Spam filter - report

Chalupa Petr, Štácha Martin

RPH, 2024/2025

Úvod

Tento projekt je týmová úloha, jejímž cílem bylo napsat program, který bude klasifikovat emaily jako spamy, nebo jako nezávadné, s co nejvyšší přesností. Filtr má možnost se naučit jak spam vypadá na testovací sadě emailů.

Trénování filtru

Dle specifikace začíná algoritmus trénování v metodě `train()`. Algoritmus se skládá ze dvou hlavních částí; spočítání tzv. flagů a spočítání tzv. limitů. Flagy jsou určité vlastnosti emailu, na jejichž základě se následně emaily klasifikují. Limity pak určují kolikrát je potřeba porušit nějakou takovou vlastnost, aby mohl být email díky ní klasifikován jako spam.

Flagy, které se počítají z trénovací sady dat jsou:

- Headery, které se vyskytují pouze ve spamových emailech
- Domény odesílatele, které se vyskytují pouze ve spamových emailech
- Odkazy vyskytující se pouze ve spamových emailech
- Průměrný poměr velkých a malých písmen ve spamových emailech
- Průměrný výskyt předem zvolených znaků v nezávadných emailech společně se znaky vyskytujícími se pouze ve spamových emailech (výskyt = 0)

Další flagy zjištěné externími skripty, které tvoří statické seznamy:

- Často se vyskytující slova v předmětech spamových emailů
- Často se vyskytující fráze v předmětech spamových emailů

Poté se iteračně počítají limity k jednotlivým flagům. Nejdříve jsou pro každý spamový email zjištěny všechny flagy, které (a kolikrát) má. V každé iteraci se stanoví klasifikace všech emailů na základě jeho spočítaných flagů a aktuálních limitů (viz kapitola *Klasifikace emailů*). Poté je spočítána úspěšnost filtru, kdy pokud je úspěšnost nižší, než v minulé iteraci, limity jsou vráceny do stavu minulé iterace; v opačném případě je zaznamenán aktuální stav a každý limit je upraven přičtením náhodného čísla z `<-3; 3>` (limit musí být `>= 0`). Algoritmus iteruje 150×.

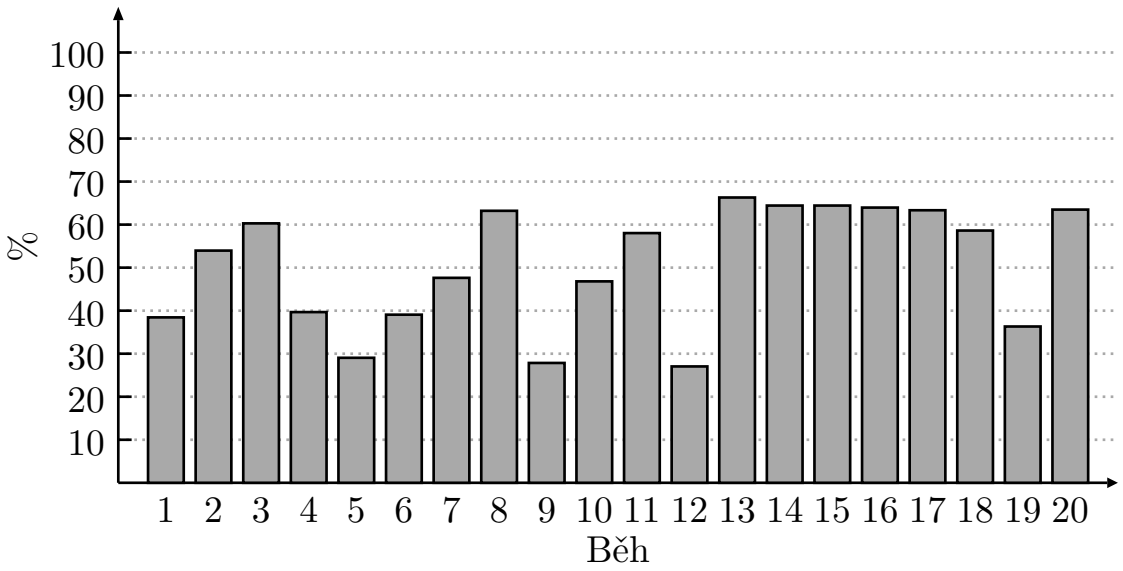
Klasifikace emailů

Dle specifikace začíná algoritmus klasifikace v metodě `test()`. Ke klasifikaci je nejprve použita metoda na počítání flagů pro všechny emaily. Poté je vyhodnocena klasifikace všech těchto emailů stejným způsobem jako při počítání limitů. Za každé porušení limitu je emailu připočten jeden strike. Aby byl email považován za spam, musí porušit alespoň 1/3 všech limitů.

```
def is_spam(self, flags):
    strikes = 0
    for flag, value in flags.items():
        if value > self.limits[flag]:
            strikes += 1
    return strikes >= round(len(self.limits) / 3)
```

Výsledky filtru

Filtr se nechová zcela deterministicky, ovšem dosahuje úspěšnosti průměrně **50,6 %** na testovacích sadách dat (viz graf). Na sadách dat automatického testování v BRUTE dosahuje úspešnosti průměrně **41,5 %**.



Rozdělení práce

Pro sdílení kódu jsme používali GitHub repozitář. Postupně jsme přidávali nové funkcionality a navzájem je vylepšovali.

Petr	**Martin**
• Analýza headerů emailu • Analýza výskytu znaků v emailu • Analýza poměru velkých a malých písmen v emailu • Algoritmus trénování filtru • Klasifikace emailů • Zpětná vazba	• Parsování emailu • Extrakce odkazů • Extrakce domén • Analýza výskytu slov a frází v předmětu emailu • Trénování filtru • Klasifikace emailů • Zpětná vazba

Závěr

Podařilo se nám úspěšně splnit zadání úlohy. Úspěšnost samotného filtru není vždy stabilní a nedosahuje takových výsledků, jaké by umožňovaly jeho reálné využití. Vylepšením stávajících algoritmů a faktorů, na základě kterých se filtr rozhoduje, by se využitelným stát ovšem mohl. Některé změny by nejspíše nemusely být ani příliš velké.

Nicméně projekt dosáhl ve větší míře očekávání a byl velmi přínosný zejména prací v týmu, ale také v jazyce Python, se kterým jsme neměli ani jeden příliš velké zkušenosti.