
Západočeská univerzita v Plzni

Umělá inteligence a rozpoznávání

KIV/UIR

# Dokumentace semestrální práce

Vojtěch Daniš́ík
A16B0019P
danisik@students.zcu.cz
doba řešení: 10 dní

11. května 2018

# Obsah

# 1 Popis problémů

## 1.1 Zadání

Zadání semestrální práce: Ve zvoleném programovacím jazyce navrhněte a implementujte program, který umožní automaticky detekovat události z krátkých textových zpráv. Implementujte alespoň tři různé algoritmy pro tvorbu příznaků reprezentující textovou zprávu, alespoň dvě různé metody detekce údálostí dle vlastní volby. Ohodnoťte kvalitu detekčních algoritmů na anotovaných datech (použijte metriky přesnost, úplnost a F-míra).

## 1.2 Popis problémů

V této úloze lze najít několik problémů:

- První problém této úlohy spočívá v reprezentaci textové zprávy. Čím lepší reprezentace zpráv bude, tím rychlejší a přesnější bude výsledek detekčních metod.
- Druhý problém této úlohy spočívá ve vhodně zvolené metodě počítající vzdálenost mezi dvěma body. Špatná volba metody může znamenat znepřesnění detekčních metod.
- Poslední problém je ve výběru detekčních algoritmů. Pokud se zvolí nevyhovující detekční algoritmus, tak se to velice projeví na výsledné anotaci a kvalitě.

# 2 Analýza problému

V 1. kapitole jsme si popsali všechny problémy, které tato úloha přináší. V této kapitole se budeme věnovat jejich analýze.

## 2.1 Reprezentace textové zprávy

Abychom mohli textovou zprávu reprezentovat, budeme potřebovat vytvořit příznaky.
Pro tvorbu příznaků lze použít model **Bag of Words** (BoW). V tomto modelu je dokument/text reprezentován jako "taška" slov, ve které jsou obsaženy všechny různá slova (zachovává mnohočetnost).
Jako další algoritmus pro tvorbu příznaků lze použít **Term Frequency** (TF) a **Term Frequency- Inverse Document Frequency** (TF-IDF). Metoda TF vyjadřuje, jak často se slovo vyskytuje dokumentu/textu. Hodnota TF

pro slovo v dokumentu se vypočítá pomocí vzorce na obrázku č. 1, kde čitatel je počet výskytů slova, pro které se počítá hodnota TF v aktuálně prohledávaném dokumentu/textu, a jmenovatel je součet výskytů všech slov v prohledávaném dokumentu.

$$\mathrm{tf_{i,j}} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

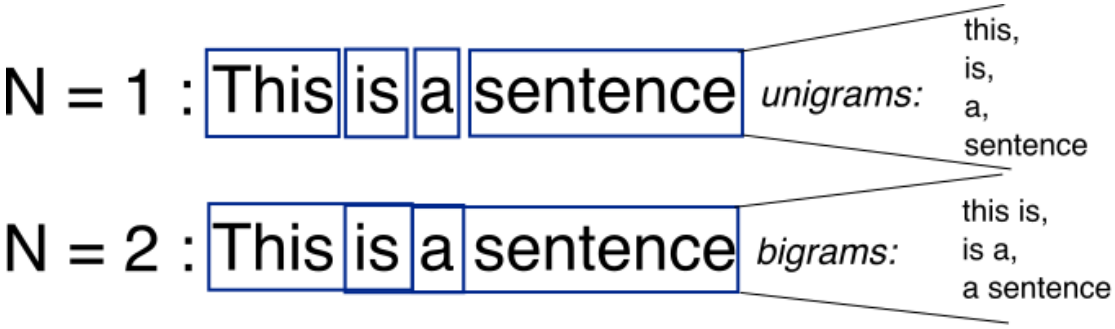
Obrázek 1: Výpočet hodnoty TF

Metoda TF-IDF je spojení TF a Inverse Document Frequency, jejíž hodnota reprezentuje "důležitost"slova. Čím více se slovo vyskytuje v dokumentech, tím méně je důležité .Hodnota IDF pro slovo v dokumentu se vypočítá pomocí vzorce na obrázku č. 2, kde čitatel je velikost databáze dokumentů a jmenovatel je počet dokumentů, které obsahují slovo, pro které se IDF hodnota počítá.

