

**UNIVERSITÀ POLITECNICA DELLE MARCHE**  
**FACOLTÀ DI INGEGNERIA**  
Dipartimento di Ingegneria dell'Informazione  
Corso di Laurea Magistrale in Ingegneria Informatica e dell'Automazione

---



**TESI DI LAUREA**

**Definizione e progettazione di un framework per la gestione di documenti giudiziari**

**Defining and designing a framework for the management of court documents**

Relatore

Prof. Domenico Ursino

Correlatore

Dott. Francesco Cauteruccio

Candidata

Giulia Manoni

---

**ANNO ACCADEMICO 2021-2022**

*L'uomo deve perseverare nell'idea che l'incomprensibile sia comprensibile;  
altrimenti rinunciarebbe a cercare.*

J. W. Goethe, "Massime e riflessioni"

## Sommario

La digitalizzazione è uno degli strumenti più utilizzati per ottimizzare i processi interni di grandi aziende e uffici. In questo progetto di tesi la digitalizzazione è stata considerata in ambito giuridico. In particolare, essa è stata valutata nelle Corti di Appello nell'ambito del progetto Uni 4 Justice. Dopo aver presentato una panoramica di tale progetto, consideriamo la situazione reale della Corte di Appello di Ancona, fornendo dei suggerimenti in merito alla possibilità di applicare il progetto stesso. Successivamente viene progettato, implementato e testato un metodo per l'estrazione di informazioni da testi giuridici, tramite l'utilizzo di algoritmi di Data Science e, nello specifico, del Natural Language Processing.

**Keyword:** Natural Language Processing, Data Science, Sentenze, Note, Ontologia, Parser, Spacy, Neo4j, Corti di Appello.

<b>Introduzione</b>	<b>1</b>
<b>1 Il progetto Uni 4 Justice</b>	<b>4</b>
1.1 Introduzione	4
1.2 Rilevanza rispetto al contesto territoriale	5
1.3 Rilevanza e coerenza con la strategia nazionale per la Giustizia Digitale	7
1.4 Rilevanza e coerenza con le disposizioni normative tramite modelli organizzativi	8
1.5 Complementarietà con altri strumenti di programmazione rilevanti	11
1.6 Articolazioni e contenuti del progetto	14
1.6.1 Raccolta dei bisogni e delle eccellenze per l'UPP	14
1.6.2 Strumenti operativi di monitoraggio e consultazione per il potenziamento dell'UPP	14
1.6.3 Strumenti di formazione delle competenze per l'efficacia dell'UPP	15
1.6.4 Modelli organizzativi e dei processi per l'arretrato	16
1.6.5 Modelli di trasformazione digitale avanzata e delle interfacce HCI	16
1.6.6 Attivazione delle sperimentazioni dei modelli organizzativi, giuridici, tecnologici	17
1.6.7 Monitoraggio delle sperimentazioni dei modelli organizzativi, giuridici, tecnologici	17
1.6.8 Ridefinizione modelli formativi permanenti di eccellenza	17
1.6.9 Formazione alle competenze digitali e alla comunicazione attiva	17
1.6.10 Reclutamento dello staff	18
<b>2 Uno sguardo alla situazione sul campo</b>	<b>19</b>
2.1 Incontro con il Presidente della Corte di Appello di Ancona	19
2.1.1 Situazione informatica e digitale attuale	19
2.1.2 Esigenze di innovazione digitale	20
2.1.3 Contestualizzazione nel Progetto Uni 4 Justice	20
2.2 Gestione informatica della sezione civile	20
2.2.1 Sistema informativo: SICID	20
2.2.2 Problematiche e richieste del personale	20
2.3 Gestione informatica della sezione penale	21
2.3.1 Sistema informativo: SICP	21
2.3.2 Problematiche e richieste del personale	21

<b>3</b>	<b>Il sistema documentale: Sharepoint</b>	<b>24</b>
3.1	Panoramica . . . . .	24
3.1.1	Posizionamento competitivo . . . . .	24
3.1.2	Punti di forza . . . . .	26
3.2	Configurazione . . . . .	27
3.2.1	Online . . . . .	27
3.2.2	In locale . . . . .	27
3.3	Sito in SharePoint . . . . .	29
3.3.1	Ruolo amministratore . . . . .	33
3.3.2	Gesione e permessi dei membri del team . . . . .	33
3.3.3	Condivisione e manipolazione dei documenti . . . . .	33
3.4	La ricerca . . . . .	34
3.4.1	Campo di azione . . . . .	34
3.4.2	Personalizzazione . . . . .	35
<b>4</b>	<b>Introduzione ad Akoma Ntoso</b>	<b>38</b>
4.1	Akoma Ntoso . . . . .	38
4.1.1	Accesso libero . . . . .	39
4.1.2	Obiettivo . . . . .	40
4.1.3	Obiettivi strategici . . . . .	40
4.2	Architettura del documento . . . . .	41
4.2.1	Strutture di markup . . . . .	42
4.3	Schema . . . . .	42
4.3.1	Schema generale . . . . .	42
4.3.2	Schema personalizzato . . . . .	45
<b>5</b>	<b>Definizione dell'approccio proposto</b>	<b>46</b>
5.1	Il Natural Language Processing . . . . .	46
5.1.1	Aspetti fondamentali . . . . .	46
5.2	L'ontologia . . . . .	47
5.2.1	Costruzione teorica della struttura di base . . . . .	47
5.3	Il parser . . . . .	48
5.3.1	Applicazione del "Part of Speech" . . . . .	48
5.4	Gestione dei risultati . . . . .	48
<b>6</b>	<b>Applicazione dell'approccio proposto: implementazione</b>	<b>50</b>
6.1	Creazione manuale dell'ontologia . . . . .	50
6.1.1	Individuazione delle parole dell'ontologia . . . . .	51
6.2	Algoritmo relativo al parser automatico . . . . .	51
6.2.1	Utilizzo della libreria "Spacy" . . . . .	51
6.2.2	Implementazione per l'estrazione di pattern . . . . .	51
6.3	Gestione grafo delle parole individuate . . . . .	54
6.3.1	Utilizzo del software "Neo4j" . . . . .	55
6.3.2	Setting del driver . . . . .	55
6.3.3	Query del grafo ottenuto . . . . .	58
<b>7</b>	<b>Applicazione dell'approccio proposto: analisi dettagliata del case study</b>	<b>64</b>
7.1	Descrizione dataset . . . . .	64
7.2	Ontologia estrapolata . . . . .	65
7.2.1	Le parole più frequenti . . . . .	65
7.2.2	Risultati delle query effettuate . . . . .	68

---

<b>8</b>	<b>Discussione in merito al lavoro svolto</b>	<b>71</b>
8.1	Applicazione dell'algoritmo con le note delle sentenze . . . . .	71
8.1.1	Modifiche necessarie . . . . .	71
8.2	Ontologia attesa . . . . .	72
8.3	Possibili query di interesse . . . . .	73
	<b>Conclusioni</b>	<b>75</b>
	<b>Bibliografia</b>	<b>77</b>
	<b>Ringraziamenti</b>	<b>81</b>

---

## Elenco delle figure

---

3.1	Logo Microsoft SharePoint . . . . .	25
3.2	Magic Quadrant Content Service Platform 2021 . . . . .	26
3.3	Esempio di uno <i>spazio</i> con immagine 360° . . . . .	31
3.4	Home page <i>Sito1</i> . . . . .	32
3.5	Sezione <i>Pagine</i> . . . . .	33
3.6	Elenco delle <i>analisi</i> . . . . .	34
3.7	Heatmap giornaliera e oraria degli accessi . . . . .	35
3.8	Esempio relativo alla sezione <i>Documenti</i> . . . . .	35
3.9	Ricerca della parola <i>diagramma</i> . . . . .	36
3.10	Template per l'aggiunta di una regola di query . . . . .	36
3.11	Template per le origini dei risultati . . . . .	37
3.12	Template per i tipi di risultati delle ricerche . . . . .	37
3.13	Template per le impostazioni della ricerca . . . . .	37
4.1	Akoma Ntoso . . . . .	38
7.1	Ontologia creata dalle recensioni degli hotel . . . . .	67
7.2	Esempio di una parte del grafo ottenuto in <i>Neo4j</i> . . . . .	68
7.3	Tabella dei risultati ottenuti . . . . .	69
8.1	Ontologia ipotizzata per le note delle sentenze . . . . .	74

Questo progetto di tesi si sviluppa a partire dal progetto *Uni 4 Justice*, il cui obiettivo è quello di utilizzare i risultati del progresso tecnologico al fine di migliorare i processi giuridici delle Corti di Appello.

*Uni 4 Justice* è un progetto proposto dall'*Alma Mater Studiorum-Università degli Studi di Bologna*, la quale ha ottenuto il supporto di tredici Università e cinque Corti di Appello, diversamente dislocate nel territorio nazionale.

Tale progetto nasce dall'esigenza dei nostri sistemi giudiziari di secondo livello di allinearsi al contesto europeo. In particolare, *Uni 4 Justice* vuole consentire un aumento della digitalizzazione, in modo da velocizzare i sistemi informativi delle Corti di Appello, ed oltretutto consentire un migliore accesso della popolazione ai servizi erogati dalle Corti stesse (basti guardare il "successo" del Processo Civile Telematico, che è risultato di grande aiuto nel periodo pandemico).

La buona riuscita del progetto fa affidamento su figure come tesisti, borsisti e assegnisti di ricerca, che permettano la raccolta dei requisiti nelle diverse Corti, per poter evidenziare le criticità presenti nel territorio ed, infine, il monitoraggio continuo per l'attuazione del progetto. Inoltre, per avere una base solida, *Uni 4 Justice* ha specificato alcuni indicatori, definiti a livello europeo, da tenere sempre in considerazione nella progettazione, implementazione e realizzazione del progetto stesso.

Altri punti cardine del progetto sono la scelta dello sviluppo *Agile*, che comporta l'avanzamento del progetto sulla base dell'analisi delle *best-practice*, e l'utilizzo della *Data Science* per l'estrazione della conoscenza dalle note delle sentenze. In particolare, quest'ultimo aspetto è un vero e proprio punto di svolta, perché applicare l'*Intelligenza Artificiale* ai testi giuridici aumenta le informazioni utili riguardanti molteplici aspetti degli stessi. Un esempio di ciò sono le informazioni sulle decisioni prese in merito a determinati casi, oppure le agevolazioni per i giudici nelle ricerche delle sentenze simili ai casi che loro devono giudicare. Ciò è dovuto alla capacità che ha la *Data Science* di estrapolare informazioni dai dati passati, individuando pattern che si ripetono, e identificando cose che manualmente o "ad occhio" nessuna persona riuscirebbe mai a fare. Oltretutto, essa, basandosi sull'applicazione di algoritmi automatici, consente l'analisi di enormi quantità di dati in breve tempo.

Uno dei punti cardine per l'attuazione del progetto è quello della formazione dei dipendenti degli uffici giuridici, al fine di preparare il personale addetto alla gestione dei nuovi sistemi, per evitare il mancato utilizzo di alcune funzionalità che vengono introdotte appositamente con l'obiettivo di velocizzare determinate procedure.

Per capire al meglio tutti gli aspetti che impattano sul progetto *Uni 4 Justice* siamo partiti da un'attenta analisi del progetto stesso, sintetizzando tutte le sue sezioni e sotto-sezioni.

Ci siamo, quindi, dedicati alla presentazione di tutti gli argomenti previsti nel progetto, partendo dall'introduzione, passando per la rilevanza del progetto nei diversi contesti e terminando con il suo sviluppo secondo determinati articoli definiti e dettagliati.

Dopo l'attenta analisi del focus del progetto abbiamo considerato la situazione attuale della Corte di Appello di Ancona, entrando nel "vivo" della questione. Siamo partiti dai sistemi informativi delle sezioni civile e penale e abbiamo raccolto le richieste che le Cancellerie ci hanno riferito, nonché le problematiche attuali. A questo punto si è avuta coscienza del "gap" che vi è tra la situazione reale della Corte e il punto di partenza che *Uni 4 Justice* ha preso come riferimento per l'attuazione del progetto.

Per colmare tale gap, l'idea è quella di utilizzare un tool specifico, ossia *Microsoft SharePoint*. Abbiamo, quindi, descritto tutte le funzionalità di questo tool che possono essere utilizzate per rispondere alle esigenze delle cancellerie della Corte di Appello di Ancona. Partendo da una panoramica introduttiva, si è passati, poi, alla descrizione dei siti, nonché alle possibili alternative di creazione degli stessi, in base alla gestione degli utenti che più si desidera attuare. È stata esaminata, anche, la gestione della ricerca nel sito, al fine di introdurre la ricerca intelligente basata sugli algoritmi di *Data Science*.

Siamo passati, poi, alla descrizione di *Akoma Ntoso*, che è stato prescelto come formato *standard* in *Uni 4 Justice* per la creazione e manipolazione dei documenti giudiziari. Si sono esaminati, quindi, una serie di marcatori, accettati dallo *standard*, e, per ciascuno di essi, è stato esaminato il rispettivo utilizzo.

Dopodiché, è stato definito, progettato e implementato l'approccio da noi proposto per l'estrazione di conoscenza dalle note delle sentenze. Tale approccio si concentra su tre passaggi fondamentali, ovvero la creazione dell'ontologia, la realizzazione del parser automatico e la gestione dei risultati ottenuti. Siamo partiti dalla spiegazione teorica e prettamente astratta delle tre parti che compongono il metodo, quindi dalla creazione dell'ontologia a partire dalle parole più frequenti, e abbiamo continuato con la creazione di un parser automatico in base alla semantica dei termini, per poi discutere della gestione dei risultati ottenuti a valle delle fasi precedenti, pur sempre in termini astratti.

Siamo, poi, passati ad un livello di dettaglio maggiore, entrando nello specifico dell'implementazione. Abbiamo, pertanto, visto gli strumenti utilizzati e abbiamo considerato le parti più importanti del codice, commentando riga per riga come funziona l'algoritmo nei punti più interessanti.

Questo approccio si è però concretizzato in un contesto diverso da quello giuridico, in quanto non erano ancora disponibili le note delle sentenze; quindi si è deciso di trovare un caso di studio che avesse testi strutturalmente simili, al fine di creare l'algoritmo e testarlo, per poi ottenere dei risultati concreti. Abbiamo, così, deciso di fornire delle linee guida per la modifica di alcune righe di codice dell'algoritmo, per poterlo eseguire sulle note non appena disponibili.

Questo progetto di tesi è composto da nove capitoli strutturati come di seguito specificato:

- Nel Capitolo 1 verrà illustrato il progetto *Uni 4 Justice* in tutte le sue parti, dalle problematiche che sono state individuate alle soluzioni che il progetto propone per farvi fronte.
- Nel Capitolo 2 verrà mostrata la situazione attuale della Corte di Appello di Ancona, entrando nel dettaglio delle esigenze innovative che le cancellerie civile e penale richiedono.
- Nel Capitolo 3 verrà introdotto *Microsoft SharePoint* come soluzione per preparare la Corte di Appello di Ancona all'attuazione del progetto *Uni 4 Justice*. In particolare, verrà descritto il tool in dettaglio, a partire dalle alternative nella creazione e personalizzazione dei diversi siti, con un'attenzione specifica per la funzione di ricerca.

- Nel Capitolo 4 verrà spiegato *Akoma Ntoso*, standard per la gestione dei documenti giudiziari; si potranno capire, quindi, i "tag" consentiti dallo standard e il loro utilizzo.
- Nel Capitolo 5 verrà delineato l'approccio proposto, partendo dal *Natural Language Processing*, spostandosi sulla definizione dell'ontologia, per poi passare alla spiegazione del parser e finire con la gestione dei risultati.
- Nel Capitolo 6 verrà definita l'implementazione effettiva dell'approccio proposto nel capitolo precedente, entrando nel dettaglio dell'algoritmo e delle righe del codice stesso, spiegando tutti i punti critici.
- Nel Capitolo 7 verrà descritto il caso di studio preso in esame per l'esecuzione dell'algoritmo, mostrando i risultati ottenuti tramite l'approccio proposto.
- Nel Capitolo 8 verranno elencate tutte le modifiche da fare all'approccio proposto al fine di applicarlo alle note delle sentenze, non appena disponibili.
- Nel Capitolo 9, infine, verrà effettuato un recap dell'intero lavoro svolto, suggerendo alcuni sviluppi futuri.

---

## Il progetto Uni 4 Justice

---

*In questo capitolo si proveranno a sintetizzare le idee su cui Uni 4 Justice si è sviluppato, cercando di carpire i punti chiave su cui il progetto si concentra. Si metteranno in risalto le criticità del sistema giudiziario italiano che Uni 4 Justice intende affrontare con un processo di digitalizzazione, tramite le proposte che il Progetto vuole attuare, per un miglioramento nelle Corti di Appello.*

*L'obiettivo di questo capitolo è quello di descrivere, tramite un "percorso" all'interno del Progetto, i punti di svolta che sono stati definiti in esso, a partire dall'identificazione degli atenei e delle Corti di Appello che hanno aderito al progetto, passando per l'individuazione degli strumenti e delle tecniche che il Progetto intende utilizzare, fino ad arrivare ad una descrizione delle azioni su cui esso si struttura.*

### 1.1 Introduzione

*Uni 4 Justice* è un progetto che trae origine dalla volontà di promuovere il cambiamento organizzativo, tecnologico e professionale di una parte del sistema giudiziario, al fine di allinearsi al contesto europeo. Esso si basa sulla collaborazione tra il mondo universitario e quello della ricerca scientifica e il sistema giudiziario di alcune corti di appello. Concretamente, il rapporto è stato stretto con:

- *Corte di appello di Ancona;*
- *Corte di appello di Bologna;*
- *Corte di appello di Trento;*
- *Corte di appello di Trieste;*
- *Corte di appello di Venezia.*

Le Università partner che collaborano al progetto sono le seguenti:

- *Alma Mater Studiorum - Università degli Studi di Bologna (soggetto proponente).*
- *Università degli Studi di Ferrara.*
- *Università degli Studi di Modena e Reggio Emilia.*
- *Università degli Studi di Parma.*
- *Università di Verona.*

- *Università Ca' Foscari di Venezia.*
- *Università degli Studi di Padova.*
- *Università degli Studi di Trento, inclusa la sezione distaccata di Bolzano.*
- *Università Politecnica degli Studi delle Marche.*
- *Università degli Studi di Macerata.*
- *Università degli Studi di Camerino.*
- *Università degli Studi di Urbino "Carlo Bo".*
- *Università degli Studi di Udine.*
- *Università degli Studi di Trieste.*

## 1.2 Rilevanza rispetto al contesto territoriale

Il progetto *Uni 4 Justice* assume rilevanza nei confronti della macro area 2, la quale include i cinque distretti giudiziari delle corti di appello sopra citati.

Nella scelta della macro area da prendere in considerazione sono state valutate diverse variabili, sia strutturali che funzionali. La decisione, poi, è ricaduta sui distretti che compongono la macro area 2, i quali si caratterizzano da quattro aspetti legati alle specificità territoriali, ovvero:

- *La distribuzione del contenzioso.* L'analisi della distribuzione dei contenziosi, nei vari distretti, ha fatto emergere che il maggior numero di cause si concentrano in materia di diritto commerciale, in particolare controversie legate ai contratti, nonché in materia di tutela dei diritti delle famiglie. Altri settori in cui il contenzioso è particolarmente presente, con conseguente assorbimento di molteplici risorse del sistema giudiziario, sono la protezione internazionale e le procedure concorsuali.
- *La distribuzione degli spazi.* La problematica dello spazio, e della necessità di un processo di digitalizzazione e di innovazione, ha assunto ancora maggior rilevanza durante il periodo pandemico, motivo per il quale è stato necessario ricorrere in maniera massiva allo "smart-working" e affrontare, quindi, la mancanza di un sistema digitale performante. In questa tematica rientra, anche, l'esigenza di riorganizzare gli spazi materiali e gli uffici, per migliorare le funzionalità e ottenere, quindi, un'ottimizzazione dei processi.
- *Innovazioni tecnologiche e organizzative.* Il radicale cambiamento dato dall'innovazione tecnologica permette di massimizzare l'efficienza operativa. Uno degli impatti significativi della digitalizzazione, nel mondo della giustizia, è imputabile al PCT (Processo Civile Telematico), il quale ha cambiato l'assetto che era insito sin dal principio, e ha permesso di decongelare la situazione nel campo giuridico, in un momento di grande difficoltà, come l'emergenza sanitaria dovuta al COVID-19. Questo è un segnale importante; esso ci permette di affermare che in molti casi la digitalizzazione non solo è di aiuto, ma è necessaria. Questo processo di digitalizzazione è iniziato nel 2016, quando la Direzione Generale dei Sistemi Informativi e Automatizzati (DGSIA) del Ministero della Giustizia identifica TIAP come gestore documentale nazionale. L'obiettivo che si poneva questo sistema era quello di "pervenire alla digitalizzazione del fascicolo attraverso la scannerizzazione - o acquisizioni di file digitali - la classificazione, la

codifica e l'indicizzazione dei fascicoli con possibilità di ricerca, consultazione, esportazione e stampa di interi fascicoli e/o di singoli atti"<sup>1</sup>. Nel tempo, però, ci si è resi conto che questo non era abbastanza, ma che ci fosse l'esigenza di un'informatizzazione a 360°, che coinvolgesse tutto il fascicolo digitale, in quanto la digitalizzazione deve coinvolgere l'intero sistema.

- *La dinamica del personale.* Con riferimento a questo aspetto sono stati identificati tre componenti necessari per una durevole capacità di orientamento nell'azione di buon governo. Le componenti individuate sono le seguenti:
  - disponibilità di una base di dati sempre aggiornata;
  - verifica dei fatti nel caso di ipotesi di lavoro non andate a buon fine;
  - analisi sistemica dovuta al carattere multilivello su cui si basa il sistema giuridico.

Per affrontare tali tematiche, *Uni 4 Justice* basa il proprio funzionamento su due principi cardine. Essi sono:

- Mettere al centro il ciclo di produzione, validazione sul posto e diffusione della conoscenza come catalizzatore di qualità della giustizia.
- Intendere le collaborazioni inter-istituzionali come collaborazioni basate sulla partecipazione e sull'inclusività, nonché sulla chiarezza dei ruoli.

Per fare ciò il progetto deve basarsi su due aspetti metodologici di lavoro; i quali si sviluppano nel seguente modo:

- Preparazione della sede giudiziaria oggetto di ricerca, ancor prima di avviare la fase operativa delle ricerche.
- Apprendimento orizzontale, attuabile tramite delle tabelle di analisi.

*Uni 4 Justice* sviluppa la prima fase del progetto attraverso quattro azioni, ovvero:

- *La predisposizione dei progetti.* Si rivela di fondamentale importanza la fase preliminare di progettazione e di preparazione delle condizioni alla base dei progetti di ricerca. Ciò è realizzabile attraverso processi di digitalizzazione dei servizi e l'implementazione di misure orientate a migliorare i servizi stessi, e a raggiungere, quindi, una migliore efficienza amministrativa.
- *La formazione.* È indispensabile attuare un profondo piano di formazione per tutti gli operatori di ricerca che collaborino con gli uffici giudiziari, in modo che essi possano condividere il proprio *know-how* con il personale amministrativo degli uffici.
- *L'elaborazione della metodologia di monitoraggio.* Si predispone una griglia utile per la rilevazione dei risultati ottenuti dai progetti di ricerca. Questo consente di effettuare una comparazione tra i risultati ottenuti con questo progetto tramite le griglie di rilevazione internazionale. Ciò si concretizza con l'individuazione di indicatori che vengono adottati sia dallo European Justice Scoreboard, sia dal CEPEJ che dall'OCSE.
- *La valorizzazione del dialogo con gli atenei.* È di cardinale importanza far leva sul dialogo tra i diversi soggetti coinvolti nel progetto, coordinando le singole equipe di ricerca. Ciò è rilevante al fine di estrapolare le maggiori informazioni e idee possibili dai ricercatori e borsisti.

---

<sup>1</sup>Il sistema documentale TIAP - tiap- CMS.

### 1.3 Rilevanza e coerenza con la strategia nazionale per la Giustizia Digitale

Il progetto mira a contribuire alla trasformazione digitale del Sistema Giustizia e all'accrescimento nel sistema universitario delle competenze digitali specializzate nel campo giuridico. La Presidenza del Consiglio dei Ministri ha adottato la "*Strategia per la Crescita Digitale 2014-2020*" finalizzata ad un cambiamento radicale della politica pubblica in materia di *public governance*.

I punti di svolta per attuare questa crescita digitale sono i seguenti:

- Creare un linguaggio digitale per le procedure esistenti, trasformando in flussi di interfaccia i processi di lavoro.
- Mettere al centro la qualità sostenibile ed efficiente, assicurando la tutela dei diritti sull'intero territorio.
- Promuovere l'accesso e la leggibilità dei processi e delle procedure del sistema giuridico, pur essendo quest'ultimo stratificato in livelli diversi. Per raggiungere tale scopo, ci si basa sulla digitalizzazione, sulla dematerializzazione dei procedimenti e sull'erogazione delle risposte su richiesta in tutto il territorio.

*Uni 4 Justice* si applica nel contesto sopra riportato tramite quattro dimensioni. Esse sono:

- *La natura infrastrutturale delle piattaforme digitali.* Viene introdotto il concetto di investimento, rispetto alla precedente idea di spesa fine a se stessa. La spesa pubblica per le infrastrutture digitali, per le reti e per la connettività va intesa come un investimento e non più come una voce passiva del bilancio dello Stato. Questo perchè un costo finalizzato all'obiettivo può effettivamente apportare un avanzamento nella realtà del sistema attuale. Ovviamente, tali risorse pubbliche non devono essere sprecate in investimenti volatili, ed è quindi necessario affiancare l'investimento stesso a meccanismi di consolidamento connessi con le professionalità degli attori presenti. Nella pratica ciò si concretizza, già nella fase preliminare, nel fatto che il sistema si integri con i processi basati sulla partecipazione, sulla trasparenza, nonché sul monitoraggio dei risultati, al fine di perseguire l'innovazione tecnologica.
- *Il decoupling fra funzioni e territorio.* Viene valorizzato il territorio rispetto alle singole funzioni, quindi si vuole disaccoppiare la funzione stessa nei confronti della sede fisica. Il fine di ciò è poter eseguire determinate pratiche in più luoghi fisici, senza essere vincolati da un solo luogo specifico.
- *La professionalità integrale.* Vengono messe al servizio del sistema giudiziario le conoscenze tecniche basate anche sulle intelligenze artificiali.
- *La nozione di servizio.* Viene analizzata la capacità del sistema amministrativo giudiziario di rispondere alle richieste in maniera reattiva, con particolare attenzione alle situazioni svantaggiate dal punto di vista delle infrastrutture di accesso.

In questo contesto, la digitalizzazione incentiva l'accesso delle informazioni giuridiche al singolo cittadino. *Uni 4 Justice* si pone, quindi, l'obiettivo di analizzare quali siano le condizioni ottimali affinché attori altamente specializzati dal punto di vista professionale (come nel campo del diritto, della tecnologia, della scienza delle informazioni, etc.) riescano ad ottenere un'ottima interazione dei propri saperi, tenendo sempre presente che l'obiettivo nel settore pubblico è quello della tutela dei diritti della persona.

