FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Bezpečnost informačních systémů
projekt 2 - detekcia spamu

2017 Michael Halinár - xhalin01

1 Úvod

Úlohou tohto projektu je implementácia programu na detekciu spamu v emailoch. Táto problematika nie je vôbec triviálna vzhľadom na rôznorodosť spamu. Formát `.eml` obsahuje niekoľko relevantných častí pre detekciu spamu, zvlášť hlavne obsah správy, predmet, odosielatela a použitý charset[1]. Vzhľadom na to že použitý charset sa v ňom nie vždy vyskytuje, preto ho nebudeme brať do úvahy. Odosielateľ môže byť taktiež identifikátorom spamu. Napríklad ak dotyčný používa emailovú schránku `freemail`, obsah bude pravdepodobne spam. Najväčšiu výpovednú hodnotu bude mať však obsah samotnej správy a predmet.

2 Triviálny algoritmus na detekciu spamu

Prvou verziou tohto programu bola triviálna detekcia na základe kľučových slov. Predmet emailu a obsah sa spojí do jedného celku. Následne sa v texte vyhľadávajú kľučové slová ktoré s najväčšou pravdepodobnosťou budú súčasťou spamu. Patria sem slová ako napríklad: "guaranteed win", "pharmacy"a podobne. Akonáhle sa v texte nachádza čo i len jedno kľúčové slovo, bude správa považovaná za spam. Avšak toto riešenie bolo na hranici použiteľnosti, keďže na vzorke asi 1000 mailov sa ukázala schopnosť detekovať iba 20% spamu. Zároveň tento algoritmus správne detekoval pomerne malé množstvo hamu - 90%. Preto bolo potrebné nájsť iný spôsob detekcie.

3 Naive Bayes Classifier

Naive Bayes Classifier dokáže zaradiť vzorku do jednej z tried(spam/ham) na základe jej vlastností(obsahu textu). Vychádza zo vzorca:

$$P(A \mid B) = \frac{P(B|A)*P(A)}{P(B)}.$$

Kde pod $P(A \mid B)$ si môžeme predstaviť že je to pravdepodobnosť výskytu spamu A na základe jeho kľučových slov B. Uvažujme teda vzorec:

$$P(A \mid B) > P(\ulcorner A \mid B).$$

Pokiaľ pravdepodobnosť že A je spam pri daných vlastnostiach B je väčsia ako pravdepodobnosť že A nie je spam pri daných B, môžeme tento email prehlásiť za spam. Uvažujme teda vzorec že a nie je spam pri daných B:

$$P(\ulcorner A \mid B) = \frac{P(B|\ulcorner A)*P(\ulcorner A)}{P(B)}.$$

Kedže $P(B)$ sa vyskytuje v obidvoch prípadoch ako delitel, je možné ho z rovnice vylúčiť čím dostávame vzorec:

$$P(A) * P(P \mid A) > P(\ulcorner A) * P(B \mid \ulcorner A).$$

Pravdepodobnosti $P(A)$ a $P(\ulcorner A)$ sa počítaju jendoducho. Je to pomer spamu/hamu a celkového počtu emailov. Náročnejší výpočet je pravdepodobnosťi $P(B \mid A)$ respektíve $P(B \mid \ulcorner A)$. Ide o súčin pravdepodobností pre každé slovo v emaile. Táto pravdepodobnosť pre každé slovo je podiel počtu kedy sa slovo nachádza v emailoch označených ako spam a celkového počtu slov vo všetkých spam emailoch. Inými slovami povedané ak sa slovo nachádza viackrát v spamoch ako v hamoch, toto slovo pridá na pravdepodobnosti že skúmaný email je spam. Tento algoritmus je teda veľmi závislý od veľkosti trénovacej sady, preto je vhodné aby bola čo najväčšia možná.

4 Implementácia

Knižnica `nltk` poskytuje aj Naive Bayes klasifikátor. Pred použitím je nutné tento klasifikátor naučiť rozlišovať spam. Ako testovací dataset bolo využitých 3 500 mailov[2] z celkového počtu 52 000 obsahujúcich spam aj ham. Toto číslo bolo zvolené experimentálne, príliš veľa emailov malo za následok nárast veľkosti klasifikátora ktorý sa neskôr

[1] https://en.wikipedia.org/wiki/Character_encoding
[2] http://www2.aueb.gr/users/ion/data/enron-spam/

zapisoval do súboru a nebolo by možné ho odovzdať do školského informačného systému. Pre spracovanie emailu bola použitá knižnica `email` ktorá zo súboru `.eml` dokáže jednoducho získať obsah správy. Následne je vhodné získanú správu upraviť do podoby ktorú klasifikátor vie používať. Obsah správy sa teda rozdelí na tokeny, ktoré sa dalej upravia do jednotnej podoby pomocou lemmatizátoru. To znamená že slovo buy a Buying sa bude považovať za jedno a to isté. Nasleduje vyradenie nepodstatných tokenov ako "the, as"a podobne, tieto slová sa nazývajú stopwords a sú poskytované knižnocou `nltk`. Ďalej sa k týmto slovám pridá popis či sa jedná o spam alebo ham. Takto upravené pole spracovaných emailov je pripravené na trénovanie klasifikátoru. Po dokončení sa daný klasifikátor uloží do súboru pomocou knižnice `pickle` a môžeme ho použiť vo výslednom programe pre klasifikáciu.

Program príjme ľubovolný počet emailov ako argumenty. Po ich spracovaní s už spomínanou knižnicou, nasleduje klasifikácia emailu pomocou naučeneného klasifikátoru, ktorý je načítaný zo súboru. Takto sa postupne spracováva každý email. Ak klasifikátor prehlási že email je spam funkcia pre výpis vypíše výsledok výstup.

5 Použíté knižnice

- email - pre spracovanie emailu
- pickle - uloženie klasifkátoru do súboru
- nltk - tokenizácia, lematizácia, klasifikátor, stopwords
- sys - získanie argumentov

6 Návod na použitie

./antispam [1.eml 2.eml ... n.enl]

- 1.eml - názov súboru vo formáte `.eml` (možnosť zadať lubovoľný počet)