NeoMeetup: blabla

Dario Bertazioli[1], Fabrizio D'Intinosante[1], Massimiliano Perletti[1]*

Abstract
yolo

Keywords
Meetup — ADD KEYWORDS— Keyword3

[1]*Data Science, Department of Computer Science, University of Milano Bicocca, Milan, Italy*
*__Corresponding author__: m.perletti@campus.unimib.it

Contents

Introduction

Meetup è stato creato nel 2002 come piattaforma per mettere in contatto le persone nella vita reale. Fondato e guidato inizialmente da Scott Heiferman e Brendan McGovern, nel 2017 Meetup è stato acquisito da WeWork (ora The We Company).
Una volta completata la fase di registrazione, gli utenti possono:

- Selezionare i propri interessi: ciò avviene sottoscrivendo dei topic precompilati dall'applicazione in modo da favorire il sistema di raccomandazione. Questi topic spaziano tra i più svariati ambiti, da quello professionale a quello degli hobby, fino ai più comuni, relativi ad eventi sociali.

- Iscriversi a gruppi locali: una volta selezionati i topic di interesse il sistema di raccomandazione dell'applicazione suggerisce all'utente una serie di gruppi più o meno locali (a seconda dell'area di interesse selezionata dall'utente) che trattano i topic in oggetto o altri topic ad essi correlati. Ciò permette agli utenti di incontrare persone della propria zona che condividono le stesse passioni. Nel caso in cui tra i topic di interesse per l'utente ce ne siano alcuni che non trovano nessun riscontro in gruppi presenti sul territorio, l'applicazione suggerisce all'utente di creare lui stesso un gruppo con oggetto quel topic, consigliando di mettersi in contatto con altre persone che dimostrano quell'interesse, suggerendole attraverso il sistema di raccomandazione.

- Partecipare ad eventi: i gruppi locali, nel corso del tempo, organizzano eventi, incontri, meeting con oggetto i topic dichiarati dai gruppi stessi. Una volta che un gruppo ha organizzato un evento l'utente può visualizzarli sulla propria home e, una volta visionati i dettagli, decidere di comunicare se partecipare oppure no e, nel caso volesse partecipare, se ha intenzione di portare degli ospiti, il tutto in maniera non vincolante. Inoltre, per non restringere troppo la sfera di interessi degli utenti, l'applicazione permette anche di selezionare diversi filtri per la home, tra cui uno apposito per visualizzare eventi organizzati da gruppi di cui non si fa parte o un filtro apposito per visualizzare gruppi di cui l'utente non è già membro e che magari non trattano topic per cui l'utente ha dichiarato interesse, ma che essendo pur sempre gruppi locali, l'utente potrebbe gradire o trovare interessanti.

Il focus centrale dell'intero meccanismo dell'applicazione risulta quindi essere quello del "gruppo" visto come realtà associativa e fautore di momenti di aggregazione durante l'intero anno, molto spesso non guidati da una ristretta cerchia di capi ma, anzi, promotore di iniziative da parte dei suoi stessi membri. La varietà di gruppi, a seconda dei topic trattati, spazia tra:

1. Gruppi di socializzazione, che hanno come obiettivo principale quello di svolgere attività ludiche in compagnia, molto spesso anche promotori di veri e propri eventi di incontro per single.
2. Gruppi professionali, che si pongono l'obiettivo di mettere in contatto persone e professionisti di svariati campi attraverso workshop o presentazioni con il fine di fare crescere gli utenti dal punto di vista professionale.
3. Gruppi creativi, ovvero gruppi di progettazione e più vocati ad hobby ed alla pratica delle più svariate arti.

Sintetizzando, quindi, la missione di Meetup è aiutare le persone a crescere e raggiungere i loro obiettivi attraverso connessioni autentiche e reali. Dal network professionale alla birra artigianale passando per workshop di programmazione, le persone usano Meetup per uscire dalla loro comfort zone, incontrare nuove persone, imparare cose nuove, perseguire le loro passioni e trovare sostegno in comunità che le aiutano a crescere.
Ad oggi, Meetup è disponibile in 186 Paesi, e conta più di 40 milioni di membri sulla piattaforma, con più di 320 mila gruppi attivi ed una media di 12 mila eventi al giorno.

1. Goals

Con l'analisi della rete sociale di Meetup e andando a studiare le relazioni che interconnettono le persone attraverso eventi sociali organizzati dai diversi gruppi presenti in tutto il globo, è possibile avere una mappatura più o meno dettagliata degli interessi comuni che portano ad agglomerare un significativo numero di soggetti instaurando relazioni tra di essi.

