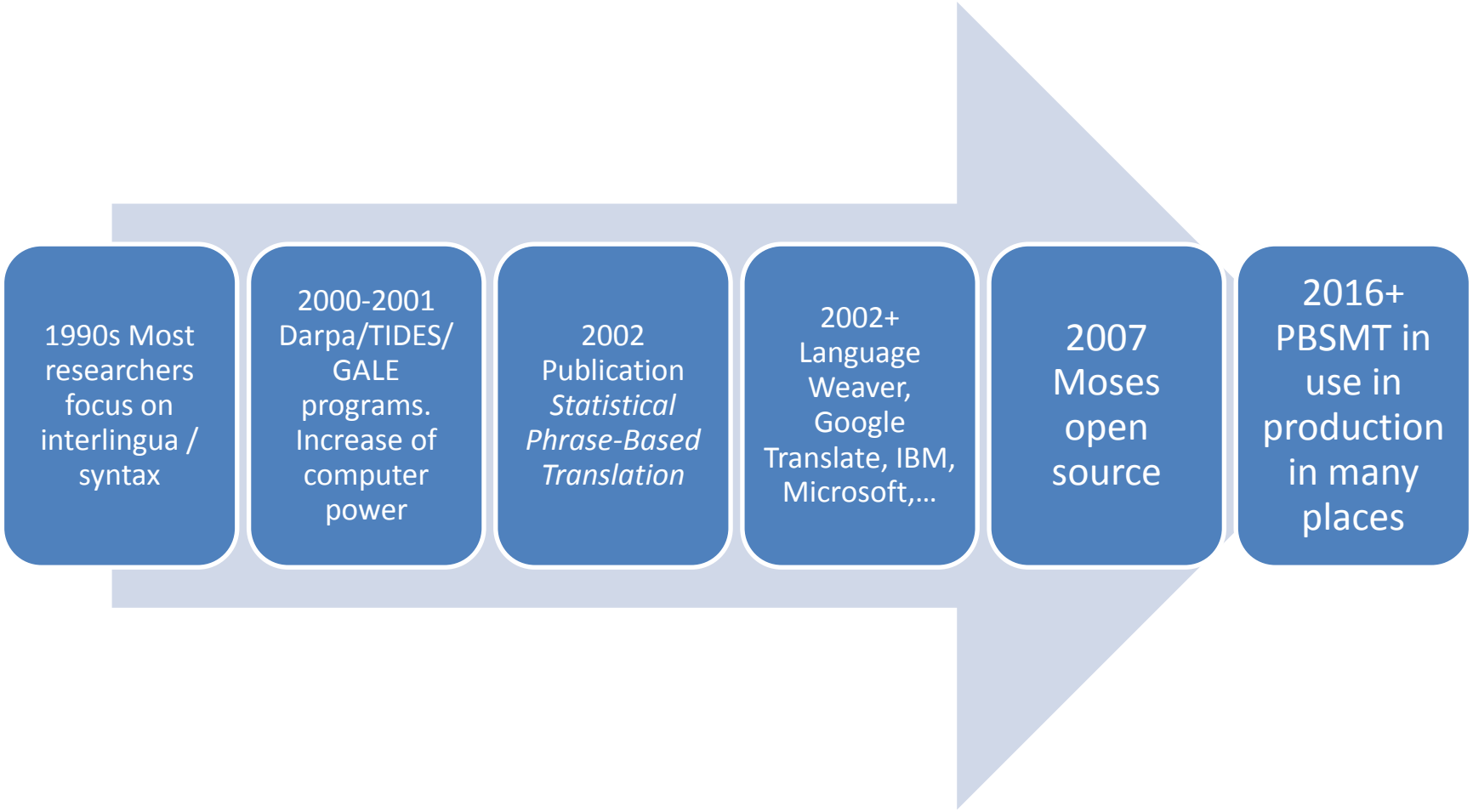
Lecture 4.2: Phrase-based SMT

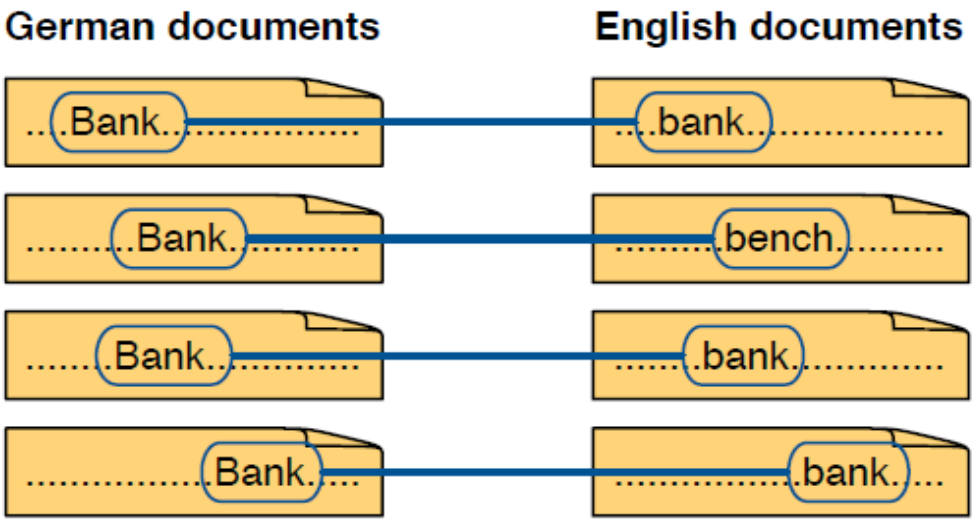
Phrase based SMT: history



Introduction to Statistical Machine Translation

SMT : how does it work ?

- Learning from data (sentence-aligned translated texts)



- New machine translation systems can be built automatically

Courtesy of Philipp Koehn, Professor at The Johns Hopkins University

Extract lexical translations

- Collect statistics