Sono stati riconosciuti tre livelli di osservazione. Essi sono:

- *Il livello individuale*, il quale si sostanzia a sua volta in due caratteristiche, ovvero:
  - La modalità con cui gli attori prendono delle decisioni, come, ad esempio, il Giudice che, nel pervenire ad una sentenza, deve strutturare il ragionamento con un metodo che tenga conto sia di conoscenze di tipo giuridico, sia di conoscenze di tipo tecnico-scientifiche. Questo approccio non deve essere di rigida contrapposizione tra il mondo giuridico e quello scientifico, ma bensì di coesione e basato su logica e razionalità.
  - La modalità con cui gli attori elaborano la propria organizzazione di lavoro e il lavoro delle persone con cui collaborano.
- *Il livello organizzativo*, il quale si basa sul rapporto che necessariamente viene a crearsi tra giustizia e scienza. Uno degli scenari in cui ciò ha maggior impatto è il potenziale di analisi nel campo giuridico, tramite lo studio delle banche dati, ma, in particolar modo, grazie all'intervento dell'intelligenza artificiale. Questo è realizzabile, quindi, tramite sistemi di diagnosi e analisi degli andamenti giurisprudenziali.
- *Il livello sistemico*, il quale si concretizza nella stratificazione del sistema giuridico nell'ottica di nuove funzioni; questo è, in parte, conseguenza degli sviluppi tecnologici e scientifici. Alcuni esempi di queste innovazioni sono lo sviluppo e la manutenzione dei sistemi informativi e delle infrastrutture tecnologiche immateriali, che si rivelano di fondamentale importanza per garantire la privacy, la cyber-security, nonché la sicurezza dell'intero sistema.

Una delle fondamenta su cui il progetto *Uni 4 Justice* deve fare affidamento è la formazione e la qualificazione professionale del personale. Infatti, dopo anni in cui si è puntato ad aumentare il numero del personale, ora è il momento di concentrarsi sulle competenze e sulla professionalità, dal punto di vista organizzativo, gestionale, economico e statistico.

Con riferimento alle professionalità e competenze utili al sistema giustizia, si è individuata la possibilità di incentivare il personale ad uno "stretching" mentale, integrato con un processo "problem driven" grazie alle conoscenze già acquisite.

È emerso che in Italia è presente una scarsa dotazione di personale con competenze organizzative e manageriali all'interno degli uffici giudiziari.

Al fine di ottenere questo necessario "upgrade" professionale, sono stati individuati due strumenti che possono essere utilizzati come programma didattico nelle università:

- dottorato con tesi di ricerca applicata; si potrebbero sviluppare corsi specifici applicati alla corte di appello e all'intera branca della statistica nel mondo giuridico;
- master professionalizzante; si potrebbe concentrare sulla riqualificazione delle competenze dei funzionari delle cancellerie.

## **1.4 Rilevanza e coerenza con le disposizioni normative tramite modelli organizzativi**

Il progresso e l'impatto della trasformazione tecnologica dell'ultimo decennio, ha comportato l'inserimento nel progetto delle particolari competenze scientifiche ed accademiche. In particolare, nell'ottica della cosiddetta socializzazione digitale e modernizzazione della Pubblica Amministrazione, *Uni 4 Justice* inserisce all'interno degli uffici giudiziari un insieme di competenze multidisciplinari di settore che riguardano *e-government, law and ICT, digital governance e data privacy law*.

Vi è sempre stato un "gap" tra la ricerca didattica e il sistema giudiziario; questo perché le tempistiche dei due "mondi" sono profondamente diverse. Per questo motivo, una delle cose fondamentali di cui necessita il progetto per una buona realizzazione è una metrica di insieme che detti il ritmo e le fasi del progetto stesso. Il flusso attraverso il quale questo diventa possibile è partire dai risultati ottenuti dallo studio effettuato dal ricercatore, rifletterli nelle organizzazioni, e da queste ottenere nuovi interrogativi per i ricercatori.

Il fine ultimo di questa interazione, tra ricerca e giustizia, è ottenere una migliore qualità della vita sociale, economica ed istituzionale, per trovare le migliori soluzioni nei vari ambiti di efficienza, gestione del personale, sostenibilità ambientale, prevenzione del rischio e ottimizzazione dell'uso della tecnologia digitale; quest'ultima, risulta essere colei che fa da catalizzatore per il cambiamento.

Un ulteriore concetto fondamentale è quello dell'*accesso*, il quale deve essere costituzionalmente garantito. Ma per poterlo garantire è necessario passare all'attuazione delle norme attualmente previste; non può, infatti, bastare la legge a tutela della garanzia, bensì è necessaria la verifica dell'esecuzione della norma stessa.

Consentire l'accesso ad un servizio pubblico si rivela essere una condizione essenziale per garantire un equilibrio tra la domanda e l'offerta. Questo concetto è stato approfondito già dall'OCSE sin dal 2015 con un progetto intitolato "Equal access for All", il quale si fonda su alcuni punti cardine. Essi sono:

- la chiarezza e la leggibilità delle procedure;
- la sostenibilità dei tempi e del servizio reso;
- la fruibilità degli spazi e delle informazioni che orientano il cittadino;
- la tempestività del processo di apprendimento (accesso per feedback);
- la disponibilità del sistema ad inserirsi in un complesso nodo di interfaccia e interoperabilità sul territorio rispetto alla policy dei dati.

Diversi fattori hanno inciso sul passaggio ad una visione più ampia dei profili di carattere formale e processuale, rispetto ad una visione più centrata che si tendeva avere nei confronti degli stessi, per potersi, quindi, concentrare sulla funzionalità, piuttosto che sulla formalità. I fattori che hanno portato a questo cambiamento sono i seguenti:

- la crescita di interesse per le forme extra-giudiziali e stragiudiziali delle controversie;
- l'aumento della domanda di giustizia;
- la difficoltà nel conciliare la specializzazione, necessaria a causa della complessità della normativa, e la visione di insieme che caratterizza la professionalità del giurista.

L'impatto che queste esigenze hanno portato ha fatto sì che le istituzioni giudiziarie fossero prese maggiormente in considerazione, in quanto è, di fatto, di forte rilevanza nella vita civile e economica della società.

Lo Stato viene considerato come *limen*, cioè ciò che segna il "dentro" e "fuori" tra la domanda di diritti e l'offerta di servizi, fra richieste di riconoscimento, appartenenza e assegnazione degli spazi legittimi o possibili.

Un'ulteriore discrasia sempre più rilevante è quella tra funzione e struttura statuale, la quale risulta accentuata dalla rivoluzione digitale, anche se, fondamentalmente, è solo una trasposizione delle stesse procedure e funzioni, ma che passano da uno spazio reale e materiale ad uno spazio digitale e immateriale. Basti pensare all'avvento dello SPID (Sistema Pubblico di Identità Digitale) che permette l'accesso nei servizi online della Pubblica Amministrazione

e di alcuni privati, offrendo, così, la possibilità al singolo cittadino di effettuare determinati *task* senza doversi presentare in un luogo specifico per poter conseguire la propria richiesta, velocizzando tutto il flusso che una singola pratica altrimenti richiederebbe.

Potrebbe risultare preoccupante l'impatto della tecnologia, ma quello che si vuole ottenere non è la sostituzione completa dei sistemi attuali della Pubblica Amministrazione con sistemi digitali sviluppati sulla base dell'intelligenza artificiale, bensì accostare al personale nel settore giuridico un sistema informatico di supporto alle decisioni, che possa fornire dati oggettivi capaci di supportare il personale a fare il proprio lavoro, senza mai sostituirlo.

Questo concetto porterebbe nell'intero sistema un aumento di fiducia da parte del cittadino, nelle sentenze emanate dai giudici in quanto gli verrebbero forniti dei dati scientifici a supporto delle motivazioni che hanno portato il giudice alla sentenza stessa, garantendone, quindi, l'eguaglianza nella decisione presa. Inoltre, aumenterebbe di gran lunga l'efficienza che nelle pratiche giuridiche spesso viene a mancare, a partire dalla questione temporale con cui le sentenze vengono erogate, a causa del sovraccarico con cui l'organismo di giustizia si trova a doversi confrontare per adempiere al proprio dovere costituzionale.

In questo contesto, spicca il ruolo cardine che solo le università possono svolgere, in quanto l'unico modo per rendere ciò possibile è tramite un percorso formativo che permetta di unire le tematiche del diritto con l'intelligenza artificiale.

Un esempio di questo tipo di applicazione è l'integrazione di un insieme di funzioni che aiutino il sapere dei giudici tramite istruttorie preliminari dei fascicoli, in accordo con le cancellerie, come eseguito già in altri paesi OCSE (Olanda, Francia e Germania).

*Uni 4 Justice* si fa portavoce di questa ideologia di base, nell'aumento della digitalizzazione in campo giuridico, tramite alcuni cambiamenti nel modo di ragionare. Il cambio di paradigma viene identificato tramite alcuni concetti su cui basarsi. In particolare:

- I borsisti di ricerca non sono da intendersi come risorse umane in aggiunta allo staff, ma creatori di capacità, di percorsi.
- L'intervento della ricerca applicata non è una forma di "manutenzione" del sistema giustizia, ma un progressivo accompagnamento ad un nuovo *modus operandi*.
- La promozione di cultura della digitalizzazione centrata sulle persone evita il cortocircuito della prospettiva sostitutiva (sostituzione fra uomo e macchina) per andare verso un'ottica di integrazione basata sulla guida consapevole.

Affinchè le proposte precedentemente spiegate riescano ad essere applicate in modo corretto nel sistema reale è necessario, in primo luogo, studiare le dinamiche operative e funzionali di ogni sezione del sistema giudiziario e annessi, dalle segreterie all'amministrazione etc., a seguito di ciò sarà possibile valutare le criticità e le potenzialità del "cambiamento migliorativo" che la digitalizzazione effettua nei processi attuali.

I punti su cui il progetto si concentra è l'inserimento delle capacità organizzative, di gestione, di comunicazione e di osservazione con il supporto di strumenti quantitativi di analisi, mantenendo il punto di vista giuridico delle procedure.

Alcuni degli spunti di analisi a cui fare riferimento sono:

- La ricostruzione sulla base della letteratura di riferimento delle condizioni processuali e normative entro cui operano i servizi di giustizia.
- Lo "shadowing", ossia la puntuale osservazione e rilevazione delle micro-attività che danno forma all'agenda quotidiana di specifici profili funzionali, come giudice, cancelliere e personale tecnico.
- L'osservazione partecipante delle prassi in essere nelle unità organizzative micro, come gruppi di lavoro, cancellerie, etc.

- La raccolta di dati quantitativi e qualitativi attraverso misurazioni e interviste.

Tramite lo "shadowing" vengono valutate le funzionalità e disfunzionalità nelle diverse sedi giudiziarie, incentivando, così, l'individuazione delle "leve" del cambiamento tramite la ricerca.

Affinchè sia possibile trasmettere, radicare e consolidare le innovazioni individuate, è necessario il supporto di più "attori", nonché quello della corralità nell'attuazione e verifica delle idee proposte come miglorie.

A valle delle proposte e considerazioni sopra descritte, si ragiona, ora, in termini di distacco dall'essere fisicamente negli uffici giudiziari, quindi, risultano necessari supporto e strumenti per attuare tale cambiamento. Ciò viene realizzato anche tramite analisi qualitative e quantitative di progetti tra istituzioni locali e uffici giudiziari, selezionando quelli che hanno ottenuto buoni risultati; ci si pone, quindi, l'obiettivo di migliorare l'organizzazione degli uffici giudiziari e la sinergia fra servizi e funzioni.

In particolare, con *Uni 4 Justice*, verranno utilizzati diversi strumenti. Alcuni di essi sono:

- *Normativa tabellare Consiglio Superiore della Magistratura*. Meccanismo di potenziamento della compliance e dell'attuazione delle azioni tra quelle di tipo giuridico e amministrativo.
- *Normativa primaria in materia digitale*. Meccanismo di rafforzamento delle capacità amministrative e gestione di tool integrati con l'intelligenza artificiale. Tale meccanismo vede le seguenti possibili applicazioni:
  - Utilizzare i concetti di ontologie semantiche, nonché l'assegnazione di un concetto semantico ad una parte di testo o file multimediale.
  - Utilizzare i dataset, ovvero un insieme di dati più o meno omogenei, dal quale poter ottenere informazioni utili per estrapolare "conoscenza" dai dati stessi. Tramite la conoscenza è, poi, possibile capire cose che, a prima vista, il singolo dato, non può fornire.
  - Utilizzare il data mining e la data analytics, come anticipato nel punto precedente, per poter descrivere, diagnosticare, predire e prescrivere qualcosa di nuovo, partendo dal dataset.
- *Normativa tecnica dei sistemi informativi della giustizia*. Utilizzo degli strumenti standard previsti dal regolamento eIDAS (910/2014), nonché Akoma Ntoso<sup>2</sup>, ELI<sup>3</sup>, ECLI<sup>4</sup>.

## 1.5 Complementarietà con altri strumenti di programmazione rilevanti

Il PON (Programma Operativo Nazionale) è lo strumento nazionale per il rafforzamento e l'innovazione della Pubblica Amministrazione concordato tra l'Italia e la Commissione Europea. In questo contesto, *Uni 4 Justice*, per poter portare a termine la realizzazione di molteplici obiettivi ambiziosi, parte da un'azione fondamentale, l'*Azione 1.4.1*. Essa è denominata "Azioni di miglioramento dell'efficienza e delle prestazioni degli uffici giudiziari attraverso l'innovazione tecnologica (informatizzazione del processo civile), il supporto organizzativo all'informazione e telematizzazione degli uffici giudiziari, disseminazione di specifiche innovazioni e supporto all'attivazione di interventi di *change management*". Questa Azione, relativa

<sup>2</sup><https://docs.oasis-open.org/legaldocml/akn-nc/v1.0/akn-nc-v1.0.html>

<sup>3</sup><https://eur-lex.europa.eu/eli-register/italy.html>

<sup>4</sup>[https://e-justice.europa.eu/content\\_european\\_case\\_law\\_identifier\\_ecli-175-it.do](https://e-justice.europa.eu/content_european_case_law_identifier_ecli-175-it.do)

all'Asse 1, è collegata ad una relativa all'Asse 2. Entrambe puntano ad aumentare l'efficienza del lavoro dei giudici tramite un modello collaborativo per il magistrato. È necessario partire, quindi, da un progetto pilota per la creazione di Uffici del Processo, per poi realizzare progetti di coordinamento tra uffici, università e ordini degli avvocati, per massimizzare gli indirizzi giurisprudenziali, e, infine, concentrarsi sul diffondere il software Consolle Assistente presso gli uffici pilota. I punti sui quali si sviluppa questa Azione sono:

- Evoluzione e completamento dei *software* e adeguamento della dotazione *hardware* per sviluppare la cooperazione fra i sistemi esistenti.
- Creazione del fascicolo penale digitale, attraverso la digitalizzazione di tutti gli atti, anche al fine di migliorare il sistema delle notifiche penali e la creazione di un sistema unico con l'integrazione dei veri progetti.
- Adottare strumenti di *e-governance* e ICT, fruibili da cittadini e imprese, con interventi di estensione dei sistemi multivideo conferenza e video telepresenza, i cui beneficiari sono gli Uffici Giudiziari di I Grado (Giudici di Pace e Tribunali).

"Uffici di Prossimità" (UdP) è uno dei progetti realizzati per implementare il PON, il quale ha permesso un maggiore accesso ai Tribunali, grazie agli Enti locali, ai sistemi informatici e all'utilizzo del Processo Civile Telematico (PCT). L'utilizzo del PCT è stata una grossa innovazione, di grande impatto, a maggior ragione nel periodo pandemico; da ciò si evince un effettivo vantaggio reale, a partire da un progetto il cui obiettivo era un cambiamento radicale, e migliorativo, basandosi sulla digitalizzazione.

Per la realizzazione di questo sono stati definiti e attivati dei modelli organizzativi, gestionali e tecnico-informatici, applicati in ogni regione del territorio. Infatti, i singoli progetti regionali, che hanno come destinatari Comuni, Uffici giudiziari e cittadini, devono effettuare determinate attività. Esse sono:

- Allestimento delle sedi degli UdP mediante l'acquisizione di beni mobili e strumentali e arredi.
- Formazione del personale (*e-learning* e *training on the job*, in presenza e *on-demand*).
- Infrastrutturazione informatica per il corretto funzionamento dei software che verranno trasferiti a seguito della modellizzazione (Redattori atti e Punti di accesso) e *help desk*.
- Comunicazione sul territorio digitale.
- Digitalizzazione dei procedimenti pendenti in materia di giurisdizione volontaria a livello regionale, necessaria per l'attivazione degli UdP.

Visti i risultati ottenuti con il PCT, il Ministro della Giustizia vuole ampliare lo strumento, per renderlo utilizzabile dai Giudici di Pace, integrando, quindi, i sistemi informativi attuali nella giustizia civile, con notifiche e comunicazioni di cancelleria in via telematica agli uffici dei Giudici di Pace.

Nel tempo, si è, di fatto, creata una situazione in cui risulta necessario l'utilizzo di alcuni *indicatori* per individuare la capacità del sistema nell'efficienza, nella tempestività e nella sostenibilità della risposta di giustizia. Tali indicatori hanno effettivamente aiutato nell'identificazione dei problemi, con conseguente risoluzione delle problematiche.

Tutto ciò è stato realizzato a partire da uno sviluppo di analisi della *qualità*, un concetto espresso sotto forma di triade, ovvero:

- qualità delle procedure, quindi qualità vista come *garanzie*;

- qualità della macchina organizzativa, quindi qualità vista come *efficienza*;
- qualità dell'interazione giustizia-società, quindi qualità vista come *uguale accesso e fiducia*.

In relazione a ciò, sono stati individuati gli indicatori che esprimono al meglio i concetti stessi. Gli indicatori considerati sono, quindi, i seguenti:

- gestione del carico di lavoro e dei tempi (CEPEJ: *clearance rate* e *disposition time*);
- numero delle sopravvenienze e carico di lavoro (CEPEJ);
- gap fra domanda e risposta (OCSE);
- tasso di discriminazione nell'accesso alla giustizia (OCSE e Nazioni Unite);
- fiducia (Europea Barometer);
- percezione del business (World Economic Forum);
- costi derivanti dai tempi di definizione dei procedimenti (World Economic Forum e Banca Mondiale);
- costi derivati dalle procedure di esecuzione (CEPEJ).

Tali indicatori, insieme ad altri, sono quelli su cui si basa *Uni 4 Justice*. Gli indicatori aggiuntivi, invece, sono quelli di rendimento connessi alle azioni di rafforzamento rispetto ad alcuni contesti. I contesti di analisi del rendimento presi in esame dal progetto sono quelli relativi:

- alle capacità amministrative (PON);
- alla qualità dei territori (POR);
- alla competitività e coesione sociale (POR);
- alla riduzione dell'arretrato e messa a regime del *clearance rate* e del *disposition time* rispetto agli impegni sottoscritti nel contesto del PNRR.

Concludiamo questa sezione evidenziando l'aspetto centrale di tutti i discorsi sopra citati. In particolare, si vuole far capire che i punti su cui *Uni 4 Justice* fa leva sono la trasformazione digitale, nonché la modernizzazione e semplificazione della Pubblica Amministrazione. Essendo il progetto multi-disciplinare ha richiamato a sé l'esigenza dell'inter-operabilità, la quale risulta molto dispendiosa, ma volta ad ottenere risultati efficaci nel sistema attuale. Infatti, essa permette una maggiore fiducia e concretezza nell'applicazione del progetto stesso rendendo produttivo e proficuo l'investimento fatto su *Uni 4 Justice*.

Tutto questo processo di cambiamento tecnologico porterà, senza alcun dubbio, molteplici vantaggi, ottimizzando gran parte, se non tutti, i flussi di processo di lavoro nel mondo della giustizia, a partire dall'orientamento della domanda e dall'aumento della tempestività. Essa porterà, anche, una soluzione su larga scala di una tra le problematiche che il sistema giuridico continua a dover affrontare.

## 1.6 Articolazioni e contenuti del progetto

*Uni 4 Justice* si prefissa diversi obiettivi. Gli obiettivi generali, con i rispettivi obiettivi operativi individuati, sono i seguenti:

- Promuovere le "best practice" organizzative Ufficio per il processo.
  - Progettare modelli di ufficio per il processo modulari e flessibili.
  - Diffondere modelli di ufficio per il processo rispondenti alle esigenze (dinamiche) di uffici giudiziari.
- Rafforzare le capacità di monitoraggio e valutazione del sistema giustizia.
  - Costruire una strategia di monitoraggio e metterla a disposizione degli uffici giudiziari.
- Rafforzare la sinergia fra formazione, professionalità e dinamica dei bisogni di funzionamento dell'amministrazione della giustizia e della giurisdizione.
  - Mappare i bisogni (e le loro traiettorie evolutive) di formazione e professionalità.
  - Costruire una strategia sinergica per la formazione e l'accompagnamento delle nuove risorse umane del sistema giudiziario.
- Affermare una visione integrata delle intelligenze per il giusto processo, che contribuisca alla leadership italiana in Europa.
  - Costruire strumenti di trasformazione digitale per la giustizia, di legal design, AI e di screening diagnostica (tramite la data analytics) giurisprudenziale, capaci di valorizzare il valore cognitivo del PCT e del PPT.
  - Costruire strumenti di monitoraggio dell'andamento delle decisioni nel sistema giustizia - sia per le decisioni giurisprudenziali, sia per quelle che riguardano l'allocazione delle risorse.

Da questo momento in poi, verranno estrapolate dalle "schede di azione e di prospetto sinottico delle linee di azione per linee di intervento" le informazioni di maggior rilevanza ai fini della tesi che si sta discutendo.

### 1.6.1 Raccolta dei bisogni e delle eccellenze per l'UPP

Avvio dell'operatività della *task force*. Azione relativa al *core* dell'intero progetto, vagliando nel territorio uno studio applicato alla mancanza di UPP o alla discontinuità di alcuni esistenti.

Si vuole supportare lo studio effettuato dai borsisti nel campo operativo, quindi negli uffici giudiziari. Inoltre, è di fondamentale importanza interagire nel miglior modo possibile con il Consiglio dell'Ordine degli Avvocati, tramite lo svolgimento di determinati incontri *critical review*, organizzati con il COA stesso.

### 1.6.2 Strumenti operativi di monitoraggio e consultazione per il potenziamento dell'UPP

Questa azione si appoggia sull'individuazione di determinati obiettivi. Essi sono:

- *Analisi giurisprudenziale*. Essa è necessaria per effettuare le ricerche in modo più semplice, per mettere a confronto casi simili, estrapolare informazioni rilevanti, analizzare statistiche e visualizzazioni così da apportare un aiuto al lavoro dell'UPP.

- *Banca dati in Akoma Ntoso delle sentenze.* Tramite lo standard internazionale e nazionale OASIS LegalDocML Akoma Ntoso raccogliere l'insieme delle sentenze disponibili. Ciò consentirà di utilizzare questa banca dati al fine di estrapolare la conoscenza dall'informazione ottenuta dai dati stessi, per poi creare una ricerca applicata alle conoscenze ricavate, ottimizzando la ricerca stessa sulla base dei concetti semantici insiti nel testo delle sentenze.
- *Statistiche e dashboard.* L'obiettivo finale è quello di ricavare statistiche e dashboard descrittive, diagnostiche, predittive e prescrittive, dalle quali non solo estrapolare conoscenza sui dati esistenti ma, in alcuni casi, effettuare predizioni futuri. L'accuratezza dell'analisi effettuata è uno dei molteplici dati da tenere in considerazione quando si lavora con questi strumenti, al fine di dare il giusto peso alla conoscenza dedotta. Quest'ultima non va mai intesa come dato certo, bensì una guida e un'informazione aggiuntiva sull'andamento futuro delle cose che si stanno analizzando.

Gli strumenti utilizzati nell'ultimo obiettivo posto in essere sono molteplici, a partire dai macro-argomenti, come la data analytics, l'intelligenza artificiale, fino ad arrivare a metodi come la classificazione, la clusterizzazione, etc..

Per poter realizzare gli obiettivi sopra elencati, sarà necessario partire da alcuni specifici "task". In particolare, essi sono:

- Definizione dei materiali rilevanti. Questa fase è di competenza degli uffici con i giuristi.
- Definizione di un insieme di passaggi con la relativa applicazione nelle sentenze. Ciò è finalizzato alla preparazione dei file su cui applicare il Natural Language Processing, l'intelligenza artificiale e la data analytics. Questo per soddisfare il secondo obiettivo, cioè creare una banca dati in Akoma Ntoso.
- Analisi giuridica dell'utilizzo dell'intelligenza artificiale nell'ambito della giustizia, per non ledere i diritti costituzionali, le norme di procedura civile e penale, e i diritti fondamentali, come, ad esempio, la privacy.
- Analisi dei criteri di anonimizzazione/pseudoanonimizzazione di documenti, come le sentenze, e dei dati, per consentire agli strumenti di intelligenza artificiale di operare senza incorrere in rischi di de-identificazione, profilazione e violazione della normativa in materia di protezione dei dati personali.
- Definizione dei sistemi di AI in modo tale che non si basino su bias cognitivi, per evitare discriminazioni di qualsiasi genere.

### 1.6.3 Strumenti di formazione delle competenze per l'efficacia dell'UPP

Si vuole sviluppare un corso di alta formazione accreditato presso le università consorziate, erogato anche in modalità *e-learning*, il quale punti a fornire tutte le competenze necessarie per il passaggio che si vuole affermare tramite la trasformazione digitale. Questo corso considera il supporto necessario per gli UUGG e per le nuove figure professionali che lavoreranno nell'UPP.

Un ulteriore corso formativo che si ritiene altresì fondamentale effettuare è quello destinato al personale degli uffici giudiziari, anche per apportare una miglioria nella gestione delle risorse umane, affinché il piano di assunzioni volga ad un costante miglioramento nelle professionalità richieste. Anche questo tipo di corso verrà erogato tramite le università partner del progetto; in tale contesto si vuole focalizzare l'attenzione sull'interazione fulcro di tecnologia e diritto.

#### 1.6.4 Modelli organizzativi e dei processi per l'arretrato

Questa azione è attuabile a partire da un'analisi della situazione attuale dei sistemi informativi presenti nei diversi uffici giuridici del territorio, per poi concentrarsi sulla modellizzazione dei flussi informativi specializzati sulla base del lavoro che quella sezione ha il compito di svolgere, concentrandosi su ciò in cui va applicata la dematerializzazione, finalizzata al miglioramento dell'organizzazione stessa. Il tutto permetterebbe di velocizzare la digitalizzazione e l'integrazione delle firme digitali.

Gli strumenti tramite i quali è possibile portare a compimento ciò che è stato illustrato sono molteplici. Sicuramente alla base ci devono essere metodologie quantitative e qualitative di ricerca, per poter analizzare l'integrazione delle modifiche che si vogliono attuare nei sistemi esistenti.

Una delle applicazioni dirette di questa azione è l'utilizzo dell'AI per bilanciare il carico di lavoro nell'agenda delle udienze del giudice, tramite il calcolo del "peso" di un fascicolo, evitando, così, ritardi rispetto a ciò che si era programmato di svolgere.

#### 1.6.5 Modelli di trasformazione digitale avanzata e delle interfacce HCI

Per poter effettuare questa azione, sono stati definiti alcuni obiettivi. Tali obiettivi sono i seguenti:

- *Interfacce Uomo-macchina e UX/UI.* Miglioramento nei tempi di ricerca nelle ridondanze in termini spaziali, inefficienza e imprecisione delle interfacce uomo-macchina.
- *Drafting assistito delle sentenze.* Utilizzo di template specifici, nel formato XML standard (Akoma Ntoso), per l'aiuto della scrittura delle sentenze, tramite tecniche di Natural Language Processing, intelligenza artificiale, semantic web, etc. al fine di velocizzare il processo di redazione della sentenza stessa e, soprattutto, di renderla strutturata a livello informatico; in questo modo sarà più facile estrapolare le informazioni su cui poi applicare le varie analisi con le tecniche sopra accennate.
- *Tassonomia ontologica.* Creazione di un prototipo minimo di ontologia giuridica dell'UPP per poter classificare in modo standard diverse cose, come, ad esempio:
  - documenti;
  - processi;
  - eventi;
  - parametri dell'analisi processual-civilistica/penalistica.

