English	CoreNLP	✓	✓		✓	✓	✓	✓		✓		✓	✓	✓	✓						✓
	GATE Cloud–ANNIE	✓	✓		✓	✓	✓	✓	✓							✓	✓	✓	✓	✓	✓
	GATE Cloud–ANNIE Measurements	✓	✓		✓	✓	✓	✓	✓											✓	✓
	OpenNLP	✓	✓		✓	✓	✓	✓						✓						✓	✓
	NewsReader											✓									
	TagMe										✓										
	Keywords NER										✓										
Italian	CoreNLP–Tint	✓	✓		✓																✓
	ItaliaNLP	✓	✓	✓	✓																
	NewsReader										✓										
	TagMe										✓										
	Keywords NER										✓										
German	CoreNLP	✓	✓		✓					✓											✓
	GATE Cloud–ANNIE	✓	✓		✓	✓	✓	✓	✓											✓	✓
	TagMe										✓										
	Keywords NER										✓										
French	CoreNLP	✓	✓			✓						✓									✓
	GATE Cloud–ANNIE	✓	✓		✓											✓	✓	✓	✓	✓	✓
	Keywords NER										✓										
Spanish	CoreNLP	✓	✓		✓	✓				✓		✓									✓
	Keywords NER										✓										

Fig. 7: Categorie di named entity disponibili per ciascun servizio all'interno di NLPHub.

Immagine tratta da Coro et al. (2020) [34].

Per quanto riguarda le entità di tipologia *Geopolitical entity* esse sono identificate unicamente dal software *"ItaliaNLP"* mentre gli altri NER, disponibili in NLPHub, classificano queste entità arbitrariamente tra le categorie di *"Organizzazione"* e *"Luogo"*.

```
1 {"text": "input text",
2   "NER1": {
3     "annotations": {
4       "annotation1": [
5         {"indexes": [i1, i2]},
6         {"indexes": [i3, i4]},
7         ...,
8         {"indexes": [ig, ig+1]},
9         ...,
10        "annotationk": [
11          {"indexes": [i1, i2]},
12          {"indexes": [i3, i4]},
13          ...,
14          {"indexes": [it, it+1]},
15        ],
16        ...,
17        "NERm": {
18          "annotations": {
19            "annotation1": [
20              {"indexes": [i1, i2]},
21              {"indexes": [i3, i4]},
22              ...,
23              {"indexes": [ig, ig+1]},
24              ...,
25              "annotationd": [
26                {"indexes": [i1, i2]},
27                {"indexes": [i3, i4]},
28                ...,
29                {"indexes": [if, if+1]}
30              ]
31            }
32          }
33        }
34      }
35    }
```

Fig. 8: Formato di output utilizzato all'interno di NLPHub.
Immagine tratta da Coro et al. (2020) [34].

Mediante un algoritmo di tipologia "*wrapping*" ciascuno dei relativi output di questi algoritmi viene reso mediante uno specifico formato. Esso, illustrato in *Figura 8* risulta conforme alle modalità dell'applicativo realizzato in quanto conforme a tali specifiche. I singoli algoritmi, disponibili all'interno di NLPHub, sono esplicitati nella seguente lista corredati da relativa descrizione:

- CoreNLP [84]: sviluppato dall'università di Stanford, *CoreNLP* si configura come uno strumento open-source contenente, oltre che a NER, diversi applicativi per le procedure di text mining quali: Part of speech tagging, sentiment analysis e morphological parsing. Attraverso NLPHub è possibile processare testi in lingua italiana mediante l'impiego del servizio "*Tint*"
- GATE Cloud [116]: realizzato per l'impiego in ambiti enterprise, offre un servizio a pagamento concepito per il processing di dati di grandi dimensioni (*big data*). L'applicativo ereditario di questo software "*ANNIE*" è adibito al task di NER inerente alla categorie illustrate nella *Figura 7*.
- OpenNLP [44]: concepito mediante il paradigma Open Source, include, oltre che a NER, diversi strumenti di NLP, quali: language detection, part-of-speech tagging, morphological parsing e tokenization. Esso, il cui nome completo è *Apache OpenNLP library*, si serve principalmente di modelli basati sul machine learning.
- ItaliaNLP [38]: progettato mediante la combinazione di algoritmi basati su regole e modelli di machine learning, si configura come un servizio *free-to-use* tramite il quale è possibile svariati task inerenti al NLP, quali:
 - Clustering
 - Sentiment analysis
 - Tokenization
 - Lemmatization
 - Part-of-speech tagging
 - NER
 - Morphological parsing

Sviluppato dall'*Istituto di Linguistica Computazionale of the National Research Council of Italy (ILC-CNR)* è adibito principalmente al supporto delle applicazioni basate su e-learning.

- NewsReader [120]: realizzato da *NewsReader European project* nel 2014, è concepito per identificare informazioni accessorie (i partecipanti e/o vincoli di tempo e luogo) inerenti alle named entity di nomi, verbi e frasi all'interno del testo. Esso è stato addestrato tramite tesi di dominio giornalistico annotati mediante il paradigma del machine learning.
- TagMe [20]: servizio adibito all'identificazione di entità associabili a pagine all'interno di *Wikipedia*. È impiegato solitamente per i task di contestualizzazione e comprensione del testo. Ampiamente utilizzato all'interno di NLPHub è adoperato anche per le operazioni di disambiguazione dei vari contesti linguistici.

Inoltre, è presente un algoritmo di tipologia "*orchestrator*", denominato "*AMERGE*". Esso è in grado di armonizzare gli output dei singoli applicativi, poc'anzi descritti, generandone un unico contenente le varie annotazioni combinate secondo quelle che risultano essere le opzioni di etichettatura preferibili dal punto di vista decisionale di questo software [34].

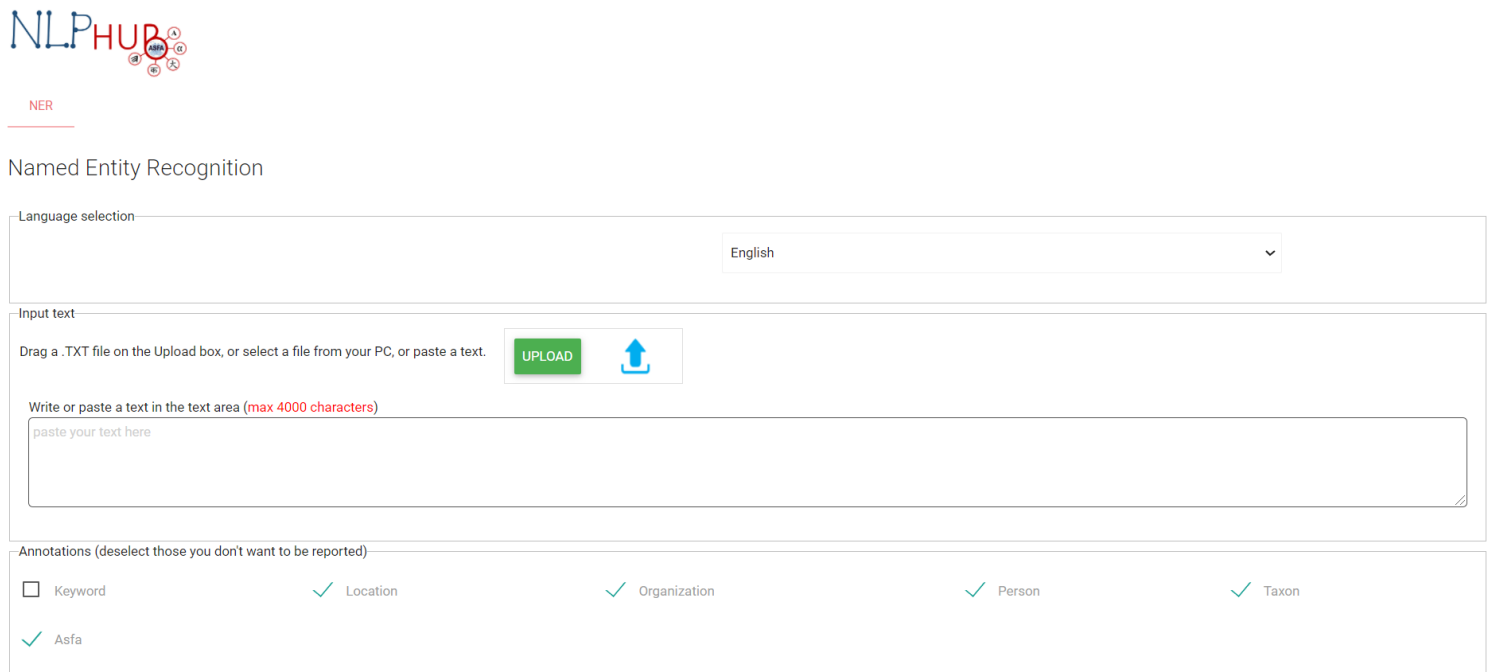
Per quanto concerne la gestione della lingua di provenienza in input, NLPHub, oltre a permetterne la selezione da parte dell'utente, incorpora uno specifico strumento adibito al riconoscimento automatico di essa. Questo è stato realizzato sulla base dell'assunzione che il software TreeTagger (che opera il Part-of-Speech Tagging), se utilizzato con testi provenienti da una lingua diversa da quella per il quale è stato impostato, tenderà a produrre in output una maggior quantità di entità riconosciute come nomi e parole non declinate. Questa caratteristica ha permesso lo sviluppo del software in questione [34].

L'applicativo sviluppato in questa tesi si concretizza all'interno di NLPHub come un'ulteriore categoria di named entity. Questa, denominata "*Taxon*", liberamente utilizzabile sia singolarmente che in combinazione con gli altri algoritmi, è gestita

in modo corretto da *AMERGE* il quale gestisce regolarmente il processo di output una volta che questo è stato reso conforme alle specifiche di funzionamento di NLPHub.

In aggiunta, è possibile utilizzare il software in questione mediante una interfaccia web appositamente realizzata³. L'integrazione con D4Science, nell'ambito dell'NLP ha reso disponibile il software realizzato anche come servizio WPS⁴.

Essa è illustrata dalla *Figura 9*.



The image shows a web interface for Named Entity Recognition (NER) on the NLP Hub. At the top left is the NLP HUB logo. Below it, the text 'NER' is underlined. The main heading is 'Named Entity Recognition'. The interface is divided into three main sections: 1. 'Language selection' with a dropdown menu currently showing 'English'. 2. 'Input text' which includes a prompt 'Drag a .TXT file on the Upload box, or select a file from your PC, or paste a text.' with an 'UPLOAD' button and an upload icon, and a text area with the placeholder 'paste your text here' and a note '(max 4000 characters)'. 3. 'Annotations (deselect those you don't want to be reported)' which contains a list of categories with checkboxes: 'Keyword' (unchecked), 'Asfa' (checked), 'Location' (checked), 'Organization' (checked), 'Person' (checked), and 'Taxon' (checked).

