4CSLL5
IBM Translation Models

Martin Emms

October 4, 2018

IBM models intro

Outline

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

Translation of Haus	**Count**
house	8,000
building	1,600
home	200
household	150
shell	50

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

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

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

  ⋮

  das Haus ist klein *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

  ⋮

  das Haus ist klein *the house is small*

  ⋮

- 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

Notation

- For reasons that will become apparent, we will use
  $\mathcal{O}$ for the language we want to translate *from*
  $\mathcal{S}$ for the language we want to translate *to*

Notation

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

Notation

- For reasons that will become apparent, we will use
  $\mathcal{O}$ for the language we want to translate *from*
  $\mathcal{S}$ for the language we want to translate *to*
- **o** is a single sentence from $\mathcal{O}$, and is a sequence $(o_1 \ldots o_j \ldots o_{\ell_{\mathbf{o}}})$; $\ell_{\mathbf{o}}$ is length **o**
- **s** is a single sentence from $\mathcal{S}$, and is a sequence $(s_1 \ldots s_i \ldots s_{\ell_{\mathbf{s}}})$; $\ell_{\mathbf{s}}$ is length **o**

Notation

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

Notation

- For reasons that will become apparent, we will use
  $\mathcal{O}$ for the language we want to translate *from*
  $\mathcal{S}$ for the language we want to translate *to*
- **o** is a single sentence from $\mathcal{O}$, and is a sequence $(o_1 \ldots o_j \ldots o_{\ell_\mathbf{o}})$; $\ell_\mathbf{o}$ is length **o**
- **s** is a single sentence from $\mathcal{S}$, and is a sequence $(s_1 \ldots s_i \ldots s_{\ell_\mathbf{s}})$; $\ell_\mathbf{s}$ is length **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 $\mathcal{V}_s$

Notation

- For reasons that will become apparent, we will use
  $\mathcal{O}$ for the language we want to translate *from*
  $\mathcal{S}$ for the language we want to translate *to*
- **o** is a single sentence from $\mathcal{O}$, and is a sequence $(o_1 \dots o_j \dots o_{\ell_{\mathbf{o}}})$; $\ell_{\mathbf{o}}$ is length **o**
- **s** is a single sentence from $\mathcal{S}$, and is a sequence $(s_1 \dots s_i \dots s_{\ell_{\mathbf{s}}})$; $\ell_{\mathbf{s}}$ is length **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 $\mathcal{O}$ and $\mathcal{S}$.

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 $\mathbf{o} \in \mathcal{O}$ 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 $\mathcal{O}$ and $\mathcal{S}$.
- in principle for any sentence $\mathbf{o} \in \mathcal{O}$ could work out the probabilities of its various translations **s** by relative frequency

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

The sparsity problem

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

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

- but even in very large corpora the vast majority of possible $\mathbf{o}$ and $\mathbf{s}$ occur zero times. So this method gives uselessly bad estimates.

The Noisy-Channel formulation

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

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

The Noisy-Channel formulation

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

$$\arg\max_{\mathbf{s}} P(\mathbf{s}|\mathbf{o}) = \arg\max_{\mathbf{s}} \frac{P(\mathbf{s},\mathbf{o})}{P(\mathbf{o})} \quad (1)$$

$$= \arg\max_{\mathbf{s}} P(\mathbf{s},\mathbf{o}) \quad (2)$$

The Noisy-Channel formulation

- recalling Bayesian classification, finding $\mathbf{s}$ from $\mathbf{o}$:

$$\arg\max_{\mathbf{s}} P(\mathbf{s}|\mathbf{o}) = \arg\max_{\mathbf{s}} \frac{P(\mathbf{s},\mathbf{o})}{P(\mathbf{o})} \quad (1)$$

$$= \arg\max_{\mathbf{s}} P(\mathbf{s},\mathbf{o}) \quad (2)$$

$$= \arg\max_{\mathbf{s}} P(\mathbf{o}|\mathbf{s}) \times P(\mathbf{s}) \quad (3)$$

The Noisy-Channel formulation

- recalling Bayesian classification, finding $\mathbf{s}$ from $\mathbf{o}$:

$$\begin{aligned}
\arg\max_{\mathbf{s}} P(\mathbf{s}|\mathbf{o}) &= \arg\max_{\mathbf{s}} \frac{P(\mathbf{s},\mathbf{o})}{P(\mathbf{o})} && (1)\\
&= \arg\max_{\mathbf{s}} P(\mathbf{s},\mathbf{o}) && (2)\\
&= \arg\max_{\mathbf{s}} P(\mathbf{o}|\mathbf{s}) \times P(\mathbf{s}) && (3)
\end{aligned}$$

