Projektseminar Angewandte Informationswissenschaft

Thorsten Brückner

Datenanalyse mit Hilfe von Twitter API und Python zum Thema:

Hype - Release von World of Warcraft – Legion am 30.08.16. In welchen Ländern wird der Release von WoW – Legion am meisten über Twitter gehyped und welche Sprachen werden am häufigsten benutzt?

Bereits fertig (Zusammenarbeit mit Raphael Katschke):

• Twitter-Crawler in Python auf Basis der Twitter-API programmieren.

1. Twitter API-Key erstellen (Access Token)
    a. Twitter Account benutzen und https://dev.twitter.com/apps/new besuchen
    b. Application erstellen mit Name, Beschreibung und URL (optional)
    c. TOS akzeptieren
    d. Twitter apllication abschließen und API Key kopieren
2. Twitter Access Token erstellen
    a. Consumer Key, Consumer Secret, Access Token, Access Token Secret kopieren und bei Theme Options einfügen
3. Crawler in Python programmieren
4. Python 0Auth2/Tweepy library benutzen
5. Accessing Data

To Do:

• Tweets zum Thema WoW-Legion erkennen und extrahieren über den Zeitraum von XX.XX.XX bis zum Release am 30.08.2016.

Mit Hilfe von GEO Tags, die Herkunftsländer von den Tweets analysieren und mit Networkx in Python grafisch darstellen.

1. Key-attributes/fields wählen. Welche Attribute seitens der Twitter-API sind nötig?
    a. Text
    b. Geo-Tag

c. Coordinates/Places

d. ID

2. Tweets mit Python fetchen:

    a. Letzten 3000 Tweets bis zum 29.08.16

    b. Streaming der Tweets vom 30.08.16

3. Daten/Tweets in CSV speichern

4. Daten/Tweets in Python importieren

5. Visualisierung mit Networkx (eventuell in Kombination mit Gephi) umsetzen

    a. Idee: Die Länder anhand von Geo-Tags identifizieren und je nach Ergebnissen diese mit Hilfe von den unten erwähnten „Graph Layouts“ umsetzen. Eine Kombination mehrerer Layouts ist möglich, wird sich aber erst herauskristallisieren, wenn die Menge der Ergenisse feststeht. Durch die Menge der Ergebnisse wird bei den unterschiedlichen eventuell die Übersicht leiden, daher diese Entscheidung.

        a.i. Benutzen von „Complete Graph“ and „Bipartite Graph“

        a.ii. Mögliche Graph Layouts (Entscheidung fällt mit Ergebnissen):

            a.ii.1. Open Ord für „besser unterscheidbare Cluster“

            a.ii.2. Yifan Hu für „dynamisches Map Clustern“

            a.ii.3. Force Atlas für „Galaktisches Clustern“

            a.ii.4. Fruchterman Rheingold für „Mass-Partikel Clustern“

            a.ii.5. Welches Layout verwendet wird, entscheidet sich anhand der Ergebnisse erst später (zwecks Übersicht)

    b. Allgemeinen und Twitter Graphen generieren

        b.i. Hinzufügen von Notes und Edges anhand von Geo-Tags/Coordinates/ID

• Ebenfalls soll die Sprache der Tweets analysiert und dargestellt werden.

1. Stoppwortliste
    a. Englisch
    b. Deutsch
    c. Französisch
    d. Spanisch
2. Langdetect Lib für Python
    a. Wörterbücher erweitern mit nötigen „Game und Szenespezifischen Wörtern/Tags in erforderlichen Sprachen“

        a.i. Z.B. Englisch → Deutsch → Spanisch → Französisch

            a.i.1.a. Warrior → Krieger → Guerrero → Guerrier

            a.i.1.b. Paladin → Paladin → Paladín → Paladin

            a.i.1.c. Hunter → Jäger → Cazador → Chasseur

            a.i.1.d. Rogue → Schurke → Pícaro → Voleur

            a.i.1.e. Priest → Priester → Sacerdote → Prêtre

            a.i.1.f. Death Knight → Todesritter → Caballero de la Muerte → Cevalier de la mort

