AI vLab - Software Heritage

Serena D’Onofrio - 1 Giugno 2023

Outline

- Software Heritage
- Transformers e Bert
- Dataset di licenze
- Grafo degli autori

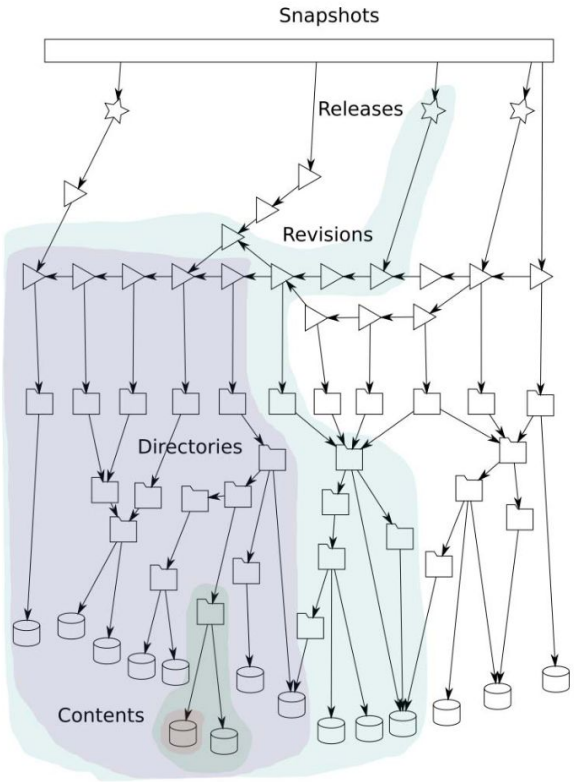
Software Heritage

Archivio universale del codice con mirror in costruzione in ENEA.

Contiene tutto il software esistente, organizzato nella struttura di un **grafo diretto aciclico** (DAG).

Obiettivi:

- Costruzione di dataset a partire dall'archivio (API, architettura di copia di SH).
- Analisi di dataset già costruiti dal team di SH.

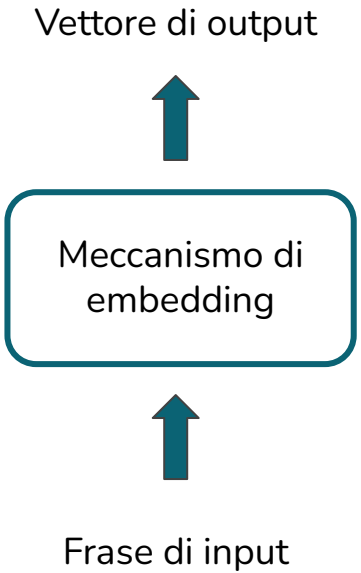


Embedding

Dataset di tipo testo: bisogna trasformarlo in un vettore n-dimensionale, senza perdere informazioni, in modo tale da poter lavorare con algoritmi di ML.

Esistono diversi tipi di embedding che permettono la rappresentazione di parole e frasi in un vettore numerico.

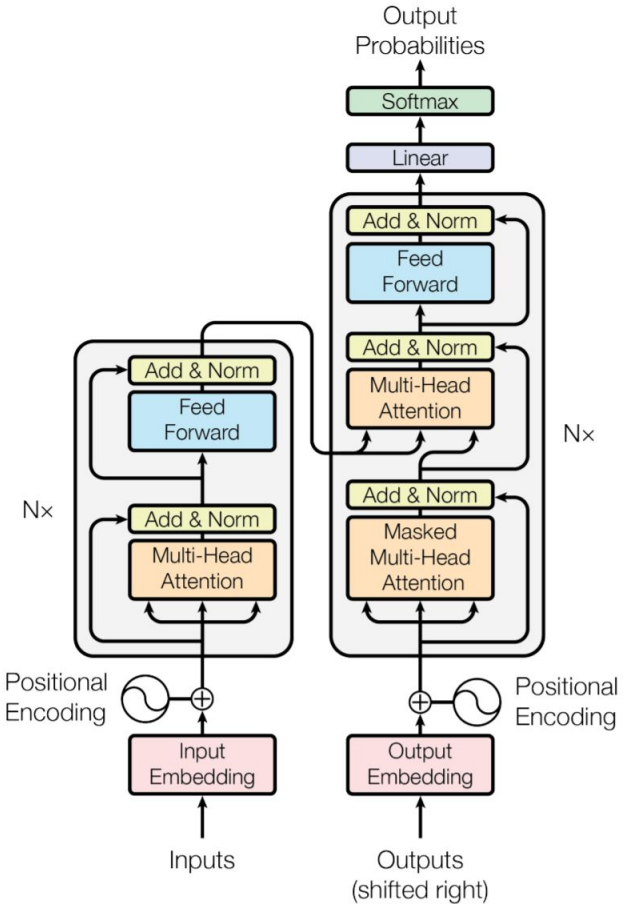
Il nostro primo approccio è basato su **BERT: Bidirectional Encoder Representation from Transformers.**