Questa idea si basa sul voler incentivare il ragionamento giuridico automatico tramite questionari e velocizzare le ricerche.

- *Dematerializzazione.* Individuazione di aree in cui poter applicare la dematerializzazione. Definizione di clustering dei casi simili e della pesatura dei fascicoli al fine di migliorare la pianificazione delle udienze, come precedentemente illustrato.
- *Legal Design.* Semplificazione della comunicazione e creazione di strumenti per la volontaria giurisdizione, finalizzati alla decongestione del carico della tutela giurisdizionale.

### 1.6.6 Attivazione delle sperimentazioni dei modelli organizzativi, giuridici, tecnologici

In questa azione si vuole creare la *task force* che effettui al meglio la fase di progettazione basata sull'individuazione delle problematiche per l'implementazione di *Uni 4 Justice*. Questo, partendo dalla strutturazione di borse di studio incentrate su diversi ambiti, ovvero:

- lavoro agile;
- digitalizzazione;
- funzionamento delle cancellerie a fronte della trasformazione digitale.

### 1.6.7 Monitoraggio delle sperimentazioni dei modelli organizzativi, giuridici, tecnologici

Questa azione si sviluppa tramite diversi passaggi. Essi sono:

- Pianificare, attuare e valutare la sperimentazione dei prototipi sviluppati precedentemente presso gli UGG e raccogliere in modo sistematico le rilevazioni dei risultati.
- Monitoraggio permanente delle sperimentazioni, integrando anche il controllo di gestione e integrandolo con il controllo della governance, anche usando diagnostiche digitali.
- Modellazione dei processi e organizzazione del lavoro nel nuovo assetto della sperimentazione per una giustizia digitale, che utilizzi anche tecniche di BPMN per favorire il flusso dei documenti digitali, seguendo workflow personalizzabili da ogni ufficio, a seconda del contesto; il tutto è finalizzato alla riduzione dei tempi, all'eliminazione dei tempi morti, alla riduzione della duplicazione dei lavori e della ridondanza delle operazioni.

### 1.6.8 Ridefinizione modelli formativi permanenti di eccellenza

In questo contesto si vuole considerare l'esperienza di eccellenza e la leadership europea del beneficiario del progetto, nonché, il tema della formazione degli operatori della giurisdizione.

In particolare, si vuole fornire uno strumento che sia in costante aggiornamento al fine di dispendere contenuti formativi, sul modello *blended training*, per i funzionari dell'UPP e per gli operatori della soluzione extra giudiziale delle controversie.

Infine, con questa azione, si prevede la creazione di un progetto di percorso formativo di eccellenza a livello europeo nel contesto del mondo digitale applicato alla giustizia. Per fare ciò, sarà necessario avvalersi delle linee di finanziamento adeguate *sub Horizon Europe*.

### 1.6.9 Formazione alle competenze digitali e alla comunicazione attiva

L'idea con cui viene applicata questa azione si concretizza tramite diverse sotto azioni. Quelle di supporto in questo contesto sono le seguenti:

- Dematerializzazione alla luce di una chiave processual-civilistica e processual-penalistica delle sperimentazioni effettuate.
- Uso degli strumenti diagnostici di AI e loro introduzione nei sistemi informatici ordinari.

- Uso di strumenti di redazione delle sentenze e loro introduzione nei sistemi del DGSIA.
- Uso di banche dati avanzate per agevolare la circolazione delle informazioni e per supportare l'*information retrieval* documentale.
- Definizione di strumenti innovativi di formazione basati su laboratori di data science per gli operatori di giustizia, finalizzati a creare quelle competenze necessarie a dialogare con strumenti sofisticati nella lettura quantitativa e qualitativa dei dati.
- Definizione di laboratori di Legal Design per creare le competenze di comunicazione visuale per una giustizia inclusiva, etica, giusta, anche in favore di un'interazione efficace tra diversi stackholder.

#### 1.6.10 Reclutamento dello staff

Quest'ultima azione si pone come obiettivo il reclutamento di personale competente e specializzato. Ciò verrà effettuato tramite diverse strategie, a partire da un'analisi dei profili curriculari dei neo-laureati e dei profili di competenze richieste dal progetto stesso, proponendo temi di sviluppo per la tesi finale, per poi concentrarsi sul ricercare ciò che si necessita, tramite borse di studio senior.

I criteri per le richieste e per l'analisi dell'offerta verranno sviluppati da alcune commissioni, tenendo in considerazione i diversi fattori che caratterizzano le molteplici università, ovvero:

- pluralità disciplinare;
- pluralità degli atenei;
- coinvolgimento nel pre-screening delle candidature degli uffici giudiziari, in un'ottica di elaborazione delle competenze necessarie per i bisogni funzionali di ricerca in situ;
- settori di approfondimento della ricerca;
- competenze linguistiche.

---

### Uno sguardo alla situazione sul campo

---

*In questo capitolo verrà delineata la situazione attuale della Corte di Appello di Ancona. Inizialmente verrà descritto l'incontro che abbiamo avuto con il Presidente della Corte di Appello di Ancona, il quale ci ha permesso di avere una visione d'insieme delle esigenze digitali che ha la Corte stessa nello specifico. Ci è stato, poi, permesso di entrare nelle cancellerie civili e penali. Ciò ci ha consentito di osservare la situazione attuale delle due cancellerie, le criticità esistenti e qualche accenno alle possibili soluzioni.*

*L'obiettivo di questo capitolo è quello di illustrare, nello specifico, la situazione attuale, al fine di rendere possibile una visione concreta dei passi successivi da effettuare per poter applicare il Progetto Uni 4 Justice.*

## 2.1 Incontro con il Presidente della Corte di Appello di Ancona

Uno dei punti di partenza su cui il Progetto Uni 4 Justice fa affidamento è l'analisi della situazione di partenza delle Corti di Appello che hanno aderito al progetto. Per questo motivo, è stato necessario incontrare il Presidente della Corte di Ancona, Dott. Luigi Catelli, il quale ci ha illustrato le esigenze informatiche della Corte stessa.

Tali esigenze nascono dalla struttura informatica sulla quale si basa la Corte, in quanto, diversi anni fa, sono stati definiti dei sistemi informativi per la gestione interna di tutte le Corti.

### 2.1.1 Situazione informatica e digitale attuale

Il discorso si è inizialmente concentrato sullo stato di digitalizzazione, nonché sull'utilizzo di un gestionale informatico, il quale utilizzo è regolamentato dal *Ministero della Giustizia*. I due sistemi utilizzati nella Corte di Appello sono i seguenti:

- *SICID* - Sistema Informativo contenzioso Civile Distrettuale;
- *SICP* - Sistema Informativo della Cognizione Penale.

Essendo questi sistemi informativi di carattere nazionale, l'accesso dei singoli dipendenti della Corte di Appello per la risoluzione dei problemi, nei quali si imbattono durante lo svolgimento dei loro compiti, non è agevole. Inoltre, si è riscontrato nei pareri dei tecnici informatici della Corte di Appello, che i due software risultano sì completi, ma potrebbero essere migliorati e, oltretutto, alcune funzioni risultano ridondanti, causando un ritardo nei tempi. Il personale amministrativo della Corte si esprime nei confronti dei sistemi informativi attuali con termini come "macchinoso", il quale fa trasparire un problema di base.

### 2.1.2 Esigenze di innovazione digitale

Vi è, quindi, una necessità di aumentare la condivisione dei file e delle informazioni, pur sempre rispettando i limiti imposti dal *Ministero della Giustizia*, il quale ha vietato l'utilizzo delle cosiddette "cartelle condivise" a causa della privacy. Motivo per il quale il Presidente della Corte ha evidenziato la necessità di rendere condivisi i documenti, pur rispettando la privacy, ad esempio tramite una gestione dei permessi. Inoltre, ad alto livello, ci è stata richiesta una migliore gestione informatica per l'organizzazione degli uffici giudiziari.

### 2.1.3 Contestualizzazione nel Progetto Uni 4 Justice

Il Presidente della Corte ci ha precisato che, a livello ministeriale, è sconsigliato il "fai-da-te" informatico; per questo motivo non sarebbe appropriato un intervento mirato alla singola Corte, ma un'unica "rivoluzione digitale" che impatti su tutto il territorio nazionale e che venga approvata e regolamentata tramite il *Ministero della Giustizia*. Si rivede, quindi, nel Progetto Uni 4 Justice, una possibilità di miglioramento effettivo, in quanto tale Progetto si fa portavoce di un'esigenza di digitalizzazione per le Corti di Appello con una strategia di applicazione nel mondo reale.

## 2.2 Gestione informatica della sezione civile

La sezione civile della Corte di Appello di Ancona risulta essere "a buon punto" a livello digitale. Infatti, ci è stato confermato che la maggior parte delle pratiche vengono svolte tramite il sistema informativo SICID.

### 2.2.1 Sistema informativo: SICID

SICID<sup>1</sup> è il sistema informativo utilizzato dalla sezione civile per eseguire le proprie pratiche. Esso permette:

- la gestione del contenzioso civile (SICC);
- la gestione della volontaria giurisdizione (SIVG);
- la gestione del diritto del lavoro (SIL).

Esso si struttura su diverse applicazioni, un database e un datawarehouse. Per ogni distretto, il database è caratterizzato da tre schemi:

- il primo riguardante il SICC e SIVIG;
- il secondo riguardante il SIL;
- il terzo riguarda le informazioni anagrafiche.

Ricordiamo, inoltre, che, per quanto riguarda la sezione civile, è già in uso il PCT - Processo Civile Telematico, il quale risulta essere di grande aiuto.

### 2.2.2 Problematiche e richieste del personale

L'unica richiesta del personale civile, nei confronti della situazione digitale attuale, è la possibilità di avere una banca dati per un accesso diretto e intelligente in tutte le sentenze di competenza della specifica Corte e delle altre Corti.

<sup>1</sup>[https://www.giustizia.it/giustizia/it/mg\\_1\\_12\\_1.wp?facetNode\\_1=0\\_21&facetNode\\_3=4\\_99&facetNode\\_2=3\\_1&previousPage=mg\\_1\\_12&contentId=SPS613400#](https://www.giustizia.it/giustizia/it/mg_1_12_1.wp?facetNode_1=0_21&facetNode_3=4_99&facetNode_2=3_1&previousPage=mg_1_12&contentId=SPS613400#)

## 2.3 Gestione informatica della sezione penale

Per quanto riguarda la sezione penale, fin da subito è stato chiaro che si trova in una situazione molto diversa dalla sezione civile nel contesto della digitalizzazione. Infatti, vi è un utilizzo del cartaceo che rallenta tutti i processi per l'iter che un fascicolo deve attraversare prima di essere archiviato.

### 2.3.1 Sistema informativo: SICP

Il SICP<sup>2</sup> è il sistema informativo utilizzato dalla sezione penale della Corte di Appello. Questo sistema integra tre sistemi informatici già esistenti. I sistemi integrati sono i seguenti:

- registro generale (*REGGE*);
- registro delle misure cautelari personali (*BDMC* e *MCR*).

Esso si struttura su un'unica base di dati.

### 2.3.2 Problematiche e richieste del personale

Per quanto riguarda la sezione penale, il sistema informativo non è sufficiente per le pratiche che questa sezione deve poter svolgere, in quanto risulta troppo rigido, non permettendo ai funzionari giudiziari di apportare determinate modifiche.

Le problematiche e conseguenti richieste del personale addetto si sono concentrate su dei macro argomenti, ciascuno dei quali si basa sull'idea di utilizzare degli automatismi che permettano di risparmiare tempo e passaggi inutili. I punti cardine delle richieste sono i seguenti:

1. Inserimento automatico dell'epigrafe di un documento, al fine di velocizzare la scrittura di documenti che hanno una struttura di base non modificabile.
2. Definizione automatica del "peso" da dare ad un fascicolo, permettendo, quindi, di definire le priorità dei diversi fascicoli.
3. Integrazione degli scadenziari degli imputati, attualmente non presente nel sistema informativo attuale, ma di prioritaria importanza per il personale amministrativo.
4. Integrazione della condivisione, nei rispetti della legge e della privacy dei dati, escludendo, quindi, le cartelle condivise.
5. Creazione del calendario dei ruoli, nonché del calendario del Presidente, ovvero quello in cui sono presenti le informazioni riguardanti gli impegni giudiziari del Presidente della Corte di Appello. Essendo, ad oggi, presente sotto forma di file *excel* non condiviso, lo si vuole condiviso e con una formattazione maggiormente fruibile.
6. Creazione di un massimario delle sentenze per i magistrati, al fine di estrapolare maggiori informazioni.
7. Integrazione della possibilità di creazione di nuovi modelli definiti dal personale amministrativo.
8. Miglioramento della gestione dei fascicoli, con l'aggiunta della conseguente decisione presa a valle della sentenza.

---

<sup>2</sup>[https://www.giustizia.it/giustizia/it/mg\\_1\\_12\\_1.wp?facetNode\\_1=0\\_21&facetNode\\_3=4\\_99&facetNode\\_2=3\\_1&previousPage=mg\\_1\\_12&contentId=SPS613400#](https://www.giustizia.it/giustizia/it/mg_1_12_1.wp?facetNode_1=0_21&facetNode_3=4_99&facetNode_2=3_1&previousPage=mg_1_12&contentId=SPS613400#)

9. Creazione di una "cronologia" di un certo fascicolo, riguardante tutti e tre i livelli di giudizio.

I due punti di partenza di tutto ciò sono stati formalizzati ad alto livello. Essi sono i seguenti:

- Realizzazione di un database aggiornato contenente i fascicoli dalla loro iscrizione alla loro definizione.
- Definizione del calendario contenente i dati del procedimento, la data di udienza, la rilevanza giuridica, l'impegno ("peso") e il nominativo del consigliere relatore.

La richiesta, ad un maggior livello di dettaglio, è stata formalizzata attraverso una definizione di ciò che si vorrebbe venga realizzato. I punti che si vorrebbe fossero soddisfatti sono i seguenti:

- Numeri di ruolo del procedimento RGNR (Registro Generale Notizie di Reato) e RG APP (Registro generale Appello).
- Tipologia del rito (dibattimento o camera di consiglio).
- Nominativi e dati dell'imputato (o degli imputati) e delle eventuali parti civili.
- Indicazione delle parti appellanti.
- Nominativi dei difensori.
- Sussistenza o meno di misure cautelari (con specificazione degli imputati cui si riferiscono) e loro tipologie.
- Data di scadenza delle misure cautelari.
- Indicazione del giudice di primo grado.
- Indicazione del numero e data della sentenza di primo grado.
- Sezioni specializzate 1 e 2.
- Indicazione dei magistrati che compongono il Collegio giudicante e del Consigliere d'appello relatore.
- Trascrizione integrale del capo di imputazione.
- Indicazione delle norme di legge violate (qualificazione giuridica del fatto).
- Data di prescrizione dei reati.
- Sintesi del fatto storico dal quale il procedimento si origina, con concisa esposizione della decisione di primo grado e dei motivi di appello.
- Data e ora di fissazione dell'udienza d'appello e di quelle eventuali di rinvio.
- Rilevanza, in scala crescente da 1 a 3, del procedimento (in termini di complessità del fatto e dei relativi profili giuridici).
- Esito del procedimento in grado di appello (dispositivo).
- Numero e anno della sentenza d'appello.

- Giorni eventualmente riservati per il deposito della motivazione.
- Testo integrale della sentenza (importandola in file *.pdf*, anche mediante collegamento all'archivio attualmente esistente).
- Eventuale proposizione di impugnazioni avverso la sentenza di appello.
- Decisione della *Corte di Cassazione*.

Tutte queste informazioni dovrebbero essere condivise; quindi potrebbero essere realizzate su una piattaforma condivisa, come, ad esempio, SharePoint; i dati dovrebbero essere disponibili per tre funzioni fondamentali. In particolare, le funzioni che si vorrebbero poter applicare ai dati sopra elencati sono le seguenti:

- ricerche sulla base di uno o più dei parametri sopra indicati;
- modifiche successive nel corso del procedimento;
- estrazione automatica per la creazione di macro, utili nelle lavorazioni dei magistrati e della cancelleria (ad esempio redazione dei ruoli, elenchi di trattazione dei procedimenti di udienza; redazione della motivazione della sentenza, etc.);
- ricerca delle parole chiavi all'interno delle sentenze esistenti.

---

## Il sistema documentale: Sharepoint

---

*In questo capitolo verrà inizialmente descritto Microsoft SharePoint, a partire dal tool stesso, per poi analizzare il suo posizionamento rispetto gli altri strumenti per la gestione documentale, i suoi punti di forza e le sue opzioni di configurazione. Successivamente, verrà spiegato come può essere creato un sito con SharePoint e verranno descritti tutti gli elementi che possono essere inseriti nel sito stesso. A questo punto, verranno illustrati i ruoli che i diversi utenti possono avere; in particolare, verrà descritto il ruolo di amministratore globale di SharePoint, oltre ai ruoli di "proprietario" e "membro", con i quali gli utenti possono essere caratterizzati nei diversi siti, nonché i permessi che questi ruoli concedono agli utenti, nei confronti della gestione delle risorse dei siti, quindi, anche della gestione dei documenti. Il capitolo, infine, si concluderà con la spiegazione del campo di azione della funzione di ricerca di SharePoint e di come questa possa essere personalizzata.*

*L'obiettivo di questo capitolo è quello di mostrare le potenzialità di SharePoint, al fine di ragionare sull'utilizzo di questo tool per attuare alcune delle modifiche proposte dal progetto Uni 4 Justice, oppure, per valutare un suo possibile utilizzo per colmare alcune lacune degli attuali sistemi informativi della Corte di Appello di Ancona.*

### 3.1 Panoramica

*Microsoft SharePoint* (Figura 3.1) nasce nel 2001 nell'omonima azienda *Microsoft*, con l'obiettivo di ottimizzare la gestione aziendale dei documenti condivisi tramite una piattaforma software. Ad oggi questa piattaforma è utilizzata in particolar modo in ambito aziendale, in quanto viene adoperata all'interno dell'organizzazione come strumento di condivisione di informazioni e decisioni. Oltre a tale utilizzo interno, le informazioni possono essere condivise anche con utenti esterni attraverso la rete. Essendo un tool rilasciato dalla Microsoft è utilizzabile tramite l'account dell'azienda stessa ed è integrato con tutto il "mondo Microsoft" (come *Microsoft OneDrive*, *Microsoft Azure*, etc.).

SharePoint si basa sullo sviluppo di siti, ricchi di contenuti e documenti, accessibili da gruppi definiti di persone, nonché proprietari, membri e visitatori del sito stesso.

#### 3.1.1 Posizionamento competitivo

Dalla analisi dei *Content Service Platform* del 2021 riportata nel Magic Quadrant di Gartner (Figura 3.2) *Microsoft SharePoint* risulta essere leader in questo settore. Al secondo posto, anche se con un notevole distacco troviamo *Hyland* (ex *Alfresco*) che risulta essere quindi un tool altrettanto "skillato".



**Figura 3.1:** Logo Microsoft SharePoint

Mettendo a confronto questi due strumenti, che risultano essere i principali leader nel diagramma, così come riportato da un autorevole sito di analisi delle performance dei software<sup>1</sup>, si nota che SharePoint presenta le seguenti feature migliori rispetto ad Hyland:

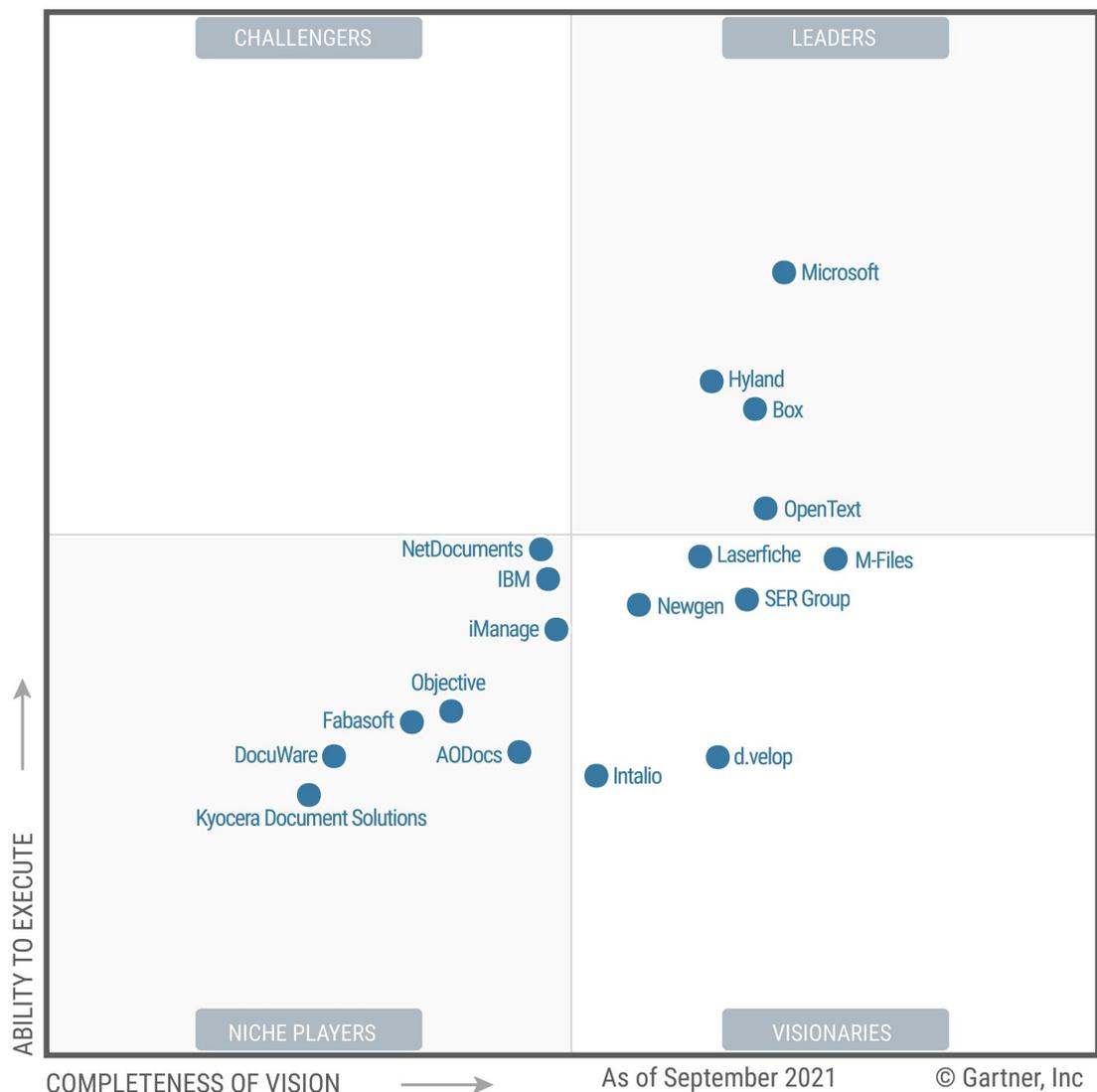
- creazione dei contenuti;
- acquisizione dei contenuti;
- aggregazione dei contenuti;
- output e presentazione dei contenuti;
- gestione del flusso di lavoro;
- controllo e gestione della versione;
- gestione della personalizzazione;
- documentazione e gestione dei registri;
- gestione della ricerca e dell'indicizzazione;
- gestione della sicurezza;
- gestione delle risorse digitali;
- report e gestione statistica.

Dal medesimo sito di comparing, Hyland risulta, invece, migliore nel *supporto al portale* e nell'*estensione di ECM*.

Le migliori performance di SharePoint sono attribuibili all'essere in cloud e all'essere prodotto da Microsoft, rendendolo una garanzia per l'utilizzatore finale. Vi è, inoltre, un problema nell'installazione della versione *Community* di Hyland che, oltretutto, presenta delle grosse limitazioni anche delle funzionalità di base, utilizzabili solo tramite la versione *Enterprise*. Per quanto riguarda l'installazione open source il bug è nel download dei pacchetti necessari, in quanto vengono installate delle librerie accessibili solo con licenze private e non gratuite, come la versione open richiederebbe, bloccando così l'intero processo di installazione.

A differenza di questa notevole limitazione, SharePoint presenta un enorme vantaggio, in quanto è possibile utilizzare direttamente l'applicazione web browser, accessibile tramite il semplice account Microsoft. Questa versione permette di effettuare tutte le operazioni possibili, ad eccezione di alcune specifiche funzionalità locali. La scelta del tool da utilizzare, quindi, è ricaduta su SharePoint, il quale si concentra sulla creazione di siti personalizzabili e condivisibili.

<sup>1</sup><https://www3.technologyevaluation.com/selection-tools/comparison-reports/17581/alfresco-vs-microsoft-sharepoint-enterprise-edition-enterprise-content-management-ecm>



**Figura 3.2:** Magic Quadrant Content Service Platform 2021

### 3.1.2 Punti di forza

SharePoint semplifica l'uso, l'organizzazione e la condivisione delle risorse, mettendo in risalto lo scambio delle idee all'interno di un team, in ogni luogo e su ogni tipo di dispositivo. Questo tool permette di condividere le best practice dei vari membri del team, migliorando la comunicazione e sfruttando appieno le conoscenze collettive, che permettono di ottimizzare le scelte future del team.

La sua nascita all'interno di Microsoft è, probabilmente, il suo punto di forza maggiore, essendo integrato con tutto il "mondo Microsoft". Questo permette l'uso di un ampio ventaglio di tool correlati, di un'ottima documentazione a cui far riferimento e, soprattutto, una grande community con la quale potersi confrontare in caso di problemi o dubbi riguardanti il tool stesso.

In aggiunta a tutto ciò, l'essere *open* non può che aumentare il suo vantaggio, essendo uno dei motivi che permettono alla community di essere in continua espansione.

## 3.2 Configurazione

Per quanto riguarda le alternative di utilizzo del software di SharePoint si presentano due diversi scenari: online o in locale. Per ciascuno di essi verrà fornito un approfondimento nel seguito della trattazione.

### 3.2.1 Online

L'applicazione web browser online, come già detto, è accessibile semplicemente con un account Microsoft365 o tramite un account SharePoint Online. In questo scenario, il servizio offerto si basa sull'utilizzo del cloud il che permette di rendere accessibile la piattaforma da qualunque tipo di device indipendentemente dalla geolocalizzazione fisica.

Per poter utilizzare l'applicazione web, sarà sufficiente accedere al sito di Microsoft Office 365<sup>2</sup>, effettuare il login, per poi cliccare sull'elenco delle applicazioni disponibili e scegliere SharePoint.

Nonostante questa versione preveda l'utilizzo della piattaforma online, vi è, comunque, la possibilità di sincronizzare le cartelle in locale sul proprio dispositivo tramite lo strumento *OneDrive*, essendo anch'esso all'interno dell'universo *Microsoft*, il che permette un'ulteriore comodità nello scambio e utilizzo dei file.

### 3.2.2 In locale

Per quanto riguarda l'utilizzo del tool in locale, si tratta di *Microsoft SharePoint Server*, installabile solo su dispositivi con Windows Server, oppure in un Datacenter, il che lo rende di ridotto utilizzo rispetto alla versione online. In questo caso sono necessari dei prerequisiti di sistema; infatti per l'installazione è richiesta l'esecuzione di un file apposito che verifichi la compatibilità.

