# Program na převedení kolekce textových dokumentů do vektorové reprezentace

Vstupem je textový soubor, jehož jednotlivé řádky reprezentují textové dokumenty (jeden řádek = jeden dokument). Řádky mají následující strukturu:

C\tTEXT\n

kde C je třída dokumentu (několik znaků), \t je tabelátor, TEXT je posloupnost libovolných znaků reprezentující obsah dokumentu, \n je znak konce řádku.

Z každého textu se odeberou HTML a obdobné tagy, entity, znaky, které nejsou písmeny (čísla, speciální znaky), a každý text se převede na tzv. *bag-of-words* reprezentaci, neboli posloupnost slov, kde pořadí není důležité.

Z každého textu se následně vytvoří vektor, kde jednotlivé prvky vektoru (atributy) odpovídají jednotlivým slovům v celé kolekci a hodnoty ve vektoru budou záviset na výskytu daných slov v tomto textu. Hodnota (váha i-tého slova v j-tém dokumentu) je součinem dvou vah – lokální a globální, a normalizačního faktoru:

$$w_{ij} = l_{ij} * g_i * n_j$$

Lokální váhy mohou být dvojího typu:

- TP (term presence) – přítomnost slova v dokumentu (1 = ano, 0 = ne),
- TF (term frequency) – počet výskytů slova v dokumentu (0 a víc).

Globální váha bude pouze jedna:

- IDF (inverse document frequency) – viz dále,
- je možnost globální váhu neuplatnit, v tom případě má tato váha hodnotu 1 (hodnota atributu pro daný dokument tak bude mít pouze hodnotu lokální váhy).

Poslední hodnotou vektoru je třída dokumentu. Každý dokument je tedy reprezentován vektorem o délce N+1, kde N je počet unikátních slov ve všech dokumentech (+1 je pro třídu dokumentu).

Pokud to bude požadováno, mohou být ze všech textů odebrána slova, která mají četnost ve všech dokumentech dohromady nízkou (např. 1). Mohou být také odebrána slova, která mají určitý malý počet znaků.

Všechny informace, které ovlivňují způsob zpracování dokumentů, stejně jako jména vstupních a výstupních souborů budou předány jako parametry skriptu. Tyto parametry budou zpracovány s využitím libovolného modulu ze sítě CPAN. Pokud nejsou některé parametry zadány, bude se pracovat s nějakými implicitními hodnotami.

Příklad vstupního souboru (fiktivní uživatelské recenze produktu):

```
_P	good :-)
_P	very good.
_N	<b>not</b> good
_N	bad!!!
_N	very, very bad
```

Příklad výstupního souboru (minimální délka slova = 1, minimální výskyt slova = 1, hodnota ve vektoru = TF):

| GOOD | VERY | NOT | BAD | _CLASS_ |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | _P |
| 1 | 1 | 0 | 0 | _P |
| 1 | 0 | 1 | 0 | _N |
| 0 | 0 | 0 | 1 | _N |
| 0 | 2 | 0 | 1 | _N |

Příklad výstupního souboru (minimální délka slova = 4, minimální počet výskytu slova = 2, hodnota TP):

| GOOD | VERY | _CLASS_ |
|---|---|---|
| 1 | 0 | _P |
| 1 | 1 | _P |
| 1 | 0 | _N |
| 0 | 0 | _N |
| 0 | 1 | _N |

**TF-IDF (term frequency-inverse document frequency) schéma**

Tento přístup je založen na myšlence, že čím vícekrát se slovo (term) v dokumentu vyskytuje, tím je důležitější (TF faktor), a čím méně častěji se slovo vyskytuje ve všech dokumentech, tím více je specifické a tudíž důležité (IDF faktor). Inverzní frekvence výskytu termu v dokumentech (inverse document frequency) se vypočítá jako

$$IDF(t_i) = \log ( N/n(_{ti}) ),$$

kde $t_i$ je i-tý term, N je počet všech dokumentů a $n(t_i)$ je počet dokumentů obsahujících term $t_i$, ($n(t_i)$ se nazývá frekvence výskytu termu v dokumentech, document frequency).

**Normalizace**

Aby se zabránilo nadhodnocení termů v dlouhých dokumentech (ve kterých se vyskytuje větší množství termů), mohou být vektory normalizovány. Jedním ze způsobů normalizace je vydělit všechny váhy jejich součtem:

$$wn_{ij} = \frac{w_{ij}}{\sum_{i=1}^{n} wij}$$

# Program pro realizaci algoritmu pro šíření aktivace v grafu

Šíření aktivace, je grafovou metoda (metoda operující nad grafy) mající svůj původ v kognitivních vědách, kde je využívána k porozumění fungování paměti a procesu formování a učení se biologických sítí.

