Statistical Machine Translation
Part III – Many-to-Many Alignments

Alexander Fraser
CIS, LMU München

2015.11.03 WSD and MT

New MT Seminar: Neural MT

- Starting this Thursday at 2pm s.t., there will be a seminar on "Neural Machine Translation"
- The goal of the seminar is to understand how deep learning is being used to do machine translation end-to-end
  - This deep learning approach is trained only on sentence pairs (not word-aligned sentence pairs)
- The paper to read this week is a classic paper on neural language models which is very accessible
- Please let me know after class if you are interested

Schein in this course

- Referat (next slides)
- Hausarbeit
  - 6 pages (an essay/prose version of the material in the slides), due 3 weeks after the Referat

Referat Topics

- We should have about 3 literature review topics and 6 projects
  - Projects will hold a Referat which is a mix of literature review/motivation and own work

Referat Topics - II

- Literature Review topics
  - Dictionary-based Word Sense Disambiguation
  - Supervised Word Sense Disambiguation
  - Unsupervised Word Sense Disambiguation

- Project 1: Supervised WSD
  - Download a supervised training corpus
  - Pick a small subset of words to work on (probably common nouns or verbs)
  - Hold out some correct answers
  - Use a classifier to predict the sense given the context

- Project 2: Cross-Lingual Lexical Substitution
  - Cross-lingual lexical substitution is a translation task where you given a full source sentence, a particular (ambiguous) word, and you should pick the correct translation
  - Choose a language pair (probably EN-DE or DE-EN)
  - Download a word aligned corpus from OPUS
  - Pick some ambiguous source words to work on (probably common nouns)
  - Use a classifier to predict the translation given the context

- Project 3: Predicting case given a sequence of German lemmas
  - Given a German text, run RFTagger (Schmid and Laws) to obtain rich part-of-speech tags
  - Run TreeTagger to obtain lemmas
  - Pick some lemmas which frequently occur in various grammatical cases
  - Build a classifier to predict the correct case, given the sequence of German lemmas as context
  - (see also my EACL 2012 paper)

- Project 4: Wikification of ambiguous entities
  - Find several disambiguation pages on Wikipedia which disambiguate common nouns, e.g. http://en.wikipedia.org/wiki/Cabinet
  - Download texts from the web containing these nouns
  - Annotate the correct disambiguation (i.e., correct Wikipedia page, e.g.

    http://en.wikipedia.org/wiki/Cabinet_(furniture) or (government)
  - Build a classifier to predict the correct disambiguation
    - You can use the unambiguous Wikipedia pages themselves as your only training data, or as additional training data if you annotate enough text

- Project 5: Moses DE-EN
  - Download and install the open-source Moses SMT system (you may want to use the virtual machine distribution)
  - Download an English/German parallel corpus, e.g., from Opus or statmt.org
  - Build a Moses SMT system for DE to EN
  - Test your system on data from Wikipedia or similar (be sure to check that the English Wikipedia does not contain this content!)
  - Perform an overall error analysis of translation quality
  - Pick some polysemous DE words and show whether Moses can correctly select all of the senses

- Project 6: Moses EN-DE
  - Download and install the open-source Moses SMT system (you may want to use the virtual machine distribution)
  - Download an English/German parallel corpus, e.g., from Opus or statmt.org
  - Build a Moses SMT system for EN to DE
  - Test your system on English data from the UN multilingual corpus
  - Perform an overall error analysis of translation quality
  - Pick some polysemous EN words and show whether Moses can correctly select all of the senses

- Project 7: Google Translate DE-EN (Compounds)
  - Make a short list of DE compounds where the head word is polysemous
  - Find text containing these compounds
  - Find also text containing the simplex head words you have selected (in all of their senses)
  - Run this text through Google Translate DE-EN, be sure to carefully save the results and record when you ran the translation
  - Perform a careful analysis of Google Translate's performance in translating these texts
    - How well does Google Translate perform on the different senses of the simplex head words?
    - How well does it translate the compounds? Is there a correlation with the simplex performance?)
    - Does Google Translate use specialized compound handling (as far as you can tell)? How does it generalize? Does it overgeneralize?

