README.md

Relazione del primo gruppo di esercizi

TLN_dicaro_1.1

CONSEGNA:

- Date delle definizioni per quattro concetti (due concreti e due astratti), calcolare la similarità fra di esse.
- Aggregare anche le definizioni secondo le dimensioni di concretezza e specificità e ri-calcolare i punteggi.
- Effettuare del pre-processing se necessario prima del calcolo.

Premessa importante

In questa prima fase abbiamo lavorato individualmente ottenendo due soluzioni coerenti ma diverse. Abbiamo, dunque, deciso di proporle entrambe perché ci sono sembrate egualmente interessanti.

I termini sono:

```
                Generico    Specifico
              ==========================
    Concreto    |   Paper     Sharpener
    Astratto    |   Courage   Apprehension
```

Versione di Roberto Demaria

SVOLGIMENTO:

- Ho importato il file excel sotto forma di data frame per poterne estrarre le informazioni più agevolmente.
- Qui sotto si può vedere la testa del frame dati creato grazie alla libreria pandas:

	Partecipante	Courage	Paper	Apprehension	Sharpener
1	2	Ability to face our own fears and do something...	Material derived from trees and used in sever...	fearful expectation or anticipation	Object used to shapen a pencil
2	3	the ability to face thing without fear	a type of material made from cellulose	A moode where one feel agitation	An object to sharpen a pencil
4	5	Inner strength thaht allow you to face particu...	Product obtained from wood cellulose. It is us...	State of disturbance	Tool used to sharpen pencils
5	6	Ability to control the fear	Flat material made from wood used for writing	Worry about the future	Little object which allow to sharpen a pencil
6	7	Ability to control fear and to be willing to d...	a short piece of writing on a particular subje...	act of understanding something, or the way tha...	tool for making something sharper

- Ho scelto di utilizzare due approcci al calcolo della similarità: uno basato su una funzione di libreria (SequenceMatcher) e uno costruendo io stesso una funzione per il calcolo della similarità.
- In questo secondo caso ho scelto un approccio bag of words e di filtrare le stopwords come fase di pre-processing, per concentrarsi sui termini più salienti.
- Ho costruito delle matrici di similarità fra le definizioni di ogni concetto.
- A partire da tali matrici ho calcolato la similarità generale per i due approcci facendo una media delle similarità escludendo la diagonale.

RISULTATI:

- Si è notato come, nel caso di termini concreti, la similarità sia più elevata di quanto non accada per i termini astratti. Questo è probabilmente dovuto alla possibilità di utilizzare degli attributi visivi per descrivere il termine.
- Nel caso dei termini astratti, invece, la mancanza di questi attributi concreti porta a definizioni meno simili fra di loro.
- I valori di similarità ottenuti con i due approcci sono quantitativamente diversi ma in entrambi i casi si può evincere questa tendenza:

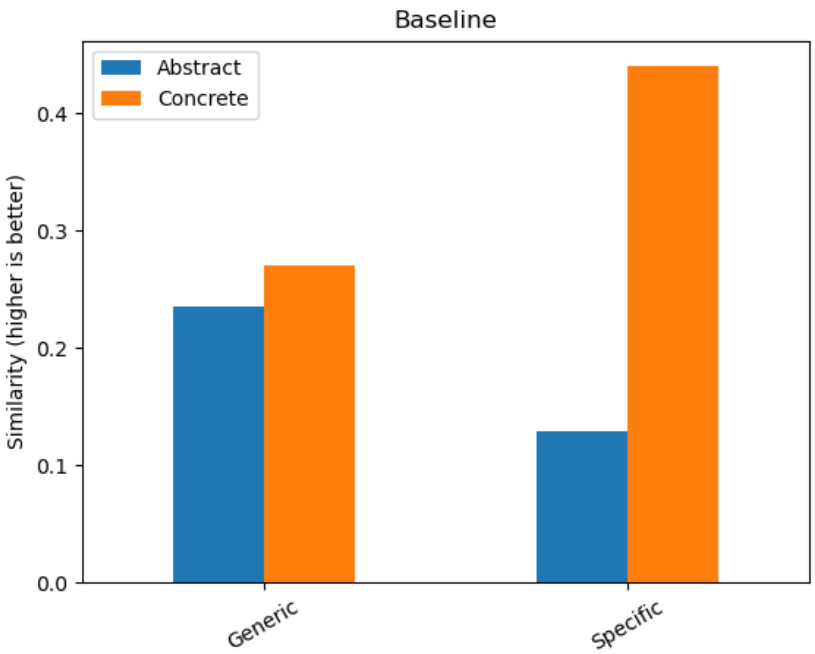
	similar	my_similarity
Paper	0.395611	0.166860
Courage	0.398625	0.137788
Apprehension	0.316237	0.087922
Sharpener	0.520719	0.332071