            a.i.1.g. Shaman → Schamane → Chamán → Chaman

            a.i.1.h. Mage → Magier → Mago → Mage

            a.i.1.i. Warlock → Hexenmeister → Brujo → Démoniste

            a.i.1.j. Monk → Mönch → Monje → Moine

            a.i.1.k. Druid → Druide → Druida → Druide

            a.i.1.l. Demon Hunter → Dämonenjäger → Cazador de demonios → Chasseur de démons

b. Text + ID (duplicates vermeiden)

    b.i. Funktion schreiben, die „Text“ als String entgegen nimmt und mit Hilfe von der Langdetect Lib die Sprache ausgibt/identifiziert

    b.ii. Anhand der Sprache wird danach zwischen verschiedenen „Dictionaries“ unterschieden und eingetragen

    b.iii. Nicht erkannte Texte manuell überprüfen und Sprache markieren

3. Plotly

    a. Als Visualisierungs-Lib benutzen um Sprachen in Diagrammen/Graphen anzuzeigen. Plotly habe ich gewählt, da mir Matplotlib sowie GEOJson verwehrt wurden, weil ich damit bereits gearbeitet hatte und Plotly mich anspricht, da es sehr variabel wirkt.

        a.i. Importieren von Plotly

            a.i.1. Simple Bar Charts vs Grouped Bar Charts vs Stacked Bar Charts

            a.i.2. Relative Barmode

            a.i.3. Dashboard – Geo Map?

        a.ii. „Bar plot“ erstellen

            a.ii.1. Anpassen vom Plot

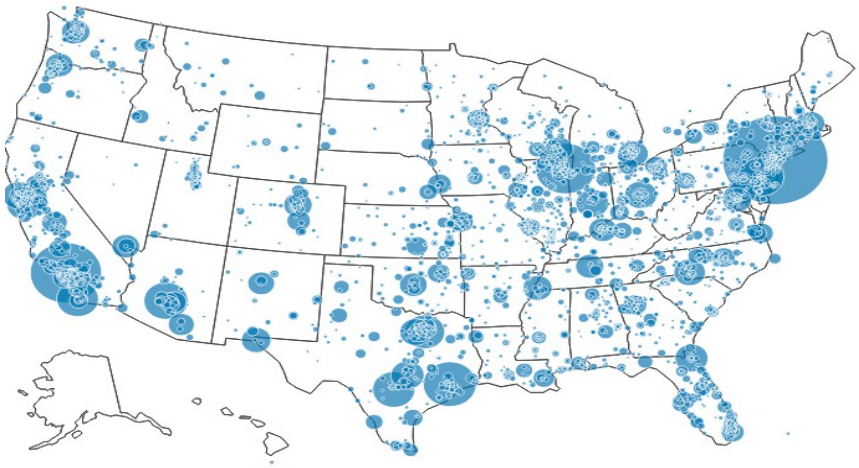
            a.ii.2. Labels den Bars/Achsen hinzufügen

            a.ii.3. Customizing Colors

                a.ii.3.a. Color Mapping

    b. Die Idee hierbei beinhaltet zwei Punkte. Zum einen möchte ich die Sprachen individuell mit Diagrammen darstellen und vergleichen und die Geo-Tag Ergebnisse von Twitter über eine Weltkarte ähnlich der USA-Abbildung darstellen. „ON TOP“ möchte ich noch die unterschiedlichen Sprachen mit verschiedenen Farben mit einbeziehen um deutlich zu machen, welche Sprache wo getwittert wurde. Das hängt allerdings davon ab, ob die Tweets der entsprechenden Sprachen auch Geo-Tags beinhalten.

4. Sollte ich Probleme bei der Darstellung mit Plotly bekommen, werde ich auf GEOJson umschwenken. Da ich allerdings schon einmal damit gearbeitet hatte, durfte ich dies nicht als erste Wahl verwenden.



Desweiteren möchte ich gerne den Versuch mit Plotly wagen um etwas Neues zu probieren.

5. Über PHP die Ergebnisse auf meiner Homepage anzeigen lassen (Optional wenn die Zeit es zulässt). Wie wir die gemeinsamen Ergebnisse darstellen können ist zur Zeit noch ungewiss. Dies ist abhängig von beiden Projektergebnissen und entscheidet sich erst später.