IBM Translation Models

Martin Emms

October 4, 2018
IBM models

Probabilities and Translation
Alignments
IBM Model 1 definitions
IBM models intro
Outline

IBM models
Probabilities and Translation
Alignments
IBM Model 1 definitions
Lexical Translation

- 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*
Collect Statistics

- Suppose a parallel corpus, with German sentences paired with English sentences, and suppose people inspect this marking how *Haus* is translated.
  
  : 
  
  *das Haus ist klein* the *house* is small
  
  : 

- Hypothetical table of frequencies

<table>
<thead>
<tr>
<th>Translation of Haus</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>house</td>
<td>8,000</td>
</tr>
<tr>
<td>building</td>
<td>1,600</td>
</tr>
<tr>
<td>home</td>
<td>200</td>
</tr>
<tr>
<td>household</td>
<td>150</td>
</tr>
<tr>
<td>shell</td>
<td>50</td>
</tr>
</tbody>
</table>
Estimation of Translation Probabilities

- from this could use relative frequencies as estimate of translation probabilities $t(e|Haus)$
- technically this is a maximum likelihood estimate – there could be others
- outcome would be

$$
tr(e|Haus) = \begin{cases} 
0.8 & \text{if } e = \text{house}, \\
0.16 & \text{if } e = \text{building}, \\
0.02 & \text{if } e = \text{home}, \\
0.015 & \text{if } e = \text{household}, \\
0.005 & \text{if } e = \text{shell}.
\end{cases}
$$
IBM models

- the so-called IBM models seek a probabilistic model of translation one of whose ingredients is this kind of lexical translation probability.
IBM models

- the so-called IBM models seek a probabilistic model of translation one of whose ingredients is this kind of lexical translation probability.
- there’s a sequence of models of increasing complexity (models 1-5). The simplest models pretty much just use lexical translation probability
IBM models

- the so-called IBM models seek a probabilistic model of translation one of whose ingredients is this kind of lexical translation probability.
- there's a sequence of models of increasing complexity (models 1-5). The simplest models pretty much just use lexical translation probability
- parallel corpora are used (eg. pairing German sentences with English sentences) but crucially there is no human inspection to find how given German words are translated to English words, ie. info is of form
  
  \[ \text{das Haus ist klein} \quad \text{the house is small} \]

  

IBM models

- the so-called IBM models seek a probabilistic model of translation one of whose ingredients is this kind of lexical translation probability.
- there’s a sequence of models of increasing complexity (models 1-5). The simplest models pretty much just use lexical translation probability.
- parallel corpora are used (eg. pairing German sentences with English sentences) but crucially there is no human inspection to find how given German words are translated to English words, ie. info is of form:
  
  \[
  \begin{align*}
  \text{das Haus ist klein} & \quad \text{the house is small} \\
  \end{align*}
  \]
  
  though originally developed as models of translation, these models are now used as models of alignment, providing crucial training input for so-called 'phrase-based SMT'.
Notation
For reasons that will become apparent, we will use $O$ for the language we want to translate \textit{from} and $S$ for the language we want to translate \textit{to}. 
Notation

- For reasons that will become apparent, we will use $\mathcal{O}$ for the language we want to translate from $S$ for the language we want to translate to.
- $o$ is a single sentence from $\mathcal{O}$, and is a sequence $(o_1 \ldots o_j \ldots o_{\ell_o})$; $\ell_o$ is length $o$. 
For reasons that will become apparent, we will use $O$ for the language we want to translate \textit{from} $S$ for the language we want to translate \textit{to}.

- $o$ is a single sentence from $O$, and is a sequence $(o_1 \ldots o_j \ldots o_{\ell_o})$; $\ell_o$ is length $o$.
- $s$ is a single sentence from $S$, and is a sequence $(s_1 \ldots s_i \ldots s_{\ell_s})$; $\ell_s$ is length $o$. 

Notation

- For reasons that will become apparent, we will use $O$ for the language we want to translate \textit{from}
  $S$ for the language we want to translate \textit{to}
- $o$ is a single sentence from $O$, and is a sequence $(o_1 \ldots o_j \ldots o_{\ell_o})$; $\ell_o$ is length $o$
- $s$ is a single sentence from $S$, and is a sequence $(s_1 \ldots s_i \ldots s_{\ell_s})$; $\ell_s$ is length $o$
- the set of all possible words of language $O$ is $V_o$
Notation

- For reasons that will become apparent, we will use $O$ for the language we want to translate from and $S$ for the language we want to translate to.
- $o$ is a single sentence from $O$, and is a sequence $(o_1 \ldots o_j \ldots o_{\ell_o})$; $\ell_o$ is the length of $o$.
- $s$ is a single sentence from $S$, and is a sequence $(s_1 \ldots s_i \ldots s_{\ell_s})$; $\ell_s$ is the length of $s$.
- The set of all possible words of language $O$ is $V_o$.
- The set of all possible words of language $S$ is $V_s$. 
For reasons that will become apparent, we will use
\( \mathcal{O} \) for the language we want to translate \textit{from} 
\( \mathcal{S} \) for the language we want to translate \textit{to}