Versione di Damiano Gianotti

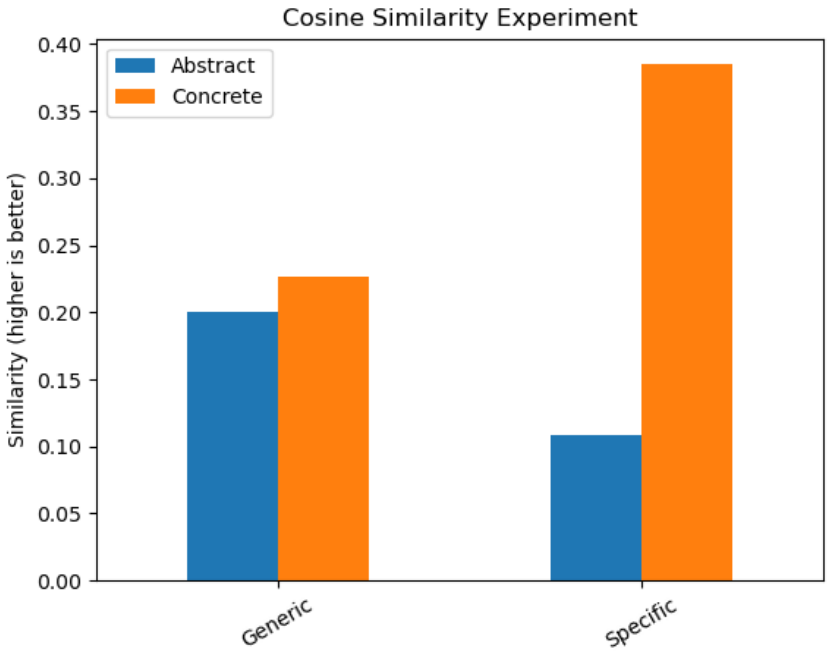
SVOLGIMENTO:

- Si è scelto di filtrare le stopwords come fase di pre-processing, per concentrarsi sui termini salienti.
- Abbiamo provato, oltre la baseline, come misura di similarità la Cosine Similarity andando ad aggregare i risultati

RISULTATI:



- Si è notato come, nel caso di termini concreti, la similarità sia significativamente più elevata di quanto non acccada per i termini astratti. Questo è probabilmente dovuto alla possibilità di utilizzare degli attributi visivi per descrivere il termine.
- Nel caso dei termini astratti, invece, la mancanza di questi attributi concreti porta a definizioni meno simili fra di loro.



TLN_dicaro_1.2

CONSEGNA:

- Dare una spiegazione dei risultati ottenuti nell'esercizio precedente

Versione di Roberto Demaria

SVOLGIMENTO:

- Ho scelto di utilizzare due approcci: il primo approccio usando TF-IDF per vedere se vi fosse un nesso con la rilevanza statistica delle parole mentre il secondo approccio basato semplicemente sulla frequenza delle parole utilizzate nelle definizioni.

RISULTATI:

- Nel primo caso ho calcolato TF-IDF delle parole e li ho sommati per ogni definizione ottenendo una lista di 28 elementi (pari al numero di definizioni date). Dopodiché ho nuovamente sommato questi elementi ottenendo il seguente risultato:

	sum tf_idf	my_similarity
Paper	72.70662	0.166860
Courage	77.31542	0.137788
Apprehension	67.99832	0.087922
Sharpener	67.98177	0.332071

- Quello che si può evincere sembra essere una maggiore rilevanza statistica delle parole per i termini generici rispetto a quelli specifici e di quelli astratti rispetto a quelli concreti. Questo ci dice che i termini generici tendono ad essere meno omogenei nelle definizioni mentre i termini specifici utilizzano più spesso le stesse parole: in particolare sharpener che ha la maggiore similarità coincide anche con il minor valore di tf_idf rivelando che le definizioni sono molto simili avendo gli stessi termini più o meno distribuiti in tutte le definizioni.
- Nel secondo approccio ho fatto alcune analisi sulle frequenze delle parole andando a contare la frequenza totale di ogni parola e poi discriminando quelle con frequenza maggiore di 1 ottenendo diverse misure quantitative da confrontare con la similarità.