- can then try to factorise $P(\mathbf{o}|\mathbf{s})$ and $P(\mathbf{s})$ 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**:

$$\begin{aligned} \arg\max_{\mathbf{s}} P(\mathbf{s}|\mathbf{o}) &= \arg\max_{\mathbf{s}} \frac{P(\mathbf{s},\mathbf{o})}{P(\mathbf{o})} && (1) \\ &= \arg\max_{\mathbf{s}} P(\mathbf{s},\mathbf{o}) && (2) \\ &= \arg\max_{\mathbf{s}} P(\mathbf{o}|\mathbf{s}) \times P(\mathbf{s}) && (3) \end{aligned}$$

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

The Noisy-Channel formulation

- recalling Bayesian classification, finding $\mathbf{s}$ from $\mathbf{o}$:

$$\begin{aligned}
\arg\max_{\mathbf{s}} P(\mathbf{s}|\mathbf{o}) &= \arg\max_{\mathbf{s}} \frac{P(\mathbf{s},\mathbf{o})}{P(\mathbf{o})} && (1)\\
&= \arg\max_{\mathbf{s}} P(\mathbf{s},\mathbf{o}) && (2)\\
&= \arg\max_{\mathbf{s}} P(\mathbf{o}|\mathbf{s}) \times P(\mathbf{s}) && (3)
\end{aligned}$$

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

The Noisy-Channel formulation

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

$$\begin{aligned} \arg\max_{\mathbf{s}} P(\mathbf{s}|\mathbf{o}) &= \arg\max_{\mathbf{s}} \frac{P(\mathbf{s},\mathbf{o})}{P(\mathbf{o})} && (1) \\ &= \arg\max_{\mathbf{s}} P(\mathbf{s},\mathbf{o}) && (2) \\ &= \arg\max_{\mathbf{s}} P(\mathbf{o}|\mathbf{s}) \times P(\mathbf{s}) && (3) \end{aligned}$$

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

Now have to start look at the details of the IBM models of $P(\mathbf{o}|\mathbf{s})$, starting with the very simplest models

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

Outline

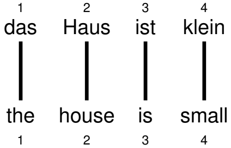
Alignments (informally)

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.

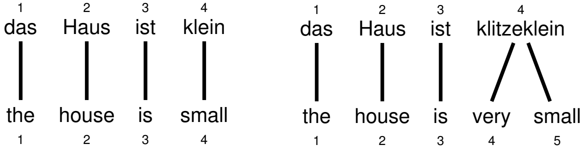
Alignments (informally)

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



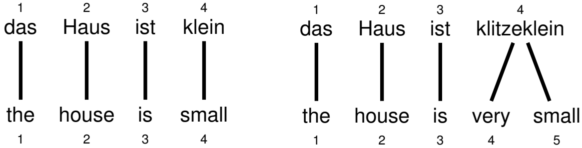
Alignments (informally)

- 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)

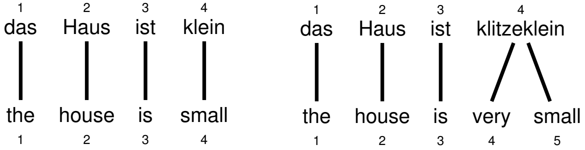
- When **s** and **o** are translations of each other, usually can say which pieces of **s** and **o** are translations of each other. eg.



- 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.



- In SMT such a piece-wise correspondence is called an 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 $\mathbf{o}$ and $\mathbf{s}$
- so a pair $\langle \mathbf{o}, \mathbf{s} \rangle$ from a sentence-aligned corpus is seen as a partial version of the fully observed case:

$$\langle \mathbf{o}, a, \mathbf{s} \rangle$$

Hidden Alignment

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

$$\langle \mathbf{o}, a, \mathbf{s} \rangle$$

- A model is essentially made of $p(\mathbf{o}, a|\mathbf{s})$, 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 $\mathbf{o}$ and $\mathbf{s}$
- so a pair $\langle \mathbf{o}, \mathbf{s} \rangle$ from a sentence-aligned corpus is seen as a partial version of the fully observed case:

$$\langle \mathbf{o}, a, \mathbf{s} \rangle$$

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

$$\arg\max_{\mathbf{s}} P(\mathbf{s}, \mathbf{o}) = \arg\max_{\mathbf{s}}([\sum_{a} p(\mathbf{o}, a|\mathbf{s})] \times p(\mathbf{s}))$$