- \( \mathbf{o} \) is a single sentence from \( \mathcal{O} \), and is a sequence \((o_1 \ldots o_j \ldots o_{\ell_o})\); \( \ell_o \) is length \( \mathbf{o} \)

- \( \mathbf{s} \) is a single sentence from \( \mathcal{S} \), and is a sequence \((s_1 \ldots s_i \ldots s_{\ell_s})\); \( \ell_s \) is length \( \mathbf{o} \)

- the set of all possible words of language \( \mathcal{O} \) is \( \mathcal{V}_o \)
- the set of all possible words of language \( \mathcal{S} \) is \( \mathcal{V}_s \)
- comments on notation in Koehn, J&M
The sparsity problem
The sparsity problem

- Suppose for two languages you have large sentence-aligned corpus $d$. Say the two languages are $O$ and $S$. 
The sparsity problem

- Suppose for two languages you have large sentence-aligned corpus \( d \). Say the two languages are \( O \) and \( S \).
- in principle for any sentence \( o \in O \) could work out the probabilities of its various translations \( s \) by relative frequency
The sparsity problem

- Suppose for two languages you have large sentence-aligned corpus $d$. Say the two languages are $O$ and $S$.
- In principle for any sentence $o \in O$ could work out the probabilities of its various translations $s$ by relative frequency

$$p(s|o) = \frac{\text{count}(\langle o, s \rangle \in d)}{\sum_{s'} \text{count}(\langle o, s' \rangle \in d)}$$
The sparsity problem

- Suppose for two languages you have large sentence-aligned corpus $d$. Say the two languages are $\mathcal{O}$ and $\mathcal{S}$.
- In principle for any sentence $o \in \mathcal{O}$ could work out the probabilities of its various translations $s$ by relative frequency

$$p(s|o) = \frac{\text{count}(\langle o, s \rangle \in d)}{\sum_{s'} \text{count}(\langle o, s' \rangle \in d)}$$

- But even in very large corpora the vast majority of possible $o$ and $s$ occur zero times. So this method gives uselessly bad estimates.
The Noisy-Channel formulation

- recalling Bayesian classification, finding \( s \) from \( o \):

\[
\arg \max_s P(s|o) = \arg \max_s \frac{P(s, o)}{P(o)}
\] (1)
The Noisy-Channel formulation

- recalling Bayesian classification, finding $s$ from $o$:

$$
\arg \max_s P(s|o) = \arg \max_s \frac{P(s, o)}{P(o)}
\tag{1}
$$

$$
= \arg \max_s P(s, o)
\tag{2}
$$
The Noisy-Channel formulation

recalling Bayesian classification, finding $s$ from $o$:

$$
\arg \max_s P(s|o) = \arg \max_s \frac{P(s, o)}{P(o)} \\
= \arg \max_s P(s, o) \\
= \arg \max_s P(o|s) \times P(s)
$$
The Noisy-Channel formulation

- recalling Bayesian classification, finding $s$ from $o$:

$$
\arg \max_s P(s|o) = \arg \max_s \frac{P(s, o)}{P(o)} = \arg \max_s P(s, o) = \arg \max_s P(o|s) \times P(s)
$$

- can then try to factorise $P(o|s)$ and $P(s)$ into clever combination of other probability distributions (not sparse, learnable, allowing solution of arg-max problem).
The Noisy-Channel formulation

- recalling Bayesian classification, finding $s$ from $o$:

$$\arg\max_s P(s|o) = \arg\max_s \frac{P(s,o)}{P(o)} = \arg\max_s P(s,o) = \arg\max_s P(o|s) \times P(s)$$

- can then try to factorise $P(o|s)$ and $P(s)$ into clever combination of other probability distributions (not sparse, learnable, allowing solution of arg-max problem). IBM models 1-5 can be used for $P(o|s)$;
The Noisy-Channel formulation

- recalling Bayesian classification, finding $s$ from $o$:

$$
\arg\max_s P(s|o) = \arg\max_s \frac{P(s, o)}{P(o)} = \arg\max_s P(s, o) = \arg\max_s P(o|s) \times P(s)
$$

- can then try to factorise $P(o|s)$ and $P(s)$ into clever combination of other probability distributions (not sparse, learnable, allowing solution of arg-max problem). IBM models 1-5 can be used for $P(o|s)$; $P(s)$ is the topic of so-called 'language models'.
The Noisy-Channel formulation

- recalling Bayesian classification, finding $s$ from $o$:

$$\arg \max_s P(s|o) = \arg \max_s \frac{P(s, o)}{P(o)} = \arg \max_s P(s, o) = \arg \max_s P(o|s) \times P(s)$$ (1)

- can then try to factorise $P(o|s)$ and $P(s)$ into clever combination of other probability distributions (not sparse, learnable, allowing solution of arg-max problem). IBM models 1-5 can be used for $P(o|s)$; $P(s)$ is the topic of so-called 'language models'.