	S	N	R	my_similarity
Paper	97	20	4.85	0.166860
Courage	99	25	3.96	0.137788
Apprehension	76	20	3.80	0.087922
Sharpener	104	18	5.78	0.332071

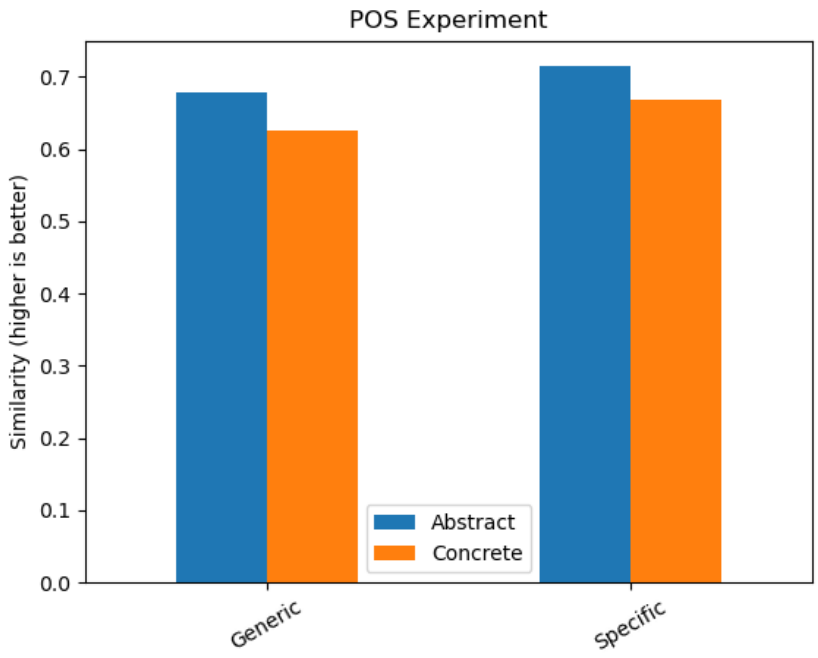
- Legenda:
  - S —> somma delle frequenze maggiori di 1
  - N —> numero di parole con frequenza maggiore di 1
  - R —> rapporto S/N
- I termini concreti presentano un valore di R significativamente superiore rispetto a quelli astratti. I valori di R rispecchiano inoltre l'andamento della similarità.

Versione di Damiano Gianotti

SVOLGIMENTO:

- Ho scelto di utilizzare approccio basato sul pos tagging delle parole
- Calcola la sovrapposizione tra i due set di definizioni pre-elaborate convertite in POS tagging.

RISULTATI:



- Quello che si può evincere sembra essere una maggiore rilevanza statistica delle parole per i termini generici rispetto a quelli specifici e di quelli astratti rispetto a quelli concreti.
- Questo sembra confermare che i termini generici tendono ad essere meno omogenei nelle definizioni mentre i termini specifici utilizzano più spesso le stesse parole:
- In particolare l'oggetto concreto e specifico (sharpener) che ha la maggiore similarità ha un valore di POS nella media, rivelando che le definizioni sono molto simili dal punto di vista del Part to Speach, avendo gli stessi termini più o meno distribuiti in tutte le definizioni.

TLN_dicaro_1.3

CONSEGNA:

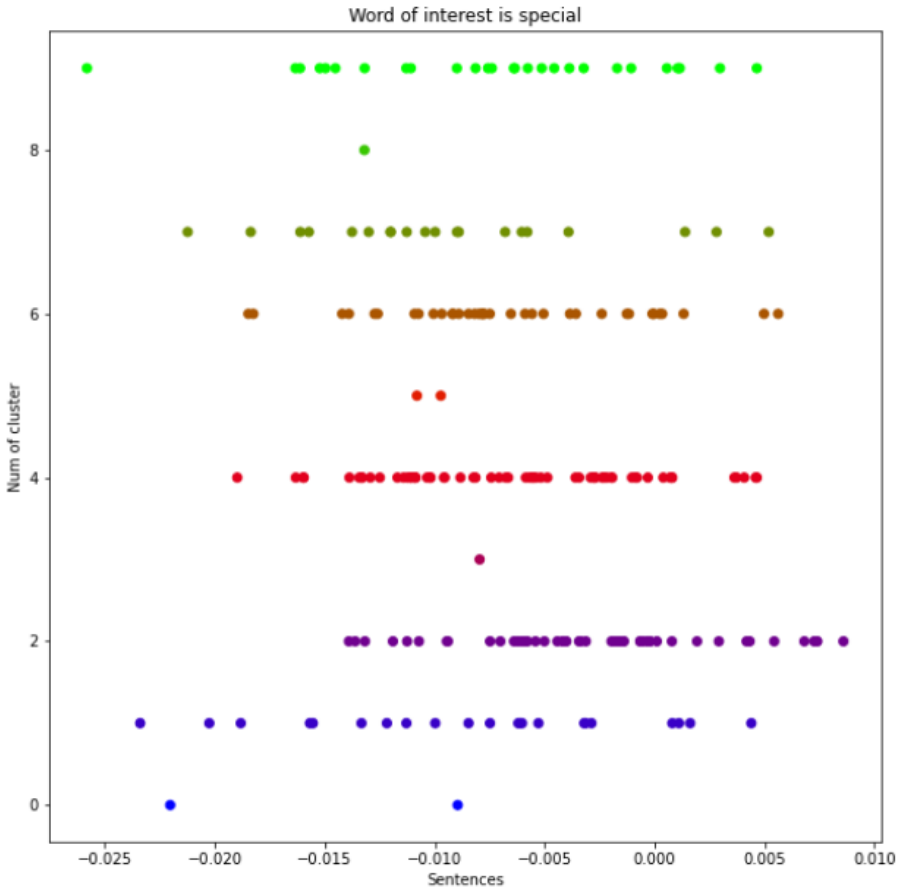
- Implementazione di un semplice sistema di WSI e della pseudo-word evaluation

