KATEDRA INFORMATIKY
UNIVERZITA PALACKÉHO V OLOMOUCI
www.inf.upol.cz

Jan Laštovička
https://www.inf.upol.cz/lide/jan-lastovicka
jan.lastovicka@upol.cz
17. listopadu 12, 771 46 Olomouc

Databáze ◇ poznámky k přednášce

12. Pokročilé podmínky

verze z 16. prosince 2024

V příkladech budeme uvažovat indexovou proměnnou `movie` danou: „Film d má název D.“ s hodnotou I_1:

$$\{\langle 1, \texttt{"The Fellowship of the Ring"}\rangle,\\ \langle 2, \texttt{"The Two Towers"}\rangle,\\ \langle 3, \texttt{"The Return of the King"}\rangle\}$$

1 Konjunkce termů

Vezměme podmínky $\theta_1, \ldots, \theta_n$. Pak

$$\theta_1 \wedge \cdots \wedge \theta_n$$

je podmínka, kterou dokument splňuje, jestliže splňuje všechny podmínky $\theta_1, \ldots, \theta_n$. Například dokument `"The Fellowship of the Ring"` splňuje podmínku `ring` $\wedge$ `fellowship`. Dále $\sigma_{\texttt{of}\wedge\texttt{ring}}(I_1)$ je rovno:

$$\{\langle 1, \texttt{"The Fellowship of the Ring"}\rangle\}.$$

a dáno: „Film d má název D, který obsahuje slovo `of` a slovo `ring`.“

Vezměme dokument Q, pak řetězec:

```
{
  "match": {
    "text": {
      "query": Q,
      "operator": "AND"
    }
  }
}
```

zapisuje podmínku $q_1 \wedge \cdots \wedge q_n$, kde $\mathrm{T}(Q) = (q_1, \ldots, q_n)$.

Například:

```
GET /movie/_search
{
```

```
  "query": {
    "match": {
      "text": {
        "query": "OF King",
        "operator": "AND"
      }
    }
  }
}

{
  "hits": {
    "hits": [
      {
        "_id": "3",
        "_source": {
          "text": "The Return of the King"
        }
      }
    ]
  }
}
```

Kvalita splnění podmínky $\theta_1 \wedge \cdots \wedge \theta_n$ je dána:

$$score(I, D, \theta_1 \wedge \cdots \wedge \theta_n) = \sum_{i=1}^{n} score(I, D, \theta_i)$$

Například dokument v předchozím příkladu splnil podmínku s kvalitou přibližně $1.3 = 0.4 + 0.9$.

2 Obecné složené podmínky

Vezměme podmínky $\theta_1, \ldots, \theta_n$. Pak podmínku $\theta_1 \wedge \cdots \wedge \theta_n$ zapíšeme řetězcem:

```
{
  "bool": {
    "must": [
      θ1,
      ⋮
      θn
    ]
  }
}
```

Například podmínku (`fellowship` $\vee$ `two`) $\wedge$ (`ring` $\vee$ `towers`) zapíšeme řetězcem:

```
{
  "query": {
      "bool": {
        "must": [
          {
            "match": {
              "text": "fellowship two"
            }
          },
          {
            "match": {
              "text": "ring towers"
            }
          }
        ]
      }
    }
}
```

Podmínku $\theta_1 \vee \cdots \vee \theta_n$ zapíšeme řetězcem:

```
{
  "bool": {
    "should": [
      θ1,
      ⋮
      θn
    ],
    "minimum_should_match": 1
  }
}
```

Například podmínku (`ring` $\vee$ `fellowship`) $\wedge$ (`towers` $\vee$ `two`) zapíšeme řetězcem:

```
{
  "bool": {
    "should": [
      {
        "match": {
          "text": {
            "query": "ring fellowship",
            "operator": "AND"
          }
```

```
        }
      },
      {
        "match": {
          "text": {
            "query": "towers two",
            "operator": "AND"
          }
        }
      }
    ],
    "minimum_should_match" : 1
  }
}
```

Pro podmínku θ je $\neg\theta$ podmínka, kterou dokument splňuje, jestliže nesplňuje podmínku θ. Například dokument `"The Return of the King"` splňuje podmínku $\neg$`ring`.

