Statistical and Neural Machine Translation

October 6th, 2016

UNIVERSITÄT DES SAARLANDES

Josef van Genabith

Josef.van_Genabith@dfki.de

BALTIC-HLT 2016

Riga, Latvia

Context: WMT 2016

- Neural MT won ¾ off all shared tasks

- Against strong state-of-the art: PB-SMT
- Honed and optimized over 15 years
- SMT > 25 years old

- NMT new kid on the block, about 2 years old …

Language

- Human languages are:
  - Elegant
  - Efficient
  - Flexible
  - Complex
- One word/sentence may mean many things
- Many ways of saying the same thing
- Meaning depends on context
- Literal and figurative language (metaphor)
- Language and culture (different ways of conceptualising the same thing)
- Word order
- Morphology
- …

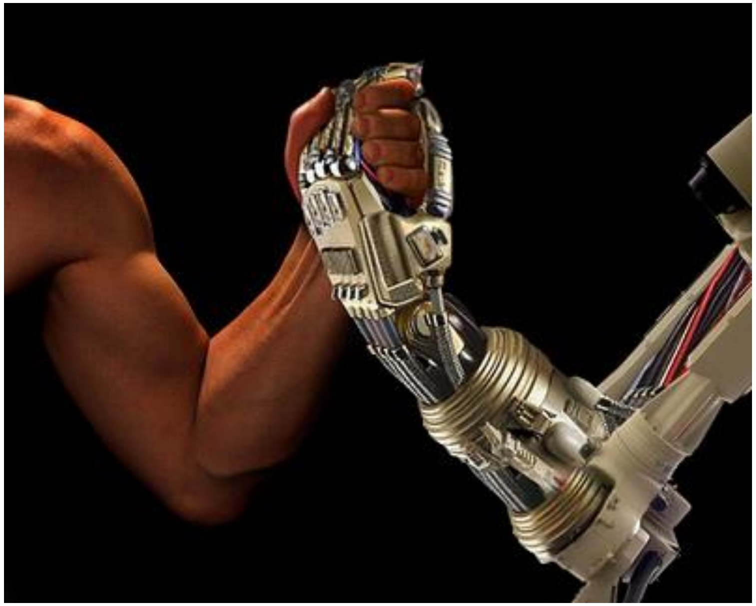


Image: http://workingtropes.lmc.gatech.edu/wiki/index.php/File:Man-vs-machine.jpg

Language is Complex

- Language is complex
- We cannot compute it exactly
- We tried: rule-based LT …

- What do we do?

- Machine Learning
  - Learns from data
  - Approximate solution ⇒ not perfect
    - Robust
    - Scalable

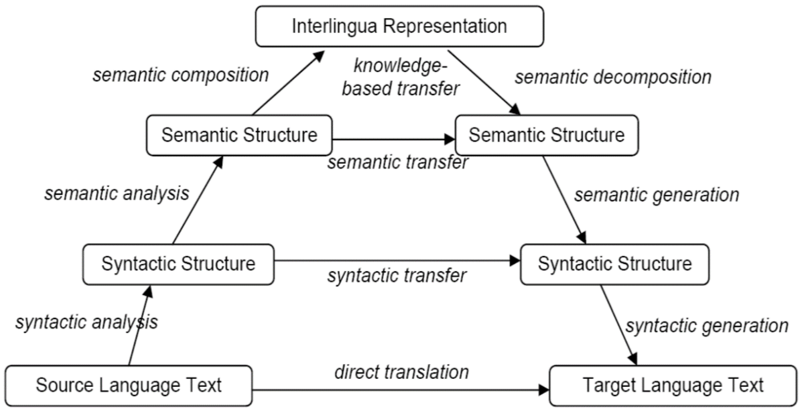
Story of Language Technology

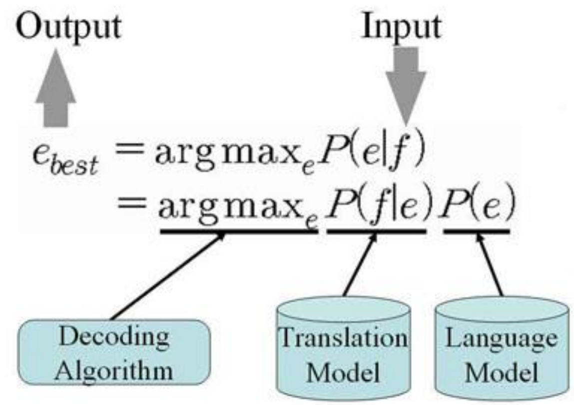
- Story of Machine Translation
  - Rule-based — direct word based, transfer based
  - Statistical I — Machine Learning I, IBM, PB-SMT
  - Statistical II — Machine Learning II: NMT, Deep Learning