- Estimate a (lexical) probability distribution
- Alignments
- Learning lexical translation models

Courtesy of Philipp Koehn, Professor at The Johns Hopkins University

Word-based MT

- One English word must be translate to one foreign word
- It just does not work!

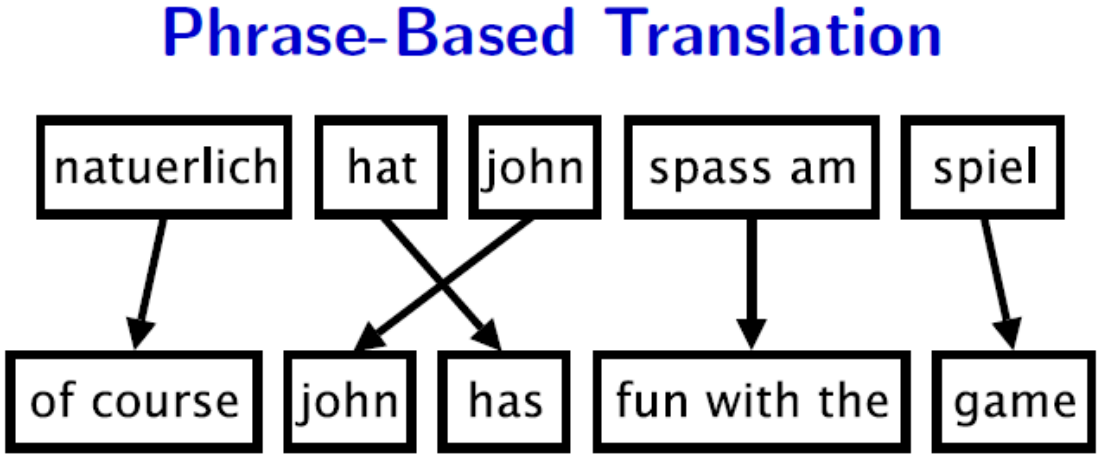
=> Phrase translation is needed

  - [en] I am => [es] soy
  - [en] the day before yesterday => [hi] परसों (*parson)*
  - [en] headache => [fr] mal de tête

