Università degli Studi di Milano Bicocca
Scuola di Scienze
Dipartimento di Informatica, Sistemistica e Comunicazione
Corso di laurea in Informatica

Sviluppo e test di tecniche di Natural Language Processing per l'Analisi Reputazionale

Relatore: *Prof.ssa Elisabetta Fersini*
Co-relatore: *Dott. Andrea Catacchio*

Relazione della prova finale di:
Lorenzo Lorgna
Matricola 829776

Anno Accademico 2019-2020

Indice

Elenco delle figure

Elenco delle tabelle

Introduzione

Il riciclaggio dei proventi illeciti rappresenta un problema di grande attualità e minaccia l'integrità e la stabilità del sistema bancario e finanziario. Esso deteriora il flusso di investimenti provenienti dall'estero e inquina vasti settori dell'economia di un Paese o addirittura di intere comunità di Stati, andando a determinare una perdita di credibilità delle relative istituzioni. L'attività di riciclaggio consiste nell'investire capitali provenienti da reato all'interno di attività lecite in modo tale da "ripulire" tali somme di denaro e immetterle nel circolo economico attraverso sbocchi perfettamente legali. Il riciclaggio del denaro avviene principalmente in tre fasi: placement stage, layering stage, integration stage. Il placement stage è la fase in cui i capitali di origine illecita vengono introdotti nel mercato. A seguire vi è il layering stage che consiste nell'andare ad eliminare qualsiasi collegamento tra i fondi riciclati e le attività illecite da cui provengono. L'ultima fase, l'integration stage, rappresenta il momento in cui si compie l'integrazione totale del denaro proveniente da attività criminose in un circuito economico legale, ottenendo così una "pulizia" del suddetto denaro per poterlo utilizzare in nuove operazioni. Il Fondo Monetario Internazionale[1] ha stabilito che l'ammontare di denaro che viene coinvolto in attività di riciclaggio oscilla tra 590 e 3200 miliardi di dollari che corrispondono circa ad una quota fra il 2 e il 5% del Prodotto Interno Lordo (PIL) globale. Tale dimensione mondiale dei fenomeni di riciclaggio e di finanziamento al terrorismo ha portato ad una serie di iniziative finalizzate ad elaborare misure di prevenzione e di contrasto dei fenomeni menzionati precedentemente. Nonostante la rilevanza di tali problematiche, al giorno d'oggi non esistono ancora strumenti e sistemi in grado di impedire definitivamente queste tipologie di attività criminali.

Il lavoro di tesi svolto è parte di un progetto destinato ad un cliente in ambito antiriciclaggio. In particolare, il progetto in questione si pone l'obiettivo di supportare i soggetti obbligati nell'esecuzione delle attività di contrasto al riciclaggio e di finanziamento al terrorismo, rendendo tali operazioni più efficaci, rapide ed efficienti.

[1] https://www.imf.org/external/index.htm

L’intento è quello di automatizzare attraverso tecniche di Machine Learning e di Natural Language Processing (NLP) parte dei processi che devono essere regolarmente svolti dai funzionari bancari e finanziari del settore antiriciclaggio. Il lavoro svolto durante lo stage è finalizzato all’automazione dell’attuale attività svolta dagli analisti bancari nell’identificare possibili clienti fraudolenti. Attualmente parte dell’identificazione viene svolta con l’obiettivo di ottenere informazioni mediante l’utilizzo di un motore di ricerca, come ad esempio Google, con parole chiavi (keywords) specifiche per trovare articoli pertinenti a frodi di riciclaggio. Questa tipologia di attività rientra nella fase di analisi reputazionale, una delle principali azioni volte a prevenire tentativi di riciclaggio del denaro. Essa ha come obiettivo quello di ricostruire l’identità digitale del soggetto indagando ad esempio sulla sua storia giudiziaria. L’automazione di queste attività di indagine e di ricerca significherebbe un grande risparmio in termini di tempo e costi ed una maggiore precisione nei risultati per gli addetti ai lavori degli istituti bancari e finanziari.

Nel primo capitolo vengono messe in luce le problematiche e le soluzioni attuali relativamente all’analisi reputazionale, presentando lo stato dell’arte. Nel secondo capitolo viene illustrato il framework proposto con una descrizione dei moduli principali realizzati e le relative funzionalità. Nel terzo capitolo viene affrontato il caso di studio analizzato, presentando i risultati ottenuti secondo specifiche metriche di valutazione. Nell’ultimo capitolo, il quarto, vengono tracciate le conclusioni e i possibili sviluppi futuri a partire dal lavoro svolto.

Capitolo 1

Analisi reputazionale: problematiche e soluzioni attuali

In questo capitolo viene affrontato il tema relativo all'analisi reputazionale mettendone in luce lo stato dell'arte e presentando le principali soluzioni tecnologiche esistenti al giorno d'oggi.

1.1 Descrizione del problema

L'analisi reputazionale rappresenta uno dei cardini nelle attività di contrasto al riciclaggio. L'obiettivo di tale attività è quello di conoscere meglio un ipotetico cliente, andando alla ricerca di ulteriori informazioni a partire da quelle ottenute dalla fase precedente, nota come "Know Your Customer". Essa ha come scopo quello di ottenere dati in merito al soggetto considerato. Un passo che contraddistingue l'analisi reputazionale è rappresentato dalle ricerche sul web basate su specifiche keywords di articoli che sono relativi al soggetto. Questa specifica attività di ricerca di notizie per mezzo di articoli di giornali online o di altri contenuti web relativamente al soggetto è nota anche come "Adverse Media Screening" o "Negative News Search". È stato stimato che in questi processi di verifica e analisi preliminari almeno il 50% delle operazioni sono completamente manuali, prevalentemente per motivi dovuti alla necessità di ricercare dati in svariate fonti e sistemi. Essendo processi manuali molto impegnativi è normale che si incorra in errori umani e si abbia perciò una ripercussione sulla qualità dei risultati. Oltre ad essere molto dispendioso in termini di energia e risorse, il processo di raccolta dati occupa circa l'80% del tempo degli analisti. Risulta dunque chiaro che l'automazione di tali operazioni porterebbe a notevoli ed evidenti vantaggi.

1.2 Stato dell'arte

Esistono attualmente diverse tecniche per affrontare il problema dell'analisi reputazionale e più in generale il problema della lotta al riciclaggio che rappresenta una delle principali

sfide nel settore bancario e finanziario. Tra queste il Data Mining rappresenta una tecnologia con grandi potenzialità in questo ambito. In particolare, diversi sono i sistemi che basano il loro funzionamento su tecniche di NLP. Tra questi sono pochi però i sistemi che elaborano le informazioni ricavate dal web, come ad esempio testi di articoli di giornali online. In generale, è possibile affermare che i progressi nel panorama dei sistemi di antiriciclaggio sono spesso contrastati dalla presenza di rigide regolamentazioni in ambito di privacy. Queste regolamentazioni limitano la quantità di Open Data e di dati condivisi e conseguentemente rendono difficoltosa la realizzazione dei suddetti sistemi. Oltre a questi ostacoli bisogna considerare anche le difficoltà del caso dovute alla globalizzazione e alla diffusione ed utilizzo sempre più rilevante di criptovalute. Quest'ultimi aspetti aprono ad una miriade di scenari difficili da prevedere e da contrastare.

Un esempio di sistema che rientra tra quelli che cercano di contrastare il reato di riciclaggio attraverso tecniche di NLP è il framework "Nextgen AML" *(Han, J., 2018) (Han, J., 2020)*. In "Nextgen AML" vengono utilizzate tecniche di NLP applicate su dati non strutturati ed eterogenei provenienti dal web. Queste tecniche vengono sfruttate in combinazione con sistemi di monitoraggio delle transazioni effettuate da un soggetto e dati provenienti da istituti bancari. L'obiettivo è quello di incrementare l'efficacia delle analisi effettuate dagli operatori andando a generare dei punteggi in base ai quali i soggetti vengono collocati automaticamente in classi di rischio. Il tutto viene corredato anche con sistemi di visualizzazione dei risultati. In particolar modo il framework in questione, a partire da contenuti testuali presenti sul web, come articoli di giornali online e tweet, esegue:

- Sentiment Analysis
- Named Enitity Recognition (NER)
- Relation Extraction (RE)
- Named Entity Linking (NEL)
- Link Analysis

L'architettura del framework è stata realizzata sulla base di diversi moduli che interagiscono tra di loro scambiandosi dati ed eseguendo le attività sopra menzionate.

