Winter School

Day 5: Discriminative Training and Factored Translation Models

MT Marathon

30 January 2009

The birth of SMT: generative models

- The definition of translation probability follows a **mathematical derivation**

$$\text{argmax}_{\mathbf{e}} p(\mathbf{e}|\mathbf{f}) = \text{argmax}_{\mathbf{e}} p(\mathbf{f}|\mathbf{e})\; p(\mathbf{e})$$

- Occasionally, some **independence assumptions** are thrown in
  for instance IBM Model 1: word translations are independent of each other

$$p(\mathbf{e}|\mathbf{f}, a) = \frac{1}{Z} \prod_i p(e_i | f_{a(i)})$$

- Generative story leads to **straight-forward estimation**
  - maximum likelihood estimation of component probability distribution
  - **EM algorithm** for discovering hidden variables (alignment)

Log-linear models

- IBM Models provided mathematical justification for factoring **components** together

$$p_{LM} \times p_{TM} \times p_D$$

- These may be **weighted**

$$p_{LM}^{\lambda_{LM}} \times p_{TM}^{\lambda_{TM}} \times p_D^{\lambda_D}$$

- **Many components** p_i with weights λ_i

$$\prod_i p_i^{\lambda_i} = exp(\sum_i \lambda_i log(p_i))$$

$$log \prod_i p_i^{\lambda_i} = \sum_i \lambda_i log(p_i)$$

Knowledge sources

- Many different **knowledge sources** useful
  - language model
  - reordering (distortion) model
  - phrase translation model
  - word translation model
  - word count
  - phrase count
  - drop word feature
  - phrase pair frequency
  - additional language models
  - additional features

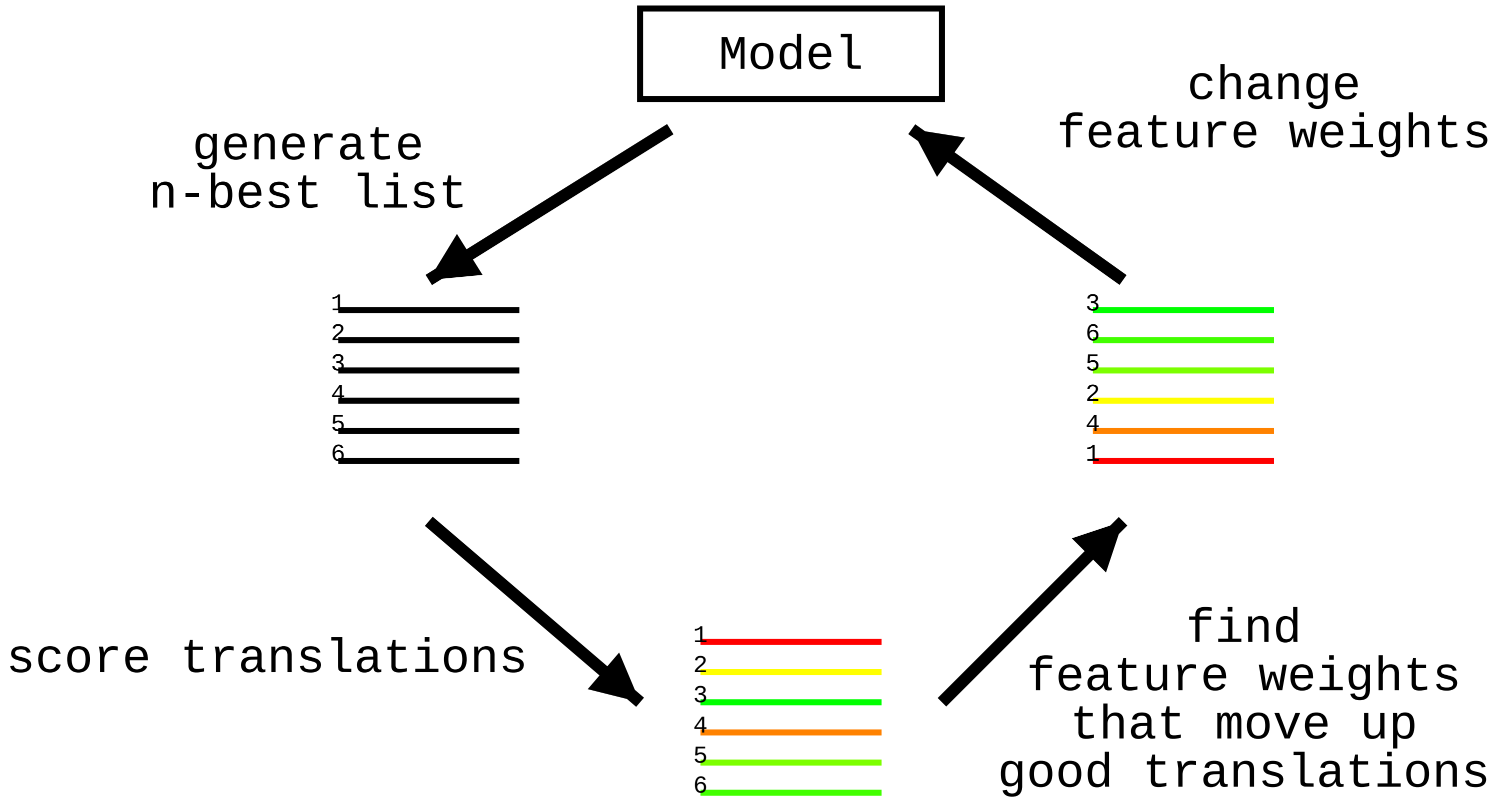
Set feature weights

- Contribution of components p_i determined by weight λ_i
- Methods
  - *manual setting* of weights: try a few, take best
  - *automate* this process
- Learn weights
  - set aside a **development corpus**
  - set the weights, so that **optimal translation performance** on this development corpus is achieved
  - requires *automatic scoring* method (e.g., BLEU)

Discriminative training

- Training set (*development set*)
  - different from original training set
  - small (maybe 1000 sentences)
  - must be different from test set
- Current model *translates* this development set
  - *n-best list* of translations (n=100, 10000)
  - translations in n-best list can be *scored*
- Feature weights are *adjusted*
- N-Best list generation and feature weight adjustment repeated for a number of iterations

Discriminative training



Discriminative vs. generative models

- Generative models
  - translation process is broken down to *steps*
  - each step is modeled by a *probability distribution*
  - each probability distribution is estimated from the data by *maximum likelihood*
- Discriminative models
  - model consist of a number of *features* (e.g. the language model score)
  - each feature has a *weight*, measuring its value for judging a translation as correct
  - feature weights are *optimized on development data*, so that the system output matches correct translations as close as possible

Learning task

- Task: *find weights*, so that feature vector of best translations *ranked first*
- Input: *Er geht ja nicht nach Hause*, Ref: *He does not go home*

Translation	Feature values						Error
it is not under house	-32.22	-9.93	-19.00	-5.08	-8.22	-5	0.8
he is not under house	-34.50	-7.40	-16.33	-5.01	-8.15	-5	0.6
it is not a home	-28.49	-12.74	-19.29	-3.74	-8.42	-5	0.6
it is not to go home	-32.53	-10.34	-20.87	-4.38	-13.11	-6	0.8
it is not for house	-31.75	-17.25	-20.43	-4.90	-6.90	-5	0.8
he is not to go home	-35.79	-10.95	-18.20	-4.85	-13.04	-6	0.6
he does not home	-32.64	-11.84	-16.98	-3.67	-8.76	-4	**0.2**
it is not packing	-32.26	-10.63	-17.65	-5.08	-9.89	-4	0.8
he is not packing	-34.55	-8.10	-14.98	-5.01	-9.82	-4	0.6
he is not for home	-36.70	-13.52	-17.09	-6.22	-7.82	-5	0.4

Och's minimum error rate training (MERT)

- **Line search** for best feature weights

```
given:  sentences with n-best list of
translations
iterate n times
    randomize starting feature weights
        iterate until convergences
            for each feature
                find best feature weight
                update if different from current
return best feature weights found in any
iteration
```

