Morphology in MT

Ondřej Bojar

April 2, 2020


Charles University
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics

Outline

- Problems caused by rich morphology.
  - Morphological richness of Czech.
  - Combinatorial explosion of Czech word forms.
  - Margin for improvement in BLEU.
- Morphology in PBMT:
  - Factored PBMT.
  - Reverse self-training.
- Morphology in NMT.
  - Subword units, BPE.

Morphological Richness (in Czech)

	Czech	English
Rich morphology	$\geq$ 4,000 tags possible $\geq$ 2,300 tags seen	50 used
Word order	free	rigid

News Commentary Corpus	Czech	English
Sentences	55,676	
Tokens	1.1M	1.2M
Vocabulary (word forms)	91k	40k
Vocabulary (lemmas)	34k	28k

Czech tagging and lemmatization: Hajič and Hladká (1998)
English tagging (Ratnaparkhi, 1996) and lemmatization (Minnen et al., 2001).

Morphological Explosion in Czech

MT chooses output words in a form:

- Czech nouns and adjs.: 7 cases, 4 genders, 3 numbers, ...
- Czech verbs: gender, number, aspect (im/perfective), ...

I	saw	two	green	striped	cats	.
já	pila	dva	zelený	pruhovaný	kočky	.
	pily	dvě	zelená	pruhovaná	koček	
	...	dvou	zelené	pruhované	kočkám	
	viděl	dvěma	zelení	pruhovaní	kočkách	
	viděla	dvěmi	zeleného	pruhovaného	kočkami	
	...		zelených	pruhovaných		
	uviděl		zelenému	pruhovanému		
	uviděla		zeleným	pruhovaným		
	...		zelenou	pruhovanou		
	viděl jsem		zelenými	pruhovanými		
	viděla jsem		...	...		

Morphological Explosion Elsewhere

Compounding in German:

- Rindfleischetikettierungsüberwachungsaufgabenübertragungsgesetz.
  "beef labelling supervision duty assignment law"

Agglutination in Hungarian or Finnish:

istua	"to sit down" (istun = "I sit down")
istahtaa	"to sit down for a while"
istahdan	"I'll sit down for a while"
istahtaisin	"I would sit down for a while"
istahtaisinko	"should I sit down for a while?"
istahtaisinkohan	"I wonder if I should sit down for a while"

Margin: Lemmatized BLEU

- Lemmatized BLEU:
  - Lemmatized MT output against lemmatized references.
  - Does not penalize errors in word forms.
    $\Rightarrow$ An indication of achievable BLEU score.

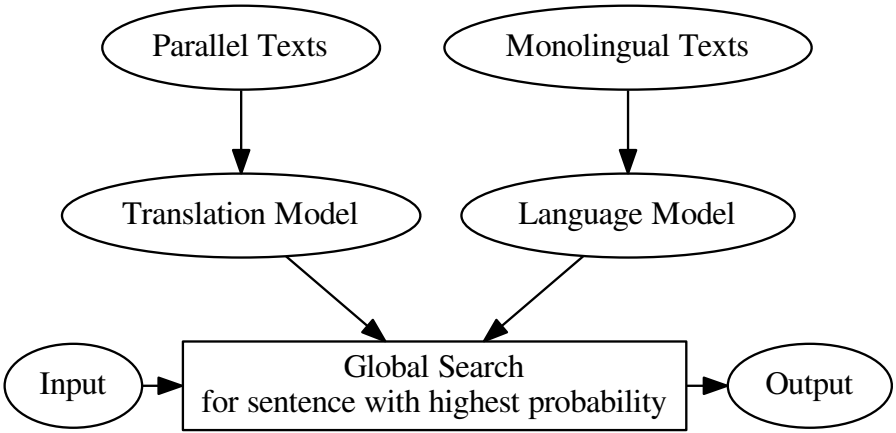
English→Czech phrase-based translation:

	PCEDT	Project Syndicate
Regular BLEU, lowercase	25.2	∼12
Lemmatized BLEU	33.6	∼20

- Margin for improvement: ∼8 points in both experiments.

Morphology in Phrase-Based MT

PBMT Main Components



LM over Forms Insufficient

Possible translations differring in morphology:

two	green	striped	cats	
dvou	zelená	pruhovaný	kočkách	← garbage
dva	zelené	pruhované	kočky	← 3grams ok, 4gram bad
dvě	zelené	pruhované	kočky	← correct nominative/accusative
dvěma	zeleným	pruhovaným	kočkám	← correct dative

- 3-gram LM too weak to ensure agreement.
- 3-gram LM possibly already too sparse!

Explicit Morphological Target Factor

- Add morphological tag to each output token:

two	green	striped	cats	
dvou	zelená	pruhovaný	kočkách	← garbage
fem-loc	*neut-acc*	*masc-nom-sg*	*fem-loc*	
dva	zelené	pruhované	kočky	← 3-grams ok, 4-gram bad
masc-nom	*masc-nom*	*masc-nom*		
	fem-nom	*fem-nom*	*fem-nom*	
dvě	zelené	pruhované	kočky	← correct nominative/accusative
fem-nom	*fem-nom*	*fem-nom*	*fem-nom*	
fem-acc	*fem-acc*	*fem-acc*	*fem-acc*	
dvěma	zeleným	pruhovaným	kočkám	← correct dative
fem-dat	*fem-dat*	*fem-dat*	*fem-dat*	

Advantages of Explicit Morphology for LM

- LM over morphological tags generalizes better.
  - p(dvě kočkách) $<$ p(dvě kočky) ...surely
    But we would need to see all combinations of *pruhovaný* and *kočka*!
    $\Rightarrow$ Better to ask if p(*fem-nom fem-loc*) $<$ p(*fem-nom fem-nom*)

    which is trained on any feminine adj+noun.
- But still does not solve everything.
  - p(dvě zelené) $\gtrless$ p(dva zelené) ... bad question anyway!
    Not solved by asking if p(*fem-nom fem-nom*) $\gtrless$ p(*masc-nom masc-nom*).
- Tagset size smaller than vocabulary.
  $\Rightarrow$ can afford e.g. 7-grams:
  p(*masc-nom fem-nom fem-nom*) $<$ p(*fem-nom fem-nom fem-nom*)

Motivation from Translation Model

Availability of translations of "knee caps" in parallel data:

Case	Surface form	50K	500K	5M	50M
nom	čéšky	●	●	●	●
gen	čéšek	–	●	●	●
dat	čéškám	–	–	●	●
acc	čéšky	○	○	●	●
voc	čéšky	○	○	○	○
loc	čéškách	–	●	●	●
instr	čéškami	–	–	–	●

⇒ You need to have 50M parallel sentences to translate:
"What's wrong with my knee caps?"

"●" ... the word was seen in the particular case,
"○" ... the surface form was seen but in a different case.
Reproduced from Huck et al. (2017b).