Queste tipologie di analisi vengono eseguite con lo scopo di fornire dunque maggiori evidenze ai fini di una valutazione finale riguardo un soggetto da parte degli operatori preposti all'attività di indagine. In particolare, rispetto ai tradizionali sistemi di lotta al riciclaggio che si basano su una complicata analisi delle transazioni che vengono effettuate da un soggetto (dunque è richiesta una forte componente umana) e che conseguentemente generano un numero elevato di falsi positivi, "Nextgen AML" si propone di diminuire drasticamente i tempi e gli sforzi nella fase di validazione di segnalazioni ritenute falsi positivi.
Mediante l'utilizzo di questo framework è stato stimato un risparmio in termini di tempi e costi che si aggira intorno al 30% rispetto alle tradizionali attività di antiriciclaggio svolte manualmente.

Un altro framework implementato con l'obiettivo di identificare reati di riciclaggio attraverso tecniche di NLP è "Pluto" *(Chen, H. Y. et al., 2019)*. Il seguente framework in particolar modo fa utilizzo di:

- Text Processing
- Paragraph embeddings
- Clustering algorithm

Queste operazioni vengono eseguite su un insieme di articoli i quali vengono poi raggruppati in base a caratteristiche simili. Attualmente il framework supporta la sola lingua cinese e di conseguenza può essere utilizzato solo con articoli scritti in lingua cinese. Un futuro obiettivo degli autori del framework sarà quello di adattarlo anche ad altre lingue. I feedback riscontrati fino ad ora sono molto positivi, infatti è stato stimato che l'utilizzo di "Pluto" per l'identificazione di possibili articoli correlati ad un soggetto cercato permette una riduzione in termini di tempi e risorse di circa il 67%.

Un differente e interessante approccio è rappresentato dall'utilizzo delle Graph Convolutional Networks (GCN*) (Weber, M. et al., 2018)*. Si tratta di un metodo innovativo che fa uso di reti neurali convoluzionali pensate per lavorare su strutture a grafo. Nel monitoraggio delle transazioni, infatti, si possono immaginare dei grafi nei quali i vertici rappresentano un soggetto con particolari attributi e gli archi rappresentano le transazioni che vengono effettuate tra il soggetto in questione e le altre entità presenti.

Alcuni tentativi di identificare operazioni sospette sono stati fatti mettendo in combinazione invece informazioni relative ad un certo numero di transazioni eseguite e l'analisi del comportamento del soggetto. In particolare, *Perez e Lavalle (2011)* hanno realizzato un modello per l'identificazione di transazioni anomale. Il lavoro si basa in una prima fase sull'ottenimento e sull'analisi del comportamento di un soggetto in base alle transazioni da lui eseguite nel passato. A seguire, l'individuazione di comportamenti a rischio frode viene effettuata mettendo in comparazione le nuove transazioni effettuate dal cliente con il modello di comportamento precedentemente realizzato.

Un recente approccio è quello relativo alla Social Network Analysis (SNA*) (Colladon, A. F., & Remondi, E., 2017)*. Questa metodologia si concentra sull'analisi delle reti sociali, ovvero l'analisi di tutte le relazioni che si instaurano tra individui, enti ed organizzazioni. Alla base di questo recente approccio si pone ovviamente la teoria dei grafi che permette un'analisi approfondita delle reti che vengono generate, con l'obiettivo di identificare qualsiasi relazione che sia di allarme per potenziali attività criminali.

Altre tecniche, basate sul Machine Learning, che sono state adottate nel tempo per far fronte alla lotta del riciclaggio di denaro, in particolar modo relativamente al monitoraggio delle transazioni, sono ad esempio: attività di Clustering, Rule-based methods, Support Vector Machines (SVM), Decision Tree (DT) *(Salehi, A., et al., 2017)*.
Il clustering è un metodo di apprendimento automatico non supervisionato che consiste in un insieme di tecniche per raggruppare oggetti in classi omogenee in modo tale che le componenti di ciascun gruppo (cluster) presentino tra loro delle similarità e dissimilarità, invece, con componenti appartenenti ad altri cluster. Nell'ambito dell'antiriciclaggio il clustering viene utilizzato per raggruppare le transazioni effettuate con conti bancari in differenti cluster in base al livello di similarità. L'utilizzo di queste tecniche permette di identificare facilmente le transazioni che risultano essere sospette. Una delle sfide principali relative a questa metodologia è la grande quantità di dati che bisogna essere in grado di considerare e trattare.
I Rule-based methods sono considerati come metodi di classificazione predettivi. Viene predisposto un set di regole espresse mediante linguaggio logico. Questo approccio è già

largamente usato ma necessita di una grande quantità di dati. Inoltre, è possibile constatare che i metodi Ruled-based si adattano difficilmente ai nuovi comportamenti da parte dei criminali. Nell'ambito dell'antiriciclaggio la classificazione è relativa alle transazioni che vengono effettuate da un certo soggetto, le quali possono essere etichettate come potenzialmente pericolose o meno.
È inoltre possibile identificare un potenziale frodatore facendo utilizzo dei DT, ovvero un'altra tipologia di modelli di apprendimento supervisionato. La struttura definita da un DT è tale per cui i nodi foglia rappresentano le classificazioni e le ramificazioni l'insieme delle proprietà che determinano tali classificazioni.

1.3 Soluzioni tecnologiche

Sono diverse al giorno d'oggi le soluzioni tecnologiche che vengono utilizzate per contrastare i reati di riciclaggio. Attualmente tra quelle presenti in commercio che fanno utilizzo di tecniche di NLP si possono menzionare:

"Aylien Adverse Media Screening Service"[2]: attraverso tecniche di NLP viene eseguita l'attività di "Adverse Media Screening" in modo molto efficiente. Inoltre, viene garantita una notevole riduzione dei falsi positivi e l'identificazione di soggetti politicamente esposti. Vengono presi in considerazioni i contenuti web provenienti da 150 paesi, analizzando tutti i contenuti presenti, come ad esempio articoli di giornali online, in 16 differenti lingue. Tutti gli articoli vengono poi tradotti in lingua inglese. Infine, vengono raggruppati tutti gli articoli trovati sulla base di similarità riscontrate e ne viene fatta una selezione di quelli più utili ai fini della ricerca.

"WorkFusion Adverse Media Monitoring (Negative News)"[3]: l'utilizzo di questo sistema permette di diminuire di circa il 70% il lavoro manuale che generalmente viene svolto dagli operatori preposti alla ricerca. Altro punto forte del sistema "WorkFusion" è la riduzione dei falsi positivi, stimata circa del 95%. Viene eseguita un'analisi di tipo Sentiment e Risk Content in combinazione con ricerche

[2] https://aylien.com/adverse-media-screening/
[3] https://start.workfusion.com/use-case/adverse-media-monitoring/

basate su particolari keywords. Queste metodologie permettono all'operatore incaricato di fare la ricerca di comprendere in modo più efficiente e veloce i contenuti degli articoli selezionati in modo tale da prendere decisioni in tempi più rapidi.

"TransparINT"[4]: il seguente sistema per l'identificazione di notizie relative ad un soggetto cercato fa uso di tecniche di Machine Learning, di NLP, di Predictive Analysis e di Data Mining. Non sono richieste terze parti per il suo funzionamento. Il sistema permette anche l'identificazione di soggetti politicamente esposti e monitora in tempo reale le news presenti sul web in modo tale di avere accesso ai contenuti più recenti.

"BlackSwan ELEMENT™ of Compliance"[5]: il sistema fa utilizzo di tecniche di NLP in combinazione con metodi di Deep Learning (DL). La grande quantità di dati non strutturati provenienti da contenuti web che vengono presi in considerazione dalle ricerche effettuate viene rappresentata per mezzo di grafi. L'analisi successiva dei grafi permette di ricavare interessanti informazioni e relazioni che possono in mettere in evidenza possibili situazioni che necessitano una più approfondita indagine.