Hidden Alignment

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

$$\langle \mathbf{o}, a, \mathbf{s} \rangle$$

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

$$\arg\max_{\mathbf{s}} P(\mathbf{s}, \mathbf{o}) = \arg\max_{\mathbf{s}}([\sum_a p(\mathbf{o}, a|\mathbf{s})] \times p(\mathbf{s}))$$

- best alignment:

$$\arg\max_a [p(\mathbf{o}, a|\mathbf{s})]$$

IBM Alignments

- Define alignment with a function,

IBM Alignments

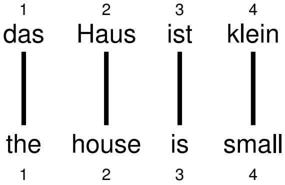
- Define alignment with a function,
  from posn j in $\mathbf{o}$ to posn. i in $\mathbf{s}$

IBM Alignments

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

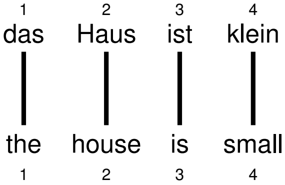
IBM Alignments

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



IBM Alignments

- Define alignment with a function,
  from posn j in **o** to posn. i in **s**
  so $a : j \to i$
- the picture



represents

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

Some weirdness about directions

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

$$a: \begin{array}{l} 1 \to 1, \\ 2 \to 2, \\ 3 \to 3, \\ 4 \to 4 \end{array}$$

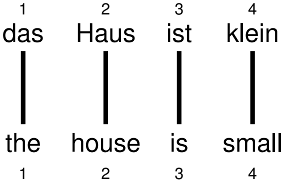
Some weirdness about directions

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

$$a: \begin{array}{l} 1 \rightarrow 1, \\ 2 \rightarrow 2, \\ 3 \rightarrow 3, \\ 4 \rightarrow 4 \end{array}$$

- Note here **o** is English, and **s** is German

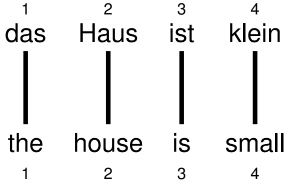
Some weirdness about directions



$$a: \begin{array}{l} 1 \rightarrow 1, \\ 2 \rightarrow 2, \\ 3 \rightarrow 3, \\ 4 \rightarrow 4 \end{array}$$

- Note here **o** is English, and **s** is German
- the alignment goes up the page, English-to-German,

Some weirdness about directions



$$a: \begin{array}{l} 1 \to 1, \\ 2 \to 2, \\ 3 \to 3, \\ 4 \to 4 \end{array}$$

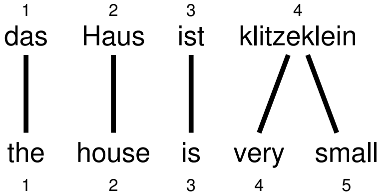
- Note here **o** is English, and **s** is German
- the alignment goes up the page, English-to-German,
- they will be used though in a model of $P(\mathbf{o}|\mathbf{s})$, so down the page, German-to-English

Comparison to 'edit distance' alignments

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

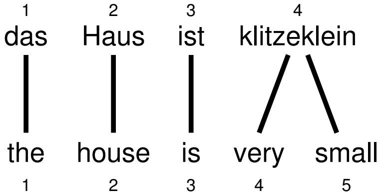
- like edit-dist alignments, its a function:
  so can't align 1 **o** words with 2 **s** words
- like edit-dist alignments, some **s** words can be unmapped to
  (cf. insertions)
- like edit-dist alignments, some **o** words can be mapped to nothing
  (cf. deletions)
- unlike edit-dist alignments, order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

N-to-1 Alignment (ie. 1-to-N Translation)



- $a : \{1 \to 1, 2 \to 2, 3 \to 3, 4 \to 4, 5 \to 4\}$

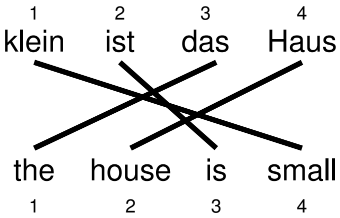
N-to-1 Alignment (ie. 1-to-N Translation)