- Project 8: Google Translate RU-DE (Pivoting)
  - Select a Russian text for which there is unlikely to be parallel English or German parallel data available (i.e., don't take a classic novel or news!). Suggestion: Wikipedia articles (on topics with no English or German)
  - Run this text through Google Translate RU-DE
    - Carefully save the results and record dates for all translations
  - Explicit pivot
    - Run this text through Google Translate RU-EN
    - Post-edit the EN output to fix any obvious major errors
    - Run the original EN output and the post-edited EN through Google EN-DE
  - Perform a careful analysis of Google Translate's performance in translating these texts
    - Is Google Translate "pivoting" when translating from RU-DE directly?
    - What are common problems in each translation?
    - Is there useful information which is easier to get from the original DE input than from the intermediate EN?
    - Does post-editing the EN help translation quality? By how much?

- A last suggestion for topics involving running translations (through Google Translate)
  - Sentence split your data manually
  - Put a blank line between each sentence
  - Then you can easily figure out which input sentence corresponds to which output sentence

- We are now done with topics (more on Referat/Hausarbeit next)
  - I am also open to your own topic suggestions (should have some similarity to one of these projects)

Referat

- Tentatively (MAY CHANGE!):
  - 25 minutes plus about 15 minutes for discussion
- Start with what the problem is, and why it is interesting to solve it (motivation!)
  - It is often useful to present an example and refer to it several times
- Then go into the details
- If appropriate for your topic, do an analysis
  - Don't forget to address the disadvantages of the approach as well as the advantages
  - Be aware that advantages tend to be what the original authors focused on!
- **List references and recommend further reading**
- **Have a conclusion slide!**

Languages

- I recommend:
- If you do the slides in English, then presentation in English (and Hausarbeit in English)
- If you do the slides in German, then presentation in German (and Hausarbeit in German)
- Additional option (not recommended):
  - English slides, German presentation, English Hausarbeit
  - Very poor idea for non-native speakers of German (you will get tired by the end of the discussion because English and German interfere)

References I

- Please use a standard bibliographic format for your references
  - This includes authors, date, title, venue, like this:
  - (Academic Journal)
  - Alexander Fraser, Helmut Schmid, Richard Farkas, Renjing Wang, Hinrich Schuetze (2013). Knowledge Sources for Constituent Parsing of German, a Morphologically Rich and Less-Configurational Language. *Computational Linguistics*, 39(1), pages 57-85.
  - (Academic Conference)
  - Alexander Fraser, Marion Weller, Aoife Cahill, Fabienne Cap (2012). Modeling Inflection and Word-Formation in SMT. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL),* pages 664-674, Avignon, France, April.

References II

- In the Hausarbeit, use *inline* citations:
  - "As shown by Fraser et al. (2012), the moon does not consist of cheese"
  - "We build upon previous work (Fraser and Marcu 2007; Fraser et al. 2012) by ..."
  - Sometimes it is also appropriate to include a page number (and you *must* include a page number for a quote or graphic)
- Please do not use numbered citations like:
  - "As shown by [1], ..."
  - Numbered citations are useful to save space, otherwise quite annoying

References III

- If you use graphics (or quotes) from a research paper, MAKE SURE THESE ARE CITED ON THE *SAME SLIDE* IN YOUR PRESENTATION!
  - These should be cited in the Hausarbeit in the caption of the graphic
  - Please include a page number so I can find the graphic quickly
- Web pages should also use a standard bibliographic format, particularly including the date when they were downloaded
- I am not allowing Wikipedia as a primary source
  - After looking into it, I no longer believe that Wikipedia is reliable, for most articles there is simply not enough review (mistakes, PR agencies trying to sell particular ideas anonymously, etc.)
- You also cannot use student work (not PhD peer-reviewed) as a primary source

- Any questions?