[4] https://transparint.com/
[5] https://blackswantechnologies.ai/applications/element-of-compliance/

Capitolo 2

Framework proposto

In questo capitolo viene presentata la struttura del framework realizzato andando ad analizzare ciascuna componente separatamente. Per ognuna di queste vengono descritti i linguaggi di programmazione e le librerie associate.

2.1 Reputazione online: il caso di studio affrontato

Il framework che è stato realizzato è in grado di effettuare una selezione sul motore di ricerca Google di tutti gli articoli che in qualche modo fanno riferimento al soggetto considerato. Il soggetto considerato nell'ambito applicativo è rappresentato da un ipotetico cliente da intendersi nello scenario bancario. Tale ricerca viene svolta facendo utilizzo di keywords che vengono stabilite dall'operatore preposto all'attività di ricerca e analisi e che in questo caso corrispondono a "condannato", "condannata" e "misure cautelari". Ogni ricerca viene effettuata seguendo uno schema preciso, "*nome+cognome+keyword*", e deve prendere in considerazione le prime tre pagine di ricerca di Google che equivalgono a trenta siti di diversa natura, come articoli di giornale e post su blog. I risultati ottenuti dovranno dunque evidenziare tutti gli articoli o contenuti accessibili via web che possono precisare o alludere alla natura giudiziaria del soggetto preso in considerazione nella ricerca.

2.2 Soluzione proposta

Figura 1: Workflow

Il framework che è stato implementato è costituito da diversi moduli che dialogano tra di loro. Il linguaggio di programmazione che è stato utilizzato è Python[6], un linguaggio di

[6] https://www.python.org/psf/

programmazione ad alto livello che supporta differenti paradigmi come quello Object-Oriented, quello imperativo e quello funzionale. Esso è definito anche come linguaggio di scripting. Python risulta essere dunque un linguaggio che combina una grande potenza con una sintassi molta chiara al tempo stesso.

La prima fase consiste nell'effettuare lo "scraping" delle pagine Google relative alle ricerche effettuate. Lo scraping ha come obiettivo quello di ottenere l'HTML (HyperText Markup Language) delle pagine risultanti dalle ricerche. Per ogni ricerca "*nome+cognome+keyword*" sono considerati i primi trenta risultati, ovvero le prime tre pagine di Google. A seguire il codice sorgente di ciascuna pagina viene analizzato per estrarre informazioni relativamente agli articoli presenti, come ad esempio il titolo e l'URL (Uniform Locator Resource).
A partire da ciascuno degli URL ottenuti viene scaricato successivamente il testo degli articoli facendo utilizzo di particolari librerie di Pyhton, specifiche per l'estrazione e il parsing di testi di articoli. Una volta ottenuti i testi di tutti gli articoli emersi dalle ricerche Google ne viene effettuata l'analisi sintattica e vengono generati degli alberi sintattici. Sulla base di quest'ultimi vengono individuati i cammini tra keywords e si procede ad un'accurata selezione degli articoli più pertinenti. Come risultato finale il framework viene presentato mediante un'interfaccia grafica minimale per rendere migliore l'interazione con l'utente finale incaricato di eseguire la ricerca.

2.2.1 Google Scraping

In questa prima fase viene estratto il codice sorgente delle pagine di ricerca Google. Le ricerche effettuate sono:

- *Nome+cognome+condannato*
- *Nome+cognome+condannata*
- *Nome+cognome+misure cautelari*

Bisogna tener presente che la ricerca che viene effettuata da Google considera di default degli AND, come relazioni logiche tra i termini sopra indicati.
Per fare in modo di considerare le prime tre pagine di risultati Google per ciascuna delle ricerche sopra indicate si agisce sulla query "string", ovvero la porzione di URL che è

posta dopo il carattere "?". In particolare, viene settato il parametro "start" con i valori 0, 10 e 20 rispettivamente per la prima, seconda e terza pagina di risultati.

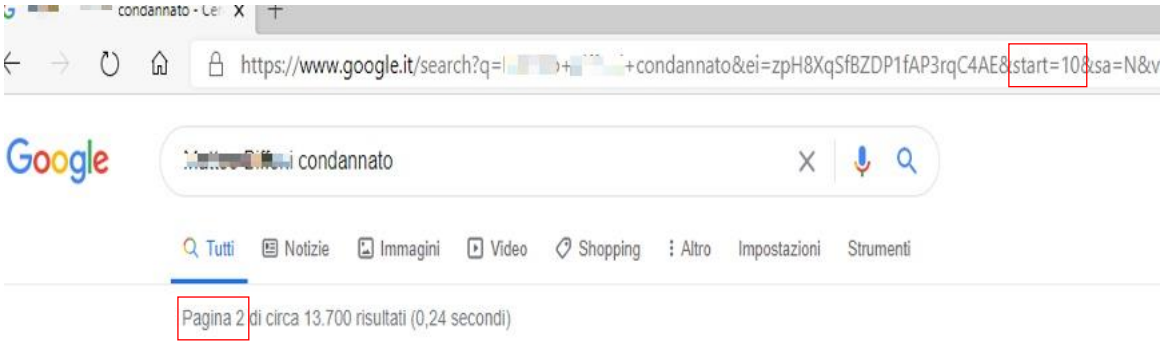

Figura 2: Query string

Il codice sorgente delle pagine di ricerca viene estratto utilizzando la libreria Requests[7], una libreria HTTP con licenza Apache2 che permette di effettuare richieste HTTP/1.1 in maniera molto semplice. Il codice HTML ottenuto è successivamente processato facendo uso della libreria Beautiful Soup[8]. La seguente libreria permette di convertire le pagine HTML ed XML in strutture composte da tag, elementi, attributi e valori rendendole delle strutture navigabili. Nella fattispecie, considerando le risorse HTML, è possibile ricercare elementi che abbiano determinate caratteristiche (tag, classi, id) nonché navigare nel DOM (Document Object Model) dei suddetti documenti HTML.

Per ogni pagina, effettuando una navigazione all'interno della struttura ad albero del codice HTML, vengono selezionati:

- Titoli degli articoli
- URL degli articoli
- Date di pubblicazione degli articoli se presenti
- Date di scaricamento degli articoli da parte dell'operatore che sta effettuando la ricerca
- Indice della pagina di Google

[7] https://requests.readthedocs.io/en/master/
[8] https://www.crummy.com/software/BeautifulSoup/bs4/doc/

Relativamente alla data di scaricamento degli articoli da parte dell'operatore e all'URL si tratta di informazioni necessarie che la Banca d'Italia richiede in maniera specifica per poter considerare una Segnalazione di Operazione Sospetta (SOS).

Figura 3: Esempio ricerca su Google

Inoltre, viene anche controllato se l'URL dell'articolo rimanda a file PDF (Portable Document Format). In questo caso gli articoli in questione vengono fin dall'inizio esclusi perché si potrebbero presentare dei problemi nelle fasi successive di scraping, soprattutto in termini di tempo e risorse, oltre che evidenti problemi di formattazione del testo nei PDF.

2.2.2 Estrazione del testo

A partire dagli URL degli articoli presenti nei risultati di Google ottenuti nella fase precedente si procede all'estrazione del testo. Questa operazione viene effettuata facendo

uso della libreria Newspaper3k[9], una libreria Python realizzata per l'estrazione e il parsing di testi di articoli di giornali presenti sul web. Mediante specifiche funzioni del modulo Newspaper3k è possibile ottenere il testo di un articolo partendo dal suo URL associato. Un limite di questa libreria è rappresentato dalla presenza in alcuni siti di paywall[10] che rendono impossibile scaricare il testo dell'articolo. Dopo aver utilizzato le funzioni di download e parsing della libreria NewsPaper3k è possibile ottenere il testo degli articoli considerati.