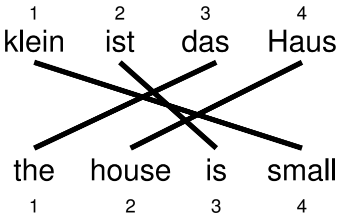
- $a : \{1 \to 1, 2 \to 2, 3 \to 3, 4 \to 4, 5 \to 4\}$
- N words of **o** can be aligned to 1 word of **s**

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

Reordering

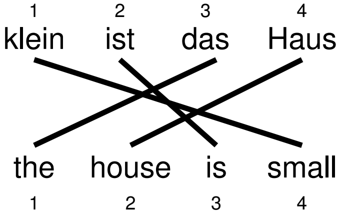


Reordering



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

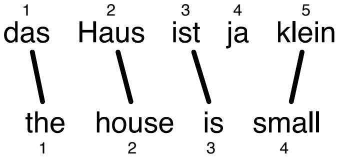
Reordering



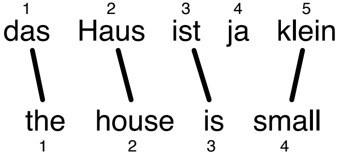
- $a : \{1 \rightarrow 3, 2 \rightarrow 4, 3 \rightarrow 2, 4 \rightarrow 1\}$
- alignment does not preserve **o** word order

  (needed when **s** words reordered during translation)

s words not mapped to (ie. dropped in translation)



s words not mapped to (ie. dropped in translation)

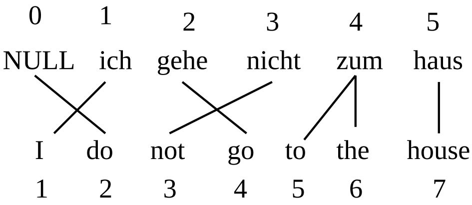


- $a : \{1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 5\}$
- some **s** words are not mapped-to by the alignment

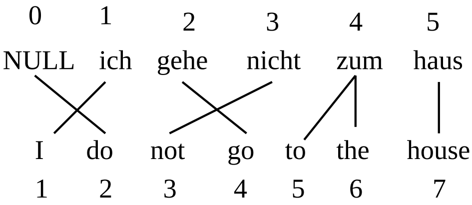
  (needed when **s** words are dropped during translation

  (here the German flavouring particle 'ja' is dropped)

- words mapped to nothing (ie. inserting in translation)

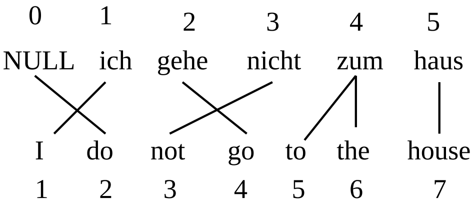


words mapped to nothing (ie. inserting in translation)



- $a : \{1 \to 1, 2 \to 0, 3 \to 3, 4 \to 2, 5 \to 4, 6 \to 4, 7 \to 5\}$

o words mapped to nothing (ie. inserting in translation)



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

IBM Model 1

- basically a hidden variable a, aligning $\mathbf{o}$ to $\mathbf{s}$ is assumed.

IBM Model 1

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

$$P(\mathbf{o}, a, L, \mathbf{s})$$

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

IBM Model 1

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

$$P(\mathbf{o}, a, L, \mathbf{s})$$

where L is length for $\mathbf{o}$ sentences, and a is an alignment from $\mathbf{o}$ sentences of length L to $\mathbf{s}$.
- $\mathbf{o}$, a, L are intended to be synchronized in the sense that if L is not the $\ell_{\mathbf{o}}$ the probability is zero. Similarly if a is not an alignment function from length L sequences to length $\ell_{\mathbf{s}}$ sequences, the probability is 0. So we will write

$$P(\mathbf{o}, a, \ell_{\mathbf{o}}, \mathbf{s})$$

Length dependency

Length dependency

- first without any assumptions, via the chain rule:

$$P(\mathbf{o}, a, \ell_{\mathbf{o}}, \mathbf{s}) = P(\mathbf{o}, a, \ell_{\mathbf{o}} | \mathbf{s}) \times P(\mathbf{s})$$

Length dependency

- first without any assumptions, via the chain rule:

$$P(\mathbf{o}, a, \ell_{\mathbf{o}}, \mathbf{s}) = P(\mathbf{o}, a, \ell_{\mathbf{o}}|\mathbf{s}) \times P(\mathbf{s})$$