- Back to SMT...
- (Finish up slides from last time)
- Last time, we discussed Model 1 and Expectation Maximization
- Today we will discuss getting useful alignments for translation and a translation model

IBM Model 1

- *Generative model*: break up translation process into smaller steps
  - **IBM Model 1** only uses *lexical translation*

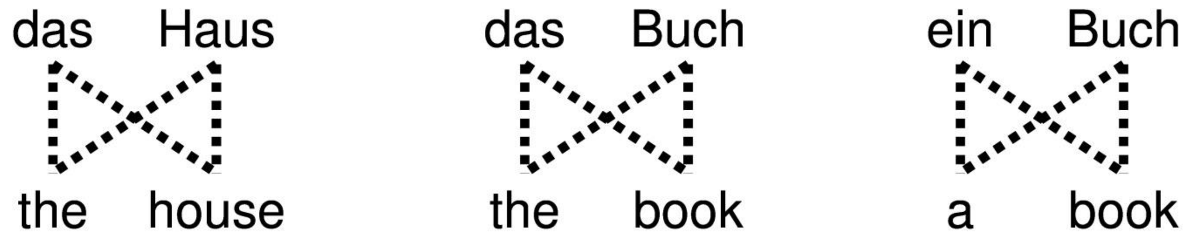
- Translation probability
  - for a foreign sentence $\mathbf{f} = (f_1, ..., f_{l_f})$ of length l_f
  - to an English sentence $\mathbf{e} = (e_1, ..., e_{l_e})$ of length l_e
  - with an alignment of each English word e_j to a foreign word f_i according to the alignment function $a : j \rightarrow i$

$$p(\mathbf{e}, a|\mathbf{f}) = \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} t(e_j | f_{a(j)})$$

  - parameter ϵ is a *normalization constant*

Slide from Koehn 2008

Convergence



e	f	initial	1st it.	2nd it.	3rd it.	...	final
the	das	0.25	0.5	0.6364	0.7479	...	1
book	das	0.25	0.25	0.1818	0.1208	...	0
house	das	0.25	0.25	0.1818	0.1313	...	0
the	buch	0.25	0.25	0.1818	0.1208	...	0
book	buch	0.25	0.5	0.6364	0.7479	...	1
a	buch	0.25	0.25	0.1818	0.1313	...	0
book	ein	0.25	0.5	0.4286	0.3466	...	0
a	ein	0.25	0.5	0.5714	0.6534	...	1
the	haus	0.25	0.5	0.4286	0.3466	...	0
house	haus	0.25	0.5	0.5714	0.6534	...	1

Slide from Koehn 2009

Higher IBM Models

IBM Model 1	lexical translation
IBM Model 2	adds absolute reordering model
IBM Model 3	adds fertility model
IBM Model 4	relative reordering model
IBM Model 5	fixes deficiency

- Only IBM Model 1 has global maximum
  - training of a higher IBM model builds on previous model
- Compuationally biggest change in Model 3
  - trick to simplify estimation does not work anymore
  - → exhaustive count collection becomes computationally too expensive
  - sampling over high probability alignments is used instead