Figura 4: Esempio articolo estratto

2.2.3 Generazione di alberi sintattici mediante spaCy

Una volta ottenuti tutti i testi degli articoli risultanti dalle ricerche effettuate mediante Google è necessario selezionare tra questi solo gli articoli utili e pertinenti che possano supportare l'analisi di ricerca dell'operatore. Per valutare o meno la bontà di un articolo

[9] https://newspaper.readthedocs.io/en/latest/
[10] è una barriera di pagamento digitale, che gli editori utilizzano per alcuni tipi di servizi offerti online riservati ai soli utenti abbonati

ai fini dei risultati, per ogni testo associato è stata effettuata l'analisi sintattica, la quale rappresenta una delle prime fasi nel Text Mining e risulta essere molto utile nell'elaborazione del linguaggio naturale del testo. L'analisi sintattica, più in generale, ha come obiettivo quello di riconoscere una frase e assegnare ad essa una struttura sintattica. Il risultato di questa analisi viene visualizzato mediante un albero sintattico (anche conosciuto come "Abstract Syntax Tree") che mette in luce i vari legami che ci sono tra le parole che costituiscono la frase. Esistono due tipologie principali di analisi sintattica che si differenziano per il tipo di grammatica su cui si basano. Da una parte vi è l'analisi sintattica basata sulla "Phrase Structure Grammar", il cui albero sintattico si basa sul formalismo delle grammatiche libere dal contesto *(Daniel Jurafsky & James H. Martin, 2019)*. Dall'altra, si può individuare l'analisi sintattica basata sulla "Dependency Grammar", la quale si basa sulle relazioni di dipendenza che vi sono tra le parole che costituiscono la frase *(Daniel Jurafsky & James H. Martin, 2019)*. L'utilizzo di una grammatica di questa tipologia permette una migliore gestione delle lingue ricche dal punto di vista morfologico e che possiedono un ordinamento libero delle parole. Per effettuare l'analisi sintattica dei testi che sono stati ricavati viene utilizzato spaCy[11] *(Yuli Vasiliev, 2020)*, una libreria open source per l'elaborazione del linguaggio naturale scritta in Python e Cython [12]che offre supporto anche per la lingua italiana. La tipologia di analisi sintattica che spaCy esegue si basa su una grammatica di tipo "Dependency". L'albero che viene generato dalla seguente analisi sintattica risulta essere un grafo $G = (V, E)$ dove i vertici V rappresentano le parole che costituiscono la frase e gli archi E rappresentano i legami tra le varie parole. Il nucleo strutturale della frase è il verbo e ad esso, tramite cammini diretti o indiretti denominati dipendenze, si collegano le altre parole che compongono la frase.

Per procedere con l'analisi sintattica del testo e la conseguente generazione di alberi sintattici è necessario utilizzare la funzione di spaCy "*nlp"* che permette di processare il testo in vari step.

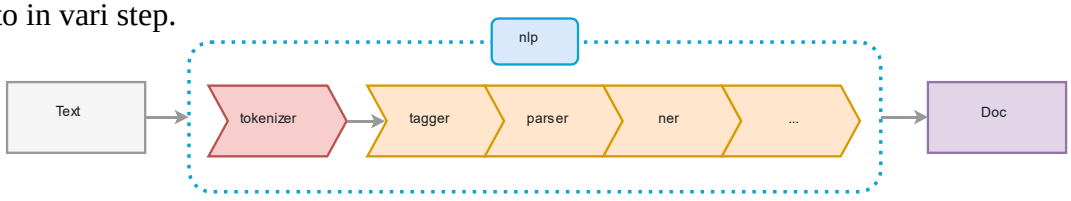

Figura 5: spaCy pipeline

[11] https://spacy.io/
[12] https://cython.org/

In particolar modo il testo viene processato e viene restituito un oggetto "Doc" che contiene tutte le informazioni relative ai tokens, incluse le loro caratteristiche e relazioni sintattiche. Un token rappresenta l'unità minima che costituisce un testo espresso in linguaggio naturale e può essere dunque una parola, un simbolo di punteggiatura o uno spazio.

Una volta che sono state eseguite le seguenti operazioni preliminari mediante spaCy, facendo uso della libreria NetworkX[13] *(Aric A. Hagberg et al., 2008),* una libreria Python, viene creato l'albero sintattico relativo al testo dell'articolo applicando la funzione "*Graph"*.

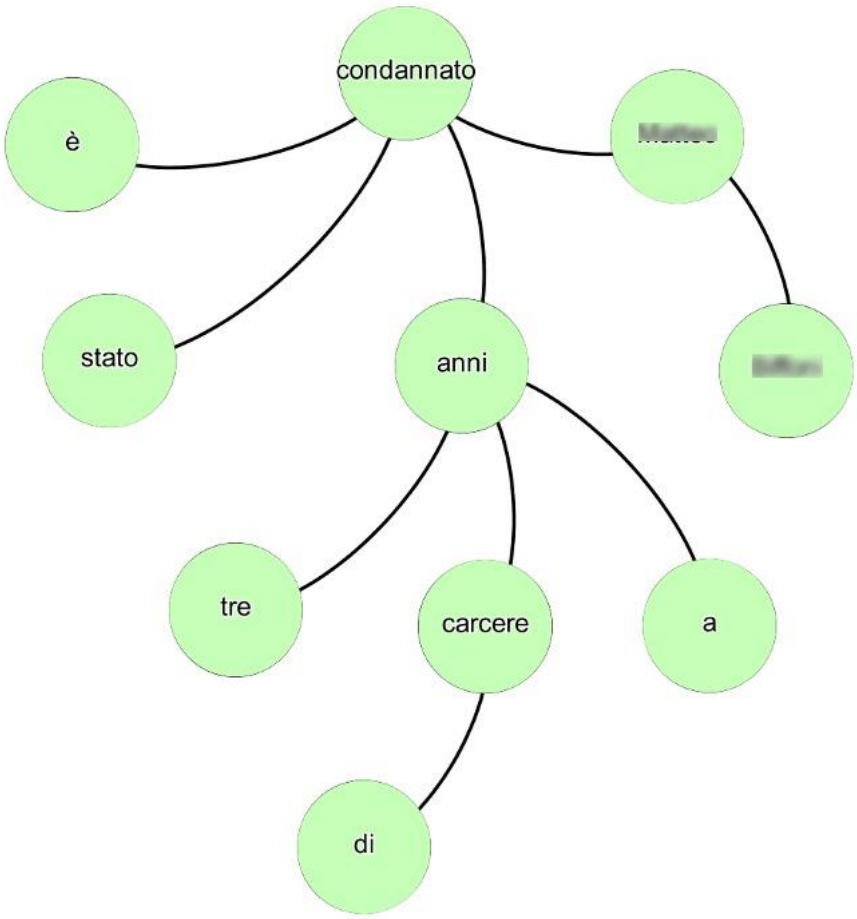

Figura 6: Albero sintattico di una frase facente parte di un testo estratto

[13] https://networkx.github.io/documentation/networkx-1.10/index.html

2.2.4 Generazione di alberi sintattici mediante Stanza

Durante la realizzazione del framework è stato utilizzato anche un DP differente da quello di spaCy. Il DP in questione è quello di Stanza[14] *(Chourdakis, E.T & Reiss, J.D, 2018)*. Quest'ultimo, rispetto a spaCy, è più moderno ed efficiente. Si riescono ad ottenere migliori risultati, in particolar modo in termini di accuratezza, ma nel caso di testi di grandi dimensioni si registrano delle prestazioni inferiori in termini di tempi. Infatti, in caso di lunghi testi viene consigliato il DP di spaCy. Stanza è dunque una libreria Python che presenta funzionalità analoghe a quelle di spaCy. Stanza offre inoltre un supporto multilingua migliore, infatti le lingue riconosciute sono più di 60. Mediante la funzione *"nlp"* il testo viene processato e a seguire mediante la libreria NetworkX ne viene generato l'albero sintattico.

