Relazione TLN sui laboratori 1-5

Patric Reineri, Alessandro Scicolone

September 9, 2025

1 Laboratorio 1: Integrazione tra WordNet e FrameNet

In questa esercitazione integriamo le risorse FrameNet e WordNet, unendo i concetti di frame e lexical unit ai synset in wordnet.

L'obiettivo principale di questo approccio è facilitare l'annotatore nella scelta del frame corretto in presenza di una lexical unit ambigua, ovvero che può attivare più frame. L'associazione automatica tra LU di FrameNet e synset di WordNet, ottenuta tramite Lesk, fornisce un supporto semantico aggiuntivo in modo tale da aiutare l'annotatore nella scelta del frame.

Nella fase di implementazione abbiamo seguito i seguenti passi:

1. **Estrazione delle frasi**: Si parte da un corpus (Brown Corpus) da cui vengono selezionate casualmente n frasi.
2. **Individuazione delle Lexical Units (LU)**: Per ogni parola della frase, dopo lemmatizzazione e POS tagging, si identificano tutte le lexical unit che attivano un frame in FrameNet.
3. **Associazione dei synset WordNet tramite Lesk**: Per ogni parola, si ricercano i synset di WordNet più rilevanti utilizzando l'algoritmo di Lesk, passando come input la parola e la frase dell'iterazione corrente (contesto).
4. **Ottenimento dei risultati**: Tutte le informazioni raccolte su frame e synsets vengono salvate in un file JSON, che può essere successivamente visualizzato anche come grafo ad albero tramite la libreria Graphviz.

Ad esempio per il lexical unit "compound" della frase "he wanted the mission compound to be effortlessly spotless .":


```
{
    "word": "compound.n",
    "frames": [
        {
            "name": "Locale_by_use",
            "definition": "Geography as defined by ..."
        }
    ],
    "synset": {
        "Name": "compound.v.05",
        "Definition": "combine so as to form a whole; mix",
        "Examples": [
            "compound the ingredients"
        ]
    }
}
```


In questo caso abbiamo solo un frame attivato, ma ad esempio nel caso dei verbi, dove vengono attivati molti frame, la presenza del synset aiuta l'annotatore di scegliere il giusto frame.

2 Laboratorio 2

In questo esercizio si parte da un file xlsx contenente, per ciascuna delle quattro categorie (Concreto-Generico, Concreto-Specifico, Astratto-Generico, Astratto-Specifico), un insieme di definizioni. L'obiettivo è calcolare per ogni termine due valori medi:

- La similarità lessicale (**SimLex**): utilizziamo CountVectorizer della libreria sklearn per calcolare le frequenze delle parole nelle definizioni. Per ogni definizione x si individuano le k parole più frequenti (top-k). Successivamente per ogni altra definizione y andiamo a vedere quante delle k parole più frequenti della definizione x sono presenti anche nella definizione y e dividiamo per k ottenendo il calcolo della copertura (se tutte le k parole sono condivise = 1.0, se nessuna = 0.0). Dopo aver fatto questo per ogni definizione del termine, il valore medio come simlex.
- La similarità semantica (**SimSem**). Si utilizza Sentence-BERT per ottenere un vettore di embedding per ciascuna definizione. Successivamente si calcola la similarità coseno tra tutti i possibili accoppiamenti di definizioni dello stesso termine e si prende la media dei valori ottenuti.

I valori medi di SimLex e SimSem per ciascun termine vengono organizzati in una tabella 2x2, con le righe corrispondenti a Generico/Specifico e le colonne a Concreto/Astratto:

```
    Results in the format (simlex, simsem):
                   Astratto          Concreto
Generico   (0.1312, 0.6529)  (0.2119, 0.6594)
Specifico  (0.0553, 0.4542)  (0.2277, 0.6325)


Average Generico SimLex: 0.1716, SimSem: 0.6561
Average Specifico SimLex: 0.1415, SimSem: 0.5433


Average Concreto SimLex: 0.2198, SimSem: 0.6459
Average Astratto SimLex: 0.0932, SimSem: 0.5536
```

Dai risultati ottenuti possiamo osservare che soprattutto per la similarità lessicale, i valori medi per i termini concreti sono più alti rispetto a quelli astratti. Questo suggerisce che le definizioni di termini concreti tendono a condividere più parole comuni, probabilmente perché descrivono oggetti o concetti tangibili che sono più facilmente rappresentabili con un vocabolario di parole specifico e condiviso. Invece dall'aggregazione per confrontare generico e specifico, vediamo una similarità semantica e lessicale più alta per i termini generici.