- Systems Engineering
- Machine Learning

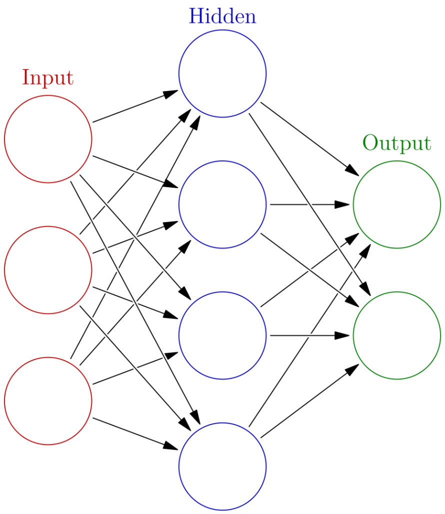
- Story is partial and biased

The Journey

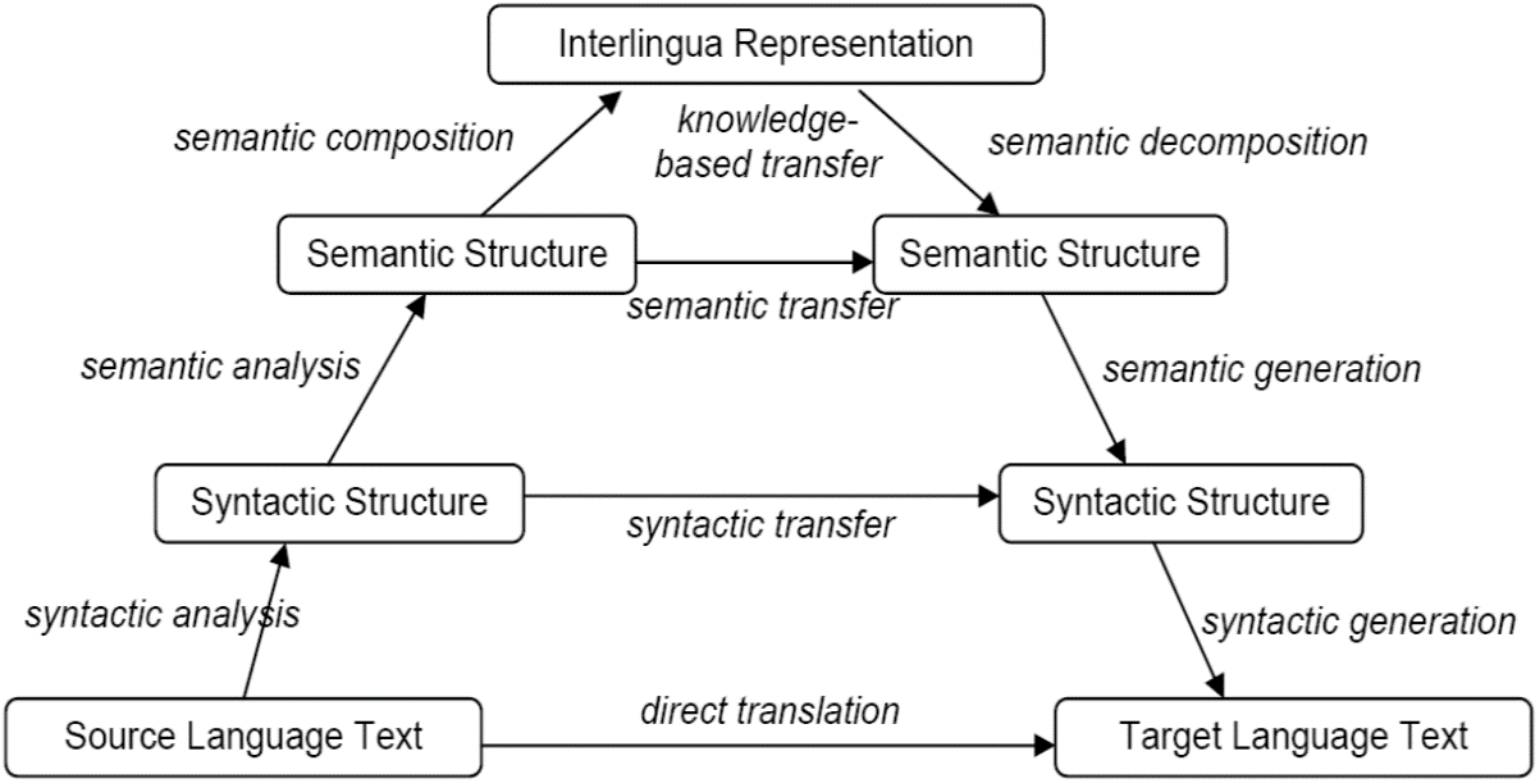




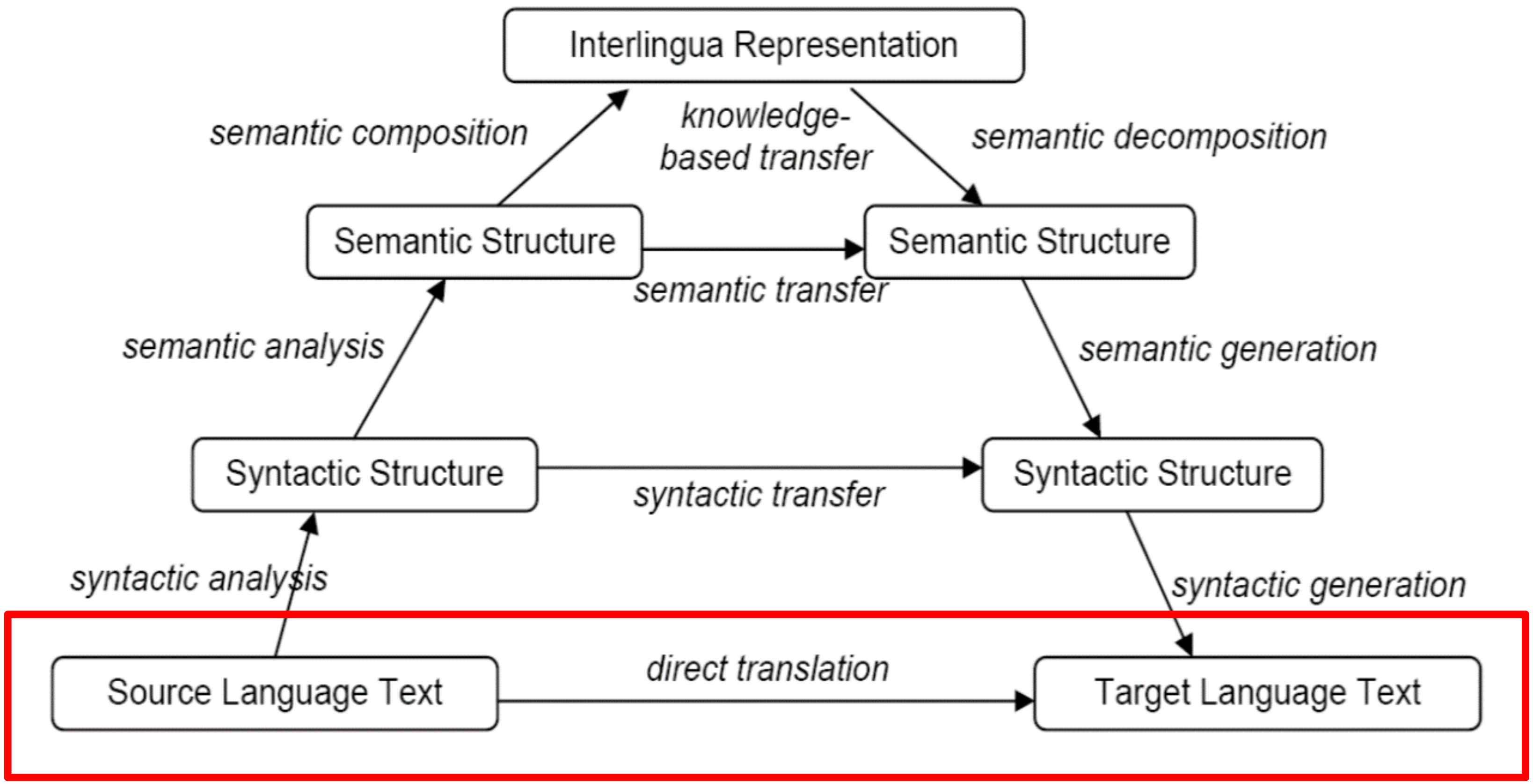
$$p(x) = \exp \sum_{i=1}^{n} \lambda_i h_i(x)$$



Rule-based Machine Translation (RBMT)



Rule-based Machine Translation (RBMT)



Rule-based Machine Translation I

- Translate word by word: “direct translation”

- Do a little bit of analysis of local source context
- Maybe a little local re-ordering in target (e.g. French adjectives tend to follow noun)

- Requires very large bilingual dictionary with rules of how to translate each word

Rule-based Machine Translation I

function DIRECT_TRANSLATE_MUCH/MANY(word) **returns** Russian translation

if preceding word is *how* **return** *skol'ko*
else if preceding word is *as* **return** *stol'ko zhe*
else if word is *much*
  if preceding word is *very* **return** nil
  else if following word is a noun **return** *mnogo*
else /* word is many */
  if preceding word is a preposition and following word is a noun **return** *mnogii*
  else return *mnogo*

Figure 25.7 A procedure for translating *much* and *many* into Russian, adapted from Hutchins' (1986, pg. 133) discussion of Panov 1960. Note the similarity to decision list algorithms for word sense disambiguation.

From: Jurafsky & Martin II

Rule-based Machine Translation I

- 10s of thousands of manually constructed entries with rules
- Systran (kind off) and other early commercial systems
- Interesting: contributes to linguistic knowledge
- Need highly skilled experts
- Time consuming & expensive
- Rule interaction hard to predict
- Long range phenomena hard to capture
- Generalisations hard to capture

Rule-based Machine Translation I

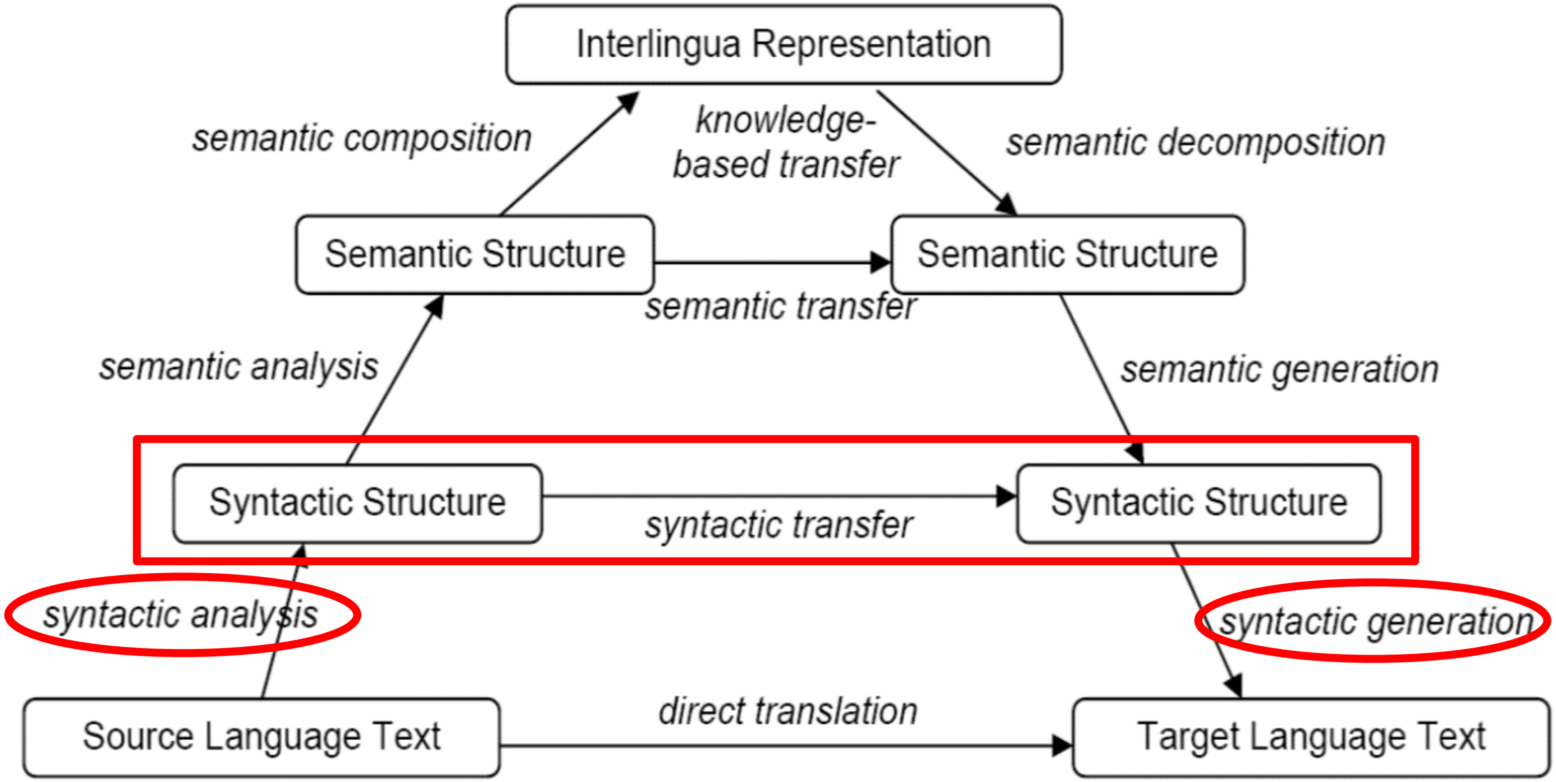
- Long range phenomena hard to capture:

EN: *Google will invest in self-driving cars*
DE: *Google wird in selbst fahrende Autos investieren*

EN: *Reuters said IBM bought Lotus yesterday*
JA: *Reuters yesterday IBM Lotus bought said*

- Need not just local but global information
- some global (syntactic/semantic) analysis

Rule-based Machine Translation II



Rule-based Machine Translation II

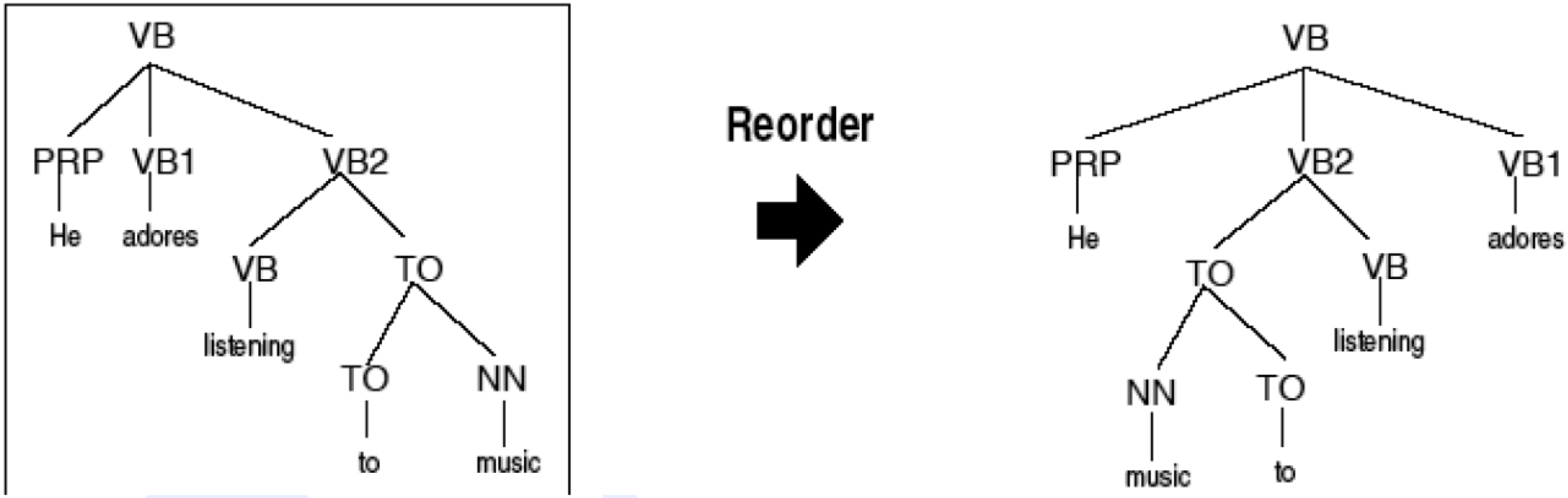
EN:	He adores listening to music	SVO
JA:	He music to listening adores	SOV

