

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

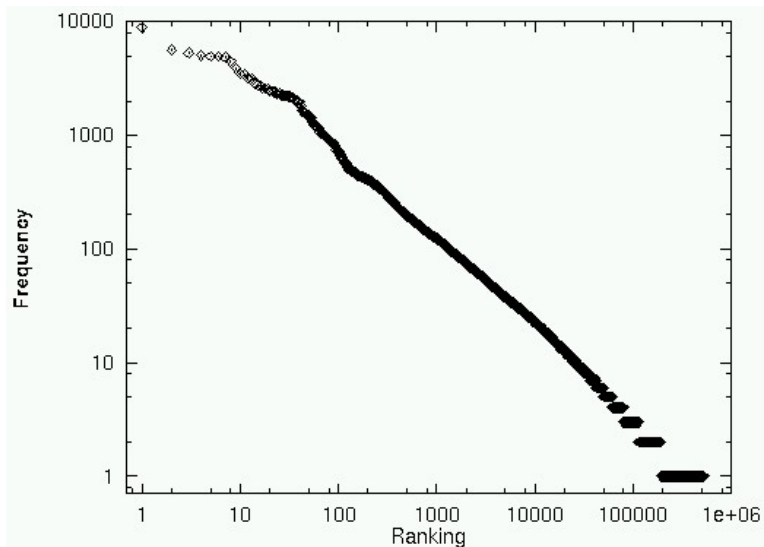
Data pro SMT – (paralelní) korpusy

- 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)

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 $r \times f = c$
- překlipy, 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

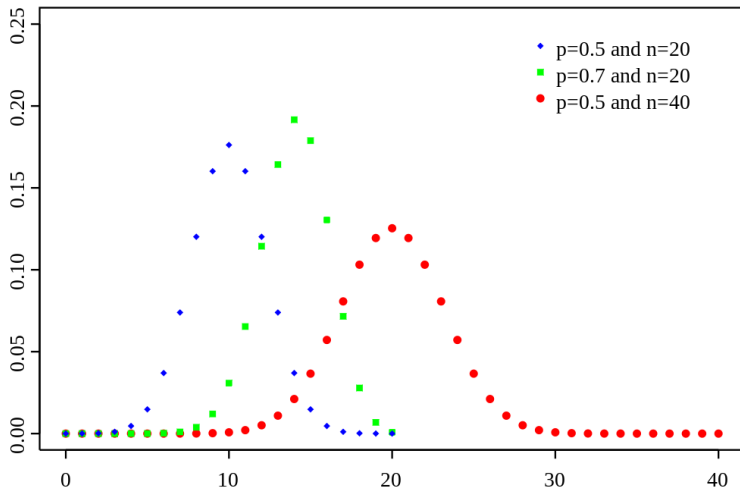
- graf hodnot pravděpodobnosti pro elementární jevy náhodné veličiny
- **rovnoměrné**: hod kostkou, mincí (diskrétní veličina)
- **binomické**: vícenásobný hod

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

- **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: $p(y|x) = p(y)$

Bayesovo pravidlo

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

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

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

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.

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 |

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.

Dobří 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 úpravění reálných počtů n -gramů na očekávané počty těchto n -gramů v libovolných datech (jiných korpusech).

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

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

Srovnání metod vyhlazování (Europarl)