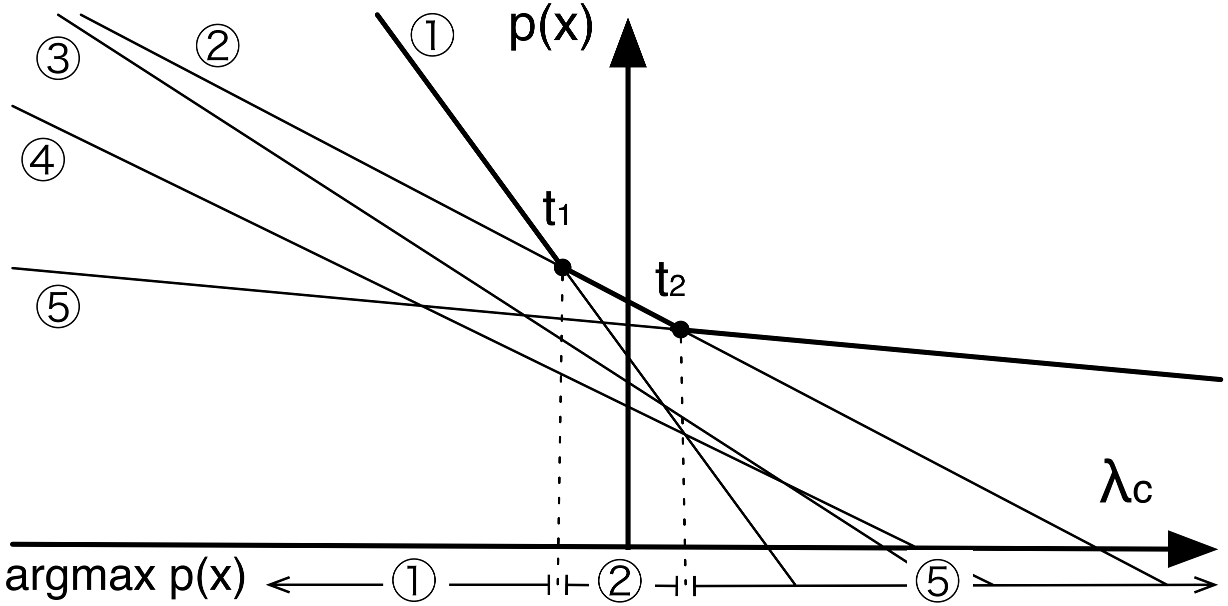
Find Best Feature Weight

- Core task:
  - find optimal value for one parameter weight λ
  - ... while leaving all other weights constant
- Score of translation i for a sentence **f**:

$$p(\mathbf{e}_i|\mathbf{f}) = \lambda a_i + b_i$$

- Recall that:
  - we deal with 100s of translations $\mathbf{e}_i$ per sentence **f**
  - we deal with 100s or 1000s of sentences **f**
  - we are trying to find the value λ so that over all sentences, the error score is optimized

Translations for one Sentence



- each translation is a line $p(\mathbf{e}_i|\mathbf{f}) = \lambda a_i + b_i$
- the model-best translation for a given λ (x-axis), is highest line at that point
- there are one a few *threshold points* t_j where the model-best line changes

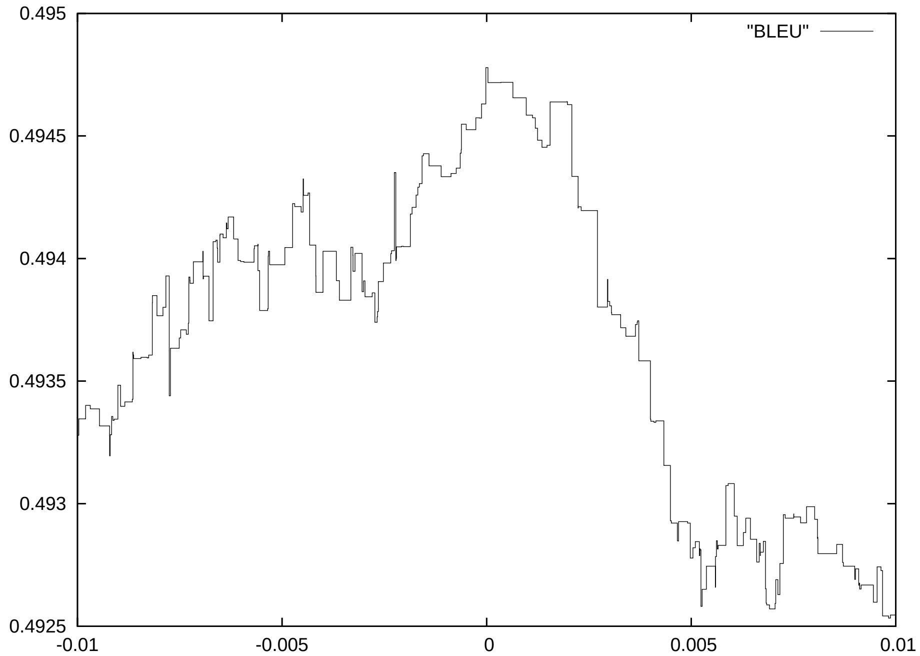
Finding the Optimal Value for λ

- Real-valued λ can have infinite number of values
- But only on threshold points, one of the model-best translation changes

$\Rightarrow$ Algorithm:
  - find the threshold points
  - for each interval between threshold points
    * find best translations
    * compute error-score
  - pick interval with best error-score

BLEU error surface

- Varying one parameter: a rugged line with many local optima



Unstable outcomes: weights vary

component	run 1	run 2	run 3	run 4	run 5	run 6
distance	0.059531	0.071025	0.069061	0.120828	0.120828	0.072891
lexdist 1	0.093565	0.044724	0.097312	0.108922	0.108922	0.062848
lexdist 2	0.021165	0.008882	0.008607	0.013950	0.013950	0.030890
lexdist 3	0.083298	0.049741	0.024822	-0.000598	-0.000598	0.023018
lexdist 4	0.051842	0.108107	0.090298	0.111243	0.111243	0.047508
lexdist 5	0.043290	0.047801	0.020211	0.028672	0.028672	0.050748
lexdist 6	0.083848	0.056161	0.103767	0.032869	0.032869	0.050240
lm 1	0.042750	0.056124	0.052090	0.049561	0.049561	0.059518
lm 2	0.019881	0.012075	0.022896	0.035769	0.035769	0.026414
lm 3	0.059497	0.054580	0.044363	0.048321	0.048321	0.056282
ttable 1	0.052111	0.045096	0.046655	0.054519	0.054519	0.046538
ttable 1	0.052888	0.036831	0.040820	0.058003	0.058003	0.066308
ttable 1	0.042151	0.066256	0.043265	0.047271	0.047271	0.052853
ttable 1	0.034067	0.031048	0.050794	0.037589	0.037589	0.031939
phrase-pen.	0.059151	0.062019	-0.037950	0.023414	0.023414	-0.069425
word-pen	-0.200963	-0.249531	-0.247089	-0.228469	-0.228469	-0.252579

Unstable outcomes: scores vary

- Even different scores with different runs (varying 0.40 on dev, 0.89 on test)

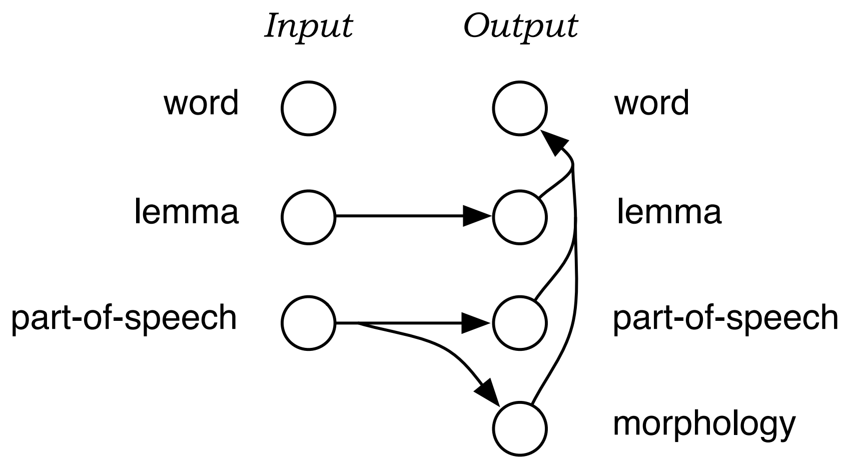
run	iterations	dev score	test score
1	8	50.16	51.99
2	9	50.26	51.78
3	8	50.13	51.59
4	12	50.10	51.20
5	10	50.16	51.43
6	11	50.02	51.66
7	10	50.25	51.10
8	11	50.21	51.32
9	10	50.42	51.79

More features: more components

- We would like to add **more components** to our model
  - multiple language models
  - domain adaptation features
  - various special handling features
  - using linguistic information

→ MERT becomes even **less reliable**
  - runs many more iterations
  - fails more frequently

More features: factored models



- Factored translation models break up phrase mapping into smaller steps
  - multiple translation tables
  - multiple generation tables
  - multiple language models and sequence models on factors

→ **Many more features**

Millions of features

- Why **mix** of discriminative training and generative models?
- Discriminative training of all components
  - phrase table [Liang et al., 2006]
  - language model [Roark et al, 2004]
  - additional features
- **Large-scale** discriminative training
  - millions of features
  - training of full training set, not just a small development corpus

Perceptron algorithm

- Translate each sentence
- If no match with reference translation: update features

```
set all lambda = 0
do until convergence
    for all foreign sentences f
        set e-best to best translation according to model
        set e-ref to reference translation
        if e-best != e-ref
            for all features feature-i
                lambda-i += feature-i(f,e-ref)
                          - feature-i(f,e-best)
```

