

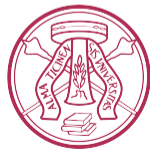
Estrarre dati dalle treebank con UDeasy senza imparare a programmare

Discorsi sul Metodo 2023

Luca Brigada Villa

University of Pavia

9 maggio 2023



UNIVERSITÀ
DI PAVIA

1 Treebank

Cosa sono
A cosa servono

2 Universal Dependencies

Cos'è UD
Formato
Annotazione secondo le guidelines

3 UDeasy

A cosa serve
Come si usa

4 Tutorial

L'italiano è una lingua SVO?
Quali parole modifica il lemma *bello*?
Stare + X
L'italiano è una lingua pro-drop?

The background features a light pink color scheme with stylized architectural elements. On the left and right sides, there are two identical sets of two columns supporting a horizontal beam, with a large arch above each. The word "Treebank" is centered within the arch on the left.

Treebank

Una treebank è una **risorsa annotata sintatticamente**.

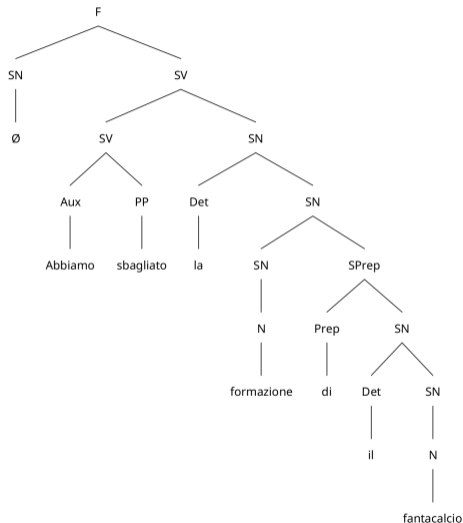
tree-bank → insieme di alberi (sintattici)

Questi alberi possono avere una struttura e seguire delle regole diverse per la loro costruzione. Identifichiamo due principali “regole”:

- alberi a costituenti
- alberi a dipendenze

A seconda della “regola” seguita, gli alberi che rappresentano la sintassi delle diverse frasi nella treebank avranno una struttura diversa.

Esempio di albero a costituenti



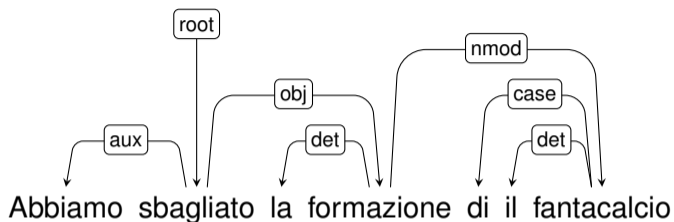
FRASE

Abbiamo sbagliato la formazione del fantacalcio

Cose da notare

- gli elementi della frase sono raggruppati quando sono vicini
- un sacco di nodi “vuoti”

Esempio di albero a dipendenze



FRASE: Abbiamo sbagliato la formazione del fantacalcio

Cose da notare:

- tutte le parole sono collegate ad un'altra (tranne una)
- non ci sono nodi "vuoti"

A cosa servono le treebank

Le treebank vengono utilizzate per diversi scopi, tra cui:

- analisi linguistica:
 - sincronica
 - diacronica
- sviluppo di modelli di NLP:
 - parser
 - analizzatori morfologici
 - language models
 - (in passato) traduzione automatica
- documentare alcune lingue

The background features a stylized architectural design with two sets of columns supporting arches. The columns are light red, and the arches are a slightly darker shade of red. The overall aesthetic is clean and modern.

Universal Dependencies

Univesal Dependencies è un progetto avviato nel 2014 e che ha tra i suoi scopi:

- sviluppare un'annotazione di treebank a dipendenze coerente tra le lingue
- favorire lo sviluppo di parser multilingui
- facilitare la ricerca linguistica in prospettiva tipologica

Per farlo, ha sviluppato uno schema di annotazione che si basa:

- sulle Stanford dependencies per la sintassi
- sul Google universal part-of-speech tagset per le parti del discorso
- sull'Intersect interlingua tagset per l'annotazione morfologica

Le treebank sono dei semplici file di testo che si possono aprire con un editor di testo:



Sono formattate in **CoNLL-U**, un formato che:

- rappresenta ogni token con la relativa annotazione in una riga
- separa le frasi con una riga vuota

