GENERAZIONE DI UN GRAFO DI CONOSCENZA IN AMBITO GIURIDICO PER I CASI DI VIOLENZA SULLE DONNE

INDICE

1

PROCESSO DI SVILUPPO

PROCESSO DI SVILUPPO DI UN KNOWLEDGE GRAPH

L'articolo «*Defining a Knowledge Graph Development Process Through a Systematic Review*» di Gyte Tamašauskaité e Paul Groth propone un metodo di sviluppo di un knowledge graph partendo da dati strutturati, semi-struturati o non strutturati

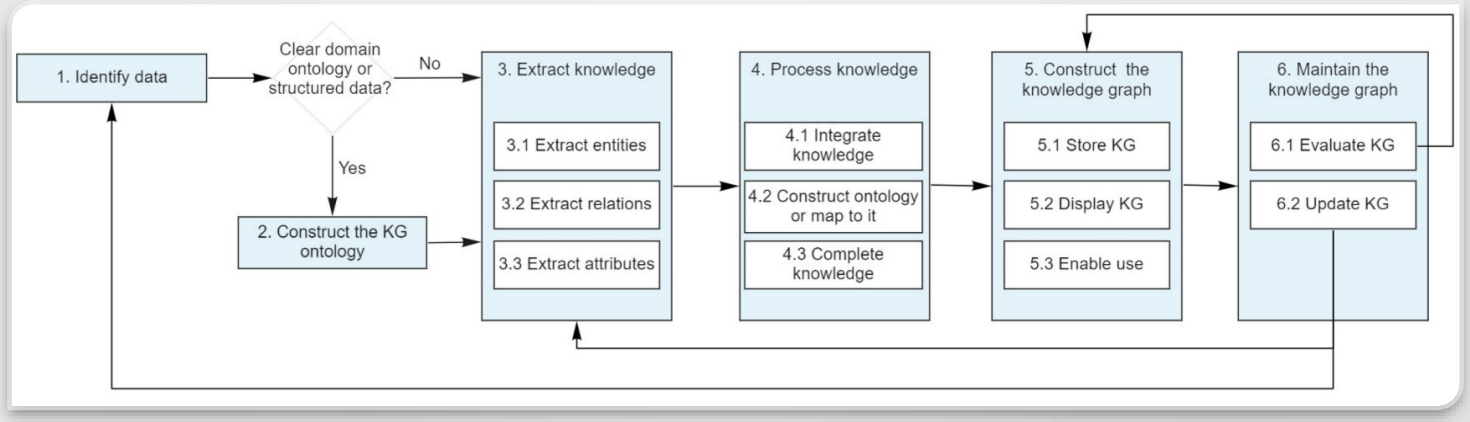
TIPOLOGIE DI SVILUPPO

- **Top-down**: si definisce un’ontologia e basandosi su questa si estrae la conoscenza dai dati
- **Bottom-up**: si estrae la conoscenza dai dati e basandosi su questa si crea un’ontologia

METODO PROPOSTO

Il metodo proposto si articola in sei diverse fasi:

1. Identificazione dei dati
2. Costruzione dell'ontologia
3. Estrazione della conoscenza
4. Processing della conoscenza
5. Costruzione del knowledge graph
6. Manutenzione



1. IDENTIFICAZIONE DEI DATI

- Si definisce il **dominio** di interesse e da quali **sorgenti** i dati verranno estrapolati, che siano essi strutturati, semi-strutturati o non strutturati

- I dati acquisiti saranno successivamente usati per l’estrazione della conoscenza

2. COSTRUZIONE DELL'ONTOLOGIA

- Necessaria nel caso si sia scelta una strategia di sviluppo **top-down**
- Si definiscono i tipi di **entità** e le **relazioni** che intercorrono tra esse usando delle ontologie di utilizzo comune come FOAF, ontologie esistenti rilevanti per il dominio di interesse, oppure definendole con linguaggi come RDFS o OWL