3 Laboratorio 3

A partire dalle definizioni raccolte per ciascun termine del file citato nel laboratorio 2, l'obiettivo è individuare il synset di WordNet che meglio rappresenta il termine stesso.

Il metodo implementato per risalire al termine a partire dall'insieme di definizioni (ricerca onomasiologica) si basa sul principio di genus et differentia:

1. Si identificano i Genus, ovvero le categorie generali che nel nostro caso sono gli iperonimi in wordnet. Questi iperonimi sono stati individuati utilizzando le parole più frequenti (top k) considerando tutte le definizioni di un termine.
2. Una volta ottenuti gli iperonimi, andiamo a cercare l'iponimo la cui definizione risulta più simile, secondo una misura di similarità semantica (simsem) calcolata tramite Sentence-BERT, alle definizioni del termine presenti nel file excel. Il termine sarà il lemma principale identificativo di quell'iponimo che massimizza la similarità.

4 Laboratorio 4: Pipeline di clustering e di Topic Modelling

Introduzione

Nel laboratorio 4 è stato scelto come dataset `GonzaloA/fake_news` da Hugging Face, contenente testi relativi a notizie false. L'obiettivo è stato quello di costruire due pipeline principali: una per il **clustering**

e una per il **topic modelling**.

Preprocessing

Prima dell'applicazione delle pipeline, i testi sono stati sottoposti a diverse trasformazioni per migliorarne la qualità:

- **Lemmatizzazione**: per ridurre le parole alle loro forme base.
- **Rimozione delle stop-words**: per eliminare articoli, congiunzioni e termini poco informativi.

Queste operazioni hanno permesso di ridurre il rumore e di uniformare le varianti lessicali presenti nel corpus.

Pipeline di Clustering

Per il clustering sono stati eseguiti i seguenti passaggi:

1. Generazione di **embeddings** con il modello `SentenceTransformer (thenlper/gte-small)`.
2. Riduzione dimensionale tramite **UMAP**, con $n\_components = 5$ e metrica coseno.
3. Applicazione di **HDBSCAN**, con `min_cluster_size=50` e metrica euclidea, per individuare i cluster.

Pipeline di Topic Modelling

Per il topic modelling è stato utilizzato **BERTopic**, integrando gli stessi modelli di embedding, UMAP e HDBSCAN. In questo modo è stato possibile estrarre i principali argomenti dai testi, mantenendo coerenza con la pipeline di clustering.

Conclusione

Le trasformazioni preliminari e le due pipeline hanno permesso di ottenere una rappresentazione più chiara dei testi del dataset, individuando strutture latenti e argomenti principali all'interno delle notizie false.

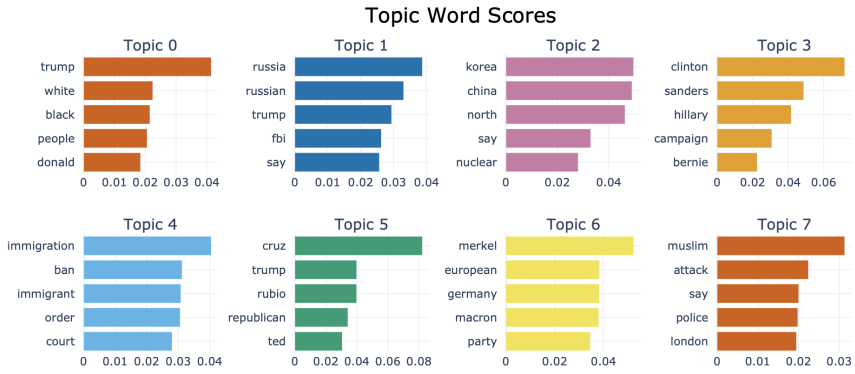

Figure 1: Risultati del laboratorio 4

5 Laboratorio 5

Per la sperimentazione è stato utilizzato il modello `microsoft/Phi-3.5-mini-instruct`, caricato tramite la libreria `transformers`. L'inizializzazione prevede:

- Verifica della disponibilità GPU.
- Caricamento del modello e del tokenizer.

- Creazione di una pipeline di `text-generation`, con parametri di campionamento controllati (`temperature=0.2`, `max_new_tokens=500`).

La generazione della risposta è stata gestita tramite una funzione dedicata, che costruisce il messaggio di input, invoca la pipeline e restituisce l'output pulito del modello.

5.1 Esercizio 5.1

Introduzione

L'obiettivo del laboratorio 5.1 è stato quello di utilizzare un **Large Language Model (LLM)** per assegnare delle **etichette (label)** ai topic emersi nel laboratorio 4. L'idea di fondo è quella di sfruttare le capacità semantiche del modello per produrre etichette descrittive e leggibili a partire dai topic, costituiti da insiemi di parole con pesi associati.

Prompt Engineering

Sono stati progettati due differenti prompt per testare l'efficacia della generazione:

- **Prompt 1:** fornisce al modello un singolo topic e richiede in output un'unica label. Per facilitarne la comprensione, è stato incluso un esempio di input-output desiderato, così da definire chiaramente il formato atteso.
- **Prompt 2:** richiede invece la generazione di una lista di etichette per più topic contemporaneamente. Anche in questo caso è stato inserito un esempio esplicativo con più topic e le rispettive label.

Le tecniche adottate per migliorare la qualità della risposta includono:

1. Inserimento di esempi concreti di input e output.
2. Contestualizzazione della definizione di "topic", per ridurre possibili ambiguità.
3. Specifica del formato di output atteso.

Risultati

I due approcci hanno mostrato differenze interessanti:

- Il **prompt 1** si è dimostrato più stabile, restituendo etichette concise e coerenti.
- Il **prompt 2**, pur fornendo un output utile, ha mostrato maggiore variabilità nella struttura, restando più generale.

In generale, la presenza di esempi e la chiarezza delle istruzioni hanno contribuito in maniera significativa alla qualità dei risultati, confermando l'importanza del *prompt engineering* nell'interazione con LLM.

5.2 Esercizio 5.2

L'obiettivo è la progettazione di un prompt per guidare il modello nell'etichettare le definizioni nell'excel del laboratorio 2 con il termine a cui esse appartengono. A tal fine sono stati progettati due prompt:

- **Prompt per i concetti concreti**: nella progettazione è stata seguita la strategia di prompting passo dopo passo, fornendo al modello una lista di passi da seguire per scegliere il termine corretto. Vengono inoltre utilizzati segni di punteggiatura per delinare l'input e l'output. Infine vengono dati al modello le seguenti informazioni: la lista dei top k termini più frequenti con score associato e un esempio di definizione. Infine abbiamo fornito un esempio di etichettatura. Il modello è riuscito ad etichettare correttamente il termine Microscopio e ad arrivare molto vicino al termine Pantalone, etichettandolo con "Gambale".

- **Prompt per i concetti astratti**: il prompt dei concetti concreti applicato a quelli astratti non ha dato i risultati attesi. In particolare il modello tendeva a concentrarsi troppo sull'esempio fornito, non riuscendo a completare correttamente la task. Pertanto abbiamo rimosso l'esempio e specificato meglio i passi, richiedendo al modello di lavorare in lingua italiana e di restituire un termine astratto sostantivo singolare, scegliendo la forma più comune e generale.

  Il modello è migliorato riuscendo a predire l'etichetta di "Pericolo" e andando molto vicino al termine "Euristica" etichettandolo con "Metodo".