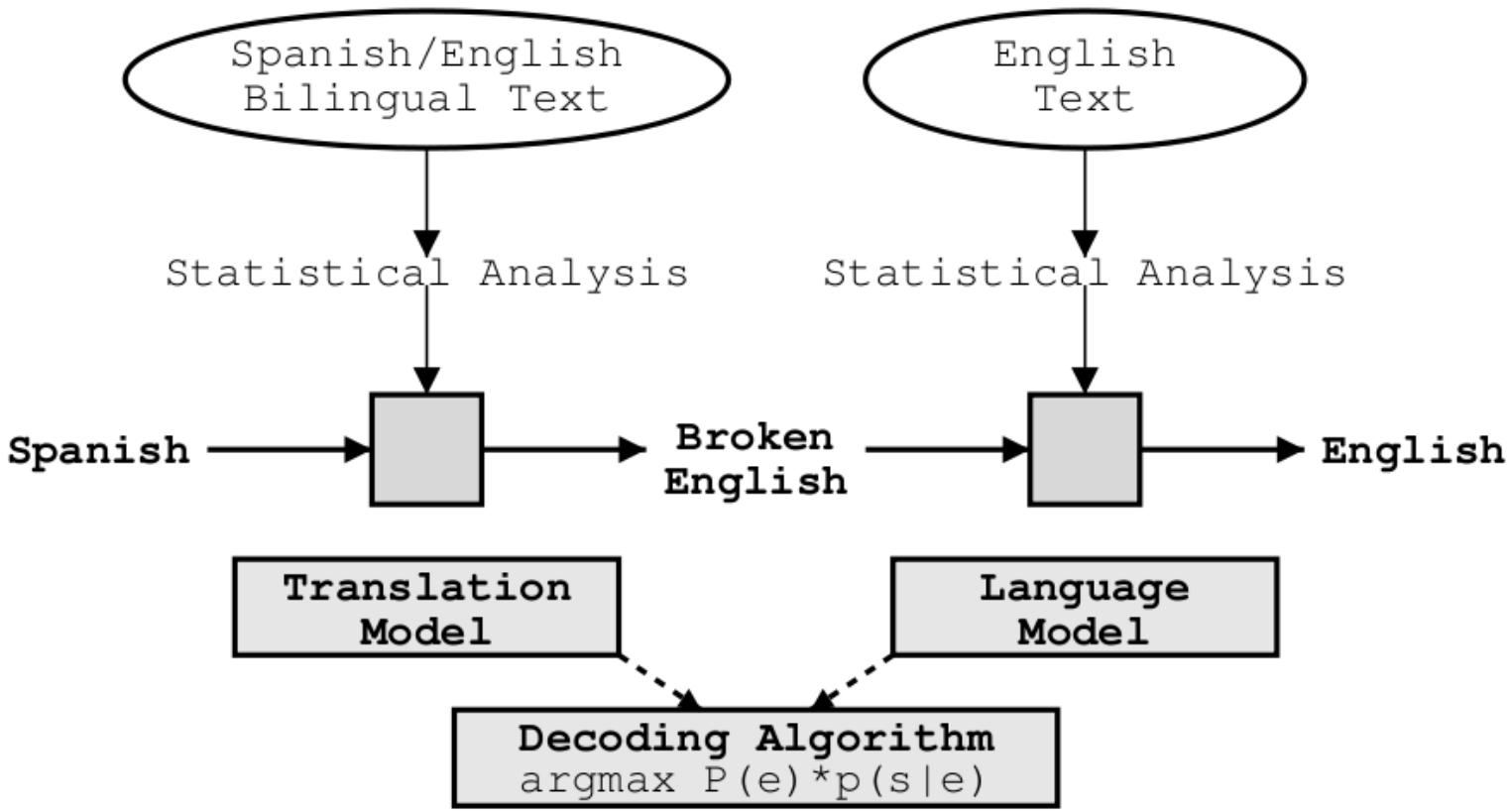
Úvod do SMT

- pravidlové systémy motivovány lingvistikou
- SMT inspirován teorií informace a statistikou
- v současnosti mnoho společností se zaměřením na SMT: Google, IBM, Microsoft, Language Weaver (2002)
- 50 miliónů stránek denně přeložených pomocí SMT
- **gisting**: stačí, má-li překlad nějaký užitek, nepotřebujeme přesný význam; nejčastější užití MT na internetu

