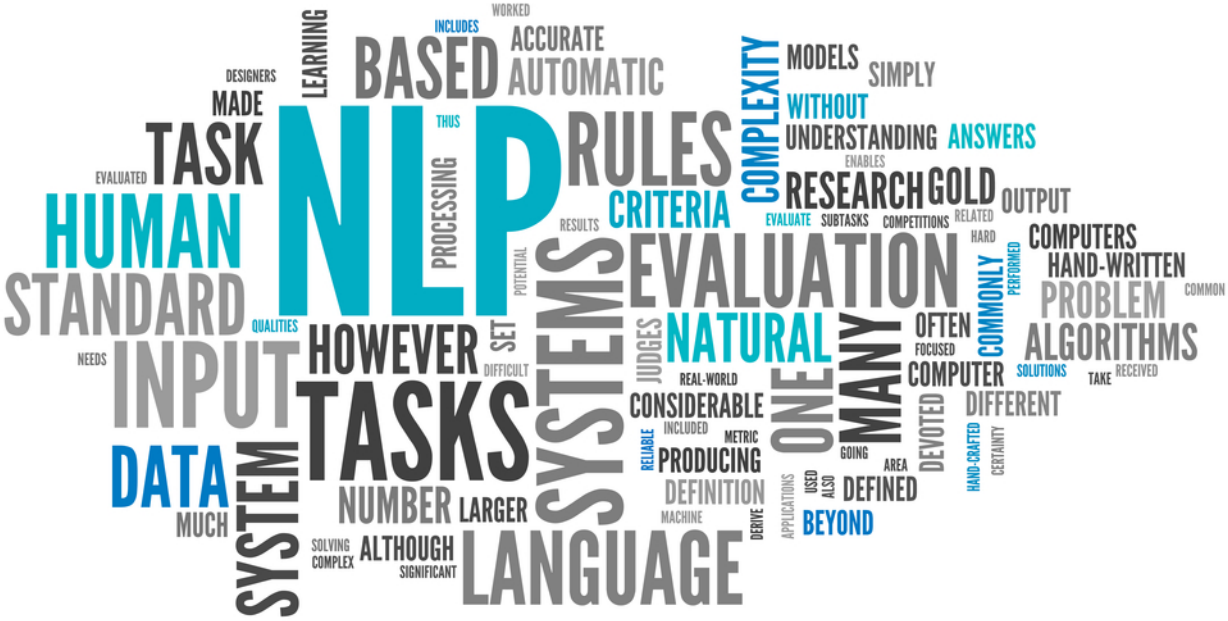
Relazione Progetto TLN

Parte terza - Esercizi svolti

Laura Ventrice



Indice

Esercizio 1: Defs

Svolgimento e risorse utilizzate

Questo esercizio consiste nel calcolare la similarità, vista come sovrapposizione lessicale, tra le definizioni del documento "*definitions.csv*" e successiva aggregazione nelle due dimensioni (concretezza/specificità).
Per lo svolgimento di questo esercizio, l'implementazione è stata suddivisa nelle seguenti fasi:

1. Importazione del documento di definizioni.
2. Preprocessing è stata inserita una fase di espansione degli slang e delle abbreviazioni. Infine sono state eliminate le stopwords ed è stata effettuata la lemmatizzazione. L'obiettivo è stato quello di generare un vocabolario comune a tutte le definizioni di un particolare concetto.
3. Creazione dei "*phrase embeddings*" in one-hot encoding per ogni definizione.
   In particolare per ognuna delle definizioni è stato creato un vettore a partire dal vocabolario di riferimento del concetto, in cui in ogni posizione è stato inserito il valore 1 se la definizione conteneva il token, 0 altrimenti.
4. Implementazione della similarità tra le definizioni di ogni concetto utilizzando la *cosine similarity* tra i phrase embeddings generati precedentemente.
   Infine per ogni concetto è stata calcolata la media delle similarità delle definizioni associate.
5. Generazione di statistiche riguardanti le 5 parole più utilizzate per definire ogni concetto, la lunghezza media delle definizioni per ogni concetto.
6. Aggregazione nelle dimensioni di concetti concreti/astratti e generici/specifici.