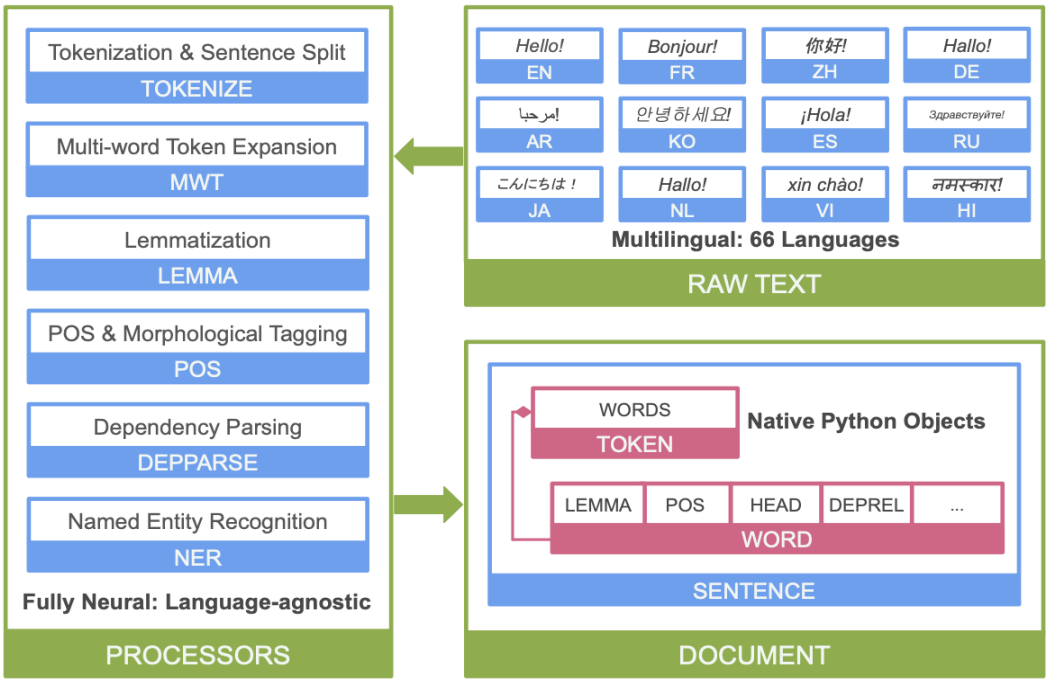

Figura 7: Stanza pipeline

2.2.5 Identificazione di cammini tra keywords

Creato per ciascun articolo il relativo albero sintattico (mediante il DP di spaCy o in alternativa di Stanza) è possibile procedere nell'analisi vera e proprio del testo. L'obiettivo è quello di identificare e selezionare solo gli articoli che riguardano il soggetto

[14] https://stanfordnlp.github.io/stanza/

considerato e che comprendono all'interno di essi alcuni dei vocaboli facenti parti del dizionario fornito. Quest'ultimi vocaboli sono ad esempio:

- Reato
- Arresti domiciliari
- Indagato
- Tribunale
- Accusato
- Condanna

Mediante alcune funzioni della libreria NetworkX, "*all_shortest_paths"* e "*shortest_path_length"*, vengono individuati tutti i possibili cammini minimi che vi sono tra il nome o il cognome del soggetto e una delle keywords facenti parte del dizionario. Se vi è almeno un cammino allora ne viene valutata la sua lunghezza. In base alle sperimentazioni effettuate inizialmente nel caso di cammini molto lunghi vi è il rischio di considerare dei risultati che si allontanano dagli obiettivi di ricerca. Per questo motivo vengono considerati solo i cammini la cui lunghezza è inferiore o uguale a 6. L'algoritmo utilizzato dalla libreria NetworkX per il calcolo delle lunghezze dei cammini è quello di Dijkstra *(Dijkstra, E. W, 1959).*

Affinché un articolo venga considerato come un risultato utile, deve contenere al suo interno almeno un cammino di lunghezza inferiore o uguale a 6 tra il nome del soggetto e una delle keywords e un cammino con gli stessi requisiti analogamente per il cognome. Inoltre, come ulteriore match di conferma, viene controllato se all'interno del testo estratto dell'articolo compare lo schema "*nome+cognome*" del soggetto considerato.

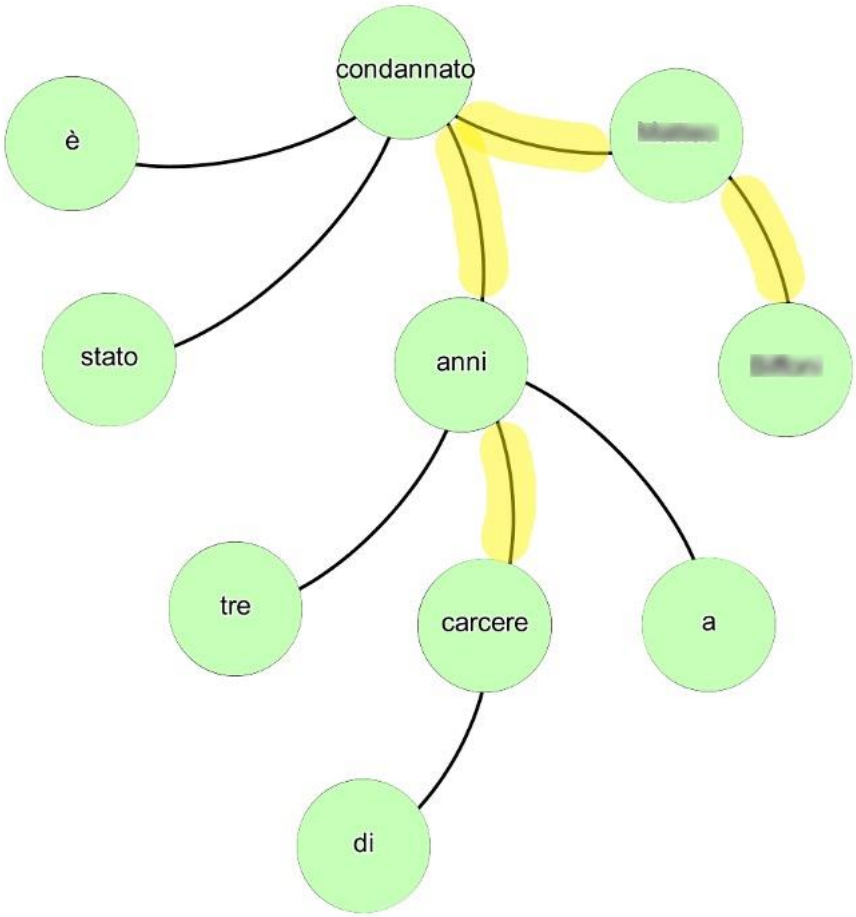

Figura 8: Cammini tra keywords individuati all'interno di una frase facente parte di un testo

2.2.6 Estrazione di articoli utili

Per ogni articolo estratto vengono dunque memorizzati:

- Titolo dell'articolo
- URL dell'articolo
- Data di scaricamento dell'articolo
- Data di pubblicazione dell'articolo
- Indice della pagina di Google
- Frasi utili estratte

Relativamente all'ultimo punto, per ogni articolo che viene considerato interessante ai fini della ricerca effettuata, viene eseguita una selezione delle sole frasi che mettono in luce in modo immediato e chiaro i punti chiave dell'articolo. Le frasi estratte che vengono memorizzate sono frasi dell'articolo che contengono contemporaneamente il nome e il

cognome del soggetto e una delle keywords che fanno parte del dizionario sopra menzionato. Queste frasi possono risultare utili all'operatore addetto alla ricerca che può avere così un'idea generale del contenuto dell'articolo che è stato selezionato.
Un esempio di frase estratta è: "*Citazione a giudizio per M****o B*****i, sindaco di P***o*". In questa frase compare il nome e il cognome del soggetto ricercato ("*M****o*", "*B*****i*") e una delle keywords appartenenti al dizionario fornito, ovvero "Citazione a giudizio".
Inoltre, gli articoli ritenuti utili ai fini della ricerca vengono ordinati sulla base dell'indice di pagina di risultati di Google. In questo modo gli articoli che sono indicizzati nelle prime pagine di Google, che conseguentemente dovrebbero avere una maggiore pertinenza, appaiono per primi nell'output visibile all'operatore.

2.2.7 Interfaccia grafica

Per rendere migliore l'esperienza d'uso per l'operatore è stata realizzata un'interfaccia grafica per il framework prodotto che ha come obiettivo quello di prendere in input i dati del soggetto su cui si vuole effettuare la ricerca e di dare in output i risultati ottenuti, evidenziando tutti gli articoli presenti su Google che possono avere un collegamento con la persona oggetto della ricerca.
Per la realizzazione di tale interfaccia è stata utilizzata PySimpleGUI[15], una libreria Pyhton che permette di creare in modo rapido e semplice interfacce grafiche.

