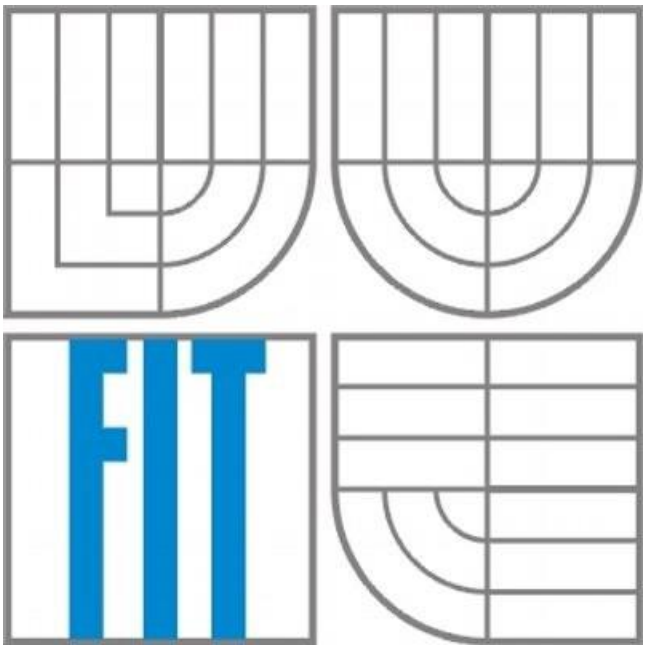

BIS – projekt 2
Detekce spamu

Ondřej Strmiska - xstrmi08

Fakulta Informačních technologií VUT v Brně
2017/2018

Jak spustit aplikaci

Pro spuštění aplikace je nutné mít ve stejné složce soubory:

- antispam.py
- Makefile
- spamWords
- spamWords2
- spamPhrases
- spamAddress

Aplikace se připraví pro spuštění provedením příkazu **„make“** ve složce s výše uvedenými soubory. Tento příkaz vytvoří nový spustitelný soubor antispam a stáhne potřebnou knihovnu.

Spuštění aplikace probíhá příkazem: **„./antispam {parametry}“**, kde parametry můžou být libovolný počet souborů (případně cesty k souborům, pokud nejsou ve stejné složce) s koncovkou .eml. Příklad spuštění je například „./antispam email1.eml email2.eml“

(De)aktivace odůvodnění

V aplikaci je defaultně zapnuté odůvodnění, které vypisuje důvod (většinou víceřádkové), proč emailu udělilo dané skóre. Pro přehlednost výsledku je možné odůvodnění vypnout „odkomentováním“ příkazu v souboru antispam.py na **171.** řádku aplikace.

#oduvodneni = "" ----> oduvodneni = ""

Informace o aplikaci

Aplikace je napsána v programovacím jazyku Python. Aplikace využívá knihovnu Email pro parsování souborů eml a knihovnu html2text pro převod html textu na plaintext. Knihovna html2text se stáhne (pokud ještě není stažená) provedením příkazu „make“.

V první fázi aplikace se nejprve naimportují potřebné knihovny. Následně se zjistí, zda byl uveden nějaký parametr (email), jinak se vypíše hláška, že nebyl uveden žádný soubor. Pokud se aplikace spustí korektně (s parametry), tak se naimportují textové soubory a vytvoří se z nich seznam String řetězců potřebné pro další fungování (podrobněji uvedeno níže).