Schéma SMT


Spanish/English
Bilingual Text
English
Text
Statistical Analysis
Statistical Analysis
Spanish
Broken
English
English
Translation
Model
Language
Model
Decoding Algorithm
argmax P(e)*p(s|e)

Nástroje SMT

- GIZA++: trénování IBM modelů, zarovnávání na úrovni slov (word alignment pomocí HMM)
- SRILM: trénování jazykových modelů
- IRST: trénování velkých jazykových modelů
- Moses: frázový dekodér, trénování modelů
- Pharaoh: předchůdce Mosese
- Thot: trénování frázových modelů
- SAMT: tree-based modely

Data pro SMT – (paralelní) korpusy

- Linguistics Data Consorcium (LDC): paralelní korpusy pro páry arabština-angličtina, čínština-angličtina atd.
  Gigaword korpus (angličtina, 7 mld slov)
- Europarl: kolekce textů Evropského parlamentu (11 jazyků, 40 M slov)
- OPUS: paralelní texty různého původu (lokalizace software)
- Acquis Communautaire: právní dokumenty Evropské únie (20 jazyků)

Pravidelné události v oblasti SMT, soutěže

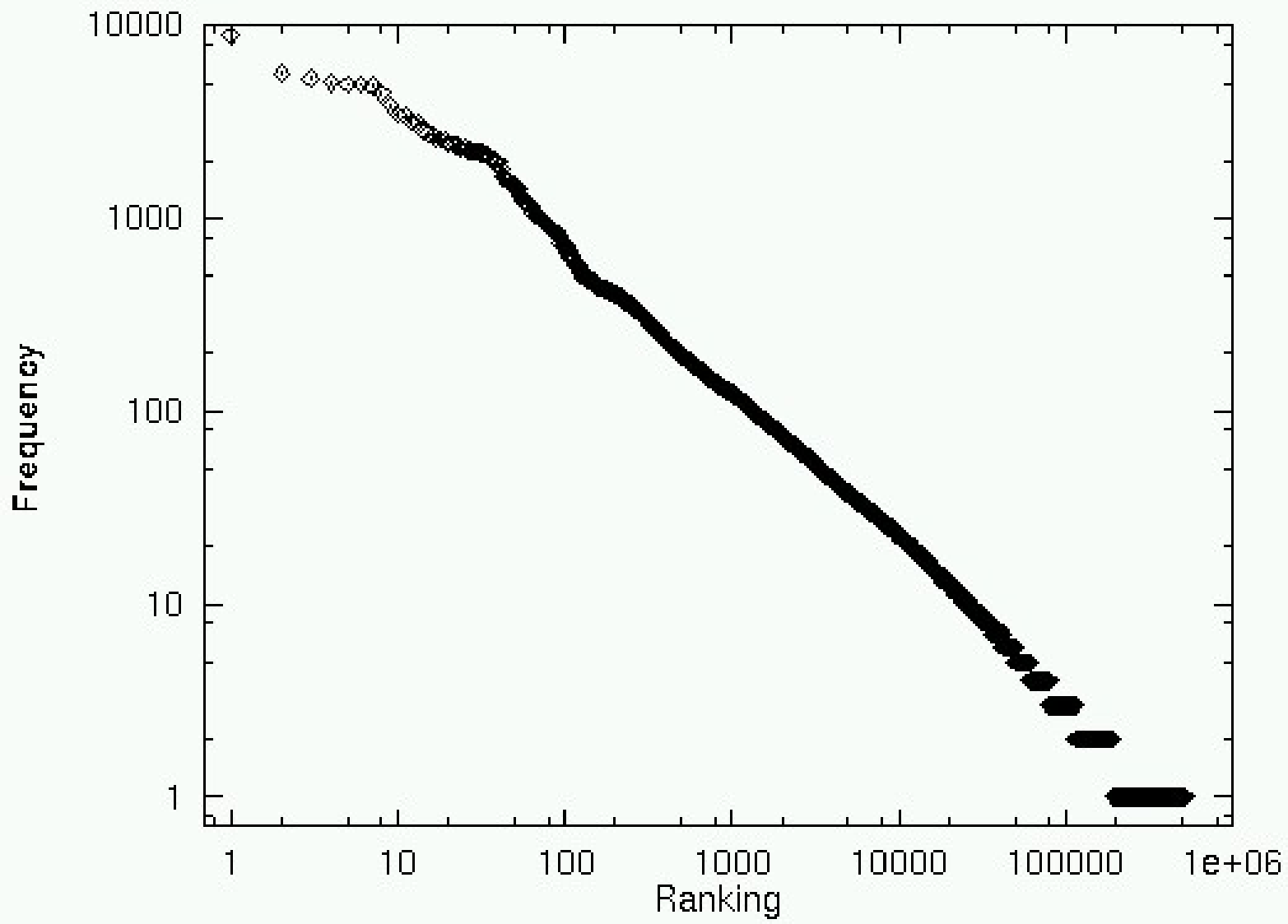
Většinou roční vyhodnocování kvality SMT. Tvorba testovacích sad, manuální vyhodnocování dat, referenční systémy.