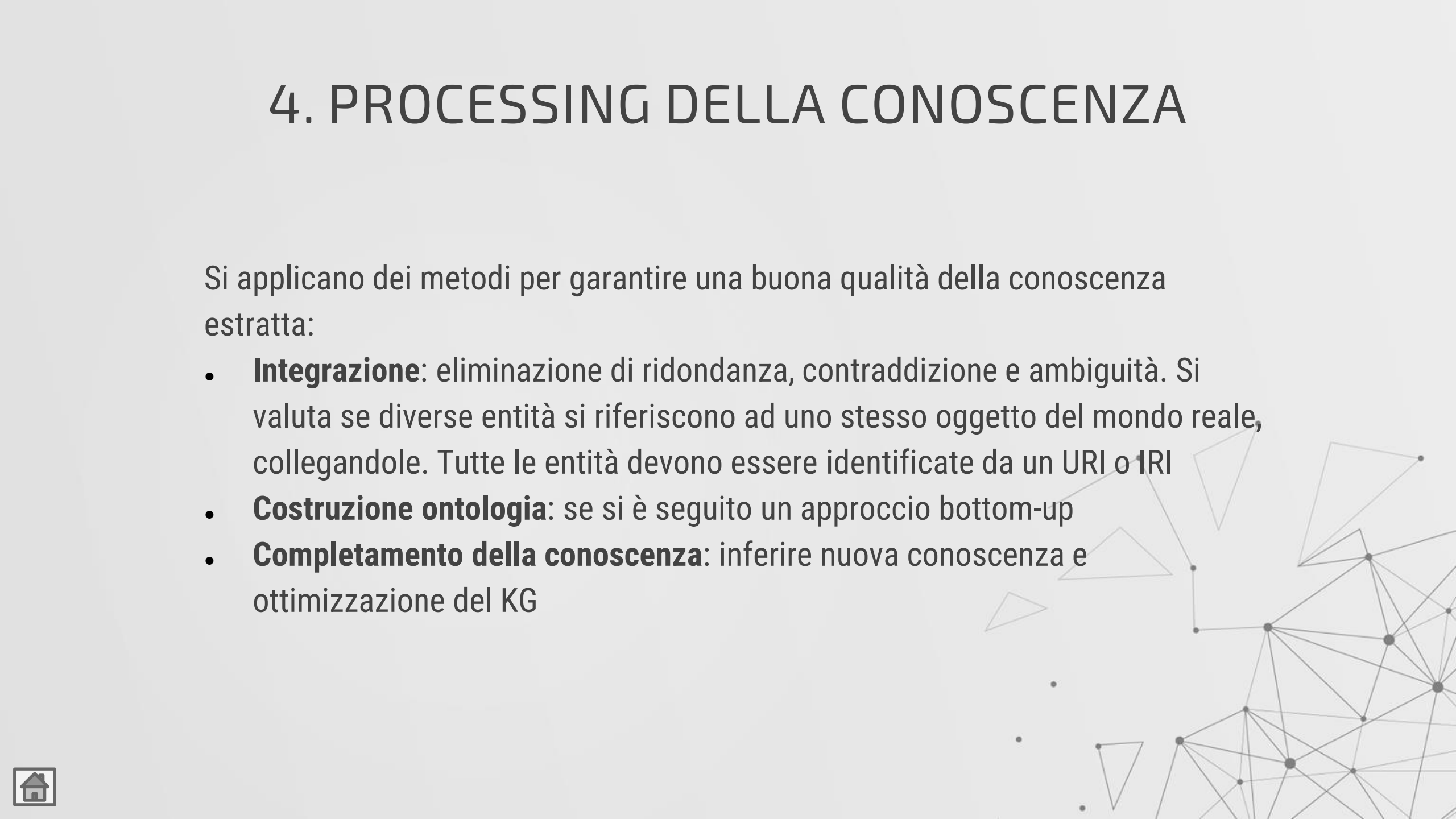
3. ESTRAZIONE DELLA CONOSCENZA

- Partendo dai dati raccolti l'obiettivo è quello di estrarre entità, relazioni e attributi. Si suddivide in tre parti:
    1. **Estrazione delle entità**: si utilizza Named Entity Recognition il quale trova e classifica le entità in categorie predefinite
    2. **Estrazione delle relazioni**: nel caso di dati non strutturati si utilizza il Natural Language Processing. Se disponibile un'ontologia si cercano le informazioni che la ricalcano.
    3. **Estrazione degli attributi**: si cercano le informazioni che permettono di descrivere meglio le entità
- Alla fine di questa fase è possibile costruire le **triple** utili per creare il KG