Figura 9: Schermata "Inserimento dati"

[15] https://pysimplegui.readthedocs.io/en/latest/

Nella prima schermata dell'interfaccia l'operatore è invitato ad inserire il nome, il cognome e il codice fiscale del soggetto considerato. Sono stati implementati dei controlli per evitare che i campi non vengano compilati o nel caso in cui il codice fiscale non fosse valido. Per quest'ultimo tipo di controllo è stata utilizzata la libreria CodiceFiscale[16].

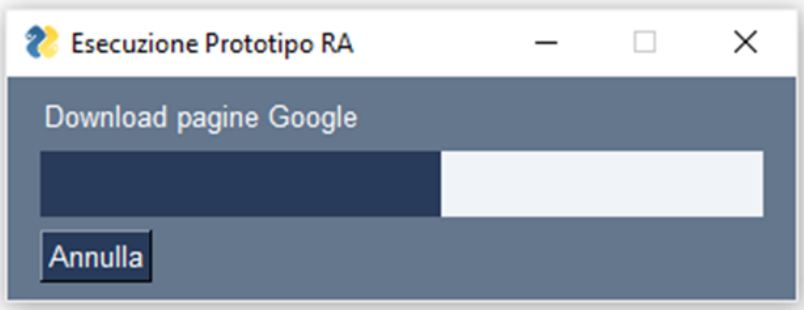

Figura 10: Schermata "Download pagine Google"

Nella successiva schermata viene scaricato il codice HTML delle pagine Google per ciascuna delle ricerche considerate. Viene mostrata una barra di progressione che indica lo stato di avanzamento dello scaricamento.

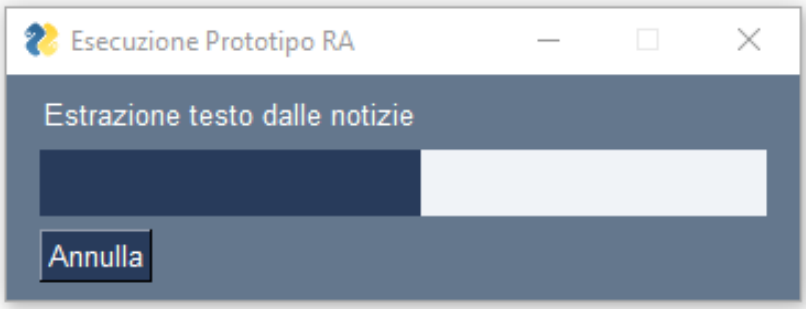

Figura 11: schermata "Estrazione testo dalle notizie"

In questa fase viene estratto per ciascun risultato trovato (articoli di giornali online) il relativo testo. Nell'esecuzione del framework è la fase che a livello computazionale richiede più risorse.

[16] https://github.com/fabiocaccamo/python-codicefiscale

Figura 12: Schermata "Selezione articoli utili per la verifica"

A questo punto, una volta che il testo è stato estratto per ciascuno degli articoli considerati, vengono individuati i cammini tra keywords a partire dagli alberi sintattici generati.

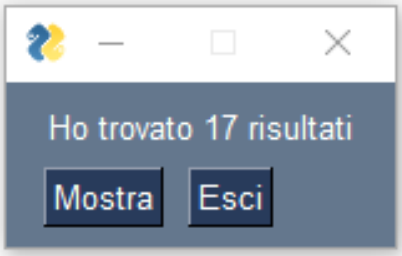

Figura 13: Schermata "Mostra risultati"

Prototipo RA
RISULTATO 1
Titolo: Fallimento del C***f, citato a giudizio M****o B*****i - gonews.it
Link*:** https://www.gonews.it/2018/09/18/fallimento-del-cf-citato-a-giudizio-m****o -b*****i /
Data di scaricamento: 12 lug 2020
Data di pubblicazione: 18 set 2018
Frasi estratte: 'Con B*****i citati a giudizio per bancarotta semplice ex amministratori del C***f ed ex membri del cda, L**a R********i, L***a C********i, L******o G****i,

V******a M****i, G*******o B***a, oltre ai tre componenti del collegio sindacale, M*****o L******i, M*****o P****i e M***o B**i.' 'Paradossali le dichiarazioni del Sindaco B*****i, che anche di fronte al rinvio a giudizio sostiene di aver trovato in eredità dalle precedenti amministrazioni questa situazione.' 'Citazione a giudizio per M****o B*****i, sindaco di P***o.'
RISULTATO 2
Titolo: C***f, il giudice respinge la richiesta del legale di B*****i: il ...
Link*:** https://www.tvpo.it/2020/03/creaf-il-giudice-respinge-la-richiesta-del-legale-di-b*****i -il-processo-va-avanti/
Data di scaricamento: 12 lug 2020
Data di pubblicazione: 6 mar 2020
Frasi estratte: C***f, il giudice respinge la richiesta del legale di B*****i: il ...

Figura 14: Una porzione della schermata dei risultati finali

In questa ultima schermata vengono presentati tutti gli articoli che sono stati selezionati in quanto interessanti. I risultati sopra mostrati sono sotto forma di elenco “scrollable”. In questo modo l’operatore successivamente potrà verificare manualmente a partire dagli URL indicati i vari articoli che sono stati selezionati.

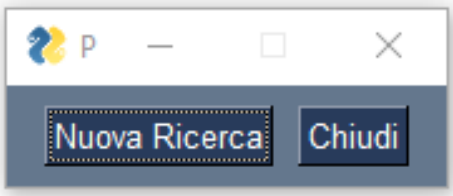

Figura 15: Schermata "Nuova ricerca"

Una volta terminata la revisione degli articoli che sono stati ritenuti utili per l'analisi di una determinata persona, l'operatore può procedere ad una nuova ricerca oppure terminare il tutto.

Capitolo 3

Analisi sperimentale

Nella fase di analisi sperimentale sono stati considerati 5 casi di studio sui quali sono state analizzate le performance del framework realizzato. In particolare, relativamente ad ogni caso di studio, è stata fatta un'analisi di efficienza ed efficacia del framework proposto, in modo tale da trarne dei risultati in termini numerici quantitativi mettendo in evidenza le sue criticità e le sue limitazioni del caso.

3.1 Casi di studio

Per i casi di studio considerati sono state prese in considerazione delle persone con precedenti penali, come ad esempio usura, bancarotta, truffa. In particolare, le persone che sono state considerate per i casi di studio sono:

- C*****a S***o: la donna residente nel sud Italia è stata condannata nel 2010 per usura, tentata estorsione e spaccio
- G*****o P****o: nel 2018 l'uomo, residente nel nord Italia, è stato condannato per bancarotta
- M****o B*****i: è il sindaco di una città toscana, nel 2018 è stato indagato per bancarotta
- Y******* L*: è un imprenditore asiatico che nel 2017 ha avuto affari economici in Italia ed è stato successivamente indagato per falso in bilancio
- D****o C*****a: l'uomo, un imprenditore italiano, ha ricevuto nel 2020 la conferma di condanna per bancarotta

I primi due soggetti sono persone comuni residenti in aree geografiche differenti. Il terzo caso di studio rappresenta, invece, una PEP, una persona dunque con una certa notorietà. Inoltre, il caso di studio relativo al sindaco considerato è interessante perché le ricerche Google mostrano diversi articoli in cui si evidenzia che il primo cittadino è stato indagato per bancarotta ma al tempo stesso emergono altrettanti articoli in cui è lo stesso sindaco che condanna dei particolari episodi. I restanti due casi di studio invece rappresentano

due imprenditori di nazionalità diversa. Infine, è possibile osservare che i 5 casi di studio presi in considerazione si riferiscono a soggetti indagati o condannati in periodi temporali diversi.

