SentiMentality – karnyújtásnyira a közvélemény

Cyber-sentiment

Kónya Leon, Sztarek Norbert
Felkészítő tanár: Kőrösi Gábor
Bolyai Tehetséggondozó Gimnázium és Kollégium, 24400 Zenta, Posta utca 18.

1. Bevezetés

Manapság rohamosan szaporodik a felhasználó által generált dokumentumok mennyisége közösségi oldalakon, hálózatokon. Ilyen oldalak például a Twitter, Facebook, Reddit, közösségi hírportálok kommentjei, politikai vitafórumok és sok más. Ezek a dokumentumok sok esetben valamilyen véleményt fejtenek ki egy témáról, érzelmet fejeznek ki.

Sok vállalkozás vagy politikai kampány sikeressége, bármekkora méretekben, függ a felhasználói, fogyasztói visszajelzéstől. Fontos számukra, hogy a visszajelzés tükrében helyes irányba tudják fejleszteni anyagi vagy eszmei terméküket, marketingjüket javítani.

A fenti gondolatmenet alapján állíthatjuk, hogy előnyös lenne számos szervezetnek statisztikai mennyiségben kinyerni a közösségi hálózatokon rejlő közvéleményt, digitalizálni, megmérni az emberek érzelmeit.

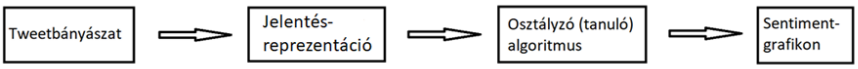
A szövegbányászat és a közvélemény-kutatás már eleve létező iparágak és kutatási területek, amelyek ezzel a témakörrel foglalkoznak, ám tapasztalataink szerint mindazon emberek számára, akik nem rendelkeznek egy előrehaladott vállalkozással vagy programozási háttérrel, nehézségek merülhetnek fel céljuk elérésében.

Ez motivált bennünket arra, hogy elkészítsünk egy szoftvert, amely vizsgálja a pozitív és negatív hangnemű felhasználói bejegyzések arányainak változását Twitteren. Lehetővé teszi programozási tapasztalattal nem rendelkező emberek számára, hogy következtetéseket vonjanak le az adatból, szociológiai kutatásokat végezzenek, feltörekvő vállakozásuk termékének hírnevét követni tudják, vagy csupán csak kielégítsék kíváncsiságukat.

2. Probléma megoldásának menete

Az elképzelésünk az volt, hogy egy olyan szoftvert készítsünk, amely képes ábrázolni egy grafikonon a Twitterről kinyert pozitív és negatív hangnemű szövegek arányának a változását. A grafikonok szelekcióját úgy képzeltük el, hogy a felhasználó megadja az időintervallum kezdetét és végét (keltezéseket), valamint egy Twitter témát (hashtaget), hogy a szoftver e témán belül az adott időszelet „szentiment-grafikonját” rajzolja ki.

A munkánk során 4 főbb lépést különítettünk el, amelyet ábra 1 szemléltet.



1. ábra: munkánk folyamatábrája

2.1. Tweetek kibányászása

A megadott követelmények alapján kérvényt lehet küldeni a TwitterAPI felületnek, amely tweetszövegeket ad vissza illetve azok további adatait. Egyetlen hátránya ennek az eljárásnak az, hogy vannak időkorlátai, így bizonyos időnél régebbi tweeteket már nem vesznek figyelembe. Erre használtunk egy TwitterScraper alkalmazást, amely a felhasználó görgetését szimulálja JavaScript eljárásokkal, így elérhetővé teszi a korlátlan tweetkinyerést. A felület kezelése Python nyelven történt.

2.2. Tweetek feldolgozása

Ahhoz, hogy szöveganalízist tudjunk végezni egy szövegen, valamilyen számszerű formába át kell alakítani. Mi erre a Doc2Vec eljárást alkalmaztunk, amelyhez át kellett néznünk pár publikációt az eljárás matematikai hátteréről, valamint a Python gensim könyvtárát, amely a vektorizációs eljárásokat biztosítja.

Maga a vektorizáció úgy történik, hogy első alkalommal, amikor végigment a program a tweeteken, felvesz minden szóra véletlenszerű értékeket, majd amikor következő pár (a mi esetünkben 50) alkalommal ez megtörténik, kontextus alapján változtat a szavak vektorjain (tehát ha hasonló kontextusban vannak jelen, akkor az értékeket egymáshoz közelebb rajta). Ezt az eljárást Doc2Vec-nek nevezzük, vagy Paragrafus vektornak (egy egyszerűsített példát láthatnak a *2. ábrán*).

Miután van számszerű adatunk, készek vagyunk arra, hogy szabályszerűségeket keressünk egy algoritmus segítségével az adatban. Ez úgy történik, hogy egy előre osztályzott, valamint a Doc2Vec-es eljárás segítségével vektorizált tweeteket az n (a mi esetünkben 400) dimenziós vektortérben elhelyez, majd egy n-1 dimenziós függvényt rajzol, amely 2 részre osztja a teret, a pozitív és negatív részre. Az alapján, hogy az új tweetek hol helyezkednek el, tudja a program meghatározni az ő számban kifejezett érzelmüket. Ezt az eljárást Logistic Regression-nek nevezzük (egy egyszerűsített példát láthatnak

a *3. ábrán*). Ezeket a metódusokat Python programozási nyelvben írtuk meg, főként a scikit-learn, valamint a gensim könyvtárak segítségével.

