



# UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

Dipartimento di Ingegneria Elettrica e Tecnologie  
dell'Informazione

Anno Accademico 2018/2019

Corso di Intelligenza Artificiale  
Prof. Flora Amato

Strumenti utilizzati:



# Trattamento del Testo

Il Natural Language Processing (NLP) è il processo che effettua il trattamento automatico del testo naturale scritto o parlato tramite un calcolatore elettronico. Esso è finalizzato all'estrazione di una conoscenza implicita, che non sia già stata acquisita, nei documenti di testo che sono per natura documenti non strutturati (Text Mining). Per fare ciò, è necessario fornire al calcolatore le conoscenze linguistiche necessarie a comprendere la struttura e il contenuto del testo. Ambienti che fanno uso dell'NLP sono il web semantico, i sistemi domanda e risposta e i Recommender System (sistemi di raccomandazione che forniscono agli utenti dei suggerimenti in base alle ricerche effettuate in precedenza). Questo processo è reso particolarmente difficile e complesso a causa delle caratteristiche intrinseche di ambiguità del linguaggio umano. La classica ricerca per parola-chiave ha forti limiti in questo senso: problemi riguardano la comparsa dei documenti erroneamente segnalati come rilevanti, oppure l'omissione di documenti rilevanti perché non contengono la parola-chiave. Per questo motivo, la ricerca si è spostata sulla semantica delle parole, in modo da ridurre le ambiguità e permettere di effettuare delle query in linguaggio naturale per concetto. Le funzionalità di gestione semantica sono basate sulla modellazione ontologica e, utilizzano quindi strumenti per la creazione e gestione delle ontologie come OWL. Le fasi principali per un trattamento semantico dei testi sono: pretrattamento, analisi morfosintattica, analisi lessico-metrica, estrazione dei termini peculiari. Il primo passo è quello del parsing (tokenizzazione), cioè individuare le singole parole (i token), ovvero successioni di caratteri dell'alfabeto compresi tra separatori (virgole, punti, ecc...). Per effettuare ciò spesso vengono usati dei software per il trattamento lessico-metrico dei testi, come ad esempio TALAC o GATE.

# GATE

Quello che noi andremo a fare è l'analisi del testo per creare delle annotazioni, o meglio attraverso tutti questi strumenti, analizzando il contenuto dei file testuali e attraverso delle regole che andremo a scrivere con un linguaggio particolare che si chiama JAPE, riusciremo ad effettuare queste annotazioni testuali. Quindi la prima cosa da fare sarà analizzare il testo e trovare le regole.

Lo strumento che noi utilizzeremo è GATE DEVELOPER.

Gate developer : ambiente che fornisce un'interfaccia grafica tramite la quale posso andare a caricare le LR , i documenti che costituiranno il corpus da un lato e da un altro andare a creare le pipeline NLP e quindi andare a comporre le processing resource a seconda delle informazioni che vogliamo estrarre per arrivare ad un'annotazione del nostro corpus. Lo scopo principale di GATE è l'annotazione dei documenti, quindi tutte le sue funzionalità sono relative a questo scopo.

In particolare GATE (General Architecture for Text Engineering) è un sistema open-source sviluppato dall'Università di Sheffield che offre agli utenti una piattaforma completa di Language Processing. È contemporaneamente un'architettura, un Integrated Development Enviroment (IDE), un Framework e una Web App.

Inoltre Gate effettua il Natural Language Processing, ovvero il trattamento del testo scritto nel linguaggio naturale. L'obiettivo di questo strumento è analizzare i file testuali per recuperare le informazioni di interesse.

Attraverso lo strumento di Gate si risale ai pattern testuali, come se fossero delle regole che si ripetono all'interno del testo e individuare elementi di interesse all'interno dei questi pattern.

Gate include un sistema completo di information extractio chiamato **ANNIE**: tramite Annie noi possiamo già lanciare una sorta di processing resource che riesce ad effettuare un'annotazione di una serie di informazioni già preconfigurate. Anche Annie è un' application resource ( è un raggruppamento logico di processing resource, cioè se ho bisogno di combinare una serie di moduli allo stesso modulo, posso creare un'applicazione) e quindi la compongo una volta e la salvo come un'application resource ,così poi la posso richiamare.

Noi abbiamo scelto 11 documenti relativi all'ambito medico, tra quelli proposti dalla professoressa. Utilizzando GATE faremo un'analisi dei documenti ed estrarremo i concetti fondamentali o le categorie per alcune tipologie di elementi.

Per l'analisi abbiamo scelto un corpus formato dai seguenti documenti:

## Language Resource

Le LR in GATE possono essere documenti, corpus (set di documenti), o annotazioni. Un'annotazione è una forma di metadato associato ad una specifica porzione di un documento definita da due "puntatori" che indicano l'inizio e la fine della stringa di riferimento all'interno del contenuto. Ogni annotazione è inoltre caratterizzata da un id numerico, da un tipo e da un insieme di feature che sono delle categorie che ne descrivono informazioni aggiuntive. Le annotazioni associate a parti di un documento vengono rappresentate come parti del documento stesso all'interno di annotation set: l'elaborazione all'interno di un testo fa sì che al termine del testo avremo un insieme di oggetti testuali che vanno a riportare informazioni relative alle annotazioni, quindi avremo una lista di oggetti rappresentati in *XML* che riportano l'id, inizio fine ecc. Così alla fine avrò una lista con un elenco di annotazioni. Un documento ha sempre un annotato set di default, al quale possono esserne aggiunti altri indefinitamente. GATE supporta automaticamente diversi formati di documenti: file di testo, *HTML*, *SGML*, *XML*, *RTF*, alcuni file *PDF*, email, alcuni formati di Microsoft Office e di OpenOffice.

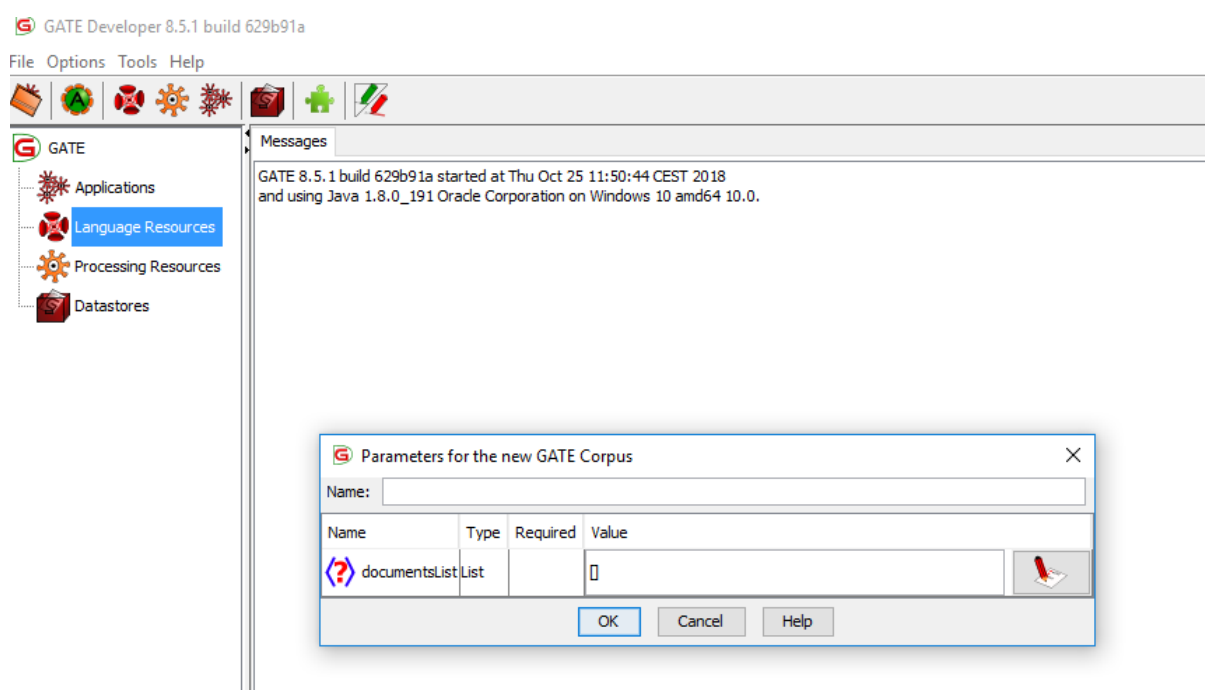
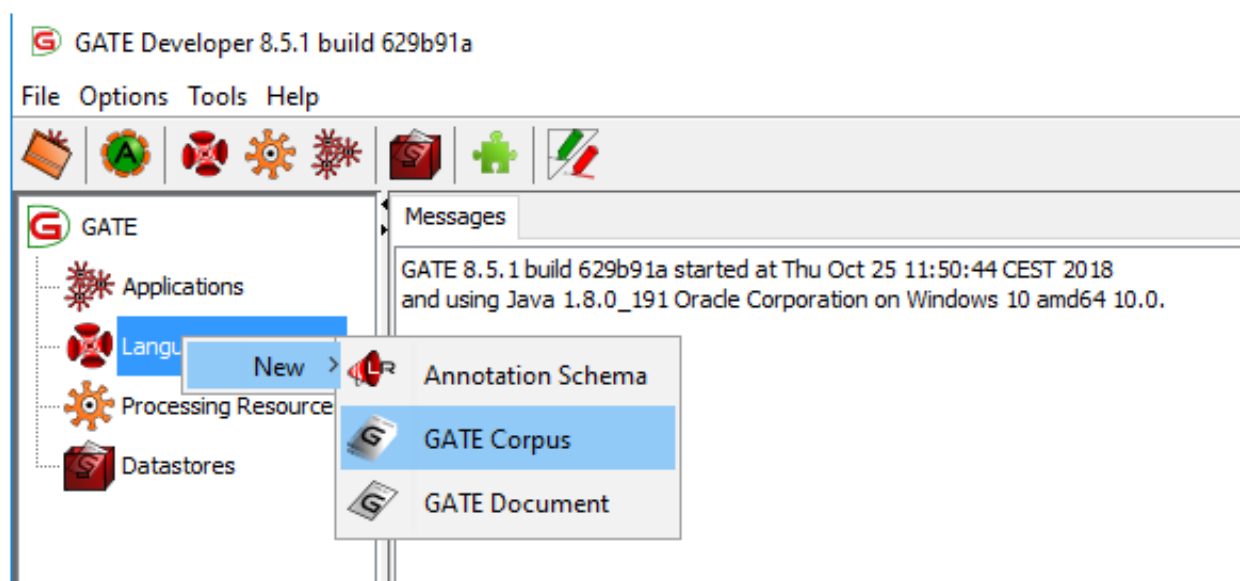
Al momento dell'importazione di un documento, GATE ricerca all'interno del contenuto l'eventuale presenza di tag, che, qualora individuati, saranno rappresentati da annotazioni create ad hoc il cui tipo corrisponderà al nome del tag e le cui features saranno estratte dagli attributi dello stesso. Una volta svolte tutte le operazioni sui documenti, questi ultimi possono essere esportati o salvati in un Data Store, per usi futuri in GATE Developer e altri componenti della suite GATE. Nel nostro caso abbiamo importato i nostri documenti attraverso i seguenti passaggi:

# Creazione del Corpus

Il corpus o corpora è un set di documenti che saranno elaborati in modo uniforme dai vari moduli selezionati dall'utente.

Una volta aggiunti tutti i documenti vado a creare un **corpus** nel quale inserirli:

- **Tasto destro** → **new** → **GATE Corpus**



Successivamente alla sua creazione vado ad inserire in esso tutti i documenti da analizzare:

- **Apro il corpus → clicco su Add → seleziono i documenti**

## Processing Resource

Le PR costituiscono le componenti di elaborazione dei task di NLP eseguibili in GATE. Esse operano sul testo non strutturato e possono essere utilizzate singolarmente, in quanto moduli indipendenti, o essere combinate insieme ad altre PR per formare delle applicazioni.

L'esecuzione di più PR all'interno di un'applicazione può essere di due tipi:

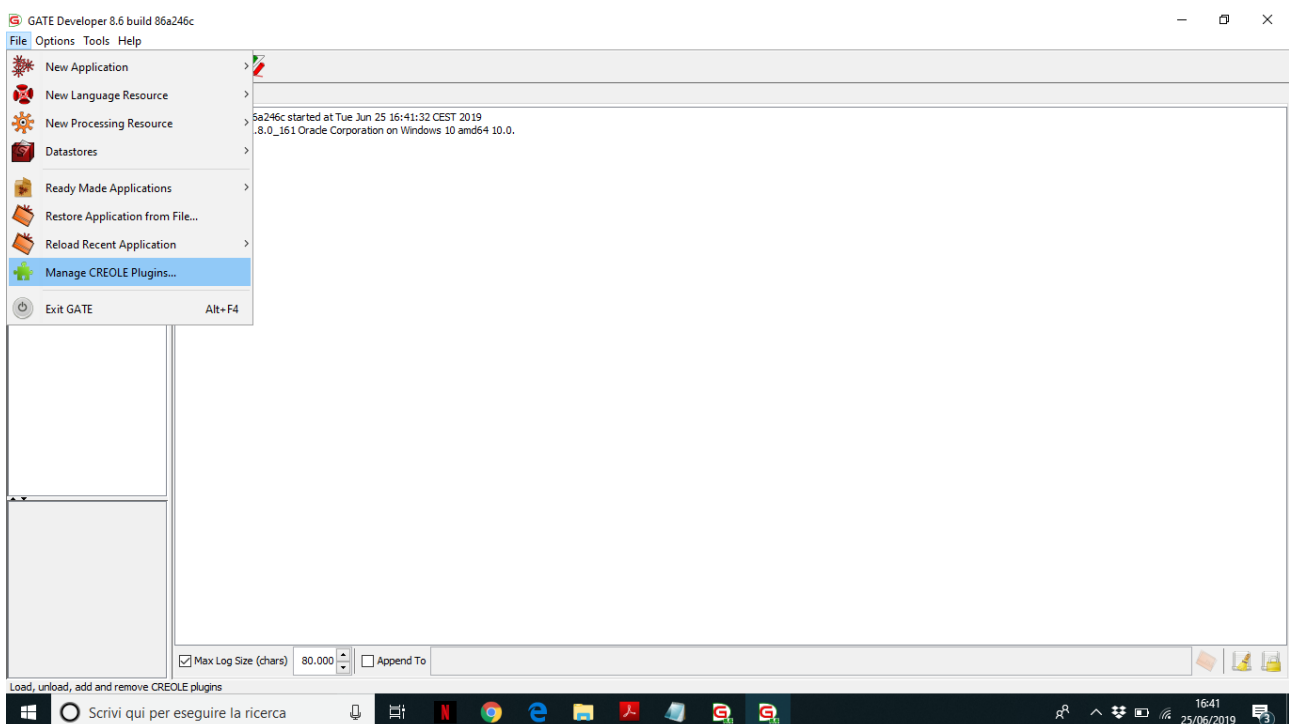
- pipeline semplice, in cui PR all'interno di un insieme ordinato sono eseguite una alla volta;
- pipeline su corpus, specifico per gruppi di PR eseguiti su documenti e corpora, che esegue sequenzialmente tutte le PR che compongono l'applicazione su un documento prima di aprire il successivo all'interno del corpus, sul quale verranno ripetute le stesse operazioni.

## CREOLE Plugins

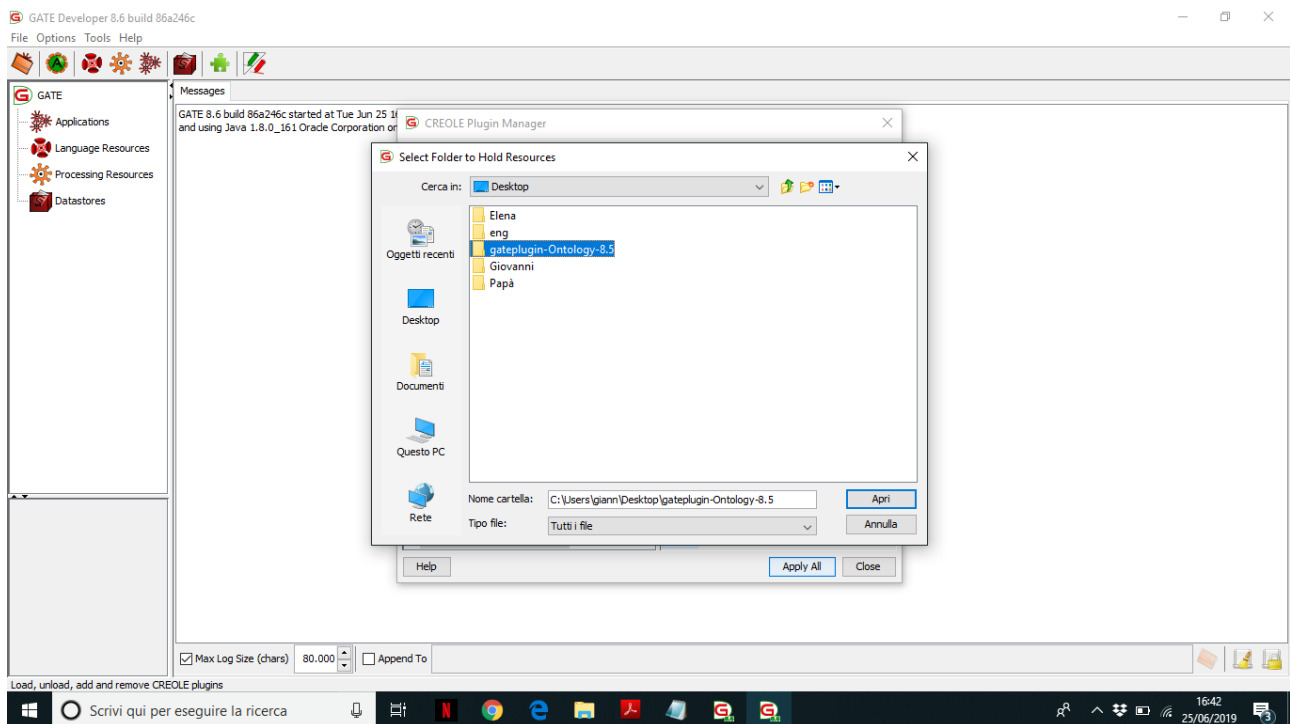
Servono a caricare le processing resources utilizzate per creare e manipolare le annotazioni sui documenti. Sono gestiti tramite la voce 'Manage CREOLE Plugins' dal menu 'File'. Per ciascun plugin vi sono due check box per caricare il plugin 'Load Now' o per aggiungerlo alla lista di plugin che vengono caricati sempre 'Load Always'. Selezionando un plugin, verrà visualizzata la lista di risorse associata e le risorse dei plugin caricati saranno disponibili nelle processing resources (facendo tasto destro → New).