Per poter installare il software, una volta proceduto con il download di Microsoft SharePoint Server 2019<sup>3</sup>, sarà necessario seguire i seguenti passi:

- effettuare doppio click sul file immagine del disco per montarlo come unità sul computer;
- passare all'unità montata;
- eseguire `splash.hta` per avviare la home di SharePoint Server 2019;
- cliccare su *Installa prerequisiti software* per lanciare la verifica dei prerequisiti, seguendo le istruzioni fino alla fine del controllo;
- fare click su *Installa SharePoint Server* dalla pagina iniziale di SharePoint Server 2019;
- quando verrà richiesto il codice *Product Key*, se non si è in possesso di una licenza, sarà possibile richiederla gratuita per 180 giorni durante i quali testare il software; passati questi giorni si potrà valutare l'acquisto o meno della licenza.

Successivamente sarà necessario installare Project Server 2019 (passaggio eseguibile solo dall'amministratore globale di Sharepoint dell'organizzazione):

- Se non appare la configurazione guidata, fare click su *Start* → *Tutte le app* → *Microsoft SharePoint Product* → *Configurazione guidata SharePoint Product*; se si apre la finestra di dialogo *Controllo account utente*, allora fare click su *Continua*.

<sup>2</sup><https://www.microsoft.com/it-it/microsoft-365/microsoft-office>

<sup>3</sup><https://www.microsoft.com/en-us/download/details.aspx?id=57462>

- Nella pagina *SharePoint Product* fare click su *Avanti*.
- Nella finestra di dialogo che notifica che potrebbe essere necessario riavviare alcuni servizi durante la configurazione, fare click su *Sì*.
- Nella pagina *Connessione a una server farm* fare click su *Crea una nuova server farm* e, poi, su *Avanti*.
- Nella pagina *Specifica impostazioni database di configurazione* eseguire le operazioni seguenti:
  - nella casella *Server database* digitare il nome del computer che esegue SQL Server;
  - nella casella *Nome database* inserire un nome per il database di configurazione o utilizzare il nome di database predefinito (*SharePoint\_Config*);
  - nella casella *Nome utente* digitare il nome utente dell'account di servizio amministratore della farm. Accertarsi di usare il formato `DOMINIO\nomeutente`;
  - nella casella *Password* digitare la password dell'utente;
  - fare click su *Avanti*.
- Nella pagina *Specifica impostazioni di sicurezza della farm* digitare una passphrase (come una password ma più lunga per questioni di sicurezza) e quindi fare click su *Avanti*.
- Nella pagina *Configurazione applicazione Web Amministrazione centrale SharePoint* attenersi alla procedura seguente:
  - selezionare la casella di controllo *Specificare il numero di porta* e digitare un numero di porta se si desidera che ne venga utilizzato uno specifico nell'applicazione Web Amministrazione centrale SharePoint, oppure lasciare deselezionata la casella di controllo *Specificare il numero di porta*, se si desidera utilizzare il numero di porta predefinito;
  - fare click su *NTLM* o su *Negozia (Kerberos)*.
- Fare click su *Avanti*.
- Nella pagina *Completamento di Configurazione guidata SharePoint Product* esaminare le impostazioni di configurazione per verificare che siano corrette e poi fare click su *Avanti*.
- Nella pagina *Configurazione completata* fare click su *Fine*. Dopo la chiusura della procedura guidata, verrà aperto il Web browser e verrà eseguita la connessione ad Amministrazione centrale.

Infine abilitare Project Server 2019; per fare ciò:

- installare *SharePoint Server 2019* utilizzando un codice Product Key Enterprise;
- creare una *farm* di *SharePoint Server 2019*;
- avviare la shell di gestione di *SharePoint 2019*;
- eseguire il comando seguente in *SharePoint 2019 Management Shell*:  
`Enable-ProjectServerLicense-Key<chiave_licenza_server_progetto>`

Questa versione in locale ha delle funzionalità aggiuntive, come, ad esempio, l'integrazione totale e diretta con altri tools (PowerBI, Power Apps, etc.). Passando, invece, al servizio in cloud, solo alcune funzionalità di questi tool possono essere utilizzate nei siti; quindi, si ottiene un'integrazione solo parziale.

A valle dell'analisi delle due alternative, la versione online risulta essere quella più efficiente ed efficace, essendo gratuita e senza limiti per l'utilizzo. Inoltre, essendo in cloud, vi è la possibilità di aumentare la capacità delle risorse e, oltretutto, viene rispettato il concetto di base del "mantenere intatta la macchina reale", apportando modifiche solo nel cloud, che possono essere annullate facilmente.

### 3.3 Sito in SharePoint

La visualizzazione, gestione e condivisione dei contenuti in SharePoint si basa sul concetto di *sito*, la cui modalità di accesso dipende dal ruolo che il proprietario associa ad ogni utente aggiunto al gruppo del sito stesso.

Esistono diversi approcci per la creazione di un sito, sulla base di quali utenti si vogliono includere nella gestione e/o visualizzazione dei contenuti. Vi sono diversi scenari possibili:

- *Sito del team*, lo si utilizza quando c'è la necessità di lavorare in uno o più team ad uno specifico progetto in quanto quasi tutti i membri potranno gestire i contenuti e le risorse. Quindi, lo scopo è quello di far collaborare degli specifici utenti scelti per creare contenuti. Vi è la possibilità di modificare le impostazioni di privacy; queste possono essere:
  - *privato* (default), in cui solo i membri autorizzati del sito possono accedere;
  - *pubblico*, dove qualsiasi utente dell'organizzazione può accedere al sito.
- *Sito di comunicazione*, quando ci si rivolge ad un ampio numero di utenti che devono poter visualizzare i contenuti, mentre la loro gestione è riservata ad un numero ristretto di utenti. Questo metodo lo si utilizza per divulgare informazioni e coinvolgere più persone per la lettura e non per la creazione di contenuti, motivo per il quale è consigliato per le presentazioni dei servizi offerti dall'organizzazione o su uno specifico progetto aziendale.

Una buona regola per centrare il concetto del sito, nonché ottimizzare la ricerca e lo scambio di informazioni, è quella di ridurre il tempo necessario impiegato dal lettore per trovare ciò che sta cercando. Questa idea si basa sul ridurre il contenuto al minimo necessario per capire il fulcro del discorso fin dalle prime informazioni riportate, minimizzando il tempo di ricerca anche a discapito dei click da effettuare per raggiungere l'informazione completa.

È importante che un sito sia personalizzato anche a livello visivo, in quanto spesso è più semplice far passare un messaggio in modo intuitivo, tramite dei dettagli sul tipo di scrittura e il colore. Nel caso in cui si volesse realizzare un sito per un importante marchio, il quale presenta degli specifici colori che richiamano il logo aziendale, sarà opportuno che il sito stesso si attenga il più possibile ai medesimi colori a cui il marchio fa riferimento.

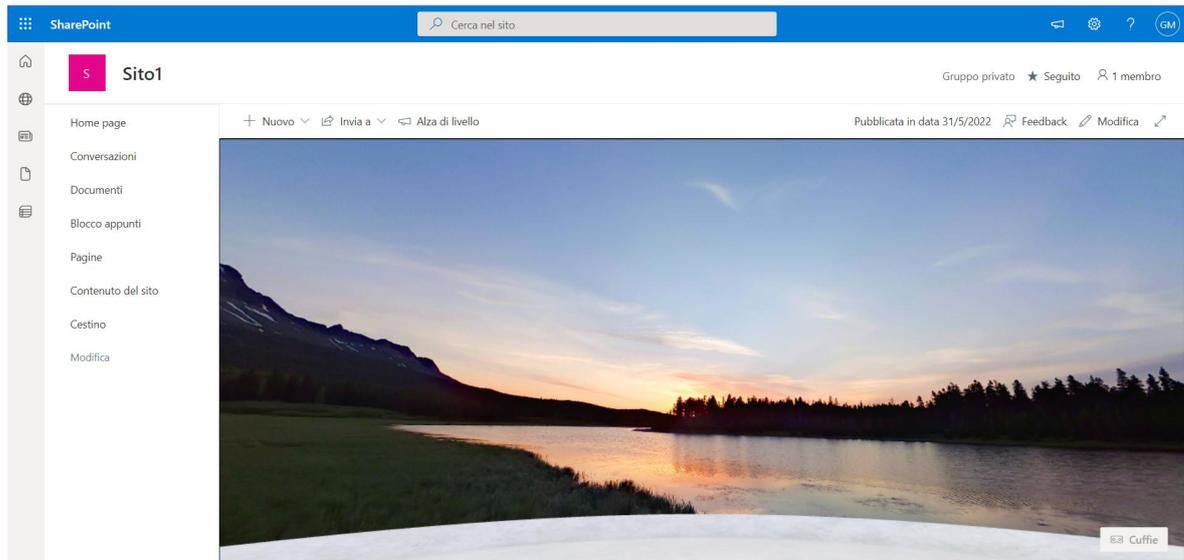
Per ogni "sito del team" ci sono una serie di alternative per poterlo personalizzare; tramite l'icona in alto a destra si valuteranno i "passaggi successivi" per migliorare il sito. Questi possono essere:

- *Applicare un modello di sito*, utile per ottimizzare il processo di creazione, scegliendo il modello che più si avvicina a ciò che si vuole condividere tramite il sito che si sta creando. Il modello può essere scelto sulla base di quelli proposti da Microsoft, oppure può essere creato dall'organizzazione, così da velocizzare il processo di creazione, pur mantenendo degli standard efficaci per la comunicazione aziendale.

- *Invitare altri membri del team*, utile per poter aggiungere altri utenti in un secondo momento. Se, ad esempio, entrasse a far parte dell'organizzazione una nuova persona che viene associata al team che si sta già occupando di un progetto, oppure se il progetto dovesse espandersi, potrebbe essere necessario integrare altre persone o, addirittura, altri team.
- *Caricare file*, shortcut per iniziare ad inserire delle risorse di tipo documentale nel sito, al fine di velocizzare il processo di creazione.
- *Condividere un post di notizie*, ulteriore shortcut per popolare sin da subito il sito, così da fornire già inizialmente qualche informazione di base aggiuntiva agli utenti che sono stati aggiunti al sito, che noteranno queste prime informazioni.
- *Cambiare l'aspetto*, scegliendo il tema, il layout, lo sfondo, il logo e l'orientamento di navigazione del sito; il miglioramento visivo del sito potrebbe aiutare nel trasmettere in maniera più chiara e intuitiva il messaggio che esso vuole far passare, in quanto, sulla base della grandezza e dei colori dei caratteri, viene posta maggiormente l'attenzione su un'informazione piuttosto che su un'altra.

Una volta scelto il tipo di sito (di team o di comunicazione) che si vuole realizzare, si possono iniziare ad inserire i contenuti, tramite l'aggiunta di un nuovo elemento tra le seguenti alternative:

- *Elenco*, che può essere creato da zero, caricato da Excel, da un elenco esistente, oppure può essere scelto uno specifico tipo di modello (registro delle assunzioni, monitoraggio progressivo del lavoro, unità di pianificazione del contenuto, etc.).
- *Raccolta documenti*, che consiste nella creazione di una sottosezione all'interno del sito.
- *Pagina*, per permettere la creazione di una vista alternativa alla pagina iniziale del sito che abbia un contesto ben definito.
- *Spazio*, verrà creato uno spazio con una visualizzazione a 360° all'interno di una nuova pagina (Figura 3.3), e verrà richiesto il tipo di struttura (raccolta, anfiteatro, arco, etc.) sulla quale posizionare il punto di vista. Successivamente, sarà possibile passare alla progettazione dello spazio, scegliendo uno sfondo (tra quelli predefiniti o scegliendo un file a 360° personalizzato) e inserendo un messaggio audio di benvenuto e un suono ambientale di sottofondo.
- *Post di notizie*, che porta alla creazione di una notizia che viene segnalata agli utenti come anteprima nella home page.
- *Collegamento notizie*, che permette di riportare una notizia esistente, senza doverla ricreare.
- *Piano*, che crea una sezione aggiuntiva che permette di mantenere lo stato di avanzamento del team, tramite una bacheca in cui aggiungere le attività (associate a determinati utenti e con le relative scadenze) e dei grafici che aiutano a capire i progressi raggiunti rispetto al piano.
- *App*, le quali possono essere aggiunte dall'elenco delle applicazioni disponibili nello Store per un sito; queste applicazioni sono state create e aggiunte dagli utenti e possono essere utilizzate a supporto del contenuto che si vuole inserire nel sito; un esempio di app è Power BI Collaboration, un semplice plug-in che permette di utilizzare alcune delle funzioni di questo tool, potendo, così, inserire i diagrammi ottenuti con Power BI nel sito stesso.



**Figura 3.3:** Esempio di uno spazio con immagine 360°

Il sito, una volta realizzato, verrà posizionato nella lista dei "siti personali" dove saranno presenti tutti i siti creati dall'utente e quelli dove l'utente è stato aggiunto finora.

Nella home page di SharePoint si trovano le ultime notizie dai vari siti personali al fine di mantenere costantemente aggiornato l'utente sulle novità, e un elenco dei siti maggiormente frequentati per poter velocizzare le varie operazioni.

Tramite le impostazioni del sito sarà possibile abilitare una o più lingue predefinite per i contenuti del sito stesso, così che, se un componente del gruppo necessita di visualizzare le informazioni in un'altra lingua, gli basterà selezionare la lingua desiderata al fine di velocizzarne la comprensione del testo.

Vediamo, nella Figura 3.4, come si presenta un esempio di sito in cui sono stati inseriti quattro documenti (nello specifico progetti universitari), che si possono vedere nella parte in basso a destra dell'immagine, due notizie di prova al centro della home page (la prima contenente il *lorem ipsum*, testo spesso utilizzato per il riempimento fittizio di spazio al fine di simulare un testo, la seconda con le informazioni di default di SharePoint); sopra la sezione documenti sono presenti alcuni collegamenti rapidi creati di default e, infine, le attività nella parte bassa dell'immagine. In quest'ultima sezione descritta sono riportate le ultime attività eseguite sul sito; in particolare, al primo posto vediamo la creazione di uno spazio (vedi Figura 3.3), mentre, di seguito, sono riportati gli inserimenti delle due notizie: "Prima Notizia" e "Seconda Notizia".

Come possiamo notare, nella parte in alto a destra del sito, vi troviamo le informazioni del sito stesso: caratteristica di sicurezza (gruppo pubblico o privato), se l'utente sta "seguendo" il sito e il numero di membri appartenenti al gruppo (cliccando si potranno vedere i nomi degli utenti che hanno accesso ed il loro ruolo). Tramite la barra verticale a sinistra è possibile spostarsi nelle diverse sezioni. Queste ultime sono le seguenti:

- *home page*, pagina iniziale del sito;
- *conversazioni*, collegamento diretto con *Outlook*, con un'anteprima di conversazione vuota effettuabile via mail i cui destinatari saranno gli utenti del gruppo del sito;
- *documenti*, sezione apposita per l'elenco di documenti del sito, con la possibilità di visualizzare, modificare o eliminare quelli presenti, oppure caricarne di nuovi;

- *blocco appunti*, collegamento diretto con *OneNote*, con la creazione di un blocco di appunti vuoto;
- *pagine*, elenco delle pagine del sito (Figura 3.5);
- *contenuto del sito*, dove sono presenti elenchi, raccolte e altre app, oltre al flusso di lavoro del sito, alle impostazioni e al cestino;
- *cestino*, per accedere agli elementi cancellati; esso ha una struttura a doppio livello per evitare ogni tipo di errore nella cancellazione dei file potendoli, così, ripristinare;
- *modifica*, per modificare i contenuti di questo pannello laterale.

Nella parte alta del sito si possono osservare altre opzioni, ovvero:

- *nuovo*; cliccando si aprirà un elenco selezionabile in cui sarà possibile scegliere il nuovo elemento che si vuole aggiungere nel sito;
- *dettagli pagina*; contiene informazioni di alto livello della pagina (nome, data e ora ultima modifica);
- *analisi*; si aprirà sulla destra un insieme di analisi relative agli accessi al sito; quelle di default sono: *visualizzatori di pagine*, *visualizzazioni della pagina*; *tempo medio impiegato per utente* (Figura 3.6) e *traffico delle pagine in base al tempo* (heatmap giornaliera e oraria in Figura 3.7);
- *pubblicata*; contiene la data di pubblicazione;
- *modifica*; consente di apportare delle modifiche.

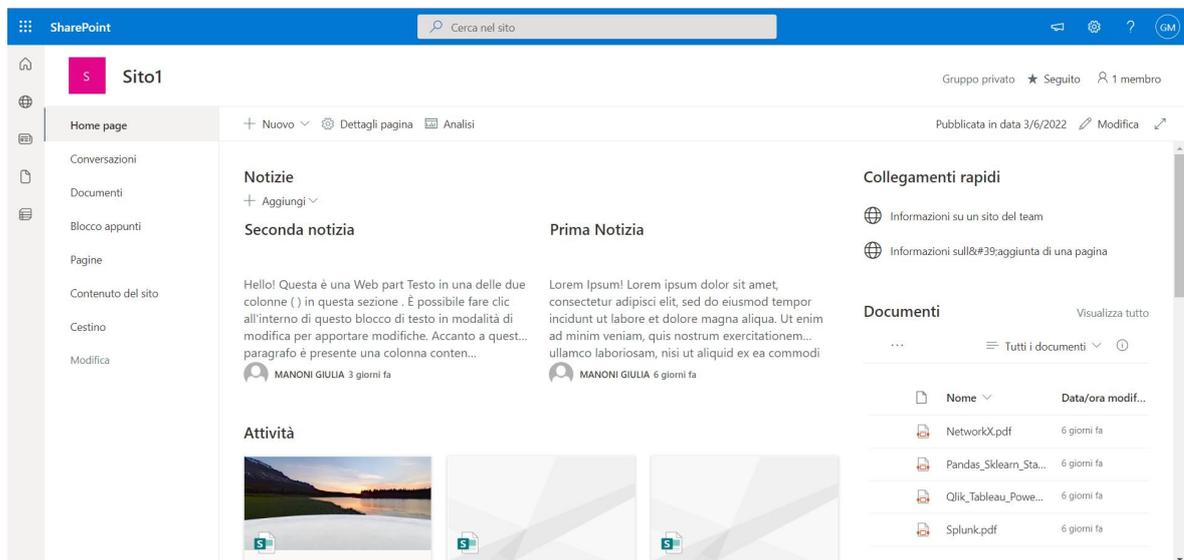


Figura 3.4: Home page *Sito1*

Nome	Modificato da	Data/ora modif...	Data/ora creazi...
Autore - Account di sistema (1)			
Home.aspx	Account di sistema	22 maggio	22 maggio
Autore - MANONI GIULIA (3)			
Lorem-ipsam.aspx	MANONI GIULIA	6 giorni fa	6 giorni fa
Seconda-notizia.aspx	MANONI GIULIA	martedì alle 15:19	martedì alle 15:18
Primo-spazio.aspx	MANONI GIULIA	martedì alle 15:23	martedì alle 15:20

**Figura 3.5:** Sezione *Pagine*

### 3.3.1 Ruolo amministratore

L'amministratore globale di Microsoft SharePoint, ovvero l'amministratore dell'azienda, ha in automatico il ruolo di amministratore di SharePoint, ma ha anche la possibilità di autorizzare come amministratore di SharePoint altri utenti selezionati per poter gestire in autonomia dei compiti che sarebbero bloccati senza la costante richiesta e autorizzazione da parte dell'amministratore globale. Un esempio di ciò è la gestione della personalizzazione della ricerca intelligente aggiunta dagli sviluppatori di SharePoint, ma utilizzabile e configurabile in un sito solo tramite l'utente amministratore di SharePoint.

### 3.3.2 Gestione e permessi dei membri del team

Il creatore del sito sarà automaticamente "settato" come proprietario del sito stesso e nel caso in cui venga creato un "sito del team privato" gli verrà richiesto di aggiungere gli utenti al gruppo con il relativo ruolo. I ruoli possibili sono i seguenti:

- *Membro*, ha un controllo limitato, in quanto può aggiungere, modificare ed eliminare elenchi, raccolte di documenti, membri del sito, documenti, pagine, post di notizie, sezioni di una pagina, web part, navigazione del sito, nonché creare o eliminare i modelli di pagina e visualizzare le metriche di utilizzo del sito.
- *Proprietario*, ha un controllo completo; quindi, oltre a tutto ciò che può fare il membro del sito, egli può modificare il tema e il logo del sito, cambiare il layout di spostamento, aggiungere o rimuovere proprietari e visitatori, modificare le impostazioni del sito e le impostazioni dei membri del sito, eliminare il sito, aggiungere un gruppo di *Microsoft365* e associare un sito a un hub (connessione tra siti sulla base dei loro ambiti).

Se il sito viene creato come sito di comunicazione, c'è un ulteriore ruolo associabile all'utente aggiunto nel gruppo, ovvero quello di *visitatore*; alla persona associata a questo ruolo viene permessa solo la visualizzazione dei contenuti senza la possibilità di apportare modifiche.

### 3.3.3 Condivisione e manipolazione dei documenti

All'interno di ogni sito c'è la sezione *Documenti* tramite la quale si possono caricare i file di interesse per quel sito. Ad esempio, è possibile vedere i documenti caricati nel Sito1 in Figura 3.8.



Figura 3.6: Elenco delle *analisi*

Se il sito è stato creato come "sito del team", allora i documenti saranno manipolabili da tutto il gruppo a prescindere dal ruolo assegnato; invece, se il sito è di comunicazione, i documenti saranno gestibili dai proprietari e membri del gruppo del sito, mentre i visitatori potranno solo visualizzarne il contenuto. Questa sezione è caratterizzata dalla lista dei documenti caricati nel sito, con i vari dettagli relativi al singolo file (data e ora di modifica, autore della modifica, etc.) che sono personalizzabili tramite le relative impostazioni, ad esempio ordinando le colonne o aggiungendone di nuove. Le impostazioni avanzate, per quanto riguarda questa vista, includono la definizione dell'URL del documento, la scelta del comportamento di apertura dei documenti da parte del browser, e tante altre opzioni. Tramite l'utilizzo di *OneDrive* è possibile sincronizzare questa cartella di documenti nel proprio computer, al fine di mantenere allineate le copie locali con quelle virtuali.

## 3.4 La ricerca

Lo scopo principale di un sito in SharePoint è quello di mantenere aggiornati gli utenti sul contenuto, quindi sulle informazioni dei documenti condivisi e sulle notizie aggiunte dai vari membri del sito stesso. L'obiettivo è quello di permettere una ricerca avanzata sull'intero sito, partendo dai singoli *documenti*, fino ai *post di notizie*.

### 3.4.1 Campo di azione

La ricerca ha un campo di azione che dipende da dove ci si trova nel momento della ricerca. In particolare, se ci troviamo nella *home page* del sito, la ricerca sarà effettuata nell'intero sito (Figura 3.9). Invece, qualora si effettuasse una ricerca trovandosi in una sezione particolare, ad esempio nei *documenti*, verrà esplorata solo la sezione del sito dove è stata effettuata la

## Traffico delle pagine in base al tempo

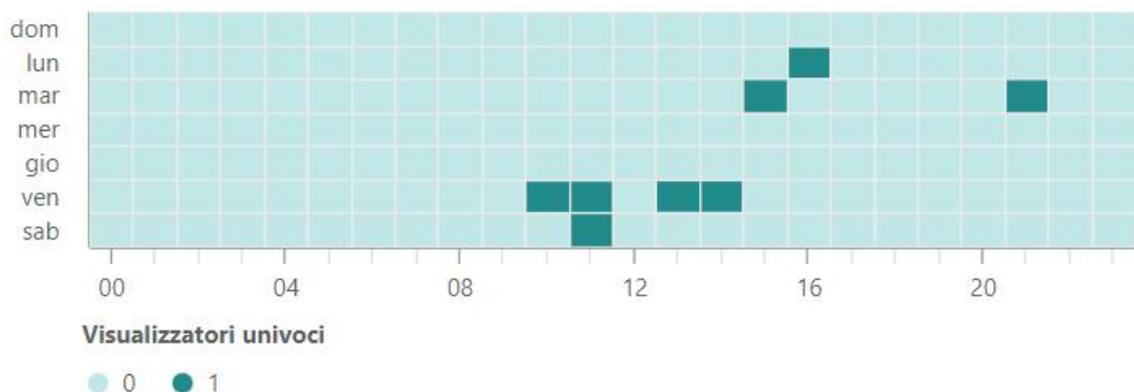
Ultimo **7** 30 90 giorni

Figura 3.7: Heatmap giornaliera e oraria degli accessi

Nome	Data/ora modif...	Modificato da	+ Aggiungi colonna
NetworkX.pdf	2 giorni fa	MANONI GIULIA	
Pandas_Sklearn_Statsmodels.pdf	2 giorni fa	MANONI GIULIA	
Qlik_Tableau_PowerBI.pdf	2 giorni fa	MANONI GIULIA	
Splunk.pdf	2 giorni fa	MANONI GIULIA	

Figura 3.8: Esempio relativo alla sezione *Documenti*

ricerca. In aggiunta, è possibile effettuare la ricerca dalla propria *home page* di SharePoint, per estendere il campo di azione a tutti i siti di cui un utente fa parte.

Sia nel caso in cui venga effettuata una ricerca ad alto livello, sia nel caso in cui riguardi un sito specifico, la ricerca è fatta all'interno dei file che vengono caricati nel sito, come possiamo vedere nell'immagine in Figura 3.9, dove viene ricercata la parola "diagramma" e viene trovata all'interno dei file *.pdf* che sono stati inseriti come risorse del sito stesso.

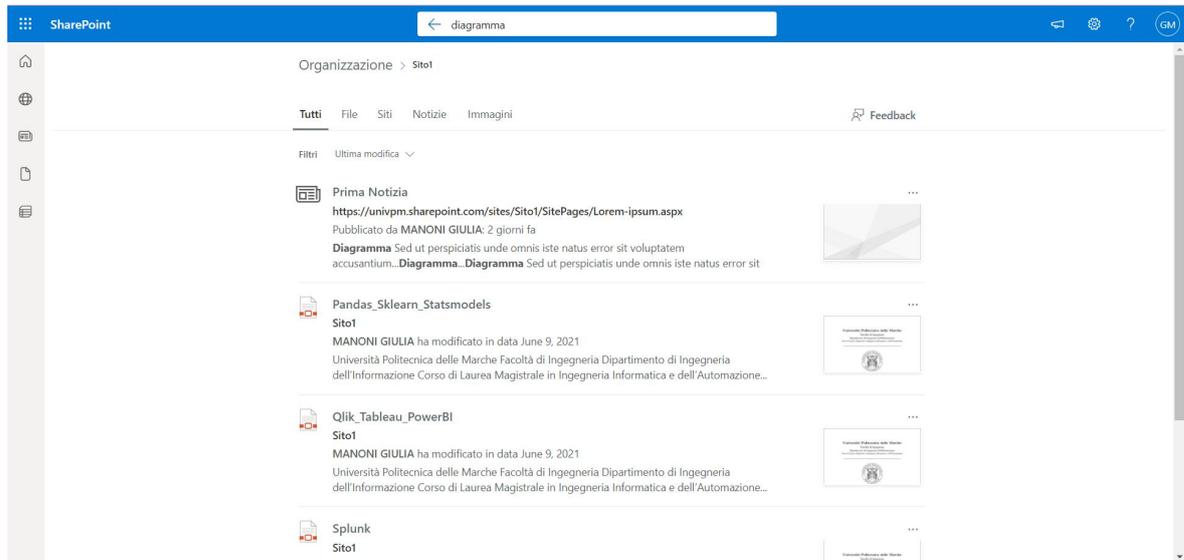
Questo tipo di ricerca risulta essere molto di aiuto per l'ispezione completa del sito; quindi, viene effettuata un'esplorazione dettagliata di ogni risorsa messa a disposizione degli utenti del sito stesso, dalle informazioni nei post e nelle varie pagine al dettaglio interno dei documenti caricati.