Un token in formato CoNLL-U ha dieci campi (colonne), separati da un carattere “tab”:

- 1 `id`: l'ID della parola, che deve essere unico all'interno della frase. Ogni parola deve avere un ID (partendo da 1).
- 2 `form`: la forma in cui compare la parola nella frase.
- 3 `lemma`: la forma di riferimento della parola.
- 4 `upos`: la universal part-of-speech della parola.
- 5 `xpos`: la part-of-speech specifica della parola.
- 6 `feats`: le feature morfologiche della parola (genere, numero, caso, modo, tempo...).
- 7 `head`: l'id della parola da cui dipende sintatticamente.
- 8 `deprel`: il tipo di dipendenza sintattica tra la parola e la sua head.
- 9 `deps`: le dipendenze sintattiche complete della parola, in formato `head:deprel`.
- 10 `misc`: eventuali informazioni aggiuntive.

Precisazioni sul formato di alcuni campi

- *multiword tokens*: l'unità di base di annotazione in UD sono le *syntactic words*. Parole che contengono clitici vengono sistematicamente separate (preposizioni articolate in italiano, verbi con clitici...). Per annotarle in conllu, si procede in questo modo:
 - l'id del multiword token compare in questa forma: PRIMO-ULTIMO
 - tutti i campi ad eccezione dell'id e della *form* rimangono vuoti nel multiword token quindi con un *underscore* (`_`)
 - nelle righe successive si annotano (normalmente) i singoli elementi che formano il multiword token
- i campi *feats* e *misc*:
 - si annotano con una serie di coppie *key-value* separate da un carattere *pipe* (`|`)
 - sia la *key* che il *value* si scrivono convenzionalmente in CamelCase (non in dromedaryCase, non in UPPERCASE, non in lowercase, non in snake_case e nemmeno in kebab-case)

Esempio di frase in formato conllu

```
# text = Abbiamo sbagliato la formazione del fantacalcio.
# id_sent = sent-001
1 Abbiamo avere AUX VA Mood=Ind|Number=Plur|Person=1|Tense=Pres|VerbForm=Fin 2 aux 2:aux _
2 sbagliato sbagliare VERB V Gender=Masc|Number=Sing|Tense=Past|VerbForm=Part 0 root 0:root _
3 la il DET RD Definite=Def|Gender=Fem|Number=Sing|PronType=Art 4 det 4:det _
4 formazione formazione NOUN S Gender=Fem|Number=Sing 2 obj 2:obj _
5-6 del _ _ _ _ _ _ _ _
5 di di ADP E _ 7 case 7:case _
6 il il DET RD Definite=Def|Gender=Masc|Number=Sing|PronType=Art 7 det 7:det _
7 fantacalcio fantacalcio NOUN S Gender=Masc|Number=Sing 4 nmod 4:nmod _
```

Il terzo campo di un token nel formato conllu è dedicato all'annotazione delle universal parts-of-speech.

La lista di POS annotabili si può trovare **qui** e include un set limitato di tag assegnabili ai token. I tag sono suddivisi in tre categorie, assegnabili a tre diversi tipologie di token:

- *open class words*: ADJ, ADV, INTJ, NOUN, PROPN, VERB
- *closed class words*: ADP, AUX, CCONJ, DET, NUM, PART, PRON, SCONJ
- *other*: PUNCT, SYM, X

Il sesto campo di un token nel formato conllu è dedicato all'annotazione delle features.

La lista di features e i valori associati ad esse si può trovare **qui** e include un set limitato di tag assegnabili ai token. I tipi di features sono divisi in due gruppi principali:

- *lexical features*: PronType, NumType, Poss, Reflex, Foreign, Abbr, Typo
- *inflectional features*:
 - *nominal*: Gender, Animacy, NounClass, Number, Case, Definite, Degree
 - *verbal*: VerbForm, Mood, Tense, Aspect, Voice, Evident, Polarity, Person, Polite, Clusivity

Per annotare la **struttura sintattica** delle frasi seguendo le guidelines di Universal Dependencies si seguono alcuni principi:

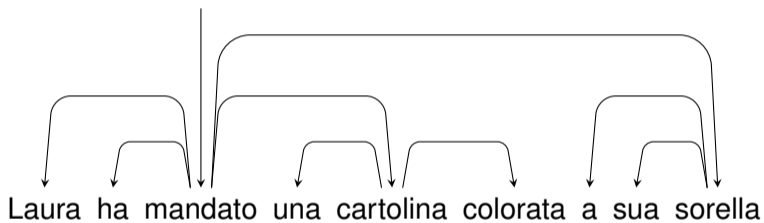
- le *content words* fanno da scheletro dell'albero sintattico
- le *function words* tendenzialmente dipendono dalle *content words*
- i modificatori dipendono dalle parole modificate

Proviamo ad annotare la frase “Laura ha mandato una cartolina colorata a sua sorella”.