Rule-based Machine Translation II

EN: *He adores listening to music*

JA: *He music to listening adores*

From: Jurafsky & Martin II

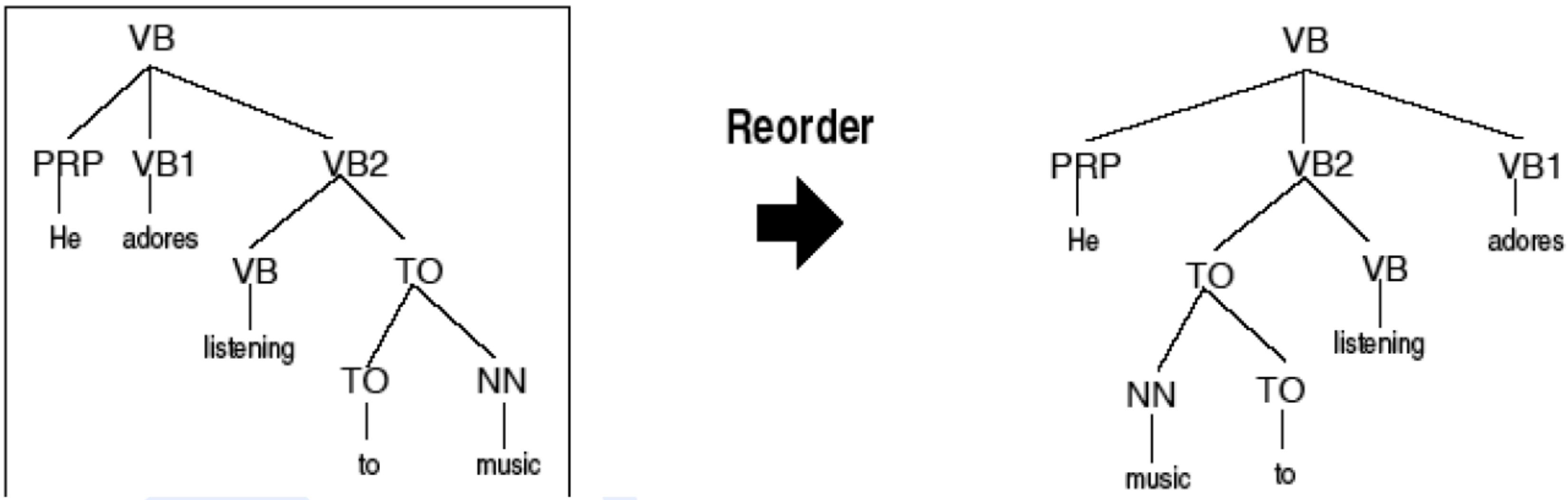


Rule-based Machine Translation II

EN: *He adores listening to music*

JA: *He music to listening adores*

From: Jurafsky & Martin II



1.	VB → PRP VB1 VB2	⇒	VB → PRP VB2 VB1
2.	VB2 → VB TO	⇒	VB2 → TO VB
3.	TO → TO NN	⇒	TO → NN TO

Rule-based Machine Translation II

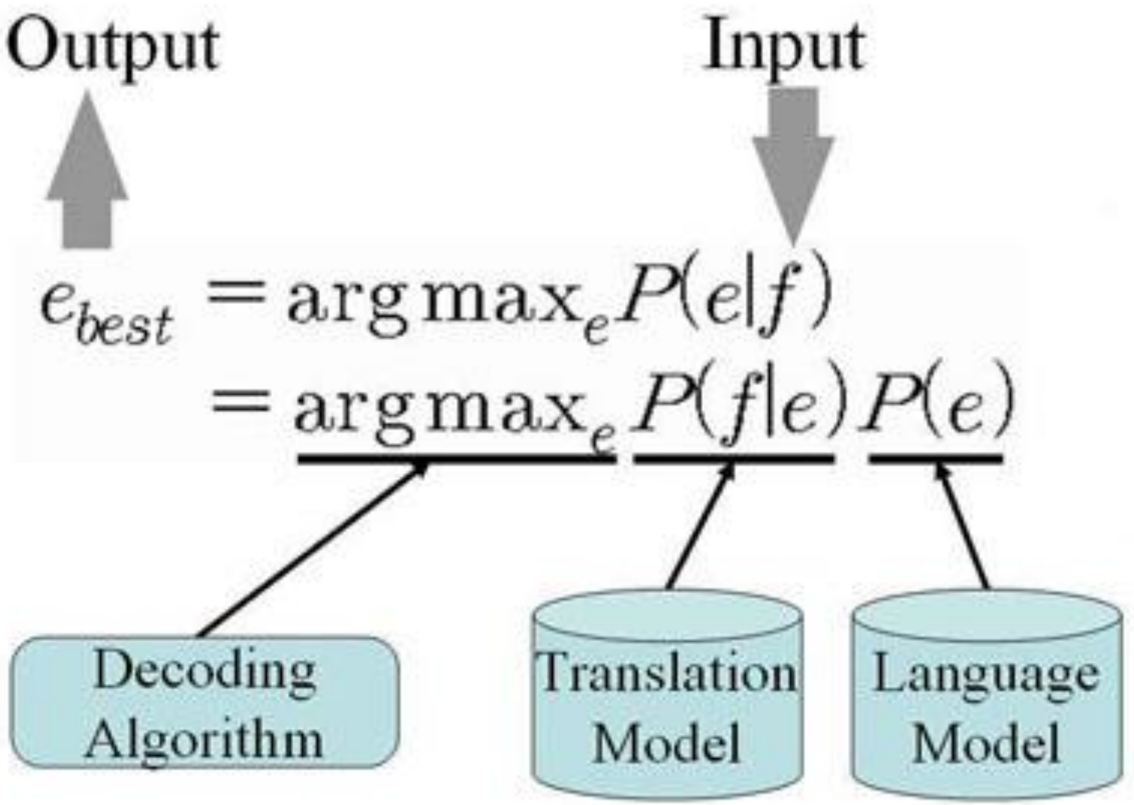
- Need a lot of resources for transfer-based MT:

- Analysis/generation lexica and grammars, as well as parsing and generation engines for source and target
- Transfer rule sets and a transfer engine for any two languages you want to translate between $n \times (n-1)$

- Interesting: strong contribution to linguistic knowledge

- Time consuming and expensive to hand-craft (… learn …)
- Not easy to achieve good coverage
- Complex phenomena
- Large rule sets
- Difficult to manage rule interactions

Statistical MT (Machine Learning) I



Statistical (Machine Learning) I

$$P(a,f|e) = \binom{m-\varphi_0}{\varphi_0} \times {p_0}^{(m-2\varphi_0)} \times {p_1}^{\varphi_0}$$

$$\times \prod_{i=1}^{l} n(\varphi_i|e_i) \times \prod_{j=1}^{m} t\left(f_j \middle| e_{a_j}\right)$$

$$\times \prod_{j:a_j \neq 0}^{m} d\left(j \middle| a_j, l, m\right) \times \prod_{i=0}^{l} \varphi_i! \times \frac{1}{\varphi_0!}$$

Recall that

$$P(f|e) = \sum_a P(a,f|e) \quad \text{and} \quad P(a|e,f) = \frac{P(a,f|e)}{\sum_a P(a,f|e)}$$

Statistical (Machine Learning) I

Mary did not slap the green witch

Maria no daba una bofetada a la bruja verde

Statistical (Machine Learning) I

Mary did not slap the green witch

Maria no daba una bofetada a la bruja verde

Statistical (Machine Learning) I

Mary did not slap the green witch

slap slap slap

Maria no daba una bofetada a la bruja verde

Statistical (Machine Learning) I

Mary did not slap the green witch

slap slap slap

Maria no daba una bofetada a la bruja verde

Statistical (Machine Learning) I

Mary did not slap the green witch

∅ slap slap slap

Maria no daba una bofetada a la bruja verde

Statistical (Machine Learning) I

Mary did not slap the green witch

∅ slap slap slap

Maria no daba una bofetada a la bruja verde

Statistical (Machine Learning) I

NULL Mary did not slap the green witch

∅ slap slap slap

Maria no daba una bofetada a la bruja verde

Statistical (Machine Learning) I

NULL Mary did not slap the green witch

∅ slap slap slap

NULL

a

Maria no daba una bofetada a la bruja verde

Statistical (Machine Learning) I

NULL Mary did not slap the green witch

Mary ∅ not slap slap slap the green witch

Mary not slap slap slap *NULL* the green witch

a

Maria no daba una bofetada a la bruja verde

Statistical (Machine Learning) I

NULL Mary did not slap the green witch

Mary ∅ not slap slap slap the green witch

Mary not slap slap slap *NULL* the green witch

Maria no daba una bofetada a la verde bruja

Maria no daba una bofetada a la bruja verde

Statistical (Machine Learning) I

NULL Mary did not slap the green witch

Mary ∅ not slap slap slap the green witch

Mary not slap slap slap *NULL* the green witch

Maria no daba una bofetada a la verde bruja

Maria no daba una bofetada a la bruja verde

Statistical (Machine Learning) I

NULL Mary did not slap the green witch

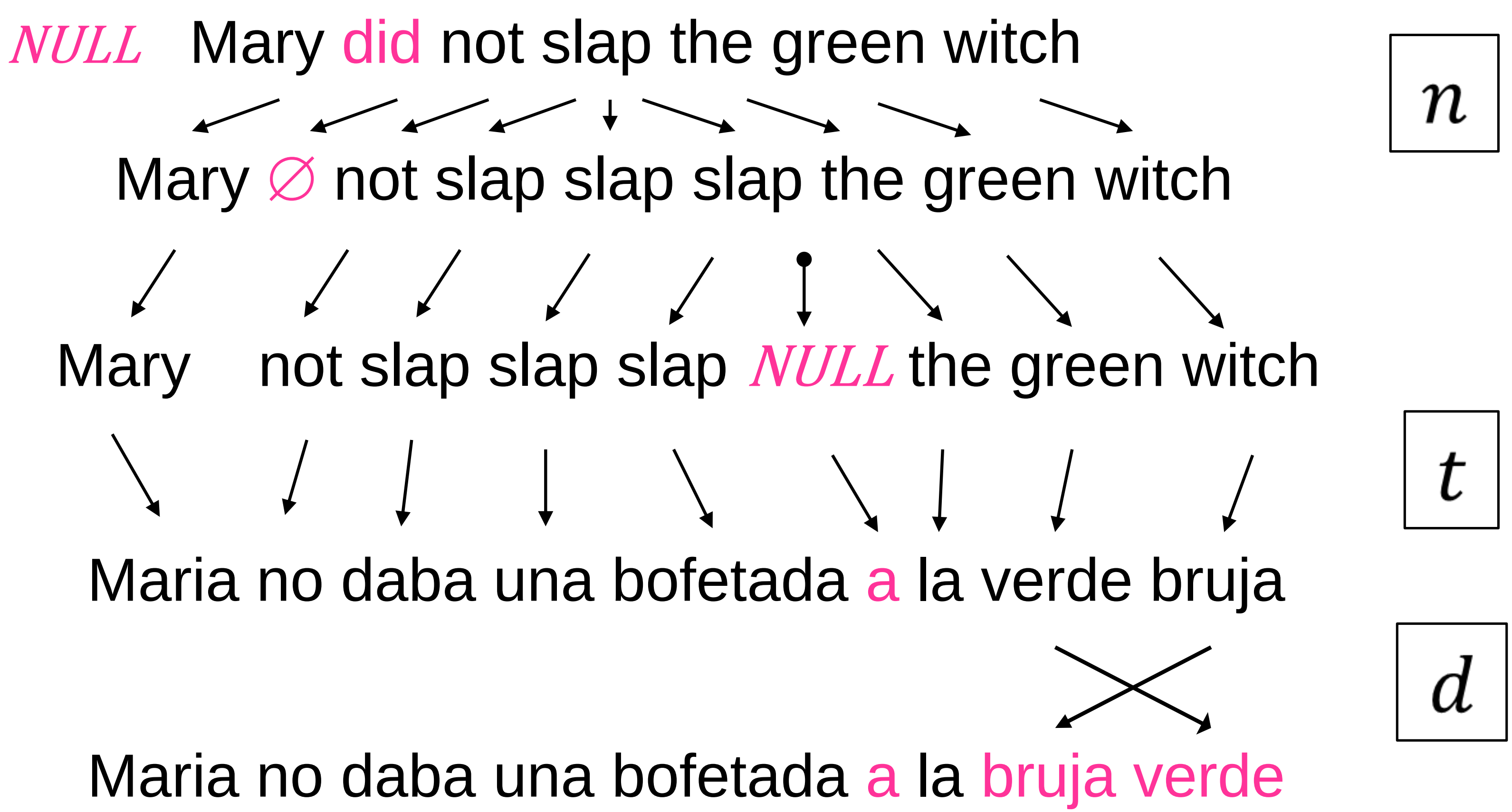
Mary ∅ not slap slap slap the green witch