Fig. 9: Pagina web attraverso la quale è possibile utilizzare l'applicativo sviluppato [34].

³Disponibile all'indirizzo <http://nlp.d4science.org/asfa/>

⁴Disponibile all'indirizzo https://services.d4science.org/group/rprototypinglab/data-miner?OperatorId=org.gcube.dataanalysis.wps.statisticalmanager.synchserver.mappedclasses.transducerers.ASFA_SPECIES_NER dopo opportuna registrazione a D4Science

3.6 Open Science, ripetibilità, riproducibilità, riuso

Come già menzionato nella sezione precedente 3.5, il sistema distribuito informatico NLPHub si configura all'interno di una e-Infrastruttura progettata per condurre le operazioni di text mining secondo il paradigma dell'Open Science (OS). Quest'ultimo è concepito per indirizzare i costrutti metodologici e tecnologici in direzione delle tre "R" inerenti al metodo scientifico, ovvero: riproducibilità, ripetibilità e riusabilità [34, 60, 18]. L'OS incoraggia l'utilizzo di piattaforme collaborative adibite alla gestione e condivisione di grandi quantità di dati. Tali strumenti, concretizzati come web-services, comportano benefici dimostrati in applicazioni recenti [79]. La conformità al paradigma in questione impone l'utilizzo di specifici standard inerenti alla progettazione dei singoli applicativi rendendo possibile l'interoperabilità tra di essi, fattore che permette il rispetto della caratteristica della riusabilità propria del paradigma OS. Questo rende conforme l'utilizzo dei vari software anche in altri domini applicativi. Tuttavia, tale paradigma non è spesso rispettato nella realizzazione di programmi per l'NLP [23, 11]. Al contrario essa rappresenta una delle caratteristiche fondanti di NLPHub il quale, utilizzando una piattaforma di cloud computing, è in grado di interconnettere, orchestrare e combinare gli output di più servizi di text mining dislocati all'interno dei relativi provider ed e-infrastructures.

4 Sistema di confronto utilizzato

4.1 Descrizione generale del sistema di confronto utilizzato

L'applicativo sviluppato è stato direttamente confrontato con *BotanicalNER* realizzato da Meraner (2019) [87]. Quest'ultimo, già introdotto nella *Sezione 2.4*, è descritto nella totalità delle relative fasi realizzative nelle sezioni successive di questo capitolo ricalcando la suddivisione delle argomentazioni direttamente dalla documentazione redatta dalla stessa autrice di tale studio, anche a partire da uno scambio epistolare con la stessa.

Il software NER realizzato nella nostra indagine differisce da questo lavoro in quanto quest'ultimo risulta essere adibito all'identificazione delle named entities appartenenti all'insieme dei nomi scientifici e vernacolari propri della branca della botanica riportati in lingua inglese o tedesca. Ciononostante esso rappresenta un sistema di confronto valido poiché comprensivo della funzione di NER per i nomi scientifici.

In *Figura 10* sono rese esplicite le fasi realizzative del suddetto progetto corredate dai rispettivi metodi, strumenti e risorse mediante i quali esso è stato sviluppato.

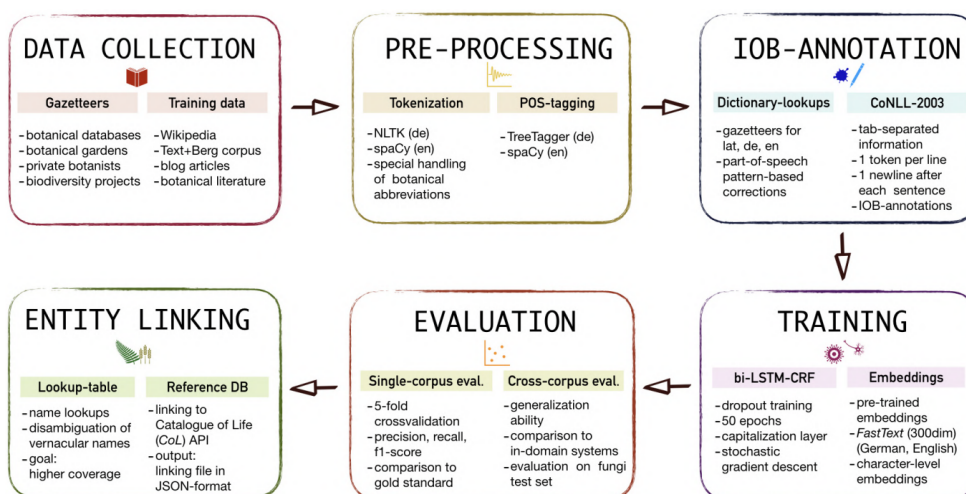


Fig. 10: Workflow del progetto realizzato da Meraner (2019) [87].

4.2 Data Collection

Una delle scelte fondamentali adottate durante lo sviluppo di BotanicalNER è rappresentata dalla decisione di suddividere i file da utilizzare per l'addestramento del modello statistico in due parti: una per l'inglese e l'altra per il tedesco includendo in entrambe i dati relativi alle entità dei nomi scientifici delle specie (riportati in lingua latina). Da questo segue che la realizzazione di tale applicativo è stata svolta indipendentemente per ognuna di esse. Questa scelta è stata dettata dal fatto che il modello utilizzato, ovvero *LSTM-CRF*, pur supportando l'addestramento in più lingue, non risulta essere in grado di poter generare un singolo modello adibito all'utilizzo multilingua. Tale fattore costituisce di per sé un impedimento al riutilizzo del modello in questione poiché, come già trattato nella sezione 3.1, uno dei requisiti fondamentali del progetto realizzato nel nostro lavoro è rappresentato dalla possibile indipendenza dalla lingua in input. Nello studio di Meraner (2019) [87] è stato utilizzato il modello *LSTM-CRF* in totale coerenza con gli obiettivi del rispettivo progetto nel quale è prevista l'identificazione anche dei nomi vernacolari delle specie. Questi sono identificabili unicamente mediante un applicativo in grado di interpretare il contesto grammaticale nel quale tali entità sono collocate. Il modello in questione è quindi capace di poter utilizzare queste caratteristiche per identificare le specie e, per la stessa ragione, non risulta essere indipendente dalla lingua in input.

La prima fase realizzativa del lavoro di Meraner è stata inerente alla generazione dell'insieme dei gazetteers da utilizzare come supporto alla creazione dei dati di training. Questi sono stati ottenuti attingendo da varie fonti e reperiti in modo distinto secondo la relativa tipologia di entità. I nomi scientifici sono stati acquisiti mediante l'aggregazione di vari knowledge base disponibili online presso le varie istituzioni del settore; nello specifico:

- Catalogue of Life (CoL) [107]
- The Global Biodiversity Information Facility (GBIF) [52]
- The International Plant Name Index (IPNI) [117]
- Multilingual Multiscript Plant Name Database (MMPND) [100]

Da questi è stata generata una lista contenente una entry per ciascuna linea.

L'insieme delle entità vernacolari è stata realizzata distintamente per le due rispettive lingue obiettivo. Per il tedesco sono state ricavate da un grande insieme di testi (strutturati e non) provenienti dalle fonti più disparate, come ad esempio articoli di botanica digitalizzati o pagine di Wikipedia. In aggiunta a queste, molte delle entità sono state da ottenute da *Info Flora* [12], *Catalogue of Life* e anche da *GBIF*.

Per quanto riguarda le entità vernacolari in lingua inglese, esse sono state ottenute in parte sempre da CoL, e da (MMPND) ma anche da *Germplasm Resources Information Network* (GRIN) [100]. Per incrementare ulteriormente la dimensione delle liste, sono state aggiunte manualmente alcune entry provenienti da articoli scientifici del settore alcuni dei quali sono stati ricavati mediante un processo di digitalizzazione eseguito appositamente per tale scopo [87].

In aggiunta a questa operazione, al fine di migliorare il grado di copertura delle liste, ciascuna di esse è stata arricchita delle possibili abbreviazioni o estensioni delle rispettive entità generate secondo regole opportunamente definite. Un esempio di tale processo è illustrato dalla *Figura 11*.

	Plant name	Translation	Generated variants
1.	<i>Eugenia floccosa</i>	a species of myrtle	E. floccosa
2.	<i>Sauergrasgewächse</i>	sedges	Sauergras-Gewächse, Sauergras-Gewächses, Sauergras-Gewächsen, Sauergras-Gewächs, Sauergrasgewächse, Sauergrasgewächses, Sauergrasgewächsen, Sauergrasgewächs
3.	<i>Vogel-Sternmiere</i>	chickweed	Vogelsternmiere (merged) Sternmiere (split)
4.	<i>Johannisbeere, Schwarze</i>	blackcurrant	Schwarze Johannisbeere (inverted full species name) Johannisbeere (only genus name)

Fig. 11: Esempio del processo di arricchimento delle liste contenenti le named entities. *Immagine tratta da [87]*

In seguito, sono stati generati quattro corpora, ognuno adibito alla rappresentazione di un differente genere testuale, aventi quindi uno stile caratteristico nonché appartenenti ad una specifica branca della letteratura scientifica inerente alla botanica. Questo è stato fatto sia per poter testare singolarmente modelli

addestrati in ognuno di essi ma soprattutto per poterli combinare in un unico corpus al fine di addestrare un modello avente capacità di "*genre-adaptation*". I corpora in questione sono:

- *Wiki Corpus*: realizzato da tutte le pagine di Wikipedia contrassegnate dalla categoria di *piante vascolari*.
- *TB Corpus*: Insieme di testi composti da annuari dello Swiss Alpine Club (ASC). Disponibile anche nelle lingue di inglese e tedesco, è stata utilizzata la versione redatta da Bubenhofer et al. (2015) [75].
- *PlantBlog Corpus*: Costruito appositamente per questo studio, è composto da articoli reperiti da blog e inerenti ad argomenti connessi alla branca della botanica. Questi testi, scritti mediante un linguaggio colloquiale, sono stati selezionati per incrementare la presenza di entità vernacolari.
- *BotLit/S800 Corpus*: Questo insieme di testi differisce nella composizione a seconda della lingua in questione. Per la parte composta dai testi scritti in tedesco è stato utilizzato principalmente il corpus BotLit/S800 (ad accesso privato) con l'aggiunta di porzioni di testo provenienti da Spescha (2009) [9], Höhn-Ochsner (1986) [61] e Bosshard (1978) [25]. Al contrario, per l'inglese è stato utilizzato il già menzionato *Species800*; ovvero il corpus realizzato per lo studio di Pafilis et al. (2013)[93].