Advantages of Explicit Morphology for TM

Main idea:

- separate translation of lemmas and morphology:

$$[\![\text{ src lemma }]\!] \Rightarrow [\![\text{ tgt lemma }]\!]$$

$$[\![\text{ src morphology }]\!] \Rightarrow [\![\text{ tgt morphology }]\!]$$

- generate target forms based on the lemma and morphology:

$$\begin{matrix} [\![\text{ tgt lemma }]\!] \\ [\![\text{ tgt morphology }]\!] \end{matrix} \Rightarrow [\![\text{ tgt surface form }]\!]$$

Factored Phrase-Based MT

- Both input and output words can have more factors.
- Arbitrary number and order of:

  Mapping steps ($\rightarrow$)

  Translate (phrases of) source factors to target factors.

  two green $\rightarrow$ dvě zelené

  Generation steps ($\downarrow$)

  Generate target factors from target factors.

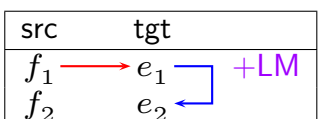
  dvě $\rightarrow$ *fem*-*nom*; dva $\rightarrow$ *masc*-*nom*

  $\Rightarrow$ Ensures "vertical" coherence.

  Target-side language models (+LM)

  Applicable to various target-side factors.

  $\Rightarrow$ Ensures "horizontal" coherence.



(Koehn and Hoang, 2007)

Translation Process in Factored PBMT

Input: *(knee caps, knee cap, Adj NN-plur)*

1. Translation step: lemma ⇒ lemma:
   (?, čéška, ?)
2. Generation step: lemma ⇒ morphological tag
   (?, čéška, N1-sg), *(?, čéška, N2-sg)*, *(?, čéška, N1-pl)*, *(?, čéška, N2-pl)*, ...
3. Translation step: morphological tag ⇒morphological tag
   ... This reorders the options, so that plural is more likely:
   (?, čéška, N1-pl), *(?, čéška, N2-pl)*, *(?, čéška, N1-sg)*, *(?, čéška, N2-sg)*, ...
4. Generation step: lemma, morphological tag ⇒ surface form
   (čéšky, čéška, N1-pl), *(čéšek, čéška, N2-pl)*, ..., *(čéškami, čéška, N7-pl)*

This and several following slides reuse slides by Philipp Koehn, MT Marathon 2009.

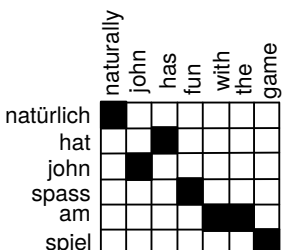
Factored PBMT Model

- Extension of the phrase-based model:
  1. Phrase Extraction for mapping steps.
  2. Extraction for generation steps.
  3. Decoding.
- Each step simply brings one or more feature functions:
  - Fits nicely into the log-linear model,
  - Weights trained by the discriminative training (MERT).
- The order of the operations is defined in configuration.

Factored Phrase Extraction (1/3)

As in standard phrase-based MT:

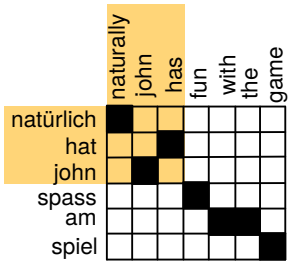
1. Run sentence and word alignment,



Factored Phrase Extraction (2/3)

As in standard phrase-based MT:

1. Run sentence and word alignment,
2. Extract all phrases consistent with word alignment.

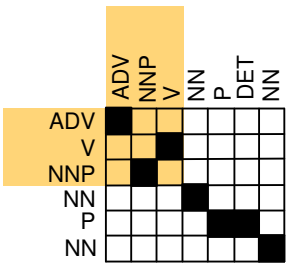


$\Rightarrow$ Extracted: natürlich hat john $\rightarrow$ naturally john has

Factored Phrase Extraction (3/3)

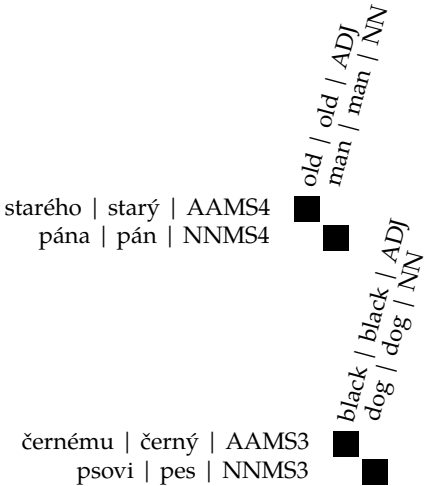
As in standard phrase-based MT:

1. Run sentence and word alignment,
2. Extract same phrases, just another factor from each word.

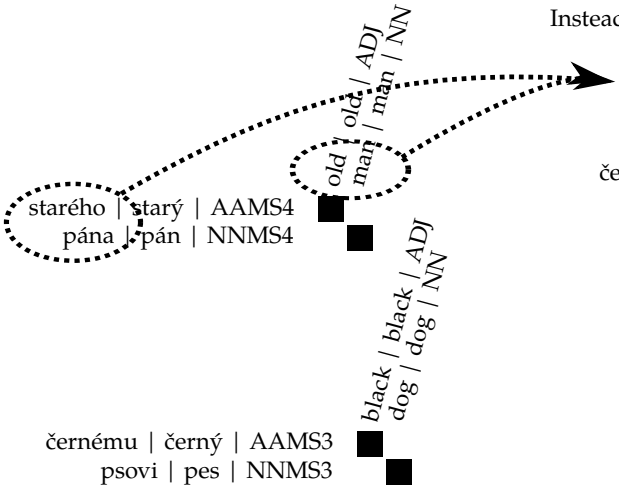


$\Rightarrow$ Extracted: ADV V NNP $\rightarrow$ ADV NNP V

The Benefit Illustrated Once More



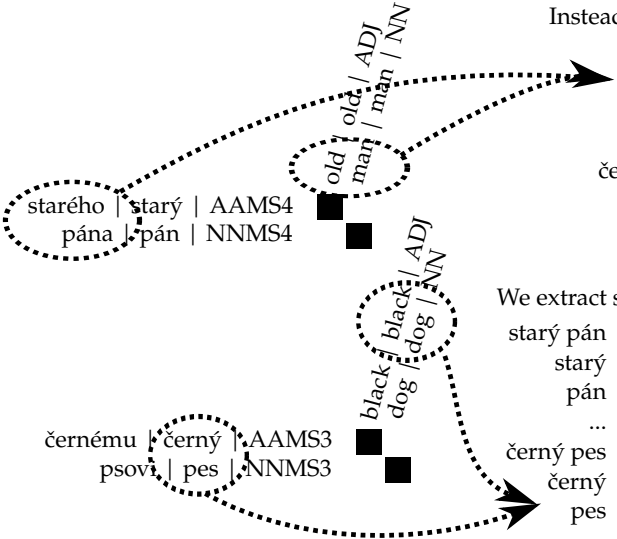
The Benefit Illustrated Once More



Instead of one phrase table based on word forms:

starého pána	=	old man
starého	=	old
pána	=	man
...		...
černému psovi	=	black dog
černému	=	black
psovi	=	dog
...		...