SVOLGIMENTO:

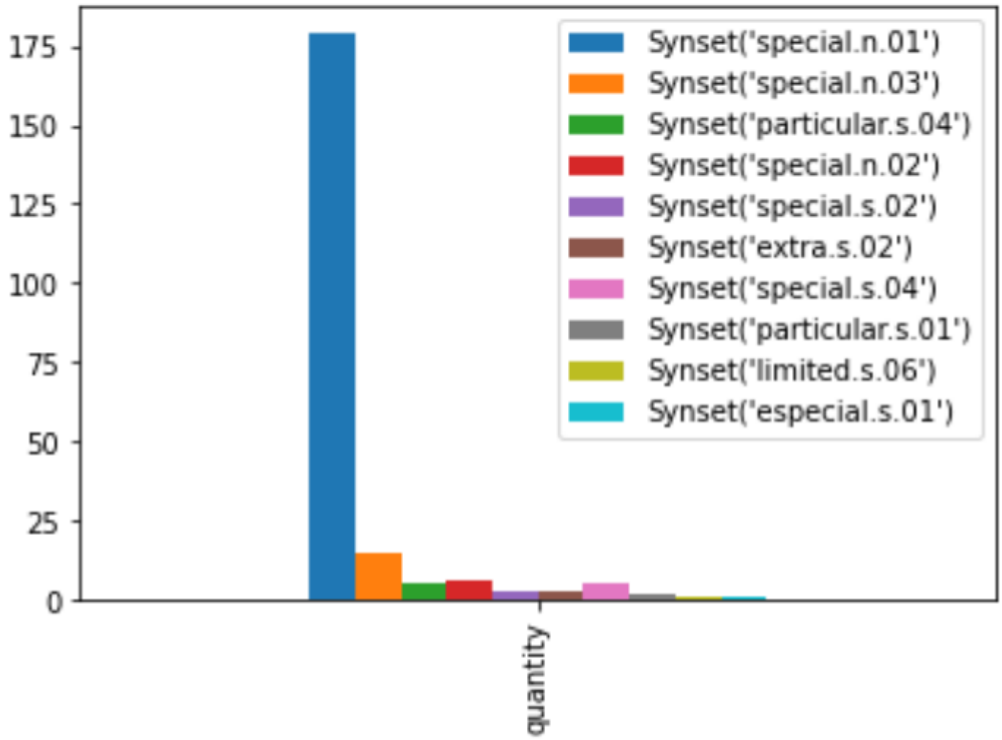
- Abbiamo usato uno spazio pre-costruito in gensin (glove-wiki-gigaword-300); questo serve a darci un embedding per una certa parola dati tutti i suoi possibili contesti.
- Abbiamo usato il brown corpus per scegliere una parola con frequenza tra 50 and 1000. Abbiamo scelto “bar” e “special” come parole di interesse (con frequenza rispettivamente 71 e 233).
- Abbiamo creato un contesto tenendo conto anche delle stop words e un embedding per la parola di interesse.
- Successivamente abbiamo applicato K-Means come algoritmo di cluster non supervisionato per predire i raggruppamenti.Ad esempio nel caso della parola “special” abbiamo scelto 10 clusters basandoci sul numero di synsets di WordNet.
- Possiamo vedere la distribuzione delle parole nei vari clusters