In particolare il plugin Ontology contiene l'attuale implementazione dell'API ontologia. Il plugin Ontology dipende da librerie che non sono disponibili nel repository Maven centrale, quindi il plugin deve essere

scaricato e installato separatamente da GitHub. Disimballando il file zip scaricato creerai una nuova versione di gateplugin-Ontology della directory, e quella directory dovrebbe essere caricata come plugin CREOLE - apri il gestore plugin, fai clic sul pulsante "+" in alto a sinistra, passa a "URL directory" scheda e selezionare la directory del plugin ontology appena decompressa. Questo aggiungerà il plugin all'elenco dei plugin noti e potrai quindi selezionare "carica ora" e / o "carica sempre" come appropriato.



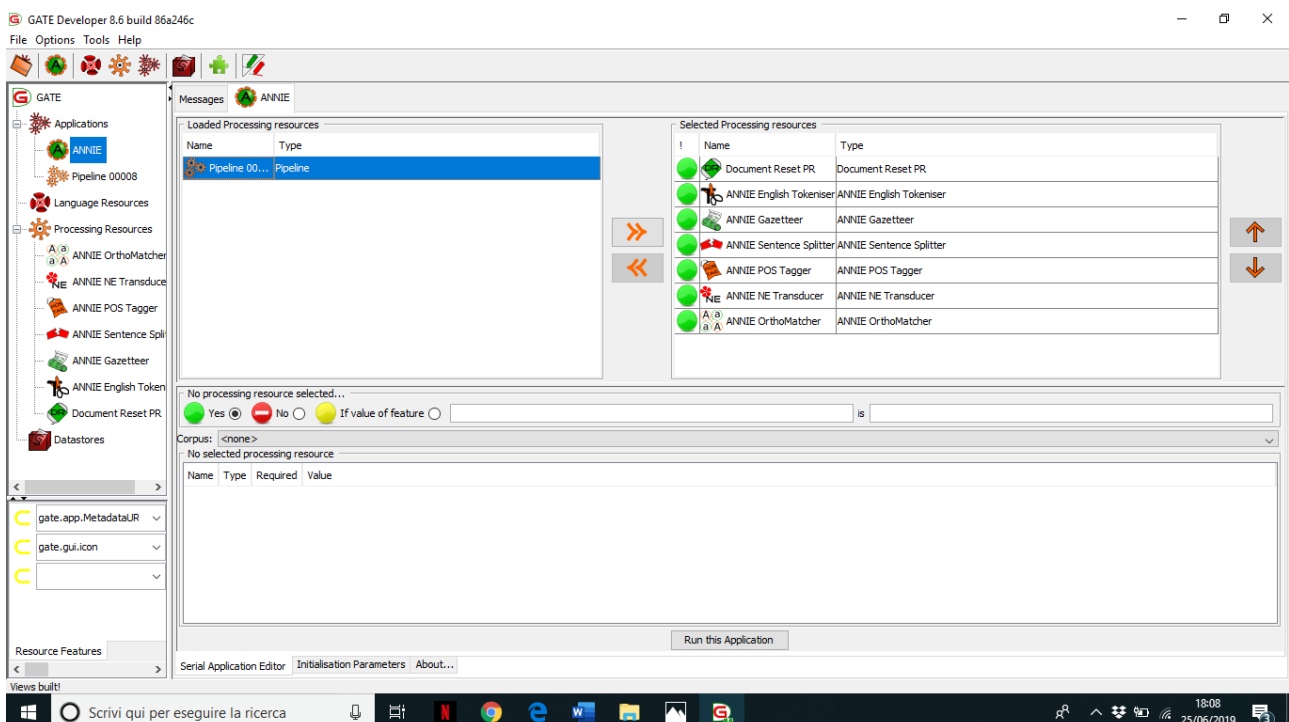
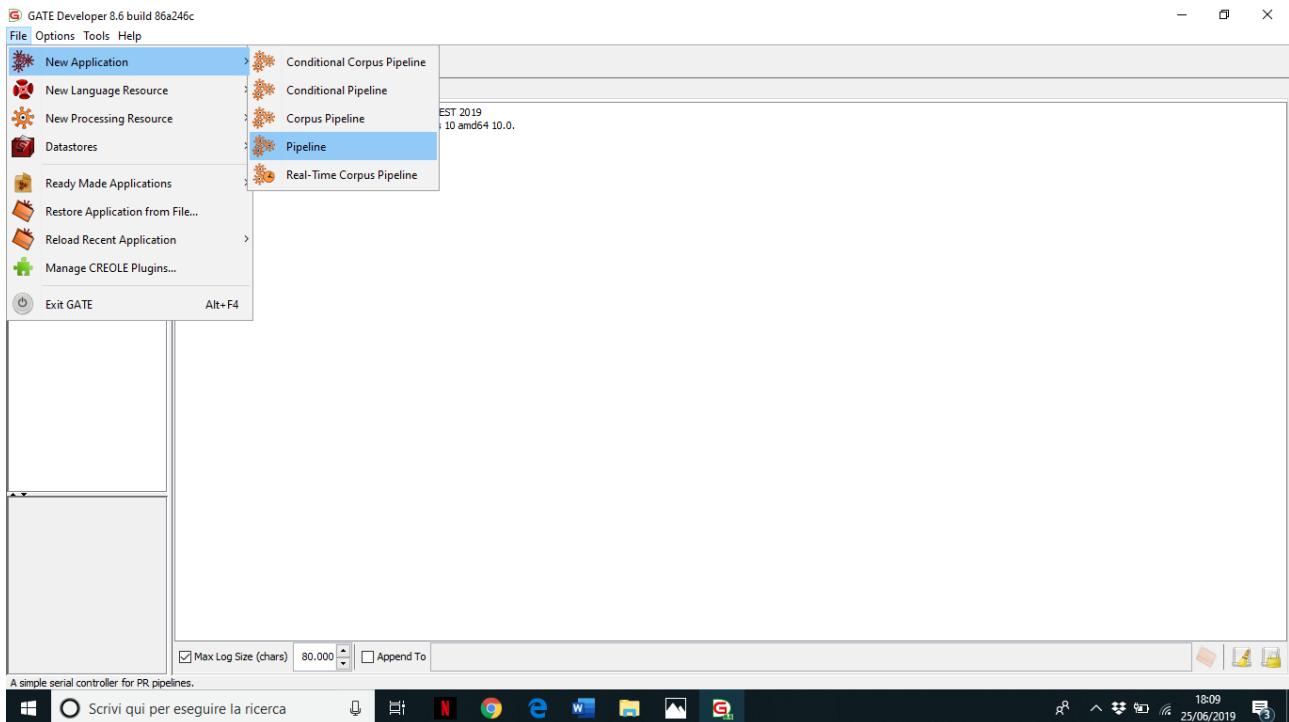
Procedendo selezionando il tipo di Plugin da noi scelto





# Creazione di un'applicazione

Per creare la nostra applicazione seguiamo i seguenti passaggi:




# ANNIE

Oltre a permettere la creazione di proprie applicazioni, GATE mette a disposizione ANNIE.

ANNIE è un sistema composto da una serie di moduli, basati su algoritmi a stati finiti e istruzioni JAPE, che permettono di elaborare annotazioni utilizzabili per vari task di IE.