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

Length dependency

- first without any assumptions, via the chain rule:

$$P(\mathbf{o}, a, \ell_{\mathbf{o}}, \mathbf{s}) = P(\mathbf{o}, a, \ell_{\mathbf{o}}|\mathbf{s}) \times P(\mathbf{s})$$

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

- concerning $\ell_{\mathbf{o}}$, still without any particular assumptions

$$P(\mathbf{o}, a, \ell_{\mathbf{o}}|\mathbf{s}) = P(\mathbf{o}, a|\ell_{\mathbf{o}}, \mathbf{s}) \times p(\ell_{\mathbf{o}}|\mathbf{s})$$

Length dependency

- first without any assumptions, via the chain rule:

$$P(\mathbf{o}, a, \ell_{\mathbf{o}}, \mathbf{s}) = P(\mathbf{o}, a, \ell_{\mathbf{o}}|\mathbf{s}) \times P(\mathbf{s})$$

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

- concerning $\ell_{\mathbf{o}}$, still without any particular assumptions

$$P(\mathbf{o}, a, \ell_{\mathbf{o}}|\mathbf{s}) = P(\mathbf{o}, a|\ell_{\mathbf{o}}, \mathbf{s}) \times p(\ell_{\mathbf{o}}|\mathbf{s})$$

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

Length dependency

- first without any assumptions, via the chain rule:

$$P(\mathbf{o}, a, \ell_{\mathbf{o}}, \mathbf{s}) = P(\mathbf{o}, a, \ell_{\mathbf{o}}|\mathbf{s}) \times P(\mathbf{s})$$

  the IBM model1 assumptions are all about $P(\mathbf{o}, a, \ell_{\mathbf{o}}|\mathbf{s})$. The assumptions can be shown by a succession of applications of the chain rule concerning $(\mathbf{o}, a, \ell_{\mathbf{o}})$
- concerning $\ell_{\mathbf{o}}$, still without any particular assumptions

$$P(\mathbf{o}, a, \ell_{\mathbf{o}}|\mathbf{s}) = P(\mathbf{o}, a|\ell_{\mathbf{o}}, \mathbf{s}) \times p(\ell_{\mathbf{o}}|\mathbf{s})$$

  An assumption of IBM Model 1 is that the dependency $p(\ell_{\mathbf{o}}|\mathbf{s})$ can be expressed as a dependency just on the length $\ell_{\mathbf{s}}$, so by some distribution $p(L|\ell_{\mathbf{s}})$.
- Usually its stated that $p(L|\ell_{\mathbf{s}})$ is uniform: ie. all L equally likely

Length dependency

- first without any assumptions, via the chain rule:

$$P(\mathbf{o}, a, \ell_{\mathbf{o}}, \mathbf{s}) = P(\mathbf{o}, a, \ell_{\mathbf{o}}|\mathbf{s}) \times P(\mathbf{s})$$

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

- concerning $\ell_{\mathbf{o}}$, still without any particular assumptions

$$P(\mathbf{o}, a, \ell_{\mathbf{o}}|\mathbf{s}) = P(\mathbf{o}, a|\ell_{\mathbf{o}}, \mathbf{s}) \times p(\ell_{\mathbf{o}}|\mathbf{s})$$

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

- Usually its stated that $p(L|\ell_{\mathbf{s}})$ 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 $p(L|\ell_{\mathbf{s}})$ are irrelevant

Alignment dependency

- we have so far

$$P(\mathbf{o}, a, \ell_{\mathbf{o}}|\mathbf{s}) = P(\mathbf{o}, a|\ell_{\mathbf{o}}, \mathbf{s}) \times p(\ell_{\mathbf{o}}|\ell_{\mathbf{s}})$$

Alignment dependency

- we have so far

$$P(\mathbf{o}, a, \ell_{\mathbf{o}}|\mathbf{s}) = P(\mathbf{o}, a|\ell_{\mathbf{o}}, \mathbf{s}) \times p(\ell_{\mathbf{o}}|\ell_{\mathbf{s}})$$

- analysing $P(\mathbf{o}, a|\ell_{\mathbf{o}}, \mathbf{s})$, a further application of the chain rule gives

$$P(\mathbf{o}, a|\ell_{\mathbf{o}}, \mathbf{s}) = P(\mathbf{o}|a, \ell_{\mathbf{o}}, \mathbf{s}) \times P(a|\ell_{\mathbf{o}}, \mathbf{s}) \tag{4}$$

Alignment dependency

- we have so far

$$P(\mathbf{o}, a, \ell_{\mathbf{o}}|\mathbf{s}) = P(\mathbf{o}, a|\ell_{\mathbf{o}}, \mathbf{s}) \times p(\ell_{\mathbf{o}}|\ell_{\mathbf{s}})$$