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

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

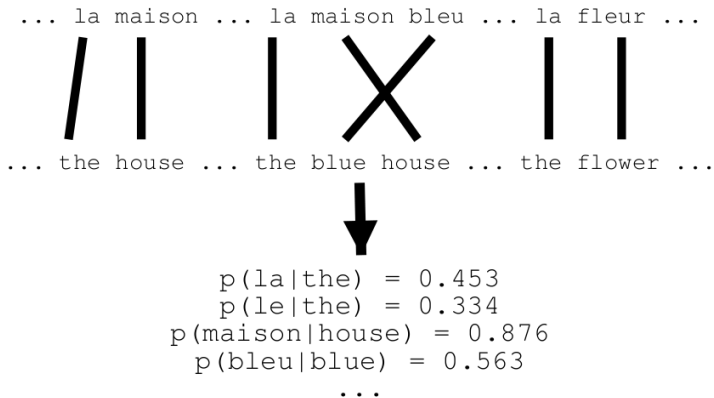
$a : 1 \rightarrow 1, 2 \rightarrow 0$

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 – výsledná fáze



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

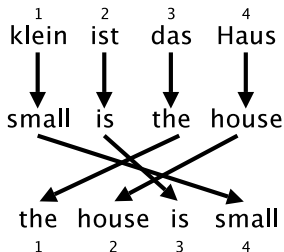
Pro IBM-1 jsou všechny možné překlady s různým uspořádáním slov stejně pravděpodobné. IBM-2 přidává explicitní model pro zarovnání, tzv. **alignment probability distribution**:

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

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-3 – distortion

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

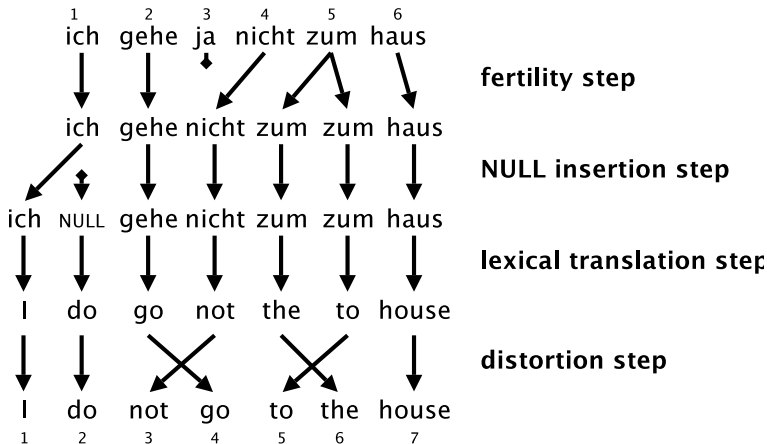
$$d(j|i, l_e, l_f)$$

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 modely

IBM-3



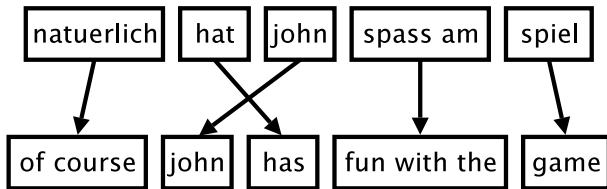
Problémy se zarovnáním slov

| | john | biss | ins | grass |
|--------|------|------|-----|-------|
| john | ■ | | | |
| kicked | | ■ | ■ | ■ |
| the | | ■ | ■ | ■ |
| bucket | | ■ | ■ | ■ |

| | john | wohnt | hier | nicht |
|------|------|-------|------|-------|
| john | ■ | | | |
| does | | ■? | | ■? |
| not | | | | ■ |
| live | | ■ | | |
| here | | | ■ | |

Frázový překladový model

State-of-the-art statistického strojového překladu. Nepřekládají se pouze samostatná slova. Když to jde, tak i celé sekvence slov.



Fráze nejsou lingvisticky motivované, pouze statisticky. Německé *am* se zřídka překládá jedním slovem *with*. Statisticky významný kontext *spass am* pomáhá správnému překladu. Klasické fráze by se dělily jinak: (*fun (with (the game))*).

Výhody PBTM

- často překládáme $n : m$ slov, slovo je tedy nevhodný atomický prvek
- překlad skupin slov pomáhá řešit překladové víceznačnosti
- můžeme se učit překládat delší a delší fráze
- jednodušší model: neuvažujeme fertilitu, NULL token atd.