Transformers

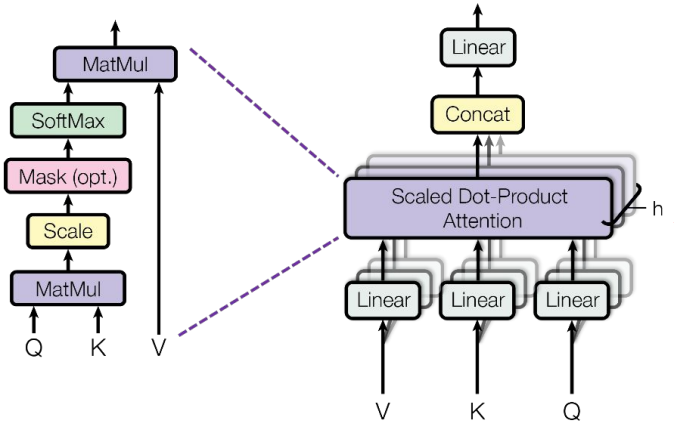
Transformers: necessità di evitare il vanishing gradient e di aumentare la velocità di training.

- Struttura encoder-decoder.
- L'input viene passato parallelamente.
- Il **Positional Encoding** (deterministico o learnable) permette di comprendere l'ordine dei dati di input.
- Il meccanismo della **Self-Attention** permette di collegare tra di loro dati lontani.

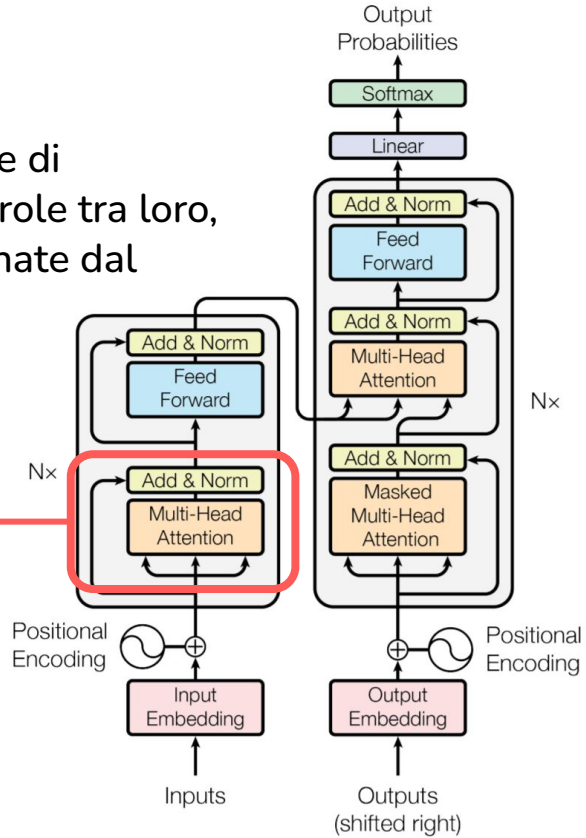


Self-Attention

Il meccanismo di Self-Attention permette di produrre un vettore di rappresentazione che tiene conto del rapporto che hanno le parole tra loro, attraverso prodotti tra matrici che vengono continuamente trainate dal sistema.