- The reason for the notation $s$ and $o$ is that (3) is the defining equation of Shannons 'noisy-channel' formulation of decoding, where an original 'source' $s$ has to be recovered from a noisy observed signal $o$, the noisiness defined by $P(o|s)$.
Now have to start look at the details of the IBM models of $P(o|s)$, starting with the very simplest models.

What all the models have in common is that they define $P(o|s)$ as a combination of other probability distributions.
Outline

IBM models
Probabilities and Translation
Alignments
IBM Model 1 definitions
Alignments (informally)
Alignments (informally)

- When \( s \) and \( o \) are translations of each other, usually can say which pieces of \( s \) and \( o \) are translations of each other. eg.
Alignments (informally)

- When s and o are translations of each other, usually can say which pieces of s and o are translations of each other. eg.

```
1 2 3 4
das Haus ist klein
1 2 3 4
the house is small
```
Alignments (informally)

- When \( s \) and \( o \) are translations of each other, usually can say which pieces of \( s \) and \( o \) are translations of each other. eg.

\[
\begin{align*}
&s & & o \\
1 & \text{das} & 2 & \text{Haus} & 3 & \text{ist} & 4 & \text{klein} \\
1 & \text{the} & 2 & \text{house} & 3 & \text{is} & 4 & \text{small} \\
1 & \text{das} & 2 & \text{Haus} & 3 & \text{ist} & 4 & \text{klitzeklein} \\
1 & \text{the} & 2 & \text{house} & 3 & \text{is} & 4 & \text{very} & 5 & \text{small}
\end{align*}
\]
Alignments (informally)

- When \(s\) and \(o\) are translations of each other, usually can say which **pieces** of \(s\) and \(o\) are translations of each other. eg.

```
1 2 3 4
das Haus ist klein
the house is small
```

```
1 2 3 4
klitzeklein
very small
```

- In SMT such a piece-wise correspondence is called an **alignment**
Alignments (informally)

- When \( s \) and \( o \) are translations of each other, usually can say which pieces of \( s \) and \( o \) are translations of each other. eg.

  \[
  \begin{array}{cccc}
  1 & 2 & 3 & 4 \\
  \text{das} & \text{Haus} & \text{ist} & \text{klein} \\
  \text{the} & \text{house} & \text{is} & \text{small} \\
  1 & 2 & 3 & 4 \\
  \text{das} & \text{Haus} & \text{ist} & \text{klitzeklein} \\
  \text{the} & \text{house} & \text{is} & \text{very small} \\
  \end{array}
  \]

- In SMT such a piece-wise correspondence is called an \textit{alignment}

- warning: there are quite a lot of varying formal definitions of alignment
Hidden Alignment
Hidden Alignment

- key feature of the IBM models is to assume there is a hidden alignment, a between o and s
Hidden Alignment

▶ key feature of the IBM models is to assume there is a hidden alignment, a between o and s

▶ so a pair \( \langle o, s \rangle \) from a sentence-aligned corpus is seen as a partial version of the fully observed case:

\[ \langle o, a, s \rangle \]
Hidden Alignment

- Key feature of the IBM models is to assume there is a hidden alignment, a between o and s.
- So a pair \( \langle o, s \rangle \) from a sentence-aligned corpus is seen as a partial version of the fully observed case:

\[
\langle o, a, s \rangle
\]

- A model is essentially made of \( p(o, a|s) \), and having this allows other things to be defined.
Hidden Alignment

- key feature of the IBM models is to assume there is a **hidden alignment**, \( a \) between \( o \) and \( s \)
- so a pair \( \langle o, s \rangle \) from a sentence-aligned corpus is seen as a partial version of the fully observed case:

\[
\langle o, a, s \rangle
\]

- A model is essentially made of \( p(o, a|s) \), and having this allows other things to be defined
- best translation:

\[
\arg \max_s P(s, o) = \arg \max_s (\sum_a p(o, a|s) \times p(s))
\]
Hidden Alignment

- key feature of the IBM models is to assume there is a hidden alignment, \textit{a} between \textit{o} and \textit{s}
- so a pair \langle o, s \rangle from a sentence-aligned corpus is seen as a partial version of the fully observed case:

\[ \langle o, a, s \rangle \]

- A model is essentially made of \( p(o, a | s) \), and having this allows other things to be defined
- best translation:

\[ \arg \max_s P(s, o) = \arg \max_s ([\sum_a p(o, a | s)] \times p(s)) \]

- best alignment:

\[ \arg \max_a [p(o, a | s)] \]
IBM Alignments

- Define alignment with a function,
IBM Alignments

- Define alignment with a function, from posn \( j \) in \( o \) to posn. \( i \) in \( s \)
IBM Alignments

- Define alignment with a function, from posn $j$ in $o$ to posn. $i$ in $s$ so $a : j \rightarrow i$
IBM Alignments

- Define alignment with a \textit{function}, from posn $j$ in $o$ to posn. $i$ in $s$
  so $a: j \rightarrow i$

- the picture

\begin{center}
\begin{tabular}{cccc}
1 & 2 & 3 & 4 \\
\hline
das & Haus & ist & klein \\
\hline
the & house & is & small \\
\hline
1 & 2 & 3 & 4
\end{tabular}
\end{center}
IBM Alignments