Phrase-based model – vzorec

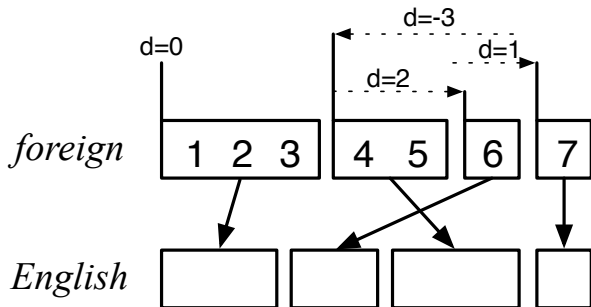
Překladová pravděpodobnost $p(f|e)$ se rozloží na fráze

$$p(\bar{f}_1^l | \bar{e}_1^l) = \prod_{i=1}^l \phi(\bar{f}_i | \bar{e}_i) d(\text{start}_i - \text{end}_{i-1} - 1)$$

Věta f se rozloží na l frází \bar{f}_i , všechna dělení jsou stejně pravděpodobná. Funkce ϕ je překladová pravděpodobnost pro fráze. Funkce d je přerovnávací model založený na vzdálenosti (**distance-based reordering model**), modelujeme pomocí předchozí fráze. start_i je pozice prvního slova ve frázi věty f , které se překládá na i tou frází věty e .

Distance-based reordering model

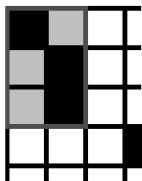
Preferuje se minimální přesun frází. Čím větší přesun (měří se na straně výchozího jazyka), tím dražší tato operace je.



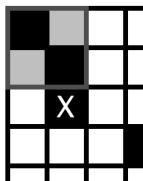
Budování překladové tabulky frází

Použijeme zarovnání slov (získané pomocí EM algoritmu pro IBM-1) a pak hledáme **konzistentní fráze**.

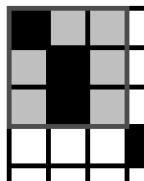
Fráze \bar{f} a \bar{e} jsou konzistentní se zarovnáním A , pokud všechna slova f_1, \dots, f_n ve frázi \bar{f} , která mají zarovnání v A , jsou zarovnaná se slovy e_1, \dots, e_n ve frázi \bar{e} a naopak.



konzistentní



nekonzistentní



konzistentní

Extrahované fráze

| | |
|--|--|
| michael assumes that he will stay in the house | michael geht davon aus / geht davon aus , dass / , dass er bleibt im haus |
| michael assumes assumes that assumes that he that he in the house michael assumes that ... | michael geht davon aus / michael geht davon aus , geht davon aus , dass geht davon aus , dass er dass er / , dass er im haus michael geht davon aus , dass ... |

Odhad pravděpodobnosti frází

Odhad pravděpodobnosti frází

$$\phi(\bar{f}|\bar{e}) = \frac{\text{count}(\bar{e}, \bar{f})}{\sum_{\bar{f}_i} \text{count}(\bar{e}, \bar{f}_i)}$$

Model statistického překladu založený na frázích

$$e^* = \operatorname{argmax}_e \prod_{i=1}^I \phi(\bar{f}_i|\bar{e}_i) d(\text{start}_i - \text{end}_{i-1} - 1) \prod_{i=1}^{|\mathbf{e}|} p_{LM}(e_i|e_1 \dots e_{i-1})$$

Vážený frázový model

$$e^* = \operatorname{argmax}_e \prod_{i=1}^I \phi(\bar{f}_i|\bar{e}_i)^{\lambda_\phi} d(\text{start}_i - \text{end}_{i-1} - 1)^{\lambda_d} \prod_{i=1}^{|\mathbf{e}|} p_{LM}(e_i|e_1 \dots e_{i-1})^{\lambda_{LM}}$$

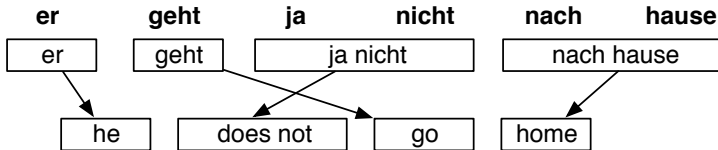
Dekódování

Máme jazykový model p_{LM} a překladový model $p(f|e)$.
Potřebujeme vyhledat z exponenciálního množství všech překladů ten, kterému modely přiřazují nejvyšší pravděpodobnost.