- NIST: National Institute of Standards and Technology; nejstarší, prestižní; hodnocení překladu arabštiny, čínštiny
- IWSLT: mezinárodní workshop překladu mluveného jazyka; překlad řeči; asijské jazyky
- WMT: Workshop on SMT; překlady mezi evropskými jazyky

Slova

- pro SMT v drtivé většině případů základní jednotka = slovo
- v mluvené řeči slova neoddělujeme: jak je od sebe oddělíme?
- SMT systémy provádí de-tokenizaci
- překlad samotný je většinou s lowercase textem
- jaká slova má angličtina $\rightarrow$ jaká slova jsou v anglických korpusech
- *the* tvoří 7 % anglického textu
- 10 nejčastějších slov (tokenů) tvoří 30 % textu (!)
- *Zipfův zákon*: r rank (pořadí ve frekvenčním seznamu slov), f = frekvence výskytu slova, c = konstanta; platí $r \times f = c$
- překlepy, čísla, vlastní jména, názvy a cizí slova

Zipfův zákon



Věty

- syntaktická struktura se v jazycích liší
- vkládání funkčních slov, která jsou typická pro daný jazyk (*the*, interpunkce)
- přerovnávání: *er wird mit uns gehen* $\rightarrow$ *he will go with us*
- některé jevy nelze přeložit na úrovni věty: anafory
- úroveň celého dokumentu: téma (topic) může pomoci při volbě vhodného překladového ekvivalentu
- v textu o jeskynních živočiších zřejmě nebude překládat *bat* jako *pálka*

Paralelní korpusy

- základní datový zdroj pro SMT
- volně dostupné jsou řádově 10 a 100 miliónů slov veliké
- je možné stáhnout paralelní texty z internetu
- vícejazyčné stránky (BBC, Wikipedie)
- problém se zarovnáním dokumentů, odstavců, . . .
- srovnatelné korpusy (comparable corpora): texty ze stejné domény, ne přímé překlady: New York Times – Le Monde
- Kapradí – korpus překladů Shakespearových dramat (FI)
- InterCorp – ručně zarovnané beletr. texty (ČNK, FFUK)

Zarovnávání vět

- věty si neodpovídají 1:1
- některé jazyky explicitně nenaznačují hranice vět (thajština)
- *It is small, but cozy. – Es is klein. Aber es ist gemütlich.*
- pro věty $e_1, \cdots e_{n_e}$ a $f_1, \cdots f_{n_f}$
- hledáme páry $s_1, \ldots s_n$
- $s_i = (\{f_{\text{start}-f(i)}, \cdots f_{\text{end}-f(i)}\}, \{e_{\text{start}-e(i)}, \cdots e_{\text{end}-e(i)}\})$