4. PROCESSING DELLA CONOSCENZA

Si applicano dei metodi per garantire una buona qualità della conoscenza estratta:

- **Integrazione**: eliminazione di ridondanza, contraddizione e ambiguità. Si valuta se diverse entità si riferiscono ad uno stesso oggetto del mondo reale, collegandole. Tutte le entità devono essere identificate da un URI o IRI
- **Costruzione ontologia**: se si è seguito un approccio bottom-up
- **Completamento della conoscenza**: inferire nuova conoscenza e ottimizzazione del KG



5. COSTRUZIONE DEL KG

- **Memorizzazione**: memorizzare il KG in un formato appropriato
- **Visualizzazione**: rendere il KG facilmente visualizzabile (es Neo4j)
- **Uso**: permettere l'interrogazione del KG

6. MANUTENZIONE

- **Valutazione**: usare feedback degli utenti per identificare problemi o mancanze del KG
- **Aggiornamento**: permettere l'aggiunta di nuova conoscenza se ci sono nuovi dati da rappresentare, partendo dalle sorgenti già considerate o da nuove

2

ONTOLOGIE GIURIDICHE

ONTOLOGIE GIURIDICHE

Nell'articolo «*The linked legal data landscape: linking legal data across different countries*» di Erwin et al. si crea un knowledge graph giuridico per l'Austria. Per quanto seguano un approccio top-down, quindi creano un'ontologia su cui si basano per popolare il knowledge graph, è importante l'uso che fanno delle ontologie **ELI** e **ECLI**, oltre che di **EuroVoc**

EUROPEAN LAW IDENTIFIER

ELI è uno standard creato per identificare i documenti legislativi e i loro metadati per gli stati europei in cui si identifica:

- eli:**LegalResource** è una creazione intellettuale distinta come un atto legale, di uno specifico eli:**type_document**, ad esempio una direttiva, che è realizzato da un eli:**LegalExpression**
- eli:**LegalExpression** ha un eli:**title** ed eli:**realizes** la versione base in una lingua particolare (eli:**language**) di un eli:**LegalResource**. È pubblicato in un eli:**Format** che è la rappresentazione fisica come HTML o PDF.

ELI segue i principi dell'ontologia **FRBRoo** utilizzata per le pubblicazioni bibliografiche

EUROPEAN CASE LAW IDENTIFIER

- ECLI identifica i case law (giurisprudenza) e definisce l'insieme minimo di metadati per i documenti giudiziari.
- Un identificatore è suddiviso in cinque parti, separate da due punti come nell'esempio «ECLI:AT:OGH0002:2016:0100OB00012.16M.1220.000» :
    1. **Sigla ECLI**
    2. **Codice dello stato** o dell'organizzazione internazionale
    3. **Codice della corte** che ha preso la decisione
    4. **Anno** della decisione
    5. **Numero univoco** ordinale della decisione

METADATI ECLI

Usa le proprietà dell'ontologia Dublin Core Metadata Initiative (DCMI), senza aggiungere nuove proprietà, ma raccomanda l'uso delle seguenti:

- dcterms:**identifier**: URL in cui è possibile recuperare la risorsa
- dcterms:**isVersionOf**: indica che una risorsa è una versione di un'altra risorsa
- dcterms:**creator**: nome completo del tribunale decidente
- dcterms:**coverage**: indica il paese in cui ha sede la corte o il tribunale
- dcterms:**date**: la data in cui è stata emessa una decisione
- dcterms:**language**: la lingua in cui è scritto
- dcterms:**publisher**: l'organizzazione responsabile della pubblicazione del documento
- dcterms:**accessRights**: definisce chi può accedere alla risorsa, pubblica o privata
- dcterms:**type**: definisce il tipo di decisione resa

RELAZIONE TRA ECLI ed ELI

Nel paragrafo 13 del documento Conclusioni del Consiglio che invitano all'introduzione dell'identificatore della legislazione europea (ELI), contenuto nella Gazzetta ufficiale dell'Unione Europea si afferma che:

«L'identificatore europeo della giurisprudenza (ECLI), applicabile su base volontaria, fornisce già un sistema europeo per l'identificazione della giurisprudenza. ELI identifica testi legislativi aventi caratteristiche diverse e più complesse, e i due sistemi sono complementari.»