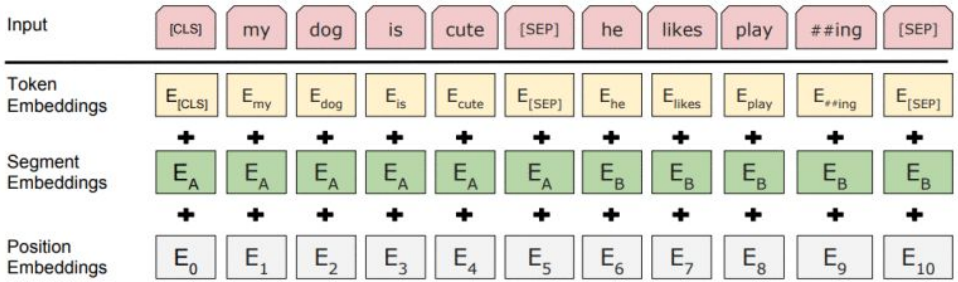


$$\text{attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$



BERT

BERT produce una rappresentazione del linguaggio basata sul contesto globale.

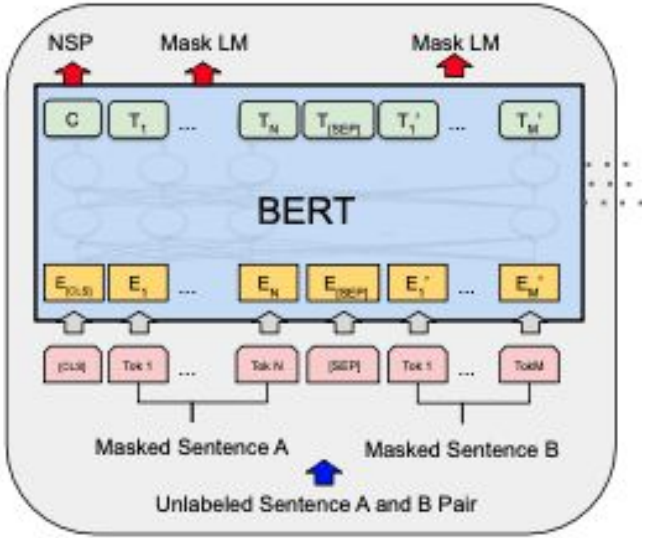


- **Struttura**: Transformer encoder.
- **Input e preprocessing**: due sequenze di token modificate in modo da avere un [CLS] iniziale e un [SEP] alla fine di ogni sequenza.
  Token embedding: rappresentazione vettoriale di ogni singolo token.
  Sequence embedding: distinzione tra le diverse sequenze.
  Positional embedding: identifica la posizione dei token e viene trainato.

BERT

Strategie di training per perfezionare la comprensione del contesto dei token

- **Masked LM:** vengono mascherati il 15% dei token per migliorare la predizione a partire dal contesto.
- **Next Sentence Prediction:** le sequenze consecutive vengono separate il 50% delle volte, per insegnare alla rete a comprendere le relazioni tra due singole frasi.

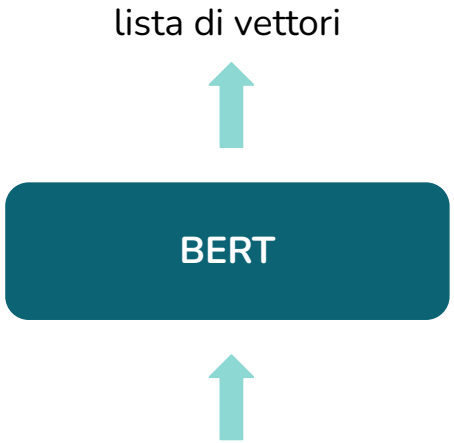


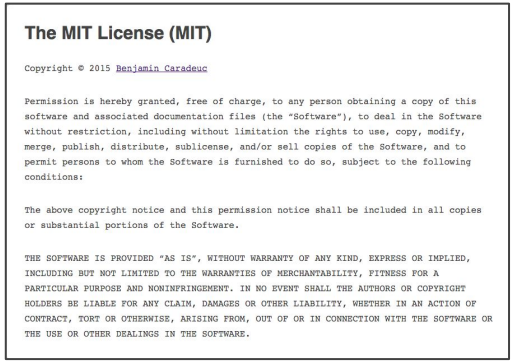
Dataset Licenze

Dataset: 8 milioni di file di testo che contengono licenze (6800 labellati).

Goal: classificazione dei file e riconoscimento delle loro caratteristiche attraverso tecniche di unsupervised learning.