P	**typ zarovnání**
0.98	1–1
0.0099	1–0 nebo 0–1
0.089	2–1 nebo 1–2
0.011	2–2

Pravděpodobnostní rozložení

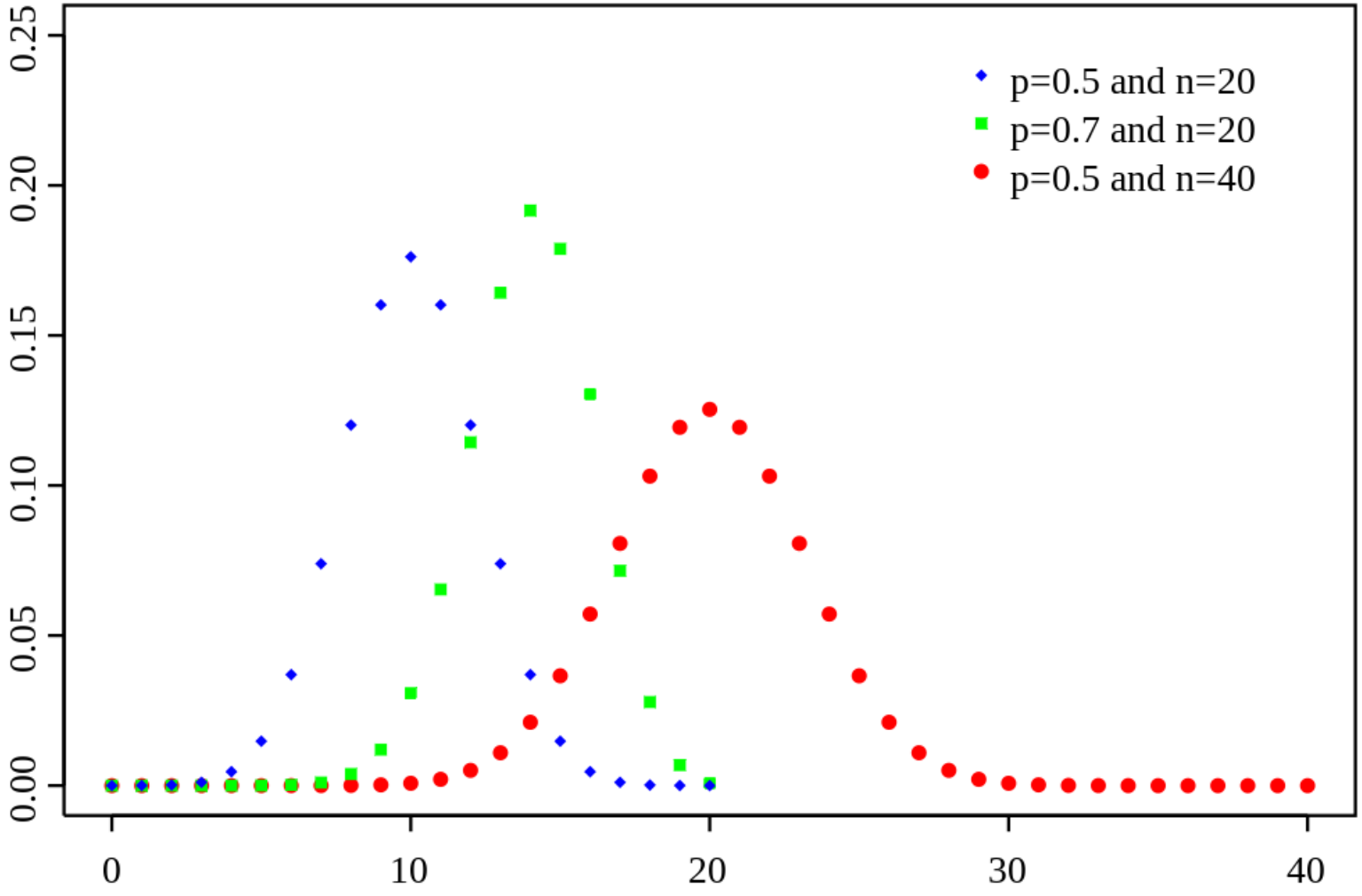
- graf hodnot pravděpodobnosti pro elementární jevy náhodné veličiny
- **rovnoměrné**: hod kostkou, mincí (diskrétní veličina)
- **binomické**: vícenásobný hod

$$b(n,k;p) = \binom{n}{k} p^k (1-p)^{n-k}$$

- **normální, Gaussovo**: spojité, dobře aproximuje ostatní rozložení; zahrnuje rozptyl