L'applicazione ANNIE contiene un set di PR core:

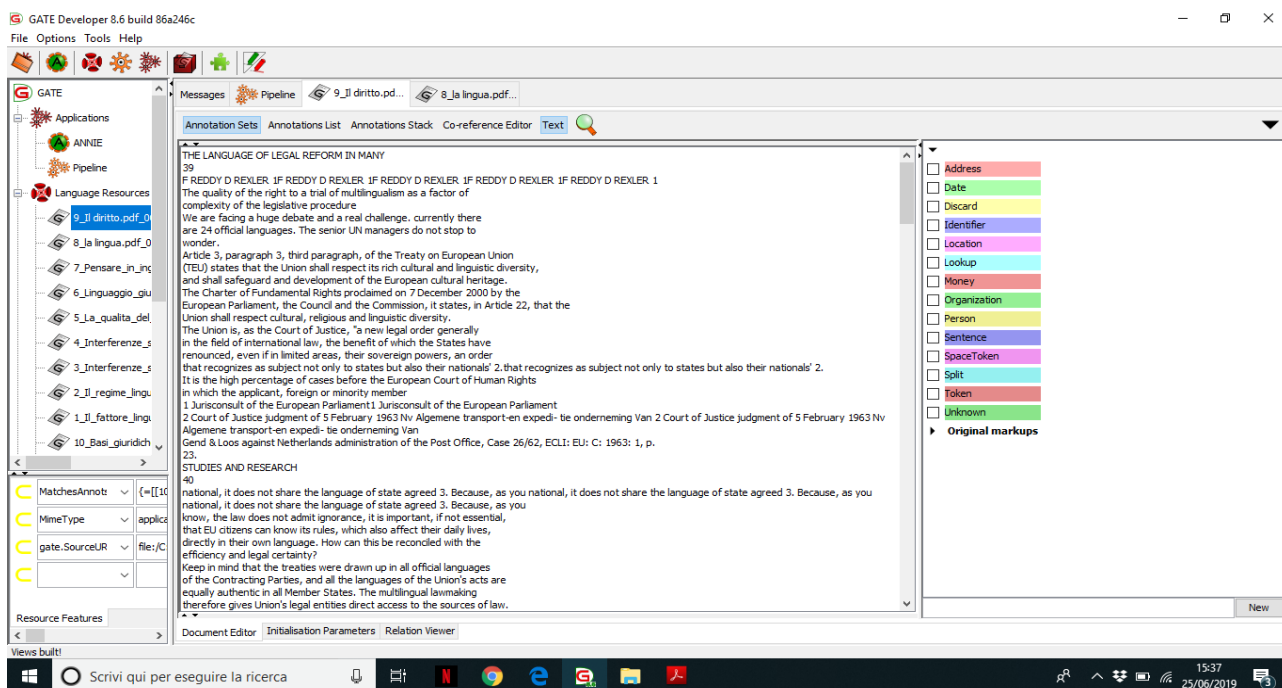
- Tokeniser;
- Sentence Splitter;
- POStagger;
- Gazetteers;
- Named Entity tagger (JAPE transducer);
- Orthomatcher (orthographic coreference);

♦ Cliccando su  verranno caricate tutte Processing Resources associate ad ANNIE e verrà creata un'applicazione che le conterrà nel giusto ordine.

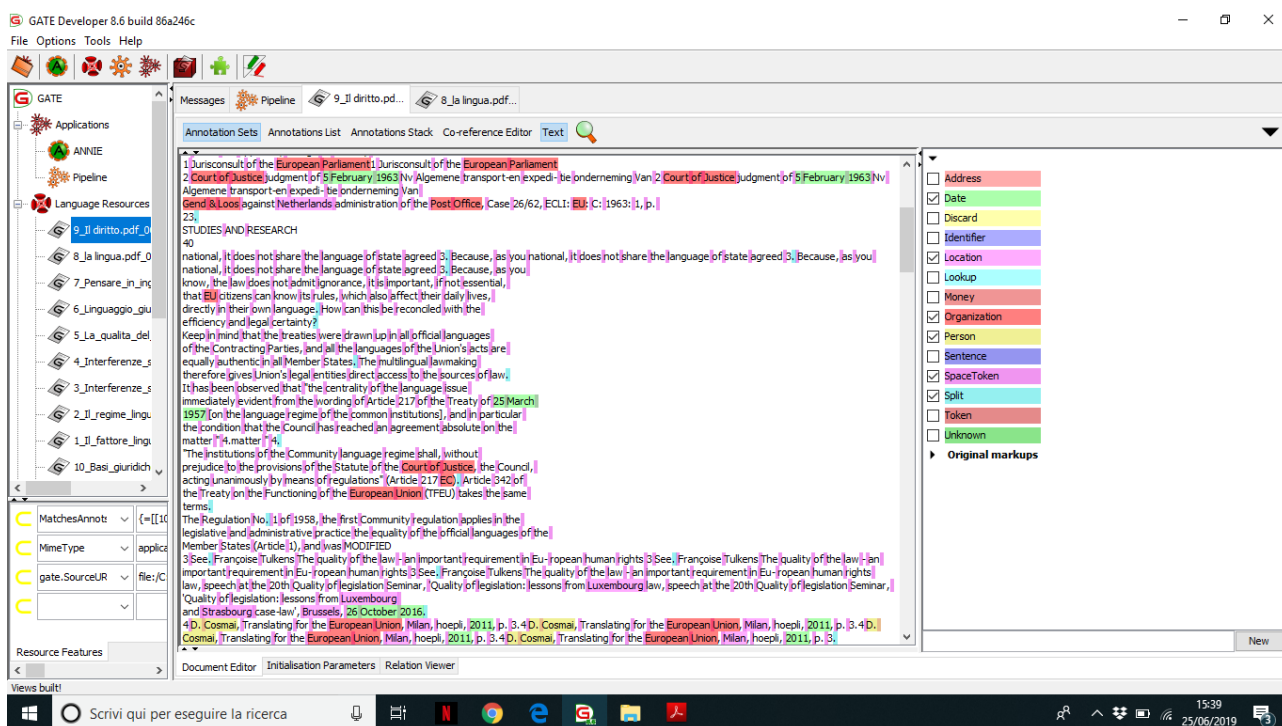
## ANNIE Application

Mandando in esecuzione tale applicazione, viene effettuata l'analisi del corpus. Riportiamo dunque i punti salienti di tale analisi:

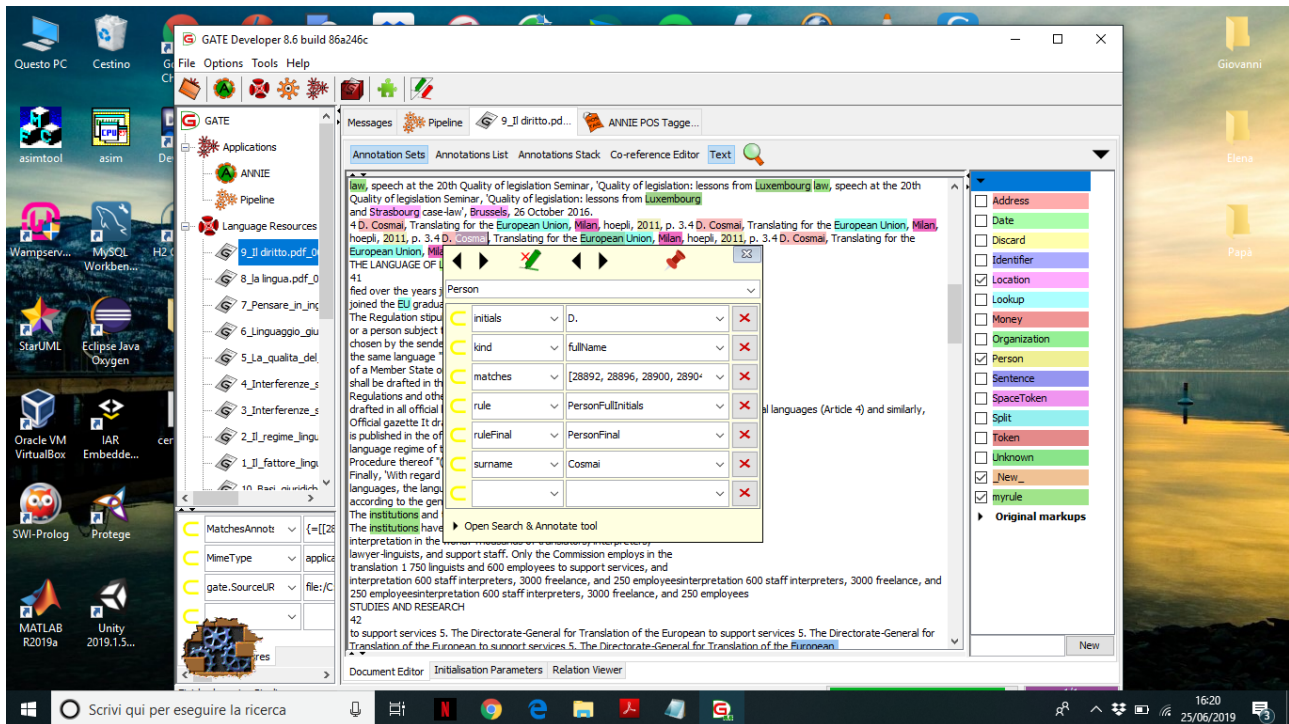
♦ Selezionando il tag **“Annotations Sets”** possiamo vedere l'insieme di annotazioni associate al nostro testo, ad es al testo 'cisti epatiche', permettendoci di effettuare una prima classificazione dei termini.



