UNIVERSITÀ DEGLI STUDI DI SALERNO



DIPARTIMENTO DI INGEGNERIA DELL’INFORMAZIONE

ED ELETTRICA E MATEMATICA APPLICATA

PROJECT RELATION

*Titolo:*

**WordNet**

*Autori:*

Caruso Oscar – Longobardi Francesco – Longobardi Giovanni

Montillo Andrea – Sammarco Enrico – Vaccaro Gennaro – Verdoliva Enrico

[**Scopo del progetto**](#_o4hv6hg2j21i) **3**

[**Prima Funzionalità: Unambiguous Definition**](#_2iyblci2cbu9) **5**

[**Seconda Funzionalità: Relation Finder**](#_m9j4ky2xsrdu) **8**

[**Terza Funzionalità: Contextual Relation Finder**](#_tof9tulfw9ip) **14**

[**Interfaccia Grafica**](#_psf3wqx9ao7r) **18**

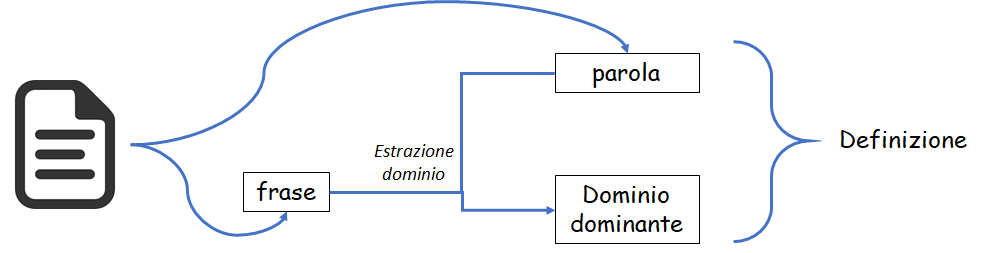
[Link GitHub: <https://github.com/franklinlong/progettoBDSW_Wordnet.git> ]

# Scopo del progetto

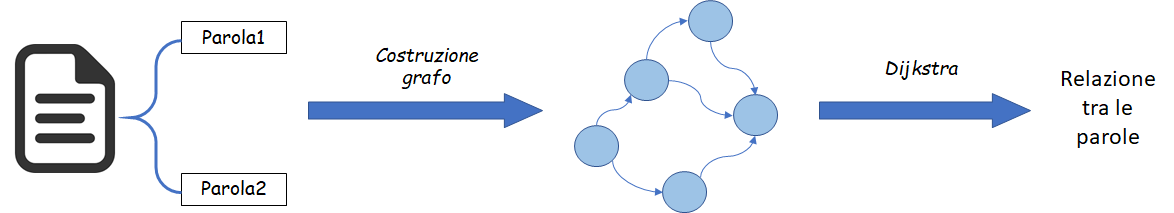
Lo scopo del progetto è la creazione di un reader di testo intelligente che attraverso l’utilizzo di WordNet e WordNet-Domain aiuti l’utente nella lettura e nella comprensione di un testo.

Sono previste tre funzionalità di seguito spiegate brevemente con l’aiuto di uno schema ad alto livello di astrazione:

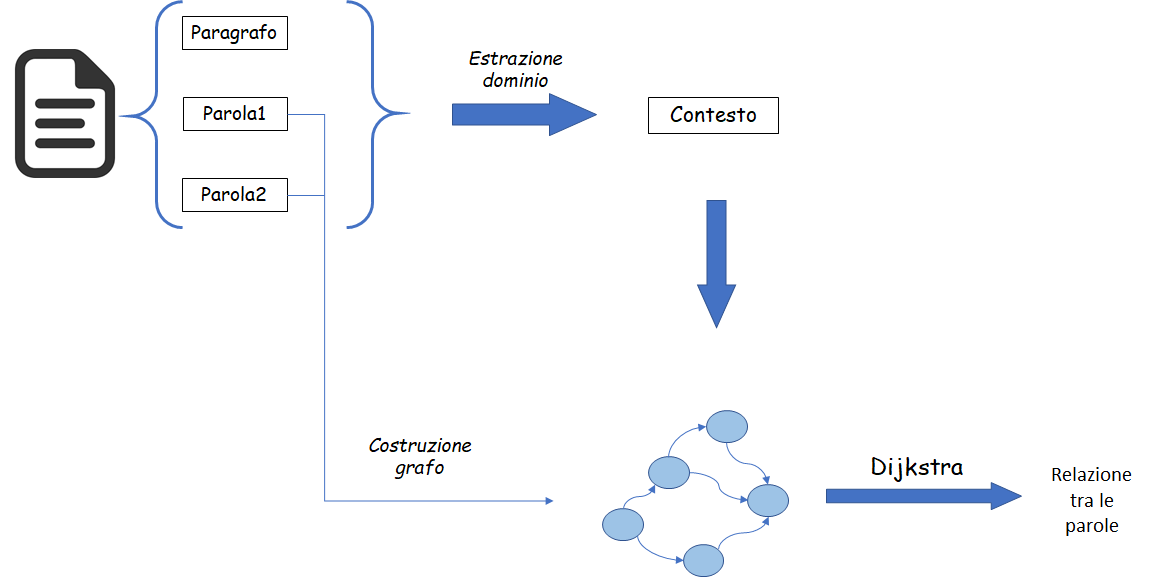
1. **Unambiguous Definition**: permette di ottenere il significato di una parola del testo, in coerenza con il contesto della frase in cui si trova la parola stessa.

*.*

1. **Relation finder**: permette di ottenere una possibile relazione tra due parole del testo. In particolare, è possibile trovare tutte le relazioni tra le due parole, oppure solo la relazione più semplice (il concetto di semplicità verrà trattato in seguito).



1. **Context relation finder**: permette di ottenere la relazione più semplice tra due parole considerando il contesto del paragrafo nel quale esse si trovano, e quindi il significato specifico delle parole in quel contesto.



L'applicazione è stata implementata, sia sul lato front-end (interfaccia grafica) che sul lato back-end, attraverso il linguaggio di programmazione ad alto livello Python. In particolare per interfacciarsi con il Thesaurus Wordnet sono state utilizzate svariate librerie, le più importanti sono *'ntlk'* e *'spacy'*.

# Prima Funzionalità: Unambiguous Definition

Data una parola, Wordnet è in grado di mostrare tutti i possibili significati per quella parola determinandone anche i vari sinonimi per ogni significato. L’insieme formato dalla parola e dai suoi sinonimi, per ogni significato, è detto *“synset”*. All’interno di una frase però, la parola selezionata dall’utente, che potrà essere di qualsiasi categoria (Sostantivo, Verbo, Aggettivo, Avverbio o Nome Proprio), avrà probabilmente uno specifico significato. Lo scopo della prima funzionalità è proprio quello di trovare e mostrare all’utente questo significato.

Per l’implementazione è stato sfruttato WordNet-Domains con l’apposita libreria “spacy-wordnet”. In particolare, WordNet-Domains è un’estensione di WordNet che aggiunge ad ogni synset almeno un’etichetta indicante il dominio (selezionato da un set di circa duecento etichette secondo una sua gerarchia proposta).

La funzione considera la parola selezionata e la frase di appartenenza. Inizialmente vengono ricavati i domini della parola selezionata, ad esempio considerando la parola ‘hate’ otterremo i domini: [‘person’, ‘psychological\_features’].

Successivamente, scorrendo la frase, parola per parola vengono ricavati i domini e si verifica che quest’ultimi siano presenti nei domini della parola selezionata tenendo conto delle ricorrenze.

Per comprendere meglio il funzionamento, si consideri ad esempio la frase “All dogs hate cats”. Le sue parole presenteranno i seguenti domini:

| **Parola** | **Domini** |
| --- | --- |
| ‘All’ | ['electronics', 'time\_period', 'school', 'card', 'aviation', 'play', 'entomology', 'mechanics', 'golf', 'radio', 'optics', 'buildings', 'philosophy', 'factotum', 'hydraulics', 'industry', 'philology', 'fishing', 'sexuality', 'color', 'enterprise'] |
| ‘dogs’ | ['buildings', 'psychiatry', 'gastronomy', 'rowing', 'fishing', 'betting', 'anatomy', 'skiing', 'skating', 'racing', 'mythology', 'biology', 'animals', 'entomology', 'veterinary', 'tourism', 'diving', 'psychology', 'town\_planning', 'animal\_husbandry', 'food', 'body\_care', 'fashion', 'folklore', 'medicine', 'transport', 'archaeology', 'free\_time', 'swimming', 'sport', 'military', 'acoustics', 'color', 'hunting', 'cycling', 'person', 'chemistry', 'mechanics'] |
| ‘hate’ | ['person', 'psychological\_features'] |
| ‘cats’ | ['biology', 'animals', 'psychiatry', 'food', 'entomology', 'surgery', 'sexuality', 'hunting', 'pharmacy', 'chemistry', 'transport', 'radiology', 'computer\_science', 'optics'] |

Si noti che in ‘dogs’ è presente il dominio ‘person’ che è anche un dominio di ‘hate’, dunque, l’occorrenza di tale dominio viene incrementata di uno.