Ricreando questa rete, l'obiettivo del progetto è quello di identificare correlazioni o tracce significative che possono determinare delle buone regole nella creazione di eventi di notevole impatto sociale; una piccola guida '**How to**' su come dev'essere organizzato un evento di successo. (CONTINUE)

2. Implementation: Architecture

2.1 Source

: WebSocket and blabla Meetup mette a disposizione parte dei dati in suo possesso relativi a membri iscritti alla piattaforma, gruppi creati ed eventi organizzati nella stessa. In particolare, i dati sono forniti in due modalita:

streaming data : aggiungi descrizione di RSVP, aggiungi esempio di rsvp (json) info sullo streaming (due settimane, 24h), numero totale di messaggi acquisiti, etcetc.

REST api : aggiungi descrizione riguardo enrichment

2.2 Nifi

: Producer and blabla Per aquisire i dati in streaming dalla piattaforma, abbiamo implementato un workflow di Nifi contenente:

- un nodo WebSocket, implementando Jetty-Client (more to add),
- un nodo per il parsing (more to add)
- un nodo ToFile (more to add): decidiamo di salvare i dati acquisiti in file system per due motivi: innanzitutto per avere una copia dei dati di backup, qual'ora si fossero verificati problemi (ad esempio, all'inizio non eravamo sicuri di aver settato correttamente il Retention time di Kafka) in secondo luogo, terminato il periodo di acquisizione, per facilitare la copia dei dati sulle altre virtual machine, in modo da implementare contemporaneamente la stessa pipeline su due macchine, per poter confrontare (quantitativamente, in termini, ad esempio, di quantità di dati importati tempo di esecuzione dei vari scripts etc.) i risultati ottenuti.
- un nodo ToKafka (more to add), per permettere l'ingestion dei dati all'interno di Kafka (vedasi sezione successiva).

2.3 Kafka

: Considerata la natura dei dati a nostra disposizione, che ricordiamo consistere flusso di streaming di messaggi (RSVP), abbiamo deciso di fare un'ingestion in Kafka, potendo successivamente esplorare, iterare, e preprocessare le informazioni tramite Python API. Abbiamo inoltre modificato i parametri di Retention-Time, in modo tale che Kafka fungesse da primo storage temporaneo, in grado di salvare all'interno di un topic tutti i messaggi acquisiti in real time streaming, per poi poterli processare successivamente senza correre il rischio di perdere qualche informazione. Come accennato, con l'API di Kafka per Python abbiamo potuto processare i dati, creando diversi csv (link alla repo) (FIX: non saprei quanto entrare nei dettagli dei vari csv), in modo da facilitare e rendere più efficente l'import nel database finale. (FIX: eventualmente accenna script per automatizzare)

2.4 Neo4j

: Storing and querying Abbiamo scelto di salvare i nostri dati in un database a grafo per via della loro struttura naturale, che ben si presta ad essere schematizzata con nodi principali, proprietà e relazioni che interconnettono tali nodi. Abbiamo provveduto quindi all'import dei dati in Neo4j, sfruttando il linguaggio Cypher e l'ottimizzazione dell'import via csv (accenno alle difficoltà di import precedenti). Per facilitare un'eventuale serializzazione del processo di import, abbiamo creato uno script (Bash) che, una volta creati i csv, consente di automatizzare le varie chiamate alla cypher-shell, dato che il processo di creazione del database risulta piuttosto laborioso ma facilmente standardizzabile (FIX: add link to import.sh). Aggiungere info quantitative: nodi totali creati,

3. Results

Fabri <3 Meetup xD.

Interesting Challenges

Circumstances where fabri has sclerated:

Symbolic Links : Per ovviare a un bug/limite dei nodi getFile/FetchFile che si verifica quando si tenta di fare un'ingestion di una elevata quantità di file in una sola volta, abbiamo escogitato una soluzione creando dei link symbolici, dividendo la creazione in sottocartelle, facendo leggere al nodo GetFile questi

link e impostando il nodo in modo che venisse eliminato il link dopo la singola lettura/ingestion. (Va detto ovviamente meglio)

Ottimizzazione script per gestione memoria : fixme.

Import CSV vs PyNeo : prestazioni dell'import via csv con Cypher nettamente superiori a quelle ottenute tramite PyNeo (Python API di Neo4j). (aggiungi esempio e amplia)

Acknowledgments

So long and thanks for all the fish.