Tuttavia, i dati raccolti, seppur sufficientemente rappresentativi sia della totalità dell'insieme dei taxa obiettivo che dei possibili generi testuali di applicazione, risultano essere ottenuti mediante una metodologia frastagliata che rende difficoltose, se non impossibili, le operazioni di riproducibilità di tale operato. È per questa motivazione che nel nostro lavoro sono stati impiegati dati unicamente provenienti dal database GBIF, fattore che permette la totale adempienza al paradigma *Open-Science*.

4.3 Preprocessing

Una volta generato l'insieme dei gazetteers e corpora; nel lavoro di Meraner questi sono stati preprocessati mediante la seguente serie di operazioni definite:

- Il testo all'interno dei corpora è stato tokenizzato utilizzando la libreria *Natural Language Toolkit* (NLTK) [80]
- Per i corpora scritti in lingua tedesca tutte le entità sono state associate al rispettivo lemma e Part of Speech (PoS) mediante il software TreeTagger [111] mentre per la lingua inglese sono state svolte le medesime operazioni utilizzando però le specifiche funzionalità della libreria *spaCy* [62].
- Mediante l'utilizzo di *regex* sono stati corretti manualmente gli errori commessi dall'applicazione *spaCy* in merito al riconoscimento dei nomi scientifici di specie all'interno dei corpora inglesi.
- Per rendere conformi i dati in questione alle fasi di addestramento dei modelli basati su LSTM-CRF, è stato utilizzato il formato "*CoNLL-2003*" sviluppato da Sang et al (2003) [109].

Tali operazioni, seppur corredate dall'utilizzo di *regex* per la riduzione della presenza di errori commessi dall'assegnazione del lemma e delle categorie PoS, non sono tuttavia esenti da altre tipologie di imprecisioni le quali, nella suddetta circostanza in cui i dati di addestramento risultano essere di ridotte dimensioni, rappresentano la causa di un decremento nelle performance del modello statistico che verrà addestrato su tali dati.

4.4 IOB-Annotation

Una volta terminate le operazioni di pre-processing, mediante l'utilizzo del paradigma *dictionary based* i corpora sono stati arricchiti con l'indicazione delle relative entità target. Nell'atto pratico ogni entry contenuta all'interno dei gazetteers è stata ricercata nei dataset scritti nella lingua corrispondente e, ad ogni riscontro positivo, queste sono state marcate utilizzando il sistema IOB (Inside, Outside, Beginning) [102] le cui iniziali indicano:

- **B**: Primo token appartenente ad un taxon o ad un nome vernacolare che può essere seguito da nessuna o più parole etichettate con *I* ma non può essere preceduto da token etichettati con *B*.
- **I**: Token successivo ad un precedente, entrambi che indicano un taxon o un nome vernacolare, il quale può essere posposto ad una di tipologia *B* o *I*
- **O**: Annotazione che indica che il token in questione è un taxon o un nome vernacolare da estrarre.

Un esempio dell'utilizzo di questo schema, all'interno dei corpora impiegati nel rispettivo studio, è indicata dalla *Figura 12*.

TOKEN	LEMMA	POS-TAG	IOB-TAG
Der	die	ART	O
Gefleckte	gefleckt	ADJA	B-de.species
Schierling	<unknown>	NN	I-de.species
Conium	<unknown>	NN	B-lat.species
maculatum	<unknown>	NN	I-lat.species
gehört	gehören	VVFIN	O
mit	mit	APPR	O
dem	die	ART	O
Wasserschierling	<unknown>	NN	B-de.species
zu	zu	APPR	O
den	die	ART	O
Doldenblütlern	<unknown>	NN	B-de.fam

Fig. 12: Esempio della struttura del dataset di addestramento al termine delle operazioni descritte nelle Sezioni 4.3 e 4.4. *Immagine tratta da [87]*

Al termine della realizzazione dei dataset in questione, sono state effettuate delle operazioni di correzione di eventuali errori mediante espressioni regolari regex utilizzate sfruttando la struttura del suddetto schema.

L'impiego di tale standard di annotazione costituisce un punto di forza per il lavoro di Meraner [87]. Esso, non solo rappresenta una metodologia di memorizzazione ampiamente condivisa ma permette, come nel caso in questione, anche di poter correggere le etichette relative ai dati sfruttando le caratteristiche proprie dello schema stesso.

4.5 Creazione del Corpus Gold Standard

Nel suo lavoro, Meraner ha realizzato un corpus *Gold Standard* per ciascuna delle due lingue obiettivo. Questa operazione è stata svolta a partire da una porzione composta dall'unione sparsa di ciascun corpus generato per le fasi di training. L'intero processo è stato svolto manualmente e sono stati realizzati due corpora (uno per l'inglese e l'altro per il tedesco) formati da 750000 token ciascuno. Questi rappresentano circa il 20% della dimensione dei rispettivi dataset combinati.

La variante in lingua inglese è stata utilizzata nel nostro studio per le fasi di valutazione le quali sono illustrate in dettaglio nelle *Sezioni 5.1, 5.2 e 5.3*.

4.6 Valutazione e trattamento delle annotazioni semi-automatiche

L'applicazione del paradigma *dictionary based* sprovvisto di un insieme di regole complesse adibite alla corretta identificazione delle entità target è la causa di parecchie imprecisioni nel sistema BotanicalNER nell'identificazione delle entità all'interno dei rispettivi corpora. Nella riga "*1st annotation round*" della figura *Figura 13* è illustrato il livello di conformità della porzione di dataset annotato mediante il paradigma *dictionary based* con il rispettivo corpus gold standard mentre, "*2st annotation round*" è inerente ai medesimi valori dopo l'applicazione di varie strategie correttive.

Queste ultime, basate su costrutti semi-automatici e pattern-based, non sono state in grado di gestire le istanze in cui, al fine di poter correggere i dati, era necessaria un'interpretazione del contesto linguistico in cui esse sono collocate. L'erronea etichettatura di questi particolari casi all'interno dei dati di addestramento costruisce un fattore significativamente negativo che rappresenta la causa di un decremento nelle performance del modello statistico che verrà addestrato su di essi.

	German				English			
	A	P	R	F	A	P	R	F
1st annotation round	95.44	89.10	81.36	85.05	97.55	90.95	79.57	84.88
2nd annotation round	98.03	96.84	90.76	93.70	98.59	94.58	89.19	91.80

Fig. 13: Livello di conformità tra il corpus gold standard ed il corrispondente realizzato con il paradigma dictionary based, prima e dopo le correzioni effettuate. *Immagine tratta da [87]*

4.7 Modello basato sull’architettura bi-LSTM-CRF

L’applicativo selezionato per la fase di training è quello di un modello basato sull’architettura *LSTM-CRF* bidirezionale; paradigma introdotto da Lample et al. (2016) [71]. Esso è stato addestrato singolarmente per le rispettive lingue obiettivo utilizzando i relativi dataset in differenti combinazioni.

La *Figura 14*, illustra i valori delle performance, articolati in due fasi, dei quat-

	German								English							
	silver standard				gold standard				silver standard				gold standard			
	A	P	R	F1	A	P	R	F1	A	P	R	F1	A	P	R	F1
1st round																
baseline	98.58	96.52	91.32	93.85	94.13	82.83	79.49	81.12	98.01	87.56	78.31	82.68	97.16	89.99	75.54	82.13
pre_emb	98.92	95.75	94.70	95.22	94.53	84.03	80.86	82.42	98.42	88.94	83.70	86.24	97.27	88.36	78.50	83.14
2nd round																
baseline	98.30	95.38	90.96	93.12	97.61	96.23	88.72	92.32	98.07	88.28	82.35	85.20	98.05	89.91	87.22	88.55
pre_emb	98.81	94.41	95.68	95.04	98.15	96.68	91.79	94.17	98.26	88.6	84.61	86.55	98.17	90.20	88.46	89.32

Fig. 14: Performace dei modelli addestrati sui dataset combinati nell’annotazione del corpus silver standard e gold standard nei due cicli di training. *Immagine tratta da [87]*

tro modelli ottenuti mediante l’addestramento nella totalità dei dati di training. Questa operazione è stata eseguita utilizzando due differenti modelli di apprendimento, entrambi basati sull’architettura bi-LSTM-CRF, ovvero:

- *"baseline"*: modello privo di conoscenza pregressa e addestrato mediante gli iperparametri forniti da Lample et al. [2016] [71].
- *"pre_emb"*: modello già addestrato con informazioni basate sulla proprietà distributiva del linguaggio mediante *Word Embeddings*.

Sempre con i medesimi dati in input, la *Figura 15* è inerente alle performance delle singole classi delle entità target ottenute mediante l'applicazione del modello *pre_emb*.

	German				English				
	No. of entities	P	R	F		No. of entities	P	R	F
de.species:	4202	90.20	92.26	91.22	en.species:	1587	77.82	76.19	77.00
de.fam:	1251	99.52	99.36	99.44	en.fam:	196	95.92	94.95	95.43
lat.species:	1580	96.58	98.58	97.57	lat.species:	846	93.03	92.81	92.92
lat.genus:	991	95.96	95.48	95.72	lat.genus:	854	92.86	85.73	89.15
lat.fam:	1067	99.72	100.00	99.86	lat.fam:	483	98.14	99.16	98.65
lat.subfam:	99	98.99	100.00	99.49	lat.subfam:	41	97.56	76.92	86.02
lat.class:	23	91.30	100.00	95.45	lat.class:	5	80.00	66.67	72.73
lat.order:	46	100.00	97.87	98.92	lat.order:	35	97.14	77.27	86.08
lat.phylum:	4	100.00	80.00	88.89	lat.phylum:	5	80.00	100.00	88.89
total/average:	9263	96.91	95.95	96.28	total/average:	4052	90.27	85.52	87.43

Fig. 15: Performace del modello *"preemb"* nelle singole classi di entities e nelle due rispettive lingue obiettivo. *Immagine tratta da [87]*

I valori inerenti all'identificazione di Genus e Species ottenuti mediante il modello costruito sulla lingua inglese sono stati utilizzati anche per il diretto confronto con il software sviluppato nel nostro lavoro (*Sezione 5.3.2*).