Dunque, considerando la parola hate, il risultato al termine dell’operazione sarà:

| **Dominio** | **Occorrenza** |
| --- | --- |
| ‘person’ | 2 |
| ‘psychological\_features’ | 1 |

Il dominio ‘dominante’ della parola ‘hate’ sarà, dunque, ‘person’. A partire da questo risultato vengono ricavati tutti i synset della parola ricercata (che ricordiamo essere ‘hate’).

Nello specifico viene effettuata una ricerca intelligente, ovvero conoscendo la categoria grammaticale di appartenenza della parola selezionata (dunque se essa è sostantivo, verbo, aggettivo, avverbio) non consideriamo tutti i synset del dominio ma solo quelli che posseggono quella categoria. Tale ricerca è necessaria per disambiguare parole che possono avere significati di categorie grammaticali differenti. Un ulteriore esempio per chiarire questo concetto può essere la parola ‘play’ che può avere significato di sostantivo (inteso come commedia) o di verbo (inteso come giocare, suonare etc.).

Verrà dunque presentata solo la definizione come verbo o come nome a seconda della categoria della parola.

In caso in cui non esistano synset per la restrizione di dominio e categoria considerata, vengono ricavati due synset:

* Il primo senza la restrizione di categoria grammaticale;
* Il secondo senza la restrizione di dominio;

Tale scelta è stata effettuata per presentare delle definizioni sempre attinenti al contesto in esame.

Ricollegando l’esempio precedente, ovvero alla frase “All dogs hate cats”, la parola selezionata ‘hate’ presenta il dominio ‘person’ e la categoria grammaticale 'v' (verbo). Dunque, Wordnet presenta un unico synset descritto nella forma ‘hate.v.01’, dove la parola ‘hate’ è la prima parola del synset, ‘v’ specifica la categoria, e ‘01’ indica la posizione della parola nel synset proposto da Wordnet.

In caso in cui vi siano due o più domini ‘dominanti’, vengono ricavati tutti i synset della parola chiave rispetto questi domini e li si ordinano in base alle occorrenze. Vi saranno uno o più synset ‘dominanti’:

* In caso ci sia un solo synset ‘dominante’, la scelta progettuale è stata di presentare anche le definizioni dei synset con occorrenza inferiore di uno (al secondo posto), se presenti;
* Se vi sono due o più synset ‘dominanti’ presenteremo tutte le definizioni di tali synset;

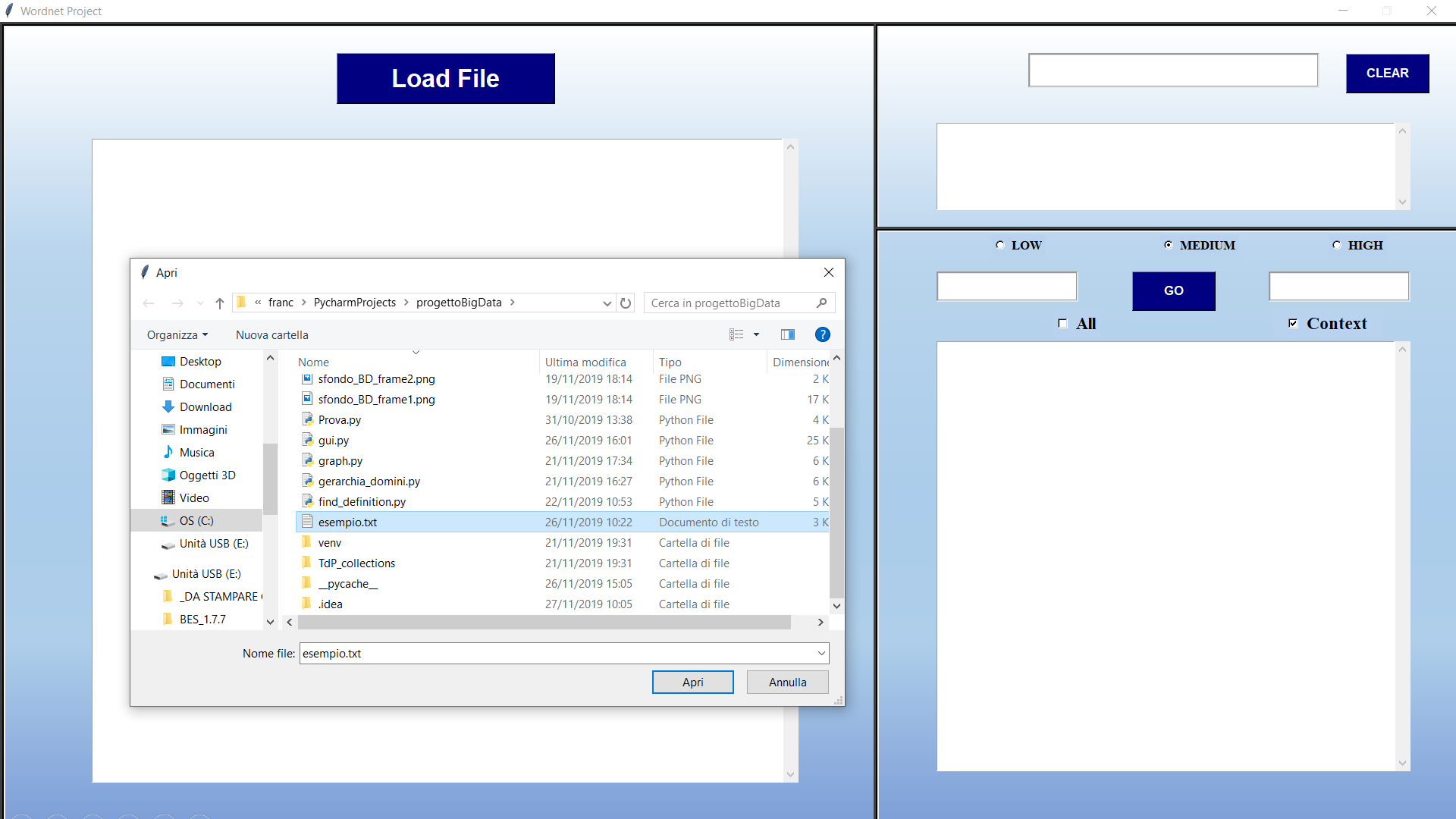
Un caso base, da tenere in considerazione, è quello in cui la parola selezionata non presenta domini. In tal caso, la scelta progettuale è stata:

* Verificare la presenza di synset su WordNet, se presenti viene mostrata la definizione del primo di essi;
* In caso contrario l’utente verrà avvertito che non vi sono definizioni disponibili;

# Seconda Funzionalità: Relation Finder

In riferimento alla funzionalità di relazionare due parole scelte dall’utente, sono state realizzate tre differenti modalità. Mentre le prime due modalità sono spiegate in maniera dettagliata e approfondita all’interno di questo paragrafo, la terza sarà spiegata nel prossimo paragrafo della relazione, essendo concettualmente diversa dalle prime due. Come riportato nella figura seguente, sull’interfaccia grafica è possibile vedere due checkbox: una etichettata con “All”, l’altra con “Context”. In particolare:

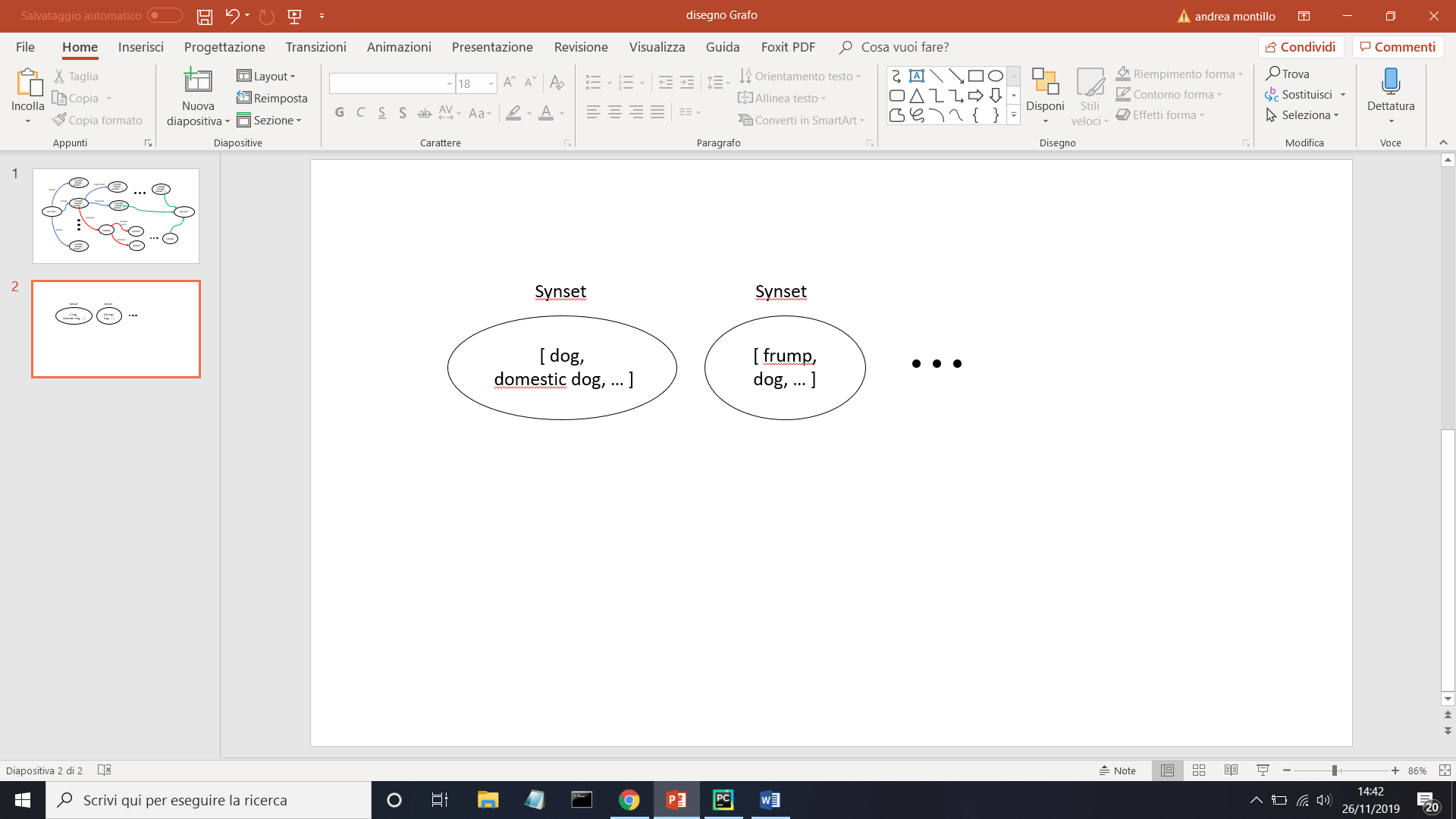
* Non selezionando nessuna delle checkbox, verrà eseguita la prima modalità che restituisce solamente la relazione più forte tra le due parole.
* Selezionando la checkbox “All” verrà eseguita la seconda modalità che restituisce tutte le relazioni trovate tra le due parole.
* Selezionando la checkbox “Context” verrà eseguita la terza modalità che restituisce la relazione tra le due parole appartenenti solo se appartenenti allo stesso paragrafo contestualizzandola rispetto al paragrafo.



*Checkbox dell’interfaccia grafica che permettono la scelta della modalità.*

Lo scopo è quello di trovare una relazione tra due parole scelte dall’utente senza tener conto del contesto in cui le parole si trovano. Essendo Wordnet un insieme di migliaia di synset, abbiamo utilizzato la teoria dei grafi per modellare il nostro problema considerando i synset come vertici e le relazioni tra essi come archi. In questo modo andiamo a creare un grafo connesso che permette l’individuazione di tutti i possibili percorsi tra le parole scelte dall’utente e anche di selezionare quella a percorso minimo, ovvero la relazione che risulta essere più semplice, attraverso l’algoritmo di Dijkstra.

In particolare, un dettaglio implementativo non trascurabile è che wordnet considera le relazioni lessicali (‘*contrario di*‘ e ‘*derivato da*‘) dei collegamenti tra lemma, ovvero i singoli membri di un synset. La scelta progettuale di utilizzare come struttura dati un grafo complesso costituito sia da Synset che da lemma non è casuale, ma dovuta alla necessità di includere all’interno del grafo anche questi ultimi, essendo molto importanti per la realizzazione di una relazione. Infatti, qualora si escludesse la presenza di lemma all’interno del grafo, tantissime relazioni (anche molto evidenti) non verrebbero restituite poiché le relazioni lessicale verrebbero ignorate.



***Esempio****: due synset relativi alla parola ‘dog’ che si differenziano per il significato. Dall’immagine è possibile notare i differenti lemma che compongono i due synset.*

Riguardo la realizzazione del grafo, essa si suddivide in diversi step:

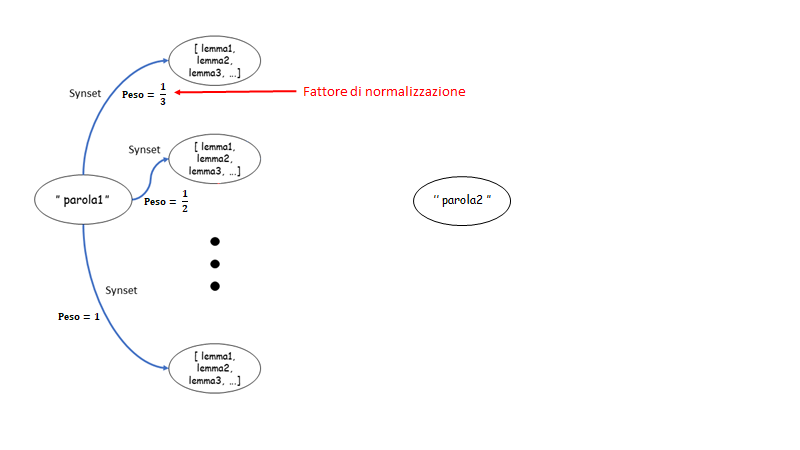
1. Si aggiungono i primi due nodi corrispondenti alle parole scelte dall’utente.



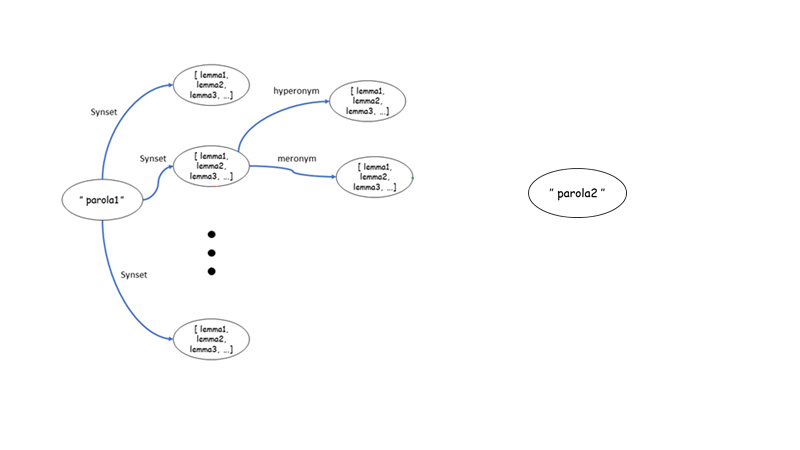
1. Il nodo contenente la prima parola scelta dall’utente viene collegato a tanti nodi quanti sono i synset della parola restituiti da Wordnet.
   * gli archi saranno etichettati con una stringa che indica la natura del collegamento. In particolare, sinonimia per il primo step dell’algoritmo (poichè wordnet restituisce tutti i sinonimi di una parola).
   * il peso di ogni arco è un semplice valore dipendente dalla frequenza d’uso della parola rispetto al synset del nuovo nodo. La frequenza d’uso di un synset è un valore definito da Wordnet che rappresenta l’importanza del particolare significato per la parola a cui il synset fa riferimento. Per questo motivo, più questo valore sarà alto, più l’arco avrà un impatto maggiore nella scelta del percorso minimo. A causa della poca omogeneità di questo valore nei diversi synset associati ad una parola (può accadere che il valore associato al secondo synset sia di due o tre ordini di grandezza inferiore rispetto a quello associato al primo synset), il peso associato all’arco è l’inverso della radice cubica della frequenza d’uso. Tale funzione è stata scelta empiricamente, dimostrando tramite esperimenti di essere quella che più uniforma gli archi del grafo, permettendo così ad ogni synset, anche quelli meno comuni, di essere presi in considerazione per la realizzazione di una relazione.

