Università degli studi di Torino
Dipartimento di informatica

Progetto di Tecnologie del Linguaggio Naturale

Implementazione di un sistema di topic modelling

Autori:
Christian Quaranta
Giulia Coucourde
Leonardo Magliolo

Anno Accademico 2022/2023

Introduzione

L'obiettivo del progetto è identificare in modo non supervisionato i principali temi presenti in grandi collezioni di documenti (Topic Modelling).
Il progetto consiste nell'implementazione di un modello Latent Dirichlet Allocation (LDA) mediante libreria Gensim per l'elaborazione del linguaggio naturale.
LDA è stato scelto come modello per affrontare il task di topic modelling per la sua efficacia, scalabilità, interpretabilità e applicabilità in contesti non supervisionati.
Gensim è stato scelto per la sua versatilità nell'NLP e semplicità di utilizzo, ci ha permesso di sperimentare diverse configurazioni e parametri per migliorare le prestazioni.
La relazione illustrerà l'approccio metodologico, l'implementazione del modello LDA tramite Gensim e i risultati ottenuti su dataset di prova, dimostrando l'efficacia del sistema nella scoperta dei temi in modo coerente e veloce.

Progettazione

Pre-processing:

E' stato scelto di attuare in fase di pre-processing la rimozione delle stopwords dal corpus utilizzato per l'addestramento del modello LDA, e successivamente sono state rimosse le parole non classificate dal POS-tagger come nomi; ciò è stato fatto per aumentare la precisione del modello e rendere più interpretabili le tematiche computate.

Latent Dirichlet Allocation:

LDA è un modello statistico generativo utilizzato per scoprire argomenti nascosti all'interno di grandi collezioni di testi. Il processo generativo dell'LDA assume che ogni documento sia una mescolanza di vari argomenti e ogni argomento sia una distribuzione di probabilità sui termini del vocabolario. L'obiettivo dell'LDA è quindi di stimare queste distribuzioni di probabilità a partire dai dati osservati (i documenti).

Il processo di generazione dei dati nel modello LDA può essere sintetizzato in:
Per ogni documento:

- Genera una distribuzione di probabilità sugli argomenti.
- Per ogni parola del documento:
    - Genera un argomento da questa distribuzione.
    - Genera una parola da questa distribuzione di probabilità dei termini per l'argomento selezionato.

L'obiettivo dell'addestramento dell'LDA è stimare i parametri che massimizzano la probabilità di osservare i dati, ovvero i documenti effettivamente presenti nel corpus.

L'addestramento avviene generalmente utilizzando l'algoritmo di inferenza di Gibbs o altre tecniche di ottimizzazione. Durante questo processo, il modello LDA cerca di assegnare in modo ottimale gli argomenti ai termini e ai documenti in modo da massimizzare la verosimiglianza dei dati osservati.

Una volta addestrato, il modello LDA può essere utilizzato per assegnare argomenti ai nuovi documenti, identificare i termini più rilevanti per ciascun argomento e, in generale, fornire una rappresentazione interpretabile delle strutture tematiche presenti nel corpus di testi.

Vengono elencati di seguito parametri e iper-parametri del modello LDA:
Parametri dell'LDA:

- Distribuzione di probabilità degli argomenti per documento: Questo parametro rappresenta la mescolanza degli argomenti all'interno di ciascun documento. È una distribuzione di probabilità sui K argomenti, dove K è il numero di argomenti specificato come iper-parametro.
- Distribuzione di probabilità dei termini per argomento: Questo parametro rappresenta la distribuzione di probabilità dei termini all'interno di ciascun argomento. Indica quali termini sono più rappresentativi per un argomento specifico.