Mary not slap slap slap *NULL* the green witch

Maria no daba una bofetada a la verde bruja

Maria no daba una bofetada a la bruja verde

Statistical (Machine Learning) I



Statistical (Machine Learning) I

NULL Mary did not slap the green witch

Maria no daba una bofetada a la bruja verde

$$\langle 1, 3, 4, 4, 4, 0, 5, 7, 6 \rangle$$

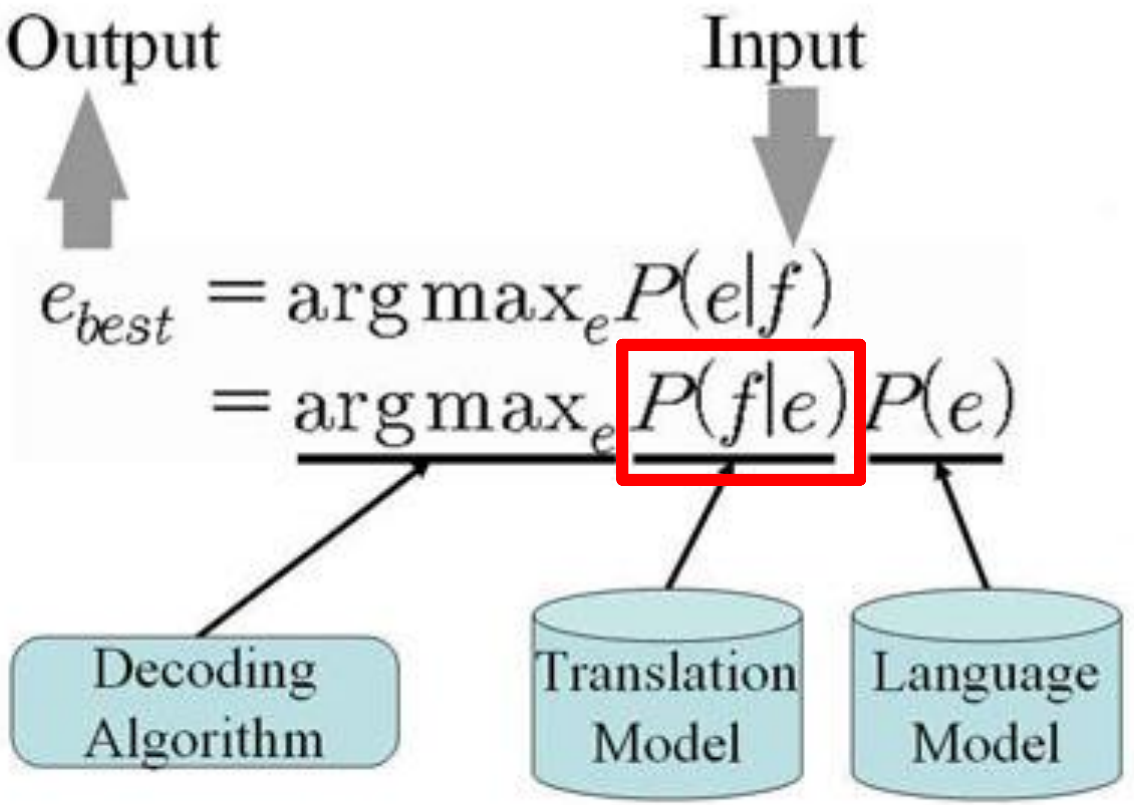
Statistical (Machine Learning) I

$$P(a,f|e) = \binom{m-\varphi_0}{\varphi_0} \times p_0^{(m-2\varphi_0)} \times p_1^{\varphi_0}$$
$$\times \prod_{i=1}^{l} n(\varphi_i|e_i) \times \prod_{j=1}^{m} t\left(f_j \middle| e_{a_j}\right)$$
$$\times \prod_{j:a_j \neq 0}^{m} d\left(j \middle| a_j, l, m\right) \times \prod_{i=0}^{l} \varphi_i! \times \frac{1}{\varphi_0!}$$

Recall that

$$P(f|e) = \sum_a P(a,f|e) \quad \text{and} \quad P(a|e,f) = \frac{P(a,f|e)}{\sum_a P(a,f|e)}$$

Statistical (Machine Learning) I



Statistical (Machine Learning) I

- IBM Models
- Word based SMT
- Master class in statistical modeling
- Its amazing that you can learn this from raw data: bi-text!
- Expectation Maximization (EM)
- Then all based on statistical decision theory

Statistical (Machine Learning) I

- What is “wrong” with this?
- Local decisions
- Lots of massive independence assumptions
- Not warranted by the data …: non-local phenomena
- Reordering is weak …
- OOVs …
- … lots more ☺
- Word salad …

Statistical (Machine Learning) I

$$P(a, f|e) = \binom{m - \varphi_0}{\varphi_0} \times p_0{}^{(m-2\varphi_0)} \times p_1{}^{\varphi_0}$$

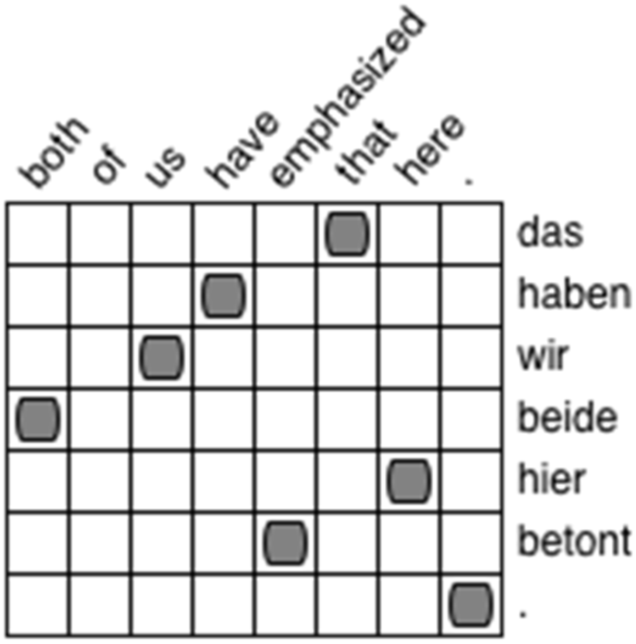
$$\times \prod_{i=1}^{l} n(\varphi_i|e_i) \times \prod_{j=1}^{m} t\left(f_j \middle| e_{a_j}\right)$$

$$\times \prod_{j:a_j \neq 0}^{m} d\left(j \middle| a_j, l, m\right) \times \prod_{i=0}^{l} \varphi_i! \times \frac{1}{\varphi_0!}$$

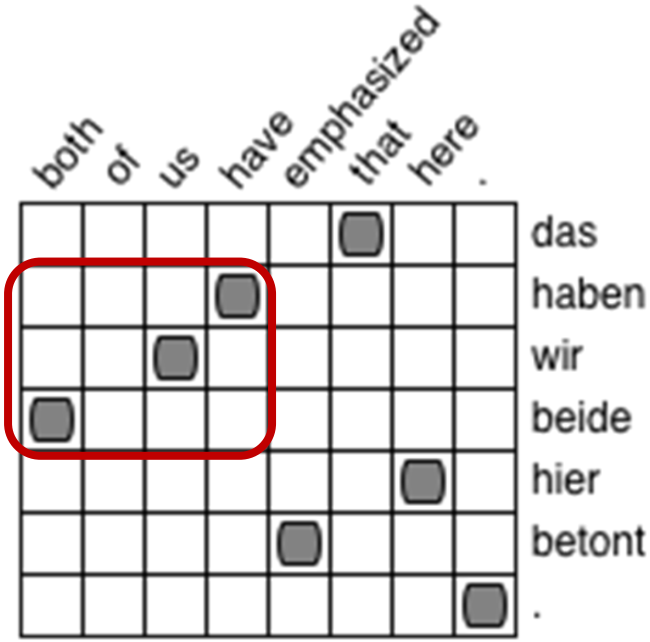
Recall that

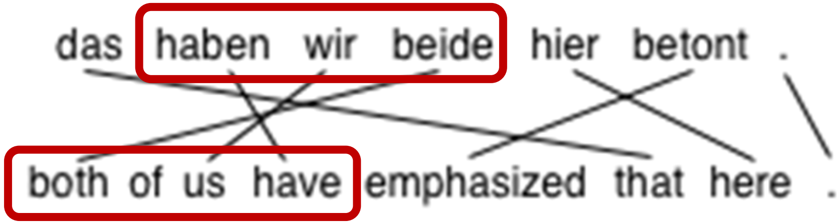
$$P(f|e) = \sum_a P(a, f|e) \quad \text{and} \quad P(a|e, f) = \frac{P(a, f|e)}{\sum_a P(a, f|e)}$$

