Získávání znalostí z textu pomocí klasifikace algoritmy strojového učení

Jan Žižka
Centrum biostatistiky a analýz CBA
Masarykova universita

zizka@cba.muni.cz

ZÍSKÁVÁNÍ ZNALOSTÍ Z NESTRUKTUROVANÝCH TEXTOVÝCH DOKUMENTŮ

Následující informace se věnuje stručnému přehledu problematiky z těchto hledisek:

- Co se rozumí pojmem *znalost*
- Jaká je obecně *forma textovch dokumentů*
- V jaké formě předložit text *algoritmu* v počítači
- Jak lze text před zpracováním vhodně *připravit*
- V čem je princip *strojového učení*
- Jak se dá strojové učení pro *získání znalosti* použít
- Které *algoritmy strojového učení* přicházejí do úvahy
- Jaký je *konkrétní postup* při učení
- Jaké *výsledky* lze očekávat

Obecná zásada při řešení uvedených problémů je:

EXPERIMENTOVAT

Získávání znalostí z nestrukturovaných textových dokumentů

Velmi vysoké objemy elektronických textových dokumentů obsahují obecně různou *znalost* danou specifickým hlediskem, tj. konkrétní aplikací, která texty využívá.

Potíž tvoří jak *množství dat*, tak i jejich forma – *přirozený jazyk*, který nemá vhodný formální popis pro vytvoření příslušné teorie umožňující automatizované zpracování.

Typická cesta získávání znalostí z elektronických nestrukturovaných textů spočívá v těchto krocích:

- zdroj → patřičný objem (obecně zašuměných) dat
- odstranění šumu → čistá data
- výběr aplikačně zajímavé části dat → informace
- zobecnění informace → znalost

Vzhledem k nedostatku formální teorie, která by na základě zadání hodnot určitých parametrů byla schopna např. požadované předpovědi, je nutno použít alternativních metod.

Poměrně velmi úspěšné výsledky poskytují algoritmy *strojového učení*, kdy příslušné parametry zvoleného algoritmu se stanovují automatizovaně ve fázi *trénování*. Vlastnosti natrénovaného algoritmu se ověří ve fázi *testování*, a pokud výsledky testování jsou přijatelné, lze naučený algoritmus použít pro danou aplikaci.

Trénovací fáze vyžaduje vhodné *učící příklady*, neboť vlastnosti algoritmu (parametry) jsou nakonec stanoveny použitými daty.

Testovací fáze používá příklady, které *nebyly* algoritmu předloženy během učení. Správná *generalizace* umožní správné zpracování informace, která se teprve objeví v budoucnu.

Representace textových dokumentů – *bag of words* (BOW)

Metody *strojového učení* převážně vidí textové dokumenty jako soubory obsahující symbolické hodnoty (“termíny”, “slova”), aniž by se zabývaly jejich významem (nanejvýš velmi mělce, např. při předzpracování dat) nebo vzájemnou závislostí.

Proto je *pořadí slov* v dokumentu zcela *bezvýznamné*, což sice eliminuje určitý informační obsah, ale výrazně zjednodušuje zpracování přirozeného jazyka z hlediska např. klasifikace. Uvedená zjednodušení umožňují se vyhnout nevyřešeným problémům se strojovým porozuměním dokumentům v přirozeném jazyce (včetně odstranění potíží s *mnohojazyčností*), ale mohou vést ke snížení efektivity (nerozeznávání významu synonym, ignorování gramatiky, ...). Přesto jsou výsledky překvapivě dobré v mnoha aplikacích, i když je nutno počítat s chybovostí.

Předzpracování – ovlivňuje výrazně kvalitu výsledku, např.:

- vyřazení obecných slov, která nemají specifický význam z hlediska aplikace (např. předložky, zkratky)
- vyřazení slov s velmi nízkou nebo vysokou frekvencí ve všech dokumentech
- vyřazení interpunkce, mezer, apod.
- převod všech písmen na malá (slovo na začátku věty a uvnitř věty je totéž při representaci typu *bag-of-words*)

Eliminace nevýznamných znaků a slov výrazně *snižuje dimensionalitu* problému (např. řádově z 10^4 na 10^3, protože každé unikátní slovo představuje jednu dimensi).