HMM Model

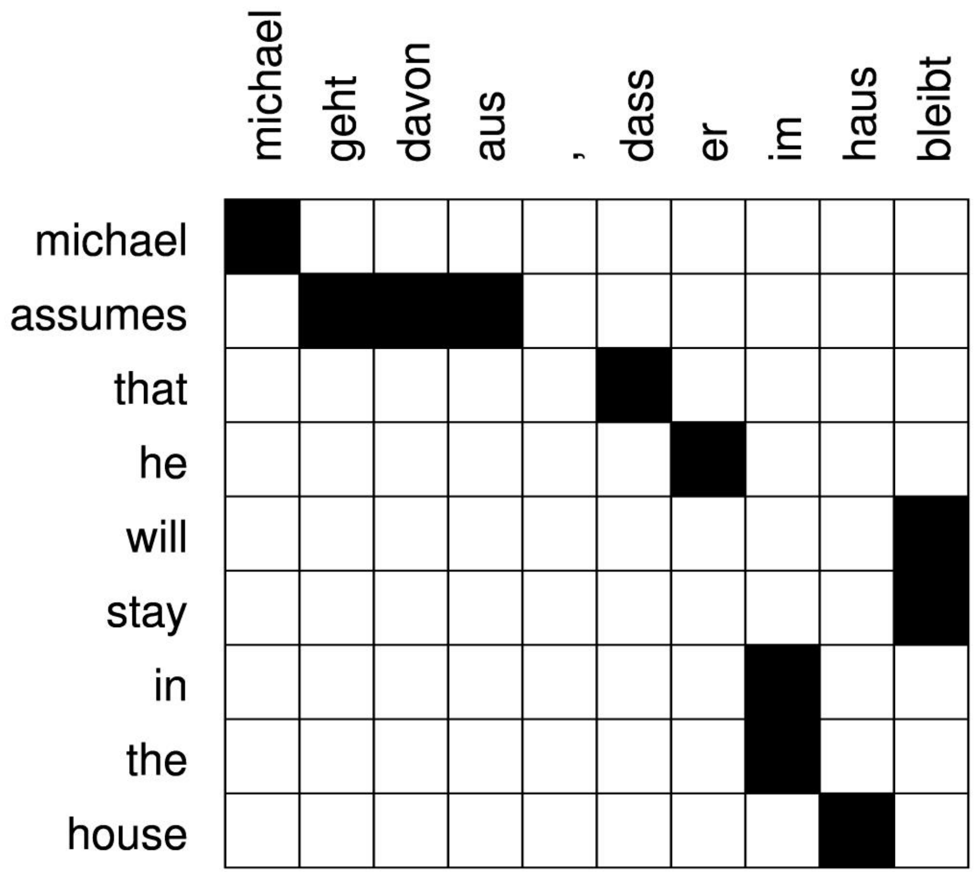
- Model 4 requires local search (making small changes to an initial alignment and rescoring)
- Another popular model is the HMM model, which is similar to Model 2 except that it uses relative alignment positions (like Model 4)
- Popular because it supports inference via the forward-backward algorithm

Overcoming 1-to-N

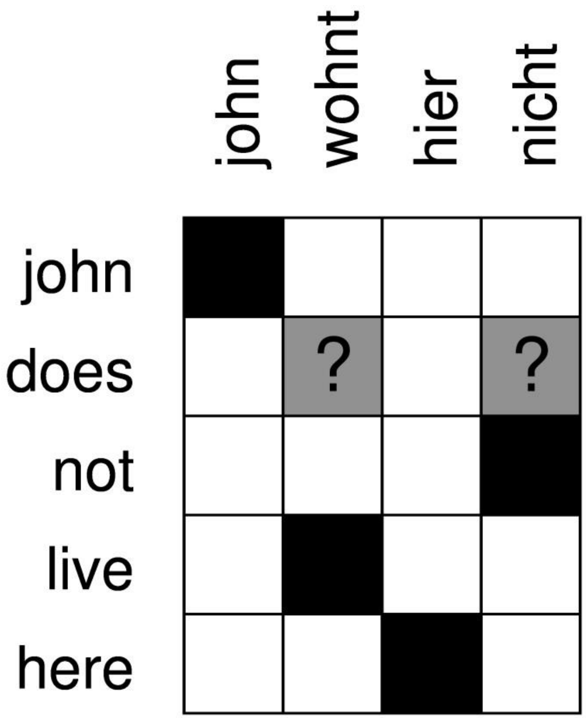
- We'll now discuss overcoming the poor assumption behind alignment functions

Word Alignment

Given a sentence pair, which words correspond to each other?