L'unica risorsa esterna utilizzata è stato il documento "*slang.txt*" che contiene gli slang per espanderli durante il preprocessing, che è stato estratto dal repository: https://github.com/rishabhverma17/sms_slang_translator

Risultati

Similarità per ogni concetto:

- Emotion: 14.515
- Person: 44.085
- Revenge: 13.418
- Brick: 37.106

Top 5 parole per concetto:

- Emotion
  [('feeling', 12), ('human', 8), ('feel', 8), ('something', 7), ('state', 4)]
- Person
  [('human', 29), ('person', 6), ('living', 4), ('individual', 3), ('certain', 3)]

- Revenge
  [('someone', 14), ('anger', 8), ('feeling', 7), ('action', 6), ('emotion', 6)]
- Brick
  [('used', 24), ('object', 16), ('material', 16), ('construction', 16), ('build', 13)]

Lunghezza media delle definizioni
- Emotion : 4.1
- Person : 3.48
- Revenge : 5.53
- Brick : 5.19

Similarità media per i concetti nelle dimensioni
- Concrete: 0.406
- Abstract: 0.14
- Generic: 0.293
- Specific: 0.253

Esercizio 2: Content2Form

Svolgimento e risorse utilizzate

Questo esercizio consiste nel effettuare la ricerca onomasiologica a partire dalle definizioni disponibili del documento "*definitions.csv*" con l'obiettivo di individuare il concetto che definiscono.

Per lo svolgimento di questo esercizio, l'implementazione è stata suddivisa nelle seguenti fasi:

1. Importazione del documento di definizioni.
2. Preprocessing in cui sono state eliminate le stopwords ed è stata effettuata la lemmatizzazione. Infine è stata aggiunta un'ultima fase di espansione degli slang e delle abbreviazioni. L'obiettivo è stato quello di generare un vocabolario comune a tutte le definizioni di un particolare concetto.
3. Eliminazione delle definizioni troppo corte, in particolare che hanno meno di 3 parole dopo il preprocessing.
4. Implementazione della ricerca onomasiologica utilizzando WordNet in particolare per ogni concetto sono state eseguite le seguenti operazioni:
   A. Estrazione dei 30 token più utilizzati per definire il concetto.
   B. Creazione di un dizionario con l'associazione tra i token estratti precedentemente e le definizioni in cui compare il token.
   C. Utilizzo del concetto di *genus* per individuare il concetto, in particolare per ogni parola frequente è stato individuato il synset utilizzando l'implementazione dell'algoritmo di *lesk* inserendo in input la parola e come contesto ognuna delle definizioni in cui compare. Successivamente viene estratto il synset più comune associato alla parola in base alle definizioni in cui compare. Infine viene definito un insieme di iperonimi a partire dagli iperonimi di ogni synset individuato.
   D. Creazione di una lista di iperonimi con associate le parole più comuni presenti nel contesto. Per ogni iperonimo il contesto viene generato a partire dalla sua definizione e gli esempi di utilizzo inseriti in Wordnet.
   E. Ordinamento degli iperonimi in base al tipo di parole che compaiono nel contesto, in particolare il valore associato viene calcolato nel seguente modo:
      $$score = lenght\_important\_words + count\_top\_5 + count\_top\_10*0.5 + count\_end*(-0.25)$$
      Quindi lo score è in funzione sia del numero di parole più frequenti associate all'iperonimo ma anche al tipo di parole in termini di popolarità.
      Lo scopo di questa definizione di score è stato quello di valorizzare gli iperonimi che contengono le prime 10 parole più frequenti e non prendere in considerazione solo la quantità di parole.
   F. Estrazione dei primi 5 iperonimi.