Inoltre, una volta effettuata la ricerca, vi è la possibilità di selezionare, tramite dei tab, *File*, *Siti*, *Persone*, *Notizie* e *Immagini*, ottenendo, così, i risultati sulla base di ciò che più interessa all'utente. Al fine di ottimizzare la ricerca che si desidera effettuare, successivamente, sarà anche possibile filtrare i risultati sulla base del tipo di file e sull'ultima modifica effettuata.

### 3.4.2 Personalizzazione

La ricerca può essere personalizzata tramite diversi passaggi, a partire dalle "Impostazioni del sito", che si trovano nella barra in alto a destra dopo aver cliccato "Contenuto del sito".

L'utente amministratore globale ha diverse alternative possibili per influenzare come viene effettuata la ricerca. In particolare, egli ha i seguenti strumenti:



**Figura 3.9:** Ricerca della parola *diagramma*

- *Regole di query della ricerca*, serve per aggiungere delle regole per promuovere alcuni risultati importanti, per mostrare risultati aggiuntivi e per definire alcune regole di classificazione (Figura 3.10).
- *Importazione configurazione di ricerca*, per utilizzare una ricerca esistente, ad esempio se si adottano degli standard aziendali e si vuole mantenere uno standard anche nel tipo di ricerche.



**Figura 3.10:** Template per l'aggiunta di una regola di query

Per quanto riguarda la visualizzazione dei risultati di una ricerca, ci sono altre possibilità, ovvero:

- *Origini dei risultati di ricerca*, per poter modificare il campo di azione della ricerca settato di default; è addirittura possibile integrare la ricerca con una ricerca esterna su Internet (Figura 3.11).
- *Tipi di risultati della ricerca*, per poter definire un modello di visualizzazione in HTML se si vuole utilizzare uno standard aziendale per ogni sito (vedi Figura 3.12).
- *Schema di ricerca*, per poter modificare l'indicizzazione sui risultati e sui tipi di informazioni delle proprietà per l'indicizzazione.
- *Impostazioni di ricerca*, per poter modificare sia il centro di ricerche, tramite un URL, sia la pagina dove i risultati devono essere visualizzati. Essa, di default, è la pagina padre, ma si potrebbe utilizzare una pagina personalizzata per la visualizzazione dei risultati (Figura 3.13).

Nome	Data di creazione	Predefinita	Stato
<b>Fornito dal servizio di ricerca (16)</b>			
Conversations	17/06/2020 12:27:55		Attive
Documents	17/06/2020 12:27:55		Attive
Items matching a content type	17/06/2020 12:27:55		Attive
Items matching a tag	17/06/2020 12:27:55		Attive
Items related to current user	17/06/2020 12:27:55		Attive
Items with same keyword as this item	17/06/2020 12:27:55		Attive
Local People Results	21/01/2020 09:09:24		Attive
Local Reports And Data Results	17/06/2020 12:27:55		Attive
Local SharePoint Results	17/06/2020 12:27:55	✓	Attive
Local Video Results	17/06/2020 12:27:55		Attive
Pages	17/06/2020 12:27:55		Attive
Pictures	17/06/2020 12:27:55		Attive
Popular	17/06/2020 12:27:55		Attive
Recently changed items	17/06/2020 12:27:55		Attive
Recommended Items	17/06/2020 12:27:55		Attive
Wiki	17/06/2020 12:27:55		Attive

**Figura 3.11:** Template per le origini dei risultati

Priorità	Nome	Data di modifica	Condizioni risultato	Azioni risultato
<b>Fornito dal servizio di ricerca (31)</b>				
1	Person	28/05/2021	Restituito dall'origine dei risultati Local People Results Corrisponde sempre per questa origine	Visualizza risultati con questo modello Elemento utenti
2	Microsoft Access	28/05/2021	Corrisponde a questi tipi di contenuto Microsoft Access	Visualizza risultati con questo modello Elemento documento di Office
3	Microsoft Excel	28/05/2021	Corrisponde a questi tipi di contenuto Microsoft Excel	Visualizza risultati con questo modello Elemento di Excel
4	Microsoft OneNote	28/05/2021	Corrisponde a questi tipi di contenuto Microsoft OneNote	Visualizza risultati con questo modello Elemento OneNote
5	Microsoft PowerPoint	28/05/2021	Corrisponde a questi tipi di contenuto Microsoft PowerPoint	Visualizza risultati con questo modello Elemento PowerPoint
6	Microsoft Publisher	28/05/2021	Corrisponde a questi tipi di contenuto	Visualizza risultati con questo modello

**Figura 3.12:** Template per i tipi di risultati delle ricerche

URL Centro ricerche:

Esempio: /CentroRicerche/Pagine o http://server/siti/CentroRicerche/Pagine

Usa le stesse impostazioni della pagina dei risultati della ricerca dell'elemento padre.

Invia query a un URL di pagina dei risultati personalizzato.

URL pagina risultati:

Esempio: /SearchCenter/Pages/results.aspx o http://server/sites/SearchCenter/Pages/results.aspx

Attiva il menu a discesa nella casella di ricerca e usa il primo nodo Navigazione ricerca come pagina dei risultati di destinazione.

**Figura 3.13:** Template per le impostazioni della ricerca

Avendo a disposizione una serie di configurazioni possibili per affinare la ricerca, si può pensare di sfruttare al meglio queste impostazioni, così da ottimizzare la ricerca nel miglior modo possibile, velocizzandola e migliorando i risultati attesi.

A valle di questa attenta analisi su SharePoint ci si è resi conto che la potenza di questo tool è tale da essere la migliore opzione per la gestione dell'intero sistema documentale come primo passo per la realizzazione del progetto *Uni 4 Justice*.

---

## Introduzione ad Akoma Ntoso

---

*In questo capitolo verrà descritto lo standard Akoma Ntoso, essendo uno dei punti di riferimenti del Progetto Uni 4 Justice. In particolare, verranno specificati i punti chiave su cui gli autori si sono basati per la realizzazione di questo standard, e verranno definiti gli obiettivi generici e strategici che essi si erano prefissati come punto di arrivo. Si passerà, poi, ad un accenno all'architettura del documento, per poi entrare nel dettaglio degli schemi di Akoma Ntoso concentrandosi, in particolar modo, sui tag previsti dallo standard stesso.*

*L'obiettivo di questo capitolo è quello di fornire una panoramica, a tratti dettagliata, di Akoma Ntoso, al fine di capire meglio la scelta dell'utilizzo di questo standard per alcuni documenti giudiziari e avere una visione d'insieme degli strumenti su cui il Progetto Uni 4 Justice fa affidamento.*

### 4.1 Akoma Ntoso

Akoma Ntoso<sup>1</sup> è, da Agosto 2018, lo standard OASIS utilizzato per rendere più agevole la creazione e la manipolazione dei documenti giudiziari. Esso prevede l'associazione di specifici "tag" ad ogni sezione del documento, al fine di renderlo "leggibile dalla macchina".

Il Progetto con cui è stato realizzato Akoma Ntoso è stato portato avanti da un'iniziativa di Africa i-Parliament Action Plan; infatti, il nome scelto per questo standard deriva dall'Africa occidentale, in particolare dalla lingua *akan*. Il termine scelto, nella lingua nativa, significa "cuori collegati" e viene espresso, da questo popolo, con il simbolo riportato in Figura 4.1, che altro non è che il logo di Akoma Ntoso.

Questo standard è, di fatto, una variante dello standard XML.



**Figura 4.1:** Akoma Ntoso

---

<sup>1</sup><http://www.akomantoso.org/>

Il concetto di "leggibile dalla macchina" è uno dei punti focali su cui verte *Akoma Ntoso*, in quanto esso permette, tramite software appositi, di catalogare i documenti e studiarne i contenuti. Ciò è reso possibile da software che vengono implementati sulla base della conoscenza della struttura con cui vengono "mappati" i documenti stessi. Questi software, lavorando con questo tipo di struttura, tramite i linguaggi di "mark-up", possono estrapolare le parole chiavi, nonché estrarre concetti semantici sapendo dove si trovano alcune frasi, quindi sapendo in quale parte del documento sono state utilizzate determinate parole. Basti pensare alla differenza di un testo se esso viene riportato come informazione descrittiva del caso che si vuole giudicare o se esso si trova all'interno di una sentenza, in quest'ultimo caso, esse non si limita semplicemente a descrivere il caso nella sua fattispecie, bensì, esprimono la decisione in merito ad esso; quindi, stesse parole possono avere significati totalmente diversi. Infatti, a "parità di frasi", si va a valutare, tramite il contesto nel quale essa si colloca, il "peso" e il significato effettivo che quella affermazione ha nella sentenza, nonché il concetto semantico che la frase assume nel contesto specifico.

Tutto questo sembra quasi richiamare il concetto di "Data Warehouse" rispetto al "Data Lake"; quest'ultimo risulta essere un insieme di dati eterogenei, "sporchi", disorganizzati e non strutturati, mentre, con il Data Warehouse si vanno a mappare i dati in informazioni. Infatti, il passaggio successivo all'utilizzo di questo standard, che trasforma i dati in informazioni, è l'estrapolazione della conoscenza in ambito giuridico.

Altri punti su cui *Akoma Ntoso* fa leva sono l'"accesso aperto" e le funzionalità avanzate, come, ad esempio, poter mantenere i riferimenti ad altri documenti, poter mantenere dati e metadati standardizzati, etc.

#### 4.1.1 Accesso libero

*Akoma Ntoso* ha permesso un accesso libero dei software ai documenti parlamentari, legislativi e giudiziari. La scelta di implementare questo standard per i documenti di queste tre macro-aree è derivata dall'osservazione che essi hanno diversi elementi in comune tra di loro, ovvero: attori, strutture, procedure, atti e processi di informazioni.

Questo standard ha permesso di rendere accessibili i dati non solo a livello nazionale, bensì anche a livello internazionale. La realizzazione di *Akoma Ntoso* si è basata su due concetti specifici, ovvero:

- *Interoperabilità semantica*, ossia il significato che i contenuti assumono sulla base della sezione in cui vengono inseriti.
- *Interoperabilità tecnica*, ossia la disponibilità dei documenti da qualsiasi parte del mondo. Infatti, la scelta di utilizzare un linguaggio come XML deriva dal rendere questi documenti accessibili e interpretabili ovunque.

È opportuno puntualizzare che i documenti che sono stati presi in esame, ovvero quelli parlamentari, legislativi e giudiziari, hanno quattro aspetti fondamentali:

- *contenuto*, cioè le parole e la punteggiatura delle frasi;
- *presentazione*, cioè le caratteristiche di formattazione delle frasi;
- *struttura*, cioè l'organizzazione delle frasi;
- *semantica*, cioè il significato delle informazioni espresse tramite le frasi.

L'intuizione umana nella lettura di un documento è da "trasmettere" alla macchina; quindi, per la realizzazione di *Akoma Ntoso*, la scelta è ricaduta sullo sviluppo di un metalinguaggio

basato sullo standard *XML*, essendo quest'ultimo un linguaggio di "mark-up" che si esprime sia con la struttura (differenziazione sulla base del ruolo nell'organizzazione del documento) che con la semantica (differenziazione sulla base del significato rispetto all'argomento del documento).

#### 4.1.2 Obiettivo

L'obiettivo raggiunto da *Akoma Ntoso* è stato quello di definire un insieme di regole per tutti i tipi di documenti di cui stiamo discutendo in questo capitolo. Infatti, con la realizzazione di questo standard, gli sviluppatori sono riusciti a soddisfare gli obiettivi che si erano inizialmente preposti. Nello specifico, tali obiettivi erano:

- *Definire un formato di documento comune.*
- *Definire un modello comune per lo scambio di documenti.* Una particolare attenzione è stata rivolta ai seguenti sotto-obiettivi:
  - Generazione di documenti indipendentemente dal tipo, dal paese, dalla lingua e dal processo tramite cui sono stati realizzati.
  - Presentazione univoca dei documenti da mostrare su schermo e stampare indipendentemente dal tipo, dal paese, dalla lingua e dal processo di generazione.
  - Accessibilità dei documenti indipendentemente dal tipo, dal paese, dalla lingua e dal processo di generazione, con il mantenimento di riferimenti tramite link, agevolando la navigazione tra i documenti.
  - Descrizione dei documenti indipendentemente dal tipo, dal paese, dalla lingua e dal processo di generazione, per permettere, ad esempio, la creazione di archivi, di strumenti di analisi e di confronto.
- *Definire uno schema di dati comune.*
- *Definire uno schema e un'ontologia dei metadati comuni*, che, tramite indicizzazione, facilita le ricerche e l'uso delle risorse online.
- *Definire uno schema comune per citazioni e riferimenti incrociati*, in quanto ogni documento ha un insieme numeroso di riferimenti ad altri documenti ufficiali. Il documento in esame può essere visto come nodo centrale, mentre i documenti linkati in esso sono visti come nodi collegati a quello centrale, ma devono essere anch'essi accessibili.

#### 4.1.3 Obiettivi strategici

La realizzazione del modello di *Akoma Ntoso* si è basata su alcuni obiettivi strategici che hanno permesso di delineare alcune caratteristiche fondamentali. Entrando nel dettaglio, i punti strategici sono i seguenti:

- Creare una "lingua franca" per lo scambio dei documenti tra le istituzioni, per accelerare il processo di elaborazione di un nuovo documento.
- Fornire un formato di archiviazione e accesso a lungo termine per questi tipi di documenti e permettere la ricerca, l'interpretazione e la visualizzazione degli stessi.
- Fornire un sistema attuabile per i sistemi parlamentari, legislativi e giudiziari nelle istituzioni, al fine di agevolare l'utilizzo dell'informatica nell'analisi dei documenti.

- Creare modelli comuni di dati e metadati per permettere l'accesso ai parlamenti e alle diverse corti, in particolare a tutti i documenti da essi prodotti.
- Rendere i documenti "autoesplicativi" ed estendibili.

Ciò che *Akoma Ntoso* ha ottenuto è descritto secondo diversi aspetti. Questi ultimi sono riportati di seguito:

- *Modello dei dati semplice*. Tale caratteristica si è basata sull'idea di semplificare, in parte, l'apprendimento, da parte dei sistemi informatici, di documenti complessi. Questi ultimi, infatti, sono basati su un linguaggio ad alto livello di dettaglio, il che ha portato ad una ricerca di "contenitori", "blocchi", "gerarchie", etc. che permettessero di associare un concetto semantico ad una specifica parte del documento.
- *Capacità di evolvere*. Tale caratteristica prende in esame una specifica problematica, ovvero l'evoluzione del linguaggio. Per garantire tale proprietà si è reso *Akoma Ntoso* capace di adattarsi ad evoluzioni e cambiamenti, così da rendere compatibili i nuovi schemi con le validazioni del sistema stesso.
- *Garanzia di correttezza*. Tale caratteristica permette la convalida dei documenti in *Akoma Ntoso*, verificando il numero e la posizione corretta di tutti gli elementi previsti dallo standard.
- *Strumenti*. Tale caratteristica considera la necessità di strumenti annessi alla realizzazione dello standard. Di seguito vengono riportati alcuni di essi:
  - *L'editor*: strumento utile, anche se non obbligatorio, per formattare correttamente il documento stesso.
  - *Il "converter"*: strumento utile per convertire un documento esistente in uno compatibile con lo standard di *Akoma Ntoso*.
  - *Il "name resolver"*: strumento utile per individuare un determinato contenuto, sulla base degli indicatori utilizzati nel documento.
  - *I "post-editing"*: strumenti utili per convalidare, arricchire e archiviare i documenti redatti.

## 4.2 Architettura del documento

Affinchè i documenti siano conformi al modello accettato da *Akoma Ntoso*, essi devono rispecchiare determinati requisiti. Nello specifico, i requisiti di conformità sono i seguenti:

- *Schema generale di Akoma Ntoso*. Tale requisito è obbligatorio e prevede che venga rispettato lo schema generale nonché le linee guida per l'utilizzo.
- *Schema personalizzato di Akoma Ntoso*. Tale requisito è facoltativo e prevede che il singolo organo emanatore possa definire alcune regole specifiche, purchè venga rispettato lo schema generale.
- *Set comune dei metadati di Akoma Ntoso*. Tale requisito è obbligatorio e prevede che venga convalidato l'elenco dei metadati degli elementi che devono essere utilizzati nel documento.
- *Set personalizzato dei metadati di Akoma Ntoso*. Tale requisito è obbligatorio e prevede che venga convalidato l'elenco dei metadati degli elementi utilizzati nel documento.

- *Estensioni locali per i metadati di Akoma Ntoso.* Tale requisito è parzialmente facoltativo e prevede l'eccezione di metadati non facenti parte di quelli comuni e personalizzati; tali metadati, quindi, vanno considerati come conformi, anche se sconosciuti e non utilizzati.

#### 4.2.1 Strutture di markup

Essendo *Akoma Ntoso* uno standard che si riferisce ad un contesto internazionale, ha la necessità di essere generale ma anche specifico e, quindi, personalizzabile.

È necessario distinguere i diversi linguaggi di marcatura previsti; in particolare, le due tipologie di "markup" consentite sono:

- *nome*, ossia l'etichetta associabile ad una parte del testo;
- *modello o tipo di contenuto*, ossia i vincoli strutturali associati al nome.

### 4.3 Schema

Passiamo, ora, alla descrizione della struttura dello schema definito per *Akoma Ntoso*, illustrando, nel dettaglio, lo schema generale e i "markup" conformi allo standard stesso.

#### 4.3.1 Schema generale

Lo schema generale definisce l'insieme minimo di vincoli che devono essere soddisfatti dai documenti per essere convalidati. Tale schema si basa su sei categorie di elementi, le quali sono riportate con il proprio esempio di riferimento:

- *Il marcatore*: questo elemento è privo di contenuto e assume significato se combinato in base a tre specifici fattori: posizione, nome e attributi. Questi elementi possono essere *placeholder* (accettati in qualsiasi posizione) o metadati (accettati solo in alcune sottosezioni dei metadati stessi).

```
<noteRef n="1" href="#note1">
```

- *L'inline*: questo elemento è un contenitore di testo posizionato in linea con una specifica parte del testo di un singolo blocco; esso è utile per definire la semantica o la struttura per la parte a cui si riferisce. Esempi di esso sono rappresentati da `term` e `ref` nell'esempio successivo.
- *Il blocco*: questo elemento è un contenitore autonomo di testo; esso permette, ad esempio, di definire un paragrafo, un'intestazione, etc. .

```
<p><def>Commission</def> means the commission established by
<ref href="#sect20">section 20</ref></p>
```

- *Il contenitore*: questo elemento è un contenitore per il raggruppamento di elementi di determinati tipi assegnati ad un nome cumulativo. Essi sono diversi, in quanto il nome del contenitore varia in base al suo contenuto.

```
<act>
<meta> ... </meta>
<preamble> ... </preamble>
<body> ... </body>
<attachments> ... </attachments>
</act>
```

- *Il contenitore gerarchico*: questo elemento è un contenitore specifico che definisce una gerarchia di contenitori; esso non può contenere testo, ma solo blocchi che contengono testo. Esso definisce, quindi, un insieme di sezioni annidate che si differenziano in base alla profondità, al titolo e alla numerazione.

```

<body>
<chapter id="chal">
<num>Chapter 1</num>
<heading>Traditional communities and ...</heading>
<paragraph id="chal-par2">
<num>2</num >
<heading>Recognition of traditional ...</heading>
<clause id="chal-par2-cla1">
<num>1</num>
<content>
<p>A community may be recognised as ...</p >
</content>
</clause>
...
</paragraph>
...
</chapter>
...
</body>

```

- *Il popup*: questo elemento indipendente è inserito in un flusso di testo *inline*. Un esempio di esso è rappresentato da *quotedStructure*:

```

<body>
<section id="secl">
<heading>Amendment of section 4 of Decree 43 of 1990 </heading>
<clause id="secl-cla1">
<num>1.</num>
<content>
<p> Section 4 of the Supreme Court Decree, 1990 (Ciskei), is hereby
amended by the substitution for subsection (2) of the following
subsection:
<mod id="secl-cla1-mod1">"
<quotedStructure id ="secl-cla1-mod1-qst1">
<subsection id="secl-cla1-mod1-qst1-sec4-ssc2">
<num>(2)</num>
<content>
<p> Notwithstanding the provision of section [1]...</p>
</content>
</subsection>
</quotedStructure>
"</mod>
</p>
</content>
</clause>
</section>
</body>

```

### Componenti dello schema generale

Gli elementi accettati dallo standard di *Akoma Ntoso* sono molteplici e specifici. Entrando nel dettaglio, quelli definiti da questo standard sono:

- *Elemento radice* <akomantoso>: tag di inizio di un documento di Akoma Ntoso.
- *Elementi del documento*, uno per ogni tipo di documento, ovvero:
  - <act>;
  - <bill>;
  - <judgement>;
  - <debateReport>;
  - <debate>;
  - <amendmentList>;
  - <officialGazette>;
  - <documentCollection>;
  - <amendment>;
  - <doc>.
- *Elementi contenitori*, per ogni parte principale delle strutture (*hierarchicalStructure*, *openStructure*, *debateStructure* e *judgementStructure*).
- *Elementi gerarchici*: <title>, <num>, etc. .
- *Elementi per la suddivisione di un dibattito parlamentare*:
  - <administrationOfOath>;
  - <declarationOfVote>;
  - <communication>;
  - <petitions>;
  - <papers>;
  - <noticesOfMotion>;
  - <questions>;
  - <address>;
  - <proceduralMotions>;
  - <pointOfOrder>;
  - <adjournment>;
  - <rollCall>;
  - <prayers>;
  - <oralStatements>;
  - <writtenStatements>;
  - <personalStatements>;
  - <ministerialStatements>;
  - <resolutions>;

- <nationalInterest>.
- *Elementi per riportare un discorso di un dibattito:*
  - <speech>;
  - <question>;
  - <scene>;
  - <narrative>;
  - <summary>.
- *Elementi per esprimere giudizi e strutture aperte:*
  - <blockList>;
  - <item>.
- *Elementi inline:*
  - <toc>, per un sommario;
  - <ref>, per un riferimento normativo;
  - <def>, per una definizione;
  - <noteref>, per una nota;
  - <recordedTime>, per il tempo di una registrazione di un intervento orale;
  - <mod>, per gli emendamenti;
  - <remark>, per commentare il dibattito.
- *Elementi generici*, utilizzati per introdurre un nuovo concetto semantico non previsto nei tag predefiniti.
- *Elementi HTML*. Hanno pressochè tutti lo stesso significato HTML, ad eccezione di alcuni. In particolare, i tag `div`, `p` e `span` vanno utilizzati, rispettivamente, come `container`, `block` e `inline`.
- *Elementi per i metadati*, suddivisi in:
  - Identificazione.
  - Pubblicazione.
  - Classificazione.
  - Ciclo di vita.
  - Analisi.
  - TemporalInfo.
  - Riferimenti.
  - Note.
  - Proprietari.

### 4.3.2 Schema personalizzato

Lo schema personalizzato definisce un insieme aperto di vincoli più rigidi, che servono per garantire la validità di un documento specifico per un determinato parlamento. Tale schema permette di aggiungere regole speciali, specifiche in base ai singoli paesi, pur rimanendo validi i vincoli definiti dallo schema generale, al fine di mantenere il documento conforme con lo standard *Akoma Ntoso*.

---

## Definizione dell'approccio proposto

---

*Questo capitolo si concentra sul "mondo" del Natural Language Processing, mostrando i passaggi teorici che hanno permesso di implementare l'algoritmo per l'analisi di testi. Inizialmente vi è un accenno alle fasi che caratterizzano l'NLP, per poi passare ad un'astrazione teorica ed al dettaglio dei passaggi che hanno portato a creare l'ontologia e il parser, concludendo con i passaggi che hanno permesso di estrarre la conoscenza del dataset originario.*

*L'obiettivo di questo capitolo è quello di delineare teoricamente gli "step" necessari per estrapolare i concetti fondamentali nel campo di applicazione dell'algoritmo, che verrà poi concretizzato, descrivendolo nello specifico, nel capitolo successivo.*

### 5.1 Il Natural Language Processing

Il *Natural Language Processing* è uno degli strumenti di analisi dei dati che cerca di estrarre informazioni in modo automatico, studiando il linguaggio naturale. Il contesto in cui ci approcceremo al linguaggio naturale, nello specifico, è la sua espressione in forma scritta, quindi, analizzando dei testi.

#### 5.1.1 Aspetti fondamentali

L'NLP è un processo di analisi che si concretizza tramite lo sviluppo di alcune fasi ben precise. Esse sono le seguenti:

- *L'elaborazione morfologica*, la quale consiste nell'estrazione delle singole parole di un testo, creando, per ognuna, un *token*.
- *L'analisi sintattica*, la quale consiste nella verifica della correttezza grammaticale delle frasi.
- *L'analisi semantica*, la quale consiste nella verifica della correttezza del significato delle frasi.
- *L'analisi pragmatica*, la quale consiste nella scelta del concetto che una frase assume nel contesto in cui si trova.

Il *Part Of Speech Tagging* è una delle tecniche che l'NLP utilizza per lo studio dei testi; in particolare, essa permette di assegnare ad un token un significato semantico, avvalendosi dell'analisi grammaticale della frase in cui la parola è utilizzata.

## 5.2 L'ontologia

L'ontologia, in ambito informatico, è una struttura dati contenente un insieme di parole derivate dal "mondo di interesse", estrapolando informazioni, al fine di concettualizzare il dominio in esame.

### 5.2.1 Costruzione teorica della struttura di base

Prima di passare ad una fase puramente automatica, è opportuno effettuare una prima fase manuale, nonché un'attenta analisi delle parole più utilizzate. Questo approccio permette di identificare quelle parole che assumono un'accezione fondamentale all'interno del dominio stesso, in quanto il concetto delle frasi dipende esso stesso dalle singole parole che vengono utilizzate. Quindi, partendo dai concetti fondamentali del mondo di interesse, è possibile poi estrapolare le informazioni che lo caratterizzano, associando il giusto significato alle parole in base al contesto. Per portare all'attenzione l'importanza del contesto, e del concetto semantico, basti pensare alla parola "Ferrari"; essa può indicare sia la famosa casa automobilistica che l'azienda vitivinicola.

La creazione dell'ontologia, quindi, parte dalla ricerca delle parole più frequenti e dalle relazioni che intercorrono tra di esse, facendo attenzione alla sinonimia, quindi prendendo in considerazione i sinonimi che vengono utilizzati.

A partire dal dataset, la prima cosa effettuata è stata la ricerca manuale delle parole più frequenti, che avessero un significato nel dominio dei dati. Dopodiché, è stato necessario visionare uno ad uno un paio di centinaia di testi forniti dal dataset. La scelta di annotare le parole con la rispettiva frequenza, quest'ultima aumentata di volta in volta che la parola compariva nel testo, è ricaduta su un foglio *excel*; questo per velocizzare la raccolta dei dati.

Per ogni parola è stata conteggiata sia la sua frequenza nel totale dei testi, ovvero il numero totale di volte che quella parola è stata trovata, sia il numero di testi in cui viene utilizzata, non considerando il numero di volte che essa viene ripetuta nel singolo testo.