Problem: overfitting

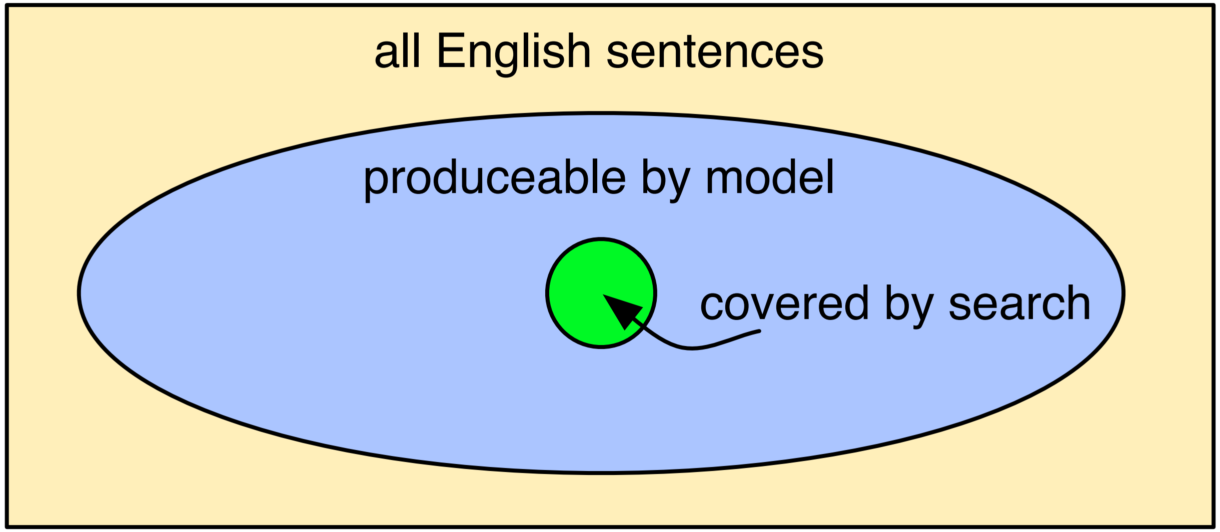
- Fundamental problem in machine learning
  - what works best for training data, may not work well in general
  - **rare, unrepresentative features** may get too much weight
- **Especially severe problem** in phrase-based models
  - **long phrase pairs** explain well *individual sentences*
  - ... but are less general, *suspect to noise*
  - EM training of phrase models [Marcu and Wong, 2002] has same problem

Solutions

- **Restrict to short phrases**, e.g., maximum 3 words (current approach)
  - limits the power of phrase-based models
  - ... but not very much [Koehn et al, 2003]
- **Jackknife**
  - collect phrase pairs from one part of corpus
  - optimize their feature weights on another part
- IBM direct model: **only one-to-many** phrases [Ittycheriah and Salim Roukos, 2007]

Problem: reference translation

- Reference translation may be anywhere in this box



- If produceable by model → we can compute feature scores
- If not → we can not

Some solutions

- **Skip sentences**, for which reference can not be produced
  - invalidates large amounts of training data
  - biases model to shorter sentences
- Declare candidate translations closest to reference as **surrogate**
  - closeness measured for instance by smoothed BLEU score
  - may be not a very good translation: odd feature values, training is severely distorted

Experiment

- Skipping sentences with unproduceable reference **hurts**

Handling of reference	BLEU
with skipping	25.81
w/o skipping	29.61

- When including all sentences: surrogate reference picked from 1000-best list using maximum *smoothed BLEU score* with respect to reference translation
- Czech-English task, **only binary features**
  - phrase table features
  - lexicalized reordering features
  - source and target phrase bigram
- See also [Liang et al., 2006] for similar approach

Better solution: early updating?

- At some point the reference translation **falls out** of the search space
  - for instance, due to *unknown words*:

Reference:	`The group attended the meeting in Najaf ...`
System:	`The group meeting was attended in UNKNOWN ...`

only update features involved in this part

- Early updating [Collins et al., 2005]:
  - stop search, when reference translation is not covered by model
  - only update **features involved in partial** reference / system output

Conclusions

- Currently have proof-of-concept implementation
- Future work: Overcome various technical challenges
  - reference translation may not be produceable
  - overfitting
  - mix of binary and real-valued features
  - scaling up
- More and more features are unavoidable, let's deal with them

Factored Translation Models

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

Statistical machine translation today

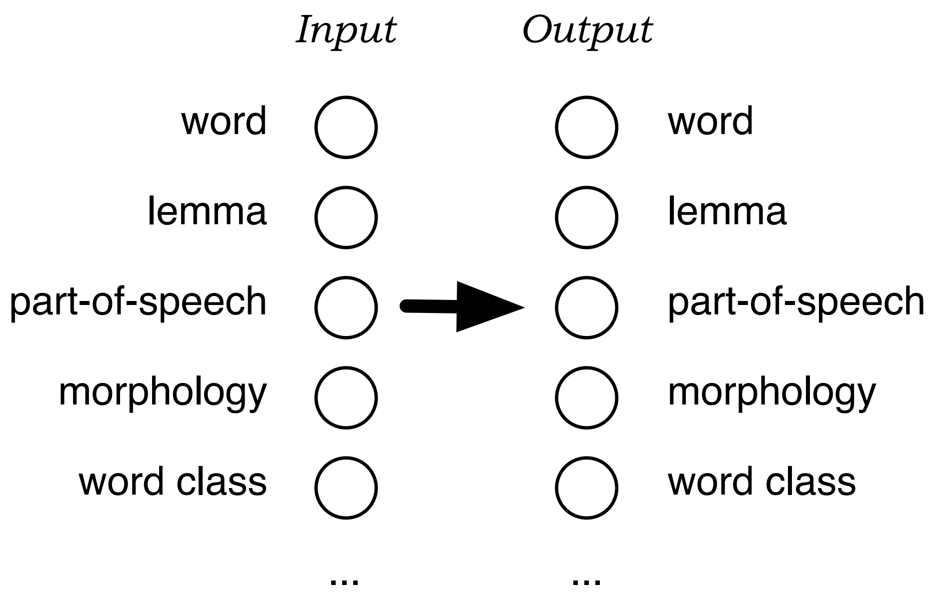
- Best performing methods based on **phrases**
  - short sequences of words
  - no use of explicit syntactic information
  - no use of morphological information
  - currently best performing method
- Progress in **syntax-based** translation
  - tree transfer models using syntactic annotation
  - still shallow representation of words and non-terminals
  - active research, improving performance

One motivation: morphology

- Models treat *car* and *cars* as completely different words
  - training occurrences of *car* have no effect on learning translation of *cars*
  - if we only see *car*, we do not know how to translate *cars*
  - rich morphology (German, Arabic, Finnish, Czech, ...) → many word forms
- Better approach
  - analyze surface word forms into **lemma** and **morphology**, e.g.: *car +plural*
  - translate lemma and morphology separately
  - generate target surface form

Factored translation models

- **Factored represention** of words



- Goals
  - **Generalization**, e.g. by translating lemmas, not surface forms
  - **Richer model**, e.g. using syntax for reordering, language modeling)

Related work

- **Back off** to representations with richer statistics (lemma, etc.)
  [Nießen and Ney, 2001, Yang and Kirchhoff 2006, Talbot and Osborne 2006]
- Use of additional annotation in **pre-processing** (POS, syntax trees, etc.)
  [Collins et al., 2005, Crego et al, 2006]
- Use of additional annotation in **re-ranking** (morphological features, POS, syntax trees, etc.)
  [Och et al. 2004, Koehn and Knight, 2005]

→ we pursue an *integrated approach*

- Use of syntactic **tree structure**
  [Wu 1997, Alshawi et al. 1998, Yamada and Knight 2001, Melamed 2004, Menezes and Quirk 2005, Chiang 2005, Galley et al. 2006]

→ may be *combined* with our approach

Factored Translation Models

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

Decomposing translation: example

- **Translate** lemma and syntactic information **separately**

$$\begin{bmatrix} \text{lemma} \end{bmatrix} \Rightarrow \begin{bmatrix} \text{lemma} \end{bmatrix}$$

$$\begin{bmatrix} \text{part-of-speech} \\ \text{morphology} \end{bmatrix} \Rightarrow \begin{bmatrix} \text{part-of-speech} \\ \text{morphology} \end{bmatrix}$$

Decomposing translation: example

- **Generate surface** form on target side

$$\begin{bmatrix} \text{surface} \end{bmatrix}$$

$$\Uparrow$$

$$\begin{bmatrix} \text{lemma} \\ \text{part-of-speech} \\ \text{morphology} \end{bmatrix}$$

Translation process: example

Input: *(Autos, Auto, NNS)*

1. Translation step: lemma ⇒ lemma
   (?, car, ?), (?, auto, ?)
2. Generation step: lemma ⇒ part-of-speech
   (?, car, NN), (?, car, NNS), (?, auto, NN), (?, auto, NNS)
3. Translation step: part-of-speech ⇒ part-of-speech
   (?, car, NN), (?, car, NNS), (?, auto, NNP), (?, auto, NNS)
4. Generation step: lemma,part-of-speech ⇒ surface
   (car, car, NN), (cars, car, NNS), (auto, auto, NN), (autos, auto, NNS)

Factored Translation Models

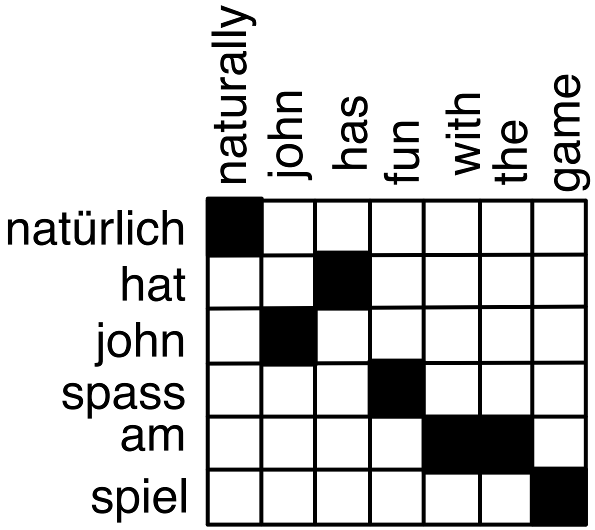
- Motivation
- Example
- **Model and Training**
- Decoding
- Experiments