- Define alignment with a function, from posn \( j \) in \( o \) to posn. \( i \) in \( s \), so \( a : j \rightarrow i \)

- the picture

```
1 2 3 4
1 2 3 4

das Haus ist klein
the house is small
```

represents

\[ a : \{1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 4\} \]
Some weirdness about directions

\[
\begin{array}{cccc}
1 & 2 & 3 & 4 \\
\text{das} & \text{Haus} & \text{ist} & \text{klein} \\
\text{the} & \text{house} & \text{is} & \text{small}
\end{array}
\]

\[
a : 1 \rightarrow 1, \\
2 \rightarrow 2, \\
3 \rightarrow 3, \\
4 \rightarrow 4
\]
Some weirdness about directions

1 2 3 4

das Haus ist klein
| | | |
| the house is small

a: 1 → 1,
   2 → 2,
   3 → 3,
   4 → 4

Note here o is English, and s is German
Some weirdness about directions

$\begin{align*}
\text{das Haus ist klein} &\quad a : \quad 1 \rightarrow 1, \\
\text{the house is small} &\quad 2 \rightarrow 2, \\
&\quad 3 \rightarrow 3, \\
&\quad 4 \rightarrow 4
\end{align*}$

- Note here o is English, and s is German
- the alignment goes up the page, English-to-German,
Some weirdness about directions

$\text{das Haus ist klein}$

\begin{align*}
1 & \rightarrow 1, \\
2 & \rightarrow 2, \\
3 & \rightarrow 3, \\
4 & \rightarrow 4
\end{align*}

Note here $\text{o}$ is English, and $\text{s}$ is German.

The alignment goes up the page, English-to-German,

they will be used though in a model of $P(o|s)$,
so down the page, German-to-English.
Comparison to 'edit distance' alignments

in case you have ever studied 'edit distance' alignments . . .

- like edit-dist alignments, its a function:
  so can’t align 1 o words with 2 s words
- like edit-dist alignments, some s words can be unmapped to
  (cf. insertions)
- like edit-dist alignments, some o words can be mapped to nothing
  (cf. deletions)
- unlike edit-dist alignments, order not preserved: so $j < j' \not\Rightarrow a(j) < a(j')$
N-to-1 Alignment (ie. 1-to-N Translation)

- das Haus ist klitzeklein
- the house is very small

\[ a : \{1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 4, 5 \rightarrow 4\} \]
N-to-1 Alignment (ie. 1-to-N Translation)

\[ \begin{align*}
&\text{1} \quad \text{2} \quad \text{3} \quad \text{4} \\
&\text{das} \quad \text{Haus} \quad \text{ist} \quad \text{klitzeklein} \\
\downarrow & \downarrow & \downarrow & \downarrow \\
&\text{the} \quad \text{house} \quad \text{is} \quad \text{very} \quad \text{small} \\
&\text{1} \quad \text{2} \quad \text{3} \quad \text{4} \quad \text{5}
\end{align*} \]

- \( a : \{1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 4, 5 \rightarrow 4\} \)

- \( N \) words of \( o \) can be aligned to \( 1 \) word of \( s \)

  (needed when 1 word of \( s \) translates into \( N \) words of \( o \))
Reordering

1 2 3 4
klein  ist  das  Haus

1 2 3 4
the  house  is  small
Reordering

1  2  3  4
klein  ist  das  Haus
	he  house  is  small

$$\rightarrow a : \{1 \rightarrow 3, 2 \rightarrow 4, 3 \rightarrow 2, 4 \rightarrow 1\}$$
Reordering

> a : \{1 → 3, 2 → 4, 3 → 2, 4 → 1\}

> alignment does not preserve o word order

(needed when s words reordered during translation)
s words not mapped to (ie. dropped in translation)

das Haus ist ja klein
the house is small
s words not mapped to (ie. dropped in translation)

\[
\begin{align*}
\text{das \ Haus \ ist \ ja \ klein} \\
\text{the \ house \ is \ small}
\end{align*}
\]

- \( a : \{1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 5\} \)

- some s words are not mapped-to by the alignment
  
  (needed when s words are dropped during translation
  
  (here the German flavouring particle 'ja' is dropped)
0 1 2 3 4 5
NULL ich gehe nicht zum haus
I do not go to the house
1 2 3 4 5 6 7
0  1  2  3  4  5
NULL ich gehe nicht zum haus

I do not go to the house

ширим a: {1 → 1, 2 → 0, 3 → 3, 4 → 2, 5 → 4, 6 → 4, 7 → 5}
o words mapped to nothing (ie. inserting in translation)

\[
\begin{array}{ccccccc}
0 & 1 & 2 & 3 & 4 & 5 \\
\text{NULL} & \text{ich} & \text{gehe} & \text{nicht} & \text{zum} & \text{haus}
\end{array}
\]

\[
\begin{array}{cccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\text{I} & \text{do} & \text{not} & \text{go} & \text{to} & \text{the} & \text{house}
\end{array}
\]