- analysing $P(\mathbf{o}, a|\ell_{\mathbf{o}}, \mathbf{s})$, a further application of the chain rule gives

$$P(\mathbf{o}, a|\ell_{\mathbf{o}}, \mathbf{s}) = P(\mathbf{o}|a, \ell_{\mathbf{o}}, \mathbf{s}) \times P(a|\ell_{\mathbf{o}}, \mathbf{s}) \quad (4)$$

- The next assumption is that the dependency $P(a|\ell_{\mathbf{o}}, \mathbf{s})$ can be expressed as dependency just on $\ell_{\mathbf{s}}$ and $\ell_{\mathbf{o}}$,

Alignment dependency

- we have so far

$$P(\mathbf{o}, a, \ell_{\mathbf{o}}|\mathbf{s}) = P(\mathbf{o}, a|\ell_{\mathbf{o}}, \mathbf{s}) \times p(\ell_{\mathbf{o}}|\ell_{\mathbf{s}})$$

- analysing $P(\mathbf{o}, a|\ell_{\mathbf{o}}, \mathbf{s})$, a further application of the chain rule gives

$$P(\mathbf{o}, a|\ell_{\mathbf{o}}, \mathbf{s}) = P(\mathbf{o}|a, \ell_{\mathbf{o}}, \mathbf{s}) \times P(a|\ell_{\mathbf{o}}, \mathbf{s}) \quad (4)$$

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

Alignment dependency

- we have so far

$$P(\mathbf{o}, a, \ell_{\mathbf{o}}|\mathbf{s}) = P(\mathbf{o}, a|\ell_{\mathbf{o}}, \mathbf{s}) \times p(\ell_{\mathbf{o}}|\ell_{\mathbf{s}})$$

- analysing $P(\mathbf{o}, a|\ell_{\mathbf{o}}, \mathbf{s})$, a further application of the chain rule gives

$$P(\mathbf{o}, a|\ell_{\mathbf{o}}, \mathbf{s}) = P(\mathbf{o}|a, \ell_{\mathbf{o}}, \mathbf{s}) \times P(a|\ell_{\mathbf{o}}, \mathbf{s}) \tag{4}$$

- The next assumption is that the dependency $P(a|\ell_{\mathbf{o}}, \mathbf{s})$ can be expressed as dependency just on $\ell_{\mathbf{s}}$ and $\ell_{\mathbf{o}}$, and furthermore that the distribution of possible alignments from length $\ell_{\mathbf{o}}$ sequences to length $\ell_{\mathbf{s}}$ sequences is a uniform distribution
- There are $\ell_{\mathbf{o}}$ members of $\mathbf{o}$ to be aligned, and for each there are $\ell_{\mathbf{s}} + 1$ possibilities (including NULL mappings), so there are $(\ell_{\mathbf{s}} + 1)^{\ell_{\mathbf{o}}}$ possible alignments,

Alignment dependency

- we have so far

$$P(\mathbf{o}, a, \ell_{\mathbf{o}}|\mathbf{s}) = P(\mathbf{o}, a|\ell_{\mathbf{o}}, \mathbf{s}) \times p(\ell_{\mathbf{o}}|\ell_{\mathbf{s}})$$

- analysing $P(\mathbf{o}, a|\ell_{\mathbf{o}}, \mathbf{s})$, a further application of the chain rule gives

$$P(\mathbf{o}, a|\ell_{\mathbf{o}}, \mathbf{s}) = P(\mathbf{o}|a, \ell_{\mathbf{o}}, \mathbf{s}) \times P(a|\ell_{\mathbf{o}}, \mathbf{s}) \tag{4}$$

- The next assumption is that the dependency $P(a|\ell_{\mathbf{o}}, \mathbf{s})$ can be expressed as dependency just on $\ell_{\mathbf{s}}$ and $\ell_{\mathbf{o}}$, and furthermore that the distribution of possible alignments from length $\ell_{\mathbf{o}}$ sequences to length $\ell_{\mathbf{s}}$ sequences is a uniform distribution
- There are $\ell_{\mathbf{o}}$ members of $\mathbf{o}$ to be aligned, and for each there are $\ell_{\mathbf{s}} + 1$ possibilities (including NULL mappings), so there are $(\ell_{\mathbf{s}} + 1)^{\ell_{\mathbf{o}}}$ possible alignments, so this means

$$p(a|\ell_{\mathbf{o}}, \ell_{\mathbf{s}}) = \frac{1}{(\ell_{\mathbf{s}} + 1)^{\ell_{\mathbf{o}}}}$$