Používá se **heuristické** prohledávání. Nemáme tedy garantováno, že nalezneme nejpravděpodobnější překlad.

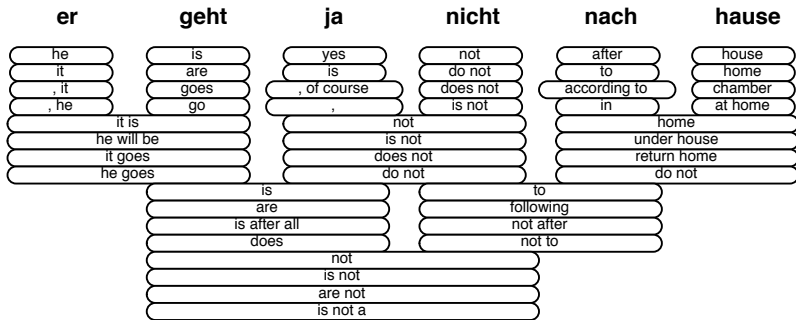
Chyby překladu jsou způsobeny 1) chybou v prohledávání, kdy není nalezen nejlepší překlad v celém prohledávacím prostoru a 2) chybou v modelech, kdy i nejlepší překlad podle pravděpodobnostních funkcí není ten správný.

Překlad věty po frázích



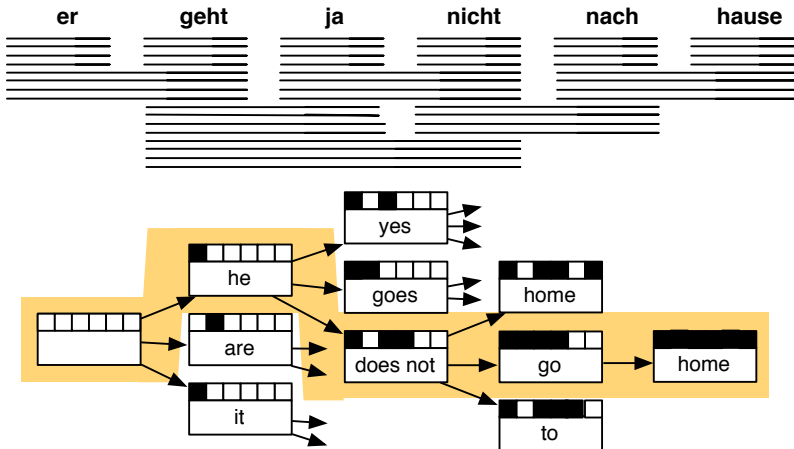
V každém kroku překladu počítáme předběžné hodnoty pravděpodobností z překladového modelu, přerovnávacího modelu a jazykového modelu.

Prohledávací prostor překladových hypotéz



Rozšiřujeme hypotézy v exponenciálním prostoru všech možných překladů. Různými metodami se snažíme prostor zmenšit.

Budování hypotéz, beam search



Beam search

Beam search používá tzv. *breadth-first* prohledávání. Na každé úrovni prohledávacího stromu generuje všechny následovníky stavů (uzlů) nadané úrovni, uspořádává je podle heuristik. Ukládá ovšem pouze omezený počet nejlepších stavů na každé úrovni (beam width). Pouze tyto stavy jsou prozkoumávány dále. Čím větší beam width, tím méně stavů je prořezáno. Při nekonečné šířce jde o breadth-first prohledávání. Šířka určuje spotřebu paměti. Nejlepší konečný stav ovšem nemusí být nalezen, může být v nějaké fázi prořezán.