When it all started

- *Statistical Phrase-Based Translation (2002) Philipp Koehn , Franz Josef Och , Daniel Marcu*
  - *Philipp Hoehn built Moses*
  - *Franz Josef Och built Google Translate*
  - *Daniel Marcu created* LanguageWeaver company (now SDL Language Weaver)

Phrase-based translation in a nutshell



- Foreign input is segmented in phrases
  - any sequence of words, not necessarily linguistically motivated
- Each phrase is translated into English
- Phrases are reordered

Courtesy of Philipp Koehn, Professor at The Johns Hopkins University

Theory behind PBSMT

- Translating from language *f* to language *e* (***f***rench - ***f***oreign into ***e***nglish)
- Find the best English sentence out of a French one
- $\tilde{e} = \arg\max_{e \in e^*} p(e|f)$

English translation is the translation having the highest probability "English knowing French"

- $p(e|f) = \frac{p(f|e)p(e)}{p(f)}$ (Bayes' rule)
- $\tilde{e} = \arg\max_{e \in e^*} p(f|e)p(e)$

$\Rightarrow$Replace it by: finding the best English sentence knowing the probability of French to be translated as such combined with "best fluent English"

- $\tilde{e} = \arg\max_{e \in e^*} p(f|e)p(e)$
- $p(f|e)$ → Phrase table (phrase translation probabilities)
- $p(e)$→Language model (probability of being “good” English)

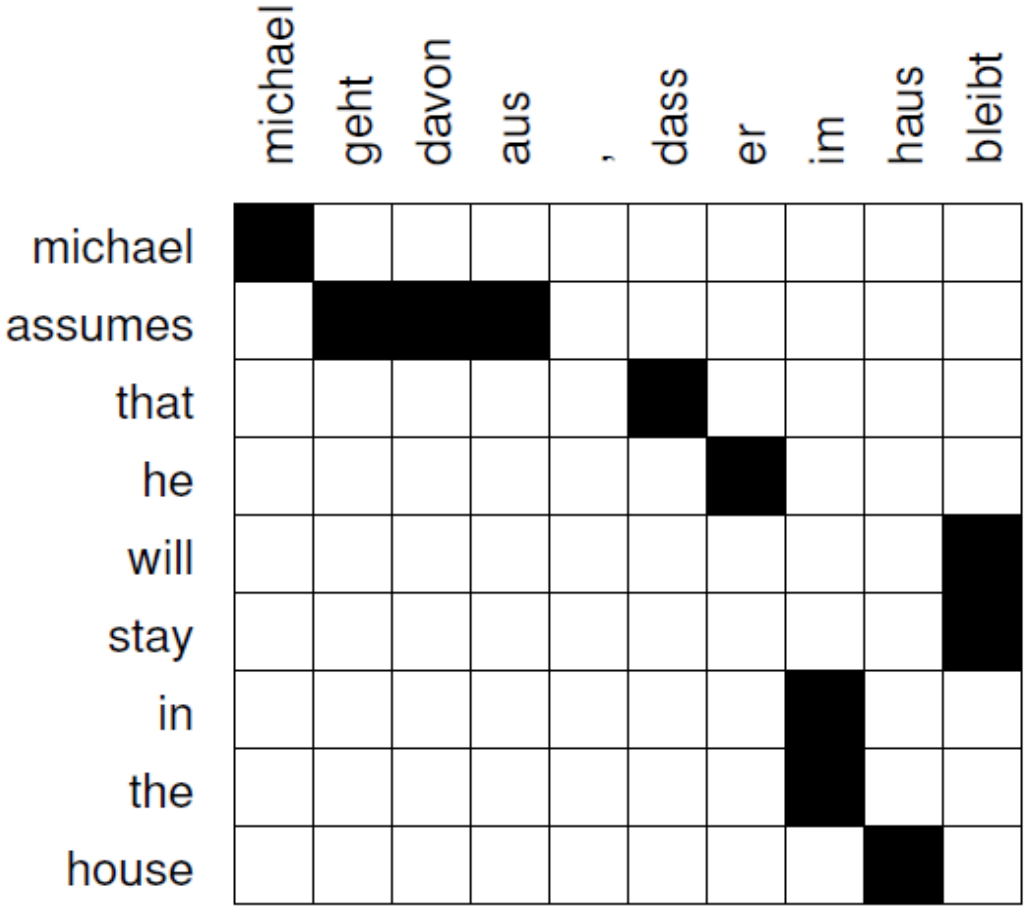
Lexical translation (word)