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

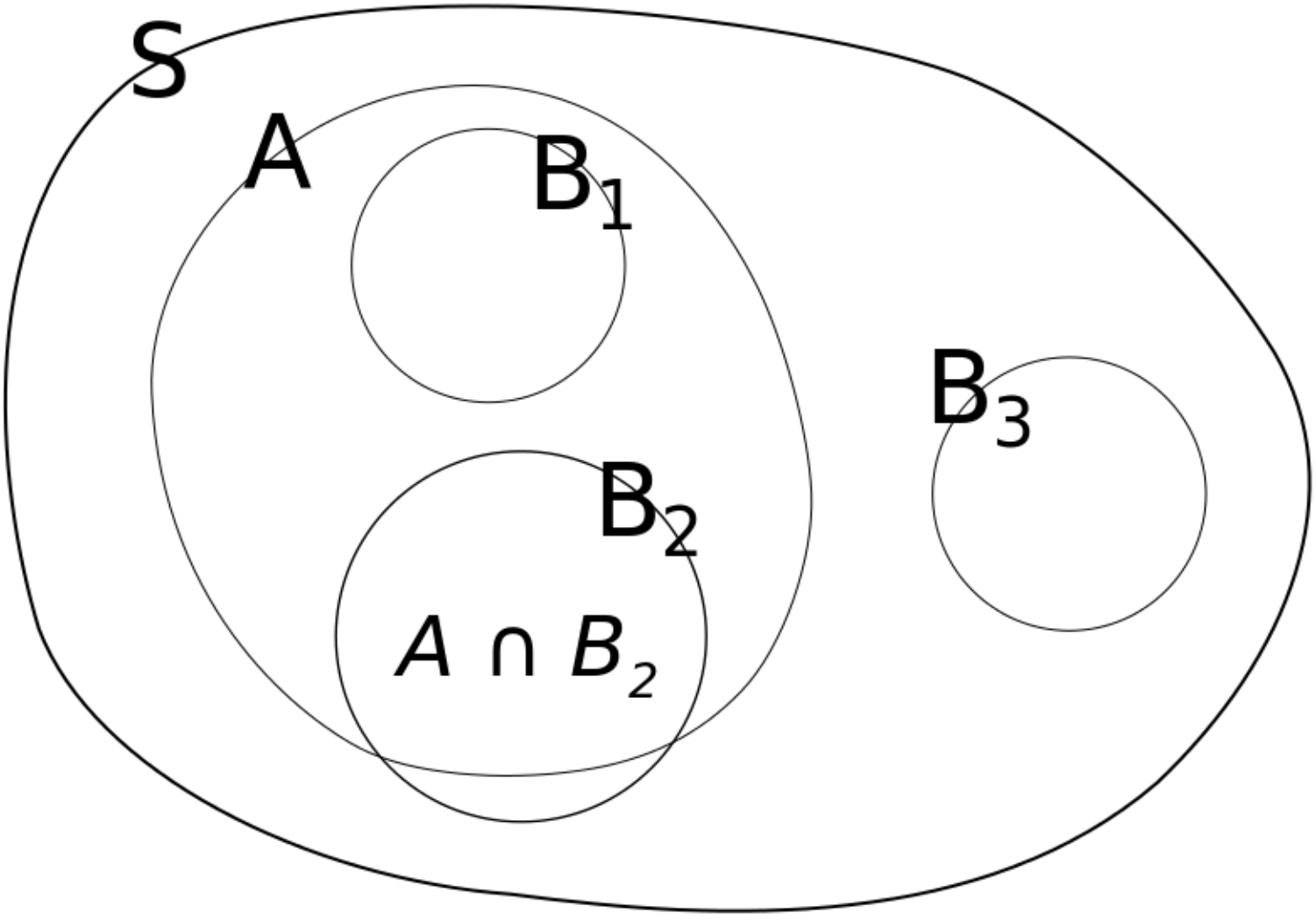
Binomické rozložení



Statistika I

- náhodná proměnná, pravděpodobnostní funkce, ...
- máme data, chceme spočítat rozložení, které nejlépe tato data vystihuje
- **zákon velkých čísel**: čím víc máme dat, tím lépe jsme schopni odhadnout pravděpodobnostní rozložení
- např.: hod falešnou kostkou; výpočet π
- nezávislé proměnné: $\forall x, y : p(x, y) = p(x).p(y)$
- **spojená (joint) pravděpodobnost**: hod mincí a kostkou
- **podmíněná pravděpodobnost**: $p(y|x) = \frac{p(x,y)}{p(x)}$
  pro nez. proměnné platí: $p(y|x) = p(y)$

Podmíněná pravděpodobnost



Shannonova hra

Pravděpodobnostní rozložení pro následující znak v textu se liší v závislosti na předchozích znacích.