$$\mathrm{idf_i} = \log \frac{|D|}{|\{j : t_i \in d_j\}|}$$

Obrázek 2: Výpočet hodnoty TF-IDF

Další metoda pro vytvoření příznaků z textové zprávy se nazývá **N-gram**. N-gram je definován jako sled N po sobě jdoucích položek z dané posloupnosti. Tato posloupnost může být posloupností slov, písmen nebo čehokoliv jiného. Ukázka fungování N-gramu na obrázku č. 3




Obrázek 3: Ukázka fungování N-gramu

## 2.2 Výpočet vzdálenosti mezi dvěma body

Pro výpočet vzdálenosti mezi dvěma body lze využít několik metod, nejznámější a nejpoužívanější jsou: **Euklidovská vzdálenost** a **Manhattanská vzdálenost**.
Euklidovská vzdálenost je přímá vzdálenost mezi dvěma body v euklidovském prostoru. Vzorec pro výpočet euklidovské vzdálenosti mezi dvěma body je zobrazen na obrázku č. 4.

$$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\| = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}.$$

Obrázek 4: Vzorec pro výpočet euklidovské vzdálenosti

Manhattanská vzdálenost je součet absolutních rozdílů dvou zadaných vektorů v kartézské soustavě souřadnic. Vzorec pro výpočet manhattanské vzdálenosti mezi dvěma body je zobrazen na obrázku č. 5.

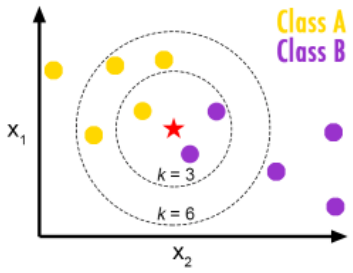
$$d_1(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_1 = \sum_{i=1}^{n} |p_i - q_i|,$$

Obrázek 5: Vzorec pro výpočet manhattanské vzdálenosti

## 2.3 Detekční algoritmy

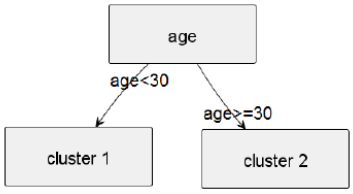
### 2.3.1 Detekční algoritmy využívající metodu Učení s učitelem

Mezi tyto algoritmy patří například: K-nejbližší sousedé (K-NN), Rozhodovací strom (Decision trees).
**K-NN**: Na startu algoritmu se trénovací anotovaná data rozdělí do předpřipravených skupin. Následně se do algoritmu předají neanotovaná data. Samotný průběh algoritmu spočívá ve vypočítání vzdálenosti neanotovaného prvku se všemi anotovanými. Algoritmus následně vybere K nejbližších sousedů a podle nich se určí třída, do které má být neanotovaný prvek zařazen. Příklad na obrázku č. 6.




Obrázek 6: Ukázka algoritmu K-NN

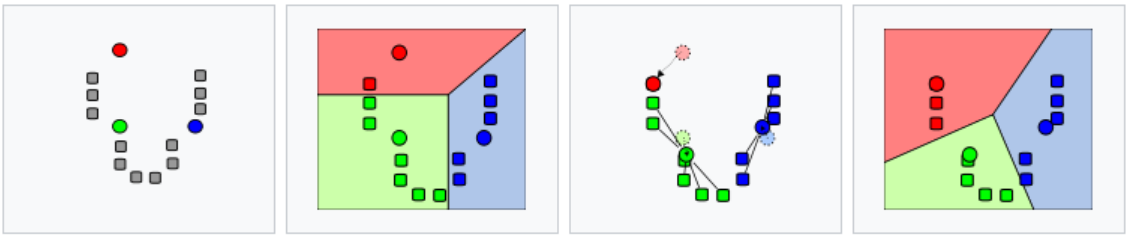
**Rozhodovací strom**: U tohoto algoritmu se vytváří strom, jehož uzly reprezentují rozhodování podle jedné (vybrané) vlastnosti objektu. Tento strom se musí vytvořit z množiny daných objektů, které poskytne učitel/jiný algoritmus. Největším problémem tohoto algoritmu je vytvoření samotného stromu. Strom totiž musí co nejlépe odlišovat objekty, aby byl algoritmus užitečný. Příklad na obrázku č. 7.