Per annotare una frase come “Laura ha mandato una cartolina colorata a sua sorella”:

- 1 identifichiamo le content words: “Laura”, “mandato”, “cartolina”, “colorata”, “sorella”
- 2 per ognuna di esse, identifichiamo la head:
 - “Laura” ← “mandato”
 - “cartolina” ← “mandato”
 - “colorata” ← “cartolina”
 - “sorella” ← “mandato”
- 3 per ognuna delle parole rimaste (*function words*), identifichiamo la head:
 - “ha” ← “mandato”
 - “una” ← “cartolina”
 - “a” ← “sorella”
 - “sua” ← “sorella”

Annotare secondo le guidelines di UD: struttura



Annotare secondo le guidelines di UD: relazioni sintattiche

Una volta strutturata la frase, si può passare a etichettare ciascuna delle relazioni sintattiche presenti in essa. Per farlo è possibile scegliere l'etichetta tra le label proposte da UD (consultabili **qui**).

Per la nostra frase sceglieremo queste:

- `root`: indica la root della frase; si assegna al token che non ha una `head`
- `nsubj`: sta per “nominal subject”; si assegna al soggetto sintattico della frase
- `obj`: sta per “object”; si assegna al secondo argomento più importante dopo il soggetto
- `iobj`: sta per “indirect object”; si assegna ad alcuni tipi di argomenti dei verbi (sempre *core arguments*). Assegnato solitamente al *recipient*
- `det`: sta per “determiner”; questa etichetta si assegna al link tra una testa nominale e il suo determinante. Prende la sottocategoria `poss` per i possessivi
- `case`: si usa per qualsiasi *syntactic word* che marca il caso (preposizioni, postposizioni...)
- `amod`: sta per “adjectival modifier”; si assegna all'aggettivo o al sintagma aggettivale che modifica un nome o un pronome
- `aux`: sta per “auxiliary”; si assegna alla *function word* associata ad un predicato verbale che esprime categorie come tempo, aspetto, modo, diatesi...

Annotare secondo le guidelines di UD: relazioni sintattiche

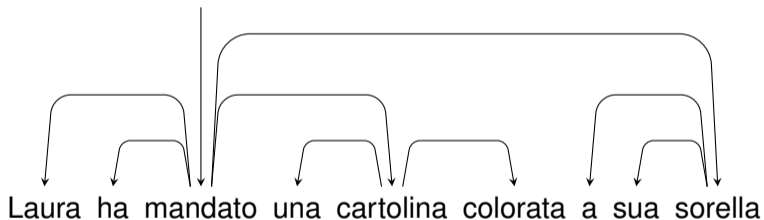
Proviamo ad assegnare queste etichette alle relazioni sintattiche della frase di prima:

root
det

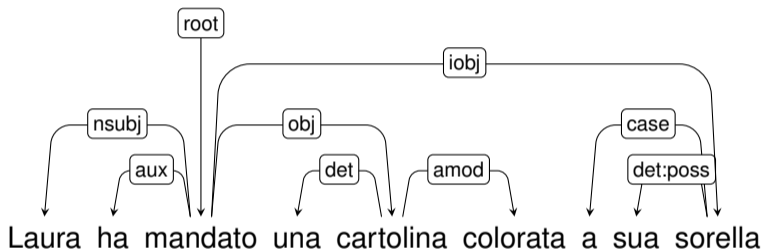
nsubj
case

obj
amod

iobj
aux



Annotare secondo le guidelines di UD: relazioni sintattiche



The background features a stylized architectural design with two sets of columns supporting arches. The columns are light red, and the arches are a slightly darker shade of red. The text 'UDeasy' is centered within the central arch.

UDeasy

UDeasy è un tool che permette di interrogare ed estrarre occorrenze dalle treebank con un'interfaccia grafica che ha l'obiettivo di rendere questa operazione il più semplice possibile.