The Benefit Illustrated Once More



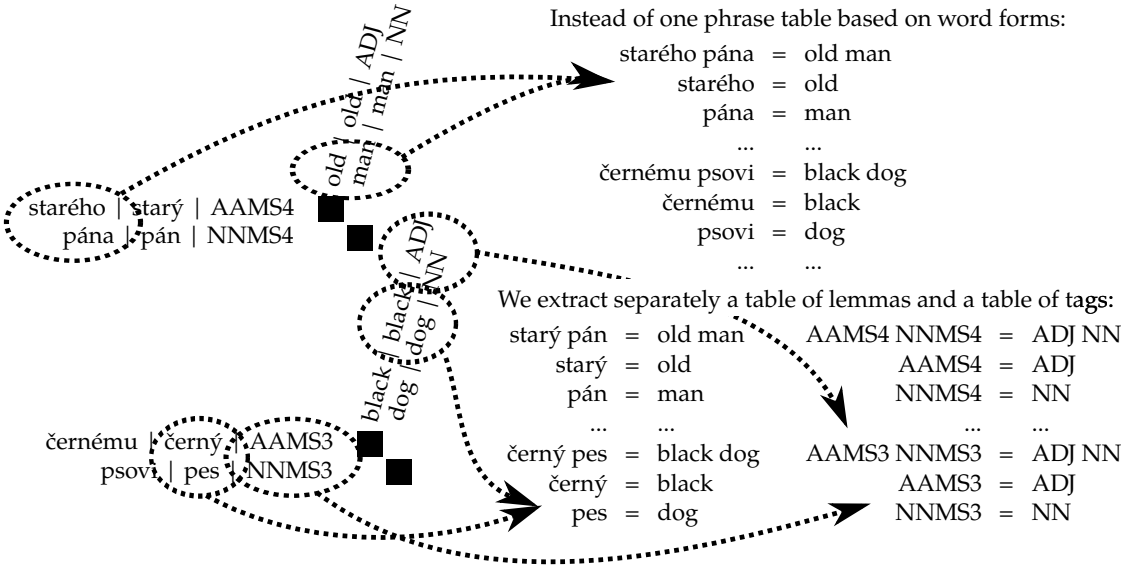
Instead of one phrase table based on word forms:

starého pána	=	old man
starého	=	old
pána	=	man
...		...
černému psovi	=	black dog
černému	=	black
psovi	=	dog
...		...

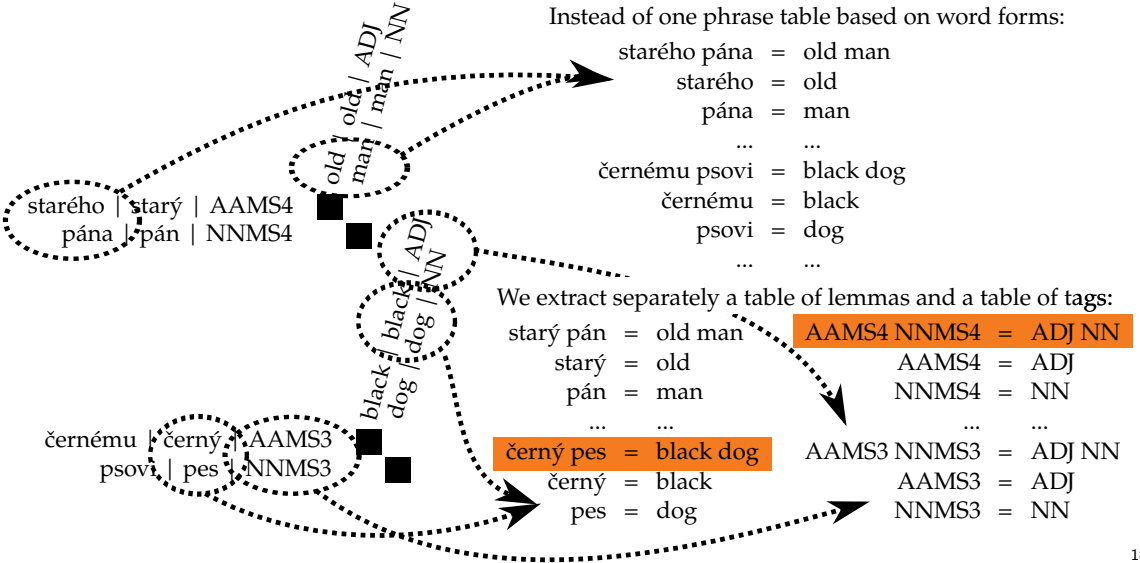
We extract separately a table of lemmas and a table of tags:

starý pán	=	old man
starý	=	old
pán	=	man
...		...
černý pes	=	black dog
černý	=	black
pes	=	dog

The Benefit Illustrated Once More



The Benefit Illustrated Once More

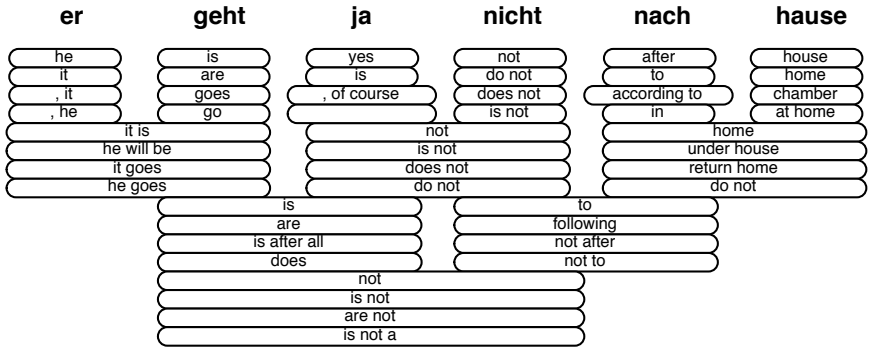


Factored Phrase-Based MT

See the following slides by Philipp Koehn (Fri Jan 30, 2009, pp. 49–75):

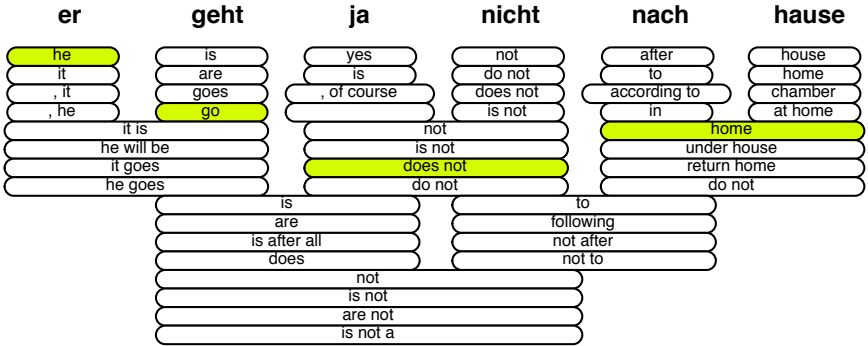
- Decoding
  - Reminder of standard phrase-based decoding
  - Factored model decoding
- Experiments
  - esp. Alternative decoding paths

