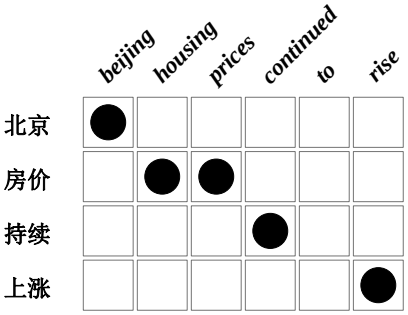
第五讲 短语翻译表构造 –概率估计

学习目标: 深刻理解基于短语统计机器翻译系统中短语翻译表概率估计的基本步骤，可自行开发概率估计模块。

短语翻译表概率估计是统计机器翻译系统构建的第四步，经过第三步短语抽取后，获得基于短语系统使用的翻译短语对，概率估计的作用是对翻译短语对的正确性进行合理的评估。

本讲学习内容:

- **示例短语对集合**



短语对	
北京	*beijing*
持续	*continued*
上涨	*rise*
房价	*housing prices*
持续	*continued to*
上涨	*to rise*
北京 房价	*beijing housing prices*
房价 持续	*housing prices continued*
持续 上涨	*continued to rise*
北京 房价 持续	*beijing housing prices continued*
房价 持续	*housing prices continued to*
北京 房价 持续	*beijing housing prices continued to*
房价 持续 上涨	*housing prices continued to rise*
北京 房价 持续 上涨	*beijing housing prices continued to rise*

图 1 示例短语对集合

概率估计主要进行四个分数的计算，即双向[1]短语翻译概率、双向词汇化权重。首先，在图 1 上方给定的含有词对齐的句对中，通过上一讲中的短语对抽取算法抽取出 14 条与词对齐保持一致的短语对，短语概率估计是在图 1 结果的基础上进行的[2]。

[1] 正向:“源语言->目标语言”方向；反向:“目标语言->源语言”方向

[2] 在进行概率估计时，短语对集合需要保留词对齐信息

- **双向短语翻译概率**

➢ “源语言->目标语言”短语翻译概率

$$\Pr(\bar{e} \mid \bar{f}) = \frac{count(\bar{f}, \bar{e})}{\sum_{\bar{e}_i} count(\bar{f}, \bar{e}_i)} \tag{1}$$

在公式(1)中，短语翻译概率使用极大似然估计（*maximum likelihood estimation*）进行计算。其中 $count(\bar{f}, \bar{e})$ 表示源言语与目标语言短语对 $(\bar{f}, \bar{e})$ 在大规模双语平行句对中出现的频次，$\sum_{\bar{e}_i} count(\bar{f}, \bar{e}_i)$ 表示以 $\bar{f}$ 作为源语言端短语的短语对在大规模双语平行句对中出现的频次。

➢ “目标语言->源语言”短语翻译概率

$$\Pr(\bar{f} \mid \bar{e}) = \frac{count(\bar{e}, \bar{f})}{\sum_{\bar{f}_j} count(\bar{e}, \bar{f}_j)} \tag{2}$$

反向的短语翻译概率与正向短语翻译概率计算方式相同，在公式(2)中，$count(\bar{e}, \bar{f})$ 表示目标语言与源语言短语对 $(\bar{e}, \bar{f})$ 在大规模双语平行句对中出现的频次，$\sum_{\bar{f}_j} count(\bar{e}, \bar{f}_j)$ 表示以 $\bar{e}$ 作为目标语言端短语的短语对在大规模双语平行句对中出现的频次。

当使用的含有词对齐信息的双语平行句对的规模比较大时，抽取出来的短语对集合文件是非常大的，文件大小甚至会达到几个 GB 或几十 GB。所以，在使用公式(1)、公式(2)计算短语翻译概率时，需要对文件进行外部排序，以避免文件内容全部加载至内存中。以公式(1)为例，如果对抽取出来的短语对集合文件按照源语言端短语进行排序，这样具有相同源语短语的短语对在文件中将是依次出现的，此时仅需要同时读入有限的短语对至内存中便可进行条件概率分布分数的计算。

在基于短语的统计机器翻译系统中，经常仅仅使用双向的短语翻译概率。在这种情况下，数据的稀疏性或不可靠的数据源可能会产生一些问题。如果短语 $\bar{e}$ 和 $\bar{f}$ 都只出现一次，那么短语翻译概率 $\Pr(\bar{e} \mid \bar{f}) = \Pr(\bar{f} \mid \bar{e}) = 1$，这通常过高的估计了这种短语对的可靠性。为了判断不经常出现的短语对是否可靠，通常做法是将短语对分解成词的翻译，这样就可以检查短语对的匹配程度，这种方法称为词汇化加权，该方法是一种基本的平滑方法。