Model

- Extension of *phrase model*
- Mapping of foreign words into English words broken up into steps
  - **translation step**: maps foreign factors into English factors (on the phrasal level)
  - **generation step**: maps English factors into English factors (for each word)
- Each step is modeled by one or more *feature functions*
  - fits nicely into log-linear model
  - weight set by discriminative training method
- Order of mapping steps is chosen to optimize search

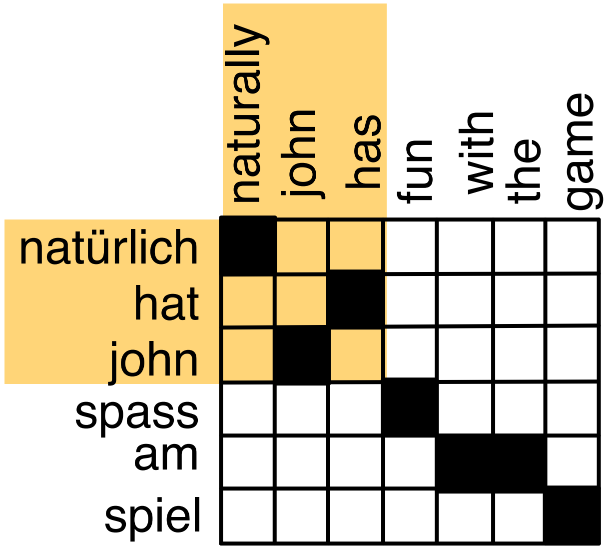
Phrase-based training

- Establish word alignment (GIZA++ and symmetrization)



Phrase-based training

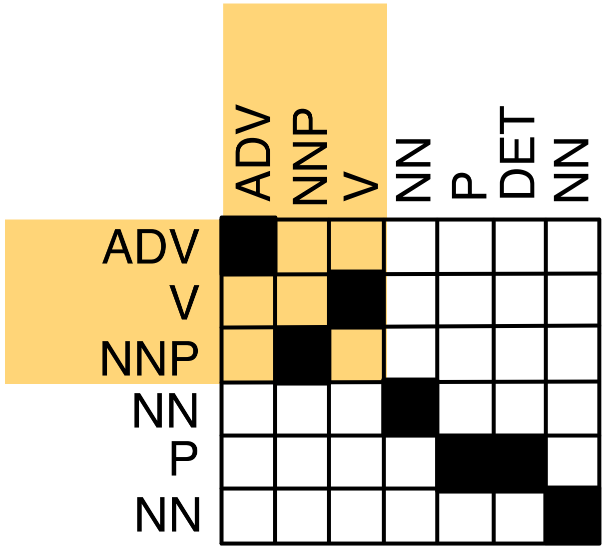
- Extract phrase



⇒ *natürlich hat john* — *naturally john has*

Factored training

- Annotate training with factors, extract phrase



$\Rightarrow$ *ADV V NNP — ADV NNP V*

Training of generation steps

- Generation steps map target factors to target factors
  - typically trained on target side of parallel corpus
  - may be trained on additional monolingual data
- Example: *The*/DET *man*/NN *sleeps*/VBZ
  - count collection
    - count(*the*,DET)++
    - count(*man*,NN)++
    - count(*sleeps*,VBZ)++
  - evidence for probability distributions (max. likelihood estimation)
    - p(DET|*the*), p(*the*|DET)
    - p(NN|*man*), p(*man*|NN)
    - p(VBZ|*sleeps*), p(*sleeps*|VBZ)

Factored Translation Models

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

Phrase-based translation

- Task: *translate this sentence* from German into English

er	**geht**	**ja**	**nicht**	**nach**	**hause**

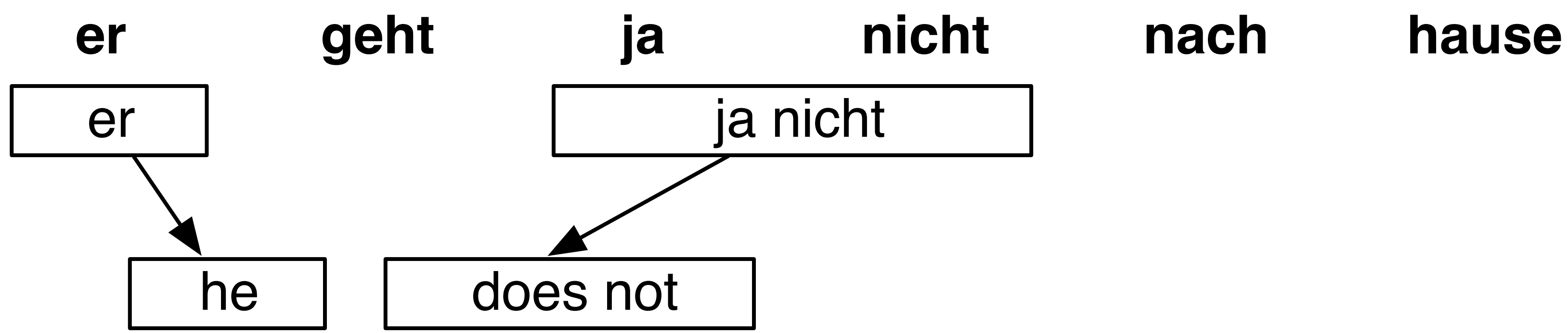
Translation step 1

- Task: translate this sentence from German into English

- *Pick* phrase in input, *translate*

Translation step 2

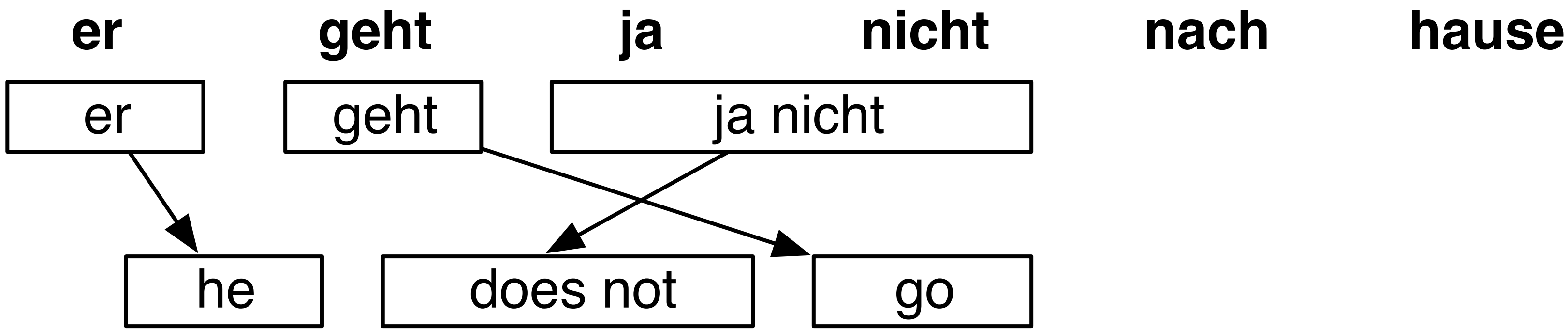
- Task: translate this sentence from German into English



- Pick phrase in input, translate
  - it is allowed to pick words *out of sequence* (**reordering**)
  - phrases may have multiple words: *many-to-many* translation

Translation step 3

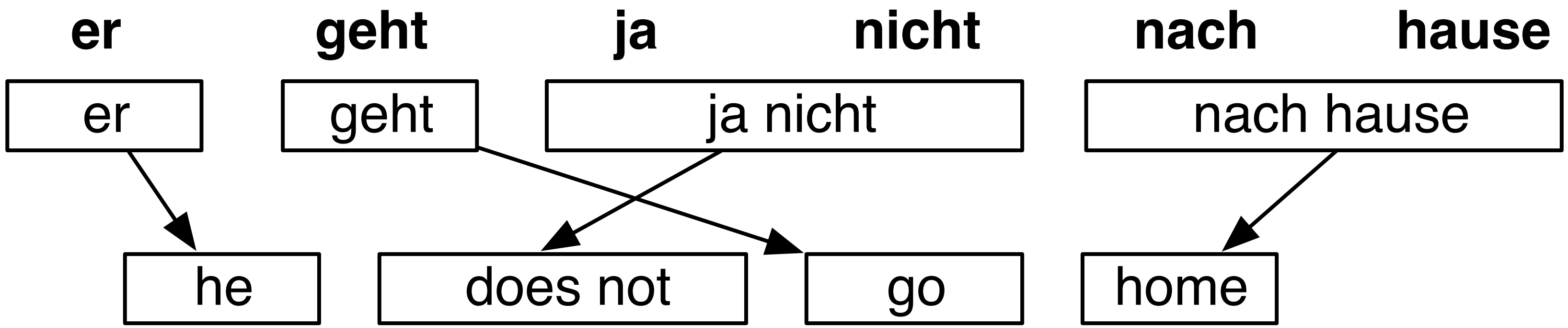
- Task: translate this sentence from German into English