(e.g. frequenza d’uso = 27 → peso arco = )

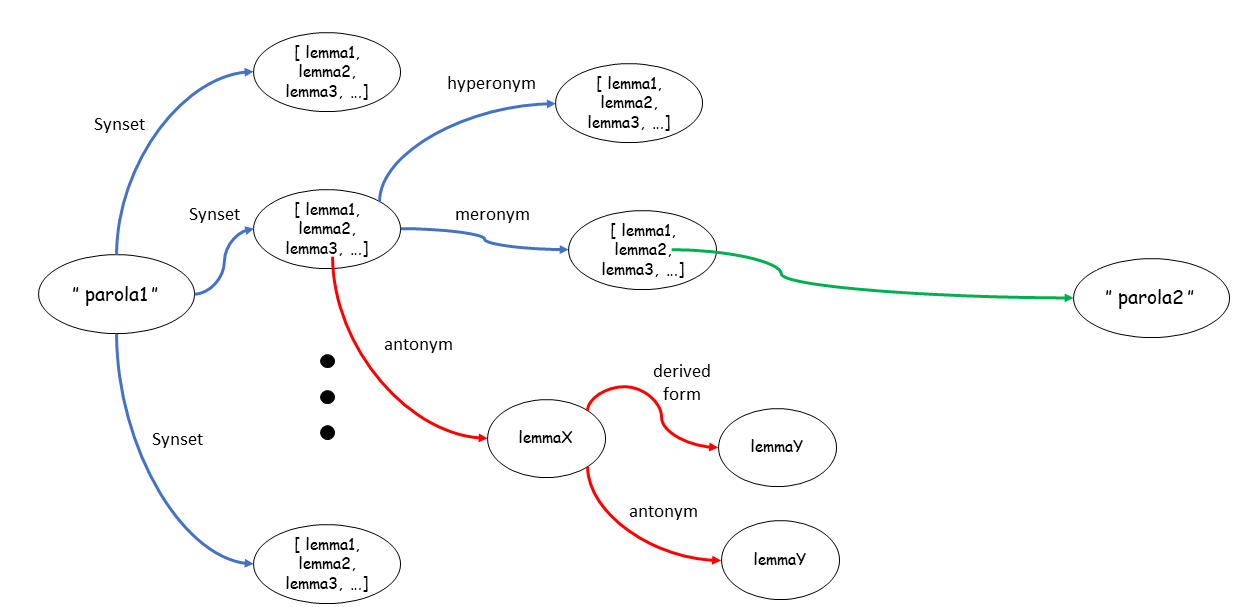
* + viene memorizzato il peso minore tra quelli calcolati precedentemente, e quindi il peso dell’arco che collega la parola iniziale con il synset dal significato più frequente per quella parola. Questo valore viene nominato ‘*fattore di normalizzazione*’, che verrà utilizzato per il calcolo dei pesi degli archi.



1. Viene a questo punto avviato un algoritmo iterativo che per ogni nodo del grafo ricerca altri synset da inserire nel grafo, che si collegano al nodo attraverso una delle relazioni standard di Wordnet (hyperonym, meronym, also see, ...)
   * gli archi saranno etichettati con una stringa che indica la natura del collegamento.



* + Il peso degli archi corrisponde al prodotto tra il fattore di normalizzazione e un valore dipendente dalla natura del collegamento. (e.g. valore\_meronimia = ; fatt\_di\_norm = → peso arco = )
  + si controllano, per ogni lemma dei synset aggiunti, le relazioni lessicali che ci permettono di individuare nuovi lemma da inserire nel grafo. Se un lemma corrisponde alla seconda parola scelta dall’utente, viene creato il collegamento con essa (senza etichetta e con peso 0.01).



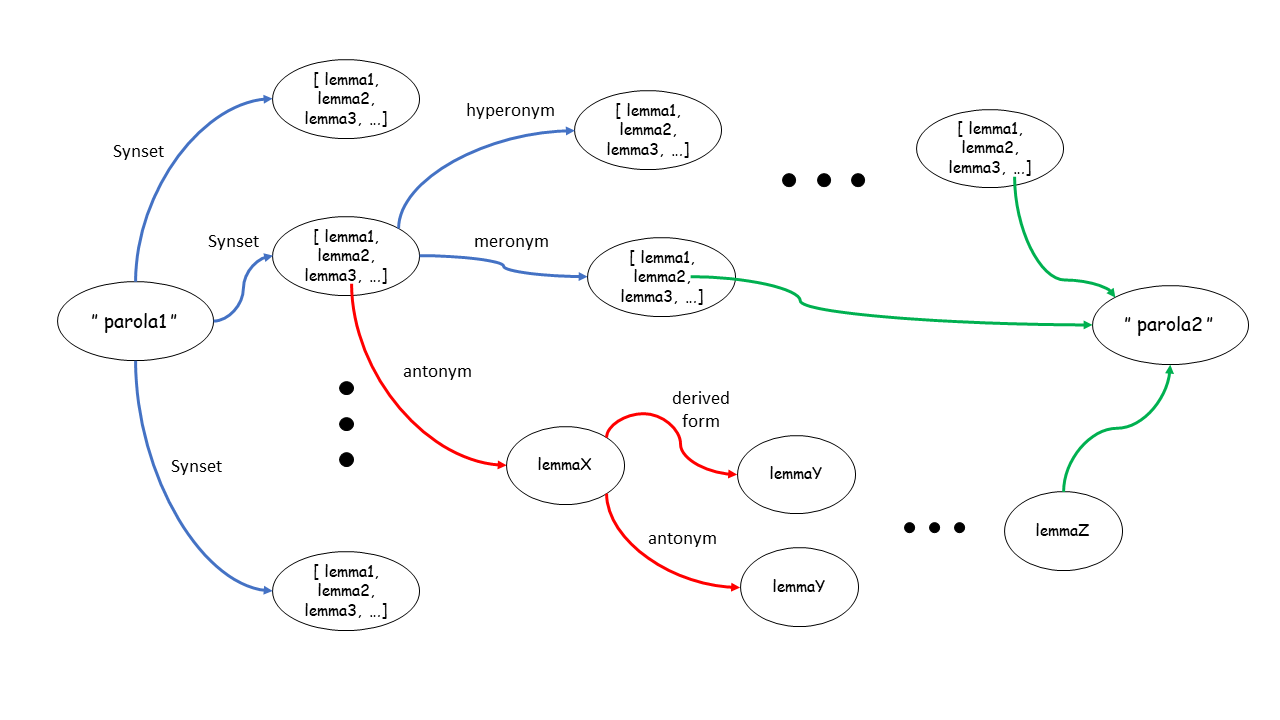
1. L’algoritmo termina dopo che sono state effettuate un numero di iterazioni scelto dall’utente attraverso l’interfaccia grafica.

Si noti come il grafo sia formato da nodi di diverso tipo, ovvero i primi due nodi inseriti sono stringhe, poi vengono aggiunti nodi che corrispondono a dei Synset e altri che corrispondono a dei Lemma. Questo perché, a livello implementativo, i nodi del grafo hanno un attributo *‘element’* che non è vincolato ad un tipo unico.

Si noti inoltre che il calcolo dei pesi degli archi è finalizzato a dare più importanza (un peso totale minore) ai percorsi che comprendono i significati più comunemente utilizzati per la parola iniziale e ai percorsi con relazioni ritenute più forti, importanti.

(e.g. meronimia è ritenuto più importante di iperonimia →

→ valore\_meronimia < valore\_iperonimia )



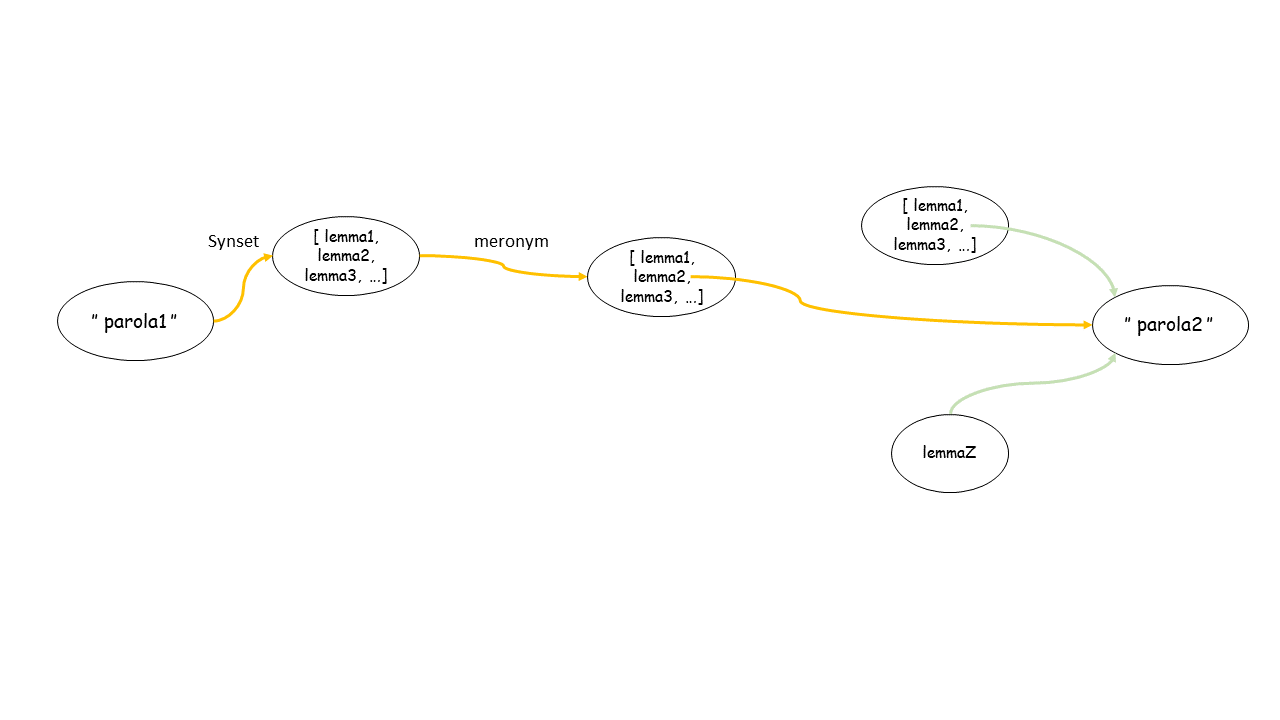
*Schema logico finale del grafo in cui si evidenziano tutti i suoi componenti.*