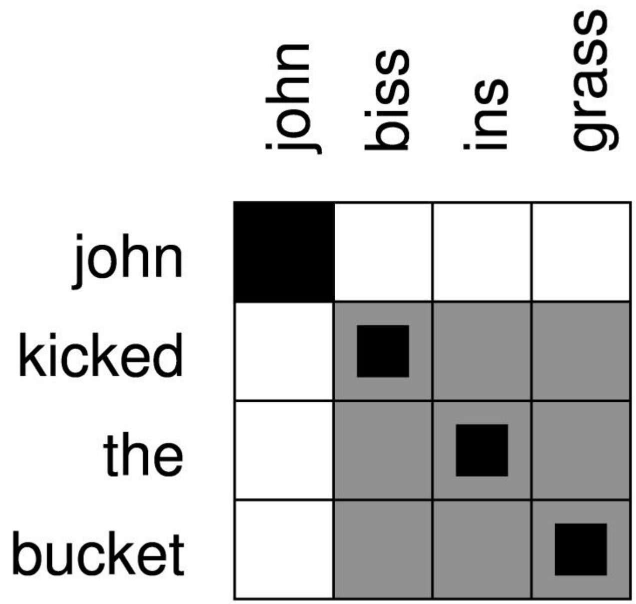


Word Alignment?



Is the English word does aligned to the German wohnt (verb) or nicht (negation) or neither?

Word Alignment?

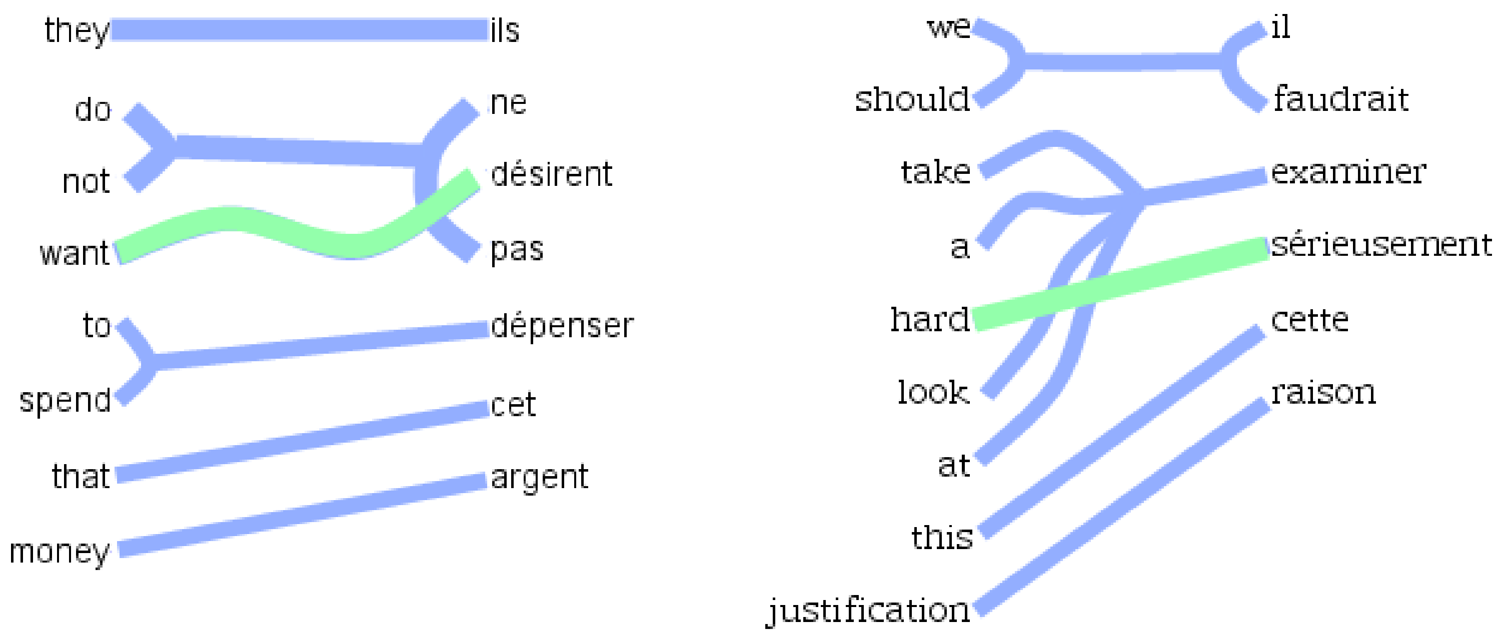


How do the idioms kicked the bucket and biss ins grass match up?
Outside this exceptional context, bucket is never a good translation for grass