5 Risultati

5.1 Metriche di valutazione dei risultati

Le metriche utilizzate nel nostro studio sono costruite sulla base del paradigma della classificazione binaria che si realizza mediante le seguenti unità:

1. True positive (TP)
2. False positive (FN)
3. False negative (FP)

Queste ultime, il cui utilizzo all'atto pratico è illustrato nella *Sezione 5.2*, sono impiegate per la realizzazione dei seguenti indici:

1. *Precision*
2. *Recall*
3. *F1 score*.

Essi, ampiamente utilizzati sia nella valutazione degli applicativi di NER che, più in generale, nella disciplina della NLP, risultano essere condivisi anche dall'*Organizzazione internazionale per la standardizzazione (ISO)* la quale definisce tali indici in "*ISO 5725-1: 1994: accuracy (trueness and precision) of measurement methods and results-part 1: general principles and definitions*" (1994) [50]

Per quanto concerne l'indice di *precision*, esso è indicativo della probabilità di successo della classe positiva nell'ambito della classificazione binaria. Come indicato dalla formula sottostante viene calcolato tramite la divisione tra l'ammontare dei TP ed il totale di positivi (TP, FP) [1]:

$$\text{Precision} = \frac{TP}{TP+FP}$$

Formula utilizzata, ai fini della valutazione dell'applicativo realizzato, per il calcolo dell'indice di precision

L'indice di recall è orientativo del grado di sensibilità dell'applicativo nell'identificazione della classe positiva. Si realizza mediante il quoziente tra i TP e la somma con gli stessi ed i FN [1]:

$$\text{Recall} = \frac{TP}{TP+FN}$$

Formula utilizzata, ai fini della valutazione dell'applicativo realizzato, per il calcolo dell'indice di recall

Ideata per essere valida anche in condizione in cui sono presenti dati sbilanciati, l'*F1 score* combina i valori di *precision* e *recall* realizzando una media armonica tra questi due valori. La formula è la seguente [6]:

$$\text{F1 score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 * TP}{2 * TP + FP + FN}$$

Formula utilizzata, ai fini della valutazione dell'applicativo realizzato, per il calcolo dell'indice di F1 score

È opportuno chiarire che in questo studio non è indicato l'indice di *Accuracy* poiché non è stato possibile fornire, in un ambito così complesso, una definizione valida per l'unità binaria dei *True negative*. Questo, all'atto pratico, è da attribuire al fatto che data la natura delle entità target composte arbitrariamente da una parola (nel caso dei Genus) e più (nel caso delle Species) non è risultato possibile identificare nelle altre parole del testo un valore concordante con esse in termini quantitativi.

5.2 Approccio alla valutazione dei risultati

Le fasi valutative inerenti ai risultati dell'applicativo realizzato sono state svolte utilizzando due differenti approcci. Il primo è adibito alla valutazione delle performance riguardanti il task di identificazione ed estrazione di tag associabili a documenti mentre il secondo è concepito per il confronto diretto con il lavoro di Meraner (2019) [87]. Ne consegue che essi siano basati su differenti criteri di interpretazione delle unità binarie di classificazione. Queste, in ambo i casi, sono concretizzate dalle parole marcate all'interno del corpus gold standard e dall'output del nostro software ottenuto mediante l'analisi del testo che compone il gold, privo dell'esplicita etichettatura delle entità.

Il primo degli approcci utilizzati per le fasi di valutazione dell'applicativo basa l'identificazione delle categorie binarie tramite la seguente metodologia:

1. **TP**: numero di entità che risultano essere identificate sia all'interno del corpus gold che dal nostro software.
2. **FP**: quantità di occorrenze identificate dal nostro software che non risultano essere marcate all'interno del corpus gold
3. **FN**: numero di entità identificate all'interno del corpus gold ma non identificate dal nostro software.

Il metodo in questione basa il processo di valutazione sulla base del paradigma insiemistico considerando unicamente le entities distinte identificate all'interno del corpus gold standard e dall'output dell'applicativo sviluppato in questo studio. Tale metodologia di valutazione è pensata per stimare le performance del programma nel contesto dell'utilizzo pratico che si concretizza nel task di estrazione ed associazione di tag inerenti a documenti scientifici.

Al contrario, la seconda metodologia di valutazione, concepita per il diretto confronto con l'applicativo sviluppato da Meraner (2019) [87], a partire dal corpus gold, estrae le entità utilizzando le informazioni contenute nello schema BIO inerenti quindi anche alla posizione di queste ultime. Ciò di conseguenza implica che, a differenza del primo approccio, il conteggio comprenda anche le eventuali ripetizioni delle entità all'interno del testo in analisi.

I criteri stabiliti per la suddivisione degli insiemi delle unità binarie di classificazione risulta essere quindi in parte analogo con il precedente:

1. **TP**: numero totale delle entità riconosciute sia all'interno del corpus gold standard che dal nostro software considerando le rispettiva posizione all'interno del testo.
2. **FP**: quantità di entità marcate dal nostro software ma che non sono all'interno del corpus gold considerando le rispettiva posizione all'interno del testo.
3. **FN**: ammontare delle entità marcate nel corpus gold ma non riconosciute dal nostro software considerando le rispettiva posizione all'interno del testo.

5.3 Valutazione dei risultati

5.3.1 Valutazione delle performance nell'estrazione di TAG

La *Tabella 2* mostra i risultati inerenti al primo approccio di valutazione utilizzato. Concepito per il test delle performance nei task di identificazione ed estrazione di tag associabili a documenti, realizza il processo in questione utilizzando le due liste composte dalle rispettive entità distinte identificate all'interno del gold standard (Gold) e nell'output generato dal nostro software a partire dal medesimo testo. Uno dei dati più importanti illustrati in *Tabella 2* è quello inerente al

Specie identificate	783
Genus identificati	866
Specie presenti in Gold ma non in GBIF	153
Numero di Genus/Species nel Gold	1802
Precision	96,8%
Recall	100,0%
F1 score	98,4%
Complementarità del Gold in rapporto a GBIF	8,5%

Tabella 2: Risultati ottenuti dalla valutazione insiemistica

numero di specie che sono presenti all'interno del corpus gold e non in GBIF. Questo valore, espresso anche in percentuale come il grado di complementarità del Gold in rapporto a GBIF, risulta essere sensibilmente grande. Infatti 153 delle specie facenti parte del corpus gold non sono contenute all'interno di GBIF il quale però risulta essere composto da quasi due miliardi di specie [8] contro le sole 1802 del corpus gold.

La precision seppur collocata all'elevata cifra di 96.8%, risulta comunque inferiore all'F1 score (98.4%). La recall (100%) è indicativa del fatto che la totalità delle entità presenti all'interno del corpus gold che sono anche comprese in GBIF sono state tutte correttamente identificate (FN = 0).

In linea generale questi valori evidenziano come il nostro software risulti essere perfettamente in grado di identificare le specie contenute nel gazetteer il quale, composto interamente da *GBIF*, garantisce un ottimo grado di copertura anche nei casi del dominio applicativo dei nomi scientifici delle specie botaniche. Tuttavia, come evidenziato dal valore di Precision di 96.8%, la problematica più evidente è rappresentata dai casi in cui il software ha identificato entità non facenti parte dell'insieme dei nomi scientifici latini andando quindi ad incrementare il numero dei False Positive.

5.3.2 Confronto con BotanicalNER

La seconda metodologia di valutazione, la cui assegnazione delle categorie binarie è illustrata nella Sezione 5.2, è adibita al diretto confronto con *BotanicalNER* [87].

La sua applicazione è preceduta dall'illustrazione delle performance inerenti al caso in cui le entità contenute nel corpus gold siano solo quelle presenti all'interno di GBIF ponendo in relazione i dati risultanti con il caso in cui venga considerata la totalità delle entità. Questo è stato fatto per poter valutare quanto è completo ed esteso il gazetteer (GBIF) che utilizza il nostro programma al fine di individuare quelle entità che si trovano all'interno del gold ma che sfuggono al dominio del nostro sistema.

La *Tabella 3* mostra i dati risultanti da questo confronto:

GENUS	value	SPECIES	value	TOTALE	value	
TP	1037	TP	800	TP	1837	Solo le entità presenti in GBIF
FP	78	FP	0	FP	78	
FN	0	FN	0	FN	0	
Precision	93%	Precision	100%	Precision	95.93%	
Recall	100%	Recall	100%	Recall	100%	
F1 Score	96.3%	F1 Score	100%	F1 Score	97.92%	
TP	1037	TP	800	TP	1837	Tutte le entità
FP	78	FP	0	FP	78	
FN	70	FN	82	FN	152	
Precision	93%	Precision	100%	Precision	95.93%	
Recall	93%	Recall	90.7%	Recall	92.36%	
F1 Score	93%	F1 Score	95.1%	F1 Score	94.11%	

Tabella 3: *Risultati ottenuti confrontando le entities del corpus gold contenute anche all'interno del knowledge based GBIF con la totalità di esse[8].*

I dati inerenti all'identificazione della totalità delle entità mostrano come per le "Species" l'applicativo sviluppato abbia ottenuto il valore più alto di Precision (100%) ma non di Recall (90.07%), che viene influenzato dall'elevato ammontare di False Negative. Riguardo l'identificazione dei "Genus", il riconoscimento risulta avere i valori di Precision, Recall e F1 Score equamente distribuiti e rispettivamente collocati a 93%. Dal conteggio ottenuto mediante la somma delle

rispettive unità binarie di classificazione dei Genus e Species ne risultano valori ugualmente elevati che testimoniano in maniera ancor più evidente la validità del software sviluppato nel nostro studio.

Per quanto concerne i valori ottenuti considerando unicamente le entità presenti all'interno di GBIF, i valori differiscono unicamente nel conteggio dei False Negative dei Genus e per i False Positive riguardanti le Species. Dal confronto tra i dati di questi due insiemi segue che:

- Il numero di False Negative nel riconoscimento dei Genus e Species può essere ridotto mediante l'aggiornamento del gazetteers.
- L'insieme di regole adibite all'identificazione dei Genus è incrementabile mediante la ridefinizione delle stesse con relativa gestione della lista contenente i termini inglesi più utilizzati.
- Le regole definite per l'identificazione delle Species risultano essere ben definite poiché le performance, come evidenziato dal suddetto confronto, sono influenzate unicamente dalla copertura di GBIF.

La *Tabella 4* mostra i dati inerenti alla totalità delle entità (contenuti nella seconda metà della *Tabella 3*) in relazione con i risultati ottenuti da *BotanicalNER* (propri della *Figura 15*).

