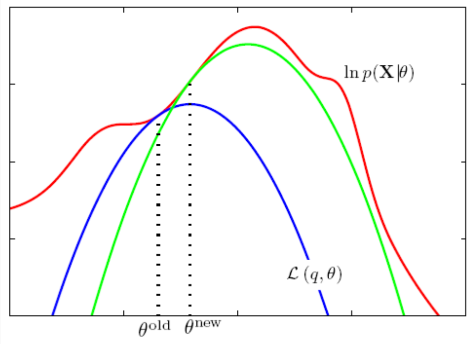
Statistical Machine Translation

-expectation-maximization-

Artem Sokolov
Computerlinguistik
Universität Heidelberg
Sommersemester 2015

material from P. Koehn, A. Ng



EM (Expectation Maximization) in a nutshell:

1. Initialize model parameters, e.g., uniform.
2. Assign probabilities to missing data. **E-step**
3. Estimate model parameters from manufactured/expected data. **M-step**
4. Iterate step 2 - 3 until convergence.

E:

$$q_i(z_i) = \frac{p(x_i, z; \theta)}{\sum_{z \in \mathcal{Z}} p(x_i, z; \theta)}$$

M:

$$\theta = \arg\max_{\theta} \sum_{i=1}^{m} \sum_{z \in \mathcal{Z}} q_i(z_i) \log \frac{p(x_i, z; \theta)}{q_i(z_i)}$$

$$x_i \to (\mathbf{e}, \mathbf{f})$$
$$z_i \to a$$
$$q_i(z_i) = p(a|\mathbf{e}, \mathbf{f})$$
$$p(x_i, z_i; \theta) = p(a, \mathbf{e}|\mathbf{f})$$

$$\sum_{i=1}^{m} \qquad \sum_{z \in \mathcal{Z}} q_i^t(z_i) \qquad \log \frac{p(x_i, z; \theta_{t+1})}{q_i^t(z_i)}$$

$$\sum_{(\mathbf{e},\mathbf{f}) \in \mathcal{D}} \qquad \sum_{a} p(a|\mathbf{e}, \mathbf{f}) \qquad \log \frac{p(a, \mathbf{e}|\mathbf{f})}{p(a|\mathbf{e}, \mathbf{f})}$$

E:

$$p(a|\mathbf{e},\mathbf{f}) = \frac{p(a,\mathbf{e}|\mathbf{f})}{\sum_a p(a,\mathbf{e}|\mathbf{f})}$$

M:

$$\theta = \arg\max_\theta \sum_{i=1}^{m} \sum_a p(a|\mathbf{e},\mathbf{f}) \log \frac{p(a,\mathbf{e}|\mathbf{f})}{p(a|\mathbf{e},\mathbf{f})}$$

IBM Model 1:

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

IBM Model 1:

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

The trick:

$$p(\mathbf{e}|\mathbf{f}) = \sum_a p(\mathbf{e}, a|\mathbf{f}) = \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} \sum_{i=0}^{l_f} t(e_j|f_i)$$

IBM Model 1:

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

The trick:

$$p(\mathbf{e}|\mathbf{f}) = \sum_a p(\mathbf{e}, a|\mathbf{f}) = \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} \sum_{i=0}^{l_f} t(e_j | f_i)$$

Final for E-step:

$$p(a|\mathbf{e}, \mathbf{f}) = \frac{p(\mathbf{e}, a|\mathbf{f})}{p(\mathbf{e}|\mathbf{f})} = \prod_{j=1}^{l_e} \frac{t(e_j | f_{a(j)})}{\sum_{i=0}^{l_f} t(e_j | f_i)}$$

Counts over 1 sentence:

$$c(e|f;\mathbf{e},\mathbf{f}) = \sum_{a} p(a|\mathbf{e},\mathbf{f}) \sum_{j=1}^{l_e} \delta(e, e_j)\delta(f, f_{a(j)})$$

Counts over 1 sentence:

$$c(e|f;\mathbf{e},\mathbf{f}) = \sum_a p(a|\mathbf{e},\mathbf{f}) \sum_{j=1}^{l_e} \delta(e,e_j)\delta(f,f_{a(j)})$$

Counts over corpus:

$$c(e|f;\mathbf{e},\mathbf{f}) = \frac{t(e|f)}{\sum_{i=0}^{l_f} t(e|f_i)} \sum_{j=1}^{l_e} \delta(e,e_j) \sum_{i=0}^{l_f} \delta(f,f_i)$$

Counts over 1 sentence:

$$c(e|f;\mathbf{e},\mathbf{f}) = \sum_{a} p(a|\mathbf{e},\mathbf{f}) \sum_{j=1}^{l_e} \delta(e,e_j)\delta(f,f_{a(j)})$$

Counts over corpus:

$$c(e|f;\mathbf{e},\mathbf{f}) = \frac{t(e|f)}{\sum_{i=0}^{l_f} t(e|f_i)} \sum_{j=1}^{l_e} \delta(e,e_j) \sum_{i=0}^{l_f} \delta(f,f_i)$$