2.3. **Statisztikai feldolgozás**

Miután meg lett minden egyes szöveg szentimentje (az, hogy 1 vagy 0) határozva, a grafikon kirajzolásához összegeznünk kellett az eredményeket. A szervergép, amely az egészet lekérte a weboldalról, ismét felküldi a weboldalnak PHP oldalakon keresztül, amely kiszámítja a „szentiment-grafikont". Ábra 4-en látható a felhasználói felületén kirajzolt grafikon, melynek elemzése a 3. fejezetben következik.

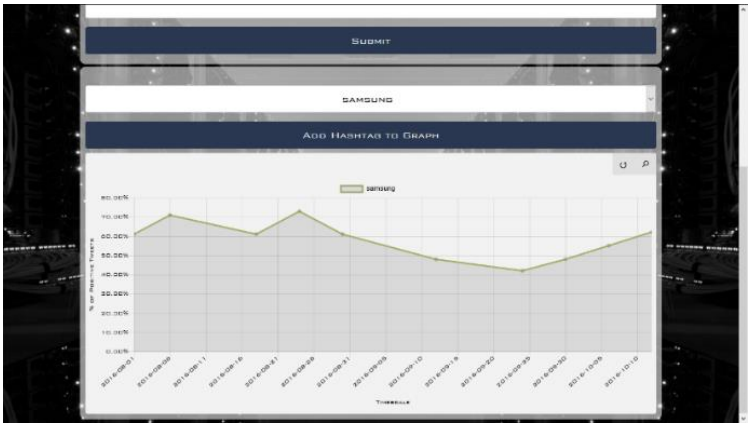
3. Elért eredmények

Ami a Samsung 2016-os évét illeti, számára nem volt túlságosan sikeres. Ha megvizsgáljuk a grafikont, egy emelkedést figyelhetünk meg augusztusban, pont abban az időben, amikor a Note 7-es telefon piacra adását bejelentette a cég. Miután piacra került a telefon, egy gyári hiba következtében világszerte elkezdtek felrobbani az akkumulátoraik, ami várhatólag a Samsung hírnevének csökkenését eredményezte.

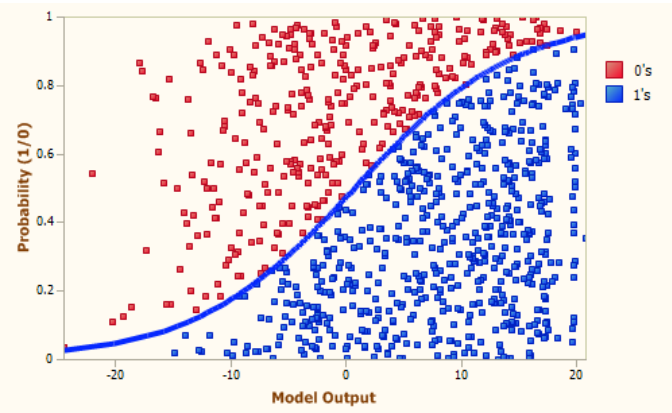
Ezt a feltételezést igazolja a grafikonunk, amely augusztus vége felé komoly süllyedést jelez.

Ez a példa, illetve tétel alapján állíthatjuk, hogy az általunk használt eljárás valósan tükrözi a valóvilágban történő eseményeket, vagyis megbízható, közvélemény-kutatásra alkalmas szoftvert hoztunk létre.

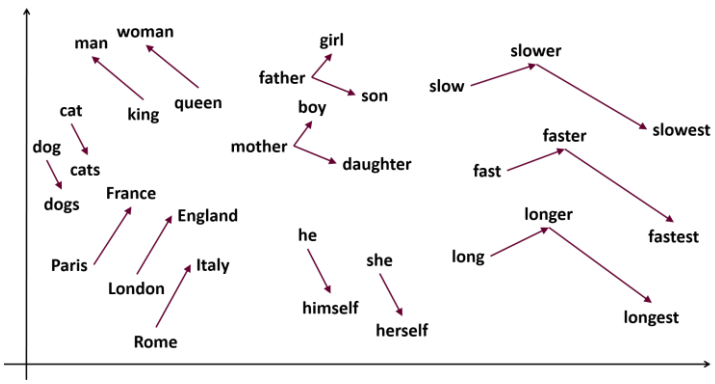
Mivel a kezdetekben kikötött célunkat elértük, felmerül a kérdés, hogy hogyan tovább. Saját, központi szervergép igénybe vétele aránylag költséges lenne, így egy félig decentralizált paradigma kidolgozásán gondolkodunk. Alapjában véve minden felhasználó, aki a standardnál nagyobb mennyiségű adathoz szeretne hozzáférni, annak hozzá kell járulnia a „közvélemény kibányászásához", olyan formában, hogy processzorerejének egy részét felajánlja a rendszer számára, és egy ún. proof of work-öt számít ki, amelyet ellenőrizhet bárki. Ez valójában olyan, mintha egy állandó függvény eredményét számítaná ki a felhasználó, bemenetként véve egy időintervallumot és egy témanevet.



2. ábra: Samsung 2016-os grafikonja a felhasználói felületen



3. ábra: bináris klasszifikáció szemléltetése



4. ábra: vektoros jelentésreprezentáció szemléltetése