I valori di Precision, così come quelli di F1 score, risultano essere, in ambo le categorie di identificazione, superiori a quelli di *BotanicalNER*. La Recall invece risulta essere più alta per *BotanicalNER*. Ciò è dato dal fatto che il nostro sistema, come dimostrato anche nei precedenti confronti (illustrati dalle *Tabelle 2 e 3*) risulta essere eccessivamente sensibile nell'identificazione delle Species, spesso riconoscendo entità non facenti parte di questo insieme ed andando quindi ad incrementare il numero dei False Positive.

Tuttavia, questi dati testimoniano come l'utilizzo del paradigma rule-based risulti essere perfettamente valido nel task di NER delle specie scientifiche dimostrandosi in grado di essere competitivo anche se direttamente confrontato, come nel caso in questione, con altre tipologie di software basate su modello statistico.

		TP	FP	FN	Precision	Recall	F1 Score
BotanicalNER	Species	819	61	63	93,03%	92,81%	92,92%
	Genus	949	73	158	92,86%	85,73%	89,15%
Nostro sistema	Species	800	0	82	100%	90,70%	95,10%
	Genus	1037	78	70	93,00%	93,00%	93,00%

Tabella 4: *Confronto del nostro sistema con Botanical NER [87]*

Dal confronto tra i due approcci utilizzati per la valutazione del sistema sviluppato (rispettivamente illustrati dalle *Tabelle 2, 3, 4*), emerge che c'è differenza tra la valutazione del nostro sistema quando le entità si considerano come un insieme privo di ripetizioni e quando si tiene in conto la posizione delle entità nel testo. Questa osservazione implica che la componente contestuale delle entità identificate è importante per il nostro sistema.

5.4 Tempi di esecuzione

Al termine delle fasi di sviluppo, l'applicativo realizzato nel nostro studio è stato testato al fine di poter valutare il tempo dell'esecuzione delle operazioni di NER. Queste sono state svolte distintamente sia per la componente di identificazione dei taxa che per quella dei Thesauri, utilizzando un dispositivo avente le seguenti caratteristiche:

1. Memoria RAM: 8 GB DDR4
2. Processore: AMD Ryzen 3 2200u con Radeon Vega Mobile GFX x4
3. Sistema operativo: Ubuntu 20.04.2 LTS 64-bit

In aggiunta, mediante le medesime metodologie di valutazione, sono state anche testate le tempistiche di esecuzione di *BotanicalNER* [87] al fine di contestualizzare al meglio questa tipologia di valutazione. La *Tabella 5* mostra i risultati di tale procedimento. I valori indicati nelle colonne dei tre rispettivi NER sono espressi in secondi mentre le unità che scandiscono i vari test riportano il numero di caratteri, in aggiunta è disponibile il rapporto tra questi ultimi due valori per ciascuno di essi. La componente adibita al riconoscimento dei thesauri, seppur utilizzando le medesime funzionalità di ricerca (realizzate dalla classe *EfficientSearchInText*) già illustrate nella *Sezione 3.2*, risulta avere tempi di esecuzione significativamente inferiori. Ciò è da attribuire alla dimensione dei gazetteers i quali, nella componente di NER inerente ai nomi scientifici, risulta essere di dimensioni molto più cospicue. L'ammontare in questione supera i 100 megabyte di dati contro i meno di 1000 kilobyte della componente adibita all'identificazione dei thesauri.

I valori di *BotanicalNER* risultano essere sensibilmente maggiori. Questo è dato dall'elevato costo computazionale che il modello LSTM-CRF bidirezionale richiede per poter eseguire le operazioni di NER.

N° di caratteri	N° secondi Thesaurus NER	Rapporto Thesaurus NER	N° secondi Taxon NER	Rapporto Taxon NER	N° secondi BotanicalNER	Rapporto Botanical NER
10000	0,1172	0,000012	5,13	0,0005	9,0303	0,0009
50000	1,369	0,000027	10,923	0,0002	42,5994	0,0009
100000	1,681	0,000017	19,659	0,0002	82,9479	0,0008
500000	3,394	0,000007	129,338	0,0003	397,1012	0,0008
1000000	5,422	0,000005	279,652	0,0003	810,5938	0,0008
5000000	22,173	0,000004	2537,89	0,0005	4022,35	0,0008

Tabella 5: *Tempistiche di esecuzione (espresse in secondi) di BotanicalNER e del software NER realizzato nel nostro studio nei due differenti domini di applicazione.*

Tali dati sono resi ancor più evidenti dalla *Figura 16* in cui è possibile constatare l'andamento nostro dell'applicativo in contrasto con quello di BotanicalNER. Quest'ultimo, come in generale per gli algoritmi basati su rete neurale, ha un andamento lineare. Il nostro software si comporta in maniera più efficiente avendo tempi di esecuzione sensibilmente inferiori. Inoltre, essendo una crescita quasi lineare, risulta possibile la previsione il calcolo dei tempi di elaborazione di documenti a partire dalla dimensione (espressa in numero di caratteri) degli stessi.



Fig. 16: Tempistiche di esecuzione (esprese in millisecondi) del software NER realizzato in contrapposizione con BotanicalNER

6 Discussione

6.1 Sommario e valutazione qualitativa del sistema e dei risultati nei confronti delle soluzioni attuali.

I requisiti richiesti dalla FAO inerenti all'applicativo sviluppato nel nostro studio (*Sezione 1.1*) non erano soddisfatti da nessun software esistente (*Sezione 2.4*). L'impiego del paradigma rule-based, seppur maggiormente legato alla staticità delle regole che dominano l'identificazione delle entità, ha permesso l'attuazione di diverse funzionalità che si sono rivelate fondamentali nel successo del progetto (*Sezione 3.1*).

In primo luogo è stata incorporata conoscenza esperta nell'identificazione dei nomi scientifici all'interno delle regole di riconoscimento del software. Tale processo è stato svolto utilizzando le informazioni contenute all'interno dell'articolo "*Taxa Merging Discussion*" (2012) [64] e sviluppate nel contesto del progetto europeo i-Marine, che aveva tra i suoi scopi anche quello di riportare in maniera rigorosa anche questo tipo di conoscenza. Queste informazioni, illustrate in dettaglio nella *Sezione 2.2*, sono state redatte da esperti del relativo settore e descrivono puntualmente le varie problematiche inerenti all'identificazione dei nomi scientifici latini. La trattazione è corredata da esemplificazioni atte alla rappresentazione delle varie casistiche comprensive della totalità delle possibili eventualità verificabili in tal senso. Tutte queste indicazioni costituiscono l'insieme di conoscenza mediante la quale sono state generate le regole che dominano il processo di NER nel software sviluppato (*Sezione 3.2*).

Coerentemente al paradigma rule-based è stato necessario utilizzare più gazetteers adibiti alla rappresentazione dell'insieme di entità identificabili dal programma. Questo scenario è generalmente percepito come una limitazione poiché circoscrive in maniera rigida il dominio applicativo di tali software. Ciononostante, sfruttando l'elevata quantità e qualità dei dati FAIR, ampiamente disponibili presso le varie istituzioni del settore, si è riusciti ad ovviare a questa problematica volgendola come un concreto punto di forza. Nell'atto pratico è stato utilizzato il collettore di dati GBIF [8] per i Genus e Species ed il dataset *ASFA* [46] per l'identificazione dei thesauri. Essi sono stati rielaborati mediante una serie di operazioni volte al fine di poter garantire la totale riproducibilità dell'intero es-

perimento presentato (*Sezione 3.1.1*).

L'integrazione all'interno della piattaforma di cloud computing NLPHub costituisce una risorsa significativa per il lavoro realizzato il quale, mediante tale piattaforma, è reso fruibile tramite un servizio web dedicato, disponibile sia in modalità stand-alone [43] che all'interno dell'infrastruttura D4Science [18]. Quest'ultima permette di usufruire di esose risorse computazionali e di poter condividere facilmente il lavoro svolto con eventuali collaboratori. Ogni operazione eseguita all'interno di questa infrastruttura è marcata mediante un relativo identificativo al fine di garantire la totale riproducibilità delle azioni svolte (*Sezione 3.4 e 3.5*).

Il finanziamento che ha permesso la progettazione del nostro lavoro risulta essere di natura prettamente indiretta poiché il software sviluppato è di fatto costruito dall'opportuno riutilizzo di dati FAIR. Questi ultimi, direttamente sovvenzionati dalla comunità scientifica, risultano essere in totale coerenza al paradigma Open Science (*Sezione 3.6*). Tale filosofia la si può di fatto riscontrare in ciascuna delle fasi realizzative del progetto. Il collettore di dati GBIF è sovvenzionato da contributi annui provenienti da 43 entità governative nazionali mentre il dataset contenente i thesauri è realizzato dalla FAO e quindi con i fondi stanziati dagli stati membri di tale organizzazione [5]. Infine, D4Science e i-Marine sono il risultato di progetti pubblici europei.

I vantaggi dell'uso di dati FAIR non è solo inerente al costo di realizzazione di un applicativo come quello presentato, ma anche della sua manutenzione ed estensione. La totale adempienza al paradigma Open science facilita tali procedure le quali, coerentemente al formato e alla disponibilità dei dati FAIR utilizzati, risulta essere semplificata ed in larga parte anche automatizzabile. In aggiunta, come dimostrato nella *Sezione 5.4*, l'utilizzo del paradigma rule-based, in complicità con un'opportuna ottimizzazione delle funzionalità di ricerca, ha garantito la realizzazione di un software efficiente avente tempi di esecuzione inferiori rispetto al software di riferimento utilizzato. Oltre che a necessitare di risorse computazionali sensibilmente limitate, il programma realizzato, sfruttando le caratteristiche peculiare delle entità target e del paradigma rule-based, è in grado di conseguire le operazioni di NER indipendentemente dalla lingua in input.

Dal diretto confronto con *BotanicalNER* emerge che il nostro sistema raggiunge performance superiori nell'identificare i nomi scientifici delle specie. Questo fattore, in concomitanza alle scelte progettuali realizzate a partire da dati di origine

conformi alle proprietà FAIR e all'utilizzo di una metodologia di identificazione costruita sul paradigma rule-based (che si contrappone agli approcci basati su modello statistico in quanto rende esplicita ogni operazione eseguita), ha reso il software sviluppato nel nostro studio conforme ai requisiti richiesti dalla FAO (*Sezione 5*). È opportuno chiarire che il fatto che BotanicalNER risulti essere meno efficace nell'identificazione dei nomi scientifici delle specie è da imputare alla natura di tale progetto che, a differenza del nostro, è concepito per sfruttare le informazioni contenute all'interno del contesto grammaticale, mediante il modello statistico LSTM-CRF bidirezionale, al fine di identificare anche i nomi vernacolari delle specie [87].