- **双向词汇化翻译概率**

➢ “源语言->目标语言”词汇化加权

$$\Pr_{lex}(\bar{e} \mid \bar{f}, a) = \prod_{i=1}^{I} \frac{1}{\left|\{ j \mid (i, j) \in a \}\right|} \sum_{\forall (i,j) \in a} \omega(e_i \mid f_j) \tag{3}$$

词汇化加权（*lexical weighting*）特征是将源语言端和目标语言端短语分解成词汇，进而检查词汇间的匹配程度。即源语言端短语 $\bar{f}$ 中词汇 $f_1 \ldots f_J$ 与目标语言端短语 $\bar{e}$ 中词汇 $e_1 \ldots e_I$ 的匹配程度。其中 $\omega(e_i \mid f_j)$ 计算公式如公式(4)所示，该公式可以从含有词对齐的大规模平行句对中进行估计。在公式(4)中，$count(f_j, e_i)$ 表示的是词对 (f_j, e_i) 在大规模双语平行句对中出现的频次，$\sum_{e_i} count(f_j, e_i)$ 表示以 f_j 为源语言端词汇的词对在大规模语料中出现的频次。

$$\omega(e_i \mid f_j) = \frac{count(f_j, e_i)}{\sum_{e_i} count(f_j, e_i)} \tag{4}$$

以图 1 中短语对“北京 房价 持续 上涨，*beijing housing prices continued to rise*”为例，公式(3)的具体计算方式如下所示：

$$\begin{aligned} \Pr_{lex}(\bar{e} \mid \bar{f}, a) = {} & \omega(\text{beijing} \mid \text{北京}) * \omega(\text{housing} \mid \text{房价}) * \\ & \omega(\text{prices} \mid \text{房价}) * \omega(\text{continued} \mid \text{持续}) * \\ & \omega(\text{to} \mid \text{NULL}) * \omega(\text{rise} \mid \text{上涨}) \end{aligned}$$

公式(3)是一个二重循环问题，在外层循环中，从目标语言端第一个词汇遍历至最后一个词汇，将概率值进行连乘；在内层循环中，当前目标语言端词汇为 e_i，计算不同 f_j 翻译为 e_i 的概率和的均值。

➢ “目标语言->源语言”词汇化加权

“目标语言->源语言”方向词汇化加权与公式(3)相似，具体如公式所示。

$$\Pr_{lex}(\bar{f} \mid \bar{e}, a) = \prod_{j=1}^{J} \frac{1}{\left|\{ i \mid (i, j) \in a \}\right|} \sum_{\forall (i,j) \in a} \omega(f_j \mid e_i) \tag{5}$$

在公式(5)中，$\omega(f_j \mid e_i)$ 计算如公式(6)所示。其中公式(6)说明与公式(4)类似。

$$\omega(f_j \mid e_i) = \frac{count(e_i, f_j)}{\sum_{f_j} count(e_i, f_j)} \tag{6}$$

以图 1 中短语对“北京 房价 持续 上涨, *beijing housing prices continued to rise*”为例，公式(5)的具体计算方式如下所示：

$$\begin{aligned}\Pr_{lex}(\bar{f}\,|\,\bar{e},a) = {} & \omega(\text{北京}\,|\,\text{beijing})* \\ & \frac{1}{2}[\omega(\text{房价}|\text{housing}) + \omega(\text{房价}|\text{prices})]* \\ & \omega(\text{持续}|\text{continued})*\omega(\text{上涨}|\text{rise})\end{aligned}$$

此处具体计算方式的解释与上文相似，在此不再赘述。至此，短语翻译表中最常使用的 4 个特征介绍完毕。

注：附件“*phrase.translation.table*”为在实际的汉英短语翻译系统中使用的短语翻译表中的一部分。该短语翻译表共分为五个域，第一个域为“源语言”端短语；第二个域为“目标语言”端短语；第三个域为分数域，其中前四个分数为本文讲的双向短语翻译概率和双向词汇化权重，这四个分数在概率值基础上通过取 log 获得；第 4 个域为短语对出现的频次；最后一个域为词对齐信息。

- **参考资源**

1. *NiuTrans* 源码：*http://www.nlplab.com/NiuPlan/NiuTrans.ch.html* 下载 *NiuTrans* 源码包， *NiuTrans* 源码包目录*/NiuTrans/src/NiuTrans.PhraseExtractor/*下，*ruletable_scorer.cpp* 与 *ruletable_scorer.h* 两文件实现短语翻译表概率估计功能。

2. Philipp Koehn. 2010. *Statistical Machine Translation*. Cambridge, UK: Cambridge University Press.

3. Tong Xiao, Jingbo Zhu, Hao Zhang and Qiang Li. 2012. *NiuTrans: An Open Source Toolkit for Phrase-based and Syntax-based Machine Translation*. In *Proc. of ACL 2012*, page 19-24.