Word Alignment with IBM Models

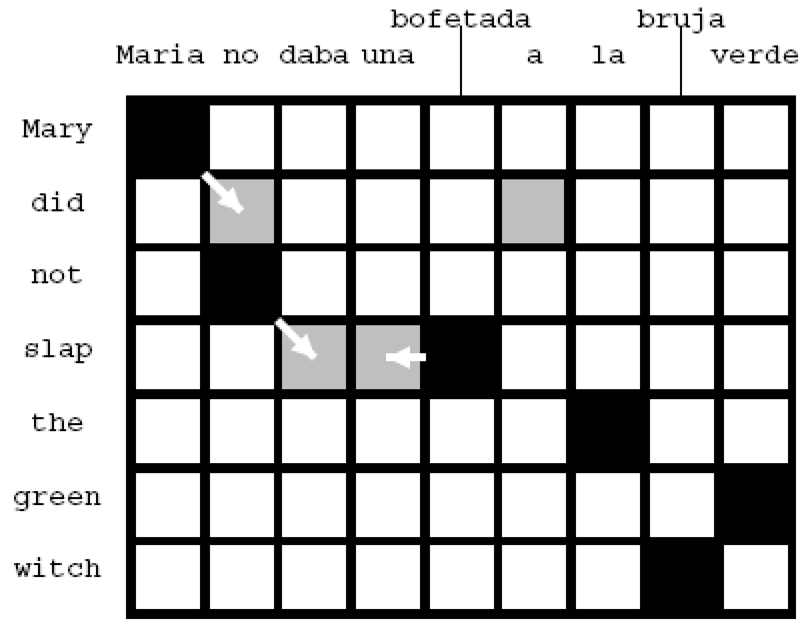
- IBM Models create a **many-to-one** mapping
  - words are aligned using an alignment function
  - a function may return the same value for different input (one-to-many mapping)
  - a function can not return multiple values for one input (no many-to-one mapping)
- Real word alignments have **many-to-many** mappings

IBM Models: 1-to-N Assumption



- 1-to-N assumption
  - Multi-word "cepts" (words in one language translated as a unit) only allowed on target side. Source side limited to single word "cepts".
  - Forced to create M-to-N alignments using heuristics