Doplňujeme postupně znaky (malá abeceda a mezera).
Některé znaky nesou více informace (jsou uhádnuty později).

Bayesovo pravidlo

$$p(x|y) = \frac{p(y|x).p(x)}{p(y)}$$

- příklad s kostkou
- $p(x)$ – prior
- $p(y|x)$ – posterior

Statistika II

- střední hodnota (diskrétní): $\mathrm{E}\,X = \sum_I s_i.p_i$
- rozptyl: $\sigma^2 = \sum_{i=1}^{n} [x_i - \mathrm{E}(X)]^2 p_i$
- očekávaná hodnota: $\mathrm{E}[X] = \sum_{x \in X} x.p(x)$

SMT – princip noisy channel

Vyvinut Shannonem (1948) pro potřeby samoopravujících se kódů, pro korekce kódovaných signálů přenášených po zašuměných kanálech na základě informace o původní zprávě a typu chyb vznikajících v kanálu.

Příklad s OCR. Rozpoznávání textu z obrázků je chybové, ale dokážeme odhadnout, co by mohlo být v textu (jazykový model) a jaké chyby často vznikají: záměna l-1-I, rn-m apod.

$$\begin{aligned} e^* &= \arg\max_e p(e|f) \\ &= \arg\max_e \frac{p(e)p(f|e)}{p(f)} \\ &= \arg\max_e p(e)p(f|e). \end{aligned}$$

SMT – komponenty noisy channel principu

- jazykový model:
  - jak zjistit $p(e)$ pro libovolný řetěz e
  - čím víc vypadá e správně utvořené, tím vyší je $p(e)$
  - problém: co přiřadit řetězci, který nebyl v trénovacích datech?
- překladový model:
  - pro e a f vypočítej $p(f|e)$
  - čím víc vypadá e jako správný překlad f, tím vyšší p
- dekódovací algoritmus
  - na základě předchozího najdi pro větu f nejlepší překlad e
  - co nejrychleji, za použití co nejmenší paměti

Jazykové modely

Noam Chomsky, 1969

But it must be recognized that the notion “probability of a sentence” is an entirely useless one, under any known interpretation of this term.

Fred Jelinek, 1988

Anytime a linguist leaves the group the recognition rate goes up.

Jak pravděpodobné je pronesení české věty *s*?

Ke snídani jsem měl celozrnný ...

Jazykové modely

Noam Chomsky, 1969

But it must be recognized that the notion “probability of a sentence” is an entirely useless one, under any known interpretation of this term.

Fred Jelinek, 1988

Anytime a linguist leaves the group the recognition rate goes up.

Jak pravděpodobné je pronesení české věty s?

Ke snídani jsem měl celozrnný ...

chléb $>$ pečivo $>$ zákusek $>$ mléko $>$ babičku

Jazykové modely

- zajímá nás určování pravděpodobnosti následujícího slova
- jazykový model je pravděpodobnostní rozložení nad všemi možnými sekvencemi slov daného jazyka

Pravděpodobnost sekvencí slov

p_{LM}(včera jsem jel do Brna)
p_{LM}(včera jel do Brna jsem)
p_{LM}(jel jsem včera do Brna)

Použijeme

- podmíněnou pravděpodobnost $P(X|Y)$ a
- joint probability (společná pravděpodobnost)

Jazykové modely

- LM pomáhají zajistit **plynulý výstup** (správný slovosled)
- LM pomáhají s **WSD v obecných případech**
- pokud má slovo více významů, můžeme vybrat nejčastější překlad (*pen* $\rightarrow$ pero)
- ve speciálních textech nelze použít, ale
- LM pomáhají s **WSD pomocí kontextu**
- p_{LM}(i go home) $\geq$ p_{LM}(i go house)

N-gramové modely

- n-gram je nejdůležitější nástroj ve zpracování řeči a jazyka
- využití statistického pozorování dat
- dvojí využití ve strojovém překladu:
  - po slovech *I go* je častější *home* než *house* apod.
  - *I go to home* vs. *I go home*
- generování jazyka

Generování unigramy

To him swallowed confess hear both. Which. Of save on trail for are ay device and rote life have Every enter now severally so, let.

Generování trigramy

Sweet prince, Falstaff shall die. Harry of Monmouth's grave. This shall forbid it should be branded, if renown made it empty.