Observed words dependency

Observed words dependency

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

$$P(\mathbf{o}, a|\ell_{\mathbf{o}}, \mathbf{s}) = P(\mathbf{o}|a, \ell_{\mathbf{o}}, \mathbf{s}) \times \frac{1}{(\ell_{\mathbf{s}} + 1)^{\ell_{\mathbf{o}}}} \tag{5}$$

Observed words dependency

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

$$P(\mathbf{o}, a|\ell_{\mathbf{o}}, \mathbf{s}) = P(\mathbf{o}|a, \ell_{\mathbf{o}}, \mathbf{s}) \times \frac{1}{(\ell_{\mathbf{s}} + 1)^{\ell_{\mathbf{o}}}} \tag{5}$$

- finally concerning $P(\mathbf{o}|a, \ell_{\mathbf{o}}, \mathbf{s})$ 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 $\mathbf{s}$ that it is aligned to, that is, $s_{a(j)}$, so

Observed words dependency

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

$$P(\mathbf{o}, a|\ell_{\mathbf{o}}, \mathbf{s}) = P(\mathbf{o}|a, \ell_{\mathbf{o}}, \mathbf{s}) \times \frac{1}{(\ell_{\mathbf{s}} + 1)^{\ell_{\mathbf{o}}}} \tag{5}$$

- finally concerning $P(\mathbf{o}|a, \ell_{\mathbf{o}}, \mathbf{s})$ 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 $\mathbf{s}$ that it is aligned to, that is, $s_{a(j)}$, so

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

Observed words dependency

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

$$P(\mathbf{o}, a|\ell_{\mathbf{o}}, \mathbf{s}) = P(\mathbf{o}|a, \ell_{\mathbf{o}}, \mathbf{s}) \times \frac{1}{(\ell_{\mathbf{s}} + 1)^{\ell_{\mathbf{o}}}} \tag{5}$$

- finally concerning $P(\mathbf{o}|a, \ell_{\mathbf{o}}, \mathbf{s})$ 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 $\mathbf{s}$ that it is aligned to, that is, $s_{a(j)}$, so

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

- and $P(\mathbf{o}, a|\ell_{\mathbf{o}}, \mathbf{s})$ becomes

$$P(\mathbf{o}, a|\ell_{\mathbf{o}}, \mathbf{s}) = \prod_j [p(o_j|s_{a(j)})] \times \frac{1}{(\ell_{\mathbf{s}} + 1)^{\ell_{\mathbf{o}}}} \tag{6}$$

The final IBM Model 1 formula

The final IBM Model 1 formula

$$P(\mathbf{o}, a, \ell_{\mathbf{o}}|\mathbf{s}) = \prod_{j}[p(o_j|s_{a(j)})] \times \frac{1}{(\ell_{\mathbf{s}}+1)^{\ell_{\mathbf{o}}}} \times p(\ell_{\mathbf{o}}|\ell_{\mathbf{s}})$$

The final IBM Model 1 formula

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

or slightly more compactly

$$P(\mathbf{o}, a, \ell_{\mathbf{o}} | \mathbf{s}) = \frac{p(\ell_{\mathbf{o}} | \ell_{\mathbf{s}})}{(\ell_{\mathbf{s}} + 1)^{\ell_{\mathbf{o}}}} \times \prod_j [p(o_j | s_{a(j)})] \tag{7}$$

the 'generative' story

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

the 'generative' story

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 $\ell_{\mathbf{o}}$, according to a distribution $p(\ell_{\mathbf{o}}|\ell_{\mathbf{s}})$

the 'generative' story

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 $\ell_{\mathbf{o}}$, according to a distribution $p(\ell_{\mathbf{o}}|\ell_{\mathbf{s}})$
2. choose an alignment a from $1 \ldots \ell_{\mathbf{o}}$ to $0, 1, \ldots \ell_{\mathbf{s}}$, according to distribution $p(a|\ell_{\mathbf{s}}, \ell_{\mathbf{o}}) = \frac{1}{(\ell_{\mathbf{s}}+1)^{\ell_{\mathbf{o}}}}$

the 'generative' story

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

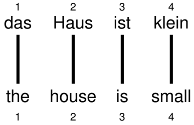
1. choose a length $\ell_{\mathbf{o}}$, according to a distribution $p(\ell_{\mathbf{o}}|\ell_{\mathbf{s}})$
2. choose an alignment a from $1 \dots \ell_{\mathbf{o}}$ to $0, 1, \dots \ell_{\mathbf{s}}$, according to distribution $p(a|\ell_{\mathbf{s}}, \ell_{\mathbf{o}}) = \frac{1}{(\ell_{\mathbf{s}}+1)^{\ell_{\mathbf{o}}}}$
3. for $j = 1$ to $j = \ell_{\mathbf{o}}$, choose o_j according to distribution $p(o_j|s_{a(j)})$