Translation options



- *Many translation options* to choose from
  - in Europarl phrase table: *2727 matching phrase pairs* for this sentence
  - by pruning to the top 20 per phrase, *202 translation options* remain

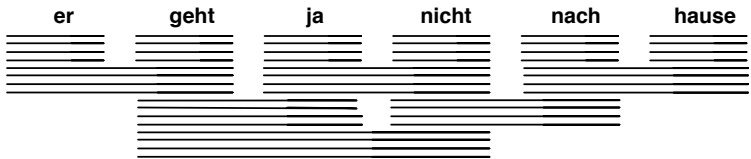
Translation options



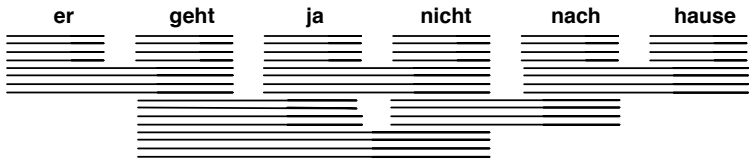
- The machine translation decoder does not know the right answer

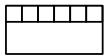
→ *Search problem* solved by heuristic beam search

Decoding process: precompute translation options

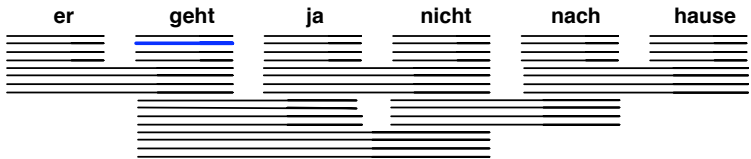


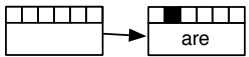
Decoding process: start with initial hypothesis



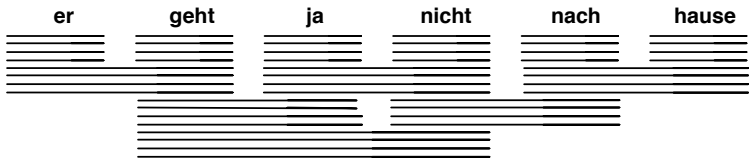


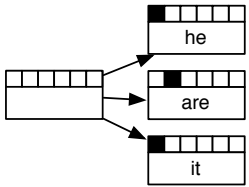
Decoding process: hypothesis expansion



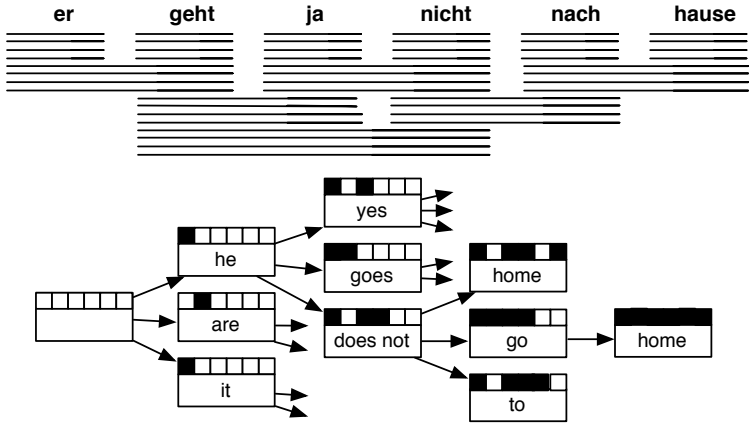


Decoding process: hypothesis expansion

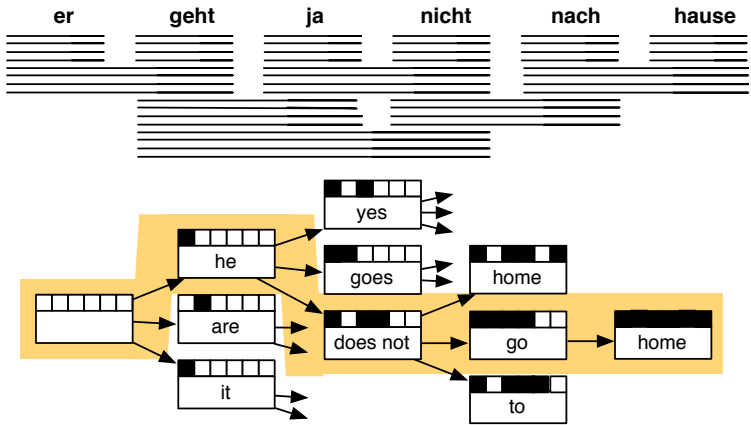




Decoding process: hypothesis expansion



Decoding process: find best path



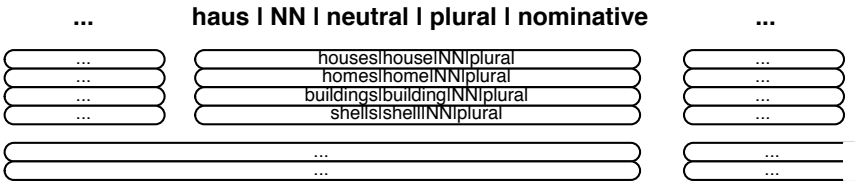
Factored model decoding

- Factored model decoding introduces *additional complexity*
- Hypothesis expansion not any more according to simple translation table, but by *executing a number of mapping steps*, e.g.:
  1. translating of *lemma* → *lemma*
  2. translating of *part-of-speech, morphology* → *part-of-speech, morphology*
  3. generation of *surface form*
- Example: *haus|NN|neutral|plural|nominative*
  → { *houses|house|NN|plural, homes|home|NN|plural,*
  buildings|building|NN|plural, shells|shell|NN|plural }
- Each time, a hypothesis is expanded, these mapping steps have to applied

Efficient factored model decoding

- Key insight: executing of mapping steps can be *pre-computed* and stored as translation options
  - apply mapping steps to all input phrases
  - store results as *translation options*

  → decoding algorithm *unchanged*



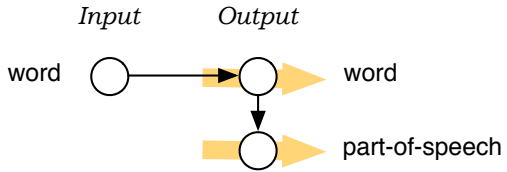
Efficient factored model decoding

- Problem: *Explosion* of translation options
  - originally limited to 20 per input phrase
  - even with simple model, now 1000s of mapping expansions possible
- Solution: *Additional pruning* of translation options
  - *keep only the best* expanded translation options
  - current default 50 per input phrase
  - decoding only about 2-3 times slower than with surface model