*Si osservi come il grafo sia formato da vertici di diversi tipo (stringhe, synset e lemma).*

*Sono mostrate: in blu le relazioni tra synset, in rosso quelle tra lemma e in verde quelle con la seconda parola scelta dall’utente.*

A questo punto, il grafo è stato creato e a seconda della modalità selezionata, ovvero se si vuole ottenere solo la relazione più semplice oppure tutte le relazioni tra le parole scelte, abbiamo:

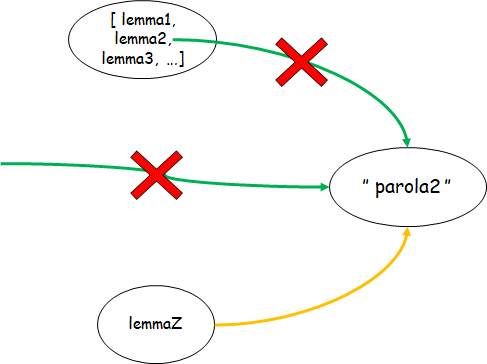
* calcolo del percorso minimo attraverso l’algoritmo di Dijkstra.



*In giallo il percorso minimo del grafo precedentemente mostrato.*

* calcolo di più percorsi tra le due parole utilizzando tante volte Dijkstra quanti sono gli archi che permettono di raggiungere la parola di destinazione.

In particolare, ad ogni iterazione, viene utilizzato l’algoritmo di Dijkstra, come nella prima modalità, per ottenere la relazione più semplice tra le parole. Quindi l’arco con la quale viene collegata la parola finale viene cancellato, così da ottenere nell’iterazione successiva un nuovo percorso minimo diverso dal precedente (almeno per l’arco finale) sempre attraverso l’algoritmo di Dijkstra.



*In giallo il percorso minimo nell’attuale iterazione. Le ‘X’ simboleggiano l’eliminazione del collegamento in quanto già utilizzato in una iterazione precedente.*

Va osservato che con questo algoritmo per la seconda modalità non saranno riportate tutte le possibili relazioni tra le parole, difatti si avrebbe un numero sproporzionato di relazioni data la grande dimensione del grafo. Si è scelto di limitare le relazioni trovate a quelle che differiscono per il collegamento finale in quanto si è visto che questa scelta porta ad avere comunque molte relazioni ma tutte abbastanza rilevanti qualitativamente.

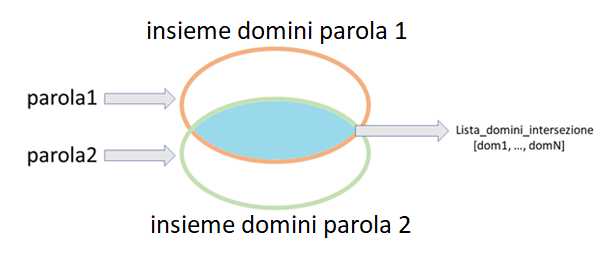
Una volta trovate le relazioni, ovvero i percorsi, componiamo le frasi da mostrare all’utente considerando le etichette dei vertici e degli archi, dove quelle dei vertici corrispondono al significato del synset mentre quelle degli archi corrispondono al tipo della relazione.

# Terza Funzionalità: Contextual Relation Finder

Lo scopo di questa funzione è di ottenere una relazione tra due parole appartenenti allo stesso paragrafo che sia contestualizzata rispetto al paragrafo in questione. Per raggiungere lo scopo si sono utilizzati algoritmi già presentati nella prima e nella seconda funzionalità. Ad esempio per il calcolo della relazione verrà comunque sfruttata la teoria dei grafi e l’algoritmo di creazione del grafo presentato precedentemente, con le opportune modifiche effettuate considerando il dominio del paragrafo. Comunque, partendo dall’inizio, di seguito vengono elencati e discussi, in ordine, tutti i passi dell’algoritmo:

Check ed estrazione paragrafo: si è ritenuto necessario che le parole selezionate appartengano allo stesso paragrafo per evitare di trovare relazioni contestuali “forzate” e, dunque, subito dopo la selezione viene controllato che le parole appartengano allo stesso paragrafo e avviene l’estrazione del paragrafo stesso dal testo (viene considerato un paragrafo il testo presente tra due coppie di \n e dunque è necessario che il testo sia formattato in questo modo per permettere la corretta estrazione).

Lista ordinata dei domini di interesse: la relazione contestualizzata può esistere solo se le due parole selezionate hanno uno o più domini in comune, dunque per entrambe le parole, tramite la libreria WordNet-Domain, viene raccolto l’insieme di domini correlato alla parola e dall’intersezione di questi due insiemi viene ricavata la lista di domini che contiene i domini attraverso i quali è possibile costruire la relazione contestualizzata.



Come vedremo negli step successivi, questi domini verranno utilizzati uno alla volta per tentare di ottenere la relazione e dunque è necessario ordinarli in funzione della loro importanza all’interno del paragrafo.

L’importanza di un dominio è data dalla sua ricorrenza nel paragrafo ovvero quante parole del paragrafo sono collegate a questo dominio, quindi è stato assegnato un contatore ad ogni dominio contenuto nella lista intersezione ed è stata definita una routine in cui vengono analizzate tutte le parole del paragrafo: viene incrementato il contatore riferito ad un determinato dominio ogni volta che una delle parole del paragrafo possiede quel determinato dominio in WordNet-Domain, e dopo aver analizzato tutto il paragrafo, i domini vengono ordinati in funzione del valore di questi contatori.

Estrazione gerarchia di domini per costruzione grafo: la costruzione del grafo, che verrà spiegata a breve, si basa su un solo dominio tra quelli presenti nella lista di intersezione, ovvero comuni ad entrambe le parole che si vuole contestualizzare. Questi domini hanno in comune solamente il fatto di essere associati in qualche modo sia alla prima che alla seconda parola scelte dall’utente, ma ciò non implica necessariamente che, preso un dominio qualsiasi dalla lista, esista una relazione tra le due parole riguardante quello specifico dominio. Per questo motivo, attraverso un algoritmo iterativo, si tenta di realizzare una relazione contestualizzata partendo dal primo dominio e, qualora non ritornasse nulla, riprovando con un secondo dominio fino a scorrere l’intera lista:

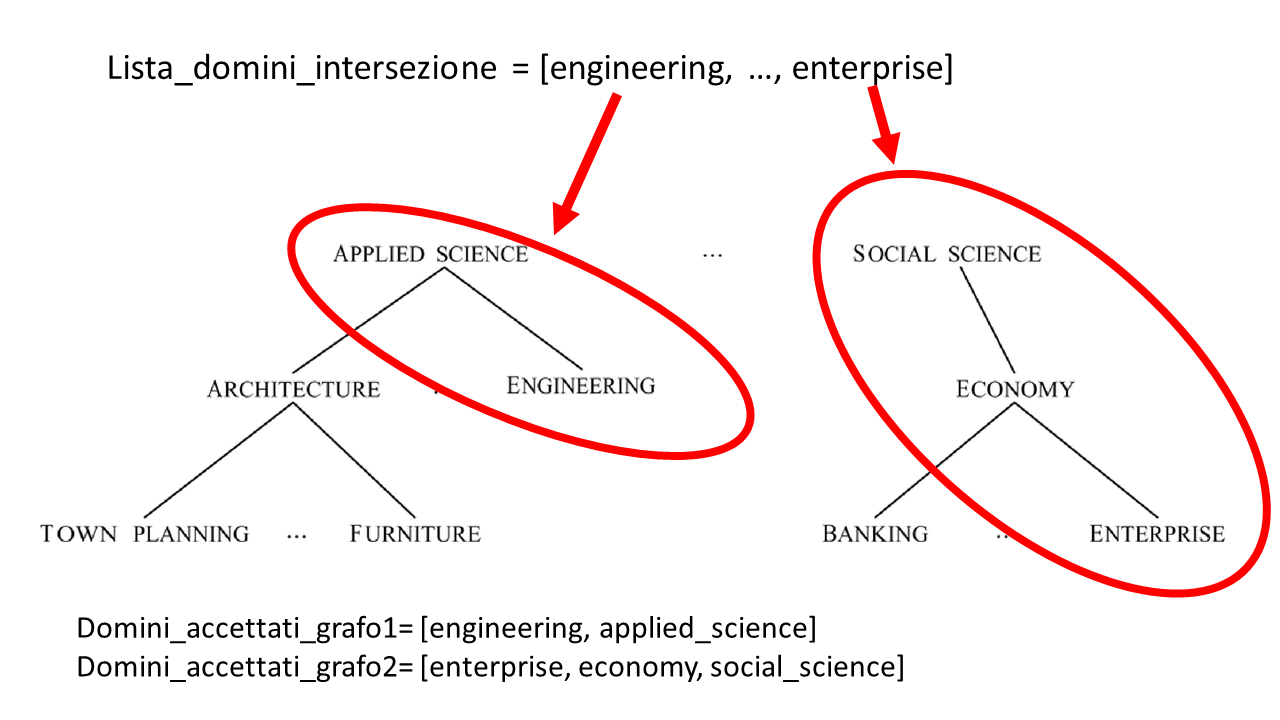
* relazione trovata : la si restituisce e quindi l’algoritmo termina l’esecuzione.
* relazione non trovata: si passa al successivo dominio della lista ordinata costruendo un nuovo grafo (se non esiste un dominio successivo la relazione non può essere trovata e dunque viene comunicato all’utente attraverso una stampa nel textbox).

Come vedremo nel passo successivo, un nodo per essere aggiunto al grafo deve essere collegato al dominio in questione. Il criterio di selezione è stato però allargato anche ai domini gerarchicamente superiori poichè prendere solo i synset del determinato dominio si è rivelata una scelta troppo stringente per ottenere una relazione.

WordNet-Domain non permette di esplorare direttamente la gerarchia dei domini e quindi, allo scopo di ottenere il padre e gli antenati del dominio che si sta analizzando, abbiamo definito un dizionario con elementi coppie del tipo:

[dominio\_figlio, dominio\_padre]

Per i domini radice il dominio\_padre è stato settato a None e tramite questa struttura dati, dunque, otteniamo la lista di domini che utilizzeremo nel prossimo passo per popolare il grafo.

*Esempio di estrazione dei domini accettati a partire dalla lista*

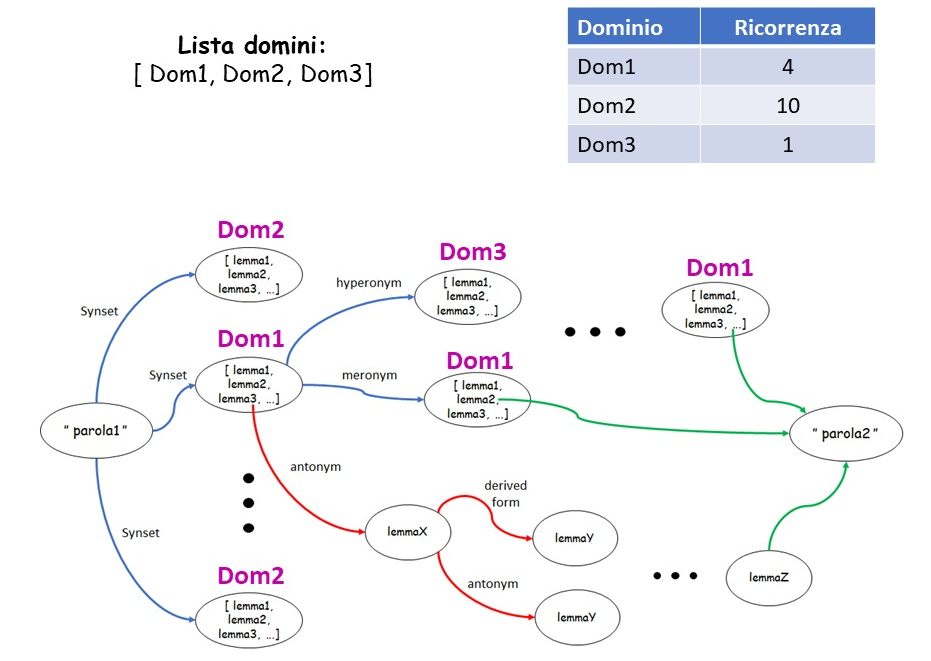
*generata dall’intersezione dei domini.*

Popolamento del grafo e dominio della relazione : Il grafo viene costruito sfruttando lo stesso algoritmo presentato nel precedente capitolo di questa relazione con qualche differenza:

* I synset che vengono mano a mano trovati attraverso i collegamenti semantici vengono effettivamente aggiunti al grafo solo se possiedono almeno uno dei domini della ‘*lista domini accettati*’
* il peso degli archi è sempre unitario. Infatti essendo i nodi del grafo già strettamente correlati (in quanto appartengono ad uno stesso dominio), non si è ritenuto utile introdurre una differenziazione ulteriore tra i vari archi. Ogni percorso sarà qualitativamente buono, verrà selezionato semplicemente il percorso più breve (grazie, come al solito, all’algoritmo di Dijkstra)

Inoltre, si è ritenuto opportuno analizzare quale dei domini accettati permette effettivamente l’aggiunta del synset al grafo in modo da ottenere un “dominio della relazione”. A livello implementativo si è scelto di aggiungere un dizionario come attributo del grafo che contiene le coppie [dominio, contatore] e dunque ogni volta che un synset viene aggiunto al grafo si incrementa il contatore associato al dominio che ha permesso la sua aggiunta e al termine della creazione del grafo viene selezionato come “dominio della relazione” il dominio con l’occorrenza maggiore.

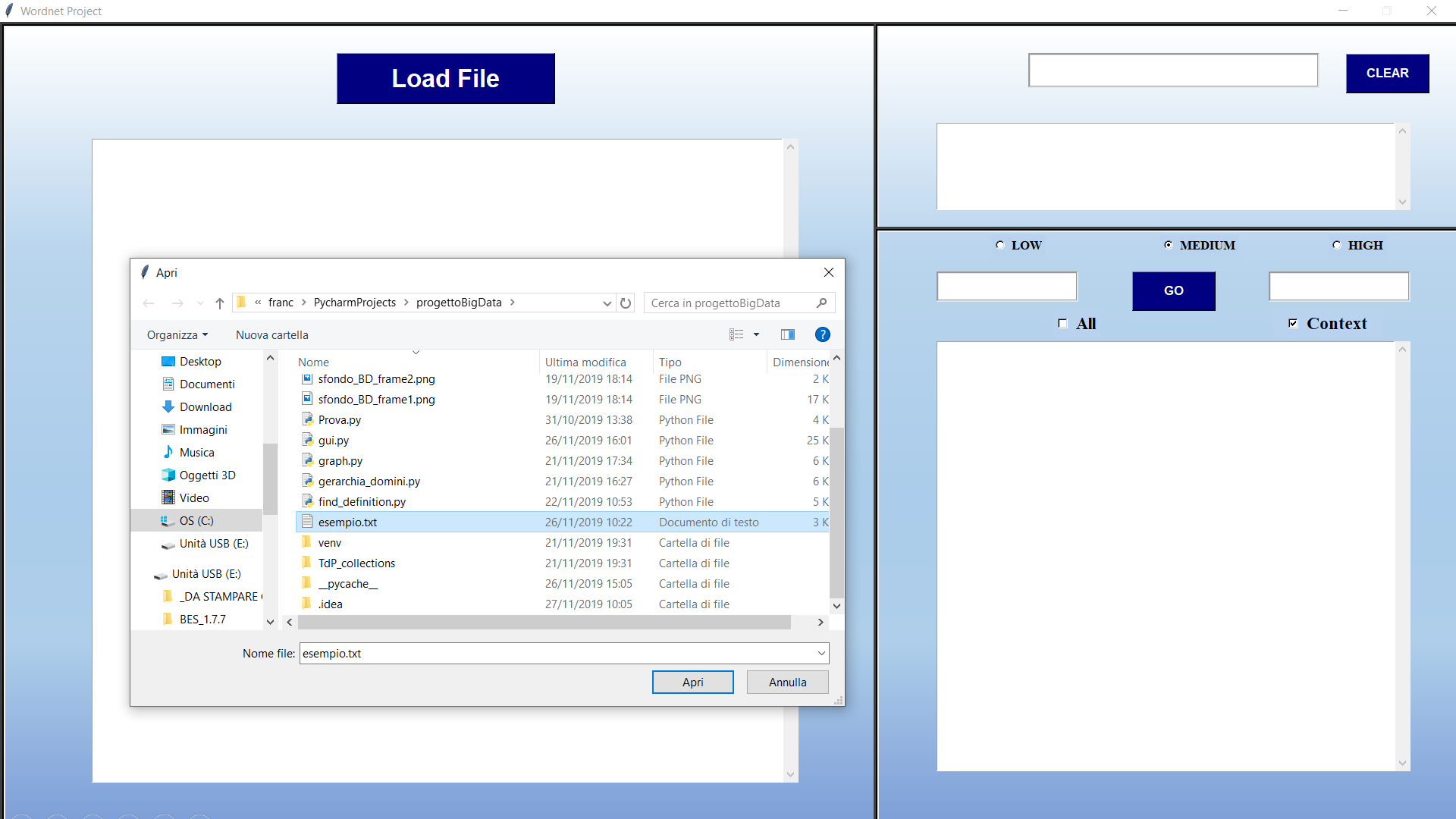
Nel caso in cui il grafo non risulti connesso, ovvero non esiste una relazione tra le due parole nel contesto considerato, si riparte con la costruzione di un nuovo grafo considerando il successivo dominio della lista intersezione.