- How to translate a word → look up in dictionary

  Haus ➔ house, building, home, household, shell

- Multiple translations
  - some more frequent than others
  - for instance: house, and building most common
  - special cases: Haus of a snail is its shell

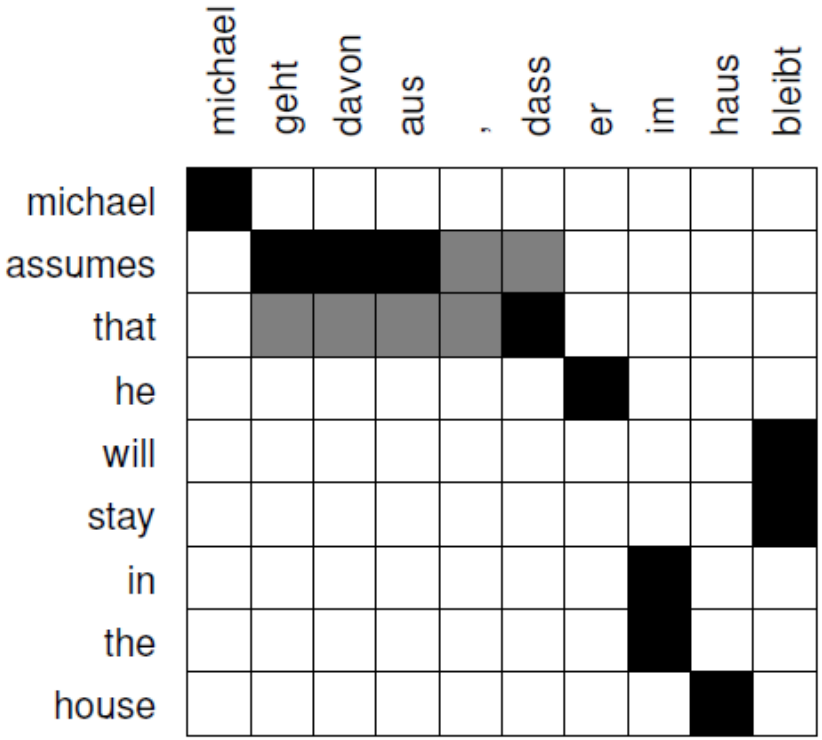
From aligned words to aligned phrases



Preparing the data : word alignment

Courtesy of Philipp Koehn, Professor at The Johns Hopkins University

Extracting Phrases from Data



Given a word alignment: extract phrase pairs, estimate probabilities

Courtesy of Philipp Koehn, Professor at The Johns Hopkins University

From parallel phrases to probabilities

Phrase Translation Table

- Phrase Translations for "den Vorschlag"

English	$\phi(e\|f)$	English	$\phi(e\|f)$
the proposal	0.6227	the suggestions	0.0114
's proposal	0.1068	the proposed	0.0114
a proposal	0.0341	the motion	0.0091
the idea	0.0250	the idea of	0.0091
this proposal	0.0227	the proposal ,	0.0068
proposal	0.0205	its proposal	0.0068
of the proposal	0.0159	it	0.0068
the proposals	0.0159	...	...