Příklad representace textu pomocí metody *bag-of-words* (BOW), kde se ignoruje interpunkce, členění textu do řádků, velká a malá písmena, dvojjazyčnost (anglické termíny v české větě), pořadí slov, které může mít velký význam (např. *machine learning – strojové učení* a *learning machine – učící stroj* má zcela odlišný význam), a vynechají se obecná slova. Vznikne tedy slovník (seznam symbolů) používaný pro trénink nějakého zvoleného algoritmu:

anglické bag bow české členění dvojjazyčnost ignoruje interpunkce learning machine má malá metody mít může obecná odlišný písmena pomocí pořadí příklad representace řádků slov stroj strojové termíny textu učení učící velká velký větě vynechání význam words zcela

Representace BOW v předchozím příkladu vynechala např. spojku *'a'*, zvratné *'se'*, a některé další symboly nemající pro konkrétní aplikaci význam (včetně interpunkce).

Další redukci dimensionality lze docílit např. převodem slov na základní tvar (kmen, lemma). V předchozím příkladu by bylo možné redukovat vzniklý slovník (*infinitiv*, 1. *pád*, *jednotné číslo*, *jediný rod*, apod.), takže dimensionalita klesne o 4:

mít má stroj strojové učení učící velká velký

↓

mít stroj učit velký

Tzv. *lemmatizace* je ovšem jazykově závislá. Pro angličtinu existuje zjednodušený systém *Porter stemming*, kde se prostě odřezávají koncovky, což není dokonalé, ale z praktického hlediska účinné.

Výskyt slov – existuje více možností, jak je vyjádřit, např.:

- binární – '1' slovo se v dokumentu vyskytuje, '0' slovo se nevyskytuje (váha slova = 1 nebo 0)
- frekvenční – váha slova je dána četností jeho výskytu
- tf-idf – *term frequency-inverted document frequency*: četnost slova v dokumentu (representace dokumentu daným slovem) vůči počtu dokumentů, v nichž se dané slovo vyskytuje (v čím více dokumentech se dané slovo vyskytuje, tím nižší je jeho diskriminační hodnota)

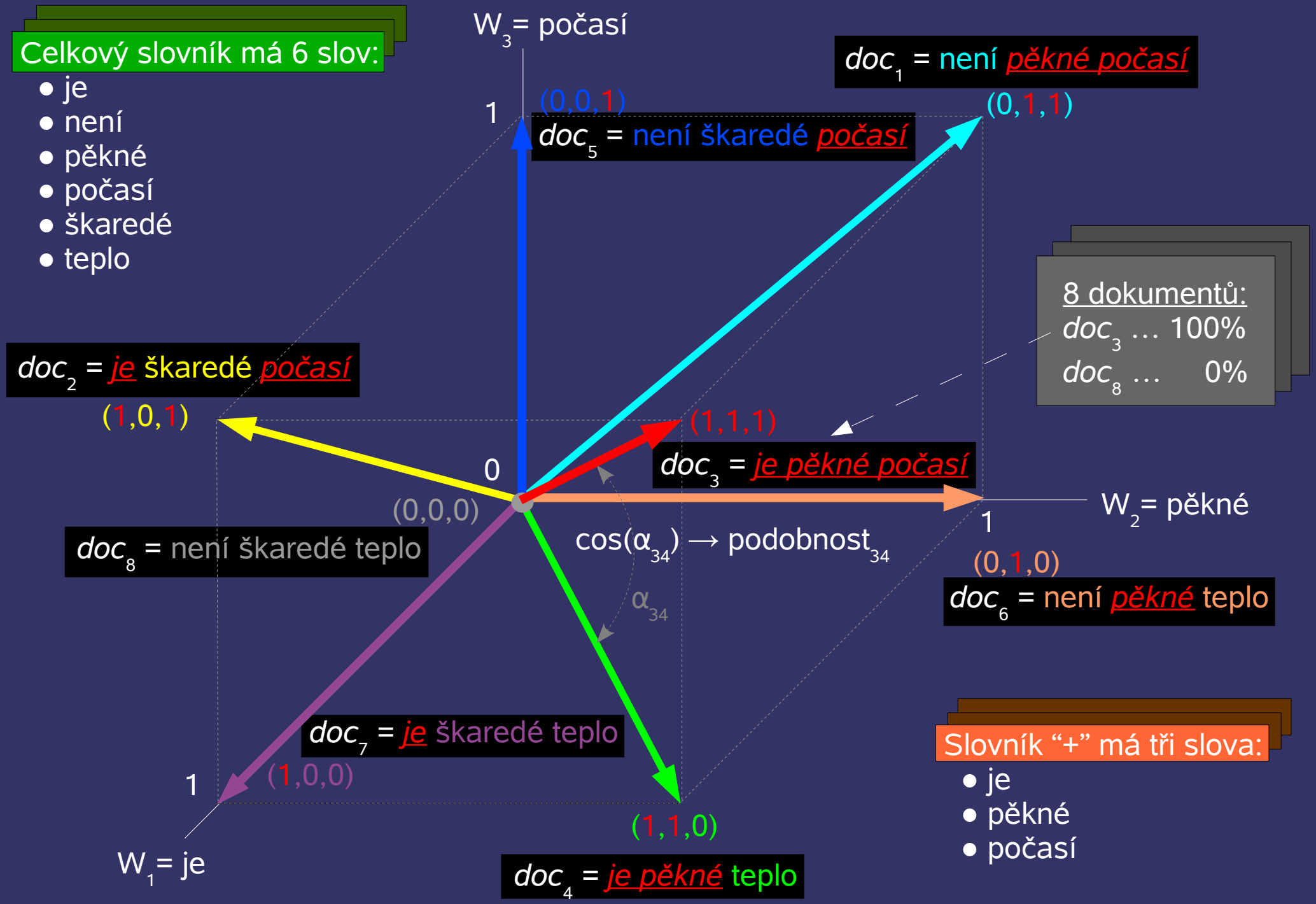
Optimální representaci výskytu slov je obvykle nutno zjistit *experimentálně*, protože dosavadní zkušenosti ukazují, že je aplikačně závislá a navíc pro různé algoritmy může být různá.