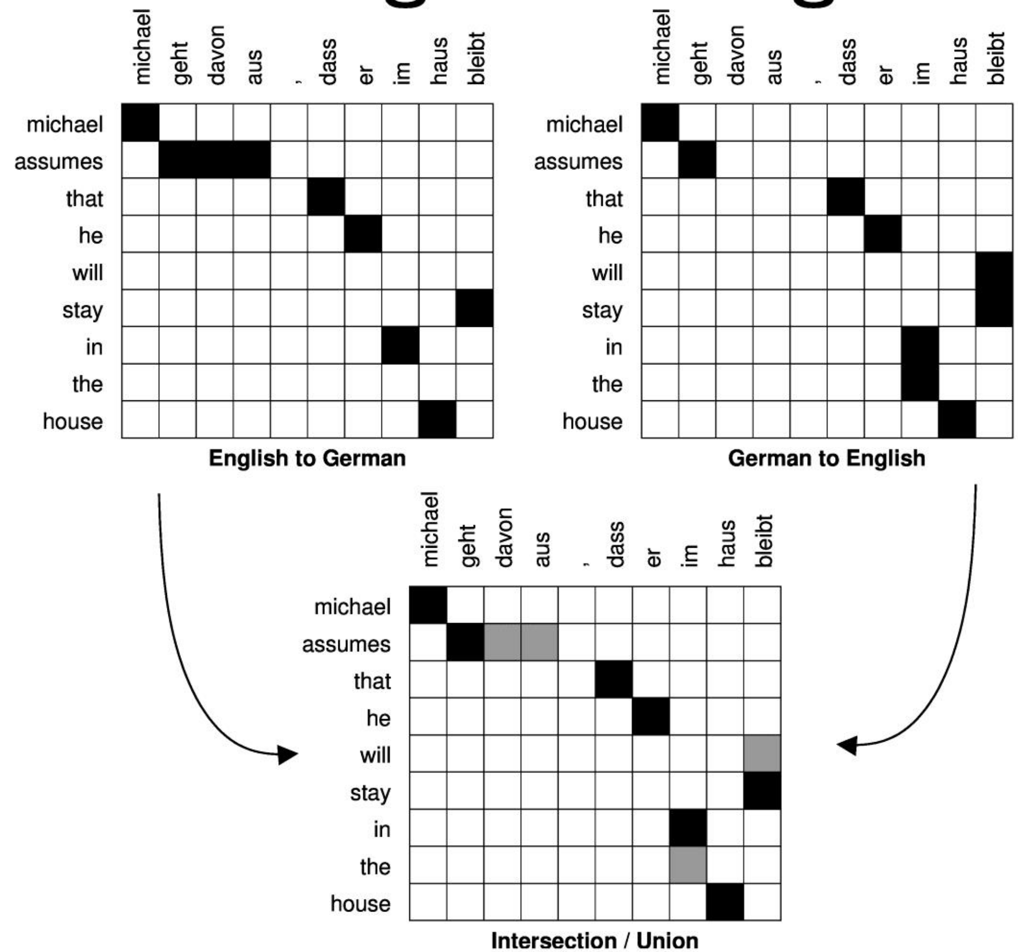
Symmetrizing word alignments



- *Grow* additional alignment points [Och and Ney, CompLing2003]

Slide from Koehn 2008

Symmetrizing Word Alignments



- Intersection of GIZA++ bidirectional alignments
- Grow additional alignment points [Och and Ney, CompLing2003]

Slide from Koehn 2009

Growing heuristic

grow-diag-final(e2f,f2e)

```
neighboring = {(-1,0),(0,-1),(1,0),(0,1),(-1,-1),(-1,1),(1,-1),(1,1)}
alignment A = intersect(e2f,f2e); grow-diag(); final(e2f); final(f2e);
```

grow-diag()

```
while new points added do
    for all English word e ∈ [1...e_n], foreign word f ∈ [1...f_n], (e, f) ∈ A do
        for all neighboring alignment points (e_new, f_new) do
            if (e_new unaligned OR f_new unaligned) AND (e_new, f_new) ∈ union(e2f,f2e) then
                add (e_new, f_new) to A
            end if
        end for
    end for
end while
```

final()

```
for all English word e_new ∈ [1...e_n], foreign word f_new ∈ [1...f_n] do
    if (e_new unaligned OR f_new unaligned) AND (e_new, f_new) ∈ union(e2f,f2e) then
        add (e_new, f_new) to A
    end if
end for
```

Slide from Koehn 2009

Discussion

- Most state of the art SMT systems are built as I presented
- Use IBM Models to generate both:
  - one-to-many alignment
  - many-to-one alignment
- Combine these two alignments using symmetrization heuristic
  - output is a many-to-many alignment
  - used for building decoder
- Moses toolkit for implementation: www.statmt.org
  - Uses Och and Ney GIZA++ tool for Model 1, HMM, Model 4

- However, there is newer work on alignment that is interesting!

Where we have been

- We defined the overall problem and talked about evaluation
- We have now covered word alignment
  - IBM Model 1, true Expectation Maximization
  - Briefly mentioned: IBM Model 4, approximate Expectation Maximization
  - Symmetrization Heuristics (such as Grow)
    - Applied to two Viterbi alignments (typically from Model 4)
    - Results in final word alignment

Where we are going

- We will define a high performance translation model
- We will show how to solve the search problem for this model (= decoding)