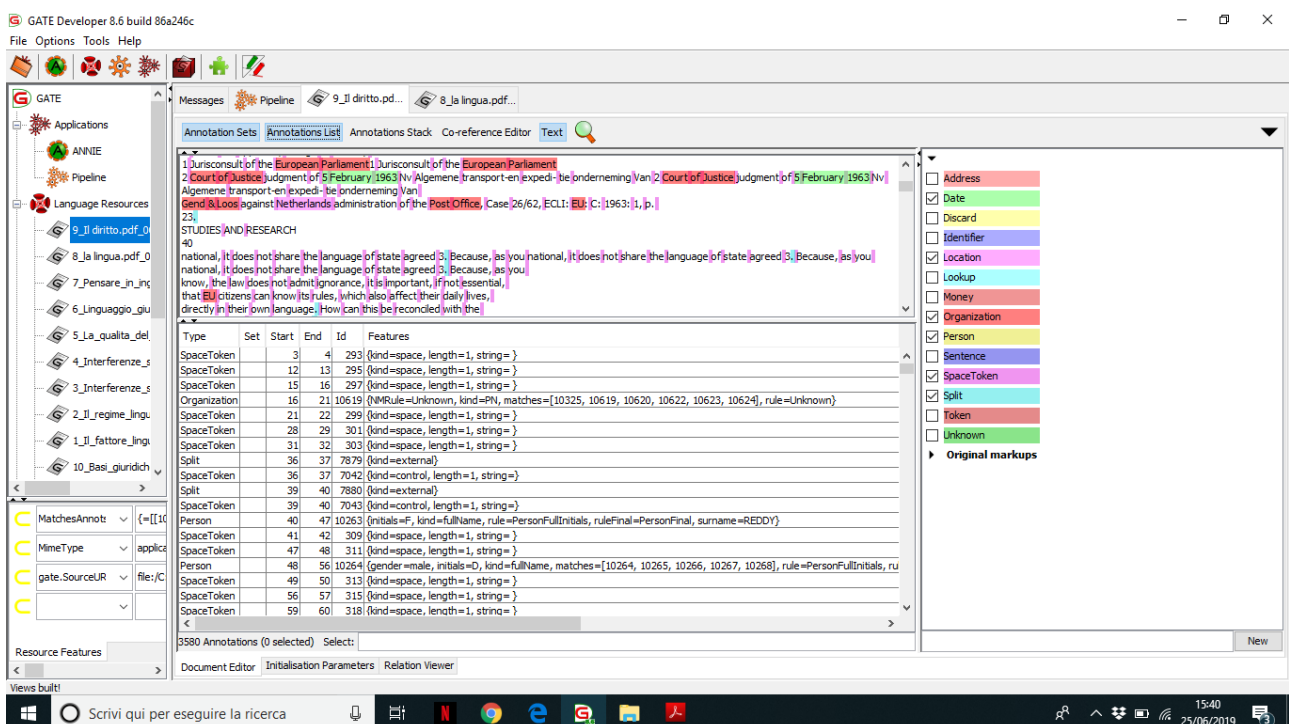
Spuntando le categorie diverse presenti a lato possiamo notare come il programma è in grado di analizzare il testo e riconoscere le parole di ambiti diversi



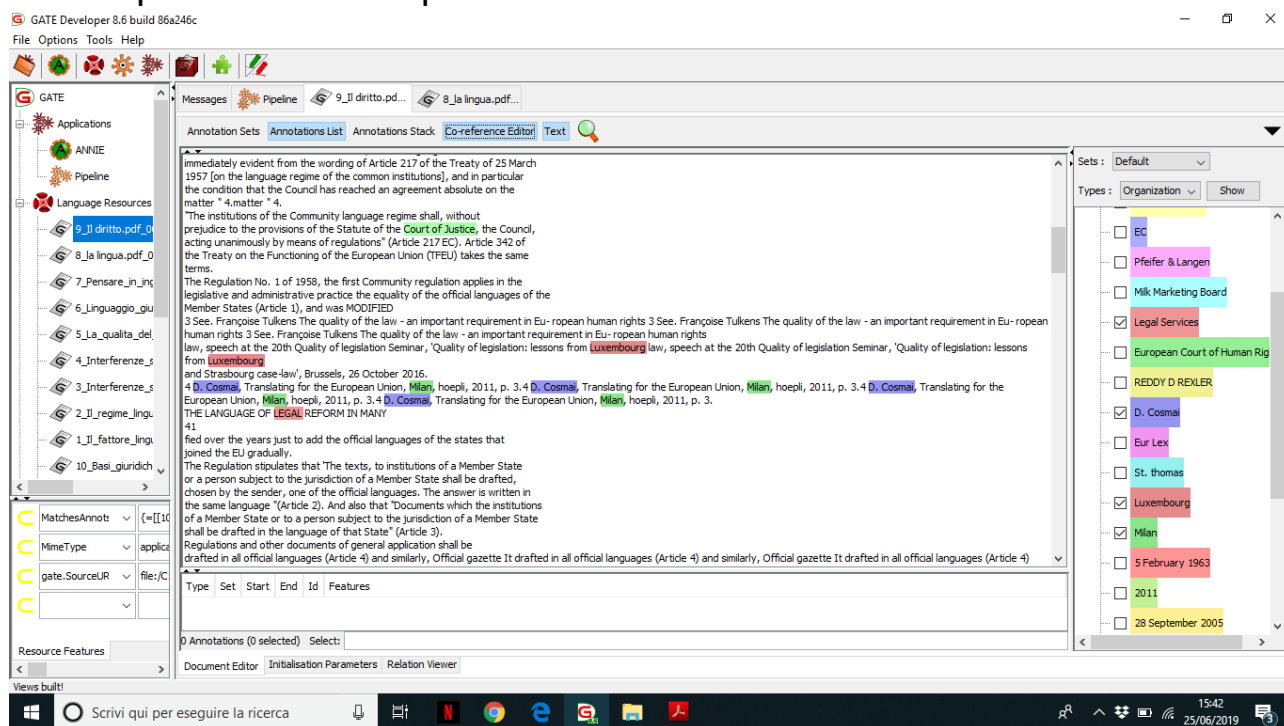
◆ Posizionandoci col cursore su una particolare annotazione, possiamo ricavare ulteriori informazioni sull'elemento selezionato.



◆ Selezionando contemporaneamente anche il tag “Annotations List” possiamo ottenere il tipo, la posizione del testo, ID e Features delle annotazioni selezionate.



◆ Spostandoci infine sul tag **“Co-reference Editor”** possiamo notare come GATE, mediante l’**OrthoMatcher**, riesca a riconoscere lo stesso soggetto anche quando esso è espresso in frasi e modi differenti.