Obrázek 7: Ukázka algoritmu Decision trees

### 2.3.2 Detekční algoritmy využívající metodu Učení bez učitele

Mezi tyto algoritmy patří například: K-means.
**K-means**: Algoritmus předpokládá, že shlukované objekty lze chápat jako body v euklidovském prostoru a že počet shluků K je předem dán. Shluky jsou definovány svými centroidy, což je střed shluku vypočítaný ze všech bodů v jednom shluku. Algoritmus postupuje iterativně tak, že se vyjde z nějakých centroidů, přiřadí do nich body, přepočítá centroidy do té doby, než se poloha centroidů ustálí. Příklad na obrázku č. 8.



Obrázek 8: Detekování pomocí algoritmu K-means

# 3 Návrh řešení

Jako parametrizační metody vyberu BoW, TF a TF-IDF (N-gram není pro tuto úlohu ideální z důvodu nevyhovující reprezentace slova). Pro výpočet vzdálenosti mezi dvěma vektory použiji Euklidovskou vzdálenost z důvodu přesnějšího výpočtu při počítání vzdálenosti mezi dvěma vektory. Jako detekční algoritmy použiji K-Means a K-NN (pro K = 1) pro zobrazení rozdílů mezi metodou **Učení s učitelem** (K-NN) a metodou **Učení bez učitele** (K-Means).

# 4 Popis řešení

Pro reprezentaci slovníku využiji HashMapu a jako klíč použiji slovo. Tím zajistím, že se ve slovníku každé slovo bude vyskytovat pouze jednou.

## 4.1 Popis tříd

### 4.1.1 Cluster

Třída, která reprezentuje jednotlivé shluky. Má 2 důležité atributy: vektor reprezentující střed shluku (double[ ]) a list tweetů patřících do clusteru.

### 4.1.2 EventType

Enumerate třída, obsahuje všechny typy eventů. Typy eventů:

- POLICY - politika (po)
- INDUSTRY - průmysl (pr)
- AGRICULTURE - zemědělství (ze)
- SPORT - sport (sp)
- CULTURE - kultura (ku)
- CRIMINALITY - kriminalita (kr)
- WEATHER - počasí (pc)
- OTHER - jiné (ji)
- NOT_EVENT - není event (-)

### 4.1.3 Quality

Třída sloužící pro výpočet kvality detekčních metod. Používané metriky jsou **přesnost** (precision), **úplnost** (recall) a **F-míra** (F-measure).

### 4.1.4 Tweet

Třída reprezentující tweet. Důležité atributy u této třídy jsou: vektor reprezentující tweet (double [ ]), typ eventu určený ze souboru (EventType), typ eventu určený detekční metodou (EventType).