6.2 Potenziali utilizzi

Sulla base delle caratteristiche spiegate in *Sezione 6.1*, l'Organizzazione delle Nazioni Unite per l'alimentazione e l'agricoltura (FAO) ha deciso di adottare la soluzione proposta nel nostro studio per l'etichettatura di Abstract inerenti alla pesca e all'acquacoltura. Nello specifico, il nostro software è impiegato per l'estrazione ed assegnazione di tag a tali documenti che si concretizza mediante operazioni di etichettatura basate sulle named entities estratte.

La sua applicazione è pensata per essere integrata nel sistema NLPHub attraverso il quale esso è combinabile mediante l'utilizzo di *Orchestrator* con altri algoritmi NER. Questi concorrono nella generazione di un output combinato nel quale risultano essere marcati non solo nomi scientifici e thesauri (utilizzabili in contemporanea) ma anche relativi alle entità di:

1. Keyword
2. Location
3. Organization
4. Person

Inoltre, dato l'elevato grado di copertura di GBIF, il nostro software può essere utilizzato anche con altri taxa oltre che a quelli delle scienze della pesca e acquacoltura. Questo è dimostrato anche dal confronto effettuato e riportato nel dominio della Botanica (*Sezione 5.3.2*).

Come già affermato in *Sezione 6.1* e dimostrato parzialmente in 5.4 il programma realizzato nel nostro studio risulta essere eseguibile anche tramite macchine aventi risorse computazionali moderate. Questo fattore permette non solo di agevolare l'esecuzione all'interno del sistema di cloud computing ma anche di ridurre i possibili costi delle procedure di implementazione. In aggiunta, la struttura dell'algoritmo adibito alle operazioni di ricerca risulta essere notevolmente personalizzabile, fattore che rende possibile eventuali modifiche mirate all'ottimizzazione di tale processo in relazione alle caratteristiche hardware di una nuova macchina ospitante.

L'utilizzo del paradigma rule-based in concomitanza all'impiego di grandi gazetteer,

facilita non solo le operazioni di aggiornamento ma permette anche di poter testare il programma mediante un differente dataset da impiegare come supporto alle fasi di annotazione. Se ad esempio si volesse utilizzare il database *WoRMS* [37] sarebbe necessario unicamente svolgere delle operazioni volte alla conformazione della struttura di tale dataset al fine di renderlo compatibile con i requisiti strutturali del programma realizzato senza andare ad intaccare il codice di cui esso è composto.

6.3 Estensioni future

6.3.1 Estensioni previste dal committente del progetto

Il software realizzato risulta essere totalmente in linea con le caratteristiche prefissate a monte delle operazioni di sviluppo. Queste, appartenenti al primo stadio di progettazione previsto per tale applicativo, sono state ideate per essere compatibili con le ulteriori quattro fasi contemplate anteriormente dell'inizio della definizione dell'applicativo.

I documenti che contengono Abstract inerenti alle scienze acquatiche e della pesca sono memorizzati generalmente nei formati di Portable Document Format (PDF) e "doc/docx". Il secondo stage previsto è inerente proprio allo sviluppo di uno o più algoritmi in grado di riconoscere tali formati e di essere in grado di identificare la componente di Abstract al loro interno (attraverso la funzionalità di automatic Abstract identification) al fine di ottenere il testo necessario, in formato *plain text*, per permettere l'esecuzione del nostro NER. A differenza dei documenti prodotti mediante Microsoft Office Word, i file in formato PDF possono indurre il computer a commettere errori di interpretazione del contenuto poiché il testo al loro interno è spesso strutturato mediante modalità che favoriscono la visualizzazione a discapito dell'interpretazione dei caratteri dai quali esso è composto. Per ovviare a questa problematica, sempre nella seconda fase di sviluppo, è prevista l'implementazione di una componente in grado di realizzare un processo di riconoscimento ottico dei caratteri (OCR) che rende capace la macchina di decifrare tale contenuto senza il verificarsi di imprecisioni.

Il terzo stage riguarda le configurazioni di integrazione dell'applicativo con i vari servizi di NER disponibili mediante l'utilizzo di NLP Hub. Secondo gli obiettivi previsti per il progetto, il software realizzato, una volta identificata la lingua che compone il testo in input deve configurarsi diversamente nelle seguenti metodologie di identificazione delle classi di entità in rapporto con la stessa:

1. **Inglese:** Location, Person, Organization, Keyword, Species
2. **French:** Location, Person, Keyword, Species
3. **Italian:** Location, Person, Organization, Keyword, Species

4. **Spanish:** Location, Person, Organization, Keyword, Species
5. **German:** Location, Person, Organization, Keyword, Species

La quarta fase di implementazione è inerente all'ampliamento del supporto multilingua del programma. Nello specifico per le lingue di russo, portoghese, greco e olandese. La caratteristica dell'indipendenza dalla lingua in input, propria del programma sviluppato, è da considerarsi come valida ma ancora sprovvista di opportuni test che mirino alla verifica della stessa. Inoltre, la gestione dei diversi alfabeti facenti parte delle lingue in questione, può essere la causa di relative problematiche inerenti all'interpretazione della codifica dei caratteri delle stesse. In aggiunta è previsto l'iniziale supporto nell'identificazione di Keyword per le lingue di Coreano e Cinese. Queste impongono un sviluppo fortemente relazionato con gli altri algoritmi di NER disponibili in NLP Hub, fattore che incrementa il livello di collaborazione generale che vi è tra il software sviluppato e le funzionalità proprie di tale infrastruttura.

L'ultima fase di sviluppo è inerente alla realizzazione di una metodologia che permetta all'utente di poter utilizzare una lista personalizzata, composta dalle entità di thesauri, da impiegare come gazetteer per l'identificazione degli stessi. Questa funzionalità dovrà essere realizzata mediante l'implementazione dei seguenti costrutti:

1. Predisponendo il programma agli utilizzi con insiemi di dati esterni, incrementando quindi l'elasticità dello stesso.
2. Fornendo una navigazione guidata per permettere all'utente di inserire la propria lista di termini.
3. Realizzando uno standard solido al quale tale lista debba aderire in relazione alla forma delle entry contenute al suo interno.

6.3.2 Altre estensioni

In aggiunta a quelle che risultano essere le estensioni stabilite di pari passo alla progettazione del progetto, trovano collocazione in questa sezione una serie di possibili implementazioni ragionate sulla base delle caratteristiche attualmente presenti nel programma stesso.

Come già parzialmente suggerito nella *Sezione 6.2*, l'applicativo realizzato risulta essere costruito in linea con le proprietà strutturali del paradigma rule-based, mediante una netta separazione tra dizionari annessi all'annotazione (gazetteers) e le regole definite per il riconoscimento delle entità target. Essendo GBIF un dataset FAIR si può automatizzare il processo di aggiornamento rendendo automatiche le operazioni di download di tale insieme di dati.

Infine, l'incorporazione di un sistema di taxa name matching, come ad esempio BiOnym [21] (illustrato nella *Sezione 2.3*) permetterebbe di aumentare l'informazione allegata ad ogni nome scientifico identificato aggiungendo informazione su sinonimi, nomi comuni e codici di riferimento in molteplici knowledge base di riferimento, come ad esempio: Catalogue of life [115], FishBase [51], WoRMS [127] ecc.

7 Acronimi

NER = Named Entity Recognizer

FAO = Food and Agriculture Organization of the United Nations

ASFA = Aquatic Sciences and Fisheries Abstracts

NERC = Named Entity Recognition and Classification

MUC 7 = Message Understanding Conference

GNI = Global Names Index

GNA = The Global Names Architecture

GBIF = Global Biodiversity Information Facility

GUI = Graphical User Interface

TAF = Taxonomic Authority File

GSay = Genus Species Authority year

DWG = International Taxonomic Database Working Group

DNN = Deep neural network

LSTM = Long short-term memory

CRF = Conditional random field

LSTM-CRF = Long short-term memory - Conditional random field

EF = Education First

UTF-8 = Unicode Transformation Format, 8 bit

WPS = Wi-Fi Protected Setup

GARR = Italian Academic and Research Network

ILC-CNR = Institute for Computational Linguistics «A. Zampolli» National Research Council of Italy

OS = Open Science

CoL = Catalogue of Life

IPNI = The International Plant Name Index

MMPND = Multilingual Multiscript Plant Name Database

GRIN = Germplasm Resources Information Network

ASC = Swiss Alpine Club

NLTK = Natural Language Toolkit

PoS = Part of Speech

IOB = Inside Outside Beginning

TP = True positive

FP = False positive

FN = False negative

A = Accuracy

P = Precision

R = Recall

F1 = F1-Score

PDF = Portable Document Format

FAIR = Findability, Accessibility, Interoperability, and Reuse of digital assets

OCR = Optical character recognition

Bibliografia

- [1] The 3 pillars of binary classification: Accuracy, precision & recall. <https://medium.com/@yashwant140393/the-3-pillars-of-binary-classification-accuracy-precision-recall-d2da3d09>. Controllata il: 2021-11-2.
- [2] Aquatic sciences and fisheries abstracts (asfa). <https://www.fao.org/fishery/asfa/en>. Controllata il: 2021-11-18.
- [3] Asfa - aquatic sciences and fisheries abstracts. <https://www.fao.org/about/en/>. Controllata il: 2021-11-18.
- [4] Asfa - aquatic sciences and fisheries abstracts. <https://www.fao.org/policy-support/mechanisms/mechanisms-details/en/c/428685/>. Controllata il: 2021-11-18.
- [5] Country profiles. <https://www.fao.org/countryprofiles/en/>. Controllata il: 2021-11-2.
- [6] The f1 score. <https://towardsdatascience.com/the-f1-score-bec2bbc38aa6>. Controllata il: 2021-11-2.
- [7] Main page. http://wiki.i-marine.eu/index.php/Main_Page. Controllata il: 2021-11-2.
- [8] GBIF Secretariat (2021). Gbif backbone taxonomy, 2021. [Online; controllata il: 29-September-2021].
- [9] Placidus a Spescha. *Beschreibung der Val Tujetsch (1806)*. Chronos, 2009.
- [10] Mohammed Abdelrahim, Carlos Merlos, et al. Hybrid machine learning approaches: A method to improve expected output of semi-structured sequential data. In *2016 IEEE Tenth International Conference on Semantic Computing (ICSC)*, pages 342–345. IEEE, 2016.
- [11] Oluwasegun Adedugbe, Elhadj Benkhelifa, and Russell Champion. A cloud-driven framework for a holistic approach to semantic annotation. In *2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 128–134. IEEE, 2018.
- [12] Aeschimann and C. Heitz. *Synonymie. Synonymie index der schweizer flora und der angrenzenden gebiete*, 2005.