- \( a : \{1 \rightarrow 1, 2 \rightarrow 0, 3 \rightarrow 3, 4 \rightarrow 2, 5 \rightarrow 4, 6 \rightarrow 4, 7 \rightarrow 5\} \)

- some o word are mapped to nothing by the alignment

(needed when o words have no clear origin during translation)

The is no clear origin in German of the English 'do'

formally represented by alignment to special NULL token
Outline

IBM models
- Probabilities and Translation
- Alignments
- IBM Model 1 definitions
IBM Model 1
IBM Model 1

- basically a hidden variable $a$, aligning $o$ to $s$ is assumed.
IBM Model 1

- basically a hidden variable $a$, aligning $o$ to $s$ is assumed.
- in more detail, IBM model 1 will define a probability model of

\[ P(o, a, L, s) \]

where $L$ is length for $o$ sentences, and $a$ is an alignment from $o$ sentences of length $L$ to $s$. 
IBM Model 1

- basically a hidden variable $a$, aligning $o$ to $s$ is assumed.
- in more detail, IBM model 1 will define a probability model of

$$P(o, a, L, s)$$

where $L$ is length for $o$ sentences, and $a$ is an alignment from $o$ sentences of length $L$ to $s$.

- $o$, $a$, $L$ are intended to be synchronized in the sense that if $L$ is not the $l_o$ the probability is zero. Similarly if $a$ is not an alignment function from length $L$ sequences to length $l_s$ sequences, the probability is 0. So we will write

$$P(o, a, l_o, s)$$
Length dependency
Length dependency

- first without any assumptions, via the chain rule:

\[ P(o, a, \ell_0, s) = P(o, a, \ell_0 | s) \times P(s) \]
Length dependency

- first without any assumptions, via the chain rule:

\[ P(o, a, \ell_o, s) = P(o, a, \ell_o | s) \times P(s) \]

the IBM model1 assumptions are all about \( P(o, a, \ell_o | s) \). The assumptions can be shown by a succession of applications of the chain rule concerning \((o, a, \ell_o)\)
Length dependency

- first without any assumptions, via the chain rule:

\[ P(o, a, \ell_o, s) = P(o, a, \ell_o|s) \times P(s) \]

the IBM model1 assumptions are all about \( P(o, a, \ell_o|s) \). The assumptions can be shown by a succession of applications of the chain rule concerning \((o, a, \ell_o)\)

- concerning \( \ell_o \), still without any particular assumptions

\[ P(o, a, \ell_o|s) = P(o, a|\ell_o, s) \times p(\ell_o|s) \]
Length dependency

- first without any assumptions, via the chain rule:

\[ P(o, a, \ell_o, s) = P(o, a, \ell_o|s) \times P(s) \]

- the IBM model 1 assumptions are all about \( P(o, a, \ell_o|s) \). The assumptions can be shown by a succession of applications of the chain rule concerning \((o, a, \ell_o)\)

- concerning \( \ell_o \), still without any particular assumptions

\[ P(o, a, \ell_o|s) = P(o, a|\ell_o, s) \times p(\ell_o|s) \]

An assumption of IBM Model 1 is that the dependency \( p(\ell_o|s) \) can be expressed as a dependency just on the length \( \ell_s \), so by some distribution \( p(L|\ell_s) \).
Length dependency

- first without any assumptions, via the chain rule:

\[ P(o, a, \ell_o, s) = P(o, a, \ell_o|s) \times P(s) \]

the IBM model 1 assumptions are all about \( P(o, a, \ell_o|s) \). The assumptions can be shown by a succession of applications of the chain rule concerning \((o, a, \ell_o)\)

- concerning \( \ell_o \), still without any particular assumptions

\[ P(o, a, \ell_o|s) = P(o, a|\ell_o, s) \times p(\ell_o|s) \]

An assumption of IBM Model 1 is that the dependency \( p(\ell_o|s) \) can be expressed as a dependency just on the length \( \ell_s \), so by some distribution \( p(L|\ell_s) \).

- Usually it's stated that \( p(L|\ell_s) \) is uniform: ie. all \( L \) equally likely
Length dependency

- first without any assumptions, via the chain rule:

$$P(o, a, \ell_o, s) = P(o, a, \ell_o | s) \times P(s)$$

the IBM model 1 assumptions are all about $P(o, a, \ell_o | s)$. The assumptions can be shown by a succession of applications of the chain rule concerning $(o, a, \ell_o)$

- concerning $\ell_o$, still without any particular assumptions

$$P(o, a, \ell_o | s) = P(o, a | \ell_o, s) \times p(\ell_o | s)$$

An assumption of IBM Model 1 is that the dependency $p(\ell_o | s)$ can be expressed as a dependency just on the length $\ell_s$, so by some distribution $p(L | \ell_s)$.