Factored Translation Models

- Motivation
- Example
- Model and Training
- Decoding
- **Experiments**

Adding linguistic markup to output



- Generation of POS tags on the target side
- Use of high order language models over POS (7-gram, 9-gram)
- Motivation: syntactic tags should enforce syntactic sentence structure model not strong enough to support major restructuring

Some experiments

- English–German, Europarl, 30 million word, test2006

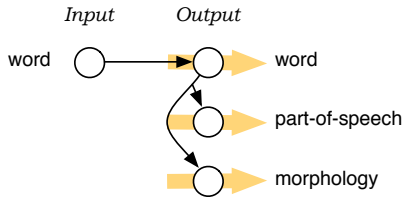
Model	BLEU
best published result	18.15
baseline (surface)	18.04
surface + POS	18.15

- German–English, News Commentary data (WMT 2007), 1 million word

Model	BLEU
Baseline	18.19
With POS LM	19.05

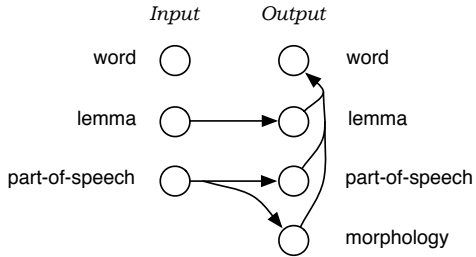
- Improvements under sparse data conditions
- Similar results with CCG supertags [Birch et al., 2007]

Local agreement (esp. within noun phrases)



- High order language models over POS and morphology
- Motivation
  - *DET-sgl NOUN-sgl* good sequence
  - *DET-sgl NOUN-plural* bad sequence

Morphological generation model



- Our motivating example
- Translating lemma and morphological information more robust

Initial results

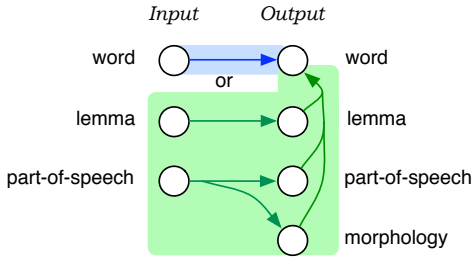
- Results on 1 million word News Commentary corpus (German–English)

System	In-doman	Out-of-domain
Baseline	18.19	15.01
With POS LM	19.05	15.03
Morphgen model	14.38	11.65

- What went wrong?
  - why back-off to lemma, when we know how to translate surface forms?

  → loss of information

Solution: alternative decoding paths



- Allow both surface form translation and morphgen model
  - prefer surface model for known words
  - morphgen model acts as back-off

Results

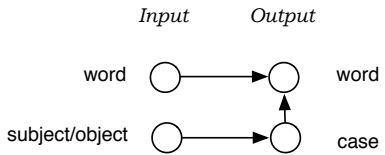
- Model now beats the baseline:

System	In-doman	Out-of-domain
Baseline	**18.19**	**15.01**
With POS LM	19.05	15.03
Morphgen model	14.38	11.65
Both model paths	**19.47**	**15.23**

Adding annotation to the source

- Source words may **lack sufficient information** to map phrases
  - English-German: what case for noun phrases?
  - Chinese-English: plural or singular
  - pronoun translation: what do they refer to?
- Idea: **add additional information** to the source that makes the required information available locally (where it is needed)
- see [Avramidis and Koehn, ACL 2008] for details

Case Information for English–Greek



- Detect in English, if noun phrase is subject/object (using parse tree)
- Map information into case morphology of Greek
- Use case morphology to generate correct word form

Obtaining Case Information

- Use syntactic parse of English input
  (method similar to semantic role labeling)

Results English-Greek

- Automatic BLEU scores

System	devtest	test07
baseline	18.13	18.05
enriched	18.21	18.20

- Improvement in verb inflection

System	Verb count	Errors	Missing
baseline	311	19.0%	7.4%
enriched	294	5.4%	2.7%

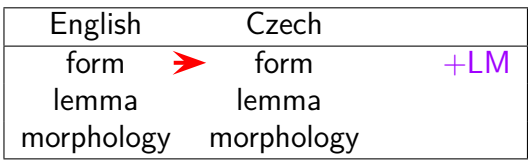
- Improvement in noun phrase inflection

System	NPs	Errors	Missing
baseline	247	8.1%	3.2%
enriched	239	5.0%	5.0%

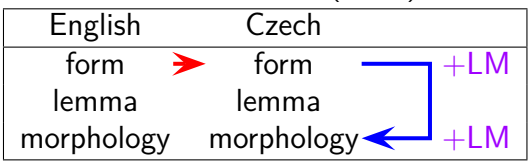
- Also successfully applied to English-Czech

Translation Scenarios for En→Cs

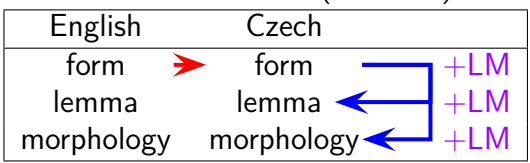
Vanilla



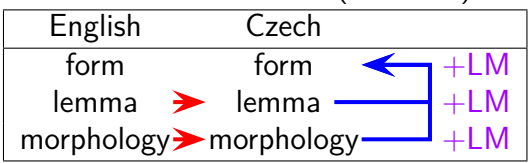
Translate+Check (T+C)



Translate+2·Check (T+C+C)



2·Translate+Generate (T+T+G)



Details on Translate+Check

- Drawback: Morphological tags increase target-side complexity:

word form → word form

green	striped
zelený	pruhovaný
zelené	**pruhované**
zelení	pruhovaní
zelených	**pruhovaných**
zeleným	pruhovaným

word form → word form morphological tag

green	striped
zelený$_{\text{sg,masc,nom}}$	pruhovaný$_{\text{sg,masc,nom}}$
zelené$_{\text{sg,fem,gen}}$	**pruhované**$_{\text{sg,fem,gen}}$
zelené$_{\text{sg,fem,dat}}$	**pruhované**$_{\text{sg,fem,dat}}$
zelené$_{\text{pl,fem,nom}}$	**pruhované**$_{\text{pl,fem,nom}}$
zelení$_{\text{pl,masc,nom}}$	pruhovaní$_{\text{pl,masc,nom}}$
zelených$_{\text{pl,masc,loc}}$	**pruhovaných**$_{\text{pl,masc,loc}}$
zeleným	pruhovaným

- Benefit: more robust LMs, e.g. trained on morphological tags only.
  - p(fem,nom masc,loc) < p(fem,nom fem,nom) … observed on all adjectives.
  - p(zelené pruhovaných) < p(zelené pruhované) … much sparser.

