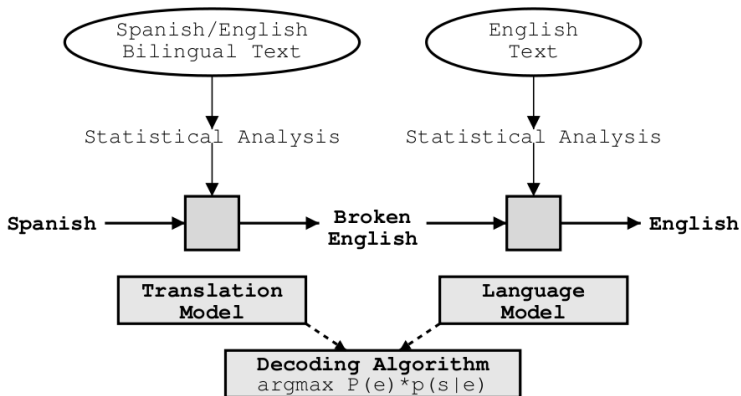


- 1 Úvod do statistického strojového překladu
- 2 Jazykové modely
- 3 Překladové modely

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



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 Consortium (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 unie (20 jazyků)

Pravidelné události v oblasti SMT, soutěže

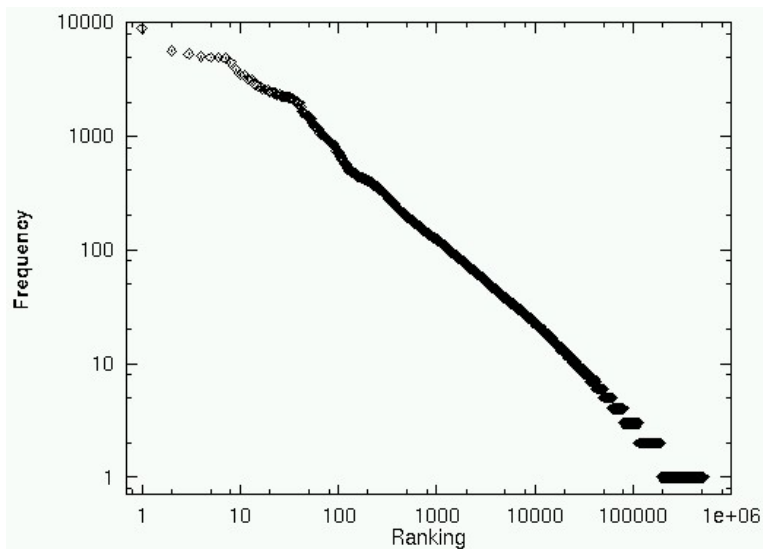
Většinou roční vyhodnocování kvality SMT. Tvorba testovacích sad, manuální vyhodnocování dat, referenční systémy.

- 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 → 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* → *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 dramát (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, \dots, e_{n_e} a f_1, \dots, f_{n_f}
- hledáme páry s_1, \dots, s_n
- $s_i = (\{f_{\text{start}-f(i)}, \dots, f_{\text{end}-f(i)}\}, \{e_{\text{start}-e(i)}, \dots, e_{\text{end}-e(i)}\})$

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

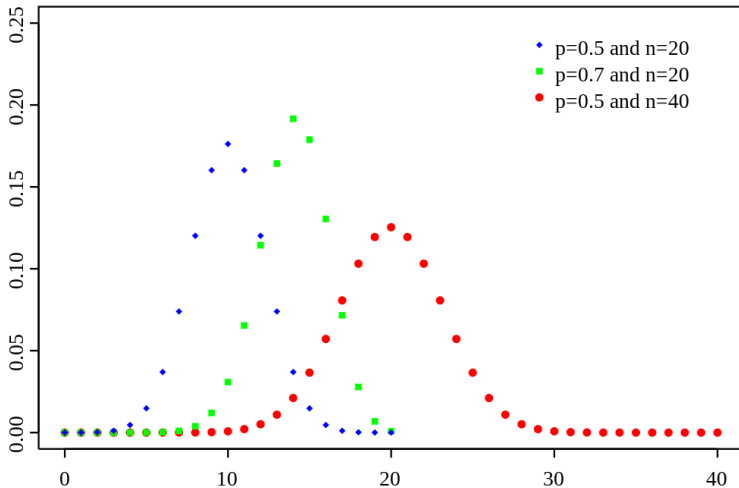
- graf hodnot pravděpodobnosti pro elementární jevy náhodné veličiny
- **rovnoměrné**: hod kostkou, mincí (diskrétní veličina)
- **binomické**: vícenásobný hod