Risultati

- Emotion
  Top words: ['feeling', 'human', 'feel', 'something', 'state', 'being', 'living', 'concept', 'certain', 'animal', 'sensation', 'mind', 'express', 'emotion', 'mental', 'range', 'situation', 'think', 'make', 'good', 'bad', 'arising', 'form', 'percieve', 'towards', 'others', 'sentiment', 'entity', 'throw', 'word']
  Risultato della ricerca: [('emotional_state.n.01', ['emotion', 'good', 'state']), ('feeling.n.01', ['emotion', 'state', 'feel', 'feeling']), ('idea.n.01', ['good', 'mind', 'think']), ('body.n.01', ['animal', 'being', 'human']), ('being.n.01', ['state', 'being'])]

- Person
  Top words: ['human', 'person', 'certain', 'ability', 'single', 'living', 'homo', 'sapiens', 'individual', 'answer', 'question', 'mean', 'may', 'say', 'generic', 'describe', 'precise', 'feature', 'belonging', 'group', 'society', 'mammal', 'descending', 'ape', 'entity', 'sentient', 'see', 'touch', 'member', 'specie']
  Risultato della ricerca: [('people.n.01', ['group', 'human']), ('person.n.01', ['person', 'human']), ('unit.n.03', ['group']), ('currency.n.01', ['ape']), ('organism.n.01', ['ability', 'living'])]

- Revenge
  Top words: ['someone', 'anger', 'feeling', 'action', 'reaction', 'act', 'something', 'emotion', 'person', 'hurting', 'done', 'bad', 'towards', 'wrong', 'negative', 'consequence', 'resulting', 'generally', 'arising', 'another', 'way', 'hurt', 'revenge', 'return', 'damaging', 'usually', 'wrongdoing', 'describes', 'classified', 'good']
  Risultato della ricerca: [('return.n.10', ['return', 'good', 'act', 'action']), ('resistance.n.01', ['act', 'action', 'something', 'feeling']), ('pain.n.02', ['emotion', 'feeling']), ('feeling.n.01', ['emotion', 'feeling']), ('quality.n.01', ['someone', 'something'])]

- Brick
  Top words: ['used', 'object', 'construction', 'material', 'build', 'building', 'made', 'clay', 'block', 'something', 'usually', 'brick', 'house', 'constructing', 'element', 'like', 'piece', 'shape', 'wall', 'aim', 'basic', 'parallelepiped', 'tool', 'resistnat', 'polygonal', 'different', 'size', 'red', 'generally', 'cunstruction']
  Risultato della ricerca: [('building_material.n.01', ['building', 'build', 'material', 'used', 'constructing']), ('implement.n.01', ['piece', 'used', 'tool']), ('ceramic.n.01', ['material', 'made']), ('artifact.n.01', ['made', 'object']), ('filler.n.01', ['used'])]

Esercizio 3: Hanks

Svolgimento e risorse utilizzate

Questo esercizio consiste nell'applicazione della teoria di Hanks a partire dall'individuazione di un verbo transitivo che abbia almeno 2 argomenti. Il verbo scelto per lo svolgimento di questo esercizio è *gestire*, in inglese *handle* che è un verbo polisemico che ha diversi tipi di soggetti e oggetti.
Per lo svolgimento di questo esercizio, l'implementazione è stato suddiviso nelle seguenti fasi:

1. Estrazione di circa 900 frasi che contengono il verbo *handle*, a partire da un corpus di articoli di Medium.
2. Individuazione di soggetto e oggetto del verbo, in ognuna delle frasi estratte, utilizzando l'albero a dipendenze generato da SpaCy.
3. Individuazione del synset per ogni soggetto e oggetto utilizzando l'implementazione dell'algoritmo di Lesk di nltk, inserendo in input anche la frase in cui compare la parola e la Part Of Speech individuata da SpaCy al fine di limitare la ricerca del synset.
4. Individuazione del supersenso per ogni oggetto e soggetto utilizzando la funzione *lexname* di WordNet in nltk.
5. Combinazione dei significati dei verbi seguendo la teoria di Hanks.
6. Individuazione del synset del verbo handle nelle frasi con l'algoritmo di Lesk e infine creazione dei cluster semantici.