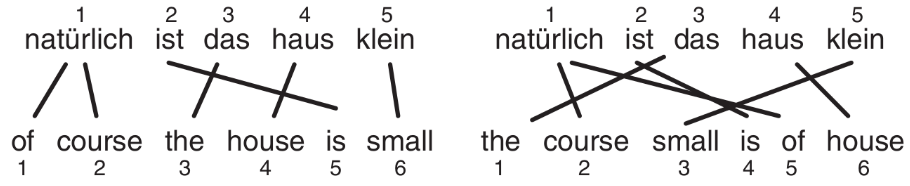
Final for M-step:

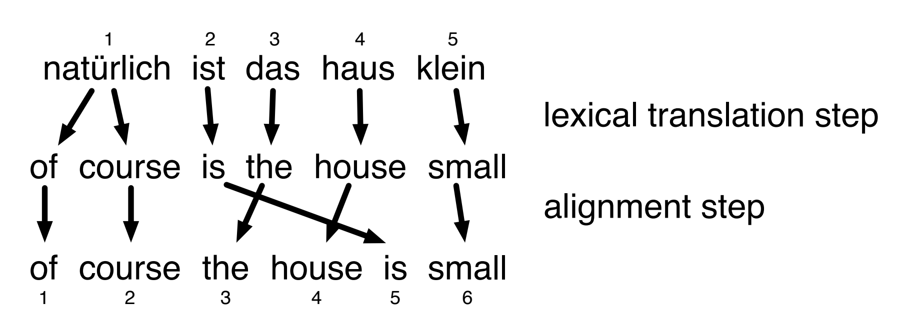
$$t(e|f) = \frac{\sum_{(\mathbf{e},\mathbf{f})} c(e|f;\mathbf{e},\mathbf{f})}{\sum_{e} \sum_{(\mathbf{e},\mathbf{f})} c(e|f;\mathbf{e},\mathbf{f})}$$

EM pseudo code

```
Require: set of sentence pairs (e, f)
Ensure: translation prob. t(e|f)
  initialize t(e|f) uniformly
  while not converged do
      {initialize}
      count(e|f) = 0 for all e, f
      total(f) = 0 for all f
      for all sentence pairs (e,f) do
          {compute normalization}
          for all words e in e do
              s-total(e) = 0
              for all words f in f do
                  s-total(e) += t(e|f)
              end for
          end for
          {collect counts}
          for all words e in e do
              for all words f in f do
                  count(e|f) += t(e|f) / s-total(e)
                  total(f) += t(e|f) / s-total(e)
              end for
          end for
      end for
      {estimate probabilities}
      for all foreign words f do
          for all English words e do
              t(e|f) = count(e|f) / total(f)
          end for
      end for
  end while
```

IBM Model 2 vs. Model 1




1
natürlich
2
ist
3
das
4
haus
5
klein
lexical translation step
of
course
is
the
house
small
alignment step
of
course
the
house
is
small
1
2
3
4
5
6

IBM Model 1: lexical translation, all reorderings / alignments are equally likely; *still used for initialization*
IBM Model 2: adds explicit reordering / alignment model
IBM Model 3: adds fertility model
IBM Model 4: adds improved reordering model
IBM Model 5: fixes deficiency

IBM Model 1:

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

IBM Model 1:

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

IBM Model 2:

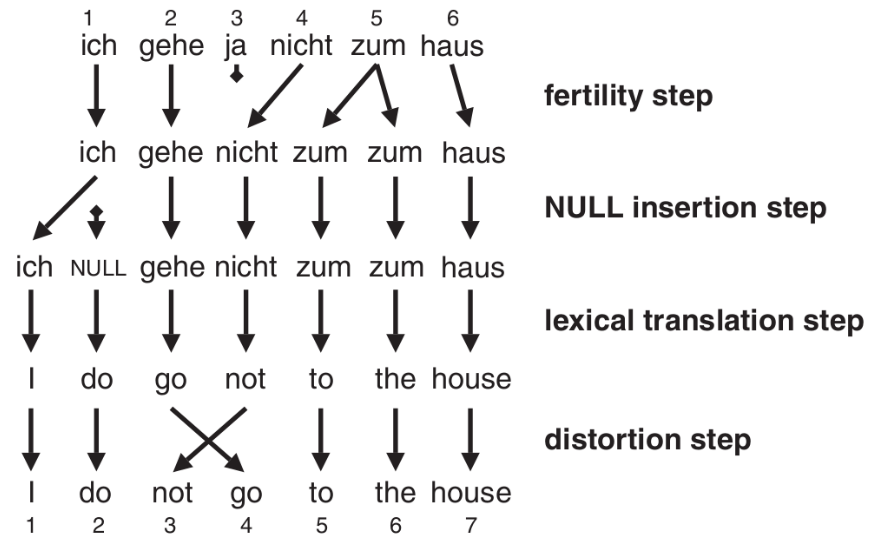
$$p(\mathbf{e}, a|\mathbf{f}) = \epsilon \prod_{j=1}^{l_e} a(i|j, l_e, l_f) t(e_j|f_{a(j)})$$