Una volta effettuata questa prima fase, abbiamo deciso di considerare solo le parole che avessero una frequenza maggiore di 1, sia per quanto riguarda la frequenza totale che per ciò che concerne la frequenza nel numero di testi.

Una volta ottenute le parole effettivamente più ricorrenti, si è passati ad un'analisi, ancora una volta manuale, dei termini individuati, considerando il contesto del dataset.

Per prima cosa sono stati trovati i vari sinonimi utilizzati per una stessa parola, all'interno dei diversi testi, considerando, quindi, quei termini come "equivalenti".

Successivamente, con i termini rimasti a valle dei passaggi precedenti, abbiamo costruito una gerarchia di parole, in base al collegamento semantico dei termini trovati, ottenendo, così, un albero gerarchico con diversi livelli. L'albero trovato è caratterizzato dai singoli nodi contenenti le parole scelte, in base alla frequenza ed al contesto semantico, i cui "figli" sono parole semanticamente collegate ai "padri" e pur sempre frequenti, perchè scelte dall'elenco ottenuto precedentemente dopo il filtraggio della frequenza.

La decisione finale, per quanto riguarda l'ontologia e la struttura dati utilizzata, è stata quella di inserire tutte le parole dell'ontologia in un dizionario e il collegamento del termine "chiave", con i relativi "sinonimi", in un altro.

Ottenuta l'ontologia, si è passati alla creazione del parser per la gestione automatica dell'analisi dei testi, facendo utilizzo dei risultati trovati con questo approccio manuale.

## 5.3 Il parser

Una volta effettuata la costruzione manuale dell'ontologia, essa viene utilizzata come base di partenza per la creazione del parser automatico che analizzerà i testi. In particolare, esso permette di trovare le parole associate a quelle dell'ontologia, analizzando le parole successive ad esse. Una volta individuate le chiavi (parole dell'ontologia) e i suoi valori (parole associate a quelle dell'ontologia), sarà necessario prendere in esame la frequenza con cui essi sono associati a determinate chiavi. Questo è un passaggio utile per capire quali sono quelle parole che vanno a descrivere i concetti chiave del dominio stesso, caratterizzandolo.

### 5.3.1 Applicazione del "Part of Speech"

Per la creazione del parser, si può fare utilizzo del PoS tagging, descritto precedentemente, il quale permette di assegnare ad un token un concetto semantico. Esso viene, quindi, utilizzato per filtrare le parole da analizzare in base al fatto che abbiano un certo significato; verranno pertanto selezionate quelle parole che permettano di estrapolare informazioni. Ciò consente di restringere il campo di ricerca escludendo parole di non interesse, come ad esempio gli articoli e le congiunzioni, riducendo drasticamente i tempi di esecuzione del codice.

Per quanto concerne la realizzazione del parser, sono stati necessari diversi passaggi. Più specificatamente:

- Abbiamo ricercato tutte le parole dell'ontologia presenti nei testi del dataset, grazie all'utilizzo del *pattern matching* della libreria *Spacy*.
- Per ogni parola trovata, abbiamo selezionato un numero costante  $k$  di parole successive.
- Abbiamo tenuto in considerazione, per le parole successive, solo quelle che fossero sostantivi, aggettivi o numeri; a tal fine, abbiamo utilizzato il *pattern matching* della libreria *Spacy*.
- Abbiamo inserito nel dizionario, in corrispondenza della chiave (parola dell'ontologia), le parole successive trovate, rimaste dal "filtraggio" precedente.
- Abbiamo effettuato, per ogni parola dell'ontologia, il conteggio della frequenza delle parole associate trovate per ogni chiave.

Tramite il parser, abbiamo ottenuto, quindi, per ogni parola dell'ontologia trovata manualmente, un insieme di parole che vengono utilizzate per descriverla, con la rispettiva frequenza con cui la parola stessa viene utilizzata per caratterizzare quella specifica chiave. Così facendo, si è potuto creare un nuovo dataset su cui lavorare, per poter estrapolare informazioni e, quindi, conoscenza del dataset originale.

## 5.4 Gestione dei risultati

La gestione dei risultati consiste nell'interrogazione del dataset estrapolato dall'analisi dei testi di partenza. Tramite la definizione di domande ben strutturate, è possibile, quindi, ottenere informazioni, trasformando, così, queste ultime in conoscenza del dominio applicativo.

Per effettuare l'interrogazione del "nuovo dataset", la scelta è ricaduta sull'utilizzo di *Neo4j*.

*Neo4j* è un DBMS noSQL per la gestione di grandi quantità di dati sotto forma di grafi. Per poterlo utilizzare è stato necessario effettuare una fase di installazione e configurazione del driver di *Neo4j*.

Una volta organizzato l'ambiente di sviluppo, per poter inserire e gestire i dati ottenuti con l'ontologia realizzata e il parser automatico, ci siamo concentrati sulla stesura di alcune query, sulla base della struttura dei dati ottenuti. Per fare ciò, è stato necessario creare le *query* nel linguaggio utilizzato da *Neo4j* per interrogare i grafi, ovvero *Cypher*, il quale assomiglia al linguaggio *SQL*.

Infine, trovate le query da effettuare, abbiamo integrato tutto nel codice scritto in *Python*, tramite il supporto del driver precedentemente "settato", per rendere il tutto automatico e fluido.

In tutto lo sviluppo dei vari passaggi ci siamo sforzati di scrivere il codice in maniera più generica possibile; purtroppo, in alcune occasioni, è stato necessario entrare nello specifico del dataset, utilizzando, ad esempio, una certa colonna della base di dati di partenza.

---

## Applicazione dell'approccio proposto: implementazione

---

*Questo capitolo tratta dell'implementazione dell'approccio discusso teoricamente nel capitolo precedente. Per prima cosa verrà spiegato il tipo di dataset ricercato e utilizzato, per poi mostrare i passaggi che hanno portato all'individuazione delle parole dell'ontologia. Vi è poi un cambio, cioè si passa da un'implementazione puramente manuale ad una automatica. In particolare, verranno illustrate alcune parti dell'algoritmo, con la relativa spiegazione, che hanno portato all'estrazione di pattern. Il capitolo si conclude con i codici che hanno permesso la gestione delle parole individuate, a partire dalla creazione del grafo di parole in Neo4j per poi finire con le interrogazioni al grafo stesso.*

*L'obiettivo di questo capitolo è quello di dettagliare l'implementazione a livello di codice, per concretizzare ciò di cui si è discusso finora, ovvero mostrare i passaggi per l'applicazione dell'NLP ad un insieme di testi per l'estrapolazione di informazioni.*

### 6.1 Creazione manuale dell'ontologia

La prima cosa che abbiamo dovuto implementare è stata l'ontologia, che, come anticipato nel capitolo precedente, è stata costruita manualmente.

A causa della mancanza delle note delle sentenze, abbiamo dovuto ricercare un dataset che assomigliasse il più possibile alla "struttura" delle note. In particolare, la nota si presenta come un testo di 300/500 parole, caratterizzato da una sorta di "pattern" insito nel testo stesso. La struttura che ricorre nelle diverse note è la presenza di termini che si ripetono, come, ad esempio, "imputato", "divieto", "obbligo", "sanzione", "articolo", "sentenza", etc. Per questo motivo abbiamo dovuto ricercare un insieme di testi in cui determinate parole ricorressero "naturalmente", quindi "semi-strutturato" e non totalmente strutturato, anche dissociandosi radicalmente dall'argomento.

Una volta capito che l'importante era trovare un dataset che fosse semi-strutturato, prescindendo dall'argomento, abbiamo individuato il tipo di dati da ricercare, e la scelta è ricaduta su delle recensioni. Infatti, le recensioni, concentrandosi sulla descrizione di una determinata cosa, tendono a rispecchiare tutte lo stesso "format"; quindi la ricorrenza di parole specifiche è una loro caratteristica. La scelta di prescindere dal campo giuridico è stata una costrizione dovuta all'assenza di un insieme di testi giuridici di quella tipologia, cioè che avessero un massimo di parole e che fossero "semi-strutturati". Una volta creato l'iter e l'algoritmo da seguire, basterà adattarlo e specificarlo al dataset delle note delle sentenze, per poter avere velocemente il risultato desiderato.

Il primo passo è stato quello di visionare manualmente più di 200 recensioni, sia positive che negative, così da poter considerare sia quei termini con accezioni prettamente positive che quelli con accezioni negative.

### 6.1.1 Individuazione delle parole dell'ontologia

Abbiamo utilizzato *excel* per prendere nota di tutti i termini, positivi e negativi, che più caratterizzavano i singoli testi. Abbiamo, quindi, proceduto con la ricerca manuale delle parole, inserendole nuove se non presenti nella lista creata, oppure, per ogni parola già presente in lista, aumentando di uno il conteggio della sua frequenza totale e il numero di testi contenente quella parola, se quel termine non era ancora stato incontrato in quello specifico testo.

Una volta ottenuta la lista dei termini più frequenti, abbiamo dovuto restringere il numero totale di parole trovate, filtrando solo quelle con le frequenze maggiori di 1.

Dopodiché vi è stata una fase più riflessiva, in quanto abbiamo dovuto creare la gerarchia insita nelle parole trovate, in base al contesto applicativo, sviluppata sotto forma di albero gerarchico. Il collegamento tra le parole è stato creato sulla base del loro significato, ossia concentrandosi sulla loro semantica.

Per poter utilizzare l'ontologia all'interno del codice, abbiamo deciso di appoggiarci alla struttura dati del dizionario; in particolare, abbiamo creato due dizionari, uno contenente tutti i termini dell'ontologia trovata, mentre l'altro contenente il mapping tra il singolo termine e i suoi relativi sinonimi.

## 6.2 Algoritmo relativo al parser automatico

Per la creazione del parser automatico, ci siamo affidati alla tecnica del "pattern matching" della libreria *Spacy*.

### 6.2.1 Utilizzo della libreria "Spacy"

*Spacy* è una libreria di *Python* che permette di utilizzare varie tecniche per l'implementazione del *Natural Language Processing*; una di queste è la ricerca di "pattern", ovvero la ricerca di un certo termine, o sequenza di termini, all'interno di un testo. In particolare, viene definito ed utilizzato il *Matcher*, ovvero un "controllore" di testi, che ricerca le regole definite dal programmatore.

Nel nostro caso specifico, abbiamo utilizzato il *Matcher* per poter individuare tutte le parole dell'ontologia utilizzate nei diversi testi, e, per ogni parola trovata, abbiamo considerato un numero costante di parole successive e cercato quali tra queste fossero sostantivi, numeri o aggettivi, applicando nuovamente il *Matcher* per portare avanti questa ulteriore ricerca.

### 6.2.2 Implementazione per l'estrazione di pattern

Riportiamo, ora, il codice (Listato 6.1) sviluppato per implementare la ricerca ed estrazione dei pattern all'interno delle recensioni positive. Questo codice è analogo a quello riguardante le recensioni negative; per semplicità di esposizione, ci concentreremo solo su quelle positive.

```
1 #creazione di due dizionari contenenti le parole dell'ontologia come chiavi
2 label_dictP = dict.fromkeys(labels)
3 label_dict2P = dict.fromkeys(labels)
4
5 #inizializzazione dei valori dei due dizionari come array vuoti
6 for key,value in label_dictP.items():
7     label_dictP[key]=[]
8     label_dict2P[key]=[]
9
10 pattern = [{"LEMMA":{"IN":lemma_labels_}}] #definizione del pattern matching con
        le parole dell'ontologia
```

```

11 pattern1 = [{"LEMMA":{"NOT_IN":lemma_labels_}, "POS":"NOUN"}] #definizione del
    patter matching con i sostantivi
12 pattern2 = [{"LEMMA":{"NOT_IN":lemma_labels_}, "POS":"ADJ"}] #definizione del
    patter matching con gli aggettivi
13 pattern3 = [{"LEMMA":{"NOT_IN":lemma_labels_}, "LIKE_NUM":True}] #definizione del
    patter matching con i numeri
14
15 #per tutte le rensioni desiderate (h) applica la ricerca delle parole con il
    pattern matching popolando i valori del dizionario "label_dictP"
16 for y1 in range(h):
17
18     doc = nlp(df["Positive_Review"][y1]) #inizializzazione di doc associandola
    alla singola recensione positiva
19
20     matcher = Matcher(nlp.vocab) #inizializzazione del Matcher
21     matcher.add("PATTERN", [pattern]) #aggiunta del pattern di base al Matcher,
    per ricercare le parole dell'ontologia
22
23     #per tutte le parole dell'ontologia trovate
24     for match_id, start, end in matcher(doc):
25
26         key_matched = str(doc[start]) #estrazione della parola dell'ontologia che
    e' stata trovata nel testo
27         subdoc = nlp(str(doc[start+1:start+(k+1)])) #estrazione delle k parole
    successive alla parola dell'ontologia trovata
28
29         flag = [] #creazione dei vettori di check per la gestione dei negativi
30         digit = [] #creazione dei vettori di check per la gestione dei numeri
31
32         #inizializzazione dei vettori di check
33         for k_f in range(k+1):
34             flag.append(False)
35             digit.append(False)
36
37         for token in subdoc: #per ogni termine dei k successivi
38             #se c'e' un termine di negazione
39             if (token.text == "not") | (token.text == "neither") | (token.text ==
    "no") | (token.text == "t"):
40                 #imposta a True il flag nella sua i-esima posizione
41                 flag[token.i] = True
42             #per ogni lettera del termine
43             for t in token.text:
44                 #se e' un numero
45                 if t.isdigit():
46                     #imposta a True il digit nella sua i-esima posizione
47                     digit[token.i] = True
48
49         matcher2 = Matcher(nlp.vocab) #inizializzazione di un nuovo Matcher
50         matcher2.add("PATTERN1", [pattern1]) #aggiunta del pattern1 al Matcher2,
    per ricercare i sostantivi
51         matcher2.add("PATTERN2", [pattern2]) #aggiunta del pattern2 al Matcher2,
    per ricercare gli aggettivi
52         matcher2.add("PATTERN3", [pattern3]) #aggiunta del pattern3 al Matcher2,
    per ricercare i numeri
53
54         #per tutte le parole trovate dal Matcher2
55         for match_id2, start2, end2 in matcher2(subdoc):
56
57             value_matched = subdoc[start2:end2] #estrazione della parola "
    matchata" dal Matcher2 all'interno dei k successivi
58
59             #per ogni elemento del dizionario

```

```

60         for key, values in label_dictP.items():
61             #se la chiave del dizionario combacia con quella trovata dal
Matcher
62             #e se il valore trovato non e' "wasn"
63             if (key == key_matched) & (value_matched.text != 'wasn'):
64                 #se il flag del termine precedente e' True, quindi ha davanti
un termine di negazione
65                 if (flag[start2-1]):
66                     #inserisce nel dizionario in corrispondenza della parola
dell'ontologia (key) il valore trovato dal Matcher2, ma come valore negato
con "not_" davanti
67                     label_dictP[key].append("not_"+value_matched.lemma_)
68                     #se il digit di quel termine e' True, allora e' un numero
69                     elif (digit[start2]):
70                         #inserisce nel dizionario in corrispondenza della parola
dell'ontologia (key) il valore trovato dal Matcher2, ma come "_" davanti
71                         label_dictP[key].append("_"+value_matched.lemma_)
72                         #altrimenti
73                     else:
74                         #inserisce nel dizionario in corrispondenza della parola
dell'ontologia (key) il valore trovato dal Matcher2
75                         label_dictP[key].append(value_matched.lemma_)
76
77 #per tutte le chiavi e valori del dizionario popolato
78 for key, values in label_dictP.items():
79     #in corrispondenza della chiave, aggiunge nel nuovo dizionario tutti i suoi
valori associati con la rispettiva frequenza
80     label_dict2P[key].append(Counter(values))

```

**Listato 6.1:** Codice relativo all'implementazione del *parser*

L'algoritmo si sviluppa nel seguente modo:

1. Creazione e inizializzazione di due dizionari contenenti come chiavi le parole dell'ontologia e come valore un array vuoto. Questi array conterranno, poi, i valori che verranno trovati per gli specifici termini dell'ontologia. In particolare, il primo dizionario (*label\_dictP*) conterrà come valori tutte le parole associate alle chiavi; in questo caso, quindi, le parole potranno essere duplicate; il secondo (*label\_dict2P*) conterrà, comunque, le parole trovate associate alle chiavi, ma senza ripetizioni; ad ogni parola trovata verrà associata la relativa frequenza con cui essa è stata usata per descrivere quella determinata chiave.
2. Definizione dei quattro tipi di pattern che verranno utilizzati per ricercare determinate parole. Si partirà dal pattern per la ricerca delle parole dell'ontologia, per poi passare a quelli che permettono di estrapolare i sostantivi, gli aggettivi e i numeri che vengono utilizzati per descrivere i termini dell'ontologia.
3. A questo punto, per ogni recensione positiva, viene iterato il pattern matching della parola dell'ontologia; quest'ultimo è stato implementato come di seguito specificato:
  - Viene aggiunto il pattern che ricerca i termini dell'ontologia e, per ogni "match":
    - Viene estratta la parola corrispondente e vengono estratte le *k* parole successive usate per descriverla.
    - Vengono creati due vettori e vengono inizializzati a *False*. Questi serviranno per tenere traccia dei termini di negazione (*flag*) e dei numeri (*digit*) che verranno utilizzati per descrivere i termini dell'ontologia.
    - Per ogni termine, dei *k* termini associati alla parola dell'ontologia, vengono effettuati i seguenti controlli:

- \* Se è un termine di negazione, allora viene "settato" a *True* il *flag* in corrispondenza della sua posizione.
  - \* Se il termine contiene un numero, allora viene "settato" a *True* il *digit* in corrispondenza della sua posizione.
  - Creazione di un nuovo *Matcher*, denominato *Matcher2*.
  - Aggiunta dei pattern matching al *Matcher2* per l'estrazione delle parole caratterizzanti le parole dell'ontologia.
  - Ogni parola di cui il *Matcher2* ha effettuato il matching viene aggiunta come valore associato al rispettivo termine dell'ontologia, nel seguente modo:
    - \* Per ogni coppia chiave-valore del dizionario *label\_dictP*, se la chiave combacia con la parola dell'ontologia "matchata" dal primo *Matcher*, allora vengono effettuati i seguenti ulteriori controlli:
      - Se il *flag* nella posizione precedente a quella trovata dal *Matcher2* è un termine di negazione, allora quel termine viene inserito con "\_not" davanti come valore nel dizionario in corrispondenza della chiave trovata dal *Matcher*.
      - Se il *digit* della parola trovata dal *Matcher2* contiene un numero, allora quel termine viene inserito con "\_" davanti come valore nel dizionario in corrispondenza della chiave trovata dal *Matcher*.
      - Altrimenti il termine viene inserito direttamente come valore nel dizionario in corrispondenza della chiave trovata dal *Matcher*.
4. Infine, per ogni coppia chiave-valore ottenuta dagli "step" precedenti:
- Nell'altro dizionario, in corrispondenza della corrispettiva chiave, vengono inseriti tutti i valori trovati ma con la rispettiva frequenza.

### 6.3 Gestione grafo delle parole individuate

Una volta creati l'ontologia e il parser, abbiamo utilizzato la libreria *NetworkX* per creare un grafo contenente le parole dell'ontologia e i termini che le descrivono (vedi Listato 6.2).

```

1 #creazione degl grafo vuoto B
2 B = nx.Graph()
3 #per ogni chiave-valore del dizionario delle recensioni positive
4 for k,v in label_dict2P.items():
5     #per ogni termine-frequenza del valore
6     for i,f in v[0].items():
7         #aggiungi un arco tra la chiave e il termine, con f come attributo di
         frequenza positiva
8         B.add_edge(k,i,freq_positive=f,freq_negative=0)
9 #per ogni chiave-valore del dizionario delle recensioni negative
10 for k,v in label_dict2N.items():
11     #per ogni termine-frequenza del valore
12     for i,f in v[0].items():
13         #se esiste gia' un arco tra chiave e termine
14         if B.has_edge(k,i):
15             #aggiorna ad f l'attributo di frequenza negativa
16             nx.set_edge_attributes(B, {(k,i): {"freq_negative":f}})
17         else:
18             #aggiungi un arco tra la chiave e il termine, con f come attributo di
             frequenza negativa
19             B.add_edge(k,i,freq_positive=0,freq_negative=f)

```

**Listato 6.2:** Grafo in *NetworkX*

Il codice implementa i seguenti passi:

1. Creazione di un grafo vuoto.
2. Per ogni coppia chiave-valore del dizionario delle recensioni positive, vengono presi i singoli valori, estraendo i termini ( $i$ ), che descrivono le parole dell'ontologia, con la relativa frequenza ( $f$ ). Inoltre, viene aggiunto un arco tra la chiave e il valore  $i$ -esimo, con  $f$  come attributo di frequenza positiva ( $freq\_positive$ ) e con  $0$  come valore per l'attributo di quella negativa.
3. Per ogni coppia chiave-valore del dizionario delle recensioni negative, vengono presi i singoli valori, estraendo i termini ( $i$ ), che descrivono le parole dell'ontologia, con la relativa frequenza ( $f$ ). Inoltre:
  - se esiste già un arco tra la chiave e il valore  $i$ -esimo, allora viene solo modificato l'attributo della  $freq\_negative$  ad  $f$ .
  - altrimenti, viene aggiunto un arco tra la chiave e il valore  $i$ -esimo, con  $f$  come attributo di frequenza negativa ( $freq\_negative$ ) e con  $0$  come valore per l'attributo di quella positiva.

### 6.3.1 Utilizzo del software "Neo4j"

Una volta ottenuto il grafo, ci siamo resi conto che la gestione effettiva del dataset sarebbe stata molto più efficiente se effettuato tramite *Neo4j*. Per questo motivo, abbiamo deciso di utilizzare tale *DBMS noSQL* per poter manipolare il grafo ottenuto.

### 6.3.2 Setting del driver

Come prima cosa abbiamo installato *Neo4j Desktop*. Successivamente, per poter far comunicare il codice *Python* con *Neo4j* abbiamo dovuto configurare il driver tramite le righe di codice riportate nel Listato 6.3.

```

1 import neo4j
2 from neo4j import GraphDatabase
3
4 # Credenziali del database
5 uri = "bolt://localhost:7687"
6 userName = "neo4j"
7 password = "password"
8
9 # Connessione al server del database in Neo4j
10 graphDB_Driver = GraphDatabase.driver(uri, auth=(userName, password))

```

**Listato 6.3:** Configurazione *driver*

Per quanto riguarda l'*URI* bisogna inserire come porta quella che si trova all'interno del database creato come progetto in *Neo4j Desktop*. Il numero della porta lo si può controllare in corrispondenza della scritta "Bolt port" nella scheda laterale che si espande cliccando sul progetto in *Neo4j*. Per poter aprire l'interfaccia che permette di visualizzare il grafo in *Neo4j*, è necessario aprire il browser inserendo l'*URI* come *URL*, cambiando i numeri finali, ovvero il codice della porta, con i numeri che si trovano sotto il nome di "HTTP port"; in alternativa sarà sufficiente cliccare sul pulsante "Open" in corrispondenza del progetto.

Una volta effettuato il setting dell'ambiente di sviluppo per la gestione dei risultati precedentemente ottenuti, siamo passati alla realizzazione del grafo in *Neo4j*, per poi passare alle interrogazioni di quest'ultimo.

Per permettere al codice scritto in *Python* di passare le query a *Neo4j* in modo che quest'ultimo le possa eseguire, abbiamo dovuto creare una sessione, come è possibile vedere nel Listato 6.4 che mostra la query utilizzata per resettare il grafo.

```

1 resetQuery = "MATCH (n) DETACH DELETE n"
2
3 with graphDB_Driver.session() as graphDB_Session:
4
5     try:
6
7         graphDB_Session.run(resetQuery)
8
9     except:
10
11         print("EXCEPT 0->NO DELETE ALL")

```

**Listato 6.4:** Reset del grafo

L'utilizzo del costrutto *try-except* ci è servita per non mandare in blocco tutta l'esecuzione del codice nel caso di errore di connessione o di esecuzione della query da parte di *Neo4j*.

Tramite il Listato 6.5 abbiamo implementato la creazione del grafo in *Neo4j*, così da poter procedere con le interrogazioni al grafo stesso mediante questo tool.

```

1 #parola chiave per la query di creazione dei nodi e degli archi in Neo4j
2 query = "CREATE "
3
4 #per ogni parola dell'ontologia
5 for label in lemma_labels_:
6
7     #se e' una chiave del dizionario dei sinonimi
8     if label in synonymous_dict.keys():
9
10        #stringa per creare il nuovo nodo chiave
11        string_key_node = "("+str(label)+":KeyTerm{word:'"+str(label)+"'}),"
12
13        #concatenazione della stringa nella query
14        query = query+string_key_node
15
16        #per ogni sinonimo di quella chiave
17        for synonymous_value in synonymous_dict[label]:
18
19            #stringa per creare un nodo sinonimo
20            string_syn_node = "("+str(synonymous_value)+":SynTerm{word:'"+str(
synonymous_value)+"'}),"
21
22            #stringa per creare un arco tra chiave-sinonimo
23            string_syn_rel = "("+str(label)+")-[:SYNONYMOUS]->("+str(
synonymous_value)+"),"
24
25            #concatenazione della stringa nella query
26            query = query+string_syn_node+string_syn_rel
27
28        #se non ha sinonimi e non e' un sinonimo
29        elif label not in syn_val:
30
31            #stringa per creare un nodo chiave
32            string_key_node = "("+str(label)+":KeyTerm{word:'"+str(label)+"'}),"
33
34            #concatenazione della stringa nella query
35            query = query+string_key_node
36
37 #per tutti i nodi del grafo, quindi tutte le parole trovate (dell'ontologia e
quelle che le descrivono)

```

```

38 for node in B.nodes():
39
40     #se non e' una parola dell'ontologia
41     if node not in lemma_labels_:
42
43         #stringa per la creazione di un nodo termine
44         string_node = "("+str(node)+":Term{word:'"+str(node)+"'}), "
45
46         #concatenazione della stringa nella query
47         query = query+string_node
48
49 #per ogni chiave-valore delle recensioni positive
50 for k,v in label_dict2P.items():
51     #per ogni termine e frequenza del valore
52     for i,f in v[0].items():
53
54         #stringa per la creazione di un arco con attributo di frequenza positiva
55         string_rel = "("+str(k)+")-[:REL{freq_positive:"+str(f)+"}]->("+str(i)+")
56         , "
57
58         #concatenazione della stringa nella query
59         query = query+string_rel
60
61 #per ogni chiave-valore delle recensioni positive
62 for k,v in label_dict2N.items():
63     #per ogni termine e frequenza del valore
64     for i,f in v[0].items():
65
66         #stringa per la creazione di un arco con attributo di frequenza negativa
67         string_rel = "("+str(k)+")-[:REL{freq_negative:"+str(f)+"}]->("+str(i)+")
68         , "
69
70         #concatenazione della stringa nella query
71         query = query+string_rel
72
73 #creazione della sessione
74 with graphDB_Driver.session() as graphDB_Session:
75
76     try:
77         #esecuzione della query
78         graphDB_Session.run(query)
79
80         print("Ha funzionato")
81
82     except:
83
84         print("Non ha funzionato.")

```

Listato 6.5: Realizzazione del grafo in *Neo4j*

Per la creazione dei nodi e degli archi in *Neo4j* da *Python*, abbiamo scoperto che non era possibile lanciare diverse query per la creazione dei singoli nodi, in quanto si generavano continuamente nuovi grafi sconnessi; abbiamo, quindi, utilizzato un'unica query di creazione per generare un unico grafo contenente tutti i nodi e gli archi. Per questo motivo abbiamo concatenato una scritta con le singole istruzioni di creazione.