EUROVOC

- Il thesaurus EuroVoc è un thesaurus **multidominio** e **multilingue** fornito dall'Ufficio delle pubblicazioni dell'Unione europea (OP) utilizzato per classificare i documenti dell'UE in categorie per facilitare la ricerca delle informazioni.
- Si basa sullo standard Simple Knowledge Organization System (SKOS), utilizzato per rappresentare le informazioni usando RDF.
- Ogni termine di EuroVoc è di tipo skos:**Concept** e più termini possono essere aggregate in un skos:**ConceptScheme**
- I concetti sono collegati con skos:**narrower** e skos:**broader** per rappresentare la struttura gerarchica e si usa skos:**related** per le relazioni associative
- Ogni concetto ha un termine preferito e altri alternativi, indicate con skos:**prefLabel** e skos:**altLabel**

3

SISTEMI SIMILI

SISTEMI SIMILI

Di seguito sono illustrati dei sistemi che costruiscono knowledge graph in ambito giuridico. Si può notare come il processo utilizzato in ogni esempio si possa ricondurre a quello mostrato in precedenza.

Gli articoli presi in considerazione sono:

- *Developing Italy's legal knowledge graph*
- *Legal knowledge extraction for KG-based question answering*
- *Constructing a KG for Vietnamese legal cases*

ITALY'S KNOWLEDGE GRAPH

Si crea una pipeline per estrarre le informazioni dai documenti prodotti dall'Istituto Poligrafico Zecca dello Stato, rendendoli facilmente interrogabili. Il sistema è formato dai seguenti moduli:

1. **Law description extractor**: carica la banca dati dell'IPZS
2. **Law description converter**: i dati vengono convertiti in triple, usando prefissi e predicati usati in LOD
3. **Triplestore**: carica le triple in un grafo RDF dando la possibilità di interrogarlo e aggiornarlo
4. **Dereferencer**: permette di pubblicare i dati in formato RDF
5. **Graph browser**: permette di navigare la struttura del grafo
6. **SPARQL endpoint**: fornisce l'API e l'interfaccia per interrogare il grafo
7. **Dashboard**: come il precedente ma mette a disposizione query preimpostate per chi non ha familiarità con i linguaggi di interrogazione

LEGAL KNOWLEDGE EXTRACTION FOR KG-BASED QUESTION ANSWERING

Si crea un sistema in grado di rispondere a domande poste per cercare informazioni legate a specifici argomenti, concetti o entità. Per estrarre le informazioni contenute in documenti si seguono questi passi:

1. **KG extraction**: si estraggono concetti e relazioni dal testo, si assegnano URI e label RDF ai nodi. Si aggiungono anche triple speciali per tenere traccia dei frammenti di testo in cui sono stati estratti i concetti
2. **Taxonomy construction**: si estrae la tassonomia di tipi/classi dei concetti rappresentati attraverso Formal Concept Analysis
3. **Legal ODP alignment**: si allinea la struttura a design pattern di ontologie
4. **Question answering**: dato una domanda in linguaggio naturale restituisce i risultati rilevanti

KG FOR VIETNAMESE LEGAL CASES

Si crea un knowledge graph eterogeneo partendo da leggi e casi legali. I passi per la creazione del KG sono i seguenti:

1. **Data crawling**: ricerca su siti governativi di leggi e casi legali
2. **Information Extraction**: estrazione di entità e relazioni dai file ritrovati
3. **KG deployment**: si crea il grafo, i cui nodi sono del tipo *case*, *domain*, *court*, *law*. Le relazioni estratte sono nella forma: (court, *decide*, case), (case, *belongTo*, domain), (case, *basedOn*, law)

4

PIPELINE PROPOSTA

PIPELINE

Basandosi sul processo di creazione di KG visto in precedenza si può seguire una strategia bottom-up formata dai seguenti passi:

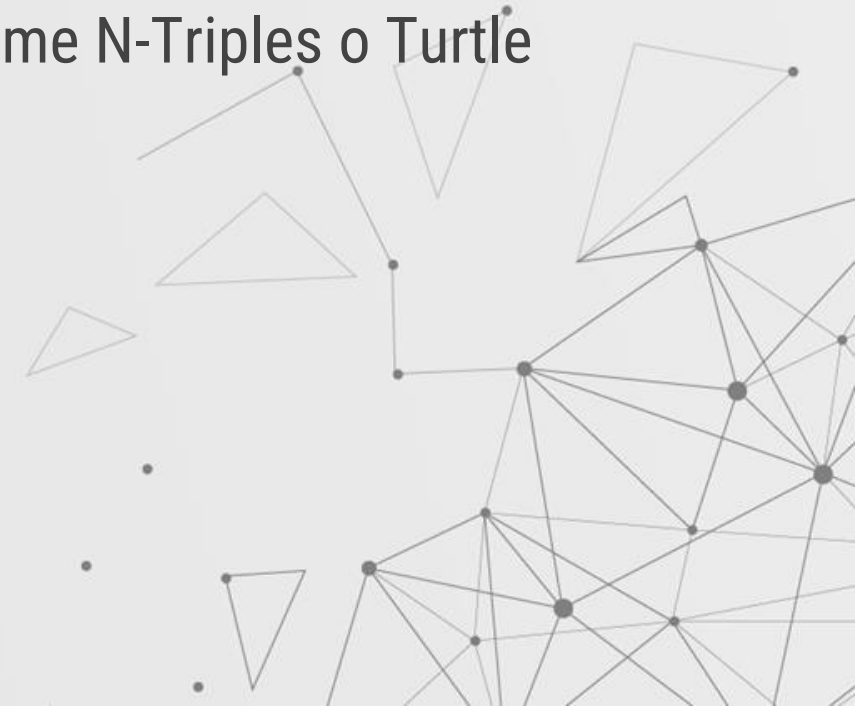
1. Estrazione della conoscenza
2. Integrazione delle triple
3. Creazione dell'ontologia
4. Costruzione del knowledge graph

1. ESTRAZIONE DELLA CONOSCENZA

- Partendo dai file contenenti le pronunce si estrae la conoscenza sotto forma di triple, utilizzando NLP o LLM.
- Delle prime informazioni sono quelle contenute nella scheda «Case Details» di ogni pronuncia (disponibili solo nell'HTML)
    - In questa tabella è stata annotata la presenza di ogni possibile campo nel documento
    - Esempio dei dati presenti

2. INTEGRAZIONE DELLE TRIPLE

- Eliminazione di eventuale ridondanza e inconsistenza nelle triple ottenute
- Identificare le entità con un URI o IRI
- Ottenere delle triple in un formato standard di RDF come N-Triples o Turtle



3. CREAZIONE DELL'ONTOLOGIA

- Si crea un'ontologia partendo da quelle esistenti nel dominio legale ed, eventualmente, espandendole
- Tra le ontologie esistenti sembra essere più importante l'uso di ECLI rispetto ad ELI, in quanto stiamo trattando della giurisprudenza e non delle leggi in senso più stretto.
    - Creato un possibile mapping tra le informazioni introdotte nel punto 1 e il contenuto di ECLI, visualizzabile in questa tabella.

4. COSTRUZIONE DEL KNOWLEDGE GRAPH

- Mettendo insieme le triple ottenute da ognuno dei documenti si crea il knowledge graph
- Permettere la visualizzazione e/o l'interrogazione utilizzando librerie come Neo4j o pyvis

5

IMPLEMENTAZIONE

IMPLEMENTAZIONE

- Una possibile implementazione della pipeline descritta in precedenza può prevedere l'utilizzo di GPT.
- Necessità di aggiungere un passo iniziale di estrazione del testo e delle sue sezioni principali, date le limitazioni al numero di token.
    - Token = blocco di testo categorizzato, normalmente costituito da caratteri indivisibili, solitamente un vocabolo

1. ESTRAZIONE DEL TESTO

Partendo dai file pdf forniti si estraggono le sezioni principali che li compongono. Questo passo può essere utile in quanto il numero di token che formano i prompt da fornire a GPT è limitato.