Il corpus utilizzando in questo esercizio è presente al seguente link:
https://www.kaggle.com/datasets/fabiochiusano/medium-articles?resource=download

Risultati

('noun.artifact', 'noun.communication') 12
('noun.communication', 'noun.communication') 10
('noun.act', 'noun.communication') 10
('noun.person', 'noun.communication') 9
('noun.communication', 'noun.cognition') 8
('noun.artifact', 'noun.act') 7
('noun.group', 'noun.attribute') 7
('noun.person', 'noun.artifact') 7
('noun.act', 'noun.cognition') 6
('noun.person', 'noun.location') 6
('noun.cognition', 'noun.artifact') 6
('noun.act', 'noun.act') 6
('noun.group', 'noun.cognition') 5
('noun.person', 'noun.act') 5
('noun.cognition', 'noun.cognition') 5
('noun.group', 'noun.act') 5

Esercizio 4: Segmentation

Svolgimento e risorse utilizzate

Questo esercizio consiste nell'implementazione di un algoritmo di text segmentation. I paragrafi utilizzati per testarlo sono stati i sommari delle pagine Wikipedia dei seguenti argomenti: "New York City", "Machine Learning", "Vincent van Gogh" e "Cubism".
Per lo svolgimento di questo esercizio, l'implementazione è stata suddivisa nelle seguenti fasi:

1. Generazione del corpus, in particolare utilizzando la libreria *wikipediaapi* per l'estrazione dei sommari degli argomenti scelti.
2. Preprocessing effettuando la lemmatizzazione e eliminando le stopwords. Infine sono stati eliminati tutti i token che non sono di tipo funzionale. Successivamente sono stati creati degli embedding per ogni frase, con la frequenza delle parole presenti.
3. Individuazione delle parole che non sono significative, in particolare che sono troppo frequenti o troppo poco.
4. Implementazione dell'algoritmo di text segmentation partendo dal testo da dover suddividere e le suddivisioni eque nel testo per evitare di avere bias durante la generazione dei segmenti.
   L'algoritmo a partire dalla segmentazione fornita calcola la **coesione intra-gruppo** nei segmenti come la media della cosine similarity delle frasi rispetto al centroide del segmento , successivamente ricerca i **punti di bassa coesione** (break point) effettuando il tentativo si spostamento della suddivisione del segmento di una frase verso destra e verso sinistra, calcolando di conseguenza la coesione intra-gruppo.
   Se nei tentativi di suddivisione si generano dei segmenti che hanno una coesione maggiore rispetto a quelli iniziali, si modifica la suddivisione dei segmenti.
   L'algoritmo termina quando non ci sono più cambiamenti nella suddivisione dei segmenti.

Risultati

La suddivisione dovrebbe essere nelle righe 19, 35 e 58. I risultati ottenuti sono [30, 41, 60].

Esercizio 5: TM-TV

Svolgimento e risorse utilizzate

Questo esercizio consiste nello svolgimento di Topic Modeling e Topic Visualization a partire da un corpus a scelta. Il corpus utilizzato per questo esercizio è il "*fetch_20newsgroup*" disponibile tramite la libreria *sklearn*, in cui sono presenti circa 10k documenti di newsgroup suddivisi in 20 sotto-argomenti.
Per lo svolgimento di questo esercizio, l'implementazione è stata suddivisa nelle seguenti fasi:

1. Caricamento del corpus tramite la libreria sklearn con suddivisione tra testi e e target.
2. Preprocessing con lemmatizzazione, eliminazione di stopwords e mantenimento di token con PoS funzionale.
3. Creazione del modello generando il dizionario con le statistiche dei token all'interno di ogni documento, filtrando i token in base alla loro frequenza. Successivamente ogni documento è stato convertito in una versione bag-of-words e infine è stato creato il modello dei topic utilizzando il modello **Latent Dirichlet Allocation** disponibile tramite la libreria Gensim.
4. Visualizzazione del modello.
5. Valutazione del modello utilizzando la Perplexity e la Coherence del modello. La prima calcola il limite di verosimiglianza per parola, utilizzando un gruppo di documenti come corpus di valutazione.
6. Analisi dei risultati prendendo in considerazione i target associati nel corpus.
7. Implementazione di Topic Visualization utilizzando le WordCloud.

Risultati

La generazione dei topic a partire dai documenti genera un bag-of-words riconducibile ai topic reali dei documenti, in particolare:

- **TM**: 0.033*"people" + 0.020*"way" + 0.019*"time" + 0.019*"government" + 0.017*"law" **Real Topic**: Politics
- **TM**: 0.045*"game" + 0.036*"drive" + 0.035*"year" + 0.032*"time" + 0.027*"soon"
  Real Topic: Sport
- **TM**: 0.027*"people" + 0.019*"time" + 0.018*"God" + 0.018*"information" + 0.017*"group"
  Real Topic: Religion
- **TM**: 0.025*"thing" + 0.021*"year" + 0.021*"time" + 0.020*"people" + 0.019*"week"
  Real Topic: Science
- **TM**: 0.034*"team" + 0.030*"year" + 0.029*"time" + 0.028*"probably" + 0.028*"car"
  Real Topic: Autos/motorcycles
- **TM**: 0.064*"problem" + 0.034*"window" + 0.027*"thank" + 0.025*"file" + 0.022*"question"
  Real Topic: Computer Science
- **TM**: 0.032*"system" + 0.028*"year" + 0.027*"time" + 0.021*"question" + 0.021*"thing"
  Real Topic: Operating System
- **TM**: 0.041*"thank" + 0.031*"card" + 0.023*"driver" + 0.019*"line" + 0.018*"chip"
  Real Topic: Eletronics - Hardware

Esercizio 6: letsplay - Frutta con la "M"

Svolgimento e risorse utilizzate

Questo esercizio consiste nel partire da una categoria, in questo caso la frutta, e una lettera e creare un gioco nel quale il sistema crea una combinazione random di input e sfida l'utente a trovare per primo un elemento della categoria che inizia per la lettera estratta.
Il sistema implementato si basa sulla risorsa semantica di WordNet.
Per lo svolgimento di questo esercizio, l'implementazione è stata suddivisa nelle seguenti fasi:

1. Individuazione del synset corretto da cui partire per la ricerca. Nello specifico partendo con la categoria frutta, la traduzione in "fruit" è polisemico, quindi l'idea è stata quella di sfruttare l'Open Multilingual Wordnet generato dall'allineamento di WordNet in inglese con altre risorse. In questo modo dalla parola "frutta" l'unico synset corretto è stato *Synset('fruit.n.01')* con la definizione: *the ripened reproductive body of a seed plant.*
2. Generazione della chiusura per iponimi a partire dal synset individuato di frutta, in modo da ottenere tutto il sottoalbero che ha come radice *Synset('fruit.n.01').*
3. Eliminazione degli elementi che non sono considerati come frutta in ambito di alimentazione in particolare legumi, semi, baccelli e spighe. Per farlo è stata utilizzata la funzione *lowest_common_hypernyms* tra gli elementi e i synset delle categorie, individuati durante l'implementazione.
   Infine sono stati eliminati gli elementi che avevano un antonimo, perchè indicavano una frutta con un aggettivo e non sarebbero stati sufficientemente specifici per lo scopo del gioco.
4. Implementazione del gioco, in particolare viene scelta la lettera casualmente, e successivamente per individuare il frutto da parte del sistema, viene generata la chiusura per iponimi, vengono eliminate le categorie non utili e a partire dalla lista generata vengono mantenuti solo gli elementi che iniziano per la lettera estratta.
   Se l'utente risponde con un frutto non contenuto nella lista il sistema stampa una frutta casuale a partire dalla lista, altrimenti vince l'utente.