Iper-parametri dell'LDA:

- Numero di argomenti (K): Questo è il principale iper-parametro dell'LDA e specifica il numero di argomenti che vogliamo scoprire nel corpus. Il valore di K deve essere scelto a priori, ed è importante trovare un valore appropriato per ottenere risultati significativi.
- Alpha (α): L'iper-parametro α controlla la distribuzione di probabilità degli argomenti per documento. Un valore più alto di α renderà i documenti più concentrati su un numero limitato di argomenti, mentre un valore più basso di α renderà i documenti più distribuiti tra tutti gli argomenti. Tipicamente, si utilizza un valore basso di α per avere una distribuzione più uniforme degli argomenti per i documenti.
- Beta (β): L'iper-parametro β controlla la distribuzione di probabilità dei termini per argomento. Un valore più alto di β renderà gli argomenti più concentrati su un numero limitato di termini, mentre un valore più basso di β renderà gli argomenti più distribuiti tra tutti i termini. Tipicamente, si utilizza un valore basso di β per avere una distribuzione più uniforme dei termini per gli argomenti.
- Iterazioni: Questo iper-parametro specifica il numero di iterazioni o epoche durante il processo di addestramento. Un valore maggiore di iterazioni può migliorare la convergenza del modello, ma aumenta anche il tempo di addestramento.

Implementazione

In aggiunta alle librerie native di Python, sono state utilizzate:

- Numpy e Pandas: Per rappresentare sotto forma matriciale i dataset.
  L'utilizzo di Numpy e Pandas ha permesso di ridurre i tempi di computazione delle soluzioni e di uniformare l'implementazione ad uno dei più comuni standard di programmazione per sistemi di AI in Python.
- NLTK: Utilizzata in fase di pre-elaborazione per effettuare le operazioni di tokenization, rimozione delle stopwords e POS-tagging.
- Gensim: Utilizzata per generare i modelli LDA a partire da un corpus e per valutare la bontà dai modelli mediante perplexity e coherence index.
- Matplotlib: Utilizzata per produrre automaticamente grafici utili al debug e all'analisi dei dati.
- PyLDAvis: Utilizzata per produrre un'Intertopic Distance Map via scaling multidimensionale dei modelli LDA computati per mezzo di Gensim

Dataset

Il dataset di riferimento utilizzato è New York Times Annotated Corpus [1] contenente oltre 1,8 milioni di articoli scritti e pubblicati dal New York Times tra il 1° gennaio 1987 e il 19 giugno 2007 con i metadati degli articoli forniti dalla New York Times Newsroom, dal New York Times Indexing Service e dalla produzione online personale di nytimes.com.

Il corpus comprende:

- Oltre 1,8 milioni di articoli (esclusi gli articoli delle agenzie di stampa apparsi durante il periodo considerato).
- Oltre 650.000 riassunti di articoli scritti da scienziati bibliotecari.
- Oltre 1.500.000 articoli contrassegnati manualmente da scienziati bibliotecari con tag tratti da un vocabolario di indicizzazione normalizzato di persone, organizzazioni, luoghi e descrittori di argomenti.
- Oltre 275.000 articoli codificati algoritmicamente che sono stati verificati manualmente dallo staff di produzione online su nytimes.com.

Date le dimensioni del dataset, abbiamo optato per l'utilizzo di una porzione ridotta di esso. Abbiamo implementato funzioni che hanno estratto gli 'abstracts' bilanciandoli in base alle diverse categorie di 'Section_name' (indicata all'interno dell'implementazione come 'label' o 'category'). Questo approccio ha permesso l'analisi di testi appartenenti a macro-argomenti in maniera equilibrata.

Test delle prestazioni

Metriche utilizzate:

- Perplexity: misura quanto bene il modello LDA è in grado di predire i documenti nel corpus di addestramento.
  Rappresenta una misura della "confusione" del modello nell'assegnare argomenti ai documenti.
  Un valore di perplexity più basso indica una migliore capacità predittiva del modello, l'obiettivo è ridurla durante l'addestramento dell'LDA.
- Coherence index: misura la coerenza semantica degli argomenti scoperti dal modello.
  Un punteggio di coerenza più alto indica una migliore coerenza semantica degli argomenti, l'obiettivo è massimizzare il punteggio per ottenere argomenti più significativi e interpretabili.
- Trade-off formula: Come osservato sperimentalmente dai dati computati a partire dai modelli calcolati dall'implementazione, aumentando il numero di topics assegnati a LDA la perplexity diminuisce a discapito di una diminuzione del coherence index.
  Al fine di individuare il modello avente 'numero di topics' che minimizzassero la perplexity e massimizzassero il coherence index simultaneamente è stato utile considerare la seguente procedura:
  - Vengono normalizzate le due metriche in modo tale che il valor minimo in modulo venga associato a 0 e il massimo in modulo ad 1, mediante la seguente funzione:

    ```
    def normalize_value_array(array):
        min_val = np.min(np.abs(array))
        max_val = np.max(np.abs(array))
        return (np.abs(array) - min_val) / (max_val - min_val)
    ```

  - Vengono infine moltiplicate le due metriche normalizzate per ottenere un indice relativo, compreso tra 0 ed 1 che stabilisce quanto bene il modello massimizzi il modulo di entrambe le metriche rispetto agli altri modelli aventi 'numero di topics' differenti.

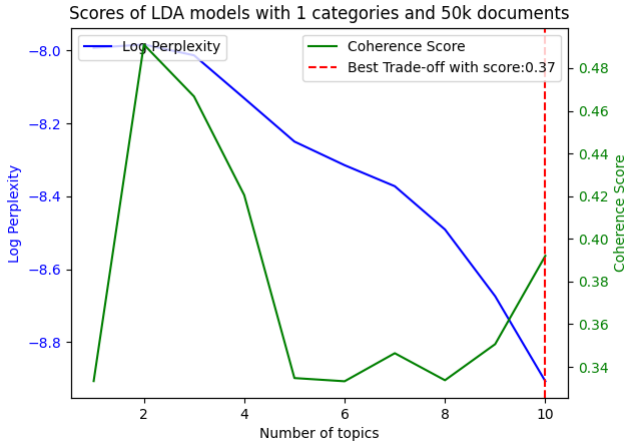
Considerato il breve tempo di computazione per una ridotta quantità di abstract (~ 4 minuti nel caso peggiore per 50.000 abstract) su macchina aventi le seguenti specifiche hardware:

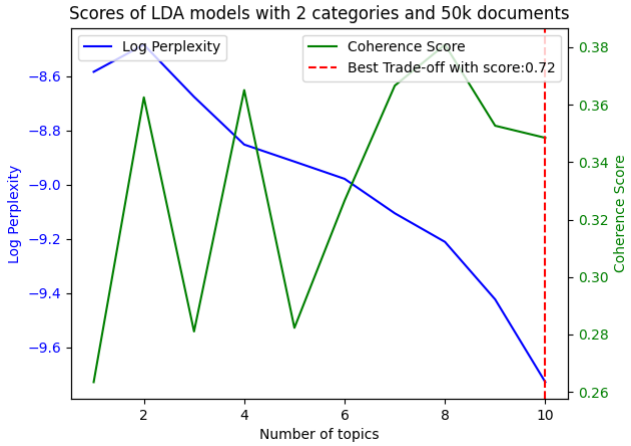
- CPU: AMD Ryzen 7 5800X
- RAM: 32 GByte, DDR4, 3600 MHz

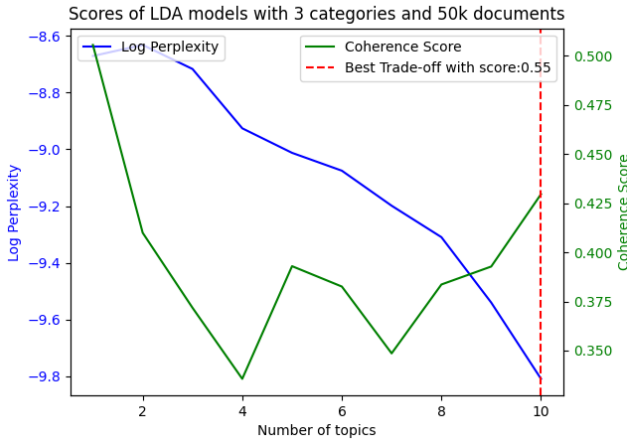
E’ stato possibile computare perplexity, coherence index e trade-off score per modelli generati a partire da datasets di 50.000 abstracts, facendo variare il numero di labels tra 1 e 10 e facendo variare l’iper-parametro ‘numero di topics’ tra 1 e 10.
Tutti i 100 modelli generati presentano i medesimi parametri di LDA di Gensim:

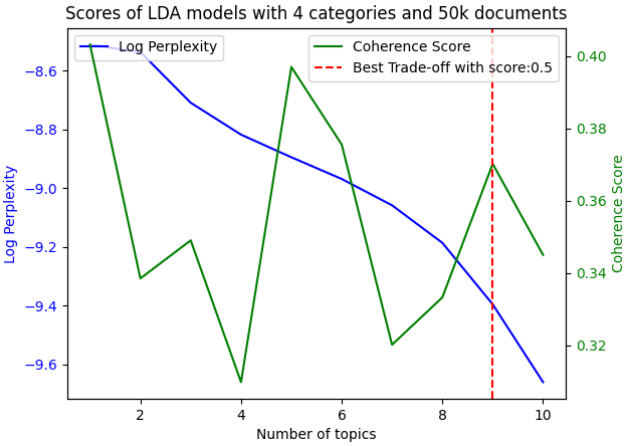
- Random_state: 100
- Update_every: 1
- Chunksize: 10
- Passes: 10 (è stato osservato che aumentare i numeri di passi non è stato in grado di aumentare significativamente le prestazioni del modello)
- Alpha: “Auto”

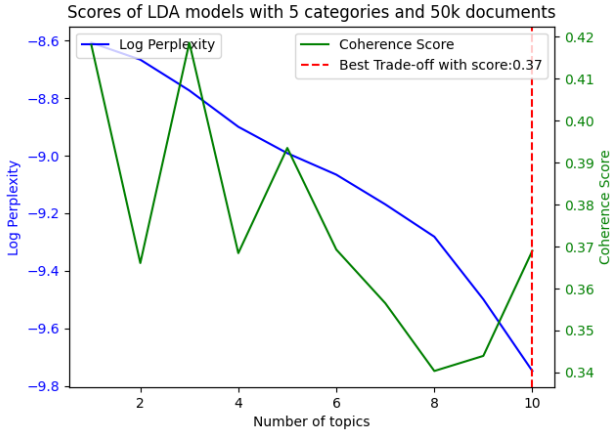
Vengono riportati di seguito i grafici ottenuti che mostrano i valori di perplexity e coherence score in relazione al numero di labels e topics:

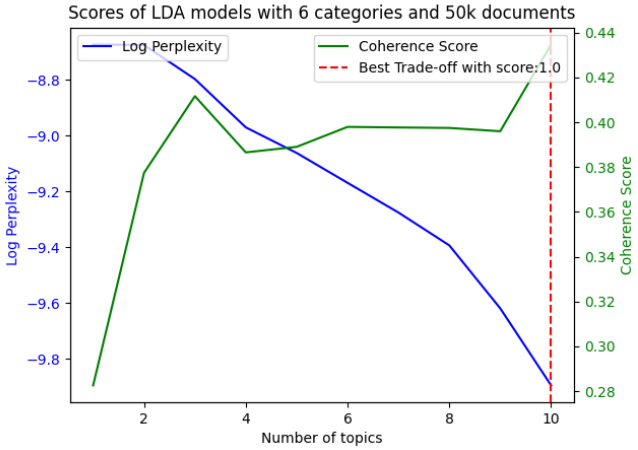


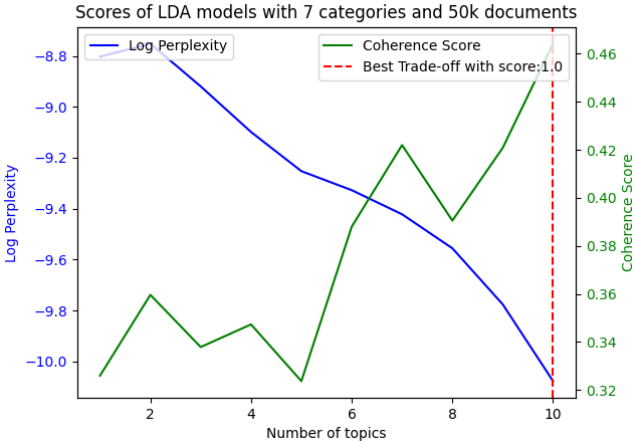


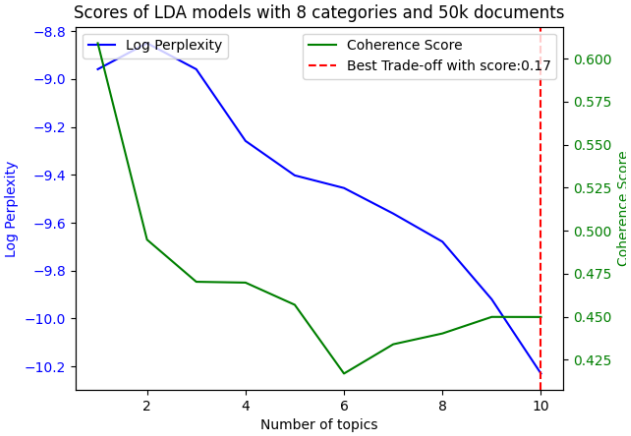


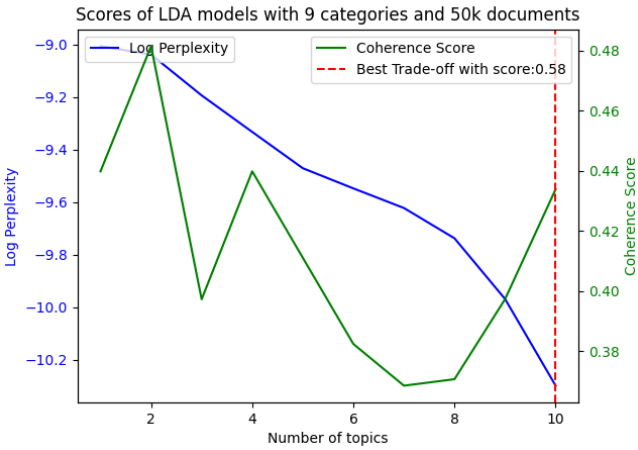


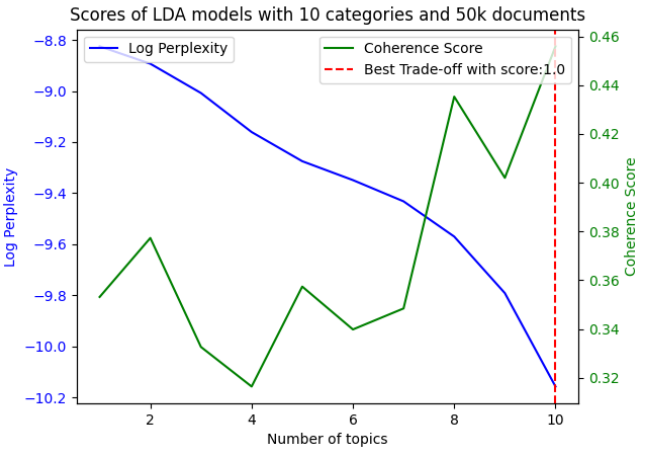

Scores of LDA models with 5 categories and 50k documents
Log Perplexity
Coherence Score
Best Trade-off with score:0.37
Number of topics


Scores of LDA models with 6 categories and 50k documents
Log Perplexity
Coherence Score
Best Trade-off with score:1.0
Number of topics