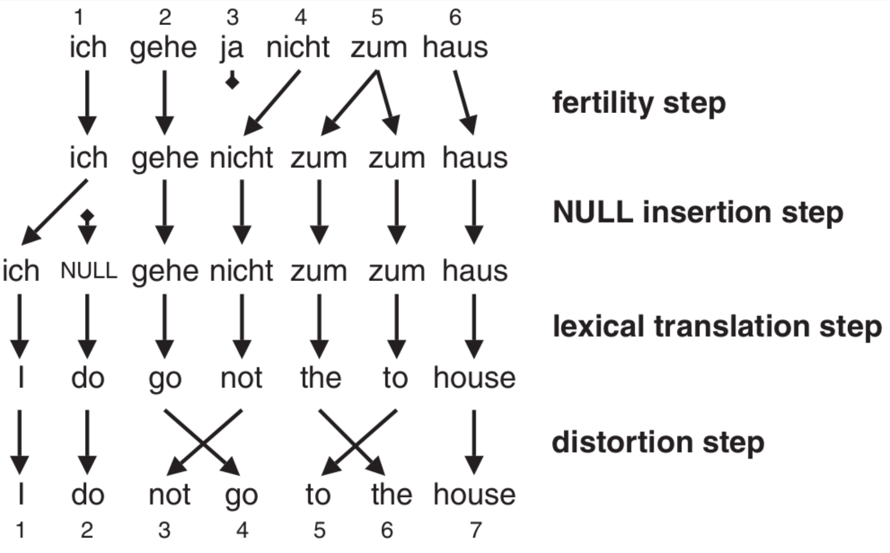
IBM Model 2:

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

Final for E-step:

$$\begin{aligned} p(a|\mathbf{e}, \mathbf{f}) &= \frac{p(\mathbf{e}, a|\mathbf{f})}{p(\mathbf{e}|\mathbf{f})} \\ &= \prod_{j=1}^{l_e} \frac{t(e_j|f_{a(j)})}{\sum_{i=0}^{l_f} t(e_j|f_i)} \end{aligned}$$


1 2 3 4 5 6
ich gehe ja nicht zum haus
fertility step
ich gehe nicht zum zum haus
NULL insertion step
ich NULL gehe nicht zum zum haus
lexical translation step
I do go not to the house
distortion step
I do not go to the house
1 2 3 4 5 6 7


1 2 3 4 5 6
ich gehe ja nicht zum haus
fertility step
ich gehe nicht zum zum haus
NULL insertion step
ich NULL gehe nicht zum zum haus
lexical translation step
I do go not the to house
distortion step
I do not go to the house
1 2 3 4 5 6 7

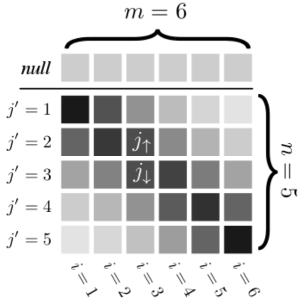
$$\text{Given}: \ \mathbf{f}, \ n = |\mathbf{f}|, \ m = |\mathbf{e}|, \ p_0, \ \lambda, \ \boldsymbol{\theta}$$

$$h(i, j, m, n) = -\left|\frac{i}{m} - \frac{j}{n}\right|$$

$$\delta(a_i = j \mid i, m, n) = \begin{cases} p_0 & j = 0 \\ (1 - p_0) \times \frac{e^{\lambda h(i,j,m,n)}}{Z_\lambda(i,m,n)} & 0 < j \leq n \\ 0 & \text{otherwise} \end{cases}$$

$$a_i \mid i, m, n \sim \delta(\cdot \mid i, m, n) \qquad 1 \leq i \leq m$$

$$e_i \mid a_i, f_{a_i} \sim \theta(\cdot \mid f_{a_i}) \qquad 1 \leq i \leq m$$



$$p(e_i, a_i \mid \mathbf{f}, m, n) = \delta(a_i \mid i, m, n) \times \theta(e_i \mid f_{a_i})$$

$$p(e_i \mid \mathbf{f}, m, n) = \sum_{j=0}^{n} p(e_i, a_i = j \mid \mathbf{f}, m, n)$$

$$p(a_i \mid e_i, \mathbf{f}, m, n) = \frac{p(e_i, a_i \mid \mathbf{f}, m, n)}{p(e_i \mid \mathbf{f}, m, n)}$$

$$\begin{aligned} p(\mathbf{e} \mid \mathbf{f}) &= \prod_{i=1}^{m} p(e_i \mid \mathbf{f}, m, n) \\ &= \prod_{i=1}^{m} \sum_{j=0}^{n} \delta(a_i \mid i, m, n) \times \theta(e_i \mid f_{a_i}) \end{aligned}$$

still quadratic time!

Optimizing for λ

$$\nabla_\lambda \mathcal{L} = \mathbb{E}_{p(a_i|e_i,\mathbf{f},m,n)}\left[h(i,a_i,m,n)\right] - \mathbb{E}_{\delta(j'|i,m,n)}\left[h(i,j',m,n)\right]$$

Advantages of fast align

- still quadratic time of the E-step
- Z_λ calculation using simple geometric progression formula in constant time!
- although optimizing for λ requires gradient descent, it also uses the same formula