Podmínku $\neg\theta$ zapíšeme řetězcem:

```
{
  "bool": {
    "must_not": θ
  }
}
```

Například podmínku $\neg$`ring` zapíšeme:

```
{
  "bool": {
    "must_not": {
      "match": {
        "text": "ring"
      }
    }
  }
}
```

Kvalita splnění podmínky $\neg\theta$ je $score(I, D, \neg\theta) = 0$. Například máme:

$$\begin{aligned} & score(I_1, \texttt{"The Return of the King"}, \texttt{of} \wedge \neg\texttt{ring}) \\ & = score(I_1, \texttt{"The Return of the King"}, \texttt{of}) \\ & \approx 0.44 \end{aligned}$$

Řetězec:

```
{
  "bool": {
    "must_not": [
      θ1,
      ⋮
      θn,
    ]
  }
}
```

je zkratkou za $\neg(\theta_1 \vee \cdots \vee \theta_n)$.

Například řetězec:

```
{
  "bool": {
    "must_not": [
      {
        "match": {
          "text": "ring"
        }
      },
      {
        "match": {
          "text": "king"
        }
      }
    ]
  }
}
```

zapisuje podmínku $\neg(\texttt{ring} \vee \texttt{king})$.

Každý dokument splňuje podmínku 1, kterou zapíšeme řetězcem:

```
{
  "match_all": {}
}
```

Kvalita splnění podmínky 1 je $score(I, D, 1) = 1$.

3 Zobecněná disjunkce

Vezměme podmínky $\theta_1, \ldots, \theta_m$ a číslo $0 \leq n \leq m$, pak

$$\bigvee_n(\theta_1, \ldots, \theta_m) = \alpha_1 \vee \cdots \vee \alpha_k,$$

kde α_i jsou všechny možné konjunkce tvaru $\theta_{j_1} \wedge \cdots \wedge \theta_{j_n}$ takové, že $j_1 < \cdots < j_n$. Platí, že $k = \binom{m}{n}$. Například:

$$\alpha_1 = \bigvee_2(\texttt{ring}, \texttt{king}, \texttt{return}) = (\texttt{ring} \wedge \texttt{king}) \vee (\texttt{ring} \wedge \texttt{return}) \vee (\texttt{king} \wedge \texttt{return})$$

Speciálně dostáváme, že:

$$\bigvee_0(\theta_1, \ldots, \theta_m) = 1,$$
$$\bigvee_1(\theta_1, \ldots, \theta_m) = \theta_1 \vee \cdots \vee \theta_m,$$
$$\bigvee_m(\theta_1, \ldots, \theta_m) = \theta_1 \wedge \cdots \wedge \theta_m.$$

Pro $n \neq 0$ řetězec:

```
{
  "bool": {
    "should": [ θ1, ..., θm ],
    "minimum_should_match": n
  }
}
```

zapisuje podmínku $\bigvee_n(\theta_1, \ldots, \theta_m)$. Například výše uvedenou podmínku α_1 zapíšeme:

```
{
    "bool": {
      "should": [
        {
          "match": {
            "text": "ring"
          }
        },
        {
          "match": {
            "text": "king"
          }
        },
        {
          "match": {
            "text": "return"
          }
        }
      ],
    "minimum_should_match" : 2
  }
}
```

Kvalita splnění podmínky $\bigvee_n(\theta_1, \ldots, \theta_m)$ dokumentem D z indexu I je dána vztahem:

$$score(I, D, \bigvee_n(\theta_1, \ldots, \theta_m)) = \sum_{i=1}^{m} score(I, D, \theta_i)$$

4 Obecná podmínka `bool`

Vezměme podmínky $\alpha_1, \ldots, \alpha_m, \beta_1, \ldots, \beta_n, \gamma_1, \ldots, \gamma_k$ a číslo $0 \leq l \leq n$, takové, že $l = 0$ implikuje $m > 0$, pak řetězec:

```
{
  "bool": {
    "must": [ α1, ..., αm ],
    "should": [ β1, ..., βn ],
    "minimum_should_match": l
    "must_not": [ γ1, ..., γk ],
  }
}
```

je zkratkou za

$$\alpha_1 \wedge \cdots \wedge \alpha_m \wedge \left(\bigvee_l (\beta_1, \ldots, \beta_n) \right) \wedge \neg(\gamma_1 \vee \cdots \vee \gamma_k).$$