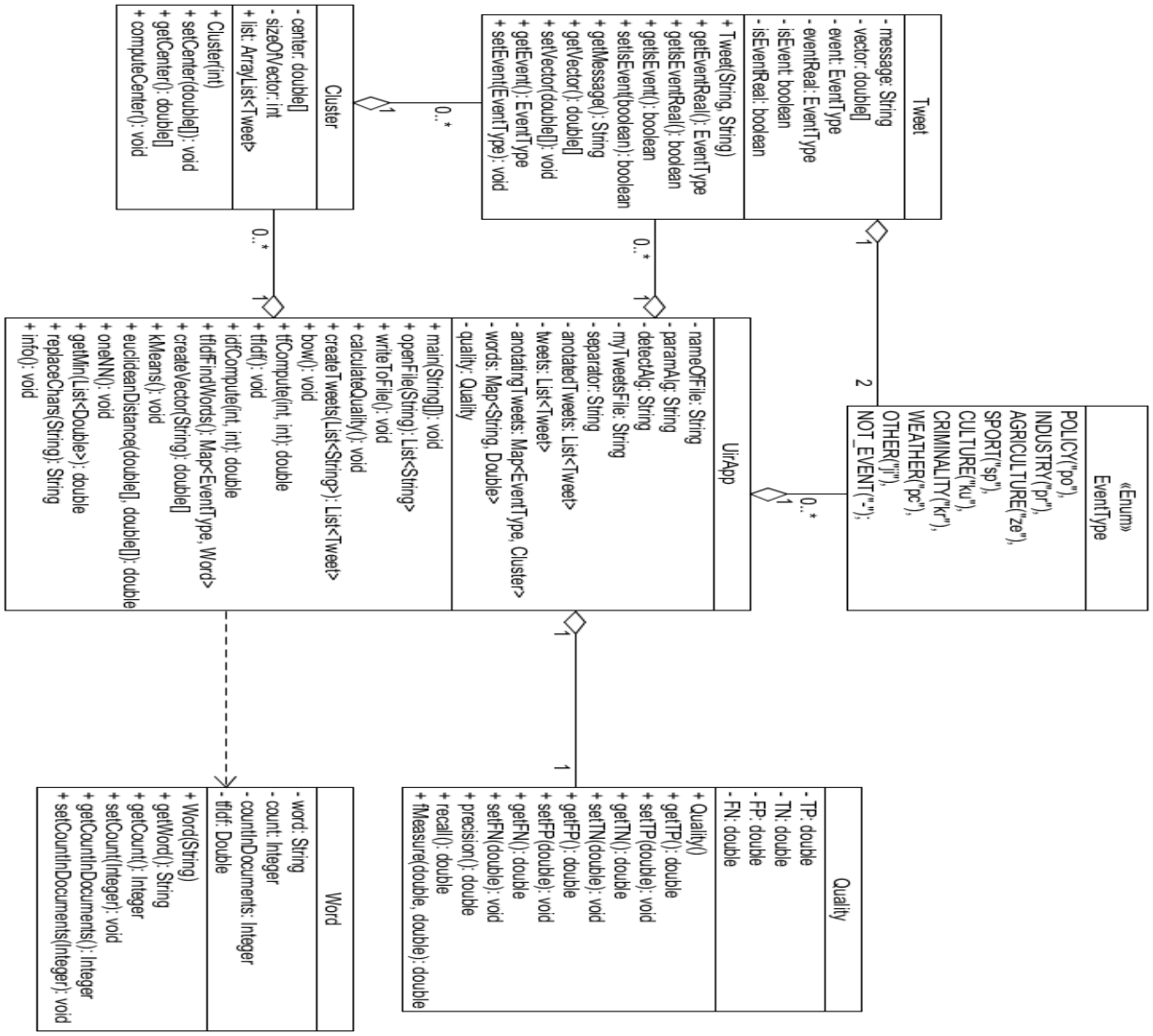
### 4.1.5 Word

Pomocná třída, která slouží pro uložení slova při použití metody TF-IDF. Důležité parametry u této metody jsou: slovo (String) a číslo reprezentující v kolika dokumentech se toto slovo nachází (Integer).

### 4.1.6 UirApp

Hlavní třída, která spouští veškeré metody. Provádí se zde načtení tweetů ze souborů do slovníku pomocí metody určené uživatelem při spouštění programu a provedení detekční metody také určené uživatelem. Po ukončení detekční metody se vypíšou všechy shluky s tweetama, provede se uložení do souboru a vypočítají se metriky pro určení kvality použité detekční metody. Tweety anotované detekční metodou se porovnávají s anotací, kterou provedl uživatel ještě před spuštěním aplikace.