- Pick phrase in input, translate

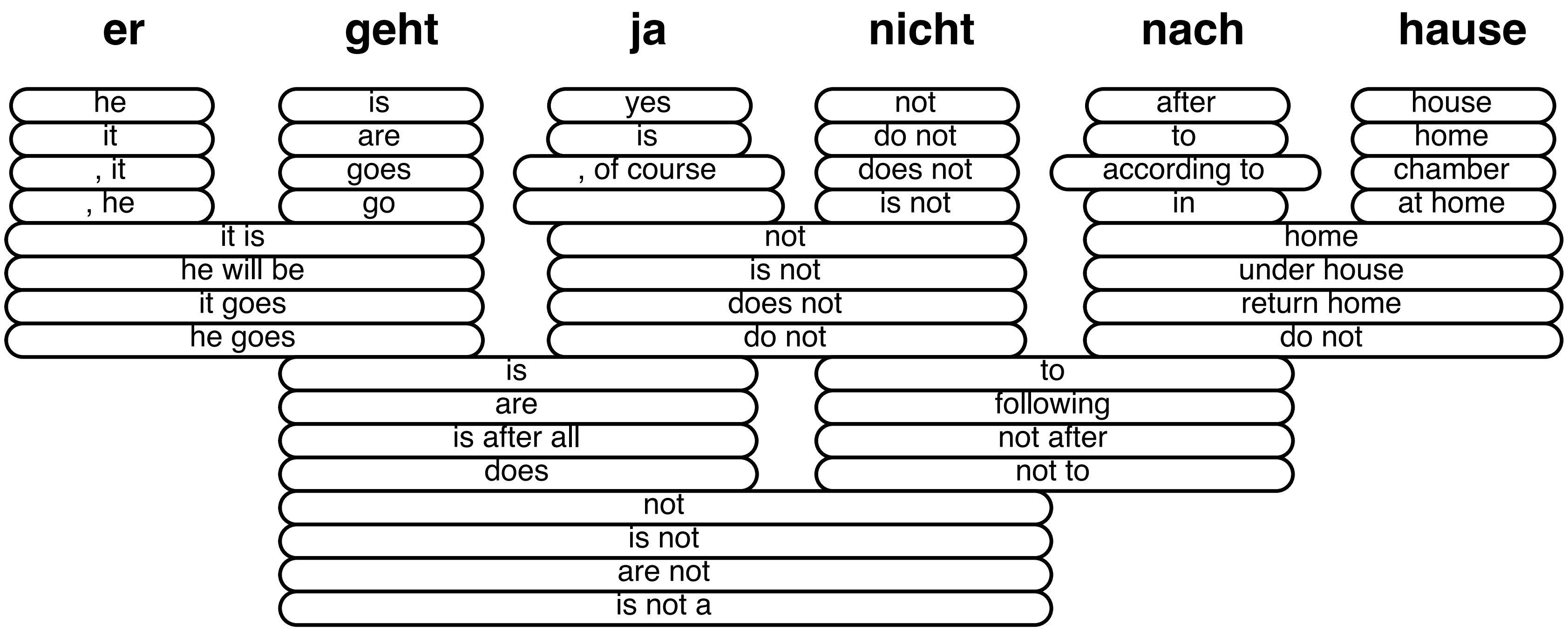
Translation step 4

- Task: translate this sentence from German into English



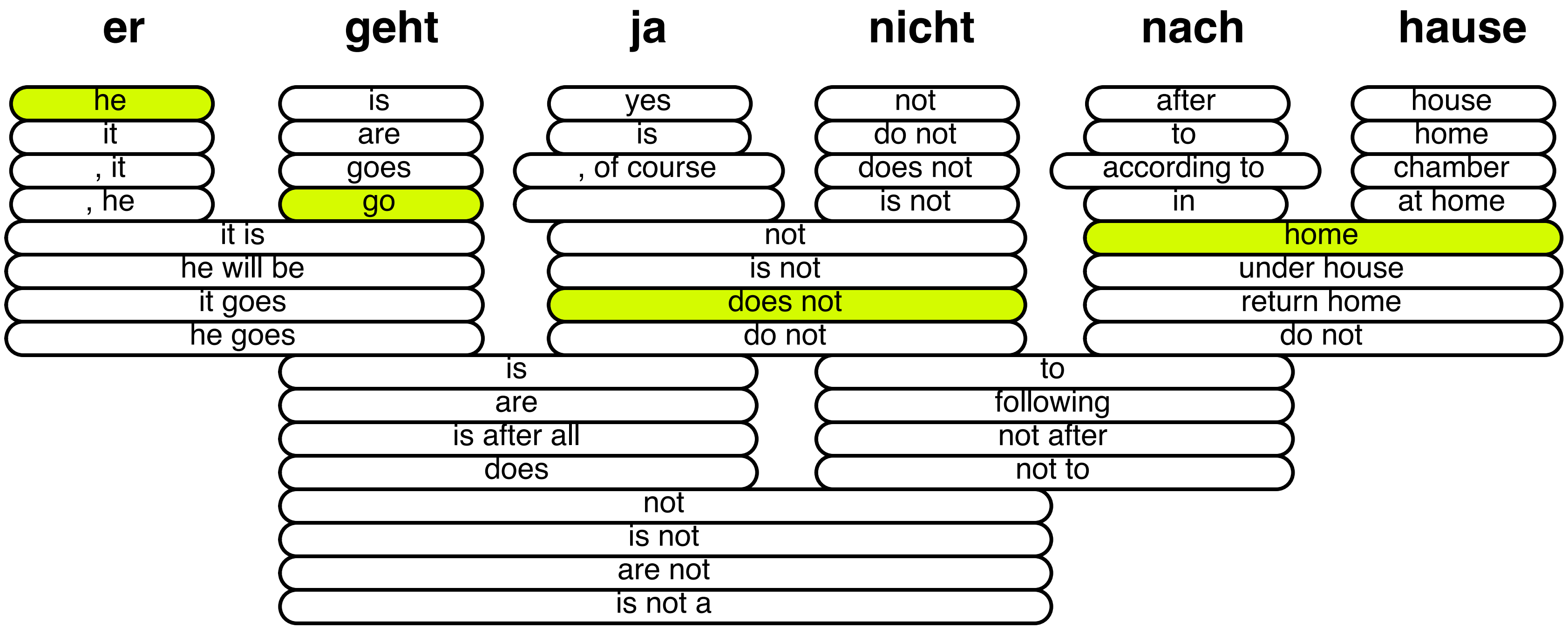
- Pick phrase in input, translate

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

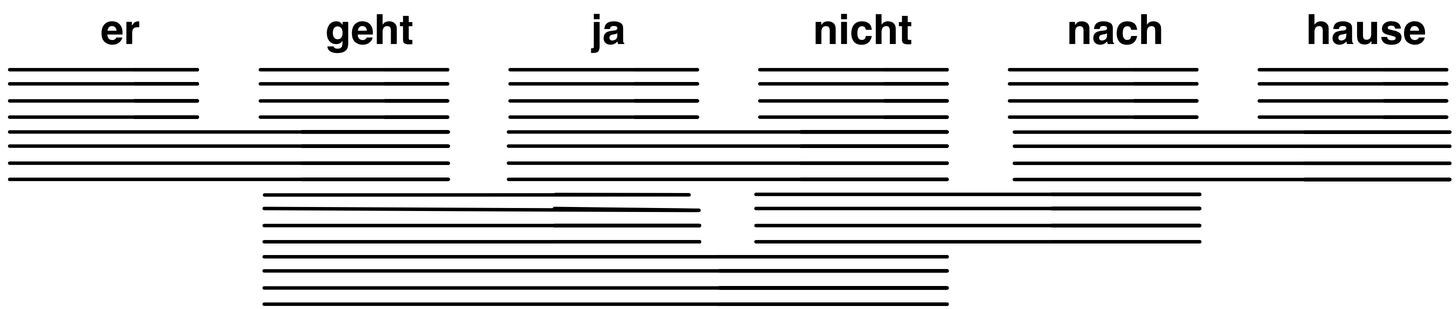
Translation options