È disponibile per tre sistemi operativi:

- Ubuntu
- MacOS
- Windows

Download

Prima di iniziare ad usare UDeasy, l'utente dovrebbe scaricare la versione del software compatibile con il proprio sistema operativo alla pagina

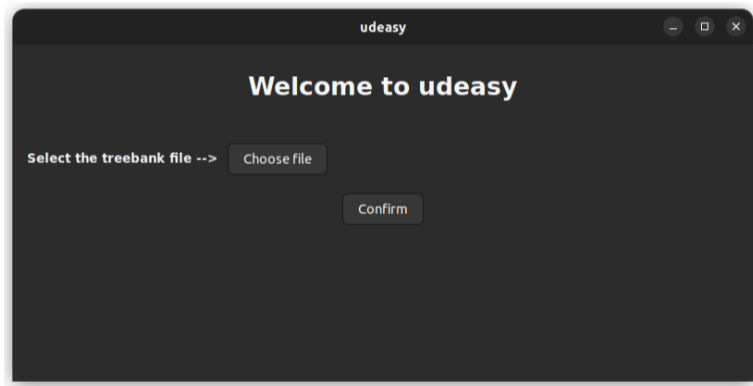
<https://unipv-larl.github.io/udeasy/download.html>.

Installazione

Dopo aver scaricato il programma, si può procedere all'installazione di UDeasy. Una volta installato si può aprire il programma con un doppio click.

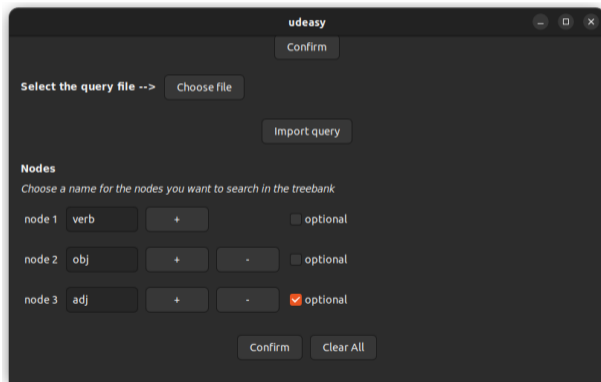
Usare UDeasy

All'apertura del programma comparirà una schermata in cui verrà chiesto all'utente di selezionare un file conllu da cui estrarre le occorrenze.



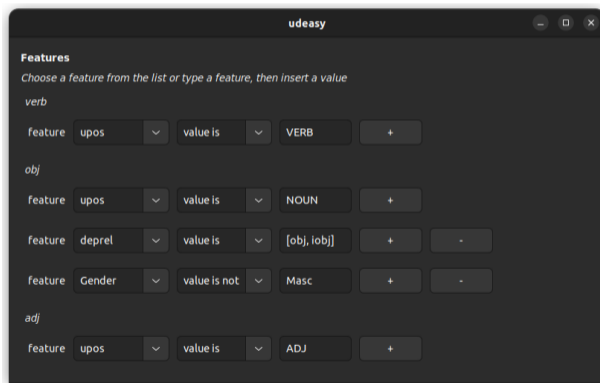
Node names

Una volta selezionato il file conllu, compariranno due sezioni nella finestra principale: una servirà ad importare una query già preparata in precedenza, l'altra per iniziare a crearne una da zero.



Features

Una volta inseriti i nomi con cui ci riferiremo ai token da cercare nella treebank, possiamo selezionare le features che questi token devono (o non devono) avere.

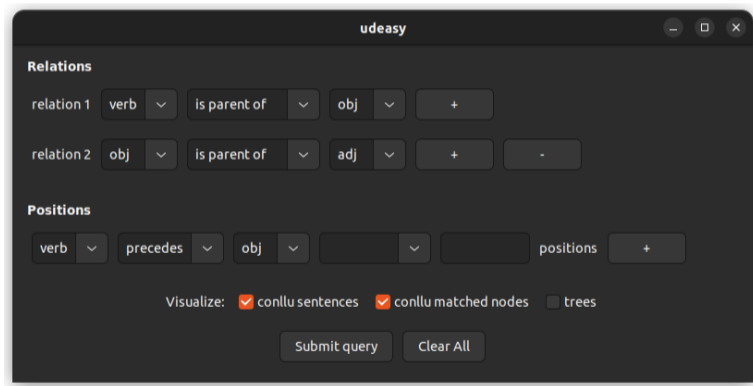


The screenshot shows a window titled "udeasy" with a "Features" section. Below the title, there is a subtitle: "Choose a feature from the list or type a feature, then insert a value". The interface is organized into sections for different parts of speech: "verb", "obj", and "adj". Each section contains a "feature" label followed by a dropdown menu, a "value is" or "value is not" dropdown menu, a text input field, and one or two buttons labeled "+" and "-".