Další obrázek ilustruje *representaci* jednotlivých textových dokumentů jako *vektory* (pro jednoduchost jsou zde koncové body vektorů dány souřadnicemi v prostoru s *binárním* vyjádřením výskytu slov – pokud by např. bylo použito *frekvenčního* vyjádření, pak jednotlivé osy jsou číselné reálné).

Vektorová representace umožňuje porovnávat *podobnost* či různost *dokumentů* jako podobnost vektorů: *délka* vektorů a *úhel* α, který navzájem svírají.

To umožňuje snadno porovnávat neznámé texty se známými např. výpočtem hodnoty $0.0 \leq cos(\alpha) \leq 1.0$, tj. hodnoty z reálného jednotkového intervalu.

ZÍSKÁVÁNÍ ZNALOSTÍ Z NESTRUKTUROVANÝCH TEXTOVÝCH DOKUMENTŮ



Metody klasifikace/kategorizace textových dokumentů

Existuje více možností strojového učení, například:

- metoda *k*-NN (*k*-nearest neighbors), nejbližší soused(é) (Eukleidova vzdálenost udává podobnost dokumentů)
- metoda generování *rozhodovacích stromů* mimalizací entropie (uzly testují jen ta relevantní "slova", která přispívají k zařazení dokumentu do správné kategorie)
- metoda *disjunktní normální formy* (vytvořená pravidla)
- metoda *support vector machines* (SVM, nalezení pouze těch textů, které tvoří oddělovací hranici mezi dvěma třídami)
- *Bayesův naivní klasifikátor* (stanovení pravděpodobnosti náležení do jedné ze tříd pomocí kombinace podmíněných pravděpodobností vypočítaných z tréninkových dat)
- a řada dalších, včetně možnosti *kombinací* metod

Jako demonstraci klasifikační metody lze použít např. jeden z nejčastěji aplikovaných algoritmů, tzv. metodu *naivního bayesovského klasifikátoru* (BNK).

BNK je založen na Bayesově teorému pro pravděpodobnostní inferenci – předpokladem je, že míra náležení kombinovaných jevů (zde výskytů slov v dokumentu) do patřičných kategorií je řízena rozloženími pravděpodobností a že optimální rozhodnutí lze najít pomocí těchto pravděpodobností a údajů z disponibilních dat z reálného světa:

$$p(h \mid D) = \frac{p(D \mid h)\, p(h)}{p(D)}$$

Počítá se pravděpodobnost hypotézy h (např. příslušnost do určité třídy) přičemž jsou dána nějaká trénovací data D.

Bayesův teorém z předchozího vztahu využívá hodnoty pravděpodobností $p(D \mid h)$, což jsou pravděpodobnosti výskytu dat D za předpokladu platnosti uvažované hypotézy h.

$p(D)$ je pravděpodobnost výskytu dat D bez jakéhokoli vztahu k jakékoli hypotéze (*apriorní* pravděpodobnost).

$p(h)$ je pravděpodobnost platnosti hypotézy h (*apriorní* pravděpodobnost), aniž jsou dosud známa nějaká data D, která svým výskytem mohou zvýšit či snížit $p(h)$.