Při spuštění cyklu se v 1. řadě načte eml soubor (viz parametr) a ten se rozparsuje. Správné rozparsování obecně jsem v tomto projektu považoval za největší problematiku, vzhledem k tomu, že se mi nepodařilo najít jednotný způsob jak rozparsovat všechny eml soubory (uvedené v příkladech). U mnoha souborech se vypisovali různé chybové hlášky (problém s kódováním, či formátem souboru). Proto mám aplikaci nastavenou tak, že je možné soubor rozparsovat až 3 způsoby. Pro 2 z těchto způsobů se využívá knihovna html2text. Z rozparsovaných souborů se vytvoří několik stringů, které jsou naplněny podle obsahu emailu (text, předmět, příjemce, odesílatel, datum). Poté již probíhá samotné vyhodnocení emailu

Hodnocení

Hodnocení probíhá pomocí skóre, které má implicitní hodnotu 1 a pokud se splňují určité podmínky, skóre roste. Pokud skóre přesáhne určitou hranici bodů, je email vyhodnocen jako spam.

Hodnocení textu

Jednou z nejdůležitějších částí je hodnocení samotného textu emailu. Celé hodnocení textu je založené na počtu poměrů celkových slov a slov a tzv. spamových slov. Čím menší je poměr tím se zvyšuje skóre emailu (blíží se k vyhodnocení jako spam). Spamová slova se získávají ze souborů spamWords a spamWords2 v těchto souborech se nachází různá slova, která se nejčastěji vyskytují ve spamových emailech. Slova jsou uvedena v angličtině, češtině a slovenštině a v různých velikostních verzí, jelikož například „spam“ není stejné jako „Spam“, ani jako „SPAM“ a nepodařilo se mi to programově podchytit, proto je tam většina slov uvedená 3x.

V souboru spamWords se nachází často slova, u kterých je velice pravděpodobné, že se budou vyskytovat ve spamech, ale je zde velká šance, že se můžou nacházet i v běžných e-mailech. Zatímco v souboru spamWords2 se nachází slova převážně se sexuálním kontextem u kterých je málo pravděpodobné, že se budou nacházet v běžných emailech.

Hodnocení probíhá tedy tak, že se vytvoří for cyklus a zkoumá se, zda se spamové slovo vyskytuje v textu. Pokud ano zvýší se proměnná *pocetSpamSlovVTextu*, se kterou se počítá ve výše zmíněném poměru. U slov ve spamWords.txt je podmínka že slovo musí být rovné slovu v textu (například spamové slovo **hot** se nazapočítá i když se nachází ve slově **hot**ovost) a započítá se pouze jednou (i když se v textu nachází víckrát). U spamWords2.txt se zvyšuje proměnná *pocetSpamSlovVTextu* více, a zvyšuje se za každý výskyt slova. Navíc se slovo započítá i když vyskytuje jenom částečně (například spamové slovo **sex** se započítá, když se nachází ve slově **sex**uální).

Dalším kritériem jsou spamové fráze, nacházející se v souboru spamPhrases, stejně jako u slov se fráze se hledají, zda se vyskytují v textu. Pokud ano, navýší se skóre emailu

Hodnocení předmětu

Hodnocení předmětu funguje v celku podobně jako hodnocení textu, za každý výskyt spam slova v předmětu se zvyšuje skóre emailu. Navíc kromě spam slov se v předmětu ověřují další věci. Například skóre se zvýší pokud předmět obsahuje vykřičník, otazník, nebo zavináč. Také se zvýší pokud je nadpis psaný velkými písmeny, nebo obsahuje aspoň částečně slovo „http“.

Další hodnocení

Aplikace ještě hodnotí kolonku adresáta u které ověřuje zda se adresát nenachází v souboru spamAddress.txt. Tento soubor je naplněn adresy, které jsem buďto našel na internetu jako spamové adresy a nebo jsem tyto adresy vytáhnul ze soukromích emailových účtů.

Také se hodnotí počet znaků v plain textu vůči počtu znaků v celém html emailu. Pokud je v html zprávě příliš velké množství tagů vůči samotnému textu, zvyšuje se skóre. Ovšem tato metoda je funkční pouze při jednom typu rozparsování.

Poslední implementovaná metoda kontroluje, zda email neobsahuje pouze html náhled, takovýto email je automaticky vyhodnocen jako spam.