Risultati

```
Welcome to the game Fruit with M..

Rules:
Starting with a randomly chosen letter, I challenge you to search for a fruit that begins with the letter drawn.
Ready to be torn?


Let's start!

Tell me a fruit starting with letter ... N
Your answer is:  nice_apple

I found a fruit before you!
The correct answer is:  nutmeg_melon  Def:  seedless orange enclosing a small secondary fruit at the apex
Bye bye!
```

```
Welcome to the game Fruit with M..

Rules:
Starting with a randomly chosen letter, I challenge you to search for a fruit that begins with the letter drawn.
Ready to be torn?


Let's start!

Tell me a fruit starting with letter ... A
Your answer is:  ananas

You win!
Bye bye!
```

Esercizio 7: letsplay - False-Friends

Svolgimento e risorse utilizzate

Questo esercizio consiste nell'estrazione di parole che sono simili come caratteri ma con significati differenti, anche chiamati *false-friends*. L'esercizio è stato svolto generando le coppie a partire dalla lingua italiana e inglese.
Per lo svolgimento di questo esercizio, l'implementazione è stata suddivisa nelle seguenti fasi:

1. Estrazione di lemmi in italiano e inglese a partire da due corpus.
2. Estrazione delle coppie che hanno **similarità grafica**, in particolare sono state utilizzate la distanza di hamming per individuare le parole che hanno una similarità alta nelle prime 3 lettere, inoltre se sufficientemente simili viene calcolata una somiglianza in termini di inserimenti/eliminazioni normalizzata nell'intervallo [0, 1].
   Successivamente viene calcolata la **similarità semantica** utilizzando la risorsa WordNet, in particolare estraendo i synsets delle parole nelle due lingue, e calcolando la *wup_similarity* per ogni coppia di synset. La similarità sarà pari al valore medio delle similarità dei synsets. Se la similarità semantica sarà sufficientemente bassa verrà mantenuta la coppia come false-friends.
3. Confronto con estrazione utilizzando i word-embedding di FastText, l'obiettivo è stato quello di vedere i risultati ottenibili con delle strutture dati più complesse in termini di semantica, utilizzando come misura di similarità la *cosine similarity*.

I corpus utilizzati sono stati per l'italiano Morph-It! e per l'inglese The British National Corpus (BNC) estratto da WordSmith Tools (https://lexically.net/wordsmith/support/lemma_lists.html).

Esercizio 8: FCA - Formal Content Analysis

Svolgimento e risorse utilizzate

Questo esercizio consiste nell'implementazione di un sistema di Ontology Learning utilizzando la Formal Content Analysis. L'argomento scelto per lo svolgimento dell'esercizio sono gli animali, con 8 features riguardanti il loro habitat e altre caratteristiche.
Per lo svolgimento di questo esercizio, l'implementazione è stata suddivisa nelle seguenti fasi:

1. Caricamento dei dati.
2. Interrogazioni, in particolare riguardanti le proprietà comuni degli oggetti o gli oggetti comuni delle proprietà (derivazione), e ottenere la coppia di oggetti o proprietà (concetti formali) più vicina.

Risultati

	Forest	Savana	Acquatic (salt)	Freshwater	Viviparous	Oviparius	Cold-blood	Warm-blood
Panda	x				x			x
Koala	x				x			x
Giraffe	x	x			x			x
Jellyfish			x			x	x	
Sea turtles			x			x	x	
Octopus			x			x	x	
Frog				x		x	x	
Hippopotamus				x	x			x
Penguins			x			x		x
Lion		x			x			x
Crocodile		x	x			x	x	
Hare	x				x			x
Lizard	x					x	x