- Usually its stated that $p(L | \ell_s)$ is uniform: ie. all $L$ equally likely

- We will see in a while that for many of the vital calculations for training the model, the actually values of $p(L | \ell_s)$ are irrelevant
Alignment dependency

- we have so far

\[ P(o, a, l_o | s) = P(o, a | l_o, s) \times p(l_o | l_s) \]
Alignment dependency

- we have so far

\[ P(o, a, l_o | s) = P(o, a | l_o, s) \times p(l_o | l_s) \]

- analysing \( P(o, a | l_o, s) \), a further application of the chain rule gives

\[ P(o, a | l_o, s) = P(o | a, l_o, s) \times P(a | l_o, s) \quad (4) \]
Alignment dependency

- we have so far

\[ P(o, a, \ell_o|s) = P(o, a|\ell_o, s) \times p(\ell_o|\ell_s) \]

- analysing \( P(o, a|\ell_o, s) \), a further application of the chain rule gives

\[ P(o, a|\ell_o, s) = P(o|a, \ell_o, s) \times P(a|\ell_o, s) \]  \hspace{1cm} (4)

- The next assumption is that the dependency \( P(a|\ell_o, s) \) can be expressed as dependency just on \( \ell_s \) and \( \ell_o \),
Alignment dependency

we have so far

\[ P(o, a, \ell_o | s) = P(o, a | \ell_o, s) \times p(\ell_o | \ell_s) \]

analysing \( P(o, a | \ell_o, s) \), a further application of the chain rule gives

\[ P(o, a | \ell_o, s) = P(o | a, \ell_o, s) \times P(a | \ell_o, s) \tag{4} \]

The next assumption is that the dependency \( P(a | \ell_o, s) \) can be expressed as dependency just on \( \ell_s \) and \( \ell_o \), and furthermore that the distribution of possible alignments from length \( \ell_o \) sequences to length \( \ell_s \) sequences is a uniform distribution
Alignment dependency

▶ we have so far

\[
P(o, a, \ell_o | s) = P(o, a | \ell_o, s) \times p(\ell_o | \ell_s)
\]

▶ analysing \(P(o, a | \ell_o, s)\), a further application of the chain rule gives

\[
P(o, a | \ell_o, s) = P(o | a, \ell_o, s) \times P(a | \ell_o, s)
\] (4)

▶ The next assumption is that the dependency \(P(a | \ell_o, s)\) can be expressed as dependency just on \(\ell_s\) and \(\ell_o\), and furthermore that the distribution of possible alignments from length \(\ell_o\) sequences to length \(\ell_s\) sequences is a uniform distribution

▶ There are \(\ell_o\) members of \(o\) to be aligned, and for each there are \(\ell_s + 1\) possibilities (including NULL mappings), so there are \((\ell_s + 1)^{\ell_o}\) possible alignments,
Alignment dependency

- we have so far

\[ P(o, a, \ell_o | s) = P(o, a | \ell_o, s) \times p(\ell_o | \ell_s) \]

- analysing \( P(o, a | \ell_o, s) \), a further application of the chain rule gives

\[ P(o, a | \ell_o, s) = P(o | a, \ell_o, s) \times P(a | \ell_o, s) \quad (4) \]

- The next assumption is that the dependency \( P(a | \ell_o, s) \) can be expressed as dependency just on \( \ell_s \) and \( \ell_o \), and furthermore that the distribution of possible alignments from length \( \ell_o \) sequences to length \( \ell_s \) sequences is a uniform distribution

- There are \( \ell_o \) members of \( o \) to be aligned, and for each there are \( \ell_s + 1 \) possibilities (including NULL mappings), so there are \( (\ell_s + 1)^{\ell_o} \) possible alignments, so this means

\[ p(a | \ell_o, \ell_s) = \frac{1}{(\ell_s + 1)^{\ell_o}} \]
Observed words dependency
Observed words dependency

- this means the formula for $P(o, a|\ell_o, s)$ from (4) now looks like this

$$P(o, a|\ell_o, s) = P(o|a, \ell_o, s) \times \frac{1}{(\ell_s + 1)^{\ell_o}}$$  \hspace{1cm} (5)
Observed words dependency

- this means the formula for $P(o, a|\ell_o, s)$ from (4) now looks like this

$$P(o, a|\ell_o, s) = P(o|a, \ell_o, s) \times \frac{1}{(\ell_s + 1)\ell_o}$$

(5)

- finally concerning $P(o|a, \ell_o, s)$ it is assumed that this probability takes a particularly simple multiplicative form, with each $o_j$ treated as independent of everything else given the word in $s$ that it is aligned to, that is, $s_{a(j)}$, so
Observed words dependency

- this means the formula for $P(o, a | \ell_o, s)$ from (4) now looks like this

$$P(o, a | \ell_o, s) = P(o | a, \ell_o, s) \times \frac{1}{(\ell_s + 1)\ell_o}$$

(5)

- finally concerning $P(o | a, \ell_o, s)$ it is assumed that this probability takes a particularly simple multiplicative form, with each $o_j$ treated as independent of everything else given the word in $s$ that it is aligned to, that is, $s_{a(j)}$, so

$$p(o | a, \ell_o, s) = \prod_j [p(o_j | s_{a(j)})]$$
Observed words dependency

- this means the formula for \( P(o, a | \ell_o, s) \) from (4) now looks like this

\[
P(o, a | \ell_o, s) = P(o | a, \ell_o, s) \times \frac{1}{(\ell_s + 1)^{\ell_o}}
\]  \( (5) \)

