Freie Universität Berlin

Datenbanksysteme SS17: Projekt

Dozentin: Agnes Voisard

Bernadeta Chisarau, Dor Cohen, Mihai Renea

2 iunie 2017

---

1 Projektdokumentation

Das Team besteht aus: Bernadeta Chisarau, Dor Cohen and Mihai Renea. Da jeder von uns ein anderes Tutorium besucht, konnten wir die unterschiedlichen Gespräche aus den verschiedenen Tutorien in der Vorbereitung und Durchführung der ersten Projektiteration zusammentragen. (siehe nachfolgende Dokumentation)

2 Datenalanlyse

Der Datensatz erfasst mehr als 6000 Tweets von dem Wahlkampf zwischen den Kandidaten für US-Präsident Donald Trump und Hillary Clinton. Der Datensatz entsteht aus den folgenden Feldern:

1. *handle* - Der Autor des Tweets.
2. *text* - der Inhalt.
3. *is_retweet* - Markiert, ob es ein Retweet ist.
4. *original_author* - Falls Retweeted, der originale Autor.
5. *time* - Das Time-Stamp des Tweets (Datum und Uhrzeit).
6. *in_reply_to_screen_name* - Der Name der Person, für die das Tweet eine Antwort sein soll (inkonsistent).
7. *is_quote_status* - ?.
8. *retweet_count* - Anzahl der Retweets von einem Tweet.
9. *favorite_count* - Anzahl der Likes.
10. *source_url* - Warscheinlich die Quelle des Tweets.

Manche der obengenannten Feldern werden wir nicht in Betracht ziehen, weil sie irrelevant für unsere Zwecke, inkonsistent oder mangelhaft sind:

- *original_author* werden wir Ignorieren, da nur eine Name keine interessante Information uns geben kann. *is_retweet* werden wir aber behalten, da es eine Rolle bei der Funktion der Wichtigkeit von Tweets eine Rolle spielen könnte.

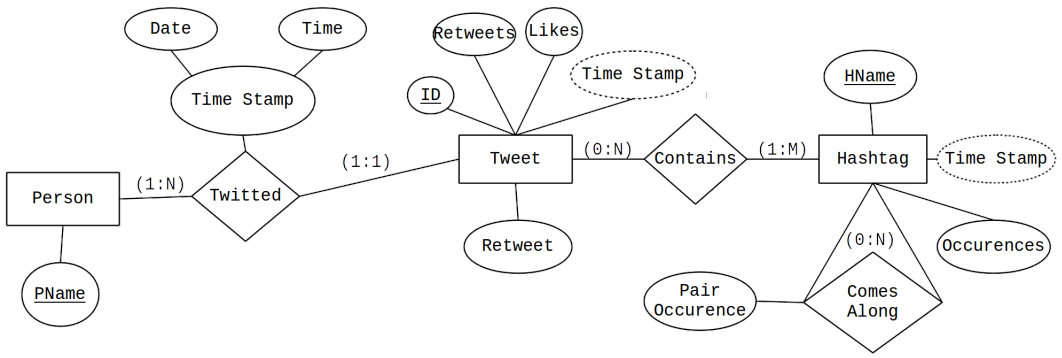
- *in_reply_to_screen_name* kann weggelassen werden, da es inkonsistent und scheinbar fehlerhaft erzeugt wurde (z.B. in der Mehrheit der Fällen, wo dieser Feld auftritt, antwortet Hillary Clinton sich selbst).
- In *is_quote_status* konnten wir keine Schablonen erkennen, deshalb liefert das uns nichts nutzbares.
- *source_url* ist irrelevant.