L'algoritmo è così strutturato:

1. Inizializzazione della stringa *query*, che conterrà l'interrogazione da lanciare.
2. Per ogni parola dell'ontologia:

- Se è una chiave del dizionario dei sinonimi, allora concatenazione della stringa *query* con la stringa per la creazione del nodo "chiave". E, per ogni sinonimo della chiave trovata, concatenazione della stringa *query* con la stringa per la creazione del nodo "sinonimo" e dell'arco chiave-sinonimo.
  - Se, invece, non è un sinonimo e non ha sinonimi, allora concatenazione della stringa *query* con la stringa per la creazione del nodo "chiave".
3. Per ogni nodo del grafo (quindi per tutte le parole trovate da entrambi i *Matcher*, se non è una parola dell'ontologia, allora concatenazione della stringa *query* con la stringa di creazione di un nodo termine.
  4. Per ogni valore delle "chiavi-valore" delle recensioni positive, estrazione di tutti i termini e di tutte le frequenze, e, poi, concatenazione dell'arco con la frequenza positiva.
  5. Per ogni valore delle "chiavi-valore" delle recensioni negative, estrazione di tutti i termini e di tutte le frequenze, e, poi, concatenazione dell'arco con la frequenza negativa.
  6. Apertura della sessione con esecuzione della query.

Per poter eseguire la query, ma mantenere il codice automatico, abbiamo deciso di concatenare una stringa per la creazione di un nodo fittizio, il quale verrà contestualmente eliminato con una query successiva; ciò ci consentirà di escludere dalla stringa finale di *query* la virgola, la quale bloccherebbe l'esecuzione, in quanto *Neo4j* non compilerebbe, rimanendo in attesa di nuove stringhe.

### 6.3.3 Query del grafo ottenuto

Le interrogazioni effettuate sul grafo creato in *Neo4j* sono state definite a priori, per poi essere implementate nel linguaggio *Cypher* e automatizzate all'interno del codice *Python*.

Illustriamo, ora, le query effettuate con le descrizioni dettagliate della loro implementazione.

Di seguito riportiamo le righe di codice che ci hanno permesso di ricavare la parola dell'ontologia con più termini associati (Listato 6.6).

```

1 res0 = {} #creazione del dizionario che conterra' i risultati della query
2 label_not_associated = [] #creazione array che conterra' i termini senza
   associazioni
3
4 #per ogni parola dell'ontologia
5 for label in lemma_labels_:
6
7     #stringa per il conteggio delle parole associate alla parola dell'ontologia
8     query_count = "MATCH (n)-[r]->() WHERE n.word = '"+label+"' RETURN COUNT(r) "
9
10    #apertura sessione
11    with graphDB_Driver.session() as session:
12
13        #esecuzione query
14        result = session.run(query_count)
15
16        #per ogni risultato ottenuto
17        for record in result:
18
19            #estrazione del valore del conteggio
20            res0[label]=record["COUNT(r)"]
21
22        #se il conteggio e' pari a 0

```

```

23         if res0[label] == 0:
24
25             #inserimento della parola dell'ontologia nell'array dei non
associati
26             label_not_associated.append(label)
27
28 #estrazione della parola con il massimo valore di associati
29 max_value = max(res0.values()); {max_key := key for key, value in res0.items()
if value == max_value}
30
31 #stringa di output
32 query1 = "La parola con piu' termini associati e' "+color.BOLD+max_key+color.END+
" con un totale di "+color.BOLD+str(max_value)+color.END+" associati."

```

**Listato 6.6:** Ricerca della parola dell'ontologia con più termini associati

L'algoritmo si sviluppa nel seguente modo:

1. Creazione di un dizionario, che conterrà i risultati della query, e di un array, che conterrà le parole che non hanno termini associati.
2. Per ogni parola dell'ontologia:
  - apertura di una sessione;
  - lancio dell'esecuzione della query *query\_count*, la quale permette, dato un nodo del grafo, il conteggio delle sue relazioni;
  - per ogni risultato:
    - recupero del conteggio effettuato;
    - se il conteggio è uguale a 0, allora la parola dell'ontologia verrà inserita nell'array contenente i nodi senza associati.
3. Ricerca della parola con il massimo valore di termini associati.
4. Concatenazione della stringa di output della query.

Di seguito riportiamo le righe di codice che ci hanno permesso di ricavare la parola dell'ontologia con più termini associati, considerando come associati anche quelli che vengono utilizzati per descrivere i sinonimi della parola dell'ontologia (Listato 6.7).

```

1 #per ogni parola dell'ontologia
2 for label in lemma_labels_:
3
4     #stringa per il conteggio delle parole associate, considerando anche gli
associati dei sinonimi, della parola dell'ontologia
5     query_count = "MATCH (n)-[:SYNONYMOUS]->()-[r]->() WHERE n.word = '"+label+"'
RETURN COUNT(r) "
6
7     #apertura sessione
8     with graphDB_Driver.session() as session:
9
10        #esecuzione query
11        result = session.run(query_count)
12
13        #per ogni risultato ottenuto
14        for record in result:
15
16            #incremento del conteggio degli associati di una dell'ontologia con
il conto degli associati dei sinonimi
17            res0[label]=res0[label]+record["COUNT(r) " ]
18

```

```

19 #estrazione della parola con il massimo valore di associati, considerando gli
    associati dei sinonimi
20 max_value = max(res0.values()); {max_key := key for key, value in res0.items()
    if value == max_value}
21
22 #stringa di output
23 query2 = "La parola con piu' termini associati, considerando i sinonimi, e' "+
    color.BOLD+max_key+color.END+" con un totale di "+color.BOLD+str(max_value)+
    color.END+" associati."

```

**Listato 6.7:** Ricerca della parola dell'ontologia con più associati considerando i sinonimi

L'algoritmo si sviluppa nel seguente modo:

1. Per ogni parola dell'ontologia vengono eseguiti i seguenti passi:
  - apertura di una sessione;
  - lancio dell'esecuzione della query *query\_count*, la quale permette, dato un nodo del grafo, il conteggio delle relazioni dei nodi sinonimi della parola dell'ontologia;
  - per ogni risultato, incremento con il nuovo conteggio calcolato (relativo agli associati "indiretti" della parola dell'ontologia), del valore del conteggio degli associati "diretti" della parola dell'ontologia.
2. Ricerca della parola con il massimo valore di termini associati, considerato l'incremento dei termini associati indirettamente.
3. Concatenazione della stringa di output della query.

Per quanto riguarda la ricerca della parola con più termini associati, considerando solo le recensioni positive, l'unica cosa che cambia, rispetto alla prima query, è l'aggiunta della condizione *r.freq\_positive>0*. Quindi, per semplicità di esposizione, riportiamo solo la riga di codice diversa (Listato 6.8). Quello che la query produce, sostanzialmente, sono i termini utilizzati con più accezioni positive degli altri termini.

```

1 query_count = "MATCH (n)-[r]->() WHERE n.word = '"+label+"'AND (r.freq_positive
    >0) RETURN COUNT(r) "

```

**Listato 6.8:** Ricerca della parola con più termini positivi associati

La query precedente è stata ampliata anche considerando i sinonimi, come per la seconda query, aumentando, quindi, la prima dicitura del *MATCH*. Riportiamo, nel Listato 6.9, l'unica riga che cambia rispetto al Listato 6.7.

```

1 query_count = "MATCH (n)-[:SYNONYMOUS]->()-[r]->() WHERE n.word = '"+label+"'AND
    (r.freq_positive>0) RETURN COUNT(r) "

```

**Listato 6.9:** Ricerca della parola con più termini positivi associati considerando i sinonimi

La terza e la quarta query sono state replicate, considerando le soli recensioni negative, cambiando la condizione in *r.freq\_negative>0* (vedi i Listati 6.10 e 6.11).

```

1 query_count = "MATCH (n)-[r]->() WHERE n.word = '"+label+"'AND (r.freq_negative
    >0) RETURN COUNT(r) "

```

**Listato 6.10:** Ricerca della parola con più termini negativi associati

```

1 query_count = "MATCH (n)-[:SYNONYMOUS]->()-[r]->() WHERE n.word = '"+label+"'AND
    (r.freq_negative>0) RETURN COUNT(r) "

```

**Listato 6.11:** Ricerca della parola con più termini negativi associati considerando i sinonimi

Per la ricerca delle parole dell'ontologia che non sono mai state usate all'interno delle recensioni negative, quindi utilizzate solo con accezioni positive, abbiamo sviluppato il codice riportato nel Listato 6.12.

```

1 list_positive = [] #creazione array che conterra' i termini che non hanno mai
   avuto accezioni negative
2
3 #per ogni parola dell'ontologia
4 for label in lemma_labels_:
5
6     #se non e' senza associati
7     if label not in label_not_associated:
8
9         #inizializzazione del flag a True
10        flag = True
11
12        #stringa per la ricerca dei valori delle frequenze positive
13        query_count = "MATCH (n)-[r]->() WHERE n.word = '"+label+"' RETURN r.
freq_positive"
14
15        #apertura della sessione
16        with graphDB_Driver.session() as session:
17
18            #esecuzione della query
19            result = session.run(query_count)
20
21            #per ogni risultato ottenuto
22            for record in result:
23
24                #estrai il valore della frequenza positiva
25                rel = record["r.freq_positive"]
26
27                #se e' "None"
28                if str(rel) == 'None':
29
30                    #set del flag a False, perche' ha avuto accezioni negative
31                    flag = False
32
33                #se flag e' rimasto True, il termine non ha mai avuto accezioni negative
34                if (flag):
35
36                    #inserisci la parola come termine mai associato negativamente
37                    list_positive.append(label)
38
39 #stringa di output
40 query7 = "Le parole che non hanno mai avuto valori negativi sono:"+color.BOLD+str
   (list_positive)+color.END

```

**Listato 6.12:** Ricerca delle parole dell'ontologia che non sono mai state usate negativamente

Il codice sviluppato prevede i seguenti passi:

1. Inizializzazione di un array vuoto per poter contenere tutti quei termini dell'ontologia che hanno avuto solo accezioni positive.
2. Per ogni parola dell'ontologia che abbia almeno un termine associato:
  - Inizializzazione del *flag* a *True*.
  - Creazione della query per la ricerca dei valori delle frequenze positive, le quali risultano uguali al valore *None* se contengono valori di frequenze negative.
  - Apertura della sessione e lancio della query.

- Per ogni risultato, estrazione del valore della frequenza positiva; qualora questa risulti uguale a "None", allora il *flag* viene portato a *False*.
- Se il *flag* è rimasto a *True*, vuol dire che quel termine non è mai stato utilizzato nelle recensioni negative; quindi, viene aggiunto nell'array contenente le parole dell'ontologia usate solo con accezioni positive.

### 3. Concatenazione della stringa di output.

La ricerca delle parole dell'ontologia che non sono mai state utilizzate con accezioni positive, è stata scritta in modo analogo alla query precedente, ma con la parola "negativo" al posto di "positivo"; quindi non riportiamo tutto il codice, ma solo la query, nel Listato 6.13.

```
1 query_count = "MATCH (n)-[r]->() WHERE n.word = '"+label+"' RETURN r.  
   freq_negative"
```

**Listato 6.13:** Ricerca delle parole dell'ontologia che non sono mai state usate positivamente

Per ricercare il termine utilizzato per descrivere più termini dell'ontologia abbiamo scritto il codice riportato nel Listato 6.14.

```
1 res3 = {} #creazione del dizionario che conterra' i risultati della query  
2  
3 #per ogni parola dell'ontologia  
4 for label in lemma_labels_  
5  
6     #stringa per la ricerca del termine che descrive piu' termini dell'ontologia  
7     query_count = "MATCH ()-[r]->(m) RETURN m.word,COUNT(r) "  
8  
9     #apertura sessione  
10    with graphDB_Driver.session() as session:  
11  
12        #esecuzione query  
13        result = session.run(query_count)  
14  
15        #per ogni risultato della query  
16        for record in result:  
17  
18            #estrazione del conteggio inserito nel dizionario in corrispondenza  
19            del termine considerato  
20            res3[record["m.word"]]=record["COUNT(r)"]  
21  
22 #ricerca del valore massimo dei risultati ottenuti  
23 max_value = max(res3.values()); {max_key := key for key, value in res3.items()  
24     if value == max_value}  
25  
26 #stringa di output  
27 query9 = "La parola associata a piu' termini dell'ontologia e' "+color.BOLD+  
28     max_key+color.END+" con un totale di "+color.BOLD+str(max_value)+color.END+"  
29     chiavi associate."
```

**Listato 6.14:** Ricerca del termine utilizzato per descrivere più parole dell'ontologia

Il codice sviluppato prevede i seguenti passi:

1. Inizializzazione di un dizionario vuoto per poter contenere tutti quei termini utilizzati per descrivere le parole dell'ontologia con il conteggio di quante ne ha associate.
2. Per ogni parola dell'ontologia:
  - Creazione della query per il conteggio delle parole a cui è associato.
  - Apertura della sessione e lancio della query.

- Per ogni risultato, estrazione del conteggio di associazioni ed inserimento nel dizionario in corrispondenza del valore del termine effettivamente considerato.
- Ricerca del valore con il conteggio massimo di associazioni.

### 3. Concatenazione della stringa di output.

Lo stesso codice è stato scritto con l'aggiunta dell'esistenza del termine solo per recensioni positive e negative; riportiamo nei Listati 6.15 e 6.16 le righe di codice con la condizione aggiunta rispetto al listato precedente.

```
1 query_count = "MATCH ()-[r]->(m) WHERE (r.freq_positive>0) RETURN m.word,COUNT(r)"
```

**Listato 6.15:** Ricerca della parola positiva utilizzata per descrivere più termini dell'ontologia

```
1 query_count = "MATCH ()-[r]->(m) WHERE (r.freq_negative>0) RETURN m.word,COUNT(r)"
```

**Listato 6.16:** Ricerca della parola negativa utilizzata per descrivere più termini dell'ontologia

---

## Applicazione dell'approccio proposto: analisi dettagliata del case study

---

*Questo capitolo si concentra sull'applicazione dell'algoritmo ad un caso pratico di studio. Il capitolo inizia con una spiegazione del dataset, fornendo, così, una visione concreta delle analisi che abbiamo effettuato. Successivamente vengono riportati l'ontologia creata e l'elenco delle parole più frequenti trovate. Infine, viene spiegato il metodo di esecuzione delle query con i risultati delle stesse.*

*L'obiettivo di questo capitolo è quello di illustrare i risultati ottenuti nel campo di applicazione di un dataset reale.*

### 7.1 Descrizione dataset

La ricerca del dataset, come già detto, si è concentrata sulle recensioni, proprio per la loro "semi-struttura", grazie alla presenza di parole ripetute nelle diverse recensioni. Questo ci ha permesso di "mappare" tutti quei termini che vengono utilizzati per descrivere le parole "chiave" che rappresentano il campo applicativo che si sta analizzando.

In particolare, abbiamo trovato interessante il dataset contenente più di 515.000 recensioni di hotel di lusso in tutta Europa<sup>1</sup>.

Il dataset presenta diversi attributi, i quali sono riportati di seguito:

- *Hotel\_Address*: via dell'hotel.
- *Review\_Date*: data della recensione.
- *Average\_Score*: media del punteggio dell'anno precedente dell'hotel.
- *Hotel\_Name*: nome dell'hotel.
- *Reviewer\_Nationality*: nazionalità del recensore.
- *Negative\_Review*: testo della recensione negativa; in caso di assenza, si troverà la frase "No Negative".
- *ReviewTotalNegativeWordCounts*: numero totale delle parole della recensione negativa.
- *Positive\_Review*: testo della recensione positiva; in caso di assenza, si troverà la frase "No Positive".
- *ReviewTotalPositiveWordCounts*: numero totale delle parole della recensione positiva.

---

<sup>1</sup><https://www.kaggle.com/datasets/jiashenliu/515k-hotel-reviews-data-in-europe>

- *Reviewer\_Score*: punteggio assegnato all'hotel da parte del recensore.
- *TotalNumberofReviewsReviewerHasGiven*: numero di recensioni effettuate dal recensore.
- *TotalNumberof\_Reviews*: numero totale di recensioni valide dell'hotel.
- *Tags*: tag lasciati dal recensore.
- *dayssincereview*: numeri di giorni passati dalla recensione.
- *AdditionalNumberof\_Scoring*: punteggio dato al servizio.
- *lat*: latitudine dell'hotel.
- *lng*: longitudine dell'hotel.

Tra tutti questi campi, il nostro lavoro si è concentrato su *Negative\_Review* e *Positive\_Review*.

## 7.2 Ontologia estrapolata

Una volta definito il dataset di partenza, siamo potuti passare all'estrapolazione manuale delle recensioni. In particolare, abbiamo considerato 120 righe del dataset, quindi 120 recensioni positive e 120 recensioni negative, per un totale di 240 recensioni. Appuntandoci le parole significative, con la loro frequenza, abbiamo ottenuto complessivamente 139 parole.

### 7.2.1 Le parole più frequenti

Dopo aver completato la procedura sopra esposta, e alla luce del risultato ottenuto, abbiamo deciso di escludere quelle parole che avessero entrambe le frequenze (la frequenza totale e quella relativa al numero di recensioni in cui venivano utilizzate) maggiori di 1, e abbiamo ottenuto 76 parole. Ragionando sul significato di queste parole e sul loro collegamento semantico, siamo arrivati a definire un certo insieme di parole che riportiamo di seguito:

- *location*;
- *restaurant*;
- *food*;
- *bar*;
- *church*;
- *building*;
- *hotel*;
- *price*;
- *cleaning*;
- *level*;
- *service*;
- *park*;
- *bike*;

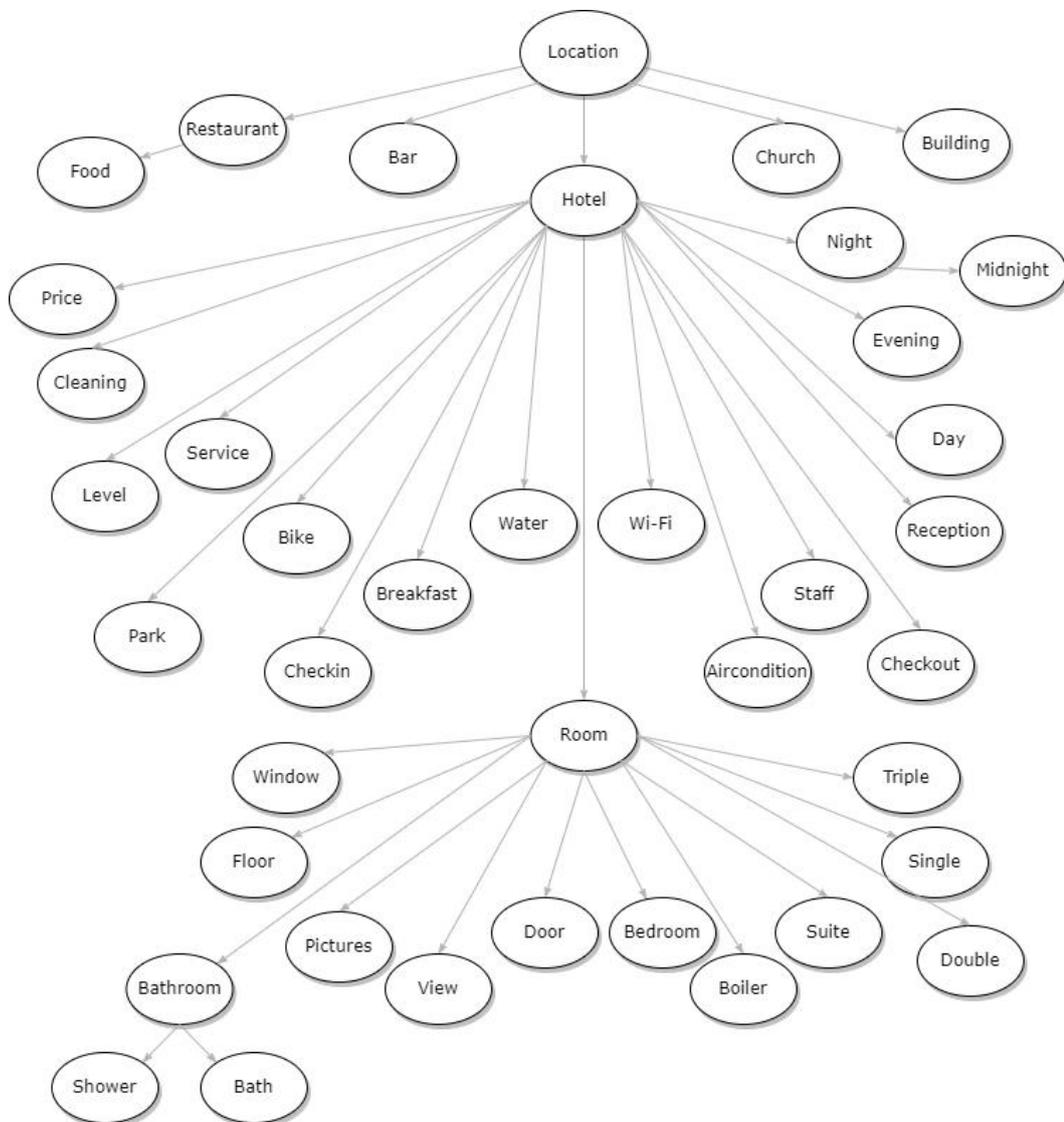
- *checkin;*
- *breakfast;*
- *water;*
- *wi-fi;*
- *aircondition;*
- *staff;*
- *checkout;*
- *reception;*
- *day;*
- *evening;*
- *night;*
- *midnight;*
- *room;*
- *windows;*
- *floor;*
- *pictures;*
- *views;*
- *door;*
- *bedroom;*
- *boiler;*
- *suite;*
- *double;*
- *single;*
- *triple;*
- *bathroom;*
- *shower;*
- *bath;*
- *area;*
- *city;*
- *place;*
- *center;*

- *caffee*.

Queste parole individuate sono state inserite in un dizionario, per poter associare loro come valori tutti quei termini che le descrivono. Poi, abbiamo analizzato queste parole residue, per realizzare un albero gerarchico con quelle più frequenti più significative e tra loro collegate, costruendo, così, l'ontologia riportata in Figura 7.1. Gli ultimi 5 termini dell'elenco non sono stati inseriti nel diagramma per una scelta ben precisa. Infatti, essendo dei sinonimi di parole presenti nell'albero, abbiamo deciso di non inserirli graficamente, ma di tenerli comunque in considerazione. Riportiamo, ora, i sinonimi che abbiamo considerato:

- "Area", "City", "Place" e "Center", considerati come sinonimi di "Location".
- "Caffee", considerato come sinonimo di "Bar".

Questo mapping è stato riportato in un altro dizionario avente come chiavi le parole "Location" e "Bar", mentre come valori i rispettivi sinonimi.

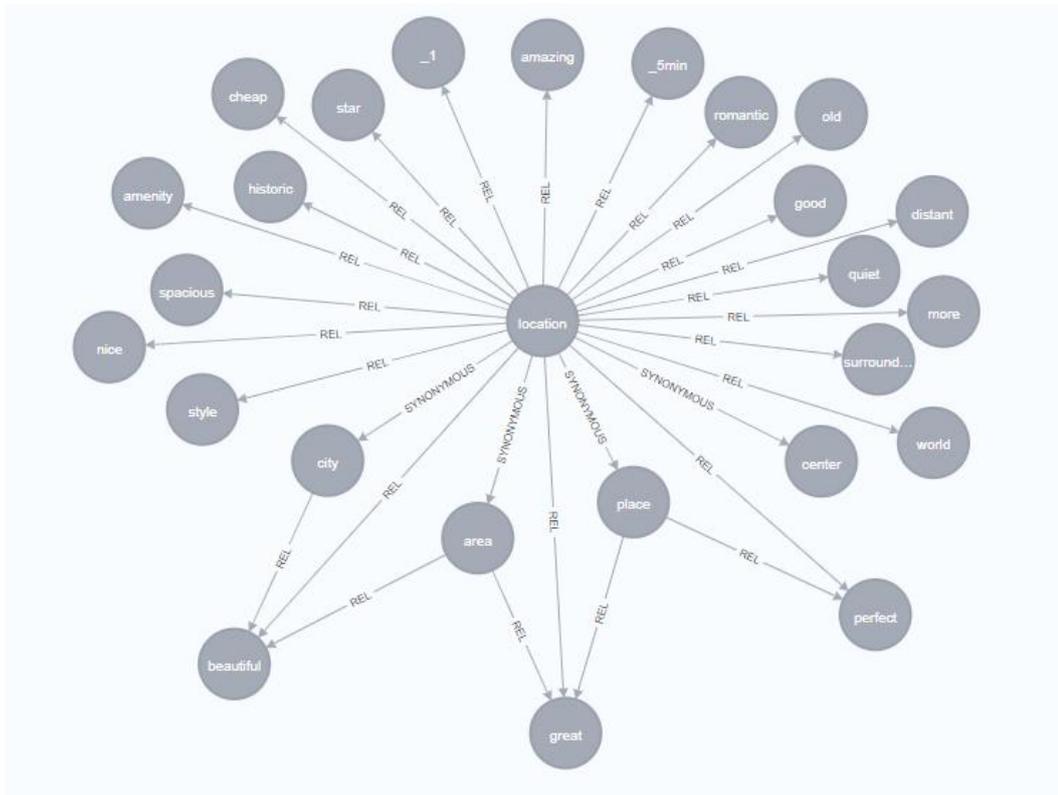


**Figura 7.1:** Ontologia creata dalle recensioni degli hotel

### 7.2.2 Risultati delle query effettuate

Prima di passare alle interrogazioni, mostriamo, in Figura 7.2, un esempio di una parte del grafo ottenuto in *Neo4j* dopo la creazione di tutti i nodi e archi.

Come si può vedere dall'immagine, la parola dell'ontologia "location" è collegata ai suoi sinonimi tramite una relazione di tipo "SYNONYMOUS". Questi sinonimi hanno, a loro volta, altri termini associati e, come si può vedere con i due termini "beautiful" e "great", questi vengono utilizzati sia per descrivere la parola "location" direttamente, che per descrivere indirettamente i suoi sinonimi. Inoltre, ci sono le altre relazioni dirette, definite "REL", le quali sono caratterizzate dagli attributi di frequenza positiva e negativa.



**Figura 7.2:** Esempio di una parte del grafo ottenuto in *Neo4j*

I risultati ottenuti dalle interrogazioni illustrate nel capitolo precedente, vengono mostrati nella tabella in Figura 7.3.

Per poterli analizzare, ci sembra opportuno effettuare un veloce "recap" delle query che abbiamo implementato; esse sono riportate nel seguente elenco:

1. *la parola con più termini associati;*
2. *la parola con più termini associati, considerando i sinonimi;*
3. *la parola con più termini positivi associati;*
4. *la parola con più termini positivi associati, considerando i sinonimi;*
5. *la parola con più termini negativi associati;*
6. *la parola con più termini negativi associati, considerando i sinonimi;*
7. *le parole che non hanno mai avuto valori negativi;*

8. le parole che non hanno mai avuto valori positivi,
9. la parola associata a più termini dell'ontologia;
10. la parola positiva associata a più termini dell'ontologia;
11. la parola negativa associata a più termini dell'ontologia.