- finally concerning \( P(o | a, \ell_o, s) \) it is assumed that this probability takes a particularly simple multiplicative form, with each \( o_j \) treated as independent of everything else given the word in \( s \) that it is aligned to, that is, \( s_{a(j)} \), so

\[
p(o | a, \ell_o, s) = \prod_j [p(o_j | s_{a(j)})]
\]

- and \( P(o, a | \ell_o, s) \) becomes

\[
P(o, a | \ell_o, s) = \prod_j [p(o_j | s_{a(j)})] \times \frac{1}{(\ell_s + 1)^{\ell_o}}
\]  \( (6) \)
The final IBM Model 1 formula
The final IBM Model 1 formula

\[ P(o, a, \ell_o | s) = \prod_j [p(o_j | s_{a(j)})] \times \frac{1}{(\ell_s + 1)^{\ell_o}} \times p(\ell_o | \ell_s) \]
The final IBM Model 1 formula

\[ P(o, a, \ell_o|s) = \prod_j [p(o_j|s_{a(j)})] \times \frac{1}{(\ell_s + 1)^{\ell_o}} \times p(\ell_o|\ell_s) \]

or slightly more compactly

\[ P(o, a, \ell_o|s) = \frac{p(\ell_o|\ell_s)}{(\ell_s + 1)^{\ell_o}} \times \prod_j [p(o_j|s_{a(j)})] \] (7)
Another way to arrive at the formula is via the following so-called 'generative story' for generating \( o \) from \( s \)
Another way to arrive at the formula is via the following so-called 'generative story' for generating $\mathbf{o}$ from $\mathbf{s}$

1. choose a length $\ell_o$, according to a distribution $p(\ell_o|\ell_s)$
the 'generative' story

Another way to arrive at the formula is via the following so-called 'generative story' for generating $o$ from $s$

1. choose a length $l_o$, according to a distribution $p(l_o \mid l_s)$
2. choose an alignment $a$ from $1 \ldots l_o$ to $0, 1, \ldots l_s$, according to distribution $p(a \mid l_s, l_o) = \frac{1}{(l_s+1)l_o}$
the 'generative' story

Another way to arrive at the formula is via the following so-called 'generative story' for generating \( o \) from \( s \):

1. choose a length \( \ell_o \), according to a distribution \( p(\ell_o|\ell_s) \)
2. choose an alignment \( a \) from \( 1 \ldots \ell_o \) to \( 0, 1, \ldots \ell_s \), according to distribution \( p(a|\ell_s, \ell_o) = \frac{1}{(\ell_s+1)\ell_o} \)
3. for \( j = 1 \) to \( j = \ell_o \), choose \( o_j \) according to distribution \( p(o_j|s_{a(j)}) \)
Example

\footnote{see p87 Koehn book}
Example

Suppose $s$ is *das haus ist klein* and $o$ is *the house is small*. Recall the alignment from $o$ to $s$ shown earlier:

```
<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>das</td>
<td>Haus</td>
<td>ist</td>
<td>klein</td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>house</td>
<td>is</td>
<td>small</td>
</tr>
</tbody>
</table>
```

$a: \{1 \to 1, 2 \to 2, 3 \to 3, 4 \to 4\}$

---

1 see p87 Koehn book
Suppose $s$ is \textit{das haus ist klein} and $o$ is \textit{the house is small}. Recall the alignment from $o$ to $s$ shown earlier:

\begin{align*}
a : \{1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 4\}
\end{align*}

we will illustrate the value of $p(o, a, \ell_o|s)$ in this case, according to the formula (7)

\begin{align*}
P(o, a, \ell_o|s) &= \frac{p(\ell_o|\ell_s)}{(\ell_s + 1)^{\ell_o}} \times \prod_j [p(o_j|s_{a(j)})]
\end{align*}

\[\text{see p87 Koehn book}\]
Example cntd
Example cntd

suppose following tables giving $t(e|g)$ for various German and English words

<table>
<thead>
<tr>
<th></th>
<th>das</th>
<th></th>
<th>Haus</th>
<th></th>
<th>ist</th>
<th></th>
<th>klein</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$t(e</td>
<td>g)$</td>
<td></td>
<td>$t(e</td>
<td>g)$</td>
<td></td>
<td>$t(e</td>
</tr>
<tr>
<td>$e$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>the</td>
<td>0.7</td>
<td></td>
<td>house</td>
<td>0.8</td>
<td></td>
<td>is</td>
<td>0.8</td>
</tr>
<tr>
<td>that</td>
<td>0.15</td>
<td></td>
<td>building</td>
<td>0.16</td>
<td></td>
<td>'s</td>
<td>0.16</td>
</tr>
<tr>
<td>which</td>
<td>0.075</td>
<td></td>
<td>home</td>
<td>0.02</td>
<td></td>
<td>exists</td>
<td>0.02</td>
</tr>
<tr>
<td>who</td>
<td>0.05</td>
<td></td>
<td>household</td>
<td>0.015</td>
<td></td>
<td>has</td>
<td>0.015</td>
</tr>
<tr>
<td>this</td>
<td>0.025</td>
<td></td>
<td>shell</td>
<td>0.005</td>
<td></td>
<td>are</td>
<td>0.005</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>small</td>
<td>0.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>little</td>
<td>0.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>short</td>
<td>0.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>minor</td>
<td>0.06</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>petty</td>
<td>0.04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example cntd