Položka:

```
"must": α
```

je zkratkou za:

```
"must": [ α ]
```

Podobně pro položky `"should"` a `"must_not"`. Dále chybějící položka `"must"` je zkratkou za:

```
"must": []
```

chybějící položka `"should"` je zkratkou za:

```
"should": []
```

pokud $k = 0$, jinak:

```
"should": 1
```

Chybějící položka `"minimum_should_match"` je zkratkou za:

```
"minimum_should_match": p
```

kde $p = 0$ pokud $m > 0$, jinak $p = 1$. Chybějící položka `"must_not"` je zkratkou za:

```
"must_not": []
```

Například:

```
{
  "bool": {
    "must": {
      "match": {
        "text": "of"
      }
    },
    "must_not": {
      "match": {
        "text": "ring"
      }
    }
  }
}
```

je zkratkou za:

```
{
  "bool": {
    "must": [
      {
        "match": {
          "text": "of"
        }
      }
    ],
    "should": [],
    "minimum_should_match": 0,
    "must_not": [
      {
        "match": {
```

```
          "text": "ring"
        }
      }
    ]
  }
}
```

a tedy zapisuje: `of` $\wedge \neg$ `ring`.

5 Přibližné vyhledávání

Uvažujme nyní jen řetězce, které jsou termy nebo prázdný řetězec označený ϵ. Přidání znaku x k řetězci w označíme xw. Například pro $x =$ `r` a $w =$ `ing` je $xw =$ `ring` nebo `g`$\epsilon =$ `g`.

Délku řetězce w označíme $|w|$. Například $|$`Ring`$| = 4$ a $|\epsilon| = 0$.

Vezměme řetězce w_1, w_2. Definujeme (**editační** nebo také **Levenštejnovu**) **vzdálenost** $d(w_1, w_2)$ řetězců w_1, w_2 následovně.

1. Pokud $w_1 = \epsilon$, pak $d(w_1, w_2) = |w_2|$,
2. pokud $w_2 = \epsilon$, pak $d(w_1, w_2) = |w_1|$,
3. pokud $w_1 \neq \epsilon$ a $w_2 \neq \epsilon$ a $xw_1' = w_1$ a $xw_2' = w_2$ pro nějaký znak x, pak $d(w_1, w_2) = d(w_1', w_2')$,
4. pokud $w_1 \neq \epsilon$ a $w_2 \neq \epsilon$ a $xw_1' = w_1$ a $yw_2' = w_2$ pro nějaké znaky x, y takové, že $x \neq y$, pak $d(w_1, w_2) = 1 + min(d(w_1', w_2'), d(w_1', w_2), d(w_1, w_2'))$.

Například: $d($`ing`$,$ `ing`$) = d($`ng`$,$ `ng`$) = d($`g`$,$ `g`$) = d(\epsilon, \epsilon) = 0$ a proto $d($`ring`$,$ `king`$) = 1 + min(d($`ing`$,$ `ing`$), d($`ing`$,$ `king`$), d($`ring`$,$ `ing`$)) = 1$.

Pro číslo $k \geq 0$ zavedeme ekvivalenci $\approx_k$ na termech takovou, že položíme $w_1 \approx_k w_2$, jestliže $d(w_1, w_2) \leq k$. Například `ring` $\approx_1$ `king`, ale `ring` $\not\approx_1$ `kin`. Ekvivalence $\approx_0$ je relace rovnosti na termech. Tedy $w_1 \approx_0 w_2$, právě když $w_1 = w_2$.

Vezměme term q, pak

$$q \approx_k$$

je podmínka, kterou dokument D splňuje, jestliže pro $\mathrm{T}(D) = t_1, \ldots, t_n$ existuje $1 \leq i \leq n$ tak, že $q \approx_k t_i$. Například dokument `"The Fellowship of the Ring"` splňuje podmínku `king` $\approx_1$, protože obsahuje term `ring` a `king` $\approx_1$ `ring`.