Statistical MT (Machine Learning) I.1

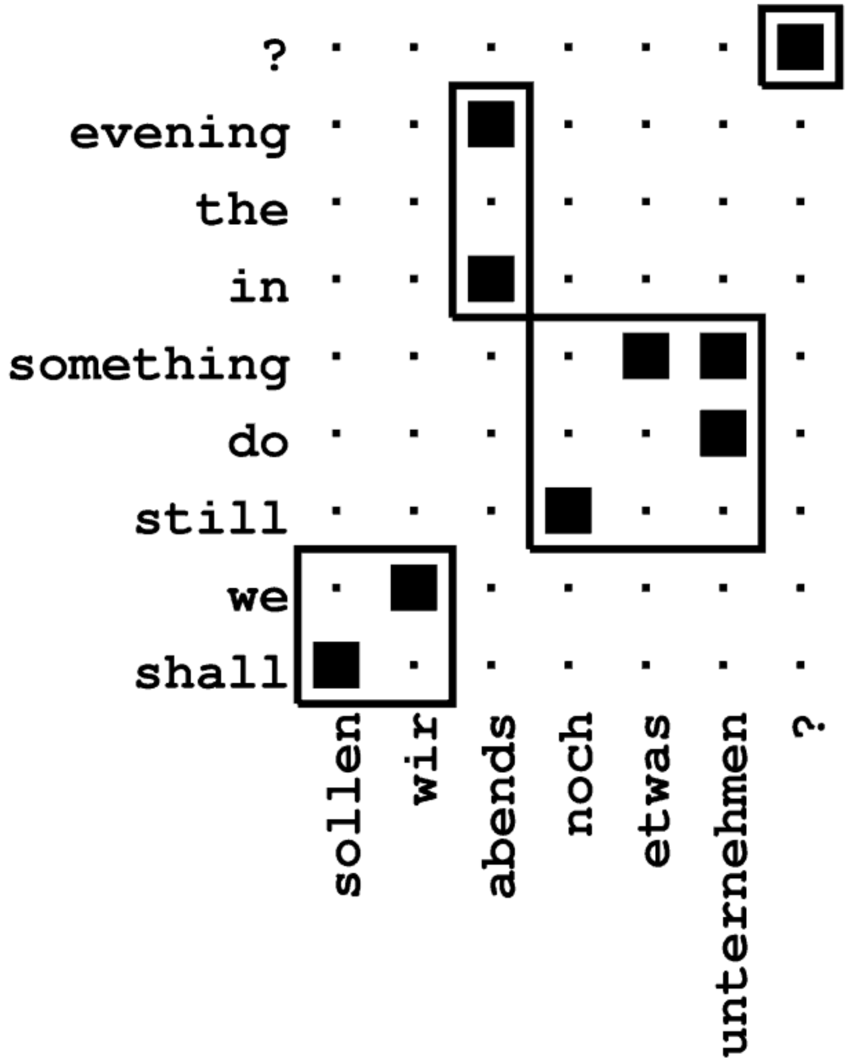


Statistical (Machine Learning) I.1

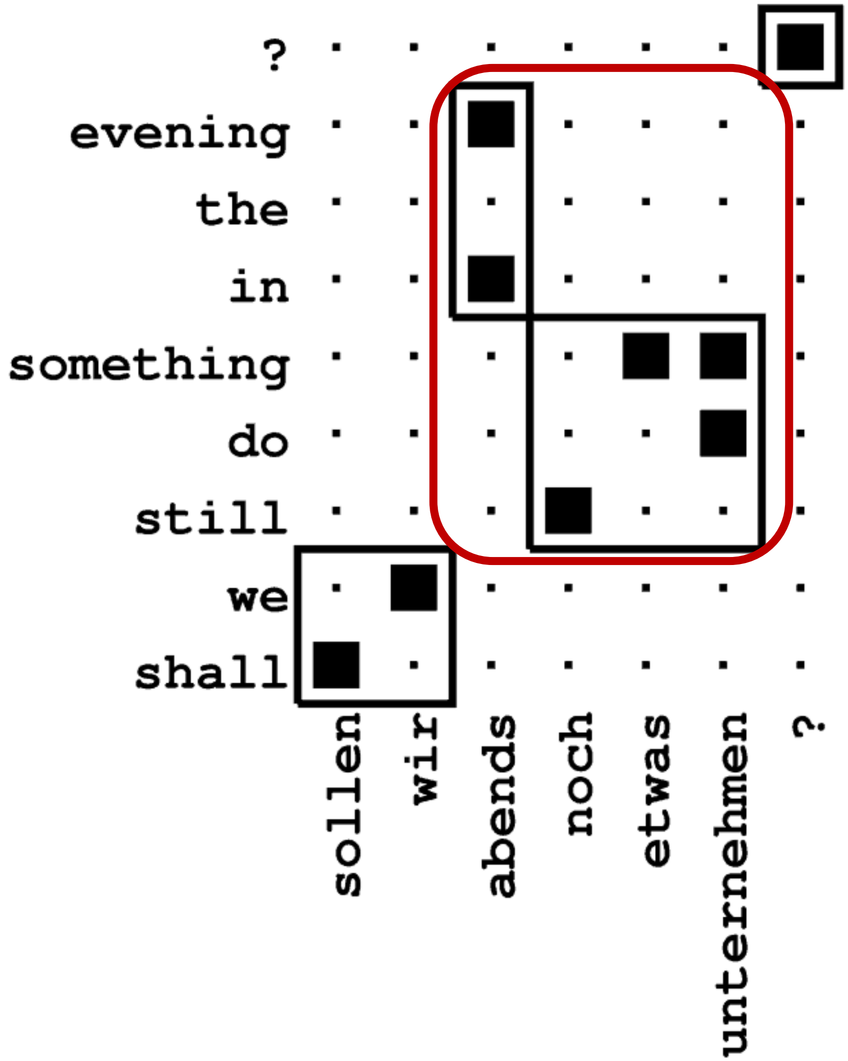




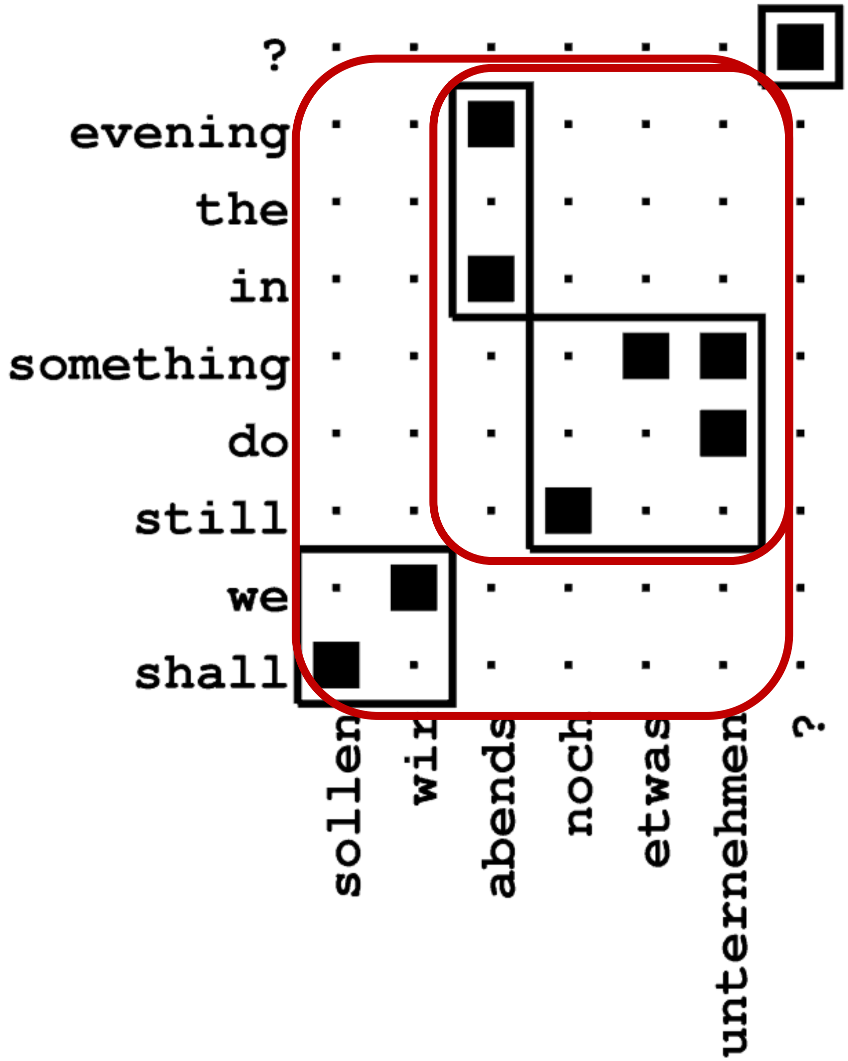
Statistical (Machine Learning) I.1



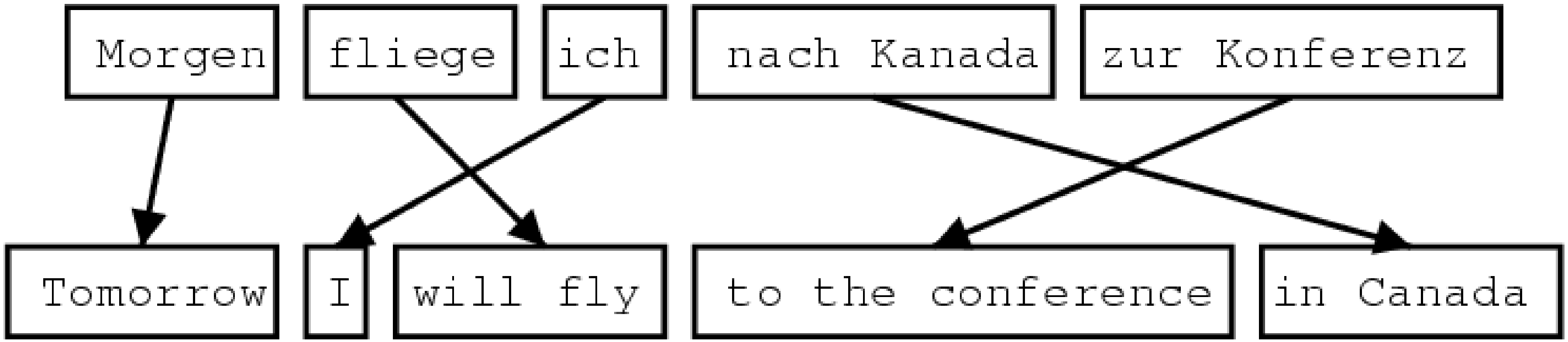
Statistical (Machine Learning) I.1



Statistical (Machine Learning) I.1



Statistical (Machine Learning) I.1



Statistical (Machine Learning) I.1

$$e_{\text{best}} = \text{argmax}_e \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i)\ d(start_i - end_{i-1} - 1)\ \prod_{i=1}^{|\mathbf{e}|} p_{LM}(e_i|e_1...e_{i-1})$$

$$e_{\text{best}} = \text{argmax}_e \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i)^{\lambda_\phi}\ d(start_i - end_{i-1} - 1)^{\lambda_d}\ \prod_{i=1}^{|\mathbf{e}|} p_{LM}(e_i|e_1...e_{i-1})^{\lambda_{LM}}$$

$$p(x) = \exp \sum_{i=1}^{n} \lambda_i h_i(x)$$

- number of feature function $n = 3$
- random variable $x = (e, f, start, end)$
- feature function $h_1 = \log \phi$
- feature function $h_2 = \log d$
- feature function $h_3 = \log p_{\text{LM}}$