N-gramové modely – naivní přístup

$$W = w_1, w_2, \cdots, w_n$$

Jak vypočítat $p(W)$? Spočítáme výskyty všech W v datech a normalizujeme je velikostí dat. Pro většinu velkých W však nebudeme mít v datech ani jeden výskyt. Úkolem je zobecnit pozorované vlastnosti trénovacích dat, která jsou většinou řídká (**sparse data**).

$$P(\text{chléb}|\text{ke snídani jsem měl celozrnný}) = \frac{|\text{ke snídani jsem měl celozrnný chléb}|}{|\text{ke snídani jsem měl celozrnný}|}$$

Markovův řetězec a Markovův předpoklad

$p(W)$, kde W je posloupnost slov, budeme modelovat postupně, slovo po slovu, užitím tzv. **pravidla řetězu**:

$$p(w_1, w_2, \ldots w_n) = p(w_1)p(w_2|w_1)p(w_3|w_1, w_2)\ldots p(w_n|w_1 \ldots w_{n-1})$$

Jelikož nemáme k dispozici pravděpodobnosti pro dlouhé řetězce slov, omezíme historii na m slov použitím **Markovova předpokladu**:

$$p(w_n|w_1, w_2, \ldots w_{n-1}) \simeq p(w_n|w_{n-m}, \ldots w_{n-2}, w_{n-1})$$

Číslo m nazýváme řádem odpovídajícího modelu. Nejčastěji se používají **trigramové** modely.

Výpočet, odhad pravděpodobností LM

Trigramový model používá pro určení pravděpodobnosti slova dvě slova předcházející. Použitím tzv. **odhadu maximální věrohodnosti** (maximum likelihood estimation):

$$p(w_3|w_1, w_2) = \frac{count(w_1, w_2, w_3)}{\sum_w count(w_1, w_2, w)}$$

trigram: *(the, green, w)* (1748)

w	**počet**	$p(w)$
paper	801	0.458
group	640	0.367
light	110	0.063
party	27	0.015
ecu	21	0.012

Kvalita a srovnání jazykových modelů

Chceme být schopni porovnávat kvalitu různých jazykových modelů (trénovány na různých datech, pomocí jakých n-gramů, jak vyhlazených apod.).

Je možné použít 2 přístupy: extrinsic a intrinsic vyhodnocení.

Dobrý model by měl přiřadit dobrému textu vyšší pravděpodobnost než špatnému textu.

Pokud máme nějaký testovací text, můžeme spočítat pravděpodobnost, jakou mu přiřazuje zkoumaný LM. Lepší LM by mu měl přiřadit vyšší pravděpodobnost.

Cross-entropy (křížová entropie)

$$H(p_{LM}) = -\frac{1}{n} \log p_{LM}(w_1, w_2, \ldots w_n)$$
$$= -\frac{1}{n} \sum_{i=1}^{n} \log p_{LM}(w_i | w_1, \ldots w_{i-1})$$

Křížová entropie je průměrná hodnota záporných logaritmů pravděpodobností slov v testovacím textu. Odpovídá míře nejistoty pravděpodobnostního rozložení (zde LM). Čím menší, tím lepší.

Dobrý LM by měl dosahovat entropie blízké skutečné entropii jazyka. Tu nelze změřit, ale existují relativně spolehlivé odhady (např. **Shannonova hádací hra**). Pro angličtinu je entropie na znak rovna cca 1.3 bitu.

Perplexita

$$PP = 2^{H(p_{LM})}$$

$$PP(W) = p_{LM}(w_1 w_2 w_3 \ldots w_N)^{-\frac{1}{N}}$$

Perplexita je jednoduchá transformace křížové entropie.

Dobrý model by neměl plýtvat p na nepravděpodobné jevy a naopak.

Čím nižší entropie, tím lépe $\rightarrow$ čím nižší perplexita, tím lépe.

Vyhlazování jazykových modelů

Problém: pokud není v datech určitý n-gram, který se vyskytne v řetězci w, pro který hledáme pravděpodobnost, bude $p(w) = 0$.

Potřebujeme rozlišovat p i pro *neviděná data*. Musí platit

$$\forall w.p(w) > 0$$

Ještě větší je problém u modelů vyšších řádů.

Snaha o úpravení reálných počtů n-gramů na očekávané počty těchto n-gramů v libovolných datech (jiných korpusech).