Example[1]

[1] see p87 Koehn book

Example[1]

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

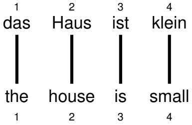


$$a : \{1 \to 1, 2 \to 2, 3 \to 3, 4 \to 4\}$$

[1] see p87 Koehn book

Example[1]

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



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

- we will illustrate the value of $p(\mathbf{o}, a, \ell_{\mathbf{o}}|\mathbf{s})$ in this case, according to the formula (7)

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

[1] see p87 Koehn book

Example cntd

Example cntd

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

das

e	$t(e\|g)$
the	0.7
that	0.15
which	0.075
who	0.05
this	0.025

Haus

e	$t(e\|g)$
house	0.8
building	0.16
home	0.02
household	0.015
shell	0.005

ist

e	$t(e\|g)$
is	0.8
's	0.16
exists	0.02
has	0.015
are	0.005

klein

e	$t(e\|g)$
small	0.4
little	0.4
short	0.1
minor	0.06
petty	0.04

Example cntd

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

das

e	$t(e\|g)$
the	0.7
that	0.15
which	0.075
who	0.05
this	0.025

Haus

e	$t(e\|g)$
house	0.8
building	0.16
home	0.02
household	0.015
shell	0.005

ist

e	$t(e\|g)$
is	0.8
's	0.16
exists	0.02
has	0.015
are	0.005

klein

e	$t(e\|g)$
small	0.4
little	0.4
short	0.1
minor	0.06
petty	0.04

let ϵ represent the $P(\ell_{\mathbf{o}} = 4|\ell_{\mathbf{s}} = 4)$ term

Example cntd

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

das

e	$t(e\|g)$
the	0.7
that	0.15
which	0.075
who	0.05
this	0.025

Haus

e	$t(e\|g)$
house	0.8
building	0.16
home	0.02
household	0.015
shell	0.005

ist

e	$t(e\|g)$
is	0.8
's	0.16
exists	0.02
has	0.015
are	0.005

klein

e	$t(e\|g)$
small	0.4
little	0.4
short	0.1
minor	0.06
petty	0.04

let ϵ represent the $P(\ell_\mathbf{o} = 4|\ell_\mathbf{s} = 4)$ term

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

Example cntd

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

das

e	$t(e\|g)$
the	0.7
that	0.15
which	0.075
who	0.05
this	0.025

Haus

e	$t(e\|g)$
house	0.8
building	0.16
home	0.02
household	0.015
shell	0.005

ist

e	$t(e\|g)$
is	0.8
's	0.16
exists	0.02
has	0.015
are	0.005

klein

e	$t(e\|g)$
small	0.4
little	0.4
short	0.1
minor	0.06
petty	0.04

let ϵ represent the $P(\ell_{\mathbf{o}} = 4|\ell_{\mathbf{s}} = 4)$ term

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

Example cntd

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

das

e	$t(e\|g)$
the	0.7
that	0.15
which	0.075
who	0.05
this	0.025

Haus

e	$t(e\|g)$
house	0.8
building	0.16
home	0.02
household	0.015
shell	0.005

ist

e	$t(e\|g)$
is	0.8
's	0.16
exists	0.02
has	0.015
are	0.005

klein

e	$t(e\|g)$
small	0.4
little	0.4
short	0.1
minor	0.06
petty	0.04

let ϵ represent the $P(\ell_{\mathbf{o}} = 4|\ell_{\mathbf{s}} = 4)$ term

$$\begin{aligned} p(\mathbf{o}, a, \ell_{\mathbf{o}}|\mathbf{s}) &= \frac{\epsilon}{5^4} \times t(the|das) \times t(house|Haus) \times t(\text{is}|ist) \times t(small|klein) \\ &= \frac{\epsilon}{5^4} \times 0.7 \times 0.8 \times 0.8 \times 0.4 \\ &= 0.00028672\epsilon \end{aligned}$$