Weighted Model as Log-Linear Model

$$p(e,a|f) = \exp(\lambda_\phi \sum_{i=1}^{I} \log \phi(\bar{f}_i|\bar{e}_i) +$$

$$\lambda_d \sum_{i=1}^{I} \log d(a_i - b_{i-1} - 1) +$$

$$\lambda_{LM} \sum_{i=1}^{|\mathbf{e}|} \log p_{LM}(e_i|e_1 \ldots e_{i-1}))$$

Statistical (Machine Learning) I.1

- A modern PB-SMT system can have many components:
  - Phrase translation model (for each translation direction)
  - Reordering model
  - Language model
  - Lexical translation models (for each direction)
  - Length model
  - Segmentation model
  - Many more …
- 5 – 15 components …

$$p(x) = \exp \sum_{i=1}^{n} \lambda_i h_i(x)$$

Statistical (Machine Learning) I.1

- What's cool about PB-SMT:

- One of the most successful LTs to date
- Brought MT into our daily lives
- And into professional translation workflows: post-editing MT output
- Language agnostic, all you need is training data
- Works well for many language pairs
- …

$$p(x) = \exp \sum_{i=1}^{n} \lambda_i h_i(x)$$

Statistical (Machine Learning) I.1

- What's not so cool about PB-SMT:

- Works better for some language pairs than others
- Morphologically rich languages, OOVs, …
- Massive independence assumptions
- Makes local decisions
- Reordering pretty bad
- Based on very heterogeneous technology stacks
- Components individually estimated
- Not jointly optimized against same loss function: translation quality …!

$$p(x) = \exp \sum_{i=1}^{n} \lambda_i h_i(x)$$

Statistical (Machine Learning) I.1

- heterogeneous technology stacks: estimated independently, sometimes using heuristics and different data
  - Alignment: expectation maximization (EM) and HMMs (GIZA++)
  - Phrase extraction and scoring: based on alignment, heuristics (grow-diag-final …), MLE scoring
  - Lexical translation probabilities: alignment and MLE scoring
  - Re-ordering based on alignment positions and MLE
  - LM: count based (different ways of smoothing and back-off), often using supplementary data
  - Top level log-linear combination of feature functions setting weights, doesn't go into component models …
  - Heuristics based search …

$$p(x) = \exp \sum_{i=1}^{n} \lambda_i h_i(x)$$

Statistical (Machine Learning) I.1

- heterogeneous technology stacks: estimated independently, sometimes using heuristics and different data
- Top level log-linear combination of feature functions setting feature weights

- Individual feature functions estimated independently, sometimes using heuristics and different data, not optimized by same objective function

- Only high level feature weight settings, does not go inside components
- No guarantee that this is in any way optimal …
- Works surprisingly/amazingly well in practice ☺

$$p(x) = \exp \sum_{i=1}^{n} \lambda_i h_i(x)$$

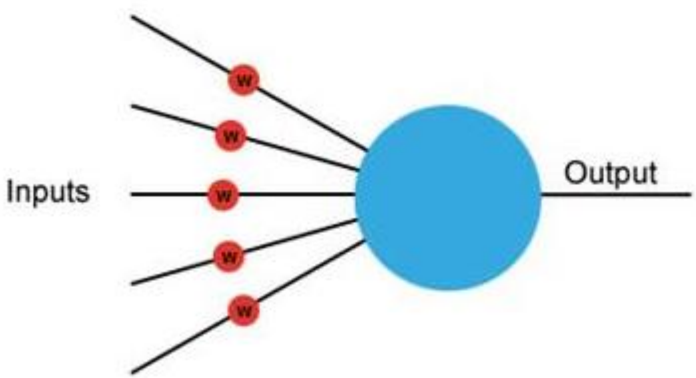
Statistical (Machine Learning) I.1

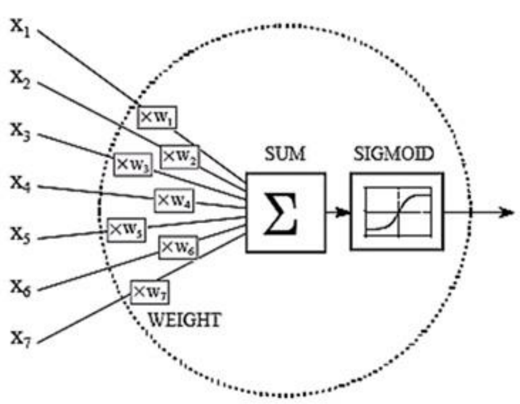
- Not only MT …
- IE Information Extraction:
- RE based tokenizer
- CRF based POS tagger
- FST based morphology
- Max-entropy NER
- RE based chunker
- Transition based dependency parser with SVM
- Perceptron-based semantic role labeler
- Clustering based relation classifier
- Graph-algorithm based NE disambiguator and linker
- Sentiment analysis component based on hand crafted sentiment lexica
- Alignment based textual entailment component
- …
- Similar for dialogue manager, QA and other complex NLP systems

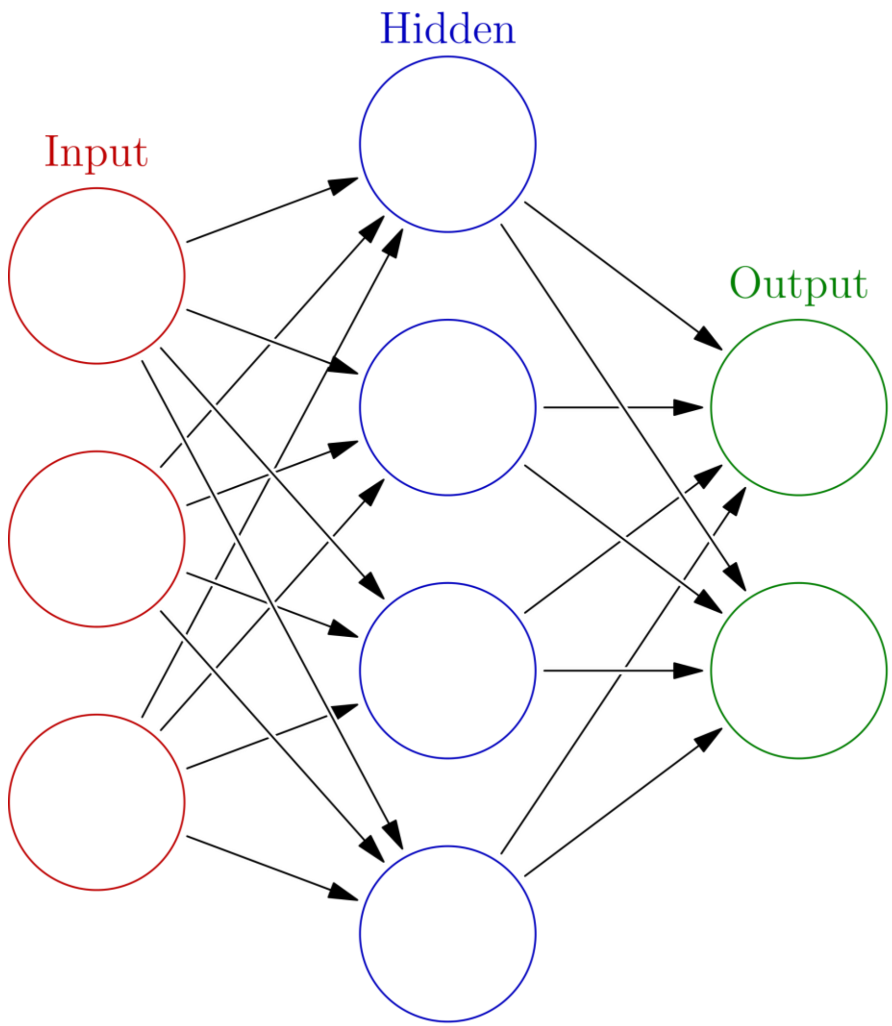
Statistical (Machine Learning) I.1

- heterogeneous technology stacks
- Motivation: best for each sub-task, compelling motivation (at first sight)
- Can have severe disadvantages:
- Difficult to
  - Maintain
  - Adapt
  - Scale
  - Requires substantial interface and standardization overhead
- Worst: almost impossible to jointly optimize end-to-end
- No end-to-end training
- No guarantee that this is in any way optimal …

Statistical (Machine Learning) II







Statistical (Machine Learning) II

- A radically different approach

- Based on a “single” simple computing device
- Artificial neural networks ANNs
- Can be scaled, stacked, cross-/inter-connected = deep neural networks DNNs
- Supports end-to-end training
- Often text-to-text end-to-end
- Avoids extensive feature engineering (can learn some itself)
- Mix supervised with non-supervised approaches
- All components are jointly optimized against same (or multiple) objective(s)

- Base technology, judiciously add external knowledge

Statistical (Machine Learning) II

- Another radically different approach

- What is the atom in linguistic computation?
- The word?
- Sub-word units: morphs?

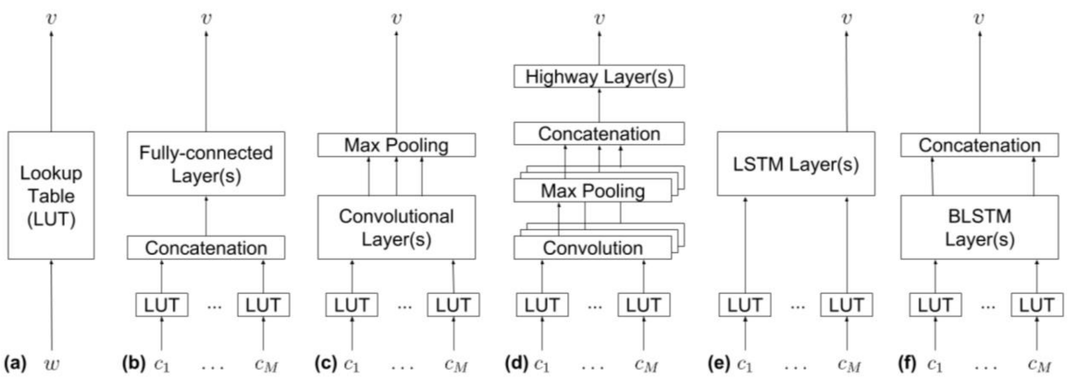
- Why not just characters?