$p(h \mid D)$ je tedy hledaná *aposteriorní* pravděpodobnost, že *pro daná data* D bude platit hypotéza h.

Naivní Bayesův klasifikátor

Výpočetní složitost lze výrazně snížit zavedením úmyslné nekorektnosti, aby bylo možno Bayesovu metodu v praxi použít:

Hodnoty atributů (*slova* na jednotlivých *pozicích* v dokumentu) *jsou navzájem podmíněně nezávislé*, tj. dokument je vlastně pozorovanou *konjunkcí* hodnot atributů.

V takovém případě se celková pravděpodobnost náležení textu do každé z možných tříd c_j počítá zjednodušeně jako součin pravděpodobností jednotlivých nezávislých jevů, tj. výskytů individuálních slov w_i v dokumentu:

$$p(w_1, w_2, \ldots, w_m \mid c_j) \rightarrow \prod_i p(w_i \mid c_j)$$

$$c_{NB} = \underset{c_j}{argmax}\left[p(c_j) \prod_{i=1}^{n} p(w_i \mid c_j) \right]$$

n … počet slovních pozic v dokumentu klasifikovaném do třídy c_j

j … index jedné z uvažovaných klasifikačních tříd (alespoň dvě)

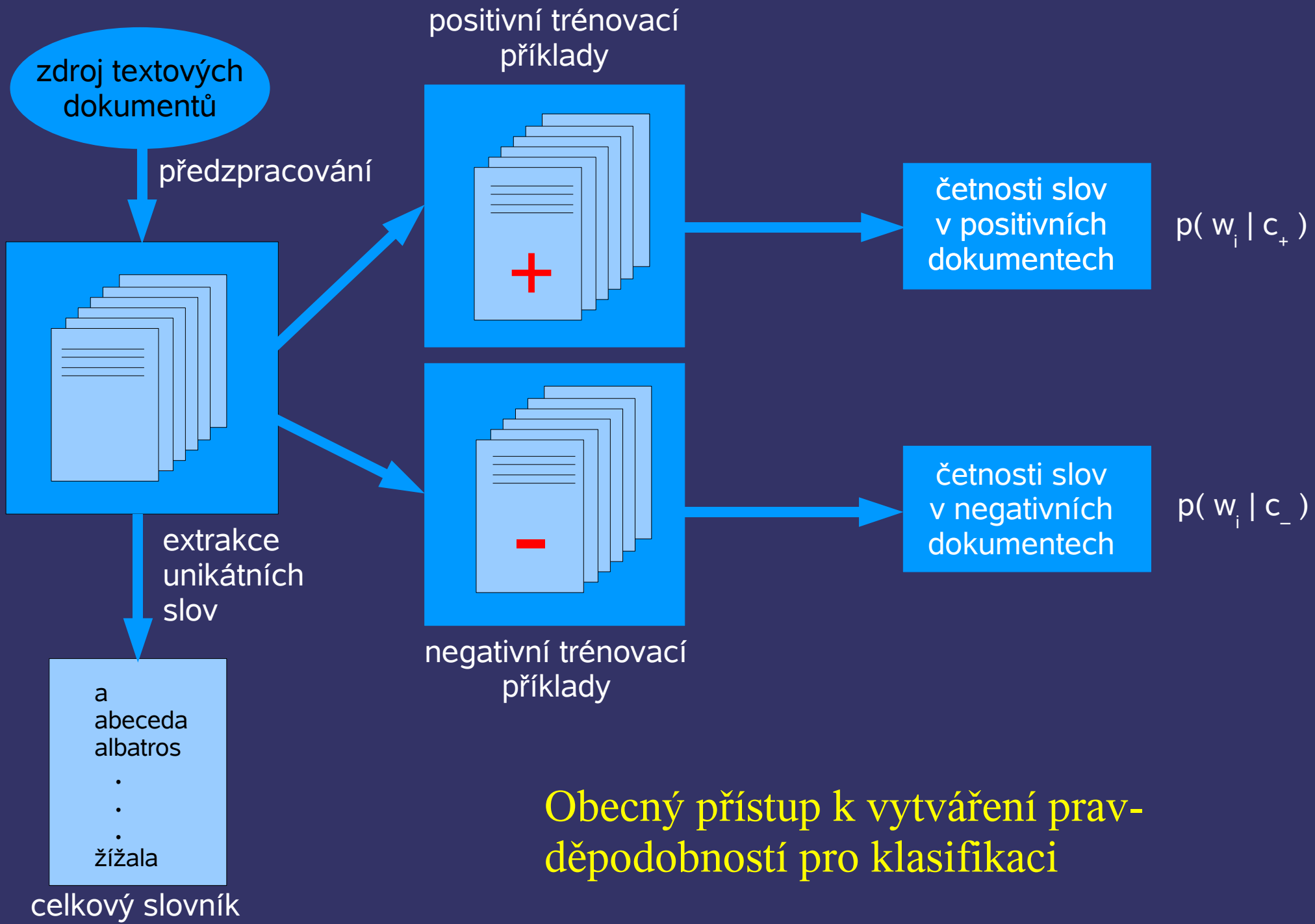
$p(c_j)$ … apriorní pravděpodobnost výskytu dokumentu v c_j

