Tweets Sentiment Analysis

MapReduce e LingPipe

Liuzzo Antonino Mauro

Sentiment Analysis

- Obiettivo Sentiment Analysis: effettuare l'analisi del sentimento (Sentiment Analysis) su dei tweet riguardante certe "entità" in base alla presenza di alcune parole chiave. In questo modo vedremo "cosa pensano" gli utenti di Twitter rispetto alle varie entità considerate.
- Con il termine Sentiment Analysis ci si riferisce all'utilizzo di tecniche di Natural Language Processing, Analisi Testuale e Linguistica Computazionale per identificare, estrarre e studiare informazioni di natura soggettiva proveniente da diverse fonti.
- L'analisi del sentimento è ampiamente applicate per analizzare social media per una varietà di applicazioni, dal marketing al servizio clienti.
- Gli approcci esistenti all'analisi del sentimento possono essere raggruppati in 4 categorie principali: Spotting di parole chiave; Affinità lessicale; Metodi statistici e Tecniche di livello concettuale.
- In questo studio ci concentreremo sulla prima categoria.

Entità

- Tra le parole che identificano le Entità sono presenti:
  - Nomi e Nickname Twitter di aziende famose
  - Nomi e Nickname Twitter di streamer Twitch famosi
  - Nomi e Nickname Twitter di marchi famosi
  - Nomi e Nickname Twitter di celebrità
- In totale il dataset contiene circa 5400 entità

Dataset

- È stato utilizzato un dataset contenenti 1,600,000 tweet senza emoticon.
- Per semplicità sono stati utilizzati tweet scritti in inglese.

MapReduce

- Per analizzare collezioni di dati grandi dimensioni in maniera efficiente, è una buona opzione utilizzare il framework MapReduce.
- MapReduce consente di effettuare computazione distribuita su una grande quantità di dati utilizzando cluster di computer.
- MapReduce è ottimizzato per Job di tipo batch.
- Il framework libera il programmatore dall'obbligo di scrivere codice parallelo. MapReduce si occupa di dividere i dati tra i nodi del cluster e di gestire i fallimenti dei nodi.
- MapReduce nasconde al programmatore tutte le complessità dovute alla gestione del sistema distribuito.
- Utilizzando MapReduce la condivisione dei dati è eliminata. Ogni nodo elabora una porzione indipendente dei dati in ingresso.
- Il compito principale del programmatore è quello di adattare il problema in modo che esso possa risolto tramite uno o più job MapReduce.
- Per l'elaborato è stata utilizzato Apache Hadoop per il calcolo distribuito e LingPipe per l'analisi del sentimento.

Struttura Catena Job MapReduce

- L'obiettivo è quello di vedere "cosa pensano" gli utenti rispetto ad una determinata entità. Per effettuare ciò si ricorre ad una catena di job MapReduce.
- 2 Job Mapreduce:
  - Il primo job verifica se i tweet citano una o più entità.
  - Il secondo job valuta i tweet relativi ad un'entità e calcola un punteggio che rispecchia il sentimento "medio" degli utenti.

Primo Job MapReduce

- Il primo job è composto solamente dal Mapper. Il Mapper prende in ingresso prende in ingresso i tweet e verifica se tali tweet contengono una o più keyword relative alle entità.
- Il Mapper prende in input delle coppie chiave-valore. La chiave è di tipo Object (non viene utilizzata), mentre il valore è di tipo Text
- Per ogni tweet , viene letta ogni parola e si vede se essa è una keyword. In caso affermativo, il mapper scrive sull'output una coppia chiave-valore in cui la chiave è data dalla keyword, mentre il valore è dato dal tweet.
- L'output è dato da coppie chiave-valore, entrambe sono di tipo Text.

- Esempi:

I hate the new Windows update → Windows — I hate the new Windows update
H&M t-shirts are awesome → H&M — H&M t-shirts are awesome

Secondo Job MapReduce: Mapper

- Il Mapper del secondo job prende in ingresso l'output del primo job MapReduce. Il Mapper utilizza un classificatore per classificare i tweet letti come "pos" (positivo) e "neg" (negativo). Il Mapper assegna un punteggio pari a 1 se il tweet è positivo o un punteggio pari a -1 se il tweet è negativo.
- Viene utilizzato un classificatore basato su LingPipe, addestrato offline tramite un dataset contenente 100.000 tweet etichettati come positivo o negativo.
- Il classificatore così addestrato ha una precisione di circa il 66,8%.
- L'output è composto da coppie di chiave-valore. La chiave è di tipo Text, mentre il valore è di Tipo IntWritable.

- Esempi:

Windows	I hate the new Windows update	→	Windows	-1
H&M	H&M t-shirts are awesome		H&M	1

Secondo Job MapReduce: Combiner

- Il Combiner è un'ottimizzazione che permettere di combinare le coppie chiave-valore date in output dal Mapper prima di inviarle al Reducer. Ciò permette di ridurre i dati che vengono trasferiti al Reducer.
- Non è garantito che il Combiner venga eseguito, pertanto l'algoritmo non deve dipendere dalla sua esecuzione.
- Possiamo vedere il Combiner come una "riduzione locale". Se il tipo di operazione lo consente (l'operazione deve essere commutativa e associativa) è possibile, come in questo caso, utilizzare il Reducer come Combiner.

- Esempio:

Secondo Job MapReduce: Reducer

- Il Reducer del secondo job prende in input una coppia chiave-valore dove la chiave è data dalla keyword e il valore è composto da una lista di numeri dati dalle valutazioni dei vari tweet. Tali numeri non necessariamente appartengono all'intervallo [-1;1], per via del Combiner
- Il Reducer restituisce in output la somma algebrica delle valutazioni. Tale valore è indice del sentimento degli utenti relativo alla determinata keyword.
- L'output è composto da coppie chiave-valore. La chiave è di tipo Text, mentre il valore è di tipo IntWritable.

- Esempio:

Link Utili

- Link Dataset Sentiment Analysis: https://www.kaggle.com/kazanova/sentiment140
- Link Dataset Training Classificatore: https://www.kaggle.com/c/twitter-sentiment-analysis2/data
- Link Dataset Nomi: https://mbejda.github.io/
- Link Repo Primo Job MapReduce: https://github.com/liuzzom/Tweet-First-Job
- Link Repo Secondo Job MapReduce: https://github.com/liuzzom/Tweets-Second-Job
- Link Repo Classificatore: https://github.com/liuzzom/Tweets-Polarity-Analysis