Primo step: vettorizzazione dei file di testo tramite BERT. L'embedding avviene frase per frase. La lista di vettori viene resa un vettore unico.





The MIT License (MIT)

Copyright © 2015 Benjamin Caradeuc

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

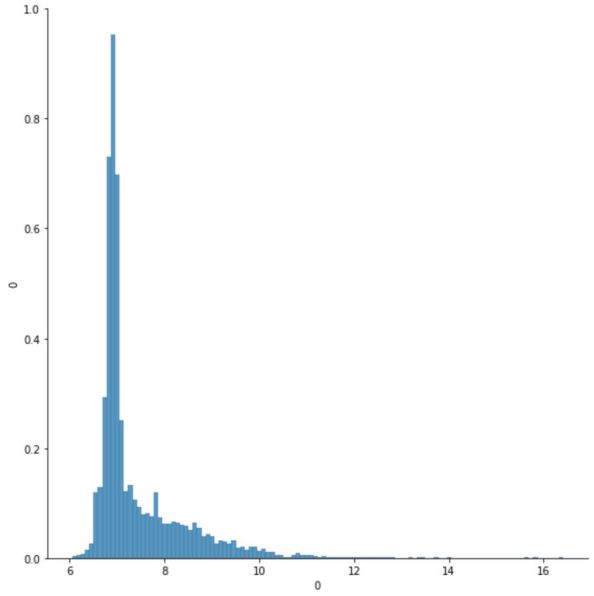
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Dataset Licenze

Vogliamo applicare algoritmi di clustering per identificare i file che hanno caratteristiche comuni. Bisogna definire una giusta “distanza” che permetta di definire una relazione tra dati simili tra di loro.

$$\text{cosine similarity} = S_C(A, B) := \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|}$$



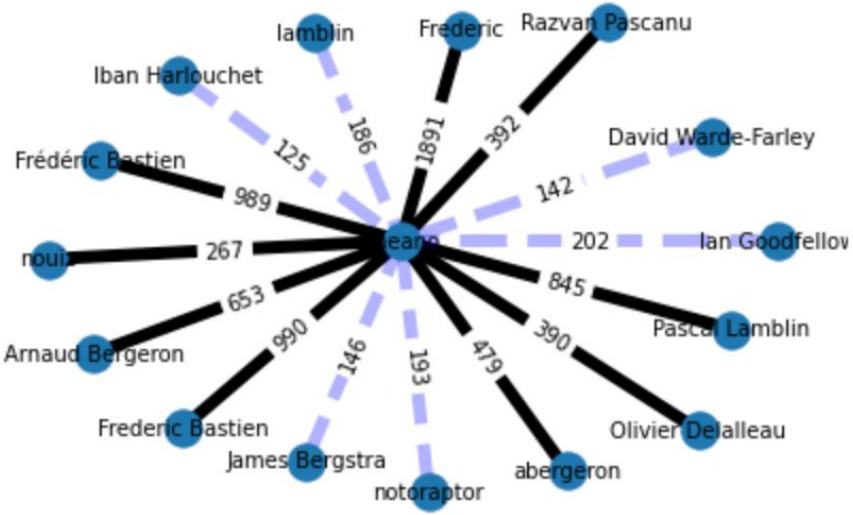
Distribuzione della norma dei vettori licenza

Grafo degli autori

Creazione di un **dataset degli autori** del software per effettuare studi storici su SH.

Partenza dalla repository di Theano *https://github.com/Theano/Theano*

- API di SH
- Environment creato da Marcello Artioli che permette di comprendere meglio la struttura di SH ed interrogarla tramite MySQL.