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

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

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

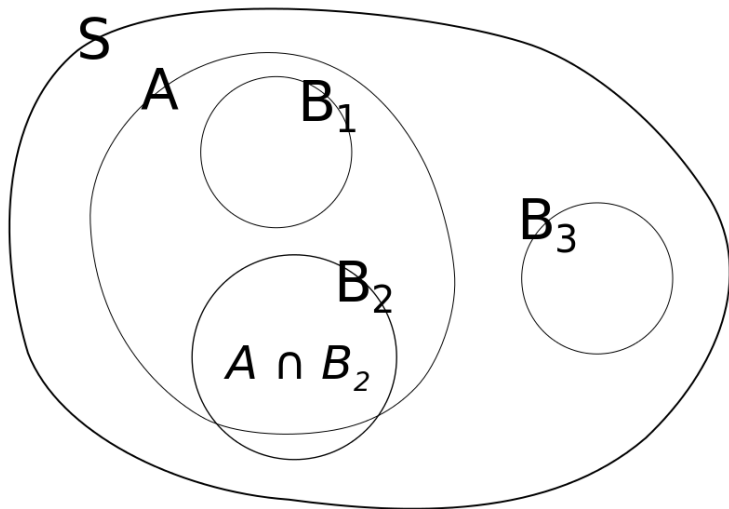
Binomické rozložení



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: $\forall x, y : p(x, y) = p(x) \cdot p(y)$
- **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

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

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

Statistika II

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

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-l, 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 vyšší 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

- 1 Úvod do statistického strojového překladu
- 2 Jazykové modely**
- 3 Překladové modely

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

$p_{LM}(\text{včera jsem jel do Brna})$

$p_{LM}(\text{včera jel do Brna jsem})$

$p_{LM}(\text{jel jsem včera do Brna})$

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* → pero)
- ve speciálních textech nelze použít, ale
- LM pomáhají s **WSD pomocí kontextu**
- $p_{LM}(i \text{ go home}) \geq p_{LM}(i \text{ go house})$

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

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

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**:

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

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

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

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):

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

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

<i>w</i>	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)

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

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

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

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

Perplexita je jednoduchá transformace křížové entropie.

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

Čím nižší entropie, tím lépe → 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 úpravu 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.

- 1 Úvod do statistického strojového překladu
- 2 Jazykové modely
- 3 Překladové modely**

Lexikální překlad

Standardní slovník neobsahuje informace o tom, jak často se překládá dané slovo na své různé překladové ekvivalenty.

key → *klíč*, *tónina*, *klávesa*

Jak často jsou zastoupeny jednotlivé překlady v překladech?

key → *klíč* (0.7), *tónina* (0.18), *klávesa* (0.12)

Potřebujeme lexikální překladové pravděpodobnostní rozložení p_f s vlastností

$$\sum_e p_f(e) = 1$$

$$\forall e : 0 \leq p_f(e) \leq 1$$

$$p_{\text{klíč}}(\text{key})? p_{\text{mrkev}}(\text{carrot})$$

Zarovnání slov, zarovnávací funkce

Překlady si často neodpovídají v počtu slov ani ve slovosledu.
Zavádí se *alignment function*

$$a : j \rightarrow i$$

kde j je pozice odpovídajícího slovo v cílové větě (čeština), i je pozice ve zdrojové větě (angličtina).

i:	1	2	3	4	5
	the	castle	is	very	old
	ten	hrad	je	velmi	starý
j:	1	2	3	4	5

a je funkce, tedy pro každé slovo w_e z cílové věty existuje právě jedno slovo w_f ze zdrojové věty.

Zarovnání slov – další případy

- jiný slovosled:

it was written here

bylo to zde napsané

$a : 1 \rightarrow 2, 2 \rightarrow 1, 3 \rightarrow 4, 4 \rightarrow 3$

- jiný počet slov:

jsem maličký

i am very small

$a : 1 \rightarrow 1, 2 \rightarrow 1, 3 \rightarrow 2, 4 \rightarrow 2$

- slova bez překladových ekvivalentů:

have you got it ?

máš to ?

$a : 1 \rightarrow 1, 2 \rightarrow 4, 3 \rightarrow 5$

- opačný případ, přidáme nové slovo NULL, pozice 0:

NULL laugh

smát se

$a : 1 \rightarrow 1, 2 \rightarrow 0$

IBM model 1

Nemůžeme hledat p_f pro jednotlivé věty. Překlad rozložíme do menších kroků, budeme používat p_f pro slova. Tomuto přístupu se říká *generative modeling*.