3.2 Misure di valutazione delle performance

Per misurare le performance del framework realizzato sono stati analizzati i risultati sia da un punto di vista di efficienza che di efficacia.
Relativamente alla valutazione dell'efficienza del framework proposto sono stati analizzati i tempi di esecuzione usando i due DP adottati, rispettivamente spaCy e Stanza. Le suddette analisi hanno permesso inoltre di stabilire la lunghezza minima ottimale dei cammini da ricercare tra keywords all'interno del testo in modo tale da trovare risultati utili all'identificazione di un possibile frodatore.
Per quanto riguarda l'efficacia sono stati osservati effettivamente quanti articoli utili ai fini della ricerca sono stati selezionati nell'output sul numero totale di articoli estratti. Il numero totale di articoli estratti è rappresentato da tutti gli articoli che vengono ricavati dalle tre ricerche incrociate su Google andando ad eliminare i possibili duplicati che si generano. Come metriche sono state utilizzate l'Accuracy (accuratezza), la Precision (precisione), la Recall (richiamo) e la F-score (punteggio f).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

L'Accuracy indica l'accuratezza del modello.

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

La Precision indica l'abilità del modello di non etichettare un'istanza positiva che in realtà è negativa. È dunque il rapporto tra veri positivi e la somma di veri positivi e falsi positivi.

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

La Recall, invece, indica l'abilità del modello di trovare tutte le istanze positive. È definita come il rapporto tra i veri positivi e la somma dei veri positivi e dei falsi negativi.

$$F - score = 2 \times \frac{Recall \times Precision}{Recall + Precision}$$

Questa metrica, la F-score, rappresenta una media armonica delle metriche Precision e Recall.

3.3 Analisi dei risultati

Facendo uso del DP di spaCy e di Stanza, sono stati calcolati i risultati dell'esecuzione del framework per i casi di studio proposti al variare della lunghezza minima dei cammini da ricercare all'interno dei testi tra specifiche keywords. Facendo variare la lunghezza minima dei cammini da 3 a 10 è stato possibile constatare che 6 rappresenta il valore ottimale da assegnare alla lunghezza minima dei cammini da ricercare. Oltre il valore 6 i risultati utili, per i casi di studio considerati, rimangono costanti (la tabella completa è riportata nell'*Appendice A*). Per valori inferiori a 6, invece, si ha una graduale diminuzione del numero di articoli estratti. Nelle tabelle sotto riportate "Numero articoli estratti" fa riferimento al numero degli articoli che sono mostrati in output all'operatore preposto alla ricerca. Il "Numero articoli ritenuti utili", invece, rappresenta il numero effettivo di articoli che fanno riferimento al soggetto considerato selezionati manualmente dall'operatore. Dalle tabelle è inoltre possibile evincere che utilizzando il DP di spaCy, come previsto, si ottengono risultati migliori in termini di tempo, pur trattandosi di differenze che non superano mai i 50 secondi. Tendenzialmente il tempo richiesto per effettuare una ricerca in merito ad un soggetto oscilla tra i 2 e i 3 minuti.

Tabella 1: Valutazione tempistiche e lunghezza cammino minimo ottimale per il caso di studio 1

Soggetto	***DP***	***Numero articoli estratti***	***Numero articoli ritenuti utili***	***Tempo medio (min)***	***Lunghezza cammino minimo***
M****o B*****i	spaCy	9	2	2,2	3
		13	2		4
		17	2		5
		17	**2**		**6**
	Stanza	11	2	2,5	3
		15	2		4

		16	2		5
		17	**2**		**6**

Tabella 2: Valutazione tempistiche e lunghezza cammino minimo ottimale per il caso di studio 2

Soggetto	***DP***	***Numero articoli estratti***	***Numero articoli ritenuti utili***	***Tempo medio (min)***	***Lunghezza cammino minimo***
G*****o P****o	spaCy	8	8	3,3	3
		9	9		4
		10	10		5
		10	**10**		**6**
	Stanza	8	8	3,5	3
		10	10		4
		10	10		5
		10	**10**		**6**

Tabella 3: Valutazione tempistiche e lunghezza cammino minimo ottimale per il caso di studio 3

Soggetto	***DP***	***Numero articoli estratti***	***Numero articoli ritenuti utili***	***Tempo medio (min)***	***Lunghezza cammino minimo***
C*****a S***o	spaCy	0	0	2,5	3
		1	1		4
		1	1		5
		1	**1**		**6**
	Stanza	1	1	3	3
		1	1		4
		1	1		5
		1	**1**		**6**

Tabella 4: Valutazione tempistiche e lunghezza cammino minimo ottimale per il caso di studio 4

Soggetto	***DP***	***Numero articoli estratti***	***Numero articoli ritenuti utili***	***Tempo medio (min)***	***Lunghezza cammino minimo***
Y******g L*	spaCy	12	10	2	3
		17	13		4
		19	13		5
		19	**13**		**6**
	Stanza	11	10	2,2	3
		18	13		4
		18	13		5
		19	**13**		**6**

Tabella 5:Valutazione tempistiche e lunghezza cammino minimo ottimale per il caso di studio 5

Soggetto	***DP***	***Numero articoli estratti***	***Numero articoli ritenuti utili***	***Tempo medio (min)***	***Lunghezza cammino minimo***
D****o C*****a	spaCy	32	32	2,1	3
		35	35		4
		37	37		5
		38	**38**		**6**
	Stanza	33	33	2,4	3
		36	36		4
		36	36		5
		38	**38**		**6**

Per valutare l'efficacia del framework per ogni ricerca effettuata e per ciascun soggetto sono state analizzate specifiche metriche, quali Accuracy, Precision, Recall e F-score. Le metriche sono state calcolate a partire dal numero di articoli scaricati a seguito delle ricerche effettuate su Google. Il seguente numero non tiene conto dei duplicati, motivo

per cui non assume mai il valore 90 come, a livello teorico, invece dovrebbe accadere (per ogni soggetto vengono effettuate 3 ricerche Google considerando per ciascuna di esse i primi 30 risultati). I risultati sono stati calcolati sia per spaCy sia per Stanza e considerando il valore del cammino minimo pari a 6.

Tabella 6: Valutazione efficacia risultati

Caso di studio	***DP***	***Numero risultati ricerche Google***	***Accuracy***	***Precision***	***Recall***	***F-score***
1	spaCy	63	0,730	0,117	0,500	0,190
	Stanza	63	0,730	0,117	0,500	0,190
2	spaCy	63	0,984	1	0,909	0,952
	Stanza	63	0,984	1	0,909	0,952
3	spaCy	74	0,959	1	0,250	0,400
	Stanza	74	0,959	1	0,250	0,400
4	spaCy	59	0,694	0,684	0,520	0,590
	Stanza	59	0,694	0,684	0,520	0,590
5	spaCy	68	0,823	1	0,760	0,863
	Stanza	68	0,823	1	0,760	0,863

In conclusione, analizzando nello specifico i risultati ottenuti è possibile constatare che per ogni ricerca effettuata vengono correttamente evidenziati diversi articoli presenti sul motore di ricerca Google che fanno riferimento al soggetto in questione. Mediante la ricerca dei cammini minimi tra keywords è possibile ottenere anche delle frasi facenti parte del testo dell'articolo che estrapolano i punti chiavi dello stesso in modo chiaro ed efficiente. È stato notato che per alcuni risultati non è possibile estrarre queste tipologie di frasi oppure lo scaricamento della data di pubblicazione dell'articolo non è possibile dal momento in cui non viene indicata. Nel complesso gli articoli vengono selezionati

con una discreta precisione e vengono corredati di utili informazioni ai fini dell’analisi reputazionale relativamente al soggetto considerato. Bisogna infine considerare che il componente, così come un umano, non può essere perfetto. Dovrà quindi essere sottoposto a revisione finale manuale da parte di un utente fisico, che confermi o meno l'attinenza degli articoli al soggetto e al contesto.

Capitolo 4

Conclusioni e sviluppi futuri

L'obiettivo fissato all'inizio del lavoro di stage era quello di andare a creare un framework che potesse replicare in modo automatico parte dei lavori che normalmente vengono svolti da analisti bancari manualmente. Diverse persone hanno manifestato grande entusiasmo nonostante la presenza di alcuni aspetti ancora da migliorare. Sarà ora compito dell'azienda garantire e mantenere questa percezione positiva mostrata rispetto al prodotto e procedere al tempo stesso gradualmente nel suo miglioramento per poter soddisfare a pieno le richieste e le esigenze pervenute.