Kvalita splnění podmínky $q \approx_k$ dokumentem D z indexu I je dána vztahem:

$$score(I, D, q \approx_k) = \sum_{i=1}^{n} boost(t_i, q) \cdot IDF(I, t_i) \cdot tf(I, t_i, D)$$

kde $t_1, \ldots, t_n$ jsou všechny termy z $\mathrm{Set}(\mathrm{T}(D))$ ekvivalentní s q vzhledem k $\approx_k$. Číslo $boost(t_i, q)$ udává podobnost termů t_i a q a klesá od 2.2 k nule.

Například:

$$score(I_1, \texttt{"The Fellowship of the Ring"}, \texttt{king} \approx_1) \approx 1.65 \cdot 0.98 \cdot 0.43.$$

Pro $1 \leq k \leq 2$ řetězec:

```
{
  "match": {
    "text": {
      "query": D,
      "fuzziness": k,
      "fuzzy_transpositions": false
    }
  }
}
```

zapisuje podmínku $q_1 \approx_k \vee \cdots \vee q_n \approx_k$, kde $\mathrm{T}(D) = (q_1, \ldots, q_n)$ a řetězec:

```
{
  "match": {
    "text": {
      "query": D,
      "operator": "AND",
      "fuzziness": k,
      "fuzzy_transpositions": false
    }
  }
}
```

podmínku $q_1 \approx_k \wedge \cdots \wedge q_n \approx_k$. Například:

```
GET /movie/_search
{
  "query": {
    "match": {
      "text": {
        "query": "ruturn ring",
        "fuzziness": 1,
        "fuzzy_transpositions": false
      }
    }
  }
}
```

```
{
   "hits": {
    "hits": [
      {
        "_id": "3",
        "_score": 1.4610269,
        "_source": {
          "text": "The Return of the King"
        }
      },
      {
        "_id": "1",
        "_score": 0.9227538,
        "_source": {
          "text": "The Fellowship of the Ring"
        }
      }
    ]
  }
}
```


Otázky a úkoly na cvičení

1. Uvažujme indexovou proměnnou `book` s charakteristickou vlastností „Kniha d se jmenuje D.“ a hodnotou:

   1 `The Life And Opinions Of Tristram Shandy`

   2 `Emma`

   3 `Nightmare Abbey`

   4 `One Day in the Life of Ivan Denisovich`

   5 `Life After Life`

   Napište výrazy pro Elasticsearch, které mají následující charakteristické vlastnosti.

   (a) „Název D knihy d obsahuje slova life a of.“

   (b) „Název D knihy d obsahuje jméno Ivan Denisovich nebo Tristam Shandy.“

   (c) „Název D knihy d obsahuje slovo life a předložku of nebo in.“

   (d) „Název D knihy d neobsahuje slovo life.“

   (e) „Název D knihy d neobsahuje jména: Emma, Shandy a Ivan.“

   (f) „Název D knihy d obsahuje aspoň dvě ze slov: in, of a life.“

   Spočítejte hodnoty výrazů a ověřte je s těmi, které spočítá systém Elasticsearch.

2. Určete hodnotu a charakteristickou vlastnost následujícího výrazu.

```
{
"bool": {
    "must": {
        "match": {
            "text": "life of"
        }
    },
    "should": {
        "match": {
            "text": "day"
        }
    }
}
```

   Hodnotu porovnejte s hodnotou získanou ze systému Elasticsearch.

3. Spočítejte editační vzdálenost následujících dvojic termů:

   (a) `day` a `da`

   (b) `day` a `dya`

   (c) `day` a `dy`

   (d) `day` a `of`

4. Určete, které dokumenty v hodnotě indexové proměnné `book` splňují podmínku `one` $\approx_2$. Odpověď ověřte systémem Elasticsearch.

5. Určete hodnotu následujícího výrazu.

```
GET /movie/_search
{
  "query": {
    "match": {
      "text": {
        "query": "lfe ifan",
        "operator": "AND",
        "fuzziness": 1,
        "fuzzy_transpositions": false
      }
    }
  }
}
```