作业一

安装好 SRILM 后，对 thchs30 训练集的文本进行统计

- `-ukndiscountn`：使用原始 Kneser-Ney 的折扣算法。
- `-kndiscountn`：使用修正 Kneser-Ney 折扣算法。

```
$ ngram-count -text train.txt -order 3 -lm 3gram.lm -interpolate -kndiscount -debug 1
```

```
train.txt: line 10000: 10000 sentences, 208224 words, 0 OOVs
0 zeroprobs, logprob= 0 ppl= 1 ppl1= 1
using KneserNey for 1-grams
modifying 1-gram counts for Kneser-Ney smoothing
Kneser-Ney smoothing 1-grams
n1 = 14914
n2 = 975
D = 0.884369
using KneserNey for 2-grams
modifying 2-gram counts for Kneser-Ney smoothing
Kneser-Ney smoothing 2-grams
n1 = 33230
n2 = 274
D = 0.983776
using KneserNey for 3-grams
Kneser-Ney smoothing 3-grams
n1 = 20000
n2 = 0
one of required KneserNey count-of-counts is zero
error in discount estimator for order 3
```

查阅 SRILM 源代码 `SRILM/src/FDiscount.cc#L423-L436`：

```
if (n1 == 0 || n2 == 0 || n3 == 0 || n4 ==0) {
  cerr << "warning: one of required modified KneserNey count-of-counts is zero\n";
  return false;
}

/*
 * Compute discount constants (Ries 1997, Chen & Goodman 1998)
 */
double Y = (double)n1/(n1 + 2 * n2);

// use ModKneserNey:discount1 since we use it in ModKneserNey::discount()
```

```
ModKneserNey::discount1 = 1 - 2 * Y * n2 / n1;
discount2 = 2 - 3 * Y * n3 / n2;
discount3plus = 3 - 4 * Y * n4 / n3;
```

可知 `-ukndiscountn` 参数会统计 `n1` 和 `n2`，`-kndiscountn` 参数会统计 `n1`、`n2`、`n3` 和 `n4`。由于训练集中大量句子都重复出现，因此导致出现次数最少的 1gram ~ 3gram 都有至少 7 个，所以 `n1=n2=n3=n4=0`。使用 Witten-Bell 折扣算法：

```
$ ngram-count -text train.txt -order 3 -lm 3gram.lm -wbdiscount -debug 1
```

```
train.txt: line 10000: 10000 sentences, 208224 words, 0 OOVs
0 zeroprobs, logprob= 0 ppl= 1 ppl1= 1
using WittenBell for 1-grams
using WittenBell for 2-grams
using WittenBell for 3-grams
warning: distributing 0.0730558 left-over probability mass over all 17199
words
discarded 1 2-gram contexts containing pseudo-events
discarded 660 3-gram contexts containing pseudo-events
discarded 20000 3-gram probs discounted to zero
writing 17200 1-grams
writing 34310 2-grams
writing 14083 3-grams
```

统计模型在测试集上的困惑度：

```
$ ngram -lm 3gram.lm -order 3 -ppl test.txt
```

```
file test.txt: 2495 sentences, 51580 words, 21226 OOVs
0 zeroprobs, logprob= -122889.2 ppl= 5508.49 ppl1= 11182.35
```