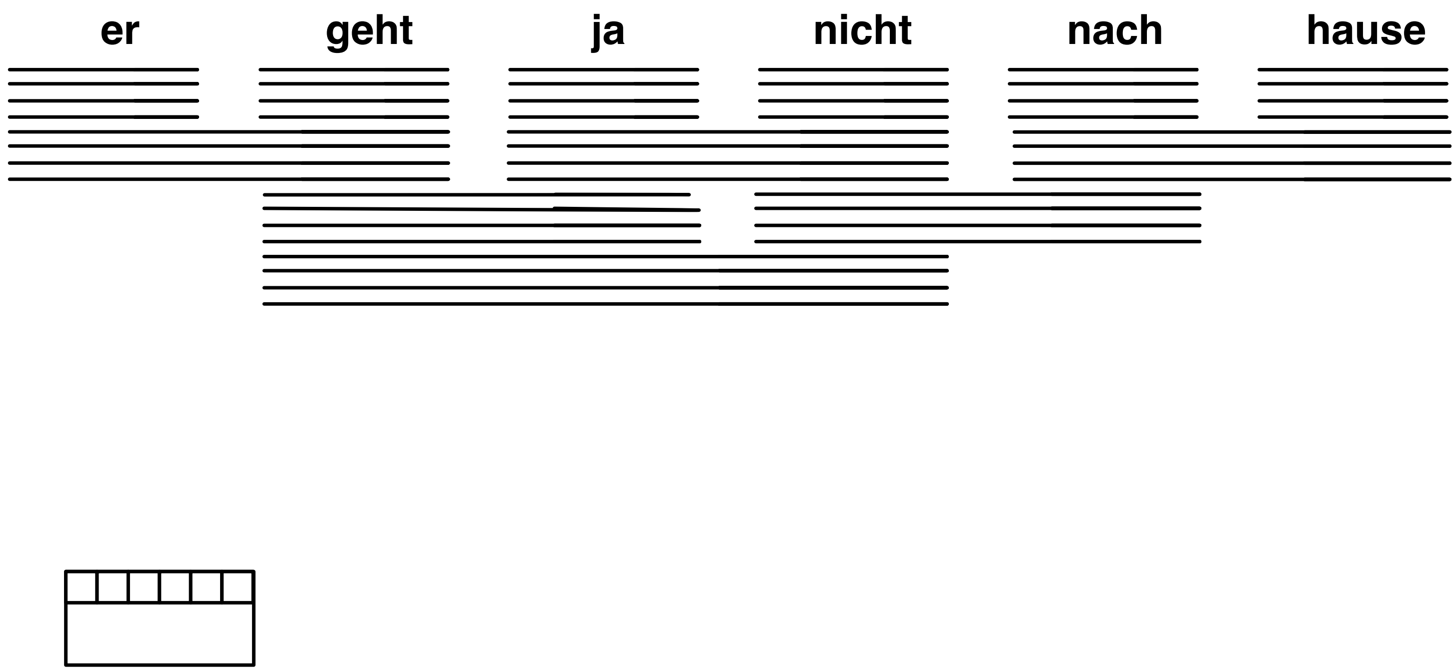
- The machine translation decoder does not know the right answer

→ *Search problem* solved by heuristic beam search

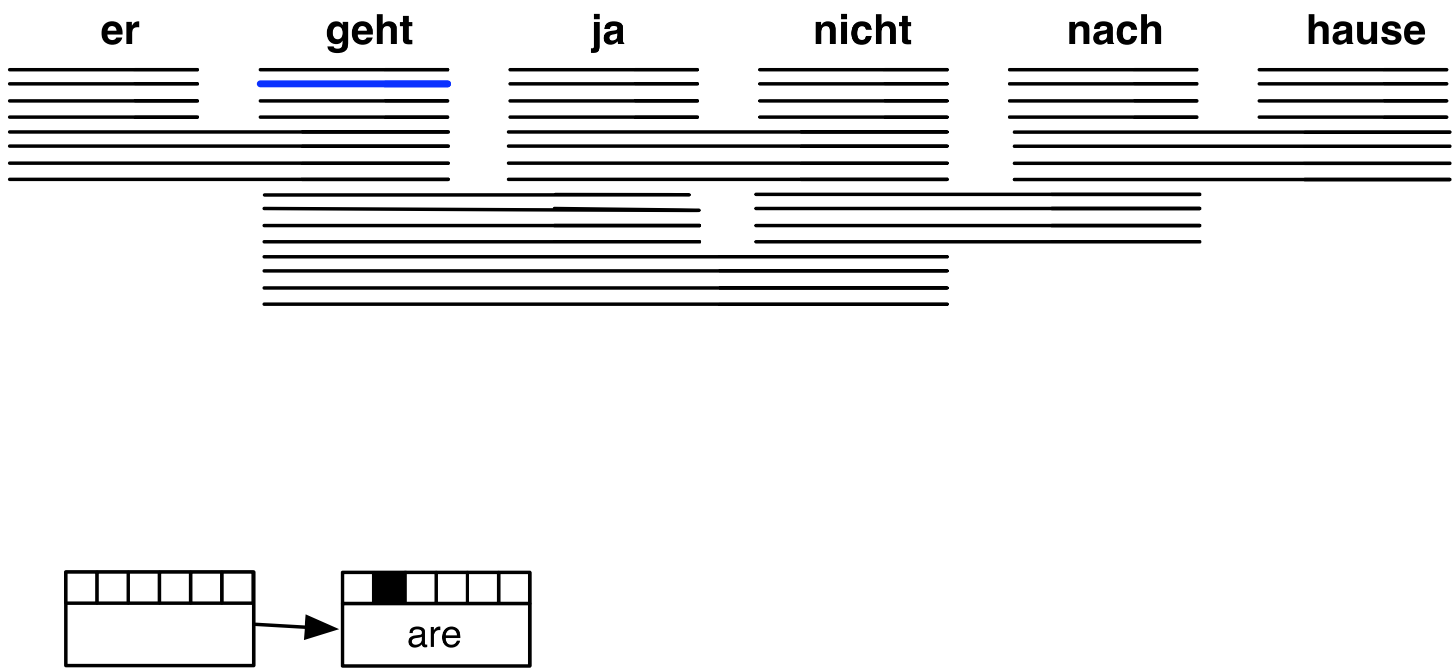
Decoding process: precompute translation options



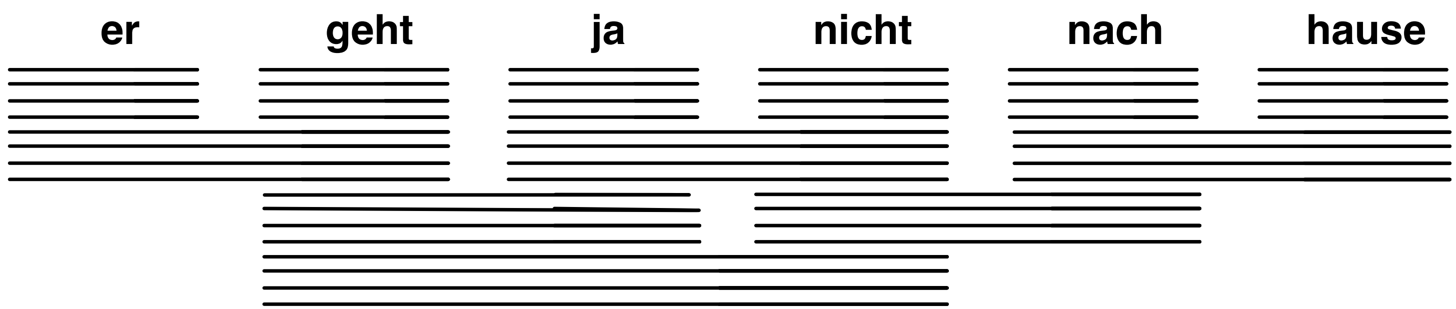
Decoding process: start with initial hypothesis