Come sviluppi futuri per avere una maggiore precisione nei risultati ottenuti si potrebbe pensare di utilizzare un sistema basato su clausole. In particolare, un noto sistema appartenente a questa tipologia è ClausIEpy [17]che risulta essere un'implementazione in Python di ClasusIE *(Del Corro, L., & Gemulla, R., 2013)* che fa uso di spaCy. ClausIE è un approccio basato su clausole relativamente al problema dell'Open Information Extraction (OIE), il cui obiettivo risulta essere quello di estrarre informazioni strutturate a partire da testi. L'idea sarebbe quella di estrarre informazioni utili sotto forma di clausole a partire dai testi degli articoli. Quest'ultime clausole potrebbero poi rappresentare un ulteriore controllo per confermare una possibile corrispondenza con il soggetto cercato. La problematica principale relativamente a ClausIEpy è il supporto alla lingua italiana. Esso nasce come un sistema basato su clausole da utilizzarsi con testi in lingua inglese. Il tentativo che è stato fatto è stato quello di tradurre manualmente in lingua italiana una porzione della libreria di ClausIEpy. Tuttavia, i risultati ottenuti non sono stati abbastanza soddisfacenti in termini di efficacia. Uno sviluppo futuro sarebbe dunque quello di considerare un sistema basato su clausole, simile a ClausIEpy, ma con il supporto anche per la lingua italiana.
Un altro miglioramento che potrebbe essere fatto è relativo alla parte di NER. La NER ha come obiettivo quello di estrarre informazioni da un testo classificando ogni singolo

[17] https://github.com/mmxgn/spacy-clausie

elemento presente all'interno di esso in categorie predefinite, come persone, organizzazioni e luoghi. Facendo uso di tecniche di NER si potrebbe implementare un ulteriore controllo all'interno del testo dell'articolo alla ricerca di elementi come il luogo di nascita del soggetto ricercato che porterebbero ad un maggior raffinamento dei risultati. In alcuni casi potrebbero essere considerati dei risultati ambigui dovuti alla presenza di frasi attive e passive che vengono individuate all'interno del testo. Discriminare i risultati sulla base della tipologia di frase estratta porterebbe ad escludere tutti i casi in cui ad esempio è il soggetto da noi considerato che "condanna" un'altra persona o un fatto. Per perseguire questo scopo si potrebbe fare uso del DP di Stanza e andare ad analizzare per ogni componente della frase il suo ruolo e il suo legame con le altre componenti della porzione di testo considerata.

Infine, l'applicativo realizzato potrebbe essere correlato di funzioni di supporto multilingua per avere una maggiore usabilità non limitandosi a ricerche basate sulla lingua italiana. Questa esigenza sorge dal momento in cui il sistema di valutazione di frode di riciclaggio di denaro dovrebbe basarsi a livello europeo sulla medesima regolamentazione per garantire una maggiore cooperazione e una maggiore trasparenza tra gli stati membri. Si tratterebbe dunque di implementare il framework in modo tale da supportare ricerche anche su siti web stranieri per poter identificare correttamente un potenziale frodatore presente in uno specifico paese.

Bibliografia

Aric A. Hagberg, Daniel A. Schult and Pieter J. Swart, *"Exploring network structure, dynamics, and function using NetworkX"*, in Proceedings of the 7th Python in Science Conference (SciPy2008), Gäel Varoquaux, Travis Vaught, and Jarrod Millman (Eds), (Pasadena, CA USA), pp. 11–15, Aug 2008

Chen, H. Y., Zou, S. X., & Sung, C. L. (2019). Pluto: A deep learning based watchdog for anti money laundering. In *Proceedings of the First Workshop on Financial Technology and Natural Language Processing* (pp. 93-95).

Chourdakis, E.T and Reiss, J.D. (2018) Grammar Informed Sound Effect Retrieval for Soundscape Generation. In DMRN+ 13: Digital Music Research Network One-day Workshop.Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton and Christopher D. Manning. 2020. *Stanza: A Python Natural Language Processing Toolkit for Many Human Languages.* In Association for Computational Linguistics (ACL) System Demonstrations. 2020.

Colladon, A. F., & Remondi, E. (2017). Using social network analysis to prevent money laundering. *Expert Systems with Applications*, *67*, 49-58.

Daniel Jurafsky & James H. Martin (2019). Constituency Parsing, *Speech and Language Processing*, pp. 232-244

Daniel Jurafsky & James H. Martin (2019). Dependency Parsing, *Speech and Language Processing*, pp. 273-297

Del Corro, L., & Gemulla, R. (2013, May). Clausie: clause-based open information extraction. In *Proceedings of the 22nd international conference on World Wide Web* (pp. 355-366).

Dijkstra, E. W. (1959). *A note on two problems in connexion with graphs*:(numerische mathematik, _1 (1959), p 269-271).

Han, J., Barman, U., Hayes, J., Du, J., Burgin, E., & Wan, D. (2018, July). Nextgen aml: Distributed deep learning based language technologies to augment anti money laundering investigation. In *Proceedings of ACL 2018, System Demonstrations* (pp. 37-42).

Han, J., Huang, Y., Liu, S., & Towey, K. (2020). Artificial intelligence for anti-money laundering: a review and extension. *Digital Finance*, 1-29.

Honnibal, M., & Montani, I. (2017). *spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.*

Pérez, D. G., & Lavalle, M. M. (2011). Outlier detection applying an innovative user transaction modeling with automatic explanation. In *2011 IEEE Electronics, robotics and automotive mechanics conference (CERMA)* (pp. 41–46). IEEE.

Salehi, A., Ghazanfari, M., & Fathian, M. (2017). Data mining techniques for anti money laundering. *International Journal of Applied Engineering Research*, 12(20), 10084-10094.

Weber, M., Chen, J., Suzumura, T., Pareja, A., Ma, T., Kanezashi, H., ... & Schardl, T. B. (2018). Scalable graph learning for anti-money laundering: A first look. *arXiv preprint arXiv:1812.00076.*

Yuli Vasiliev (2020). *Natural Language Processing with Python and spaCy: A Practical Introduction*. No Starch Press

Appendice A

Tabella 7: Valutazione tempistiche e lunghezza cammino minimo ottimale per i 5 casi di studio considerati

Soggetto	***DP***	***Numero articoli estratti***	***Numero articoli utili***	***Tempo medio (min)***	***Lunghezza minimi cammino***
M****o B*****i	spaCy	9	2	2,2	3
		13	2		4
		17	2		5
		17	2		6
		17	2		7
		17	2		8
		17	2		9
		17	2		10
	Stanza	11	2	2,5	3
		15	2		4
		16	2		5
		17	2		6
		17	2		7
		17	2		8
		17	2		9
		17	2		10
G*****o P****o	spaCy	8	8	3,3	3
		9	9		4
		10	10		5
		10	10		6
		10	10		7
		10	10		8
		10	10		9
		10	10		10
	Stanza	8	8	3,5	3

		10	10		4
		10	10		5
		10	10		6
		10	10		7
		10	10		8
		10	10		9
		10	10		10
C*****a S***o	spaCy	0	0	2,5	3
		1	1		4
		1	1		5
		1	1		6
		1	1		7
		1	1		8
		1	1		9
		1	1		10
	Stanza	1	1	3	3
		1	1		4
		1	1		5
		1	1		6
		1	1		7
		1	1		8
		1	1		9
		1	1		10
Y******g L*	spaCy	12	10	2	3
		17	13		4
		19	13		5
		19	13		6
		19	13		7
		19	13		8
		19	13		9
		19	13		10

	Stanza	11	10	2,2	3
		18	13		4
		18	13		5
		19	13		6
		19	13		7
		19	13		8
		19	13		9
		19	13		10
D****o C*****a	spaCy	32	32	2,1	3
		35	35		4
		37	37		5
		38	38		6
		38	38		7
		38	38		8
		38	38		9
		38	38		10
	Stanza	33	33	2,4	3
		36	36		4
		36	36		5
		38	38		6
		38	38		7
		38	38		8
		38	38		9
		38	38		10