suppose following tables giving $t(e|g)$ for various German and English words

| German | English  | $t(e|g)$ |
|--------|----------|----------|
| das    | the      | 0.7      |
|        | that     | 0.15     |
|        | which    | 0.075    |
|        | who      | 0.05     |
|        | this     | 0.025    |
| Haus   | house    | 0.8      |
|        | building | 0.16     |
|        | home     | 0.02     |
|        | household| 0.015    |
|        | shell    | 0.005    |
| ist    | is       | 0.8      |
|        | ’s       | 0.16     |
|        | exists   | 0.02     |
|        | has      | 0.015    |
|        | are      | 0.005    |
| klein  | small    | 0.4      |
|        | little   | 0.4      |
|        | short    | 0.1      |
|        | minor    | 0.06     |
|        | petty    | 0.04     |

let $\epsilon$ represent the $P(\ell_o = 4|\ell_s = 4)$ term
Example cntd

suppose following tables giving $t(e|g)$ for various German and English words

<table>
<thead>
<tr>
<th>das</th>
<th>Haus</th>
<th>ist</th>
<th>klein</th>
</tr>
</thead>
<tbody>
<tr>
<td>e</td>
<td>t(e</td>
<td>g)</td>
<td>e</td>
</tr>
<tr>
<td>the</td>
<td>0.7</td>
<td>house</td>
<td>0.8</td>
</tr>
<tr>
<td>that</td>
<td>0.15</td>
<td>building</td>
<td>0.16</td>
</tr>
<tr>
<td>which</td>
<td>0.075</td>
<td>home</td>
<td>0.02</td>
</tr>
<tr>
<td>who</td>
<td>0.05</td>
<td>household</td>
<td>0.015</td>
</tr>
<tr>
<td>this</td>
<td>0.025</td>
<td>shell</td>
<td>0.005</td>
</tr>
</tbody>
</table>

let $\epsilon$ represent the $P(\ell_0 = 4|\ell_s = 4)$ term

$$p(o, a, \ell_0|s) = \frac{\epsilon}{5^4} \times t(\text{the}|\text{das}) \times t(\text{house}|\text{Haus}) \times t(\text{is}|\text{ist}) \times t(\text{small}|\text{klein})$$
Example cntd

suppose following tables giving $t(e|g)$ for various German and English words

| $e$ | $t(e|g)$ | $e$ | $t(e|g)$ | $e$ | $t(e|g)$ | $e$ | $t(e|g)$ |
|-----|---------|-----|---------|-----|---------|-----|---------|
| the | 0.7     | house | 0.8     | is  | 0.8     | small | 0.4     |
| that| 0.15    | building | 0.16   | ’s  | 0.16    | little | 0.4     |
| which| 0.075  | home   | 0.02    | exists | 0.02   | short | 0.1     |
| who | 0.05    | household | 0.015  | has  | 0.015   | minor | 0.06    |
| this| 0.025   | shell  | 0.005   | are  | 0.005   | petty | 0.04    |

let $\epsilon$ represent the $P(\ell_o = 4|\ell_s = 4)$ term

$$p(o, a, \ell_o|s) = \frac{\epsilon}{5^4} \times t(\text{the}|\text{das}) \times t(\text{house}|\text{Haus}) \times t(\text{is}|\text{ist}) \times t(\text{small}|\text{klein})$$
Example cntd

suppose following tables giving $t(e|g)$ for various German and English words

<table>
<thead>
<tr>
<th>das</th>
<th>Haus</th>
<th>ist</th>
<th>klein</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e$</td>
<td>$e$</td>
<td>$e$</td>
<td>$e$</td>
</tr>
<tr>
<td>the</td>
<td>house</td>
<td>is</td>
<td>small</td>
</tr>
<tr>
<td>that</td>
<td>building</td>
<td>'s</td>
<td>little</td>
</tr>
<tr>
<td>which</td>
<td>home</td>
<td>exists</td>
<td>short</td>
</tr>
<tr>
<td>who</td>
<td>household</td>
<td>has</td>
<td>minor</td>
</tr>
<tr>
<td>this</td>
<td>shell</td>
<td>are</td>
<td>petty</td>
</tr>
</tbody>
</table>

let $\epsilon$ represent the $P(\ell_o = 4|\ell_s = 4)$ term

\[
p(o, a, \ell_o|s) = \frac{\epsilon}{5^4} \times t(\text{the}|\text{das}) \times t(\text{house}|\text{Haus}) \times t(\text{is}|\text{ist}) \times t(\text{small}|\text{klein})
\]

\[
= \frac{\epsilon}{5^4} \times 0.7 \times 0.8 \times 0.8 \times 0.4
\]

\[
= 0.00028672 \epsilon
\]