$p(w_i \mid c_j)$ … aposteriorní pravděpodobnost výskytu slova w_i v c_j

Např. konkrétní dokument d_k může být representován $n = 143$ různými slovy $w_1, w_2, w_3, \ldots, w_{143}$ a klasifikován do $j = 3$ různých tříd, např. $c_j \in$ {zajímavý, nezajímavý, neutrální}.

Pro každé c_j se spočítá pravděpodobnost, s níž tam dokument patří, a výsledek c_{NB} ($c_{Naïve\ Bayes}$) je dán maximální hodnotou *argmax*.

ZÍSKÁVÁNÍ ZNALOSTÍ Z NESTRUKTUROVANÝCH TEXTOVÝCH DOKUMENTŮ



Obecný přístup k vytváření prav-děpodobností pro klasifikaci

Trénovací texty:

w_1	w_2	w_3	c_j
je	pěkné	počasí	+
je	chladno		-
není	velmi	chladno	+
není	pěkné		-
velmi	chladno		-
chladno			-
.	.	.	.
.	.	.	.
.	.	.	.

\+ texty : celkem **6** slov
– texty : celkem 7 slov

počet unikátních slov : **6**

Klasifikovaný dokument “to není pěkné chladno”: **+** nebo **–** ?

Po vytvoření celkového slovníku z unikátních slov (je jich zde 6), výpočtu apriorních pravděpodobností (2 texty + a 4 texty – v celkem 6 textech), výpočtu aposteriorních pravděpodobností výskytu slov v + a –, a následné normalizaci lze určit výsledek:

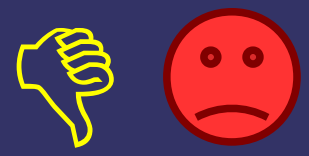
setříděný slovník :	w_1	w_2	w_3	w_4	w_5	w_6
	chladno	je	není	pěkné	počasí	velmi
četnost slova w_i v +	1	1	1	1	1	1
četnost slova w_i v –	3	1	1	1	0	1
p (w_i | +)	1/6	1/6	1/6	1/6	1/6	1/6
p (w_i | –)	3/7	1/7	1/7	1/7	0/7	1/7

$$p = p(\text{'není', 'pěkné', 'chladno'} \mid +/-) =$$
$$= p_{NBK}(\text{'není'} \mid +/-) \times p(\text{'pěkné'} \mid +/-) \times p(\text{'chladno'} \mid +/-)$$

"$w_3\ w_4\ w_1$" = "není pěkné chladno"

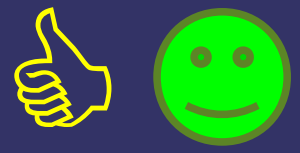
$$P^+ = p(+)\ p(w_3 = \text{'není'} \mid +)\ p(w_4 = \text{'pěkné'} \mid +)\ p(w_1 = \text{'chladno'} \mid +) =$$

$$= \frac{2}{6} \times \frac{1}{6} \times \frac{1}{6} \times \frac{1}{6} \approx \underline{0.00154}$$



$$P^- = p(-)\ p(w_3 = \text{'není'} \mid -)\ p(w_4 = \text{'pěkné'} \mid -)\ p(w_1 = \text{'chladno'} \mid -) =$$

$$= \frac{4}{6} \times \frac{1}{7} \times \frac{1}{7} \times \frac{3}{7} \approx \underline{0.00583}$$



$$P_n^+ = \frac{0.00154}{0.00154 + 0.00583} \approx 0.21$$

$$P_n^- = \frac{0.00583}{0.00154 + 0.00583} \approx 0.79$$

$$P_n^- > P_n^+ \Rightarrow \textbf{negativní}$$