10.4 子词嵌入（fastText）

英语单词通常有其内部结构和形成方式。例如，我们可以从“dog”“dogs”和“dogcatcher”的字面上推测它们的关系。这些词都有同一个词根“dog”，但使用不同的后缀来改变词的含义。而且，这个关联可以推广至其他词汇。例如，“dog”和“dogs”的关系如同“cat”和“cats”的关系，“boy”和“boyfriend”的关系如同“girl”和“girlfriend”的关系。这一特点并非为英语所独有。在法语和西班牙语中，很多动词根据场景不同有40多种不同的形态，而在芬兰语中，一个名词可能有15种以上的形态。事实上，构词学（morphology）作为语言学的一个重要分支，研究的正是词的内部结构和形成方式。

在word2vec中，我们并没有直接利用构词学中的信息。无论是在跳字模型还是连续词袋模型中，我们都将形态不同的单词用不同的向量来表示。例如，“dog”和“dogs”分别用两个不同的向量表示，而模型中并未直接表达这两个向量之间的关系。鉴于此，fastText提出了子词嵌入（subword embedding）的方法，从而试图将构词信息引入word2vec中的跳字模型 [1]。

在fastText中，每个中心词被表示成子词的集合。下面我们用单词“where”作为例子来了解子词是如何产生的。首先，我们在单词的首尾分别添加特殊字符“<”和“>”以区分作为前后缀的子词。然后，将单词当成一个由字符构成的序列来提取n元语法。例如，当$n = 3$时，我们得到所有长度为3的子词：“<wh>”“whe”“her”“ere”“<re>”以及特殊子词“<where>”。

在fastText中，对于一个词w，我们将它所有长度在$3 \sim 6$的子词和特殊子词的并集记为$\mathcal{G}_w$。那么词典则是所有词的子词集合的并集。假设词典中子词g的向量为$\boldsymbol{z}_g$，那么跳字模型中词w的作为中心词的向量$\boldsymbol{v}_w$则表示成

$$\boldsymbol{v}_w = \sum_{g \in \mathcal{G}_w} \boldsymbol{z}_g.$$

fastText的其余部分同跳字模型一致，不在此重复。可以看到，与跳字模型相比，fastText中词典规模更大，造成模型参数更多，同时一个词的向量需要对所有子词向量求和，继而导致计算复杂度更高。但与此同时，较生僻的复杂单词，甚至是词典中没有的单词，可能会从同它结构类似的其他词那里获取更好的词向量表示。

小结

- fastText提出了子词嵌入方法。它在word2vec中的跳字模型的基础上，将中心词向量表示成单词的子词向量之和。
- 子词嵌入利用构词上的规律，通常可以提升生僻词表示的质量。