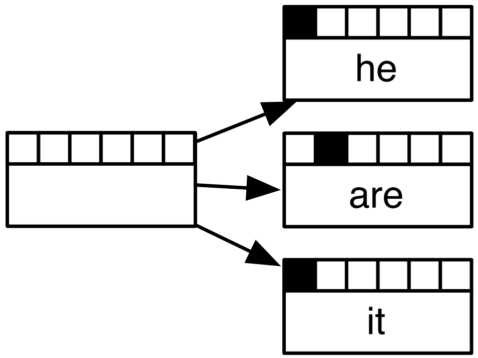


Decoding process: hypothesis expansion

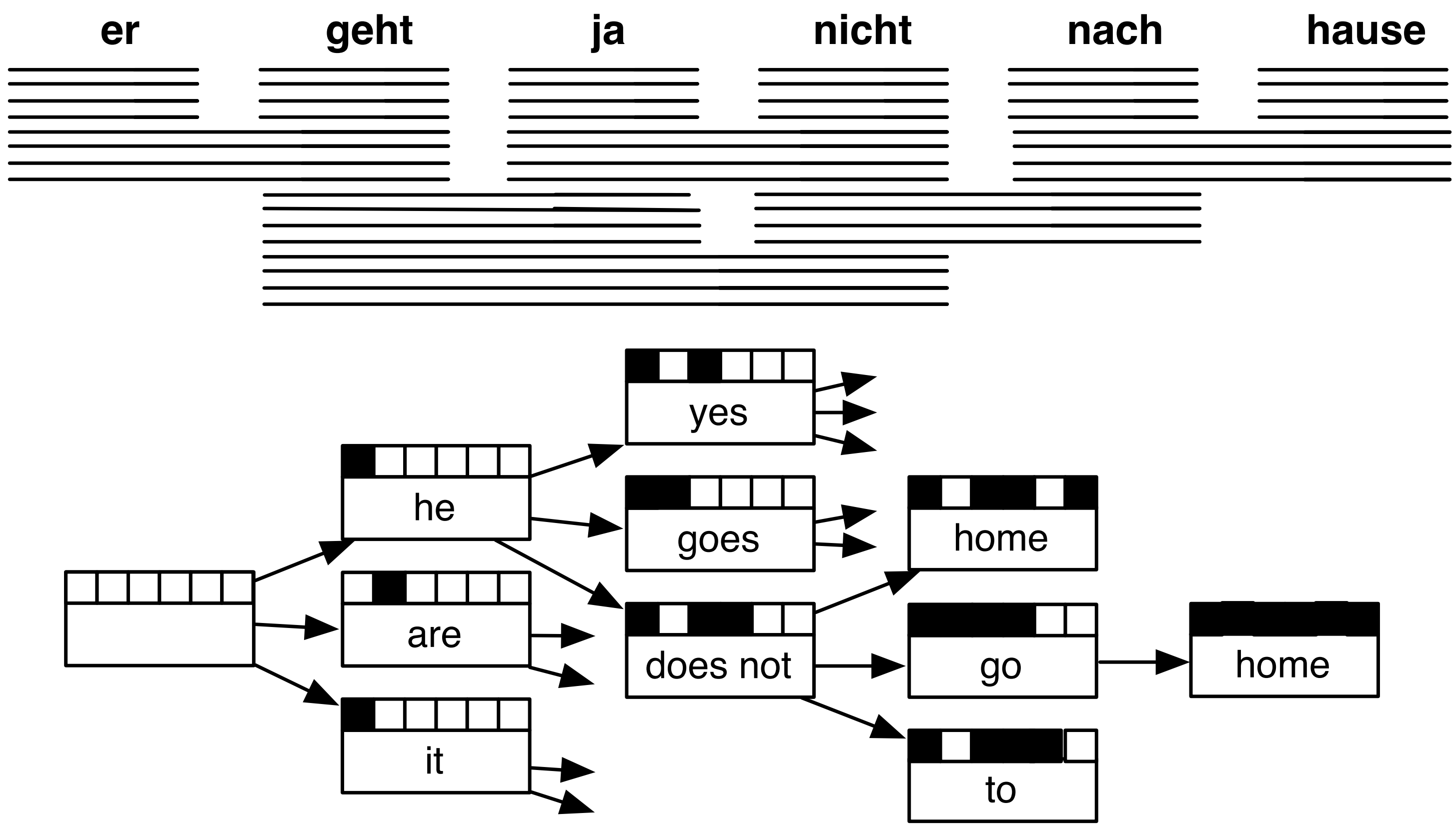


Decoding process: hypothesis expansion

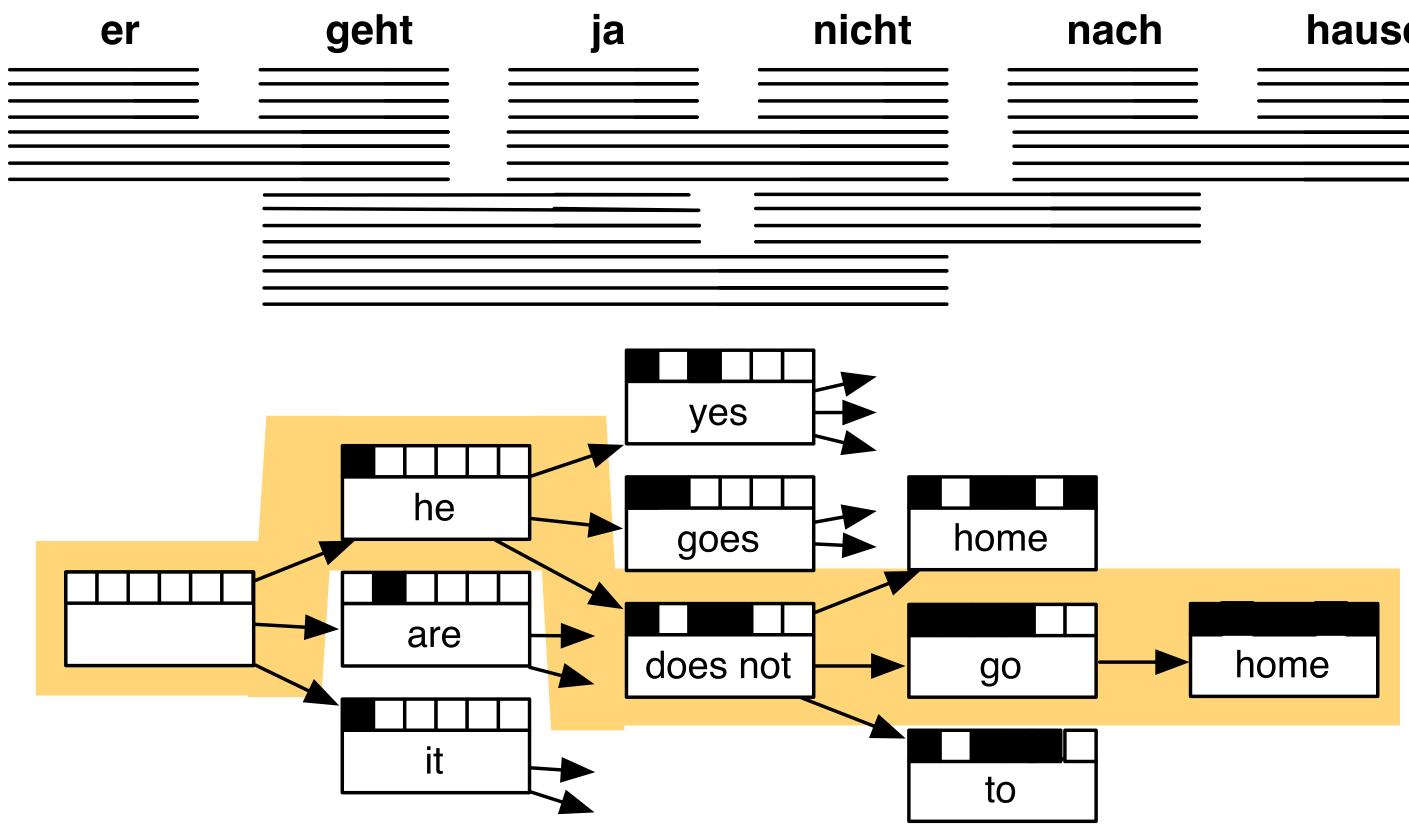




Decoding process: hypothesis expansion



Decoding process: find best path

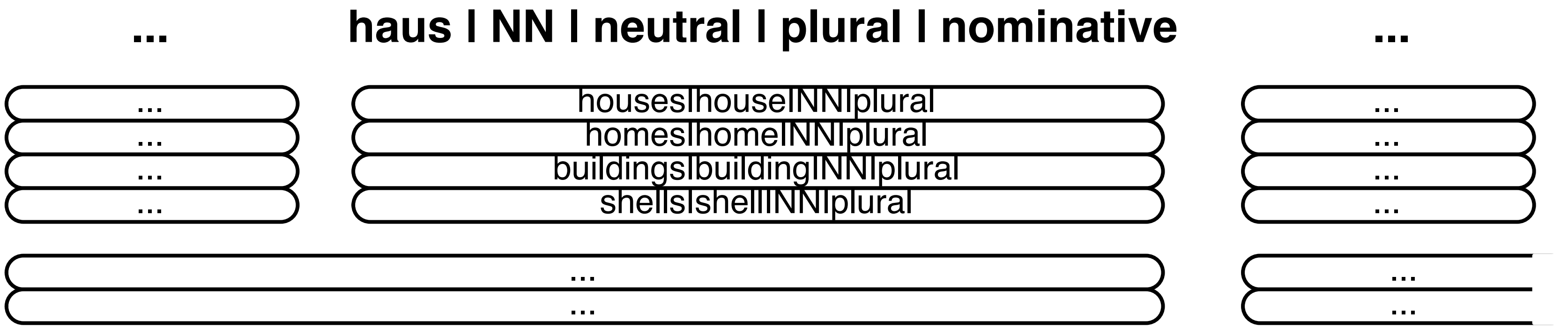


Factored model decoding

- 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*



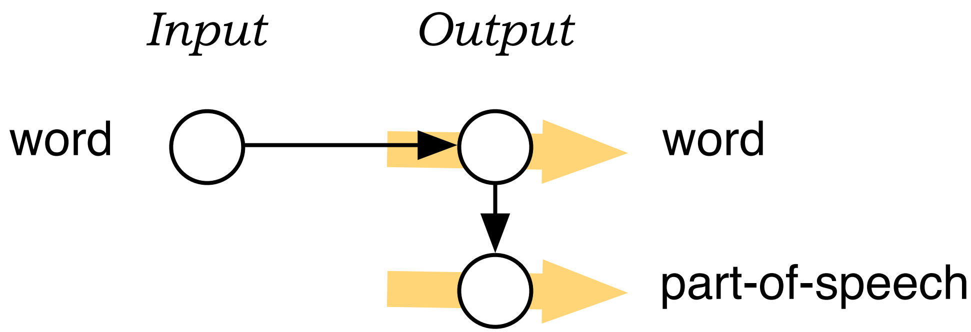
Efficient factored model decoding

- 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

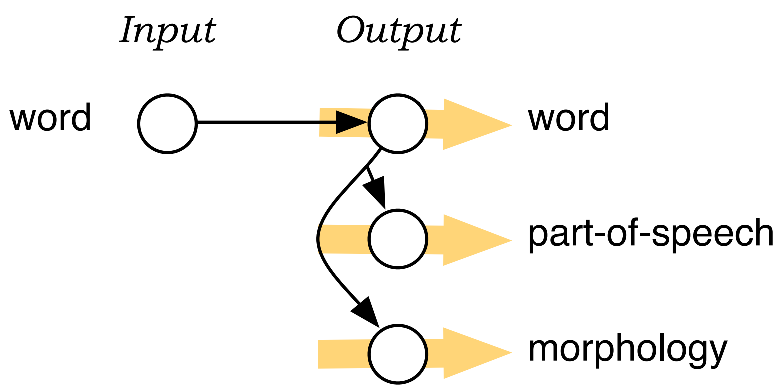
- Improvements under sparse data conditions
- Similar results with CCG supertags [Birch et al., 2007]