Part of Speech	Feature	Value	Operator	Buttons
verb	upos	value is	VERB	+
obj	upos	value is	NOUN	+
obj	deprel	value is	[obj, iobj]	+, -
obj	Gender	value is not	Masc	+, -
adj	upos	value is	ADJ	+

Relazioni sintattiche e posizioni all'interno della frase

Più in basso nella schermata sarà possibile anche specificare le relazioni sintattiche tra i token cercati e le posizioni reciproche tra di essi.



The screenshot shows a dark-themed web interface titled "udeasy". It features two main sections: "Relations" and "Positions".

Relations:

- relation 1: verb (dropdown), is parent of (dropdown), obj (dropdown), + (button)
- relation 2: obj (dropdown), is parent of (dropdown), adj (dropdown), + (button), - (button)

Positions:

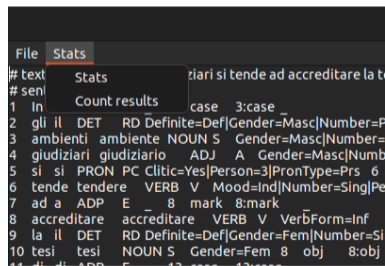
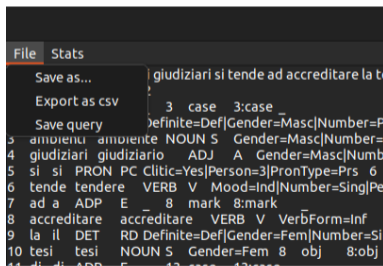
- verb (dropdown), precedes (dropdown), obj (dropdown), (empty dropdown), (empty dropdown), positions (text), + (button)

Visualize: conllu sentences conllu matched nodes trees

Buttons: Submit query, Clear All

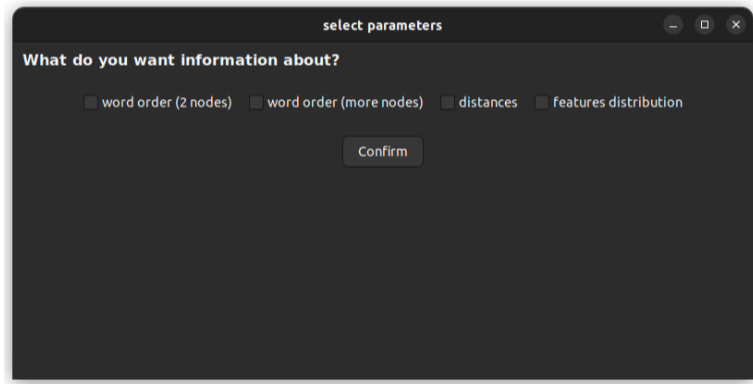
Lavorare con i risultati dell'estrazione

Una volta lanciata la query, i risultati compariranno in una nuova finestra. I risultati si potranno salvare sia come testo semplice (cioè come compagno), sia come csv (formato importabile anche in Excel o simili) selezionando i campi che si vogliono esportare. È possibile inoltre salvare la query (per riprodurla su una treebank diversa) ed estrarre statistiche, anche elaborate, dai dati estratti.

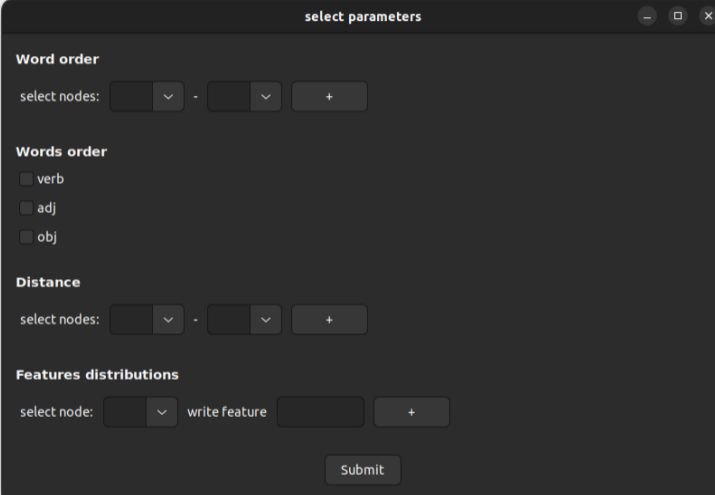


Estrarre statistiche dai risultati

UDEasy permette di estrarre delle informazioni statistiche sui dati estratti. Per farlo, si dovrà selezionare tra quelle disponibili l'informazione (o le informazioni) che si desidera estrarre dai dati.