- Uso della libreria python PyPDF2 per manipolare i file pdf
- Uso di espressioni regolari per estrapolare le diverse parti del testo

1.1 ANALISI DELLE SEZIONI

Contando il numero di potenziali sezioni in comune con tutti i documenti, tenendo in considerazione le alternative, emerge che le principali sono:

SEZIONE	FREQUENZA
INTRODUCTION / PROCEDURE	84
THE FACTS / AS TO THE FACTS	91
THE LAW / AS TO THE LAW	88
FOR THESE REASONS THE COURT	78

Questa suddivisone potrebbe non essere sufficiente e renderne necessarie ulteriori

Tabella completa

2. ESTRAZIONE DELLA CONOSCENZA

- Uso dell'API di GPT per estrarre le triple (soggetto, predicato, oggetto)
- Un possibile prompt può essere il seguente «*Create a list of relations with 3 columns in this order: source, relation name, target. Return only the the list in the form (source, relation, target). Do lemmatization for source, relation and target. Use this text:*» seguito dal testo da analizzare
- Si potrebbe passare come testo un'intera sezione o parti della stessa, in base al numero di token, e successivamente salvare le risposte ottenute in appositi file

2.1. ESEMPIO DI ESTRAZIONE DELLA CONOSCENZA

È stato inviato a ChatGPT il prompt mostrato in precedenza seguito dalla prima parte della pagina Wikipedia di Matrix. Il risultato ottenuto è il seguente:

Subject	Predicate	Object
Matrix	write	Wachowskis
Matrix	direct	Wachowskis
Matrix	be	installment
Matrix	star	Keanu Reeves
Matrix	star	Laurence Fishburne
Matrix	star	Carrie-Anne Moss
Matrix	depict	future
Matrix	depict	humanity

tabella completa

2.2. COSTI E LIMITI API GPT

- L’API GPT 3.5 Turbo ha un costo di $0,0015 per 1k token di input, $0,002 per 1k token di output
- Inoltre c’è una limitazione di 40000 Tokens Per Minute, 3 Requests Per Minute o 200 Requests Per Day
- Il numero massimo di token di input e di output è di 4097
    - questo rende necessario suddividere il testo in sezioni più piccole

6

APPENDICE: UTILIZZO API GPT

INSTALLAZIONE E AUTENTICAZIONE

- Per prima cosa installare la libreria Python: *pip install openai*
- Per utilizzare l'API è necessaria una chiave per l'autenticazione
    - Specificarla con l'istruzione

        openai.api_key = OPENAI_API_KEY

CHAT COMPLETIONS API

```
response = openai.ChatCompletion.create(
  model="gpt-3.5-turbo",
  messages=[
    {"role": "system", "content": "You are a helpful assistant."},
    {"role": "user", "content": "Who won the world series in 2020?"},
    {"role": "assistant", "content": "The Los Angeles Dodgers won the World Series in 2020."},
    {"role": "user", "content": "Where was it played?"}
  ]
)
```

- **System**: permette di modificare il comportamento dell'assistente
- **User**: fornisce richieste a cui l'assistente deve rispondere

Memorizzare l'alternarsi di questi ruoli e del relativo contenuto permette di mantenere il contesto, a costo di far aumentare il numero di token della richiesta

Parametri opzionali:

- **Temperature**: valore tra 0 e 2. Un valore più alto comporta più casualità; un valore più basso rende più deterministico
- **Top_p**: valore tra 0 e 1. Considera i token con una probabilità maggiore di quella fornita
- **Max_tokens**: numero massimo di token da ritornare

CHAT COMPLETIONS RESPONSE FORMAT


```
{
 "choices": [
  {
   "finish_reason": "stop",
   "index": 0,
   "message": {
    "content": "The 2020 World Series was played in Texas at Globe Life Field in Arlington.",
    "role": "assistant"
   }
  }
 ],
 "created": 1677664795,
 "id": "chatcmpl-7QyqpwdfhqwajicIEznoc6Q47XAyW",
 "model": "gpt-3.5-turbo-0613",
 "object": "chat.completion",
 "usage": {
  "completion_tokens": 17,
  "prompt_tokens": 57,
  "total_tokens": 74
 }
}
```


- **stop**: l'API ha ritornato il messaggio completo
- **length**: output incompleto a causa del parametro max_tokens o token limit
- **function_call**: il modello ha deciso di chiamare una funzione
- **content_filter**: contenuto omesso a causa di un filtro
- **null**: risposta in generazione o incompleta

Per accedere alla risposta generata:
response['choices'][0]['message']['content']