## 4.2 UML



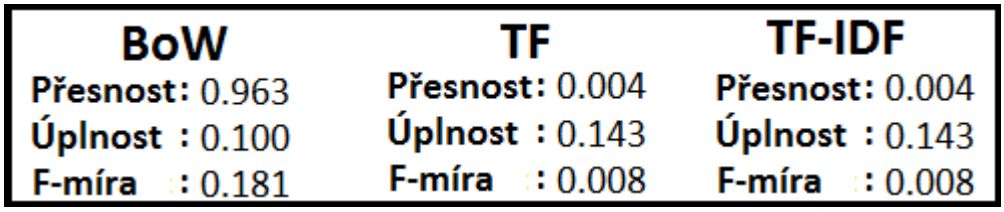

Obrázek 9: UML diagram pro UirApp

# 5 Uživatelská dokumentace

Program se spustí následujícím příkazem:
java -jar A16B0019P.jar <*datovy-soubor.csv*> <*parametrizacni-algoritmus*> <*detekcni-algoritmus*>
Formát souboru: **CSV** (**Comma-separated values**) - jednoduchý souborový formát pro výměnu tabulkových dat. Soubor ve formátu CSV sestává z řádků, ve kterých jsou jednotlivé položky odděleny znakem (v našem případě) středníkem (**;**). Soubor musí mít tuto hierarchii - zda jde o událost nebo ne (1 - ano, 0 - ne); typ události; id tweetu; jazyk; datum a čas; zpráva. Aplikace se nijak neovládá, pouze se spustí se zadanými parametry. Jako parametrizační algoritmus si lze zvolit **BoW**, **TF**, **TF-IDF**, jako detekční algoritmy si lze zvolit **K-Means** a **1-NN**.
Program po spuštění se správnými parametry zavolá parametrizační algoritmus, který vytvoří slovník všech slov ze všech tweetů a zavolá detekční algoritmus, který tweety rozdělí do 9 shluků (9 typů událostí). Po dokončení rozdělování tweetů do shluků následně program vypíše všechny shluky s tweetama a také ohodnotí kvalitu detekčního algoritmu třemi metrikami (přesnost, úplnost, F-míra), kde se porovná naše anotování a anotování detekčního algoritmu. Následně program vypíše do souboru **results.csv** výsledky anotace. Soubor má tuto hierarchii - anotace detekčního algoritmu; ruční anotace; tweet.

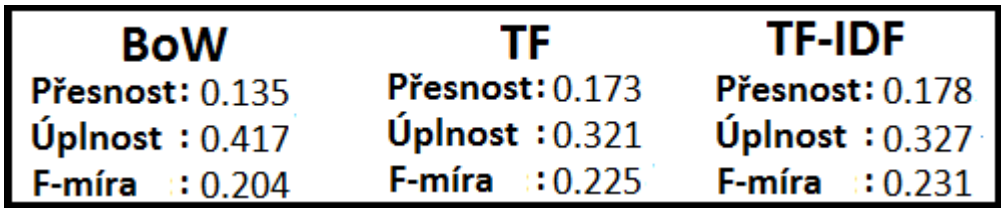
# 6 Závěr

Aplikaci jsem testoval pomocí anotovaného souboru složeného ze 2 anotovaných souborů (mého a mého spolužáka) o velikosti 400 tweetů. Hodnotila se kvalita detekčního algoritmu pomocí tří metrik (přesnost, úplnost, F-míra). Výsledky mého programu jsou na obrázcích č. 10, 11.




Obrázek 10: Výsledky pro metodu K-Means

Jelikož při používání metody K-Means vybíráme náhodné první středy shluků, výsledky nebudou nikdy stejné, jak tomu je u metody 1-NN, proto nemůžeme brát vypočítané metriky vážně.Navíc, většina výsledných přeností nám vychází velice malá (do 1 %), proto metoda K-means není vyhovující jako klasifikační metoda.




Obrázek 11: Výsledky pro metodu 1-NN

U metody 1-NN si můžeme všimnout, že pro většinu parametrizačních metod nám vychází skoro stejná přesnost i úplnost (mimo BoW, tam je rozdíl o něco větší). Metoda 1-NN podle výsledků je vyhovující jako klasifikační metoda, záleží pouze na reprezentaci tweetů (čím lepší reprezentace tweetu, tím lepší přesnost metody 1-NN).