Add-one vyhlazování (Laplace)

Maximum likelihood estimation přiřazuje pravděpodobnost na základě vzorce

$$p = \frac{c}{n}$$

Add-one vyhlazování používá upravený vzorec

$$p = \frac{c+1}{n+v}$$

kde v je počet všech možných n-gramů. To je však velmi nepřesné, neboť všech možných kombinací je většinou řádově víc než ve skutečnosti (Europarl korpus má 86,700 tokenů, tedy víc jak 7,5 mld možných bigramů. Ve skutečnosti má korpus 30 mil. slov, tedy maximálně 30 mil. bigramů.) Vyhlazování nadhodnocuje neviděné n-gramy.

Add-α vyhlazování

Nebudeme přidávat 1, ale koeficient α. Ten lze odhadnout tak, aby add-α vyhlazování bylo spravedlivější.

$$p = \frac{c + \alpha}{n + \alpha v}$$

α můžeme experimentálně zjistit: zvolit více různých a hledat pomocí perplexity nejlepší z nich. Typicky bude spíše malé (0.000X).

Deleted estimation

Neviděné n-gramy můžeme vytvořit uměle tak, že použijeme druhý korpus, případně část trénovacího korpusu. N-gramy obsažené v jednom a ne v druhém nám pomohou odhadnout množství neviděných n-gramů obecně.

Např. bigramy, které se nevyskytují v trénovacím korpusu, ale vyskytují se v druhém korpusu milionkrát (a všech možných bigramů je cca 7,5 mld), se vyskytnou cca

$$\frac{10^6}{7.5 \times 10^9} = 0.00013\times$$

Good–Turing vyhlazování

Potřebujeme upravit počet výskytů v korpusu tak, aby odpovídal obecnému výskytu v textu. Použijeme *frekvenci frekvencí*: počet různých n-gramů, které se vyskytují n-krát.

Použijeme četnost hapax legomena pro odhad četnostní nikdy neviděných dat.

$$r^* = (r+1)\frac{N_{r+1}}{N_r}$$

Speciálně pro n-gramy, které nejsou v korpusu máme

$$r_0^* = (0+1)\frac{N_1}{N_0} = 0.00015$$

kde $N_1 = 1.1 \times 10^6$ a $N_0 = 7.5 \times 10^9$ (Europarl korpus).

Ukázka Good–Turing vyhlazování (Europarl)

r	**FF**	r^*
0	7 514 941 065	0,00015
1	1 132 844	0,46539
2	263 611	1,40679
3	123 615	2.38767
4	73 788	3,33753
5	49 254	4,36967
6	35 869	5,32929
8	21 693	7,43798
10	14 880	9,31304
20	4 546	19.54487

Srovnání metod vyhlazování (Europarl)

metoda	**perplexita**
add-one	382,2
add-α	113,2
deleted est.	113,4
Good–Turing	112,9

Interpolace a back-off

Předchozí metody zacházely se všemi neviděnými n-gramy stejně. Předpokládejme 3-gramy:

nádherná červená řepa
nádherná červená mrkev

I když ani jeden nemáme v trénovacích datech, první 3-gram by měl být pravděpodobnější.

Budeme využívat pravděpodobnosti n-gramů nižších řádů, u kterých máme k dispozici více dat:

červená řepa
červená mrkev

Interpolace

Použijeme interpolaci:

$$p_I(w_3|w_1\,w_2) = \lambda_1 p(w_3) \times \lambda_2 p(w_3|w_2) \times \lambda_3 p(w_3|w_1\,w_2)$$

Pokud máme hodně dat, můžeme věřit modelům vyšších řádů a přiřadit odpovídajícím pravděpodobnostem větší váhu.

p_I je pravděpodobnostní rozložení, proto musí platit:

$$\forall \lambda_n : 0 \leq \lambda_n \leq 1$$

$$\sum_n \lambda_n = 1$$

Velké jazykové modely – počet n-gramů

Kolik je různých n-gramů v korpusu?

řád	**unikátní**	**singletony**
unigram	86 700	33 447 (38,6 %)
bigram	1 948 935	1 132 844 (58,1 %)
trigram	8 092 798	6 022 286 (74,4 %)
4-gram	15 303 847	13 081 621 (85,5 %)
5-gram	19 882 175	18 324 577 (92,2 %)

Europarl, 30 miliónů tokenů.