RISULTATI:

Embeddings e wordnet

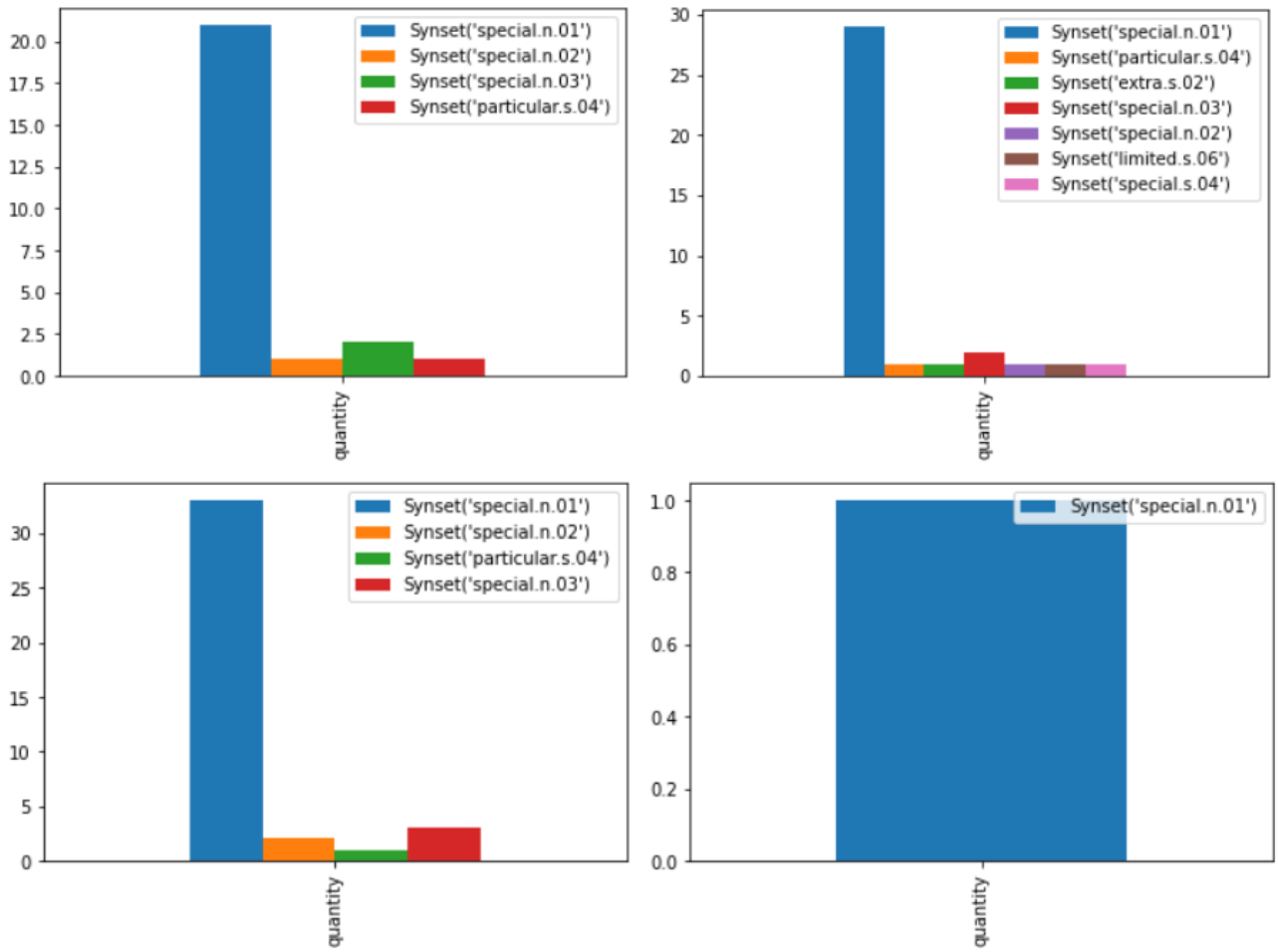


- Abbiamo provato a fare un’interpretazione di questi clusters basandoci sui synsets di WordNet e usando l’algoritmo di Lesk. In particolare applicando Lesk a tutte le frasi vengono effettivamente identificati i 10 possibili sensi




```
Counter({Synset('special.n.01'): 179,
         Synset('special.n.03'): 15,
         Synset('particular.s.04'): 5,
         Synset('special.n.02'): 6,
         Synset('special.s.02'): 3,
         Synset('extra.s.02'): 3,
         Synset('special.s.04'): 5,
         Synset('particular.s.01'): 2,
         Synset('limited.s.06'): 1,
         Synset('especial.s.01'): 1})
```