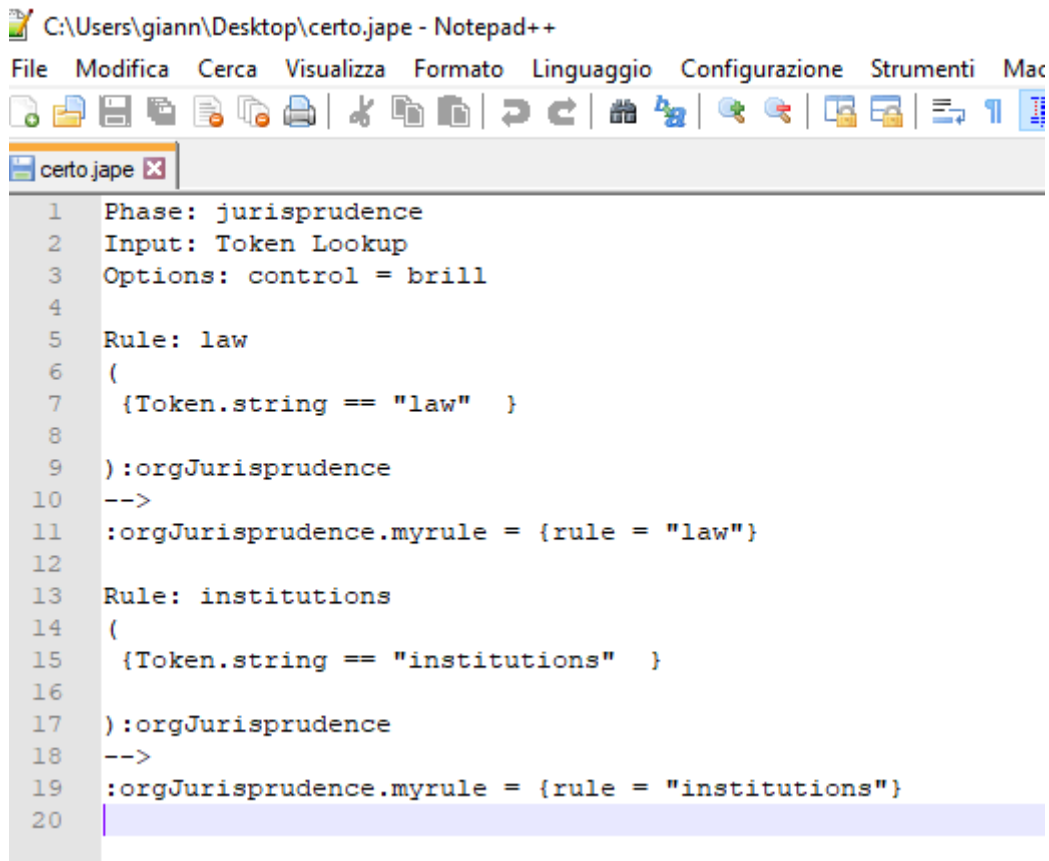


## Regola JAPE

Per effettuare queste elaborazioni testuali, GATE utilizza un set di regole espresse in linguaggio **“JAPE”**. Lo strumento permette, inoltre, agli utenti di creare le proprie regole per creare annotazioni personalizzate. Mostriamo il procedimento per la creazione e applicazione di una regola JAPE.

### Creazione della regola JAPE

Scriviamo ora la regola:



The screenshot shows a Notepad++ window with the title bar 'C:\Users\giann\Desktop\certo.jape - Notepad++'. The menu bar includes 'File', 'Modifica', 'Cerca', 'Visualizza', 'Formato', 'Linguaggio', 'Configurazione', 'Strumenti', and 'Mac'. The toolbar contains various icons for file operations and editing. The active window is 'certo.jape', which contains the following Jape script:

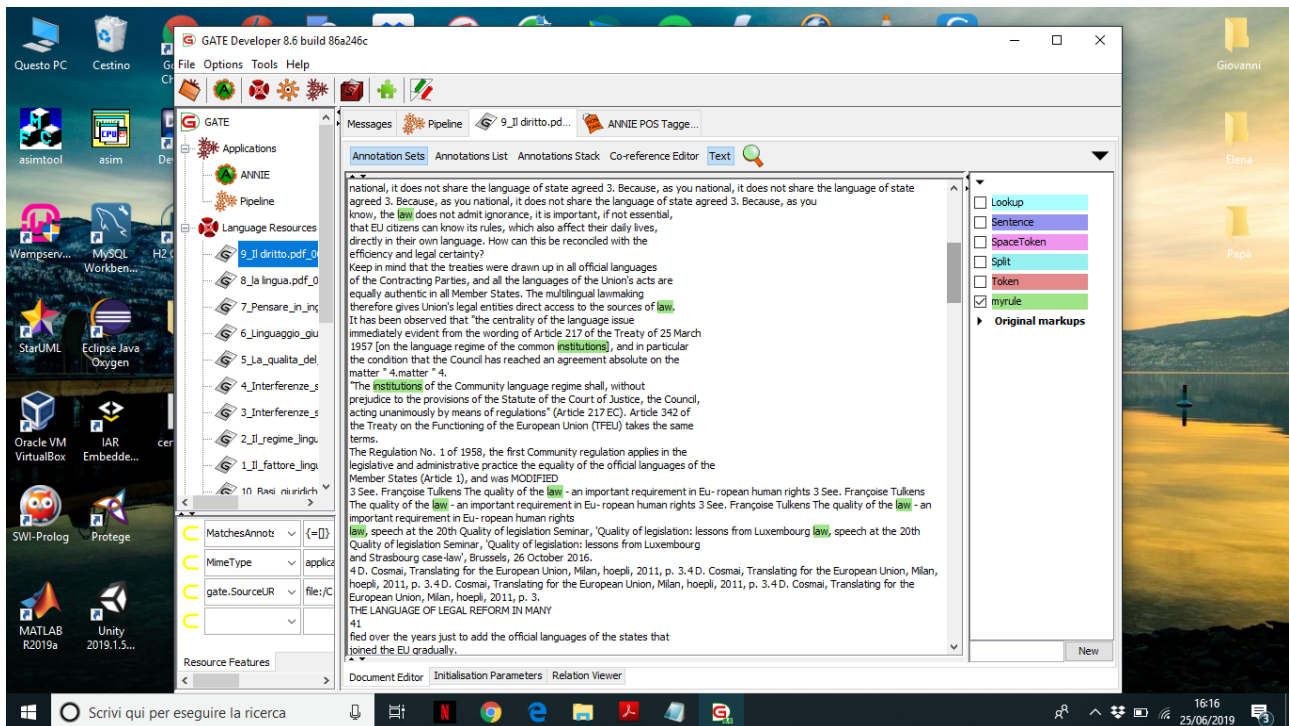
```
1 Phase: jurisprudence
2 Input: Token Lookup
3 Options: control = brill
4
5 Rule: law
6 (
7   {Token.string == "law" }
8
9 ):orgJurisprudence
10 -->
11 :orgJurisprudence.myrule = {rule = "law"}
12
13 Rule: institutions
14 (
15   {Token.string == "institutions" }
16
17 ):orgJurisprudence
18 -->
19 :orgJurisprudence.myrule = {rule = "institutions"}
20
```

## Applicazione regola JAPE

Inseriamo nella nostra applicazione la P.R. “JAPE Transducer” con la regola creata:

Poi bisogna selezionare **‘Run this Application’**.





Notiamo quindi la comparsa del nuovo tag **“myrule”** che va ad evidenziare gli elementi che, appunto, rispettano la regola.

Finita l’analisi, possiamo salvare il risultato delle nostre elaborazioni in un file **.xml**. Tale file contiene le stesse informazioni testuali che GATE ci aveva fornito, ma con una sintassi differente.

Inoltre il nostro file xml ci dà anche informazioni sulle regole utilizzate per individuare l’annotazione.

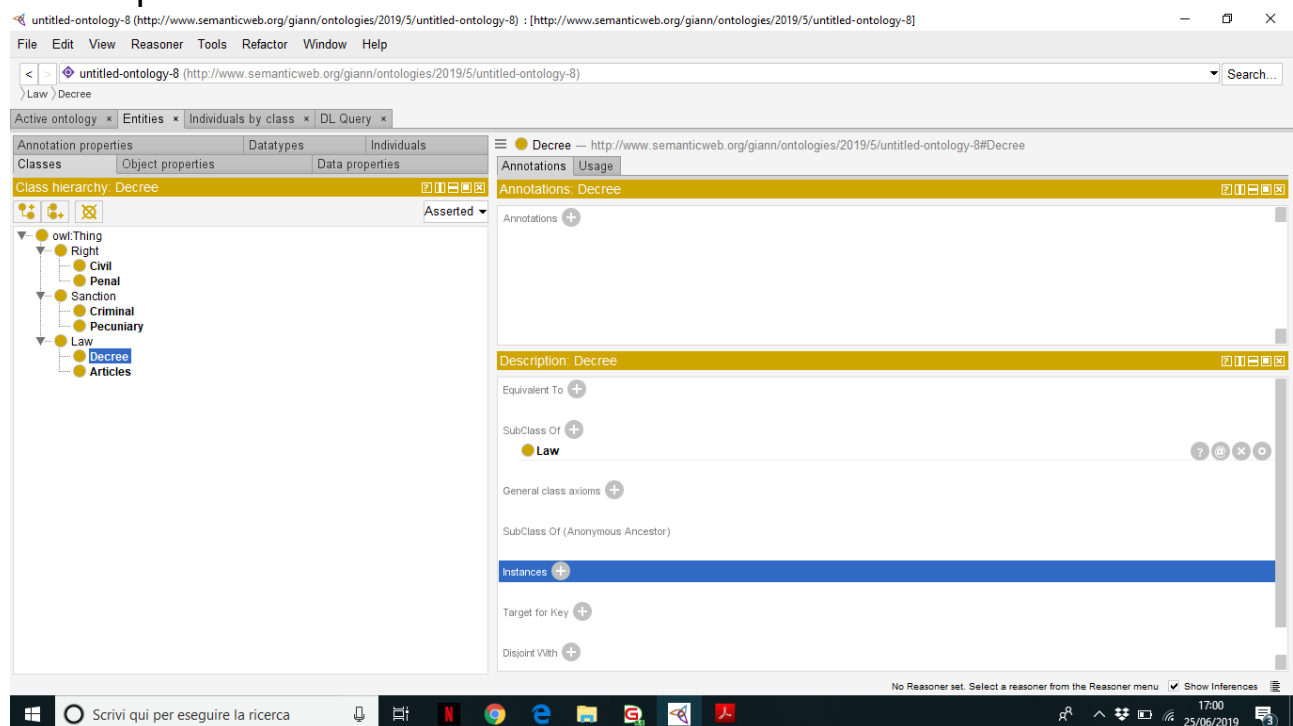
# Protégé

**Protégé è una piattaforma open source che fornisce una suite di strumenti per costruire modelli di dominio e applicazioni basate sulla conoscenza con ontologie.**