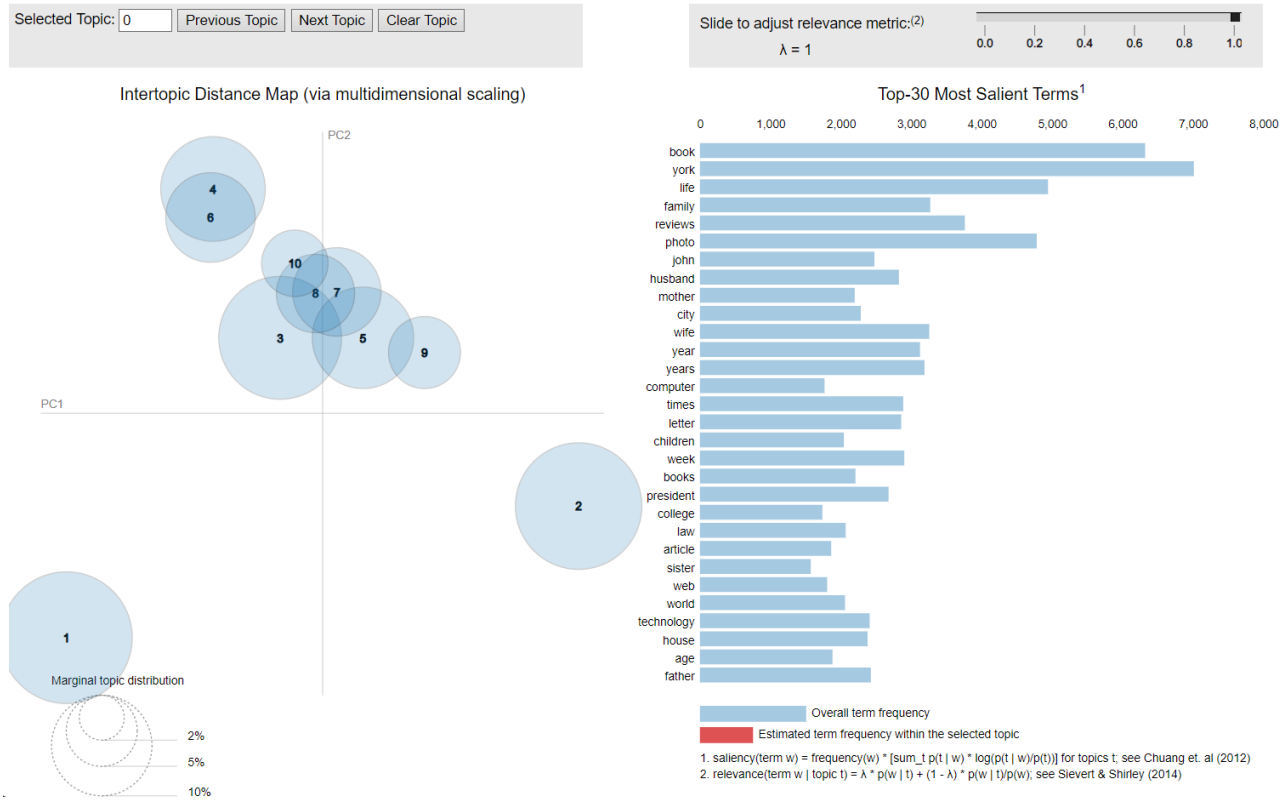

Scores of LDA models with 7 categories and 50k documents
Log Perplexity
Coherence Score
Best Trade-off with score:1.0
Number of topics


Scores of LDA models with 8 categories and 50k documents
Log Perplexity
Coherence Score
Best Trade-off with score:0.17
Number of topics


Scores of LDA models with 9 categories and 50k documents
Log Perplexity
Coherence Score
Best Trade-off with score:0.58
Number of topics


Scores of LDA models with 10 categories and 50k documents
Log Perplexity
Coherence Score
Best Trade-off with score:1.0
Number of topics

Viene riportato di seguito, per il modello avente prodotto i risultati più convincenti (10 categorie e 10 topics) l'Intertopic Distance Map generata per mezzo di PyLDAvis:



Viene indicata, infine, la rappresentazione testuale delle categorie ottenute dal modello sopracitato:

Categorie: ['Business Day', 'New York', 'U.S.', 'Opinion', 'World', 'Sports', 'Arts', 'Archives', 'Books', 'Technology']

Modello: [(0, '0.030*"world" + 0.020*"internet" + 0.017*"michael" + 0.017*"people" + 0.016*"work" + 0.012*"lives" + 0.011*"fiction" + 0.010*"director" + 0.010*"mark" + 0.009*"random"'),
(1, '0.124*"book" + 0.074*"reviews" + 0.040*"children" + 0.037*"article" + 0.028*"author" + 0.025*"board" + 0.024*"man" + 0.023*"daughter" + 0.023*"member" + 0.022*"story"'),
(2, '0.032*"wife" + 0.029*"letter" + 0.026*"president" + 0.023*"home" + 0.022*"service" + 0.021*"services" + 0.018*"st" + 0.015*"james" + 0.014*"church" + 0.014*"ny"'),
(3, '0.098*"family" + 0.066*"mother" + 0.047*"sister" + 0.043*"david" + 0.038*"grandmother" + 0.036*"june" + 0.031*"version" + 0.027*"memory" + 0.022*"april" + 0.019*"education"'),
(4, '0.034*"books" + 0.032*"law" + 0.029*"age" + 0.025*"ages" + 0.024*"woman" + 0.021*"novel" + 0.019*"friends" + 0.016*"william" + 0.012*"joseph" + 0.012*"rankings"'),
(5, '0.073*"york" + 0.033*"year" + 0.030*"times" + 0.025*"technology" + 0.025*"house" + 0.022*"brother" + 0.021*"university" + 0.019*"street" + 0.015*"march" + 0.014*"day"'),
(6, '0.058*"city" + 0.044*"college" + 0.028*"power" + 0.026*"system" + 0.025*"health" + 0.013*"records" + 0.013*"access" + 0.013*"sea" + 0.011*"rock" + 0.010*"artists"'),
(7, '0.043*"photo" + 0.029*"years" + 0.026*"week" + 0.022*"father" + 0.019*"school" + 0.015*"war" + 0.015*"time" + 0.014*"friend" + 0.014*"death" + 0.014*"photos"'),
(8, '0.088*"john" + 0.064*"computer" + 0.044*"richard" + 0.040*"robert" + 0.034*"boy" + 0.030*"sales" + 0.028*"software" + 0.025*"stephen" + 0.023*"game" + 0.020*"editor"'),
(9, '0.100*"life" + 0.056*"husband" + 0.037*"web" + 0.031*"weeks" + 0.030*"july" + 0.025*"site" + 0.021*"th" + 0.021*"sites" + 0.017*"mass" + 0.016*"computers"')].

Analisi dei risultati

E' stato possibile riscontare come i valori di perplexity e coherence index siano variati al variare del numero di topics assegnati al modello LDA: conformemente a come segnalato dalla letteratura riguardante il topic modelling si osserva una relazione inversamente proprozionale tra il numero di topics e la perplexity calcolata sul modello.

Nonostante siano state fatte prove con più dataset aventi numero di categorie differenti non c'è stato un riscontro netto circa la crescita del coherence index in prossimità del numero di categorie associato, ciò potrebbe esser spiegato considerando che

- Sebbene le categorie indicate dal NYTimes vengano considerate come distinte, esse possono condividere tematiche in comune (come nel caso di 'Arts' e 'Books'), oppure alcune categorie specifiche possono essere incluse in alcune più generali (come nel caso di 'New York' e 'U.S.').
- Topics latenti aggiuntivi possono essere presenti nei dati, in aggiunta a quelli segnalati dalle categorie degli abstract.

Fonti:

1. Evan Sandhaus. The New York Times Annotated Corpus: https://catalog.ldc.upenn.edu/LDC2008T19.