Název práce:

Spam filter

Autory:

Nazar Ponomarev, Nikita Kisel.

Stručný úvod:

Cílem bylo napsat program spam filter, který bude hodnotit emaily jako spam nebo ham.

Popis principu/algoritmu použitého spam filtru:

Spam filter se učí na datasetu a pak hodnotí emaily na základě získané informace. Spam filter na konci učení má dva slovníky. První je báze spam a ham odesílatelů. Druhý obsahuje slova a jejich výskyt ve spam nebo ham emailech. Hodnocení emailu probíhá najprve porovnáním adresy odesí latelů emailů, jestli ona se nachází v bázi spam a nenachází v hamu, filter hodnotí email jako spa m, a naopak jestli adresa se nachází v bázi ham a nenachází ve spamu, filter hodnotí email jako h am. Když žádná ze dvou možností nenastala filter pokračuje práce se druhým slovníkem. Pro kaž dé unikátní slovo z emailu filter počítá jeho výskyt ve spam a ham položkách slovníku. Jako výsled ek na konci filter má dva čísla - výskyt slov ve spamů a hamů. Podle těch čísel filter spočítá proce nt spamu v emailu. Když procent spamu v emailu větší než 60, filter hodnotí email jako spam. Čísl o 60 (ne 50) není náhodné, protože v případě chyby je lepší ohodnotit email jako ham. FIlter se u mí adaptovat k datasetu, proto existuje koeficient násobení. Když spam emailu více než ham emai lů, koeficient je > 1, a naopak, když spam emailu míň než ham emailů, koeficient je <1 . Přesný ko eficient spočítá poměrem číslu spam a ham emailů v datasetu. Také filter má jednu další možnost ve případě výjimky, když součet čísla výskytu slov ve spamů a hamů nulový, filter hodnotí email ja ko ham (protože v případě chyby je lepší ohodnotit email jako ham).

Popis způsobu trénování filtru:

Rozhodli jsme se, že budeme trénovat a testovat filter na doporučených datech v coursewaru. Byl o to lepší pro nás, protože ze začátku věděli, jak budou vypadat emaily a co můžeme z nich použít pro trénování. Pro trénování filter používá způsob tokenizaci emailů. Všimli jsme si, že filter může vždy vzít adresu odesílatele z emailu, protože první slovo v úhlových (<>) závorkách je vždycky ad resa emailu odesilatele. Pak jsme si všimli, že zpráva v emailu vždy jde po prvnímu "\n\n". Filter n a základě našich znalostí postupně zaplňuje z emailu slovník:

- Jestli email je ve báze 64, převádí ho do normálného textu.

- Dělá všechny písmena ve zprávě malými.

- Nahrzuje interpunkční znaky mezerami.

- Dělí zprávu podle mezer a nových řádků (\n).

- Odstraní neužitečné slova ze zprávy.

- Plnuje slovník vsimy zbývajícími slovy s délkou větší než 3 s výjimkou několika slov.

Filter přiřadí druhé polozke slova ve slovníku jeho výskyt ve spamu nebo hamu.

Výsledky dosažené naším spam filtrem:

Používali jsme jenom jeden základní typ filtru. Experimentovali s různými funkcemi a koefficenty u vnitř nich (používali jsme data z coursewaru):

1. Empiricky jsme se rozhodli o hodnocení emailu jako spam jenom z 60%, protože 50% je špatný případ pro chybové rozhodnutí. Kvalita filtru se zvýšila o asi 17%.

2. Odstranili jsme z porovnání slova s délkou menší než 3, kvalita filtru se zvýšila o asi 22%.

3. Udělali jsme báze spam a ham odesílatelů. Která zvýšila kvalitu filtru o asi 5%.

Stručný popis rozdělení práce v týmu:

	utils.py	corpus.py	tokenization.py	training_corpus.py	prediction_corpus.py	filter.py
Nazar Ponomarev	+	+	+	+		
Nikita Kisel	+	+			+	+

Stručný popis organizace práce v týmu:

Bydlíme na jedné koleji, proto dělali jsme filtr spolu ve studovně. Pro přenos modulu jsme používali Telegram (https://telegram.org).

Zhodnocení, závěr:

Práce byla velmi zajímavá, naučili jsme se pracovat s datami a práci spolu.

Seznam použité literatury, online zdrojů:

- https://www.youtube.com/watch?v=VDg8fCW8LdM
- https://www.w3schools.com/python/python_reference.asp
- https://en.wikipedia.org/wiki/Lexical_analysis#Tokenization
- https://docs.python.org/release/3.1.3/library/stdtypes.html#string-methods
- https://docs.python.org/release/3.1.3/library/re.html?highlight=re#module-re