Andere Informationen sind aber von besonderer Wichtigkeit für unseren Zweck:

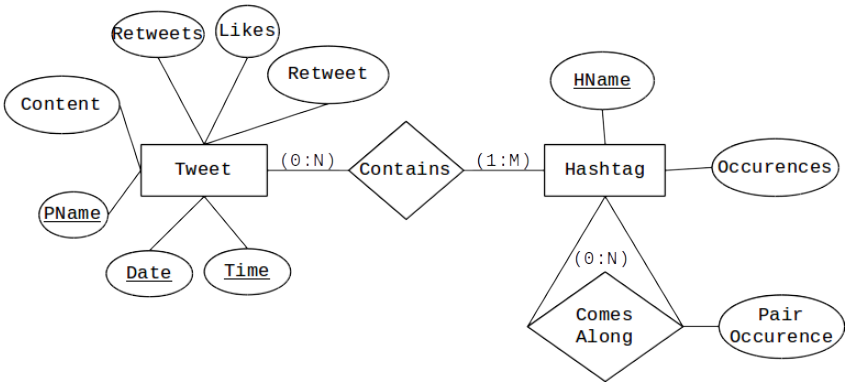
- *time* wird uns helfen die Entwicklung der Hashtags-Nutzung über die ganze Zeit zu analysieren.
- *retweet_count* und *favorite_count* werden die Schlüsselargumente fur die Modellierung der Funktion der Wichtigkeit von Tweets sein.

3 ER-Modellierung und das relationale Modell

Die folgende MinMax-Diagramm stellt die erste Überlegung für das ER-Modell vor.



Im Zuge unserer Überlegungen hat sich unser Schema an mehreren Stellen vereinfachen lassen: das ursprüngliche modulare Modell hat sich zu einem monolithisch gerichteten Modell verändert. Beispielsweise haben wir erst *Person* als Entität im ER-Diagramm dargestellt, haben aber im Nachhinein festgestellt, dass diese als Attribut der Entität *Tweet* denselben Zweck erfüllt, ohne einen Umweg gehen zu müssen über die Relation *Tweet* was *Twitted* by *Person*. Mehr dazu, ein Tweet kann jetzt mit dem Tupel (*PName, Date, Time*) eindeutig identifiziert werden.



Beschreibung des Diagramms:

- *Tweet* - entspricht jeder Zeile in dem gegebenen Datensatz.
- *Retweets, Likes* - Anzahl der Retweets und Likes von diesem Tweet. *PName* ist der Autor des Tweets. Alle drei fassen das Superkey zusammen.
- *Retweet* - Markiert, ob das Tweet ein Retweet ist.
- *Content* - Der Inhalt des Tweets.
- *Date, Time* - Die Datum und die Uhrzeit der Veröffentlichung eines Tweets, als zwei separate Attribute gespeichert, da die Informationen über die Long- bzw. Short-Term Entwicklung der Trends liefern können.
- *Contains* - Diese Relation verbindet die Hashtags mit den Tweets, in denen sie Auftauchen.
- *Hashtag* - Jeder Hashtag ist eine Entität.
- *HName* - Name des Hastags, also der Hashtag selbst.
- *Occurences* - Wie oft ein Hashtag auftaucht.
- *Comes Along* - Diese Relation fasst zwei Hashtags als Paar zusammen. *Pair Occurence* zählt wie oft ein Paar auftaucht.

Zusammenfassend ergibt sich das relationale Modell:

Tweet(PName, Date, Time, Retweets, Likes, Retweet, Content)
Contains(Pname, Date, Time, HName)
Hashtag(HName, Occurences)
Comes Along(HName1, HName2, Pair Occurence)

4 Datenbank erstellen

Die Datenbank zu erstellen ist relativ Einfach:

1. Shell-Login mit dem privilegierten Standard-PostgreSQL-Nutzer `postgres`, dann das `psql` Shell starten.
2. Neues Nutzer erstellen (soll einem UNIX-Nutzer übereinstimmen) mit dem Befehl:
   `CREATE USER <UNIX-Nutzer> WITH PASSWORD '<password>';`
3. Die Datenbank *election* erstellen mit dem Befehl:
   `CREATE DATABASE election;`
4. Letztens die Datenbank dem Nutzer zuweisen:
   `GRANT ALL PRIVILEGES ON DATABASE election to <UNIX-Nutzer>;`

Der Nutzer hat jetzt Zugriff auf die Datenbank *election* mit `psql` indem er in einem Terminal das Folgende aufruft:
`$ psql -d election`

5 Datenbankschema erstellen

- $https://github.com/derMihai/DBS_Project/blob/master/election_schema.sql$

6 Datenbereinigung und Datenimport

- Unser Code ist unter dem Link: $https://github.com/derMihai/DBS_Project$ zu finden.