Překladový model IBM-1 je definován jako

$$p(\mathbf{e}, \mathbf{a}|\mathbf{f}) = \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} t(e_j | f_{a(j)})$$

kde $\mathbf{e} = (e_1, \dots, e_{l_e})$ je cílová věta, $\mathbf{f} = (f_1, \dots, f_{l_f})$ zdrojová věta, l_e je délka cílové věty, l_f délka zdrojové věty, ϵ je normalizující konstanta, aby byl výsledný součin pravděpodobnostní rozložení. $(l_f + 1)^{l_e}$ je počet všech možných zarovnání mezi \mathbf{e} a \mathbf{f} , přičemž k l_f přičítáme 1 kvůli speciálnímu slovu NULL, t je pravděpodobnostní překladová funkce.

Výpočet překladové pravděpodobnosti

Pro výpočet $p(\mathbf{e}, \mathbf{a} | \mathbf{f})$ potřebujeme znát hodnotu funkce t pro všechna slova (věty).

K tomu budeme využívat základní zdroj pro SMT: **paralelní korpus** se zarovnanými větami.

Bohužel nemáme zarovnání slov mezi sebou. To je úkol tzv. **word-alignment**.

Ke slovu přichází **expectation-maximization (EM)** algoritmus.

EM algoritmus

- 1 inicializuj model (typicky uniformní p. rozložení)
- 2 aplikuj model na data (krok expectation)

hledáme $p(a|e, f) = \frac{p(e, a|f)}{p(e|f)}$

kde $p(e|f) = \sum_a p(e, a|f)$
- 3 uprav model podle dat (krok maximization)

upravíme počty zarovnání slova w_e na w_f (funkce c) pomocí předchozího

$$c(w_e|w_f; e, f) = \sum_a p(a|e, f) \sum_{j=1}^{l_e} \delta(e, e_j) \delta(f, f_{a(j)})$$

kde $\delta(x, y) = 1 \iff x == y$, jinak 0
- 4 opakuj E-M kroky dokud je co zlepšovat

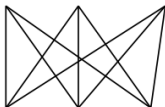
Překladová pravděpodobnost z EM algoritmu

Výsledná překladová pravděpodobnost se vypočítá pomocí c :

$$t(w_e|w_f) = \frac{\sum_{(e,f)} c(w_e|w_f; e, f)}{\sum_{w_e} \sum_{(e,f)} c(w_e|w_f; e, f)}$$

Ilustrace EM algoritmu – inicializace

... la maison ... la maison blue ... la fleur ...



... the house ... the blue house ... the flower ...

Ilustrace EM algoritmu – výsledná fáze

... la maison ... la maison bleu ... la fleur ...

... the house ... the blue house ... the flower ...



$p(\text{la}|\text{the}) = 0.453$
 $p(\text{le}|\text{the}) = 0.334$
 $p(\text{maison}|\text{house}) = 0.876$
 $p(\text{bleu}|\text{blue}) = 0.563$
 ...

IBM modely

IBM model 1 je značně jednoduchý. Neuvažuje kontext, neumí přidávat a vypouštět slova. Všechna různá zarovnání považuje za stejně pravděpodobné. Ostatní modely vždy přidávají něco navíc.

- IBM-1: lexikální překlad
- IBM-2: přidává model absolutního zarovnání
- IBM-3: přidává model **fertility**
- IBM-4: přidává model relativního zarovnání
- IBM-5: ošetřuje nedostatečnosti předchozích modelů

IBM-2

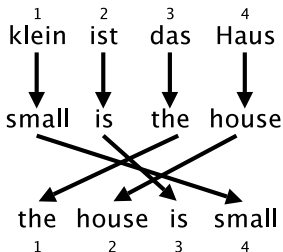
Pro IBM-1 jsou všechny možné překlady s různým uspořádáním slov stejně pravděpodobné. IBM-2 přidává explicitní model pro zarovnání, tzv. **alignment probability distribution**:

$$a(i|j, l_w, l_f)$$

kde i je pozice zdrojového slova, j pozice cílového slova.

IBM-2 – 2 kroky překladu

Překlad se tedy rozdělí na dva kroky. V prvním se přeloží lexikální jednotky, v druhém se podle modelu zarovnání přeskupí přeložená slova.



lexical translation step

alignment step

IBM-2

První krok je stejný jako u IBM-1, používá se $t(e|f)$. Funkce a i pravděpodobnostní rozložení a je v opačném směru než je překlad. Obě rozložení se kombinují do vzorce pro IBM-2:

$$p(e, a|f) = \epsilon \prod_{j=1}^{l_e} t(e_j|f_{a(j)}) a(a(j)|j, l_e, l_f)$$

$$\begin{aligned} p(e|f) &= \sum_a p(e, a|f) \\ &= \epsilon \prod_{j=1}^{l_e} \sum_{i=0}^{l_f} t(e_j|f_i) a(i|j, l_e, l_f) \end{aligned}$$

IBM model 3

Modely IBM-1,2 neuvažují vlastnost, kdy se jedno slovo přeloží na více slov, případně se nepřeloží vůbec. IBM-3 řeší tento problém zavedením **fertility**, které je modelována pravd. rozložením

$$n(\phi|f)$$

Pro každé zdrojové slovo f rozložení n říká, na kolik cílových slovo se obvykle f přeloží.

$$n(0|a) = 0.999$$

$$n(1|\text{king}) = 0.997$$

$$n(2|\text{steep}) = 0.25$$

...