- [13] N Agrawal and A Singla. Using named entity recognition to improve machine translation. *Technical report, Stanford University, Natural Language Processing*, 2012.
- [14] Beatrice Alex, Barry Haddow, and Claire Grover. Recognising nested named entities in biomedical text. In *Biological, translational, and clinical language processing*, pages 65–72, 2007.
- [15] Enrique Alfonseca and Suresh Manandhar. An unsupervised method for general named entity recognition and automated concept discovery. In *Proceedings of the 1st international conference on general WordNet, Mysore, India*, pages 34–43, 2002.
- [16] Rayner Alfred, Leow Chin Leong, Chin Kim On, and Patricia Anthony. Malay named entity recognition based on rule-based approach. *International Journal of Machine Learning and Computing*, 4(3), 300-306, 2014.
- [17] AquaDocs. Welcome to aquadocs! [Online; controllata il 30-11-2011].
- [18] Massimiliano Assante, Leonardo Candela, Donatella Castelli, Roberto Cirillo, Gianpaolo Coro, Luca Frosini, Lucio Lelii, Francesco Mangiacrapa, Pasquale Pagano, Giancarlo Panichi, et al. Enacting open science by d4science. *Future Generation Computer Systems*, 101:555–563, 2019.
- [19] Gregory V Bard. Spelling-error tolerant, order-independent pass-phrases via the damerau-levenshtein string-edit distance metric. In *Proceedings of the fifth Australasian symposium on ACSW frontiers-Volume 68*, pages 117–124. Citeseer, 2007.
- [20] Pierpaolo Basile, Annalina Caputo, and Giovanni Semeraro. Entity linking for italian tweets. In *Proceedings of the Second Italian Conference on Computational Linguistics CLiC-it*, volume 3, page 4, 2015.
- [21] Edward Vanden Berghe, Gianpaolo Coro, Nicolas Bailly, Fabio Fiorellato, Caselyn Aldemita, Anton Ellenbroek, and Pasquale Pagano. Retrieving taxa names from large biodiversity data collections using a flexible matching workflow. *Ecological Informatics*, 28:29–41, 2015.
- [22] Daniel M Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. Nymble: a high-performance learning name-finder. *arXiv preprint cmp-lg/9803003*, 1998.
- [23] Kalina Bontcheva and Leon Derczynski. Extracting information from social media with gate. In *Working with Text*, pages 133–158. Elsevier, 2016.

- [24] Andrew Eliot Borthwick. *A maximum entropy approach to named entity recognition*. New York University, 1999.
- [25] Hans Heinrich Bosshard, Hans Heinrich Bosshard, Hans Heinrich Bosshard, and Hans Heinrich Bosshard. *Mundartnamen von Bäumen und Sträuchern in der deutschsprachigen Schweiz und im Fürstentum Liechtenstein*. Bühler, 1978.
- [26] Brad Boyle, Nicole Hopkins, Zhenyuan Lu, Juan Antonio Raygoza Garay, Dmitry Mozzherin, Tony Rees, Naim Matasci, Martha L Narro, William H Piel, Sheldon J Mckay, et al. The taxonomic name resolution service: an online tool for automated standardization of plant names. *BMC bioinformatics*, 14(1):1–15, 2013.
- [27] Sergey Brin. Extracting patterns and relations from the world wide web. In *International workshop on the world wide web and databases*, pages 172–183. Springer, 1998.
- [28] A.J. Cain. taxonomy. <https://www.britannica.com/science/taxonomy>, 2020. [Online; controllata il: 27-August-2021].
- [29] Scott A Chamberlain and Eduard Szöcs. taxize: taxonomic search and retrieval in r. *F1000Research*, 2, 2013.
- [30] Hai Leong Chieu and Hwee Tou Ng. Named entity recognition: a maximum entropy approach using global information. In *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002.
- [31] Jason PC Chiu and Eric Nichols. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370, 2016.
- [32] Gianpaolo Coro, Leonardo Candela, Pasquale Pagano, Angela Italiano, and Loredana Liccardo. Parallelizing the execution of native data mining algorithms for computational biology. *Concurrency and Computation: Practice and Experience*, 27(17):4630–4644, 2015.
- [33] Gianpaolo Coro, Marco Palma, Anton Ellenbroek, Giancarlo Panichi, Thiviya Nair, and Pasquale Pagano. Reconstructing 3d virtual environments within a collaborative e-infrastructure. *Concurrency and Computation: Practice and Experience*, 31(11):e5028, 2019.

- [34] Gianpaolo Coro, Giancarlo Panichi, Pasquale Pagano, and Erico Perrone. Nlphub: An e-infrastructure-based text mining hub. *Concurrency and Computation: Practice and Experience*, 33(5):e5986, 2021.
- [35] Gianpaolo Coro, Giancarlo Panichi, Paolo Scarponi, and Pasquale Pagano. Cloud computing in a distributed e-infrastructure using the web processing service standard. *Concurrency and Computation: Practice and Experience*, 29(18):e4219, 2017.
- [36] Gianpaolo Coro, Luis Gonzalez Vilas, Chiara Magliozzi, Anton Ellenbroek, Paolo Scarponi, and Pasquale Pagano. Forecasting the ongoing invasion of ligocephalus scleratus in the mediterranean sea. *Ecological Modelling*, 371:37–49, 2018.
- [37] Mark J Costello, Philippe Bouchet, Geoff Boxshall, Kristian Fauchald, Dennis Gordon, Bert W Hoeksema, Gary CB Poore, Rob WM van Soest, Sabine Stöhr, T Chad Walter, et al. Global coordination and standardisation in marine biodiversity through the world register of marine species (worms) and related databases. *PloS one*, 8(1):e51629, 2013.
- [38] Felice Dell’Orletta, Giulia Venturi, Andrea Cimino, and Simonetta Montemagni. T2k²: a system for automatically extracting and organizing knowledge from texts. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*, pages 2062–2070, 2014.
- [39] Julianna Delua. Supervised vs. unsupervised learning: What’s the difference? <https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning>, 2021. [Online; controllata il: 8-September-2021].
- [40] dimensional research. Artificial intelligence and machine learning projects are obstructed by data issues. <https://cdn2.hubspot.net/hubfs/3971219/Survey%20Assets%201905/Dimensional%20Research%20Machine%20Learning%20PPT%20Report%20FINAL.pdf>, 2019. [Online; controllata il: 31-August-2021].
- [41] James L Edwards, Meredith A Lane, and Ebbe S Nielsen. Interoperability of biodiversity databases: biodiversity information on every desktop. *Science*, 289(5488):2312–2314, 2000.
- [42] EF. 3000 most common words in english, 2021. [Online; controllata il: 30-September-2021].

- [43] Coro et al. Nlp hub-asfa, 2021. [Online; controllata il: 19-October-2021].
- [44] Kottmann J et al. Apache opennlp, 2011.
- [45] Richard Evans and Stafford Street. A framework for named entity recognition in the open domain. *Recent advances in natural language processing III: selected papers from RANLP*, 260(267-274):110, 2003.
- [46] FAO. Asfa, 2021. [Online; controllata il: 19-October-2021].
- [47] Dimitra Farmakiotou, Vangelis Karkaletsis, John Koutsias, George Sigletos, Constantine D Spyropoulos, and Panagiotis Stamatopoulos. Rule-based named entity recognition for greek financial texts. In *Proceedings of the Workshop on Computational lexicography and Multimedia Dictionaries (COMLEX 2000)*, pages 75–78. Citeseer, 2000.
- [48] Fisheries and Aquaculture Department. Search publications. [Online; controllata il 30-11-2011].
- [49] Radu Florian, Abe Ittycheriah, Hongyan Jing, and Tong Zhang. Named entity recognition through classifier combination. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pages 168–171, 2003.
- [50] International Organization for Standardization. *ISO 5725-1: 1994: accuracy (trueness and precision) of measurement methods and results-part 1: general principles and definitions*. International Organization for Standardization, 1994.
- [51] Rainer Froese and Daniel Pauly. *FishBase 2000: concepts designs and data sources*, volume 1594. WorldFish, 2000.
- [52] GBIF. The gbif ecat programme, 2014. [Online; controllata il: 21-September-2021].
- [53] Martin Gerner, Goran Nenadic, and Casey M Bergman. Linnaeus: a species name identification system for biomedical literature. *BMC bioinformatics*, 11(1):1–17, 2010.
- [54] GNA. The global names architecture, 2014. [Online; controllata il: 21-September-2021].
- [55] GOFAIR. Fair principles), 2021. [Online; controllata il 23-11-2011].

- [56] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- [57] Ralph Grishman and Beth M Sundheim. Message understanding conference-6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, 1996.
- [58] Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. Named entity recognition in query. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 267–274, 2009.
- [59] Maryam Habibi, Leon Weber, Mariana Neves, David Luis Wiegandt, and Ulf Leser. Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*, 33(14):i37–i48, 2017.
- [60] Anthony JG Hey, Stewart Tansley, Kristin Michele Tolle, et al. *The fourth paradigm: data-intensive scientific discovery*, volume 1. Microsoft research Redmond, WA, 2009.
- [61] Walter Höhn-Ochsner. *Pflanzen in Zürcher Mundart und Volksleben: Zürcher Volksbotanik*. H. Rohr, 1986.
- [62] Matthew Honnibal and Ines Montani. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1):411–420, 2017.
- [63] IGI-Global. What is e-infrastructure, 2021. [Online; controllata il: 19-October-2021].
- [64] iMarine Project Wiki. Taxa merging discussion. http://wiki.i-marine.eu/index.php?title=Taxa_Merging_Discussion&action=edit, 2012. [Online; controllata il: 27-August-2021].
- [65] Suwisa Kaewphan, Kai Hakala, Niko Miekka, Tapio Salakoski, and Filip Ginter. Wide-scope biomedical named entity recognition and normalization with crfs, fuzzy matching and character level modeling. *Database*, 2018, 2018.
- [66] Will Kenton. Food and agriculture organization (fao), 2021. [Online; controllata il 18-11-2011].
- [67] J-D Kim, Tomoko Ohta, Yuka Tateisi, and Jun’ichi Tsujii. Genia corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl_1):i180–i182, 2003.