Come mostrato nell’immagine precedente i nodi che contengono lemmi non hanno un dominio di riferimento, questa eccezione è dovuta al fatto che gli oggetti di tipo lemma non hanno un dominio (WordNet-Domain assegna un dominio solo ai synset); per scelta progettuale abbiamo deciso di includere ugualmente questo tipo di nodi in quanto rappresentano l’aggiunta di relazioni spesso fondamentali per trovare la giusta soluzione (“opposto” e “forma lessicale derivata”) e quindi, per questi specifici nodi, le restrizioni di dominio semplicemente non vengono applicate mentre il calcolo del percorso minimo tramite Dijkstra considera normalmente il peso degli archi (unitario) che li collegano.

# Interfaccia Grafica

Al fine di semplificare l’utilizzo dell’applicazione, è stata realizzata un’interfaccia grafica di semplice utilizzo di cui se ne spiegano caratteristiche e funzionalità in questa sezione.



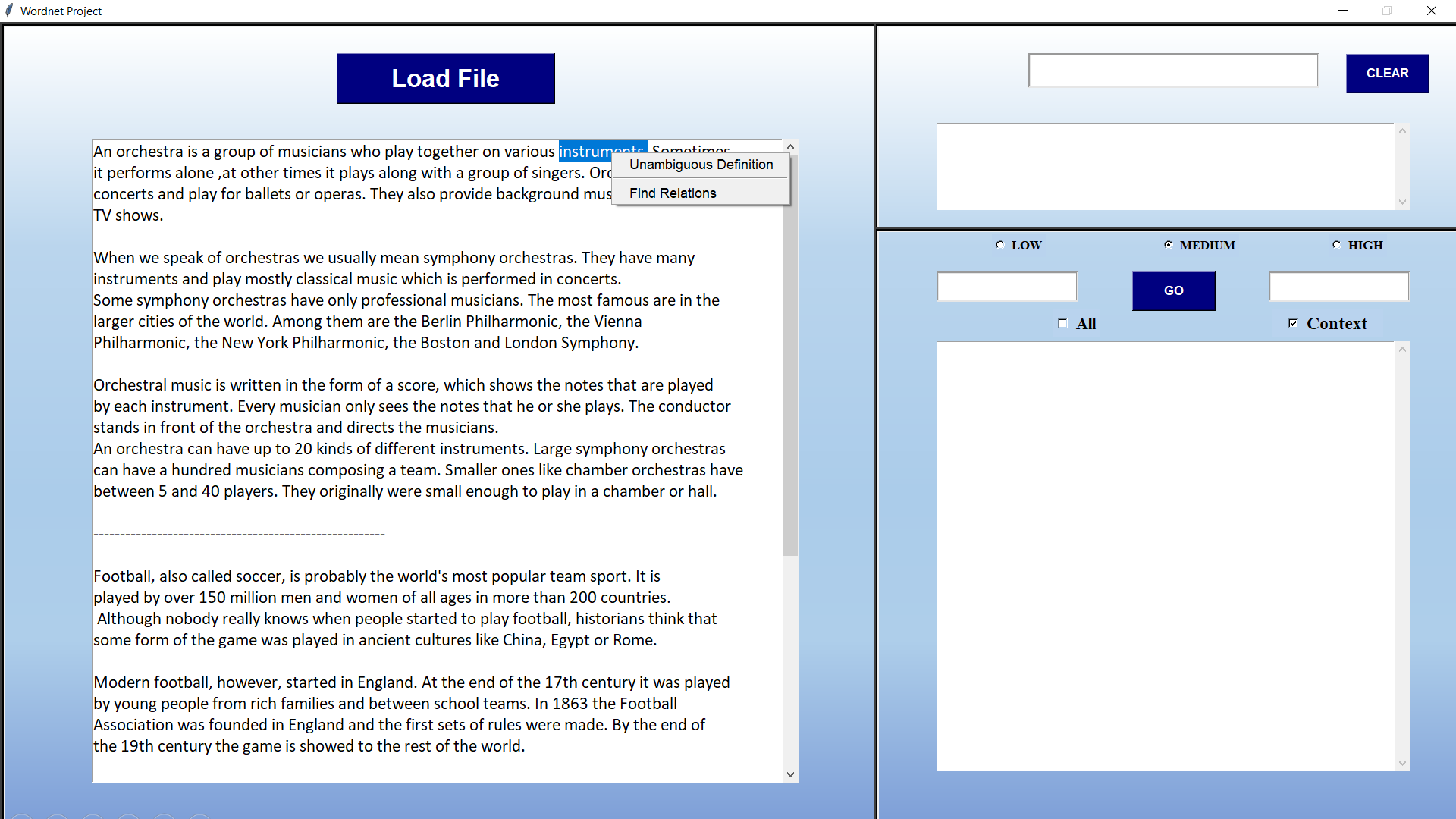
*Schermata applicazione dopo la selezione del bottone “Load File”*

Com’è possibile vedere dalla figura, all’avvio dell’applicazione l’utente potrà, dunque, selezionare un file in formato “txt” tramite la finestra di dialogo che compare dopo aver selezionato l’opzione “Load File”.

A questo punto, nel textbox della sezione di sinistra viene caricato il testo presente nel file su cui è, quindi, possibile applicare le funzionalità previste.

In particolare, è necessario selezionare la parola su cui si vuole applicare la funzionalità e, dal menù a tendina che compare dopo la pressione del tasto destro del mouse, scegliere se interrogare “Unambiguous Definition”, il cui risultato comparirà nel textbox in alto a destra, oppure “Find Relations”, il cui risultato comparirà nel textbox in basso a destra. Da sottolineare che la prima funzionalità considera non solo la parola, ma anche la frase a cui appartiene (ipotizzando che questa termini o inizi dopo un segno di interpunzione quali “.” “;” “?” “!” o una nuova linea).

L’esempio sottostante è utile per chiarire l’utilizzo delle due funzionalità.