- DFKI neural approaches to morphology and machine translation:

- Character based neural morphological tagging
- Character based neural machine translation

Statistical (Machine Learning) II

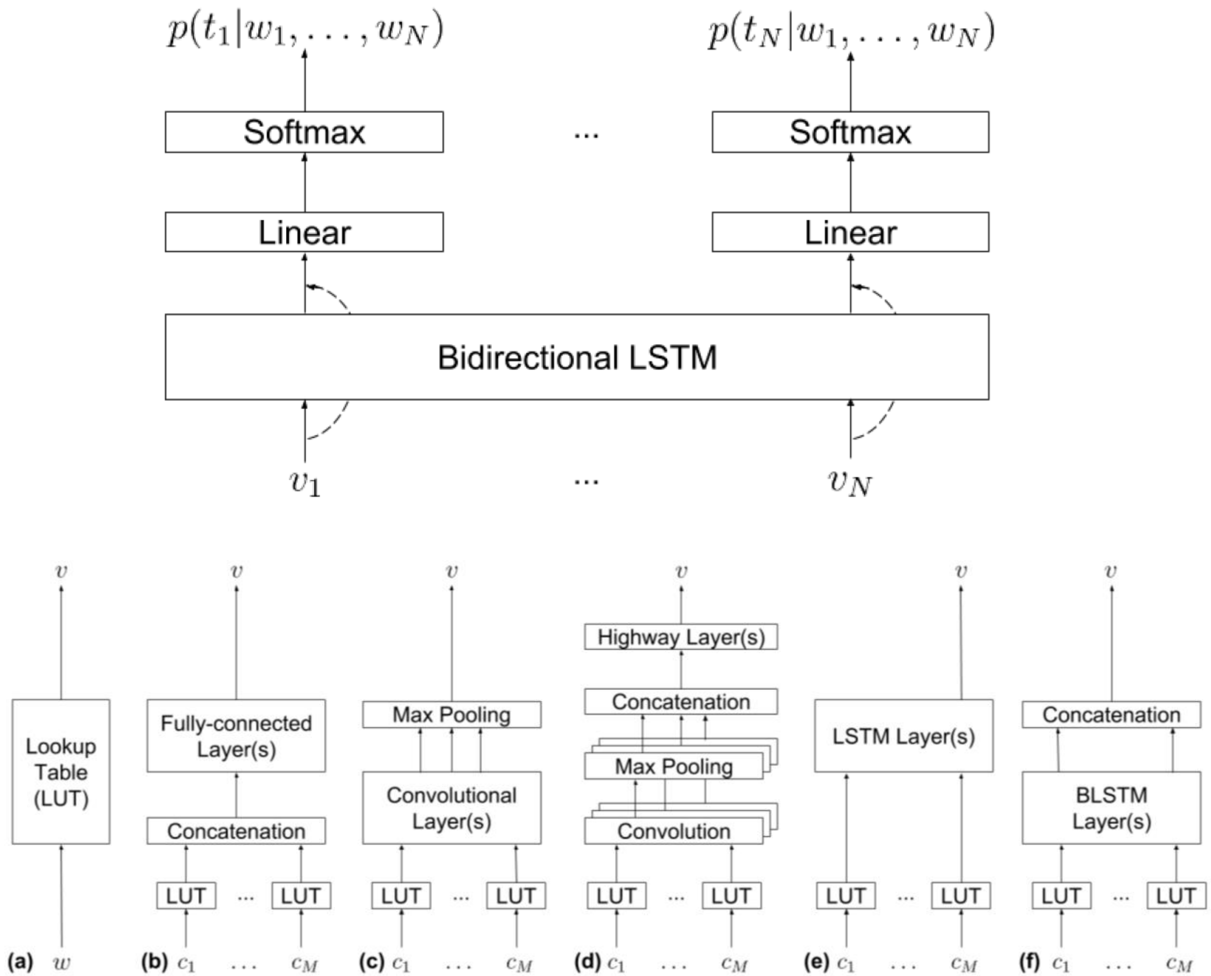
- But:
- Many different types of NNs



- Feed-forward
- Convolutional
- Recurrent (with gates or LSTMs)
- With/without attention mechanisms …

- Which ones to use for what?
- Linguistically motivated subnetworks?

DFKI Character based Morphological Tagging – Georg Heigold



DFKI Character based Morphological Tagging – Georg Heigold

	Amount of data		Number of POSMORPH tags	MarMoT		DFKI	
	train	test		dev	test	dev	test
Arabic, UD	256k	32k	320	90.80	90.87	93.19	93.63
Czech, PDT	691k	93k	1811		92.54		95.64
UD	1175k	174k	1418	93.53	93.03	96.65	96.32
Finnish, UD	163k	9k	1593	91.65	92.21	92.49	93.49
German, TIGER	760k	92k	681		88.58		93.23
Hindi, UD	281k	35k	922	88.43	88.56	90.96	91.11
Korean, SPMRL	296k	28k	1976	81.60	81.40	86.90	86.30
Romanian, UD	109k	18k	444	91.72	92.36	93.72	93.75
Russian-SynTagRus, UD	815k	108k	434	93.69	93.92	96.33	96.45
Turkish, UD	42k	9k	987	83.16	82.72	87.20	86.82

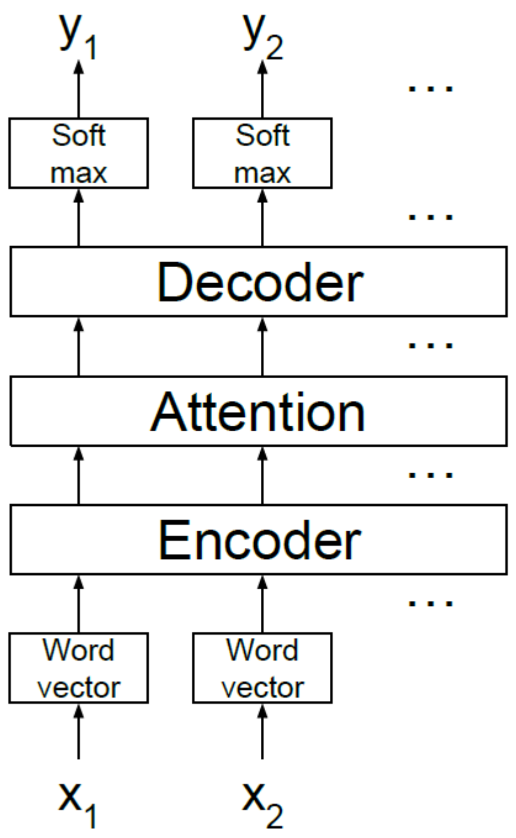
LT as end-to-end text-to-text NN: NMT

- DFKI NMT
- Character based
- Attention mechanism

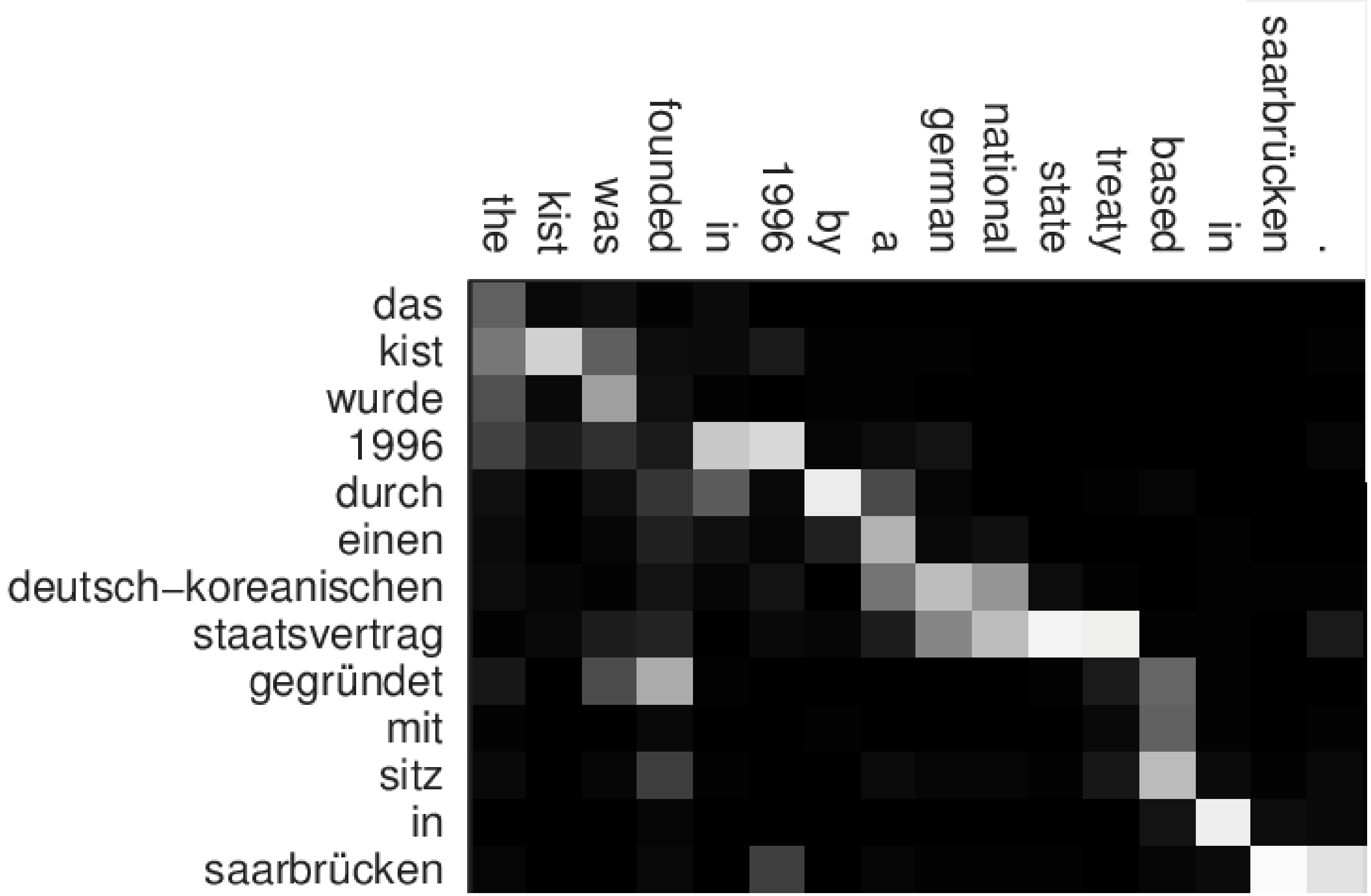
BLEU	WMT’16
PB-SMT	30.0
char-NMT	29.1 (single) 31.3 (ensemble)

Performance?