Factored Attempts (WMT09)

Sents	System	BLEU	NIST	Sent/min
2.2M	Vanilla PBMT	**14.24**	**5.175**	12.0
2.2M	T+C	13.86	5.110	2.6
84k	Vanilla PBMT	10.52	4.506	–
84k	T+C+C&T+T+G	10.01	4.360	4.0

- In WMT07, T+C worked best.
  + fine-tuned tags helped with small data (Bojar, 2007).
- In WMT08, T+C was worth the effort (Bojar and Hajič, 2008).
- In WMT09, our computers could handle 7-grams of forms.
  ⇒ No gain from T+C.
- T+T+G too big to fit and explodes the search space.
  ⇒ Worse than Vanilla trained on the same dataset.

T+T+G Failure Explained

- Factored models are "**synchronous**", i.e. Moses:
  1. Generates fully instantiated "translation options".
  2. Appends translation options to extend "partial hypothesis".
  3. Applies LM to see how well the option fits the previous words.
- There are too many possible combinations of lemma+tag.
  - $\Rightarrow$ Less promising ones must be pruned.
  - ! Pruned before the linear context is available.

- Hieu Hoang wasted a year on trying asynchronous factors.
  - Pruning hard to design (no clear comparison for partial translation options).
- In a completely different decoder Bojar and Týnovský (2009) use "delayed factors".
  - The final value generated only after the full hypothesis is ready.

Big / Long / Morphological LMs

- Our best setups used four LMs:

LM ID	Factor	Order	# Training Tokens
long	word form	7	685M
big	word form	4	3903M
morph	morph. tag	10	817M
longm	morph. tag	15	817M

- ... with complementary benefits:

long	big	long morph	big long	big morph	big long morph	all + longm
21.32	22.00	22.01	22.26	22.21	22.48	22.59

Tentative Summary

- Target-side rich morphology causes data sparseness.
- Factored setups compact the sparseness.
  ... but the search space is likely to explode at runtime.
- Explosion can be contained by pruning.
  ... but the pruning happens without linear context
  $\Rightarrow$ high risk of search errors.

Two promising techniques for handling sparseness and avoiding the explosion:

- Two-step translation (Bojar and Kos, 2010).
- Reverse self-training (Bojar and Tamchyna, 2011).

Two-Step Attempts (WMT10) 1/2

1. English → lemmatized Czech
   - meaning-bearing morphology preserved
   - max phrase len 10, distortion limit 6
   - large target-side (lemmatized LM)
2. Lemmatized Czech → Czech
   - trained on much more data
   - max phrase len 1, monotone

Src	after a sharp drop		
Mid	po+6	ASA1.prudký	NSA-.pokles
Gloss	*after+voc*	*adj+sg...sharp*	*noun+sg...drop*
Out	po	prudkém	poklesu

Two-Step Attempts (WMT10) 2/2

Training Sents		Vanilla		Two-Step		Diff
Parallel	Mono	BLEU	SemPOS	BLEU	SemPOS	B. S.
126k	126k	10.28±0.40	29.92	10.38±0.38	30.01	↗↗

Two-Step Attempts (WMT10) 2/2

Training Sents		Vanilla		Two-Step		Diff
Parallel	Mono	BLEU	SemPOS	BLEU	SemPOS	B. S.
126k	126k	10.28±0.40	29.92	10.38±0.38	30.01	↗ ↗
126k	13M	12.50±0.44	31.01	12.29±0.47	31.40	↘ ↗

Two-Step Attempts (WMT10) 2/2

Training Sents		Vanilla		Two-Step		Diff
Parallel	Mono	BLEU	SemPOS	BLEU	SemPOS	B. S.
126k	126k	10.28±0.40	29.92	10.38±0.38	30.01	↗ ↗
126k	13M	12.50±0.44	31.01	12.29±0.47	31.40	↘ ↗
7.5M	13M	14.17±0.51	33.07	14.06±0.49	32.57	↘ ↘

Two-Step Attempts (WMT10) 2/2

Training Sents		Vanilla		Two-Step		Diff
Parallel	Mono	BLEU	SemPOS	BLEU	SemPOS	B. S.
126k	126k	10.28±0.40	29.92	10.38±0.38	30.01	↗ ↗
126k	13M	12.50±0.44	31.01	12.29±0.47	31.40	↘ ↗
7.5M	13M	14.17±0.51	33.07	14.06±0.49	32.57	↘ ↘

Manual micro-evaluation of ↘↗, i.e. 12.50±0.44 vs. 12.29±0.47:

	Two-Step	Both Fine	Both Wrong	Vanilla	Total
Two-Step	**23**	4	8	-	**35**
Both Fine	7	14	17	5	43
Both Wrong	8	1	28	2	39
Vanilla	-	3	7	**23**	33
Total	**38**	22	60	30	150

- Each annotator weakly prefers Two-step
  - but they don't agree on individual sentences.

Reverse Self-Training

Goal: Learn from monolingual data to produce new target-side word forms in correct contexts.

	Source English		Target Czech
Small Parallel	a cat chased...	=	**kočka** honila...
	I saw a cat	=	viděl jsem **kočku**
Big Monolingual			četl jsem o **kočce**

Reverse Self-Training

Goal: Learn from monolingual data to produce new target-side word forms in correct contexts.

	Source English		Target Czech
Small Parallel	a cat chased...	=	**kočka** honila...
	I saw a cat	=	viděl jsem **kočku**
Big Monolingual	?		četl jsem o **kočce**

Reverse Self-Training

Goal: Learn from monolingual data to produce new target-side word forms in correct contexts.

	Source English		Target Czech
Small Parallel	a cat chased...	=	**kočka** honila...
	I saw a cat	=	viděl jsem **kočku**
Big Monolingual	?		četl jsem o **kočce**

Use reverse translation

Reverse Self-Training

Goal: Learn from monolingual data to produce new target-side word forms in correct contexts.

	Source English		Target Czech
Small Parallel	a cat chased...	=	**kočka** honila... *kočka honit... (lem.)*
	I saw a cat	=	viděl jsem **kočku** *vidět být kočka (lem.)*
Big Monolingual	?		četl jsem o **kočce** *číst být o kočka (lem.)* Use reverse translation backed-off by lemmas.

Reverse Self-Training

Goal: Learn from monolingual data to produce new target-side word forms in correct contexts.

	Source English		Target Czech
Small Parallel	a cat chased...	=	**kočka** honila... *kočka honit... (lem.)*
	I saw a cat	=	viděl jsem **kočku** *vidět být kočka (lem.)*
Big Monolingual	?		četl jsem o **kočce** *číst být o kočka (lem.)*
	I read about a cat	←	Use reverse translation backed-off by lemmas.

Reverse Self-Training

Goal: Learn from monolingual data to produce new target-side word forms in correct contexts.

	Source English		Target Czech
Small Parallel	a cat chased...	=	**kočka** honila... *kočka honit... (lem.)*
	I saw a cat	=	viděl jsem **kočku** *vidět být kočka (lem.)*
Big Monolingual	?		četl jsem o **kočce** *číst být o kočka (lem.)*
	I read about a cat	←	Use reverse translation backed-off by lemmas.