Courtesy of Philipp Koehn, Professor at The Johns Hopkins University

Language model

Language Model

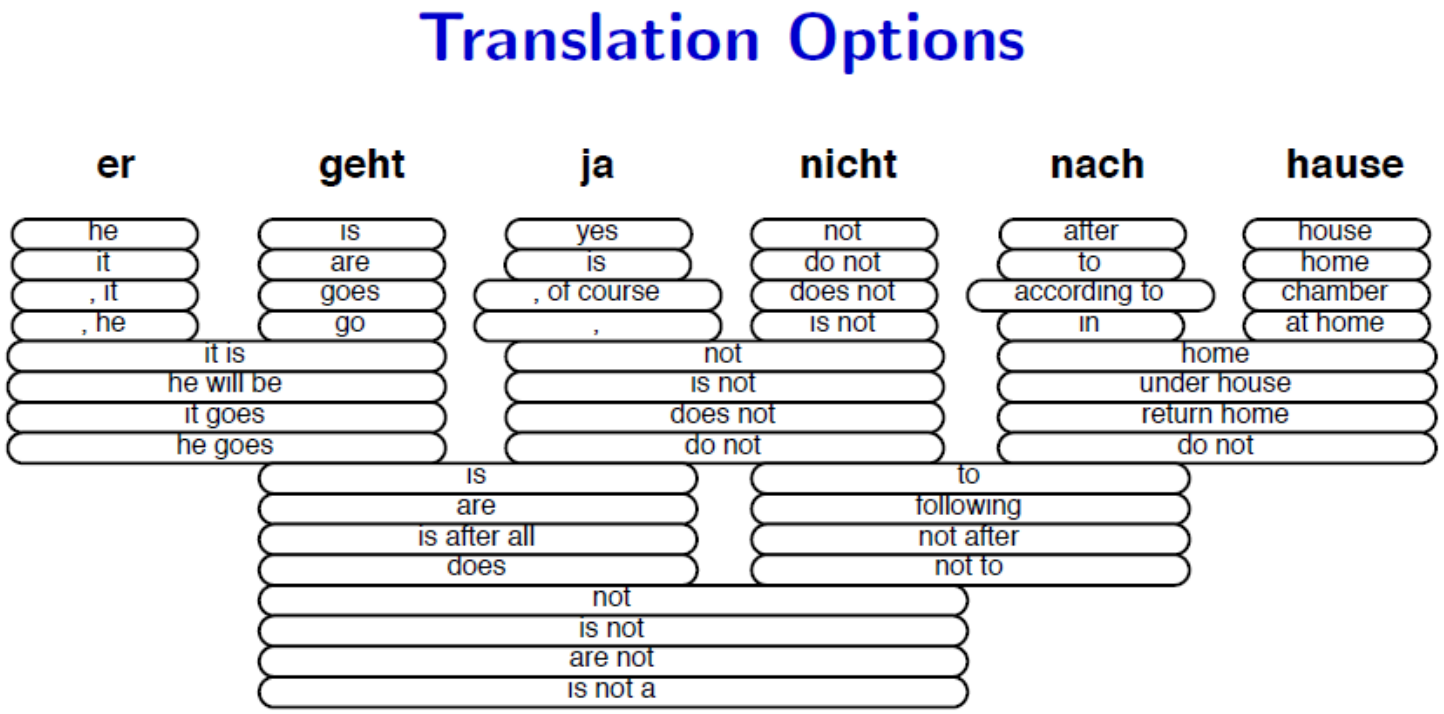
- **Language models** answer the question: How likely is a string of English words good English?
  - the house is big → good
  - the house is xxl → worse
  - house big is the → bad
- Given: English words $W = w_1, w_2, w_3, ..., w_n$ — *what is* $p(W)$?
- Limited history: only previous k words matter (here: k=2)

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

- Models trained on large amounts of monolingual text (billions of words)