- [68] Thomas A Kluyver and Colin P Osborne. Taxonome: a software package for linking biological species data. *Ecology and evolution*, 3(5):1262–1265, 2013.
- [69] Drew Koning, Indra Neil Sarkar, and Thomas Moritz. Taxongrab: Extracting taxonomic names from text. *Biodiversity Informatics*, 2, 2005.
- [70] Saul A Kripke. Naming and necessity. In *Semantics of natural language*, pages 253–355. Springer, 1972.
- [71] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016.
- [72] George Lawton. Supervised vs. unsupervised learning: Use in business. <https://searchenterpriseai.techtarget.com/feature/Comparing-supervised-vs-unsupervised-learning>, 2021. [Online; controllata il: 8-September-2021].
- [73] Nicolas Le Guillarme and Wilfried Thuiller. Taxonerd: deep neural models for the recognition of taxonomic entities in the ecological and evolutionary literature. *bioRxiv*, 2021.
- [74] Leary. Taxonfinder. <https://github.com/pleary/node-taxonfinder>, 2014.
- [75] N. Bubenhofer M. Volk F. Leuenberger and D. Wüest. Text+berg-korpus (release 151 version 01, 2015).
- [76] Chahira Lhioui, Anis Zouaghi, and Mounir Zrigui. A rule-based approach for arabic temporal expression extraction. In *2017 International Conference on Engineering & MIS (ICEMIS)*, pages 1–6. IEEE, 2017.
- [77] Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. A unified mrc framework for named entity recognition. *arXiv preprint arXiv:1910.11476*, 2019.
- [78] Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. A unified mrc framework for named entity recognition. *arXiv preprint arXiv:1910.11476*, 2019.
- [79] David S Linthicum. Cloud computing changes data integration forever: What’s needed right now. *IEEE Cloud Computing*, 4(3):50–53, 2017.

- [80] Edward Loper and Steven Bird. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*, 2002.
- [81] Tyler Tian Lu. Fundamental limitations of semi-supervised learning. Master’s thesis, University of Waterloo, 2009.
- [82] Ying Luo and Hai Zhao. Bipartite flat-graph network for nested named entity recognition. *arXiv preprint arXiv:2005.00436*, 2020.
- [83] Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*, 2016.
- [84] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.
- [85] Alireza Mansouri, Lilly Suriani Affendey, and Ali Mamat. Named entity recognition approaches. *International Journal of Computer Science and Network Security*, 8(2):339–344, 2008.
- [86] Alexa T McCray, Alan R Aronson, Allen C Browne, Thomas C Rindflesch, Amir Razi, and Suresh Srinivasan. Umls knowledge for biomedical language processing. *Bulletin of the Medical Library Association*, 81(2):184, 1993.
- [87] Isabel Meraner. *Grasping the Nettle: Neural Entity Recognition for Scientific and Vernacular Plant Names*. PhD thesis, University of Zurich, 2019.
- [88] Ivona Milanova, Jurij Silc, Miha Serucnik, Tome Eftimov, and Hristijan Gjoreski. Locale: A rule-based location named-entity recognition method for latin text. In *HistoInformatics@ TPD*, pages 13–20, 2019.
- [89] Dmitry Y Mozzherin, Alexander A Myltsev, and David J Patterson. “gn-parser”: a powerful parser for scientific names based on parsing expression grammar. *BMC bioinformatics*, 18(1):1–14, 2017.
- [90] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, 2007.
- [91] Nona Naderi, Thomas Kappler, Christopher JO Baker, and René Witte. Organismtagger: detection, normalization and grounding of organism entities in biomedical documents. *Bioinformatics*, 27(19):2721–2729, 2011.

- [92] Mai Oudah and Khaled Shaalan. A pipeline arabic named entity recognition using a hybrid approach. In *Proceedings of COLING 2012*, pages 2159–2176, 2012.
- [93] Evangelos Pafilis, Sune P Frankild, Lucia Fanini, Sarah Faulwetter, Christina Pavloudi, Aikaterini Vasileiadou, Christos Arvanitidis, and Lars Juhl Jensen. The species and organisms resources for fast and accurate identification of taxonomic names in text. *PloS one*, 8(6):e65390, 2013.
- [94] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.
- [95] Marius Pasca, Dekang Lin, Jeffrey Bigham, Andrei Lifchits, and Alpa Jain. Organizing and searching the world wide web of facts-step one: the one-million fact extraction challenge. In *AAAI*, volume 6, pages 1400–1405, 2006.
- [96] David J Patterson, J Cooper, Paul M Kirk, RL Pyle, and David P Remsen. Names are key to the big new biology. *Trends in ecology & evolution*, 25(12):686–691, 2010.
- [97] Qi Peng, Changmeng Zheng, Yi Cai, Tao Wang, Haoran Xie, and Qing Li. Unsupervised cross-domain named entity recognition using entity-aware adversarial training. *Neural Networks*, 138:68–77, 2021.
- [98] Jorrit H Poelen, James D Simons, and Chris J Mungall. Global biotic interactions: An open infrastructure to share and analyze species-interaction datasets. *Ecological Informatics*, 24:148–159, 2014.
- [99] Gorjan Popovski, Stefan Kochev, Barbara Korousic-Seljak, and Tome Eftimov. Foodie: A rule-based named-entity recognition method for food information extraction. In *ICPRAM*, pages 915–922, 2019.
- [100] Michel H Porcher et al. Multilingual multiscrypt plant name database. *University of Melbourne w ww plantnames unimelb edu au/Sorting/Mushrooms_Intro.html*, 2005.
- [101] ProQuest. Proquest. [Online; controllata il 30-11-2011].
- [102] Lance A Ramshaw and Mitchell P Marcus. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176. Springer, 1999.

- [103] Tony Rees. Taxamatch, a “fuzzy” matching algorithm for taxon names, and potential applications in taxonomic databases,”. *Proceedings of TDWG*, page 35, 2008.
- [104] Kashif Riaz. Rule-based named entity recognition in urdu. In *Proceedings of the 2010 named entities workshop*, pages 126–135, 2010.
- [105] Ellen Riloff, Rosie Jones, et al. Learning dictionaries for information extraction by multi-level bootstrapping. In *AAAI/IAAI*, pages 474–479, 1999.
- [106] Tim Rocktäschel, Michael Weidlich, and Ulf Leser. Chemspot: a hybrid system for chemical named entity recognition. *Bioinformatics*, 28(12):1633–1640, 2012.
- [107] Yuri Roskov, Thomas Kunze, L Paglinawan, T Orrell, D Nicolson, Alastair Culham, N Bailly, P Kirk, T Bourgoïn, G Baillargeon, et al. Species 2000 & itis catalogue of life, 2013 annual checklist. 2013.
- [108] ISHA SALIAN. Supervize me: What’s the difference between supervised, unsupervised, semi-supervised and reinforcement learning? <https://blogs.nvidia.com/blog/2018/08/02/supervised-unsupervised-learning/>, 2018. [Online; controllata il: 8-September-2021].
- [109] Erik F Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*, 2003.
- [110] Guido Sautter, Klemens Böhm, and Donat Agosti. A combining approach to find all taxon names (fat). *Biodiversity informatics*, 3, 2006.
- [111] Helmut Schmid. Treetagger-a language independent part-of-speech tagger. <http://www.ims.uni-stuttgart.de/projekte/complex/TreeTagger/>, 1994.
- [112] Satoshi Sekine and Chikashi Nobata. Definition, dictionaries and tagger for extended named entity hierarchy. In *LREC*, pages 1977–1980. Lisbon, Portugal, 2004.
- [113] Yusuke Shinyama and Satoshi Sekine. Named entity discovery using comparable news articles. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 848–853, 2004.
- [114] Keiji Shinzato, Satoshi Sekine, Naoki Yoshinaga, and Kentaro Torisawa. Constructing dictionaries for named entity recognition on specific domains

- from the web. In *Web Content Mining with Human Language Technologies Workshop on the 5th International Semantic Web*. Citeseer, 2006.
- [115] Reading Species and Washington Integrated Taxonomic Information System. *Catalogue of life..... annual checklist; indexing the world's known species*. School of Plant Sciences, University of Reading, 2000.
- [116] Valentin Tablan, Ian Roberts, Hamish Cunningham, and Kalina Bontcheva. Gatecloud. net: Cloud infrastructure for large-scale, open-source text processing. In *UK e-Science All hands Meeting*, 2011.
- [117] The Australian National Herbarium The Royal Botanic Gardens, The Harvard University Herbaria. The international plant names index, 2012. [Online; controllata il: 19-October-2021].
- [118] TNC. Taxonomic nomenclature checker, 2014. [Online; controllata il: 21-September-2021].
- [119] Johns Hopkins University. See the latest data in your region, 2021. [Online; controllata il: 30-September-2021].
- [120] Piek Vossen, Rodrigo Agerri, Itziar Aldabe, Agata Cybulska, Marieke van Erp, Antske Fokkens, Egoitz Laparra, Anne-Lyse Minard, Alessio Palmero Aprosio, German Rigau, et al. Newsreader: Using knowledge resources in a cross-lingual reading machine to generate more knowledge from massive streams of news. *Knowledge-Based Systems*, 110:60–85, 2016.
- [121] Aditya Satria Wibawa and Ayu Purwarianti. Indonesian named-entity recognition for 15 classes using ensemble supervised learning. *Procedia Computer Science*, 81:221–228, 2016.
- [122] Wikipedia. Organizzazione delle nazioni unite per l'alimentazione e l'agricoltura — wikipedia, l'enciclopedia libera, 2021. [Online; controllata il 18-11-2011].
- [123] Wikipedia. Systematics — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Systematics&oldid=1037384036>, 2021. [Online; controllata il: 27-August-2021].
- [124] Wikipedia. Taxonomy — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Taxonomy&oldid=1039249142>, 2021. [Online; controllata il: 27-August-2021].
- [125] Wikiversity. Taxonomy (biology) — wikiversity,, 2021. [Online; controllata il: 10-March-2021].

- [126] Edward O Wilson. The encyclopedia of life. *Trends in Ecology & Evolution*, 18(2):77–80, 2003.
- [127] WoRMS. The worms taxon matcher, 2014. [Online; controllata il: 21-September-2021].
- [128] Andre Ye. The severe limitations of supervised learning are piling up. <https://medium.com/analytics-vidhya/the-severe-limitations-of-supervised-learning-are-piling-up-eca1ecf3e113>, 2020. [Online; controllata il: 4-September-2021].
- [129] Wajdi Zaghouani. Renar: A rule-based arabic named entity recognition system. *ACM Transactions on Asian Language Information Processing (TALIP)*, 11(1):1–13, 2012.
- [130] Shaodian Zhang and Noémie Elhadad. Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts. *Journal of biomedical informatics*, 46(6):1088–1098, 2013.
- [131] Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009.
- [132] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. 2005.
- [133] Imed Zitouni. *Natural language processing of semitic languages*. Springer, 2014.