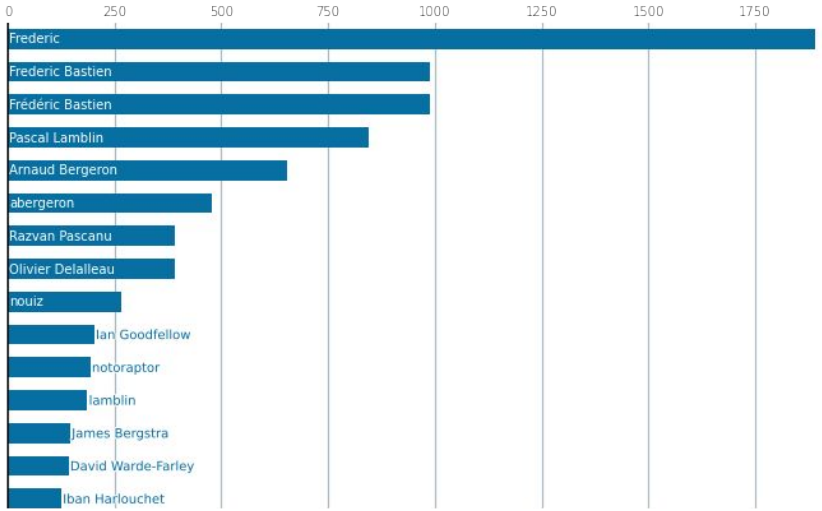


Grafo pesato dei top 15 committer di Theano

Grafo degli autori

Cosa possiamo fare con questa tipologia di dataset?

Analisi storica e sociale del **grafo degli autori** per capire la sua evoluzione nel tempo, attraverso strumenti di persistenza omologica e di teoria dei grafi.

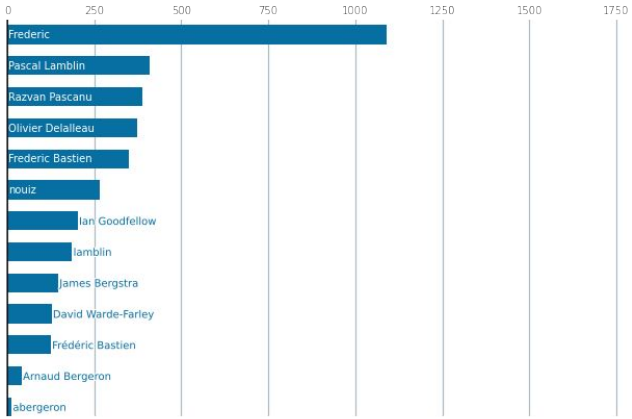


Top 15 committers di Theano

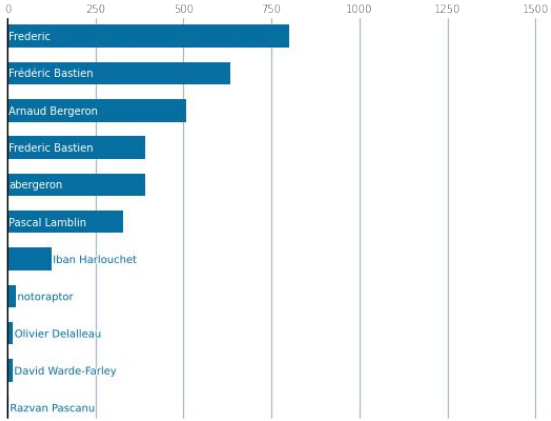
Grafo degli autori

Evoluzione temporale dei 15 top committer del repository di Theano

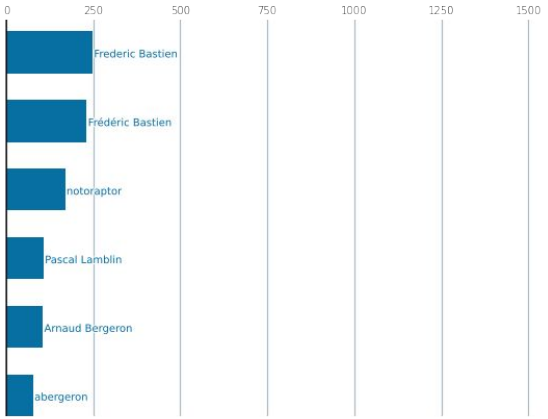
2011-2013



2011-2017



2017-2022



Prossimi step

Classificazione di Licenze

- Definire una distanza tra licenze nello spazio di embedding.
- Provare ad applicare la PCA per avere una rappresentazione più compressa delle licenze.
- Arrivare alla clusterizzazione completa e migliorare l'embedding di Bert.

Grafo degli autori

- Valutare non solo il numero di commit ma anche la loro importanza in termini di spazio disco.
- Analizzare grafi più complessi che interessano più repository insieme.
- Usare GNN per clusterizzare le repository in base alla vicinanza tra gli autori.