⇒ A new phrase learned: "about a cat" = "o **kočce**".

The Back-off to Lemmas

- The key distinction from self-training used for domain adaptation (Bertoldi and Federico, 2009; Ueffing et al., 2007).
- We use simply "alternative decoding paths" in Moses:

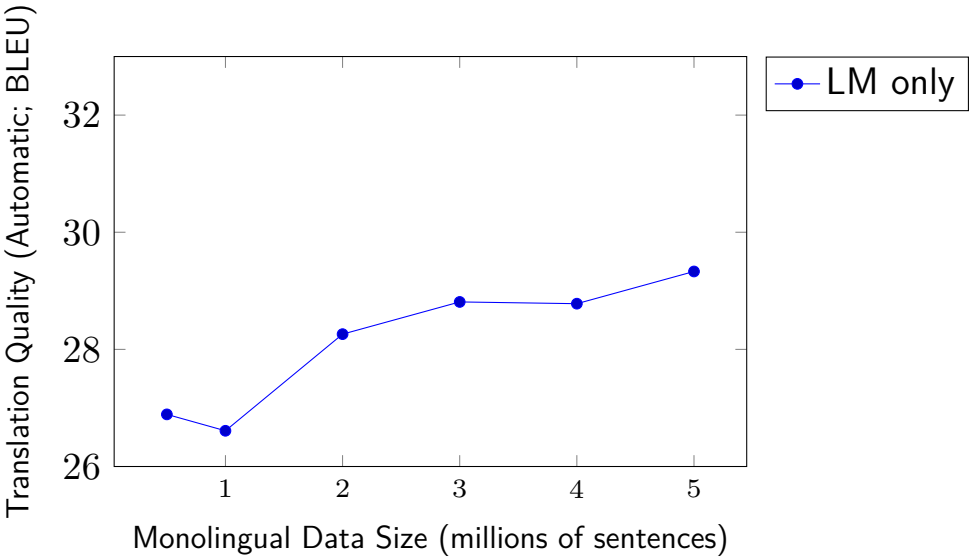
Czech		English	
form	→	form	+LM

or

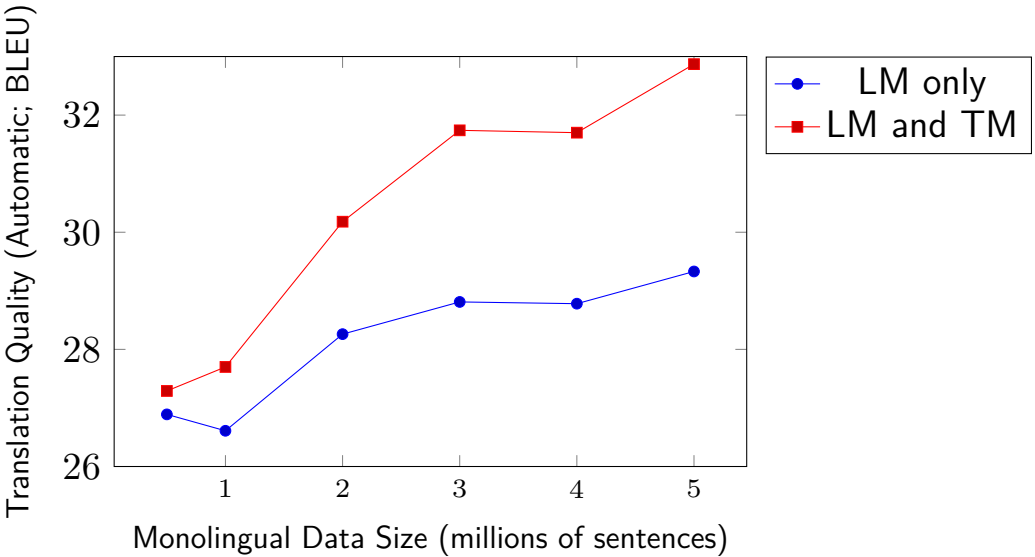
Czech		English	
lemma	→	form	+LM

- Other languages (e.g. Turkish, German) need different back-off techniques:
  - Split German compounds.
  - Separate and allow to ignore Turkish morphology.

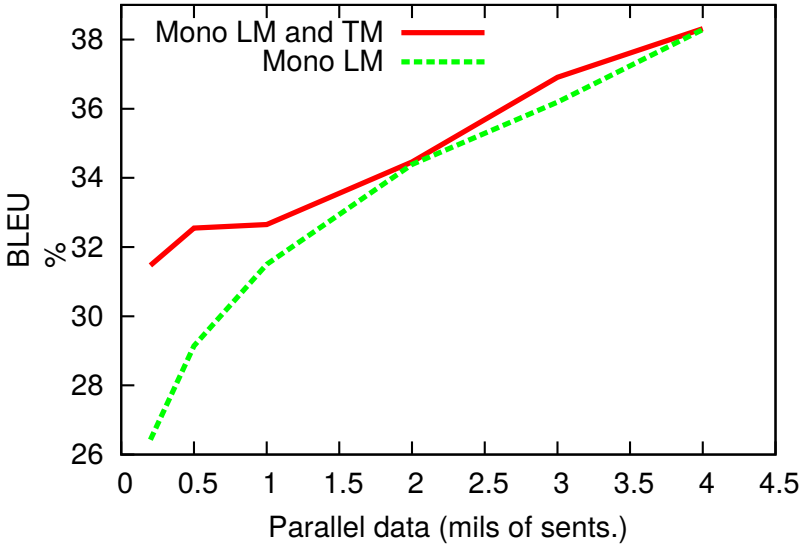
Small Parallel, Increasing Monolingual



Small Parallel, Increasing Monolingual



Increasing Para, Fixed Mono



Morphology in Neural MT

NMT: “Solved” by Segmentation

- SMT struggled with productive morphology (>1M wordforms).

  nejneobhodpodařovávatelnějšími, Donaudampfschifffahrtsgesellschaftskapitän

- NMT can handle only 30–80k dictionaries.

⇒ Resort to sub-word units.

Orig	český politik svezl migranty
Syllables	čes ký ␣ po li tik ␣ sve zl ␣ mig ran ty
Morphemes	česk ý ␣ politik ␣ s vez l ␣ migrant y
Char Pairs	če sk ý ␣ po li ti k ␣ sv ez l ␣ mi gr an ty
Chars	č e s k ý ␣ p o l i t i k ␣ s v e z l ␣ m i g r a n t y
BPE 30k	český politik s@@ vez@@ l mi@@ granty

Byte Pair Encoding (Sennrich et al., 2016)

- Given a dictionary of token types and frequences.
  1. Replace the most frequent pair of characters with a new unit.
     (Record this "merge" operation.)
  2. Repeat until the desired number of merge operations is reached.