V současnosti existuje celá řada metod spadajících do této rodiny aplikací, existuje velké množství modifikací a tím i velké množství uplatnění. Metody šíření aktivace jsou tak často nasazovány při numerických simulacích fyzikálních jevů, v robotice, v epidemických modelech, při vyhledávání informací (information retrieval) či v systémech pro doporučování.

Síť je potřeba popsat pomocí nějakého formálního prostředku. Výhodným, matematicky podloženým přístupem je teorie grafů. Celý systém (síť) je pak reprezentován orientovaným grafem. Každý orientovaný graf je charakterizován následujícími základními funkcemi:

- init(A, V) – funkce přiřazuje každé orientované hraně počáteční uzel,
- term(A, V) – funkce přiřazuje každé orientované hraně koncový uzel,
- F(V) – funkce přiřazuje každému uzlu grafu aktivační hodnotu a > 0.

Všechny objekty v sítí (např. lidé) jsou reprezentovány uzly. Uzel je charakterizován svým označením (identifikátorem), který jednoznačně umožňuje identifikovat každý z uzlů. Tento identifikátor je použit také pro definici počátečních a koncových uzlů pro vazby a může být užitečný také například při vizualizaci grafové struktury. Typ uzlu umožňuje rozlišovat mezi různými druhy uzlů a klasifikovat je do tříd. Například v systémech pro doporučování založených na využívání tagů lze rozlišit čtyři druhy objektů – aktory, zdroje, tagy a instance tagování (přiřazení tagu ke konkrétnímu zdroji určitým aktorem).

Vazba reprezentuje orientované spojení mezi dvěma uzly. Váha vazby kvantifikuje tento vztah a je typicky použita jako základ pro výpočet útlumu signálu reprezentujícího šířící se aktivační hodnotu v rámci aplikace algoritmů šíření aktivace. Hodnota menší než 1 představuje zeslabení signálu, hodnota větší než 1 signál zesiluje.

Další vlastnost vazeb, tzv. reciprocita, definuje, zda vazba mezi dvěma uzly je reciproční (vzájemná). Znamená to, že pokud mezi uzly $N_1$ a $N_2$ existuje vazba typu $T_a$ a tato vazba je reciproční, automaticky existuje také vazba typu $T_a r$ mezi uzly $N_2$ a $N_1$. Pokud je například mezi osobou $O_1$ a $O_2$ vztah takový, že osoba $O_1$ je rodičem osoby $O_2$, pak automaticky existuje vazba mezi uzly $O_2$ a $O_1$ představující skutečnost, že osoba $O_2$ je potomkem osoby $O_1$. Vazba $O_1 \rightarrow O_2$ může být zároveň jiného typu než vazba $O_1 \leftarrow O_2$, protože v určitých situacích může být třeba rozlišit mezi charakterem vztahu rodič → potomek a potomek ← rodič. Tento způsob chápání recipročních vazeb přináší výhodu zejména v procesu vytváření struktury reprezentující sociální síť, protože není nutné explicitně definovat vazby automaticky vedoucí opačným směrem často pro podstatnou část sítě.

Struktura grafu a parametry algoritmu jsou zadány v konfiguračních souborech. V nich jsou řádky následujícího typu, každý řádek začíná určitým klíčovým slovem (keyword), na základě kterého se rozhodne, co na řádku je. Řádky začínající znakem # jsou komentáře a jsou ignorovány:

```
# typy uzlů (nt = node type)
# keyword       jméno typu
nt              Osoba

# hrany (lt = link type)
# keyword       jméno typu
lt              Přítel

# keyword,jméno typu,reciproční vazba
ltr     Přítel  Přítel
# automaticky přidává k hraně typu Přítel hranu s opačnou orientací, typu Přítel

# váha vazby (lw = link weight), vztahuje se k typu vazeb
# keyword       typ vazby       váha
lw              Přítel          0.8
```

```
# konkrétní uzly
# keyword       jméno uzlu      typ uzlu
n               A1              Osoba
n               A2              Osoba
...

# konkrétní hrany
# keyword       počátek         konec   typ vazby
l               A1              A2      Přítel
l               A1              A5      Přítel
...

# počáteční hodnoty aktivace uzlů (ia = initial activation)
# keyword       uzel    aktivace
ia              A1      1
...
...
```

Algoritmy šíření aktivace iterativně předávají aktivaci (tzn. hodnotu přiřazenou každému uzlu) z uzlů, kterým byla na počátku přiřazena určitá výchozí hodnota aktivace, směrem k dalším uzlům v síti prostřednictvím svých výstupních vazeb. Tento proces obvykle pokračuje do doby, než je dosaženo určitého rovnovážného stavu, tzn. celý systém se stabilizuje na úrovni limitní distribuce aktivace, nebo jakmile je splněna určitá podmínka ukončující celý proces, například počet iterací.