Sequence models over morphological tags

die	**hellen**	**Sterne**	**erleuchten**	**das**	**schwarze**	**Himmel**
(the)	*(bright)*	*(stars)*	*(illuminate)*	*(the)*	*(black)*	*(sky)*
fem	*fem*	*fem*	*-*	*neutral*	*neutral*	*male*
plural	*plural*	*plural*	*plural*	*sgl.*	*sgl.*	*sgl*
nom.	*nom.*	*nom.*	*-*	*acc.*	*acc.*	*acc.*

- Violation of noun phrase agreement in gender
  - *das schwarze* and *schwarze Himmel* are perfectly fine bigrams
  - but: *das schwarze Himmel* is not
- If relevant n-grams does not occur in the corpus, a lexical n-gram model would *fail to detect* this mistake
- Morphological sequence model: $p(N\text{-}male|J\text{-}male) > p(N\text{-}male|J\text{-}neutral)$

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

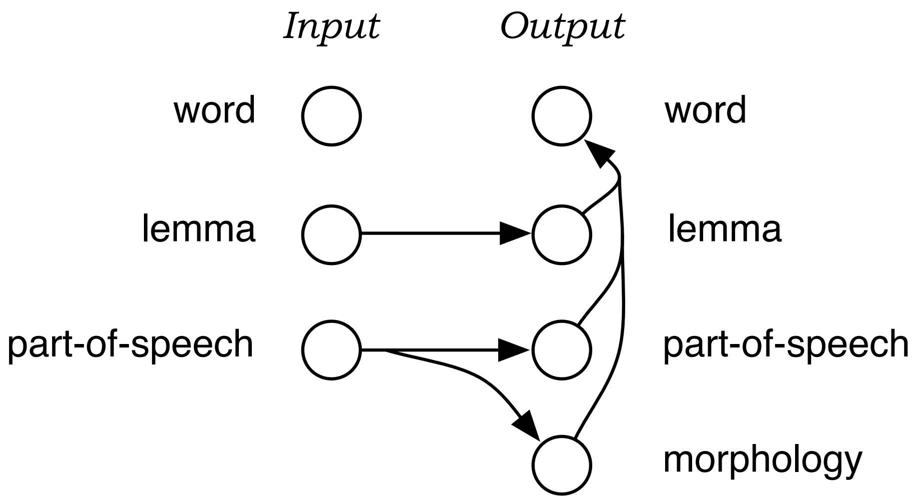
Agreement within noun phrases

- Experiment: 7-gram POS, morph LM in addition to 3-gram word LM
- Results

Method	Agreement errors in NP	devtest	test
baseline	15% in NP $\geq$ 3 words	18.22 BLEU	18.04 BLEU
factored model	4% in NP $\geq$ 3 words	18.25 BLEU	18.22 BLEU

- Example
  - baseline: *... zur zwischenstaatlichen methoden ...*
  - factored model: *... zu zwischenstaatlichen methoden ...*
- Example
  - baseline: *... das zweite wichtige änderung ...*
  - factored model: *... die zweite wichtige änderung ...*

Morphological generation model



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

Initial results

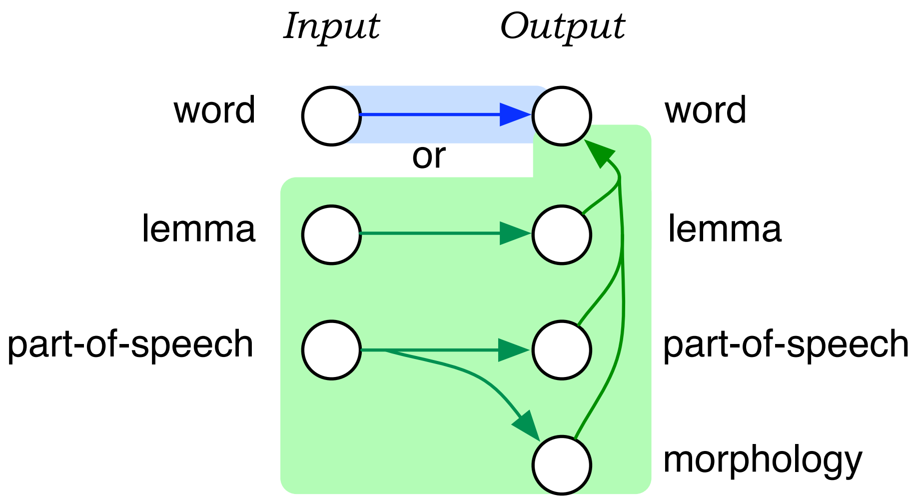
- Results on 1 million word News Commentary corpus (German–English)

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?

$\rightarrow$ 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

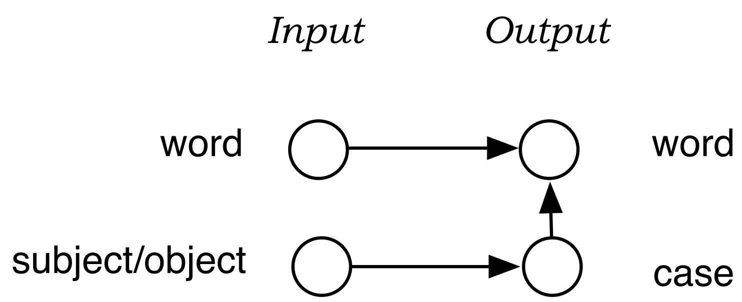
- Model now beats the baseline:

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

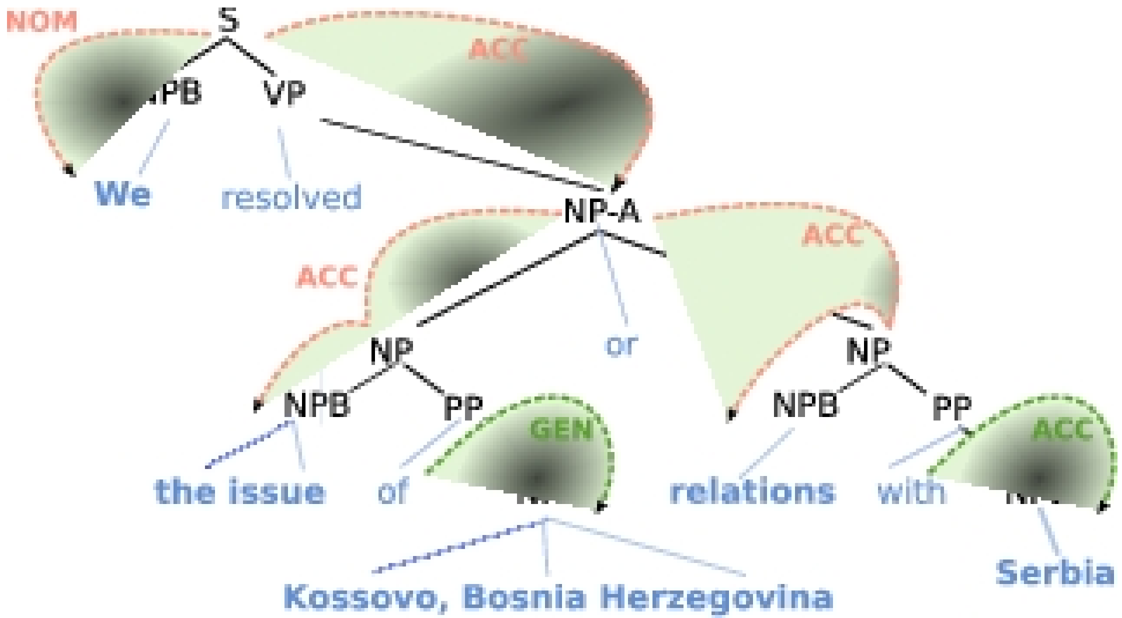
Case Information for English–Greek



- 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

Factored Template Models

- **Long range** reordering
  - movement often not limited to local changes
  - German-English: *SBJ AUX OBJ V* → *SBJ AUX V OBJ*
- Template models
  - some factor mappings (POS, syntactic chunks) may have longer scope than others (words)
  - larger mappings form template for shorter mappings
  - computational problems with this
- published in [Hoang and Koehn, EACL 2009]

Shallow syntactic features

the	**paintings**	**of**	**the**	**old**	**man**	**are**	**beautiful**
-	*plural*	*-*	*-*	*-*	*singular*	*plural*	*-*
B-NP	*I-NP*	*B-PP*	*I-PP*	*I-PP*	*I-PP*	*V*	*B-ADJ*
SBJ	*SBJ*	*OBJ*	*OBJ*	*OBJ*	*OBJ*	*V*	*ADJ*

- Shallow syntactic tasks have been formulated as sequence labeling tasks
  - base noun phrase chunking
  - syntactic role labeling
- Results presented in [Cettolo et al., AMTA 2008]