Byte Pair Encoding (Sennrich et al., 2016)

- Given a dictionary of token types and frequences.
  1. Replace the most frequent pair of characters with a new unit.
     (Record this "merge" operation.)
  2. Repeat until the desired number of merge operations is reached.

Current vocabulary	The new merge
lower lowest newer widest	

Byte Pair Encoding (Sennrich et al., 2016)

- Given a dictionary of token types and frequences.
  1. Replace the most frequent pair of characters with a new unit. (Record this "merge" operation.)
  2. Repeat until the desired number of merge operations is reached.

Current vocabulary	The new merge
lower lowest newer widest	we → we

Byte Pair Encoding (Sennrich et al., 2016)

- Given a dictionary of token types and frequences.
  1. Replace the most frequent pair of characters with a new unit.
     (Record this "merge" operation.)
  2. Repeat until the desired number of merge operations is reached.

Current vocabulary	The new merge
lower lowest newer widest	we → we
lo we r lo we st ne we r widest	

Byte Pair Encoding (Sennrich et al., 2016)

- Given a dictionary of token types and frequences.
  1. Replace the most frequent pair of characters with a new unit.
     (Record this "merge" operation.)
  2. Repeat until the desired number of merge operations is reached.

Current vocabulary	The new merge
lower lowest newer widest	we → we
lower lowest newer widest	wer → wer

Byte Pair Encoding (Sennrich et al., 2016)

- Given a dictionary of token types and frequences.
  1. Replace the most frequent pair of characters with a new unit. (Record this "merge" operation.)
  2. Repeat until the desired number of merge operations is reached.

Current vocabulary	The new merge
lower lowest newer widest	we → we
lower lowest newer widest	wer → wer
lower lowest newer widest	lwer → lwer

Byte Pair Encoding (Sennrich et al., 2016)

- Given a dictionary of token types and frequences.
  1. Replace the most frequent pair of characters with a new unit. (Record this "merge" operation.)
  2. Repeat until the desired number of merge operations is reached.

Current vocabulary	The new merge
lower lowest newer widest	we → we
lo we r lo we st ne we r widest	we r → we r
lo we r lo we st ne we r widest	st → st

Byte Pair Encoding (Sennrich et al., 2016)

- Given a dictionary of token types and frequences.
  1. Replace the most frequent pair of characters with a new unit.
     (Record this "merge" operation.)
  2. Repeat until the desired number of merge operations is reached.

Current vocabulary	The new merge
lower lowest newer widest	we → we
lo we r lo we st ne we r widest	we r → we r
lo we r lo we st ne we r wide st	st → st

- New input: Apply the recorded sequence of merges:

  newest → ne we st → ne we st ⇒ n@@ e@@ we@@ st

- Ensures that vocabulary size = alphabet + merge ops.

Flavours of Subword Units

- Byte Pair Encoding (BPE, Sennrich et al. (2016))
  `http://github.com/rsennrich/subword-nmt/`
- Google Wordpieces (Wu et al., 2016)
  Code probably unavailable, used in speech.
- SubwordTextEncoder in Tensor2tensor (Vaswani et al., 2017)
  `https://github.com/tensorflow/tensor2tensor`

STE	Blíží_ se_ k_ tobě_ **tramvaj** _ ._ Z_ **tramvaj** e_ nevysto upil i_ ._
BPE	Blíží se k tobě **tramvaj** . Z **tramva@@ je** nevy@@ stoupili .
BPE underscore	Blíží_ se_ k_ tobě_ **tramvaj@@ _** ._ Z_ **tramvaj@@ e_** nevy@@ stoupili_ ._

The best now is SentencePiece: `https://github.com/google/sentencepiece`

Performance of STE and BPE

- German→Czech T2T experiments (Macháček et al., 2018).
- The underscore trick:
  - Append "_" to tokens before learning splits.

split	underscore	BLEU
STE	after every token	18.58±0.06
BPE	after non-final tokens	18.24±0.08
BPE	after every token	13.88±0.18
BPE	-	13.69±0.66

- +5(!) BLEU points from the underscore trick.
  - If not attached at the end of the sentence.

Room for Linguistics

- Ataman et al. (2017) use a new Morfessor model Flatcat (Grönroos et al., 2014) for Turkish.
  - Considerably better than BPE.
- Huck et al. (2017a) examine English→German:
  - Compound, suffix, prefix and BPE splitting, or a cascade.
  - Suffix+BPE or Compound+suffix+BPE best.
- Macháček et al. (2018) for German→Czech:
  - Unsupervised (Morfessor) and supervised (DeriNet).
  - STE worked best.

Summary

- Rich morphology causes serious problems to token-based MT.
- Factors in PBMT allow to capture additional info.
- Rich annotation is dangerous when not treated carefully.
  Occam's razor: think twice before adding an attribute.
  - Avoid data sparseness, always provide a back-off.
  - Avoid complex models:
    - They are hard to tune (set parameters).
    - They tend to explode at runtime.
- Promising 2-step translation.
- Reverse self-training good for small data.

- NMT with subword units resolves problems with morphology.
- Still room for linguistically-adequate solutions.

References

Duygu Ataman, Matteo Negri, Marco Turchi, and Marcello Federico. 2017. Linguistically motivated vocabulary reduction for neural machine translation from turkish to english. The Prague Bulletin of Mathematical Linguistics, 108:331, Jan.

Nicola Bertoldi and Marcello Federico. 2009. Domain adaptation for statistical machine translation with monolingual resources. In Proceedings of the Fourth Workshop on Statistical Machine Translation, pages 182–189, Athens, Greece, March. Association for Computational Linguistics.

Ondřej Bojar and Jan Hajič. 2008. Phrase-Based and Deep Syntactic English-to-Czech Statistical Machine Translation. In Proceedings of the Third Workshop on Statistical Machine Translation, pages 143–146, Columbus, Ohio, June. Association for Computational Linguistics.

Ondřej Bojar and Kamil Kos. 2010. 2010 Failures in English-Czech Phrase-Based MT. In Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR, pages 60–66, Uppsala, Sweden, July. Association for Computational Linguistics.

Ondřej Bojar and Aleš Tamchyna. 2011. Improving Translation Model by Monolingual Data. In Proceedings of the Sixth Workshop on Statistical Machine Translation, pages 330–336, Edinburgh, Scotland, July. Association for Computational Linguistics.

Ondřej Bojar and Miroslav Týnovský. 2009. Evaluation of Tree Transfer System. Project Euromatrix - Deliverable 3.4, ÚFAL, Charles University, March.

Ondřej Bojar. 2007. English-to-Czech Factored Machine Translation. In Proceedings of the Second Workshop on Statistical Machine Translation, pages 232–239, Prague, Czech Republic, June. Association for Computational Linguistics.

Stig-Arne Grönroos, Sami Virpioja, Peter Smit, and Mikko Kurimo. 2014. Morfessor FlatCat: An HMM-Based Method for Unsupervised and Semi-Supervised Learning of Morphology. In Proceedings of COLING 2014, the 25th International