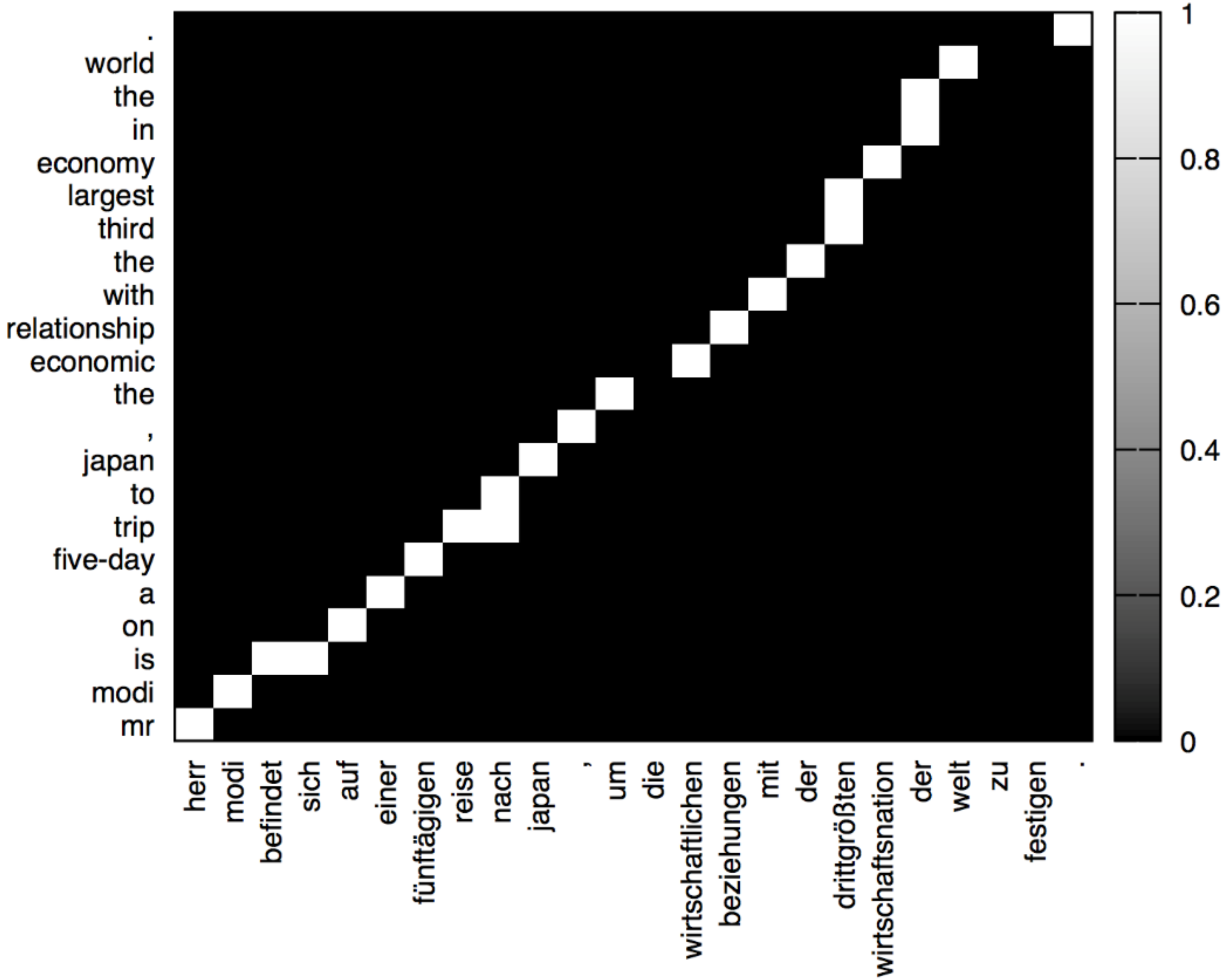
End-to-end



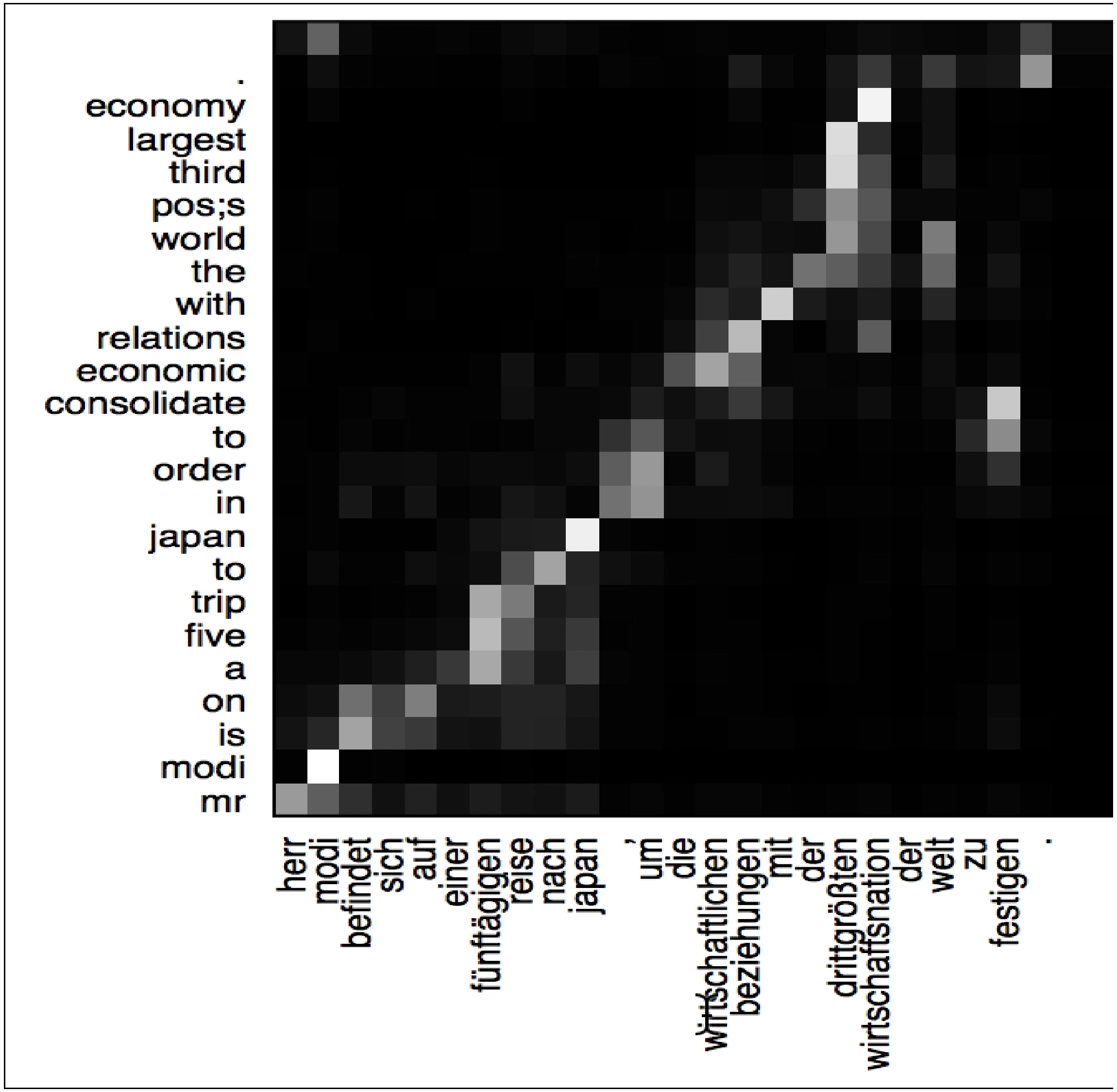
Character based Neural MT – Georg Heigold



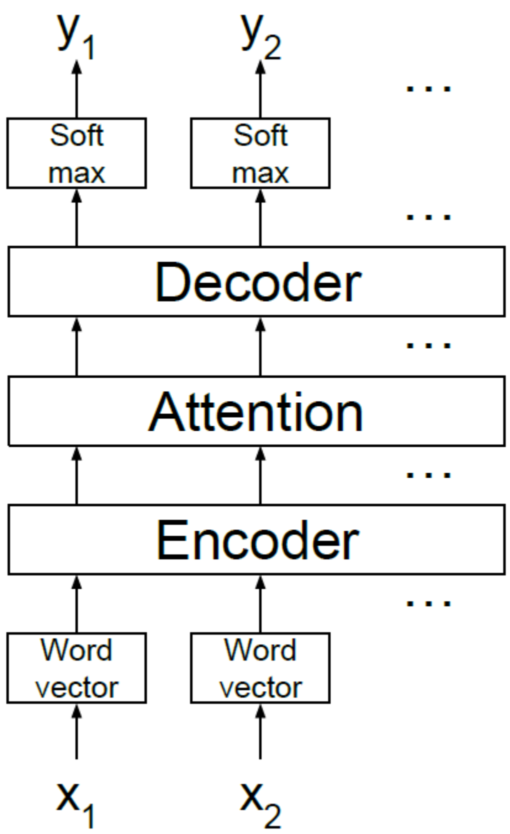
SMT: Hard Alignment



Can the Net Explain Itself?



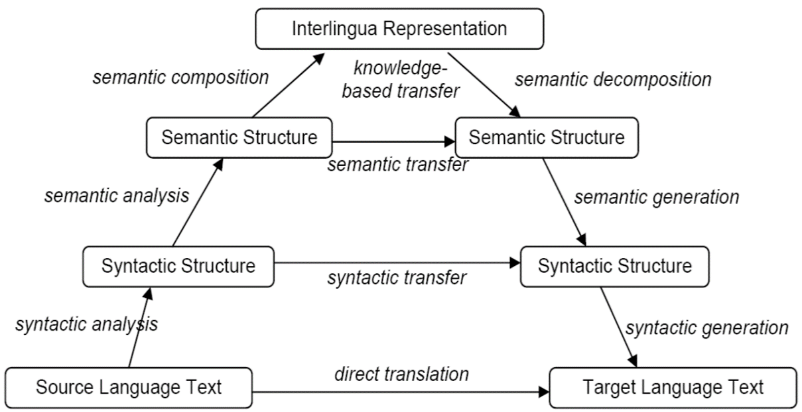
Character based Neural MT – Georg Heigold

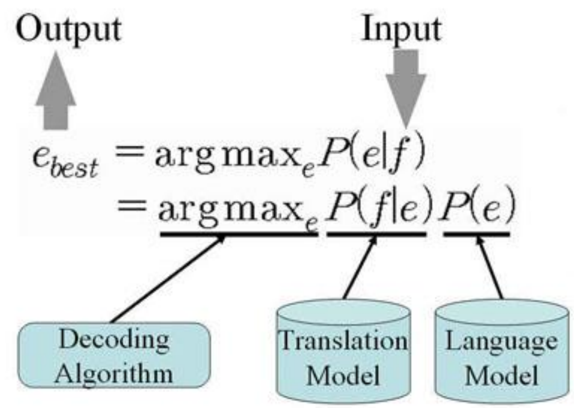


Turkish – English WMT 2016

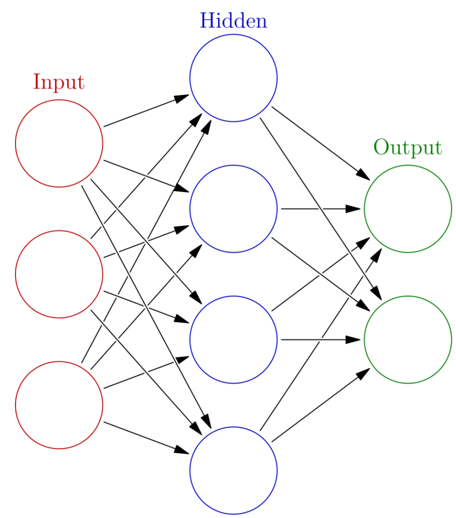
System	BLEU
PB-SMT[1]	14.8
system combination[2]	15.7
char-NMT	15.4
ensemble	16.7

The Journey





$$p(x) = \exp \sum_{i=1}^{n} \lambda_i h_i(x)$$



Take home messages?

- Linguistics, computational linguistics, HLT and NLP are “young” sciences
- Subject to “paradigm” shifts

- Move away from complex heterogeneous (and often incompatible) technology stacks to chains based on “uniform” base technology
- End-to-end, joint training against same objective(s)
- Lower barrier of entrance …?


TIME FOR QUESTIONS