**-Il primo step è individuare i concetti rilevanti all'interno del nostro corpus.**

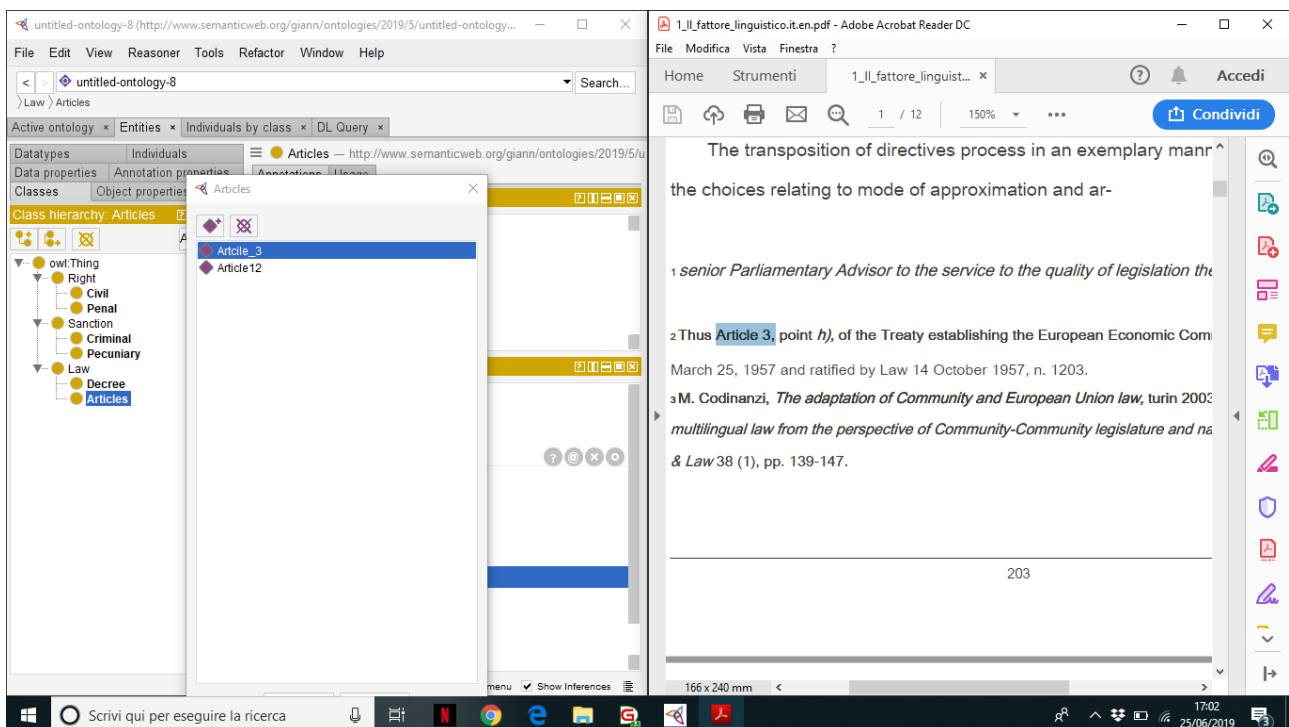
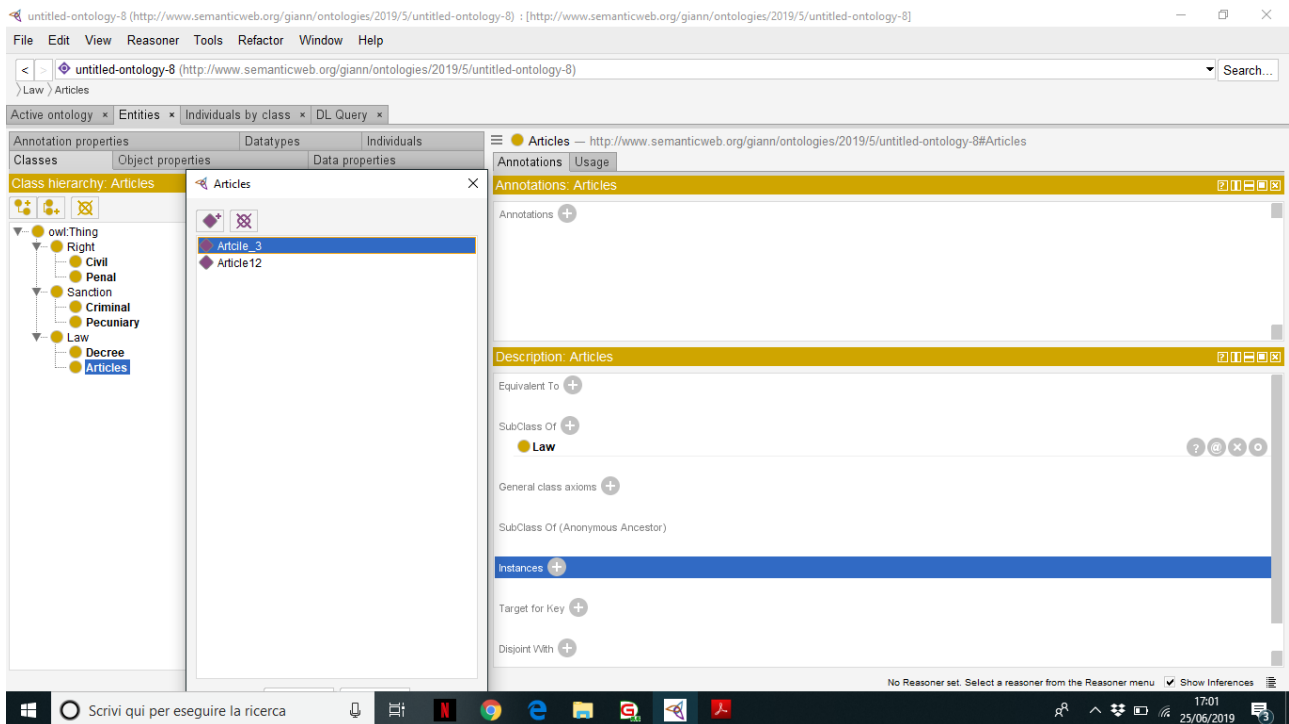
**Con tali concetti andiamo a creare delle classi e sottoclassi che li rappresentano premendo su **

**Inoltre possiamo definire delle relazioni fra le classi:**



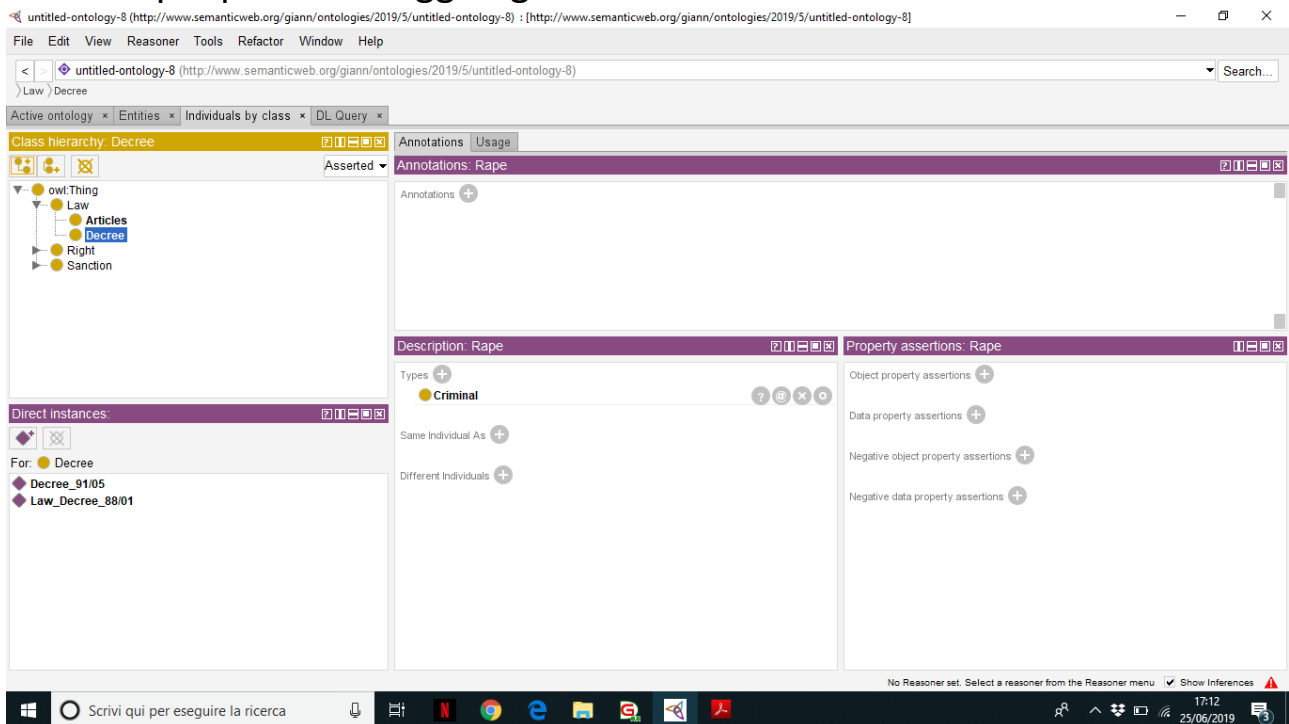
**Popoliamo le nostre classi con delle istanze prelevate dal nostro testo:**