Vložení tokenu NULL

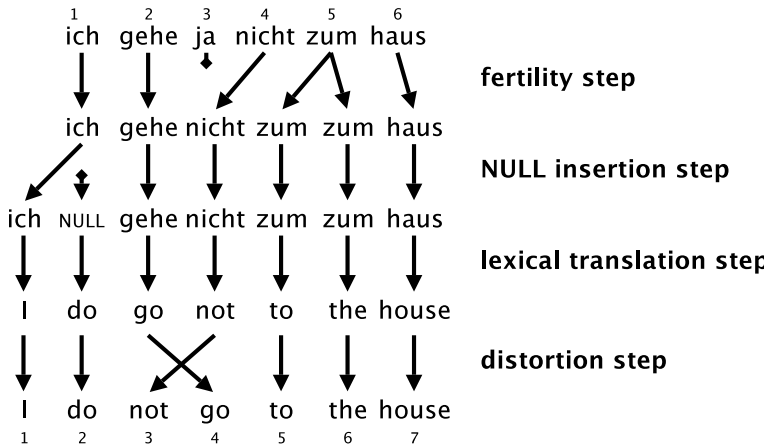
Pokud chceme správně překládat do cílového jazyka, který používá slova, jež nemají ve zdrojovém jazyce překladové ekvivalenty, musíme řešit vkládání pomocného tokenu NULL.

Nepoužívá se $n(x|NULL)$, protože vložení NULL záleží na délce věty.

Přidáme tedy další krok **vložení NULL** do procesu překladu. Používají se p_1 a $p_0 = 1 - p_1$, kde p_1 znamená pravděpodobnost vložení tokenu NULL za libovolné slovo ve větě.

IBM modely

IBM-3



IBM-3 – distortion

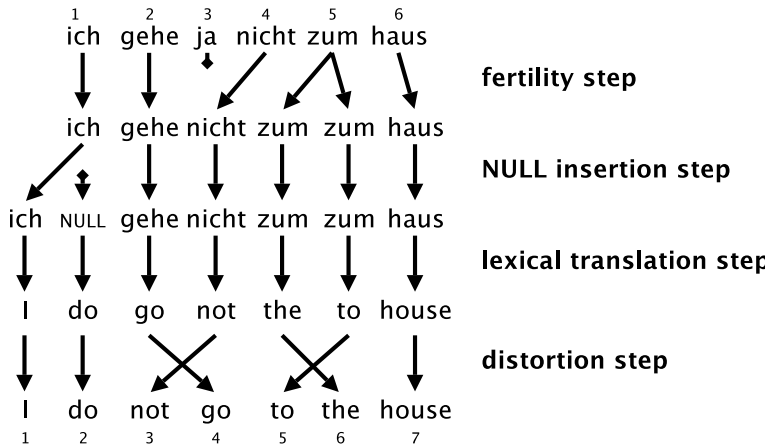
Poslední krok je téměř shodný s 2. krokem překladového procesu IBM-2 a je modelován tzv. **distortion probability distribution**:

$$d(j|i, l_e, l_f)$$

kteřá modeluje pozice v opačném pořadí: pro zdrojové slovo na pozici i modeluje pozici j cílového slova.

Proces překladu z předchozího obrázku se může drobně lišit (viz další obrázek).

IBM-3



IBM-4, IBM-5

IBM-4

Problém distorze tkví v řídkých datech pro dlouhé věty. IBM-4 zavádí tzv. **relativní distorzi**, kde změny pozic slov závisí na předcházejících slovech. Vychází z předpokladu, že se překládá po frázích, které se přesunují vcelku, případně že některé přesuny jsou více časté (angličtina: ADJ SUB, francouzština SUB ADJ apod.).

IBM-5

Tento model řeší další nedostatky předchozích modelů. Např. hlídá, aby se dvě různá zdrojová slova nedostala na jednu pozici v cílové větě atd.

Problémy se zarovnáním slov

	john	biss	ins	grass
john				
kicked				
the				
bucket				

	john	wohnt	hier	nicht
john				
does		?		?
not				
live				
here				