- Non vi è tuttavia una concordanza tra i cluster identificati con K-Means e i WordNet synsets.
- Questo significa che K-Means non riesce a separare i contesti nello stesso modo di un essere umano poiché i vari clusters non sono associabili ad uno specifico synset di WordNet;
- ogni cluster contiene frasi che possono essere assegnate a più synsets.
- Se ad esempio andiamo a vedere la distribuzione dei synsets per i primi quattro clusters (index 0, 1, 2, 3) possiamo vedere questa incongruenza:



Pseud-words

- Infine abbiamo utilizzato il meccanismo a pseudo-word per vedere se il K-Means riusciva a separare i due contesti.
- Abbiamo perciò unito le parole in “barspecial” e rifatto la stessa analisi applicando poi K-Means con 2 clusters.
- I risultati questa volta sono più incoraggianti:
  - il cluster 0 ha 55 matching per “bar” e 55 matching per “special”
  - il cluster 1 ha 13 matching per “bar” e 165 matching per “special”
- Anche se non perfettamente si può vedere che K-Means riesce ad associare i diversi contesti in maniera abbastanza precisa.
- In particolare, si può vedere dalla disomogeneità del raggruppamento, che il cluster 0 è più associato al contesto di bar mentre il cluster 1 a quello di special.
- Abbiamo anche calcolato le misure statistiche Precision, Recall e F1-score:

Metric	cluster 0	cluster 1
Precision	0.50	0.93
Recall	0.80	0.73
F1-score	0.62	0.82

TLN_dicaro_1.4

CONSEGNA:

- Implementare un sistema basato sulla teoria di Hanks per la costruzione del significato.
- Scelto un verbo transitivo (quindi valenza >= 2), recuperare da un corpus delle istanze in cui viene usato.
- Effettuare il parsing di queste frasi per identificare i supersensi di WordNet associati agli argomenti del verbo (subject e object).

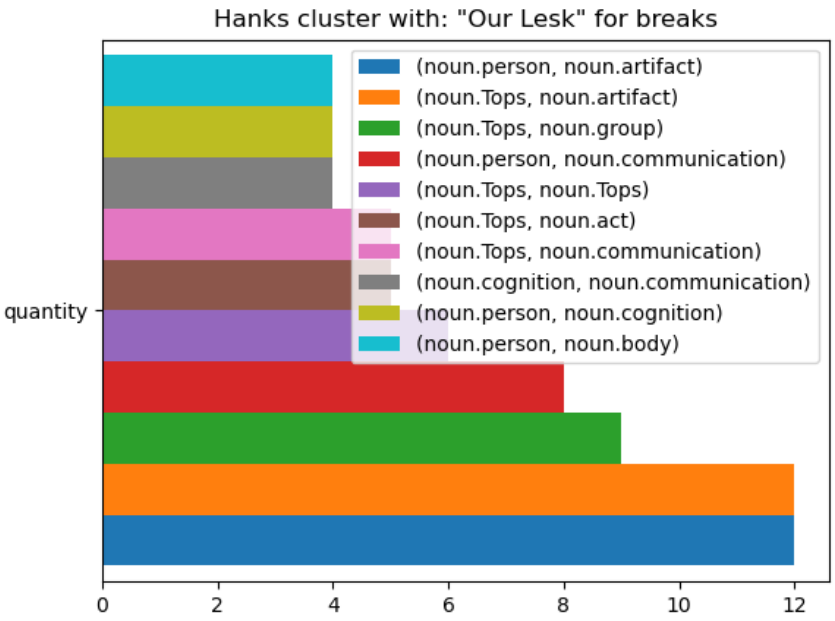
- Calcolare le frequenze di questi supersensi per i due ruoli e stampare le possibili combinazioni.

SVOLGIMENTO:

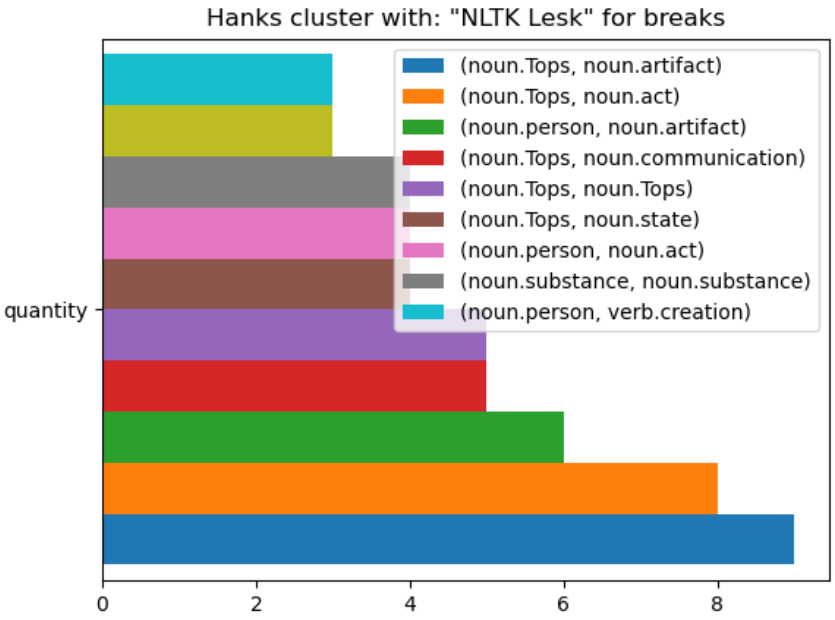
- Si è scelto il verbo 'to break', in particolare il presente terza persona singolare.
- Il corpus utilizzato è Wikipedia, da cui sono state estratte 3000 frasi, usando sketch engine
- Per il parsing a dipendenze si è usata la libreria spaCy.
- Sono state scartate quelle frasi in cui il verbo non presenta entrambi i ruoli richiesti.
- I termini che svolgono i ruoli vengono lemmatizzati e si va poi a calcolare il loro synset migliore tramite WSD (algoritmo di Lesk).
- Nel caso il soggetto sia 'he'/'she', è necessario forzare il suo synset a 'person.n.01' per evitare che venga erroneamente riconosciuto come 'elio', analogamente per 'it' sostituito ad 'artifact.n.01'
- Con questi synset si individua il relativo supersenso `lexname`, andando a calcolare poi frequenze e combinazioni possibili.

RISULTATI:

- Andiamo a creare due grafici con le migliori `k` coppie soggetto-oggetto, usando due versioni differenti dell' algoritmo Lesk



- Questo istogramma a barre rappresenta la quantità delle 10 coppie più frequenti, usando una nostra versione dell' algoritmo di lesk (approccio a bag-of-words)



- Analogamente a quanto detto sopra con la differenza che qui viene utilizzato l' implementazione di Lesk di libreria (nltk)
- Facendo riferimento alla documentazione ufficiale di wordnet e a entrambi i plot possiamo intuire le seguenti cose:
  - la maggior parte delle coppie sono del tipo (nome, nome), eccezion fatta per (noun.person, verb.creation);
  - l' elemento con maggiore frequenza è noun.Tops ovvero "beginner" unico per i nomi
  - come categorie semantiche abbiamo uno sbilanciamento verso person, artifact anche incentivato dalla nostra forzatura
- A fine dell' esperimento, analizzandolo con occhio critico, ci chiediamo se effettivamente i `lexname` di wordnet rappresentino una categoria semantica sensata e se ha una granularità/generalizzazione adeguata
- Dal nostro punto di vista, questo laboratorio è un sofisticato tentativo di rappresentare la valenza del verbo nella sua forma presente (forse la più semplice da trattare) e potrebbe aiutare a classificare o clusterizzare verbi simili. Le cose si complicano se vi vuole espandere il discorso ad altre forme verbali e/o verbi irregolari.

TLN_dicaro_1.5

CONSEGNA:

- Esperimento content-to-form usando i dati dell'esercizio 1.1
- Per ogni concetto, prendere le definizioni a disposizione, cercare in WordNet il synset corretto utilizzando il principio del "genus" per indirizzare la ricerca

SVOLGIMENTO:

- Abbiamo inizialmente verificato che ognuno dei quattro termini avesse almeno un corrispettivo synset su WordNet avendo riscontro positivo.
- Applicando il principio del genus per ogni definizione abbiamo estratto le prime n = 4 parole significative rifacendomi alla teoria che il significato sia maggiormente racchiuso all'inizio della definizione.
- Per ogni parola significativa abbiamo ricercato i rispettivi iponimi.
- Abbiamo inferito il WordNet synset in base alla massima similarità tra le nostre definizioni e quelle associate alle definizioni estratte dagli iponimi

RISULTATI:

- Non in tutti i casi, purtroppo, l'algoritmo è riuscito ad inferire il senso corretto della parola; in alcuni casi abbiamo verificato che il senso corretto non appariva all'interno degli iponimi estratti
- Questi sono i risultati complesssivi ottenuto in termini di inferenza e massima similarità ottenuta:

Concept	max similarity	synset inferred
Paper	0.75	paper.n.01
Courage	0.75	physical_ability.n.01
Apprehension	1.00	apprehension.n.01
Sharpener	0.50	acuminate.v.01

- Come possiamo vedere solo in due casi (paper e apprehension), l'algoritmo riesce ad inferire correttamente il senso a partire dalle definizioni
- Ecco invece la lista dei top 5 sensi che l'algoritmo ci presenta

COURAGE

n°	Name: Courage, dtype: object
0	[property, allows, face]
1	[ability, face, fears]
2	[ability, face, thing]
3	[inner, strength, thaht]
4	[ability, control, fear]

The best word forms for concept are

Score	Synset
0.6666666666666666	Synset('physical_ability.n.01')
0.5	Synset('confront.v.04')
0.2857142857142857	Synset('take_the_bull_by_the_horns.v.01')
0.25	Synset('lee.n.08')
0	Synset('countenance.n.03')

```
max_sym is: 0.75
my wn synset inferred for courage is: Synset('physical_ability.n.01')
The best word forms for courage concept
Score: 0.6666666666666666 for synset: Synset('physical_ability.n.01')
Score: 0.5 for synset: Synset('confront.v.04')
Score: 0.2857142857142857 for synset: Synset('take_the_bull_by_the_horns.v.01')
Score: 0.25 for synset: Synset('lee.n.08')
Score: 0 for synset: Synset('countenance.n.03')
```

```
Computing concept paper
0          [cellulose, material, cut, folded, written]
1    [material, derived, trees, used, several, cont...
2                     [type, material, made, cellulose]
3        [product, obtained, wood, cellulose, ., used]
4          [flat, material, made, wood, used, writing]
Name: Paper, dtype: object
max_sym is: 0.6666666666666666
my wn synset inferred for paper is: Synset('coloring_material.n.01')
The best word forms for paper concept
Score: 0.5 for synset: Synset('coloring_material.n.01')
Score: 0.4444444444444444 for synset: Synset('animal_material.n.01')
Score: 0.4 for synset: Synset('aggregate.n.02')
Score: 0.3333333333333333 for synset: Synset('diethylaminoethyl_cellulose.n.01')
Score: 0.2857142857142857 for synset: Synset('carboxymethyl_cellulose.n.01')
Score: 0.25 for synset: Synset('carboxymethyl_cellulose.n.01')
Score: 0 for synset: Synset('carboxymethyl_cellulose.n.01')
```

```
 Computing concept apprehension
0    [something, strange, causes, strange, feeling,...
1                  [fearful, expectation, anticipation]
2                         [moode, one, feel, agitation]
3                                  [state, disturbance]
4                                       [worry, future]
Name: Apprehension, dtype: object
max_sym is: 1.0
my wn synset inferred for apprehension is: Synset('apprehension.n.01')
The best word forms for apprehension concept
Score: 0.8571428571428571 for synset: Synset('apprehension.n.01')
Score: 0.5714285714285714 for synset: Synset('expectation.n.03')
Score: 0.5 for synset: Synset('expectation.n.03')
Score: 0.4444444444444444 for synset: Synset('emotion.n.01')
Score: 0.4 for synset: Synset('astonishment.n.01')
Score: 0.3333333333333333 for synset: Synset('affection.n.01')
Score: 0.2857142857142857 for synset: Synset('affection.n.01')
Score: 0.25 for synset: Synset('affect.n.01')
Score: 0.2222222222222222 for synset: Synset('affect.n.01')
Score: 0.18181818181818182 for synset: Synset('affect.n.01')
Score: 0 for synset: Synset('affect.n.01')
```

```
 Computing concept sharpener
0    [tool, equipped, blade, allows, sharpen, tip]
1                   [object, used, shapen, pencil]
2                        [object, sharpen, pencil]
3                   [tool, used, sharpen, pencils]
4         [little, object, allow, sharpen, pencil]
Name: Sharpener, dtype: object
max_sym is: 0.7272727272727273
my wn synset inferred for sharpener is: Synset('acuminate.v.01')
The best word forms for sharpener concept
Score: 0.5454545454545454 for synset: Synset('acuminate.v.01')
Score: 0.5 for synset: Synset('drill.n.01')
Score: 0.4 for synset: Synset('cutting_implement.n.01')
Score: 0.2 for synset: Synset('abrader.n.01')
Score: 0.16666666666666666 for synset: Synset('abrader.n.01')
Score: 0 for synset: Synset('abrader.n.01')
```