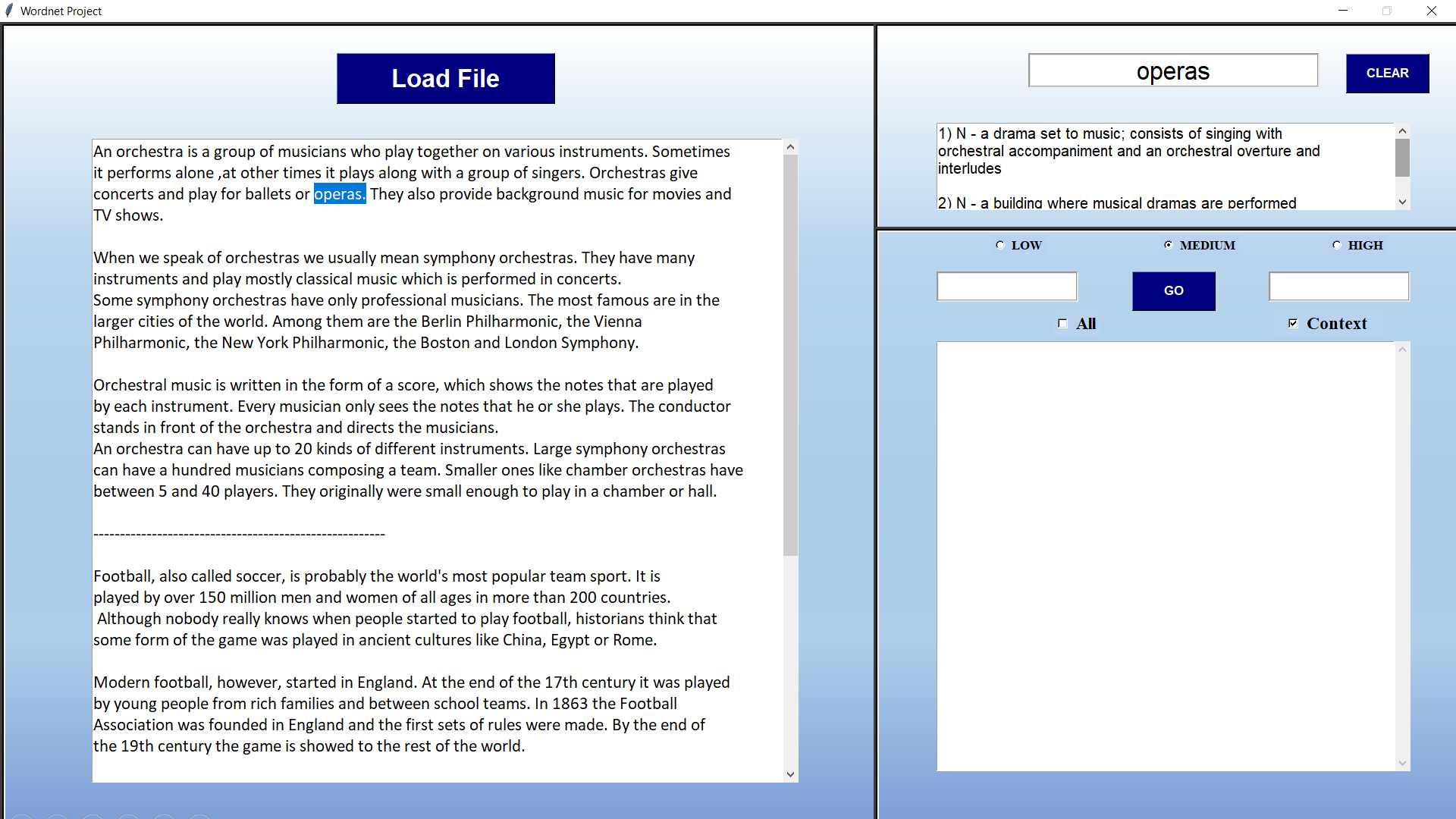
*Menù a tendina che compare dopo la pressione del tasto destro del mouse*

E’ possibile evidenziare la prima sostanziale differenza tra le funzionalità. In particolare la prima si avvia non appena viene selezionata nel menu a tendina (ovviamente dopo aver evidenziato una parola) mentre, per avviare la seconda, è necessario selezionare due parole, il livello di approfondimento (ovvero il numero di iterazioni), il tipo di funzionalità (dato che è possibile selezionarne di due tipi) e premere il bottone GO una volta pronti. Riguardo le due parole selezionate, bisogna sottolineare che queste devono appartenere allo stesso paragrafo (altrimenti viene segnalato un errore).

Come ragionevole scelta progettuale, un paragrafo è compreso tra due nuove linee vuote. Inoltre, com’è possibile vedere dalla figura, il livello di approfondimento ha come default il valore “MEDIUM” e il tipo di funzionalità ha come valore di default “Context”. Inoltre, è possibile selezionare o deselezionare “All” al fine di scegliere se si vuole ottenere come risultato tutte le relazioni oppure solo la più rilevante. Si noti che non è possibile selezionare “All” quando è attivo “Context”.

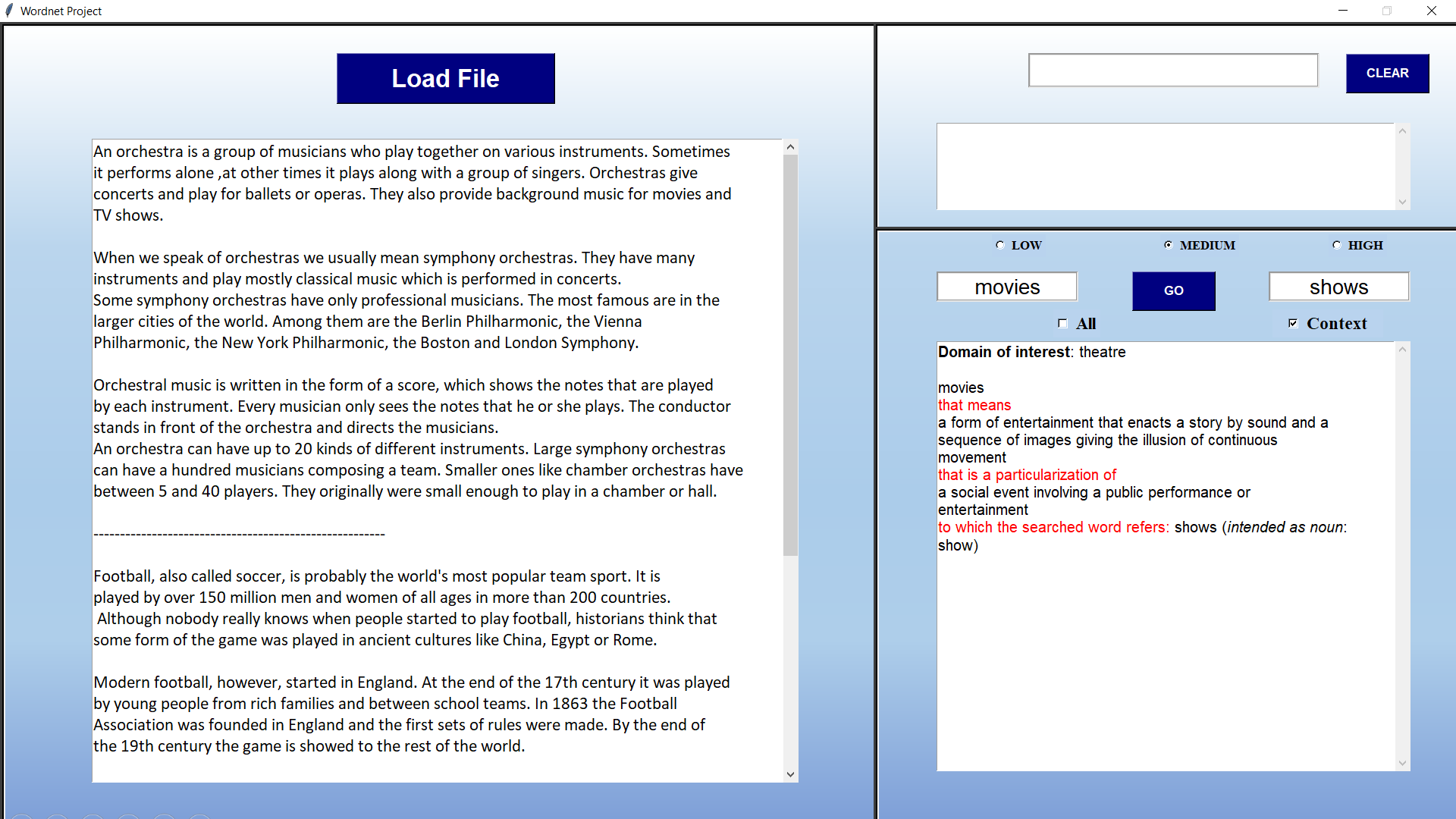
Vediamo, ora, alcuni esempi. L’obiettivo è quello di aiutare l’utente che utilizza l’applicazione a comprendere il significato dei risultati in alcuni casi d’uso.

Riguardo la prima funzionalità, che comincerà a calcolare i risultati non appena l’utente seleziona una parola e “Unambiguous Definition” dal menu a tendina, notiamo che nella label compare la parola selezionata mentre nel textbox compare un elenco di significati (il cui criterio è stato illustrato nella sezione apposita). Prima del significato, viene mostrata anche la categoria sintattica della parola mediante l’utilizzo di una lettera significativa (V per verbo, R per avverbio, S per aggettivo, N per sostantivo). E’, inoltre, presente il bottone di clear, utile per pulire il contenuto della label e della textbox.



***Esempio:*** *Prima funzionalità*

Riguardo la seconda funzionalità, come detto è possibile selezionare in maniera esclusiva il livello di approfondimento delle relazioni (“LOW”, “MEDIUM”, “HIGH”), oltre che se volere come risultato relazioni generali tra le due parole oppure relazioni dipendenti dal contesto. Alla pressione del bottone GO, l’applicazione avvia un thread che calcola quanto chiesto e fornisce i risultati nel textbox.



***Esempio:*** *Seconda funzionalità*

Le parole in rosso nel textbox sono quelle parole chiave che indicano il tipo di relazione che intercorre tra le varie righe del textbox stesso, ottenute mediante il percorso del grafo che collega le due parole.

Una differenza grafica tra la funzionalità context e quella non context che è utile evidenziare è la comparsa della stringa “Domain of interest” (evidenziata in grassetto) dato che questa compare solo nel caso in cui si richiede la relazione inerente al contesto, in quanto indica il dominio comune delle due parole estrapolato dal contesto del paragrafo.