Una volta selezionate, andranno riempiti i campi per estrarre effettivamente le informazioni sulle frequenze/ordini/cooccorrenze dei token selezionati.



The image shows a dark-themed dialog box titled "select parameters" with standard window controls (minimize, maximize, close) in the top right corner. The dialog is organized into four sections:

- Word order:** A label "select nodes:" followed by two empty dropdown menus, a minus sign "-", and a plus sign "+" button.
- Words order:** Three unchecked checkboxes labeled "verb", "adj", and "obj".
- Distance:** A label "select nodes:" followed by two empty dropdown menus, a minus sign "-", and a plus sign "+" button.
- Features distributions:** A label "select node:" followed by an empty dropdown menu, the text "write feature", another empty dropdown menu, and a plus sign "+" button.

A "Submit" button is located at the bottom center of the dialog.

The background features a light pink color scheme with stylized architectural elements. On the left and right sides, there are two identical structures, each consisting of two vertical columns supporting a horizontal entablature, which in turn supports a large, rounded arch. The word "Tutorial" is centered within the space between these two arches.

Tutorial

In questo tutorial lavoreremo con una porzione (7000 frasi, 161557 token) della treebank ISDT di Universal Dependencies. Ci chiederemo:

- 1 se l'italiano è una lingua SVO
- 2 quali parole vengono modificate dal lemma *bello*
- 3 com'è coniugato il verbo da cui dipende *stare*
- 4 se l'italiano è una lingua pro-drop

I file per questo tutorial si possono scaricare qui:

<https://unipv-larl.github.io/udeasy/tutorials.html>.

L'italiano è una lingua SVO?

Per rispondere a questa domanda dobbiamo prima pensare ad una query che possa estrarre delle occorrenze dalla treebank e successivamente andare ad analizzare l'ordine degli elementi all'interno di queste occorrenze.

- cerchiamo tre nodi che chiameremo *subj*, *verb* e *obj*
- restringiamo i risultati ai soli soggetti e oggetti nominali
- specifichiamo il tipo di dipendenze sintattiche
- specifichiamo le relazioni tra i nodi
- scegliamo le opzioni di visualizzazione che preferiamo

Quali parole modifica il lemma *bello*?

Per rispondere a questa domanda dobbiamo prima pensare ad una query che possa estrarre delle occorrenze dalla treebank e successivamente andare ad analizzare alcune caratteristiche degli elementi all'interno di queste occorrenze.

- cerchiamo due nodi che chiameremo *word* e *bello*
- per il nodo *word* non specifichiamo nessuna feature
- per il nodo *bello* specifichiamo il lemma
- specifichiamo le relazioni tra i nodi
- scegliamo le opzioni di visualizzazione che preferiamo

Com'è coniugato il verbo da cui dipende *stare*?

Per rispondere a questa domanda dobbiamo prima pensare ad una query che possa estrarre delle occorrenze dalla treebank e successivamente andare ad analizzare alcune caratteristiche degli elementi all'interno di queste occorrenze.

- cerchiamo due nodi che chiameremo *stare* e *verb*
- specifichiamo le features del nodo *stare* (`lemma`)
- specifichiamo le features del nodo *verb* (`upos`)
- definiamo la struttura sintattica di questa coppia di nodi
- scegliamo le opzioni di visualizzazione che preferiamo

L'Italiano è una lingua pro-drop?

Per rispondere a questa domanda dobbiamo prima pensare ad una query che possa estrarre delle occorrenze dalla treebank e successivamente andare ad analizzare alcune la frequenza con cui questi elementi compaiono.

- cerchiamo due nodi che chiameremo *subj* (opzionale) e *verb*
- specifichiamo le features dei due nodi (*depre1*, *upos*, *VerbForm*, *PronType*)
- definiamo la struttura sintattica
- scegliamo le opzioni di visualizzazione che preferiamo

Grazie per l'attenzione!

✉ luca.brigadavilla@unibg.it

👤 bavagliladri

👤 unipv-larl

🔗 UDeasy website