Courtesy of Philipp Koehn, Professor at The Johns Hopkins University

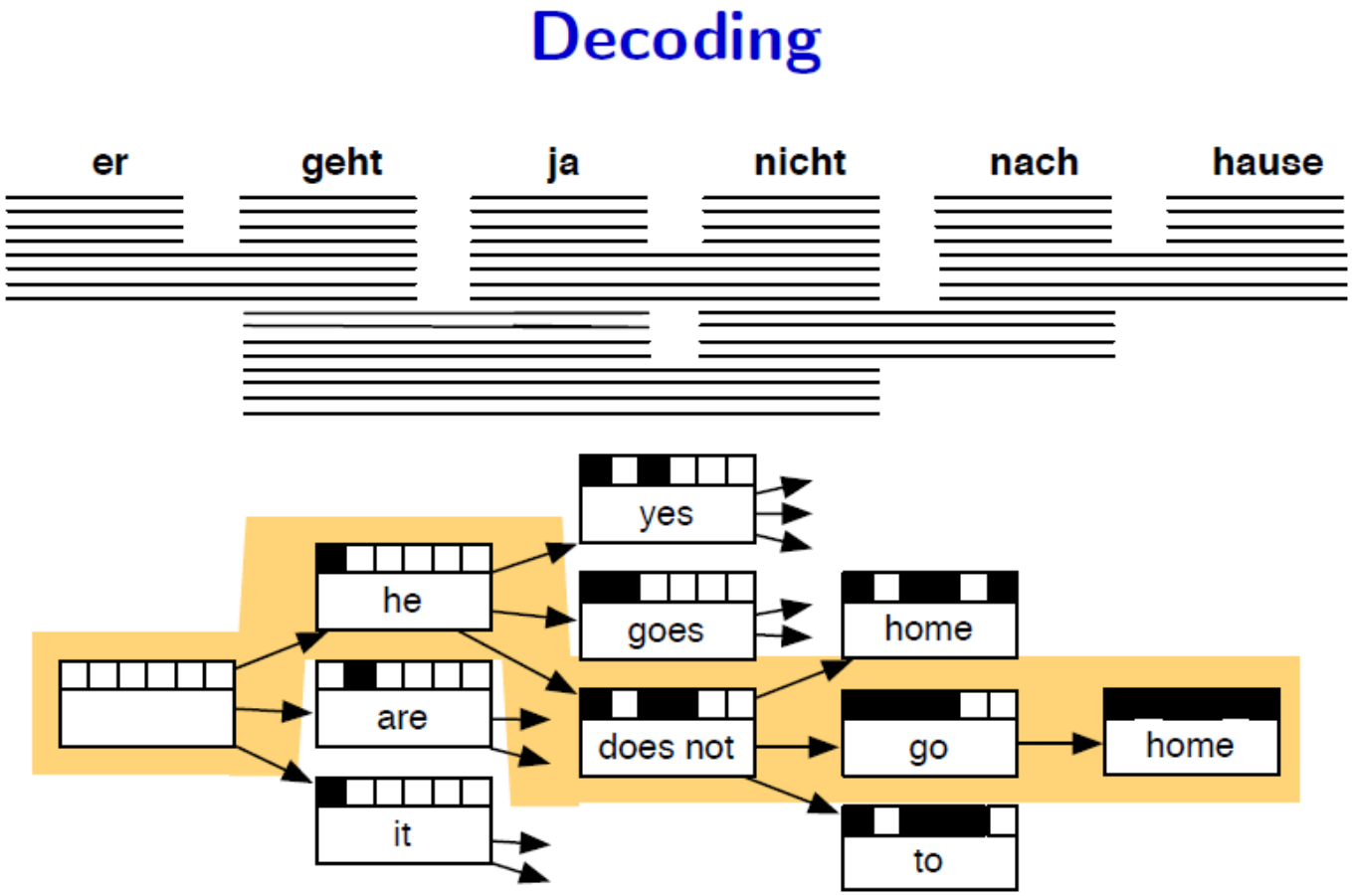
SMT : how does it work ?



- Task: find the right output phrases, put them in the right order

Courtesy of Philipp Koehn, Professor at The Johns Hopkins University

SMT : how does it work ?



Courtesy of Philipp Koehn, Professor at The Johns Hopkins University

Different open-source software

- Moses: SMT (PBSMT+tree-based), C++ - Perl/Unix
- Joshua: SMT (tree-based) Java
- Cdec: SMT (tree-based) C++/Unix
- Jane: (PBSMT+tree-based), **RWTH Aachen University, C++/Unix**
- Phrasal: PBSMT, Standford, Java
- Apertium: rule-based machine translation platform, C++/Windows-Unix
- Nematus: Neural MT open source, Python

See http://fosmt.org/, https://en.wikipedia.org/wiki/Comparison_of_machine_translation_applications