La tabella mostra tutti i risultati delle undici query considerando 2.500, 5.000, 10.000, 20.000 e 50.000 recensioni. Si può notare una linea orizzontale che mette in evidenza una separazione tra le prime 5 righe della tabella e le ultime 5, questa divisione è connotata da quali righe sono state prese in considerazione dal dataset originale. In particolare, il primo gruppo fa riferimento all'esecuzione dell'algoritmo sulle prime  $n$  righe del dataset, mentre il secondo gruppo di risultati sono stati prodotti dall'analisi di  $n$  righe del dataset selezionate randomicamente. La scelta di prendere in considerazione l'esecuzione dell'algoritmo su righe causali deriva dal fatto di voler rendere leggermente più eterogeneo il dataset.

N° rev	Query 1	Query 2	Query 3	Query 4	Query 5	Query 6	Query 7	Query 8	Query 9	Query 10	Query 11
2.500	room 686	room 686	room 302	location 380	room 384	room 394	church bike	checkin midnight boiler triple	good 54	good 38	small 22
5.000	room 1134	room 1134	room 408	location 578	room 654	room 654	church bike	midnight triple	good 64	excellent 42	small 32
10.000	room 1734	room 1734	room 678	location 808	room 1056	room 1056	church bike	triple	good 86	good 52	small 44
20.000	room 2688	room 2688	room 1072	location 1318	room 1616	room 1616	church bike	-	good 96	excellent 56	small 58
50.000	room 4228	room 4228	room 1682	location 2274	room 2546	room 2546	-	-	good 122	good 70	small 68
2.500	room 546	room 546	room 250	location 368	room 296	room 296	bike checkin	checkout midnight single	good 40	great 32	bit 18
5.000	room 892	room 892	location 358	location 554	room 540	room 540	-	caffee midnight	good 52	nice 42	small 28
10.000	room 1514	room 1514	room 632	location 1016	room 882	room 883	caffee church checkin	boiler	great 78	great 60	bit 40
20.000	room 2436	room 2436	room 1036	location 1528	room 1400	room 1400	church	boiler	good 118	good 68	good 50
50.000	room 4136	room 4136	room 1628	location 2814	room 2454	room 2455	church	-	good 120	great 68	poor 62

**Figura 7.3:** Tabella dei risultati ottenuti

Alla luce dei risultati ottenuti, possiamo affermare che, per quanto riguarda il dataset delle recensioni degli hotel, la parola sicuramente più utilizzata è *room*, seguita da *location*, che risulta essere il termine descritto da più termini positivi, considerando i sinonimi.

Il primato della parola *room* tra le recensioni dimostra che l'algoritmo è riuscito ad individuare con precisione quello che è, effettivamente, il fattore che viene più recensito nell'analisi di un soggiorno in hotel.

Un'altra considerazione interessante è che la parola *church* non viene utilizzata quasi mai con accezioni negative; questo potrebbe essere dovuto al fatto che tale termine viene usato

per recensire un albergo solo da coloro che sono interessati attivamente e positivamente dal visitare una chiesa.

Notiamo, poi, che il termine usato per descrivere più parole dell'ontologia risulta essere *good* nella maggioranza dei casi, mentre la parola più utilizzata per descrivere negativamente i termini dell'ontologia è quasi sempre *small*. Quest'ultimo risultato è sicuramente dovuto al fatto che il termine stesso è utilizzato come accezione negativa per descrivere un luogo, come, ad esempio, una stanza d'albergo.

Possiamo dire che, analizzando questi risultati nella loro interezza, essi hanno un senso logico, quindi confermano la buona riuscita dell'obiettivo che ci eravamo preposti, ossia la ricerca di quelle parole che risultano caratterizzanti per il contesto applicativo che si sta studiando.

Una volta completato tutto l'iter per l'estrazione delle parole significative, potremo ora modificare alcune parti dell'algoritmo adattandolo al dataset giuridico originale, al fine di poterlo applicare al nostro caso di studio non appena saranno disponibili le note delle sentenze.

---

## Discussione in merito al lavoro svolto

---

*Questo capitolo elenca le modifiche da apportare al codice descritto nel precedente capitolo per applicarlo al mondo giuridico. Successivamente illustra i possibili termini più frequenti estraibili manualmente dalle note delle sentenze, assieme ad una loro rielaborazione il cui obiettivo è quello di creare una possibile ontologia. Il capitolo si conclude con l'elenco di alcune alternative per le query da realizzare.*

*L'obiettivo di questo capitolo è quello di fornire delle linee guida per l'utilizzo dell'algoritmo con le note delle sentenze.*

### 8.1 Applicazione dell'algoritmo con le note delle sentenze

Una volta disponibili le note delle sentenze, basterà modificare degli aspetti nel codice, adattandolo al dataset giuridico, al fine di estrapolare conoscenza sulle sentenze stesse.

#### 8.1.1 Modifiche necessarie

Per prima cosa, definiamo i punti dove saranno necessarie le modifiche da applicare all'algoritmo, adattandolo, così, alle note delle sentenze. Più specificatamente, tali punti sono i seguenti:

- cambiare il nome del dataset da cui estrapolare le informazioni;
- modificare il dizionario contenente tutte le parole dell'ontologia;
- modificare il dizionario contenente il mapping dei vari sinonimi;
- escludere il dualismo "negativo" e "positivo" che avevamo a causa del dataset delle recensioni, quindi anche la differenza tra "frequenza positiva" e "frequenza negativa";
- cambiare l'assegnazione del *doc*, ovvero il documento che il pattern matching prende in esame;
- adattare le query al dataset.

È opportuno precisare alcuni dettagli di una delle modifiche descritte sopra, ossia l'esclusione del dualismo nelle strutture definite sulla base del negativo e del positivo. Nello specifico, possiamo dire che sarà necessario semplificare il codice nel seguente modo:

- Utilizzare solo due dizionari per la creazione delle coppie "parole ontologia-termini associati", invece dei quattro creati nel codice.
- Escludere uno dei due "pattern matching".
- Per la creazione del grafo con *NetworkX*, mantenere solo il primo iter, con la creazione diretta dei nodi, senza la doppia frequenza (positiva e negativa), escludendo la seconda parte della cella che si occupava di creare dei nuovi archi e di modificarne alcuni esistenti.
- Per la creazione del grafo in *Neo4j*, allo stesso modo, escludere la creazione dei nodi per le recensioni positive o negative, mantenendo solo una frequenza (escludendo la differenza in frequenza positiva e negativa).

## 8.2 Ontologia attesa

Ipotizziamo, ora, le parole più frequenti che potrebbero risultare dall'analisi manuale di 200 note circa. I termini che ci aspettiamo sono i seguenti:

- imputato;
- difensore;
- parte;
- condanna;
- accusa;
- avvocato;
- divieto;
- diritto;
- articolo;
- legge;
- giuria;
- giudice;
- italiano;
- sanzione;
- prigioniero;
- domiciliari;
- frode;
- dolo;
- colpa;
- norma;

- obbligo;
- assoluzione;
- condanna;
- reclusione;
- norma;
- comma;
- decreto;
- pecuniaria;
- detentiva;
- civile;
- penale;
- multa;
- restrittiva;
- anni;
- mesi;
- governo;
- costituzione;
- codice.

A questo punto, possiamo ipotizzare un'ontologia che potrebbe essere derivata a valle dell'analisi delle parole frequenti individuate. Più specificatamente, considerando le parole più frequenti evidenziate in precedenza, è possibile derivare l'ontologia mostrata in Figura 8.1.

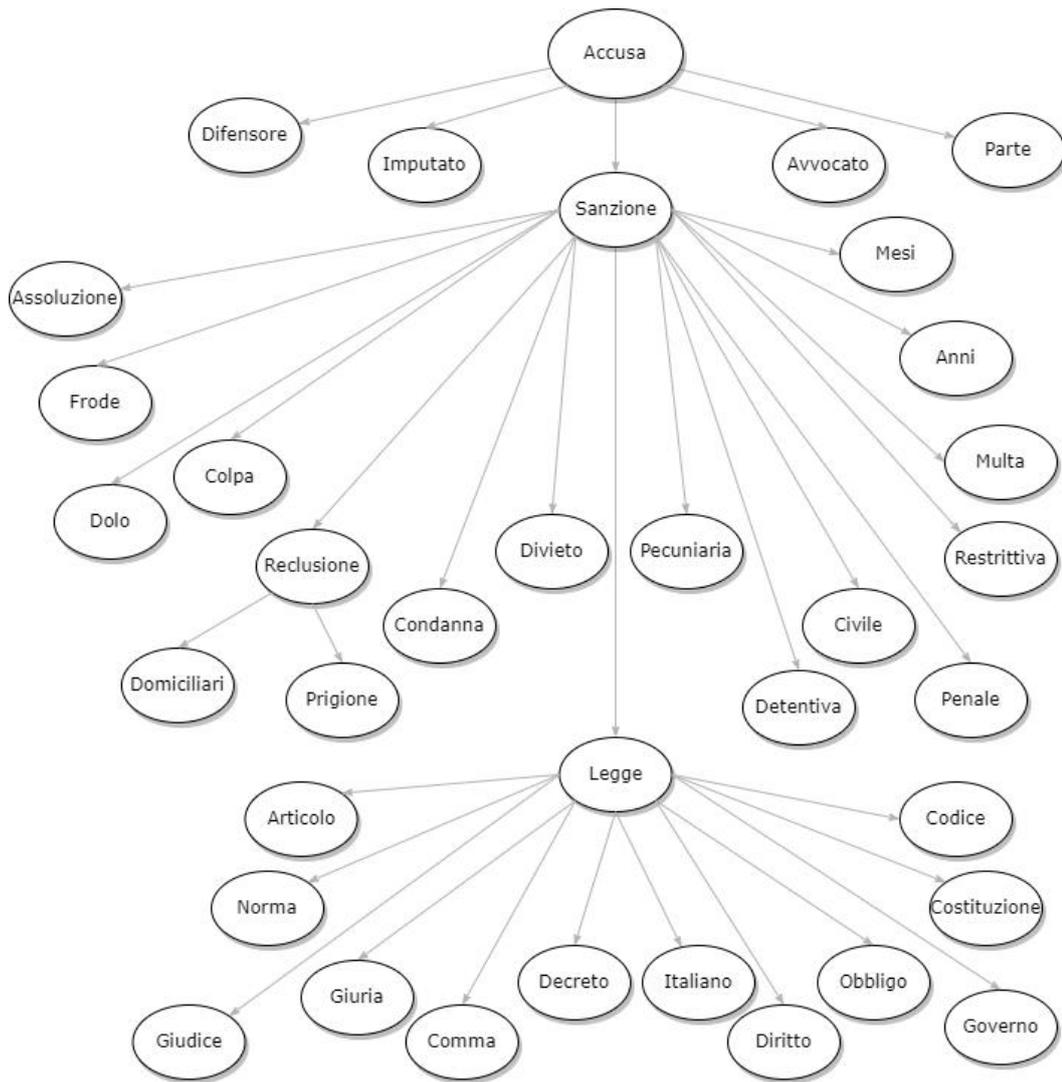
### 8.3 Possibili query di interesse

Sicuramente alcune delle query più interessanti per la nuova ontologia sono identiche a quelle implementate in precedenza, ovvero:

1. *la parola con più termini associati;*
2. *la parola con più termini associati, considerando i sinonimi;*
3. *la parola associata a più termini dell'ontologia.*

Ulteriori possibilità di interrogazioni per un dataset di questo tipo si potrebbero ottenere introducendo dei vincoli. Alcune alternative sono le seguenti:

1. *il termine più associato ad una specifica parola dell'ontologia (es. "articolo");*
2. *la parola dell'ontologia con più sinonimi;*



**Figura 8.1:** Ontologia ipotizzata per le note delle sentenze

3. le parole che non hanno mai avuto valori negativi (non considerando i "\_not\_");
4. le parole che non hanno mai avuto valori positivi (considerando solo i "\_not\_");
5. la parola positiva (non considerando i "\_not") associata a più termini dell'ontologia;
6. la parola negativa (considerando solo i "\_not") associata a più termini dell'ontologia.

Come evidenziato in precedenza, le modifiche da dover apportare al codice non sono numerose e risultano di facile implementazione; quindi, basterà seguire le indicazioni sopra riportate per poter applicare l'algoritmo alle note delle sentenze non appena queste saranno disponibili.

In questa tesi abbiamo percorso un tragitto all'interno di diversi ambiti, partendo dall'analisi del progetto *Uni 4 Justice*, in corso di sviluppo, studiando le motivazioni e i punti di forza che esso prevede di sfruttare al fine di ottimizzare i processi burocratici sulla base di uno sviluppo a livello digitale. Infatti, ci si è concentrati sulla fusione di due mondi diversi, ossia la tecnologia e la giurisprudenza, nell'attuare un processo di digitalizzazione finalizzato all'ottimizzazione dei flussi burocratici giuridici.

Abbiamo realizzato una panoramica della situazione attuale delle richieste della Corte di Appello di Ancona, con un accenno ai sistemi informatici, attualmente in uso, sia nella sezione civile che in quella penale. Tale passaggio è stato necessario per capire il "gap" che vi è tra l'obiettivo fissato dal progetto e la sua attuazione in ottica reale.

A valle di un'analisi della situazione attuale, con le richieste effettive della sezione civile e penale, abbiamo sviluppato una possibile alternativa che mira ad utilizzare il tool di *Microsoft SharePoint*, il quale, per i molteplici motivi individuati e spiegati nel relativo capitolo, sembra essere un valido strumento che si adatta alle esigenze informatiche della Corte di Appello di Ancona. Per dare una visione più concreta di questo tool, abbiamo anche realizzato un sito in *SharePoint*, spiegando le varie possibilità di personalizzazione per la gestione del sito e le alternative per la gestione degli utenti in base al tipo di sito creato.

Essendo *Akoma Ntoso* un aspetto imprescindibile su cui il progetto *Uni 4 Justice* fa affidamento, abbiamo deciso di approfondire l'argomento, mostrando in dettaglio lo standard, al fine di fornire un'opportuna documentazione, quindi una sorta di "manuale d'uso" veloce ed intuitivo. Infatti, nel capitolo dedicato, si possono trovare tutti i "tag" consentiti dallo standard e il loro specifico utilizzo.

Siamo, poi, passati alla parte di progettazione, implementazione e realizzazione dell'algoritmo per la manipolazione delle note delle sentenze con lo scopo di estrapolare i concetti fondamentali dalle note stesse. Siamo partiti dalla progettazione, ossia dalla definizione dell'approccio proposto, descrivendo i passaggi effettuati a livello astratto. Successivamente, ci siamo concentrati sull'implementazione dell'approccio, che si è sviluppato in una iniziale ricerca manuale, seguita dall'implementazione del codice. Abbiamo, poi, preso in esame un caso di studio reale, purtroppo diverso dalle note delle sentenze, in quanto mancanti, ma pur sempre con la stessa "struttura", e abbiamo concretizzato in "risultati effettivi" ciò che era stato progettato ed implementato. Infine, abbiamo definito come applicare il metodo proposto con le note delle sentenze, elencando le modifiche da fare e alcuni risultati attesi.

A valle di tutto l'iter sopra descritto, vediamo come possibilità di sviluppo futuro di questo progetto di tesi tre principali "step".

Il primo passo da fare è, sicuramente, quello di colmare il "gap" che c'è tra le esigenze della Corte di Appello di Ancona e il punto di partenza per l'attuazione del progetto *Uni 4 Justice*; questo processo di allineamento può basarsi sull'introduzione di *Microsoft SharePoint* nei sistemi informativi della Corte.

Il secondo passo è l'effettiva applicazione dell'approccio implementato, questa volta nel campo giuridico, con l'analisi delle note delle sentenze non appena queste saranno disponibili, partendo dall'estrazione manuale dell'ontologia, per poi passare all'esecuzione dell'algoritmo effettuando le modifiche che abbiamo descritto nel capitolo dedicato.

Il terzo passo, invece, è quello di valutare la complessità computazionale dell'algoritmo, cercando di ottimizzare il codice, al fine di poterlo utilizzare su una grande quantità di dati. Ad esempio, si potrebbe pensare di sviluppare un'euristica per rendere più efficiente il codice; a tal fine potrebbero risultare utili tecniche di ottimizzazione come il "*branch-and-bound*".

- BARABUCCI, G., CERVONE, L., PALMIRANI, M., PERONI, S. e VITALI, F. (2009), «Multi-layer markup and ontological structures in Akoma Ntoso», in «International Workshop on AI Approaches to the Complexity of Legal Systems», p. 133–149, Springer.
- BATES, S., SMITH, A. e SMITH, R. (2010), *SharePoint 2010 User's Guide: Learning Microsoft's Business Collaboration Platform*, Apress.
- BLAKE, C. (2011), «Text mining», *Annual review of information science and technology*, vol. 45 (1), p. 121–155.
- BOLASCO, S. e OTHERS (2005), «Statistica testuale e text mining: alcuni paradigmi applicativi», *Quaderni di statistica*, vol. 7, p. 17–53.
- CARTER, E., SCHOLL, B. e JAUSOVEC, P. (2010), *SharePoint 2010 Development with Visual Studio 2010*, Addison-Wesley Professional.
- CHOPRA, A., PRASHAR, A. e SAIN, C. (2013), «Natural language processing», *International journal of technology enhancements and emerging engineering research*, vol. 1 (4), p. 131–134.
- CHOWDHARY, K. (2020), «Natural language processing», *Fundamentals of artificial intelligence*, p. 603–649.
- COHEN, K. B. e HUNTER, L. (2008), «Getting started in text mining», *PLoS computational biology*, vol. 4 (1), p. e20.
- COLIC, N. e RINALDI, F. (2019), «Improving spaCy dependency annotation and PoS tagging web service using independent NER services», *Genomics & informatics*, vol. 17 (2).
- CONNELL, A. (2008), *Professional SharePoint 2007 Web Content Management Development: Building Publishing Sites with Office SharePoint Server 2007*, John Wiley & Sons.
- COVENTRY, P. (2010), *Microsoft SharePoint Designer 2010 step by step*, Pearson Education.
- DRISGILL, R., ROSS, J. e STUBBS, P. (2013), *Sharepoint 2013 branding and user interface design*, John Wiley & Sons.
- FAN, W., WALLACE, L., RICH, S. e ZHANG, Z. (2006), «Tapping the power of text mining», *Communications of the ACM*, vol. 49 (9), p. 76–82.
- FEINERER, I., HORNİK, K. e MEYER, D. (2008), «Text mining infrastructure in R», *Journal of statistical software*, vol. 25, p. 1–54.

- FELDMAN, R., FRESKO, M., KINAR, Y., LINDELL, Y., LIPHSTAT, O., RAJMAN, M., SCHLER, Y. e ZAMIR, O. (1998), «Text mining at the term level», in «European Symposium on Principles of Data Mining and Knowledge Discovery», p. 65–73, Springer.
- FERGUSON, R. (2002), *Special edition using Microsoft SharePoint portal server*, Que Publishing.
- FOX, S. (2010), *Beginning SharePoint 2010 Development*, John Wiley & Sons.
- GAIKWAD, S. V., CHAUGULE, A. e PATIL, P. (2014), «Text mining methods and techniques», *International Journal of Computer Applications*, vol. 85 (17).
- GARRETT, R. e COHEN-DUMANI, D. (2013), *Pro SharePoint 2013 Administration*, Springer.
- GEN, K., AKIRA, N., MAKOTO, N., YASUHIRO, O., TOMOHIRO, O. e KATSUHIKO, T. (2016), «Applying the Akoma Ntoso XML schema to Japanese legislation», *Journal of Law, Information and Science*, vol. 24 (2), p. 49–70.
- GOLDBERG, Y. (2016), «A primer on neural network models for natural language processing», *Journal of Artificial Intelligence Research*, vol. 57, p. 345–420.
- HASSANI, H., BENEKI, C., UNGER, S., MAZINANI, M. T. e YEGANEHI, M. R. (2020), «Text mining in big data analytics», *Big Data and Cognitive Computing*, vol. 4 (1), p. 1.
- HILD, E. e WACH, C. (2010), *Pro SharePoint 2010 Solution Development: Combining .NET, SharePoint, and Office 2010*, Apress.
- HILLIER, S. (2007), *Microsoft SharePoint*, Springer.
- HIRSCHBERG, J. e MANNING, C. D. (2015), «Advances in natural language processing», *Science*, vol. 349 (6245), p. 261–266.
- HOLLAND, C., CATIGNANI, M. e PORTER, C. (2011), *Microsoft SharePoint 2010 Web Applications: The Complete Reference*, McGraw-Hill.
- HOLLIDAY, J. (2011), *Professional SharePoint 2007 Records Management Development: Managing Official Records with Microsoft Office SharePoint Server 2007*, John Wiley & Sons.
- HOTH, A., NÜRNBERGER, A. e PAASS, G. (2005), «A brief survey of text mining.», in «Ldv Forum», vol. 20, p. 19–62, Citeseer.
- JOSHI, A. K. (1991), «Natural language processing», *Science*, vol. 253 (5025), p. 1242–1249.
- JUGRAN, S., KUMAR, A., TYAGI, B. S. e ANAND, V. (2021), «Extractive automatic text summarization using SpaCy in Python & NLP», in «2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)», p. 582–585, IEEE.
- KANNAN, S., GURUSAMY, V., VIJAYARANI, S., ILAMATHI, J., NITHYA, M., KANNAN, S. e GURUSAMY, V. (2014), «Preprocessing techniques for text mining», *International Journal of Computer Science & Communication Networks*, vol. 5 (1), p. 7–16.
- KAO, A. e POTEET, S. R. (2007), *Natural language processing and text mining*, Springer Science & Business Media.
- KOBAYASHI, V. B., MOL, S. T., BERKERS, H. A., KISMIHÓK, G. e DEN HARTOG, D. N. (2018), «Text mining in organizational research», *Organizational research methods*, vol. 21 (3), p. 733–765.

- LONDER, O. e COVENTRY, P. (2013), *Microsoft SharePoint 2013 Step by Step*, vol. 688, Microsoft Press.
- MALIK, S. (2010), *Microsoft Sharepoint 2010: building solutions for Sharepoint 2010*, Apress.
- MANNING, C. e SCHUTZE, H. (1999), *Foundations of statistical natural language processing*, MIT press.
- MEURERS, D. (2012), «Natural language processing and language learning», *Encyclopedia of applied linguistics*, p. 4193–4205.
- NADKARNI, P. M., OHNO-MACHADO, L. e CHAPMAN, W. W. (2011), «Natural language processing: an introduction», *Journal of the American Medical Informatics Association*, vol. 18 (5), p. 544–551.
- NOBLE, J., PIDDOCKE, R. e BAKMAND-MIKALSKI, D. (2011), *Pro SharePoint 2010 Search*, Springer.
- PALMIRANI, M. (2011), «Legislative change management with Akoma-Ntoso», in «Legislative XML for the semantic Web», p. 101–130, Springer.
- PALMIRANI, M. (2019), «Akoma Ntoso for making FAO resolutions accessible», *Knowledge of the Law in the Big Data Age. Frontiers in Artificial Intelligence and Applications*, vol. 317, p. 159–169.
- PALMIRANI, M. e VITALI, F. (2011), «Akoma-Ntoso for legal documents», in «Legislative XML for the semantic Web», p. 75–100, Springer.
- POOLE, M. (2008), *SharePoint Designer Tutorial: Working with SharePoint Websites*, Packt Publishing Ltd.
- RAINA, V. e KRISHNAMURTHY, S. (2022), «Natural language processing», in «Building an Effective Data Science Practice», p. 63–73, Springer.
- RAJMAN, M. e BESANÇON, R. (1998), «Text mining: natural language techniques and text mining applications», in «Data mining and reverse engineering», p. 50–64, Springer.
- TAN, A.-H. e OTHERS (1999), «Text mining: The state of the art and the challenges», in «Proceedings of the pakdd 1999 workshop on knowledge discovery from advanced databases», vol. 8, p. 65–70, Citeseer.
- TRYBULA, W. J. (1999), «Text mining», *Annual Review of Information Science and Technology (ARIST)*, vol. 34, p. 385–419.
- VITALI, F. e ZENI, F. (2007), «Towards a country-independent data format: the Akoma Ntoso experience», in «Proceedings of the V legislative XML workshop», p. 67–86.
- WICKLUND, P. (2011), *SharePoint 2010 Workflows in Action*, Simon and Schuster.
- WITTEN, I. H., DON, K. J., DEWSNIP, M. e TABLAN, V. (2004), «Text mining in a digital library», .
- YOUNG, S., CARAVAJAL, S., KLINDT, T. e REHMANI, A. (2014), *Professional SharePoint 2013 Administration eBook And SharePoint-videos. com Bundle*, John Wiley & Sons.

## Siti web consultati

- Corte di Appello di Ancona – <https://www.giustizia.marche.it/>
- Microsoft SharePoint – <https://www.microsoft.com/it-it/microsoft-365/sharepoint/collaboration>
- Akoma Ntoso – <http://www.akomantoso.org/>
- Microsoft SharePoint Server – <https://www.microsoft.com/it-it/microsoft-365/sharepoint/sharepoint-server>
- Spacy, Industrial-Strength Natural Language Processing – <https://spacy.io/>
- Neo4j, Graph Data Platform – <https://neo4j.com/>
- Wikipedia – [www.wikipedia.org](http://www.wikipedia.org)

---

## Ringraziamenti

---

Il primo ringraziamento è, senz'altro, per il Professore Domenico Ursino per la sua competenza, gentilezza e umanità con cui mi ha aiutata e seguita per conseguire il tirocinio e scrivere la tesi. Professore la ringrazio per esser stato una guida nell'affrontare determinate scelte importanti per il mio futuro. Ringrazio il Dott. Francesco Cauteruccio per avermi guidata nell'implementazione dell'algoritmo e ringrazio i dottorandi Gianluca Bonifazi e Luca Virgili per il supporto nel periodo del tirocinio.

Un ringraziamento speciale va a mia mamma Giovanna; senza di lei ora non sarei qui a concludere un percorso così impegnativo. Mamma, senza di te non avrei mai avuto il coraggio di provarci, la forza nell'affrontare le difficoltà che ho incontrato. Grazie; senza il tuo supporto e amore non sarei mai arrivata fin qui. Grazie per essere il mio punto di riferimento nei momenti più difficili.

Ringrazio i miei fratelli, Simona e Gianmauro, per sopportarmi ogni giorno dentro casa; grazie per darmi sempre la grinta per andare avanti.

Ringrazio il mio ragazzo Davide, per essere sempre al mio fianco da anni, sopportandomi e accettandomi per quello che sono. Davide, grazie per darmi sempre la forza per mettercela tutta e per l'aiuto che mi dai ogni giorno. Ringrazio anche Giorgio e Patrizia per essere sempre così tanto orgogliosi di me.

Ringrazio mia cugina Evelyn, per l'affetto che mi trasmette continuamente, e con lei ringrazio tutti i miei zii e cugini.

Ringrazio i miei amici storici dell'università Marco, Lara, Fabi, Chiara, Ele, Marti, Giuli, Ric e Nico per avermi accompagnata nei primi anni in questo mondo universitario. Grazie anche a Mirko e Lorenzo per aver condiviso con me il percorso magistrale, sopportandomi nello studio e nella realizzazione dei progetti.

Ringrazio tutti i miei amici Giovi, Enri, Mary, Ciuffo, Robi, Kiko, Ali, Lucio, Giovi, Giuli, Memi, Edo, Vane, Pietro, Naty, Pole, Eddi e Ani per esserci sempre.

Ringrazio mio padre e anche mia nonna Letta che, con la sua saggezza, mi ha sempre tranquillizzato prima di ogni esame.

Un ultimo ringraziamento speciale va a mio nonno Gennaro, per aver sempre creduto in me, per avermi sempre guardato con orgoglio e per avermi dato tanto amore; mi manchi.