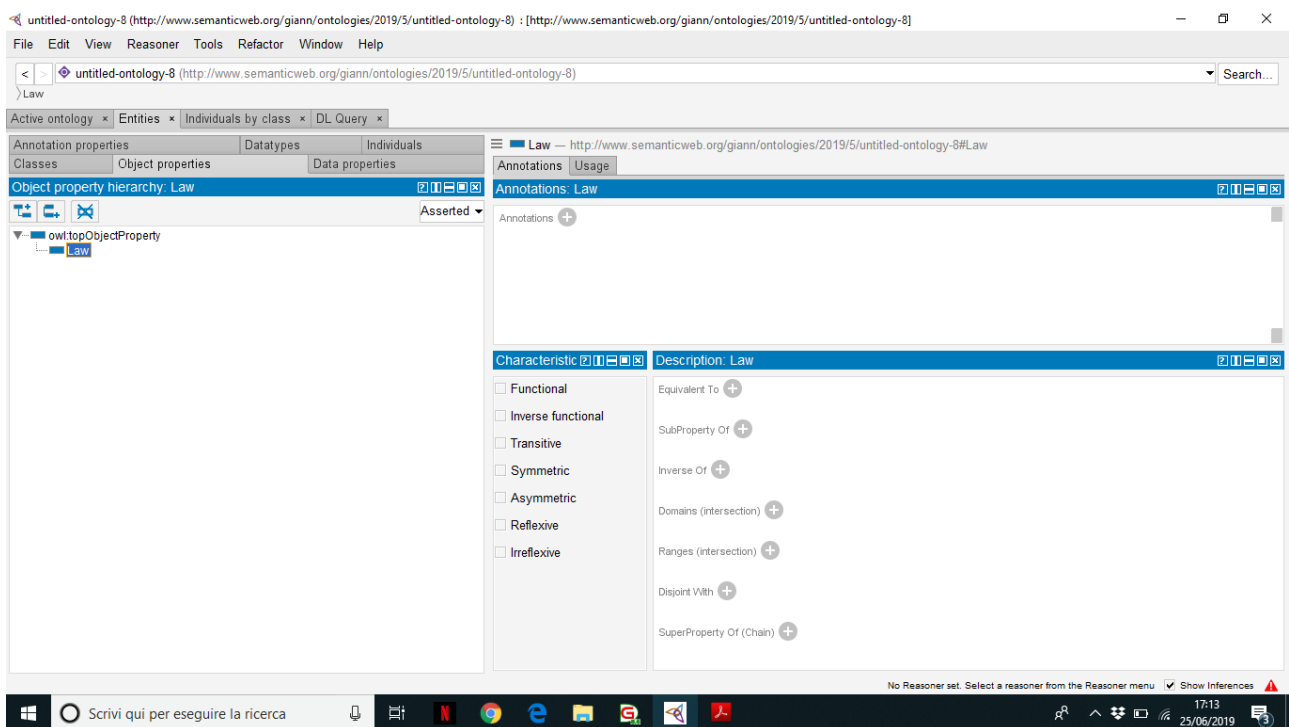


◆ Spostandoci sul tag “Individual by class” possiamo aggiungere informazioni alle istanze da noi create.

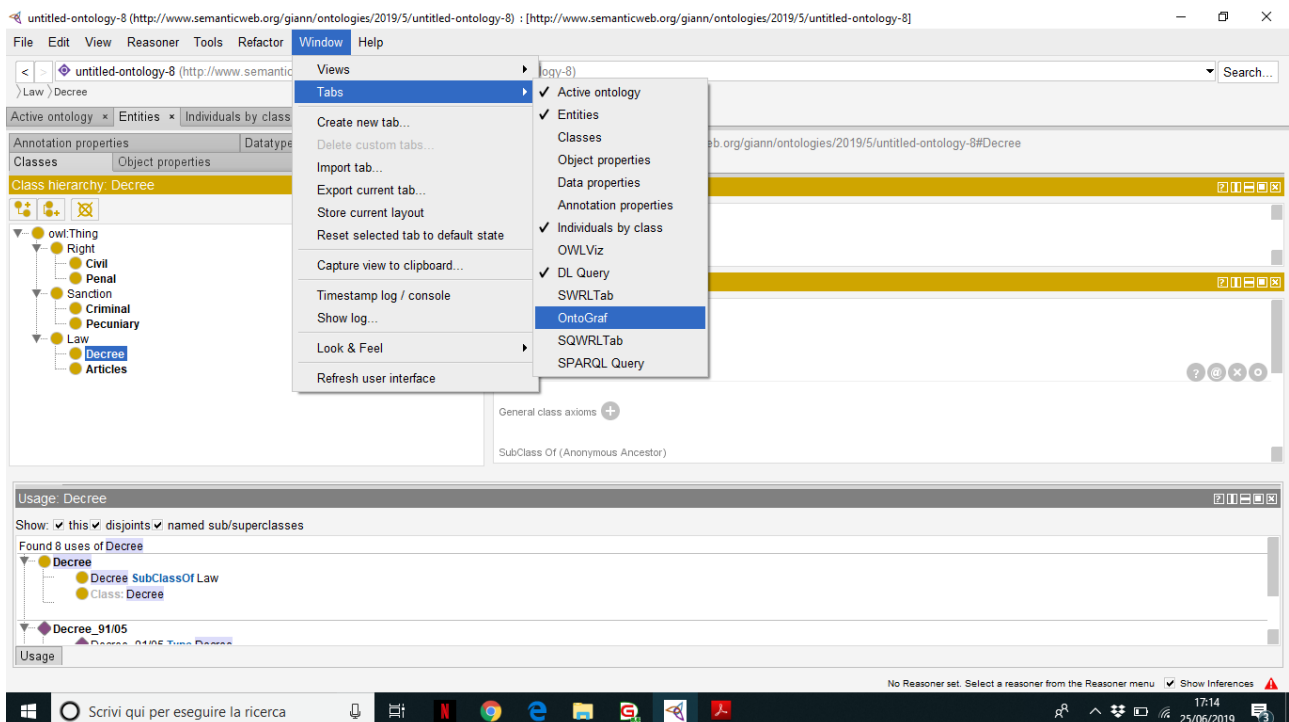
Ad esempio possiamo aggiungere delle annotazioni:



Spostandoci sul tag **“Object properties”** è possibile una vista gerarchia delle proprietà dell'oggetto, che viene presentato come un albero in cui i nodi dell'albero corrispondono alle proprietà dell'oggetto. Un nodo figlio, invece, rappresenta una proprietà dell'oggetto, che è una sottoproprietà della proprietà rappresentata dal nodo genitore. Cioè:



Alla fine del processo di popolamento, spostandoci sul tag “**OntoGraf**”, abbiamo a disposizione una serie di tools grafici che ci permettono di creare il grafo associato alla nostra Ontologia.



The screenshot displays the Protégé ontology editor interface. On the left, the 'Class hierarchy' pane shows a tree structure starting from 'owl:Thing', branching into 'Law', 'Articles', 'Decree', 'Right', 'Civil', 'Penal', 'Sanction', 'Criminal', and 'Pecuniary'. The 'Decree' class is highlighted. The main workspace shows a network graph with nodes representing classes and instances. The 'Decree' class is highlighted in green. The graph shows instances like 'Decree\_91/05', 'Rape', 'Murder', 'Velocity\_Excess', 'Law\_Decree\_88/0', 'Article\_3', 'Article12', and 'Vandalism' connected by relationships. The bottom status bar indicates 'No Reasoner set. Select a reasoner from the Reasoner menu.' and 'Show Inferences' is checked.

STUDENTI:  
OLINO GIOVANNI N46003344  
PACE MARIO N46003486  
RIZZU PIERLUIGI N46003200