Základní algoritmus šíření aktivace má následující kroky:

- Inicializace (vstup) – vytvoření grafu na základě parametrů (pomocí vhodné struktury v paměti), nastavení parametrů algoritmu, vlastností sítě a počáteční aktivace vybraných uzlů (přiřazení číselných hodnot uzlům) a vytvoření seznamu uzlů s počáteční aktivací.
- Iterace
    a) poslání signálu z aktivních uzlů (hodnota těchto uzlů > 0) uzlům, které jsou přímo dosažitelné přes hranu z těchto aktivovaných uzlů
        - hodnota odesílaná do každé hrany se vypočítá podle vzorce
            Xi * 1 / outdegree(Xi)^Beta,
          kde Xi je hodnota aktivace uzlu i, outdegree(i) je počet výstupních hran z uzlu i a Beta je parametr zadaný v konfiguračním souboru (^ je operace umocnění)
        - při průchodu hranou může signál ztratit část své hodnoty, tzn. hodnota na vstupu hrany je vynásobena vahou hrany (podle jejího typu)
        - nová hodnota uzlu je vypočítána podle vzorce
            a*X(i) + b*Input(i) + c*Output(i),
          kde X(i) je původní hodnota aktivace uzlu i, Output(i) je suma signálu poslaného z uzlu i do všech jeho výstupních hran a Input(i) je součet hodnot signálu přicházejících ze všech vstupních hran do uzlu i. A, b, c jsou parametry
        - výše uvedené operace se dějí zároveň pro celou síť, tzn. nejprve ze všech uzlů odejde určitá úroveň aktivace = na počátku vazeb se objeví nějaká hodnota, pak je tato hodnota utlumena díky váze vazby a pak do všech uzlů přiteče nějaká hodnota z jeho vstupních vazeb
    b) kalibrace
    c) redukce seznamu aktivovaných uzlů – odstranění uzlů, jejichž hodnota aktivace je menší, než prahová hodnota t, která je zadaná jako parametr
    d) výstup – seznam všech uzlů v síti a hodnot jejich aktivace
    e) rozhodnutí o ukončení iterace

**Kalibrace (normalizace)**

Normalizace je procesem, při kterém dochází k lineární transformaci hodnot aktivace uzlů tak, aby došlo ke splnění podmínek normalizace. Normalizací je umožněno například srovnat proces šíření aktivace v rámci různých sítí a vyrovnat stav vzniklý odebráním některých uzlů z procesu šíření aktivace.

Možnosti kalibrace:

- Žádná kalibrace.
- Zachování sumy hodnot uzlů, které byly aktivovány na počátku (suma aktivace uzlů, které byly na počátku aktivovány, musí zůstat konstantní, rovna sumě počáteční aktivace; poměry hodnot aktivace všech uzlů v sítí před a po kalibraci musí zůstat stejné). Pokud má síť např. 5 uzlů a po jedné iteraci šíření aktivace mají hodnoty A = 5, B = 4, C = 2, D = 0, E = 2 a na počátku celého procesu byly aktivovány uzly B a C a oběma byla přiřazena hodnota 1, musí po kalibraci (normalizaci) být suma aktivací uzlů B a C zůstat 2 (1 + 1). Uzel B tedy bude mít hodnoty 1,33333 a uzel C hodnotu 0,666667. Poměr těchto hodnot je 4 : 2 a součet je 2. Upraví se i hodnoty aktivace uzlů A a E (D je 0) tak, aby poměr mezi hodnotami aktivace zůstal zachován.
- Zachování celkové sumy hodnot aktivací uzlů (suma aktivace uzlů v celé síti musí zůstat konstantní, rovna sumě počáteční aktivace; poměry hodnot aktivace jednotlivých uzlů před a po kalibraci musí zůstat stejné). Pokud má síť např. 5 uzlů a po jedné iteraci šíření aktivace mají hodnoty A = 5, B = 4, C = 2, D = 0, E = 2 a na počátku celého procesu byly aktivovány uzly B a C a oběma byla přiřazena hodnota 1, musí po kalibraci (normalizaci) být suma aktivací všech uzlů 2 (1 + 1). Upraví se hodnoty aktivace uzlů A, B, C a E (D je 0) tak, aby poměr mezi hodnotami aktivace zůstal zachován.

Iterace se ukončí po určitém počtu opakování, které je zadáno jako parametr.

Vzorový konfigurační soubor je zadán jako příloha zadání, je obsažen i vzor výstupů pro všechny tři výše uvedené možnosti kalibrace.