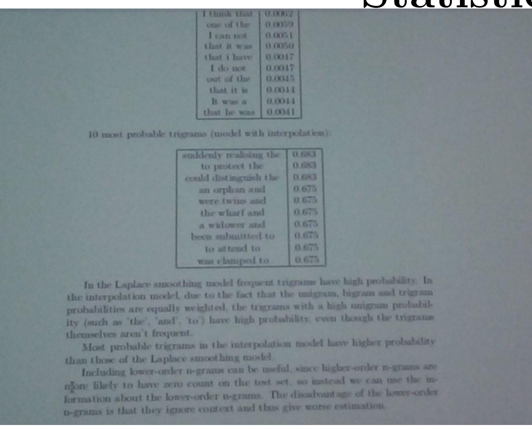
Statistical Natural Language Processing
Assignment 5

Sven Stauden 2549696
Janis Landwehr 2547715
Carsten Klaus 2554140



I think that	0.0062
one of the	0.0059
I can not	0.0051
that it was	0.0050
that i have	0.0047
I do not	0.0047
out of the	0.0045
that it is	0.0044
It was a	0.0044
that he was	0.0041

10 most probable trigrams (model with interpolation):

suddenly realising the	0.683
to protect the	0.683
could distinguish the	0.683
an orphan and	0.675
were twins and	0.675
the wharf and	0.675
a widower and	0.675
been submitted to	0.675
to attend to	0.675
was clamped to	0.675

In the Laplace smoothing model frequent trigrams have high probability. In the interpolation model, due to the fact that the unigram, bigram and trigram probabilities are equally weighted, the trigrams with a high unigram probability (such as 'the', 'and', 'to') have high probability, even though the trigrams themselves aren't frequent.

Most probable trigrams in the interpolation model have higher probability than those of the Laplace smoothing model.

Including lower-order n-grams can be useful, since higher-order n-grams are more likely to have zero count on the test set, so instead we can use the information about the lower-order n-grams. The disadvantage of the lower-order n-grams is that they ignore context and thus give worse estimation.

2/3

peplexity: 60.802,73
with interpolation: 306,39

...d results.txt

Exercise 2 1/2.5

see Pyhton file and results.txt

with $\lambda_1 = 0{,}7$
$\lambda_2 = 0{,}3$

peplexity 166,72 can be achieved

Exercise 3 - Witten-Bell Smoothing 1,5/1.5

As the word "grateful" is only followed by 12 words we have $N_{1+}(grateful) = 12$ and for "linear" we have $N_{1+}(linear) = 230$. Assuming that each of the 1000 occurrences of the both words in the text has a successor, we have $\sum_{w_n} count(grateful, w_n) = \sum_{w_n} count(linear, w_n) = 1000$. Thus we get:

$$\begin{aligned}
(1 - \lambda_{grateful}) &= \frac{N_{1+}(grateful)}{N_{1+}(grateful) + \sum_{w_n} count(grateful, w_n)} \\
&= \frac{12}{12 + 1000} = 0.0119 \\
\Leftrightarrow \lambda_{grateful} &= 1 - 0.0119 = 0.9881
\end{aligned}$$

and

$$\begin{aligned}
(1 - \lambda_{linear}) &= \frac{N_{1+}(linear)}{N_{1+}(linear) + \sum_{w_n} count(linear, w_n)} \\
&= \frac{230}{230 + 1000} = 0.187 \\
\Leftrightarrow \lambda_{linear} &= 1 - 0.187 = 0.813
\end{aligned}$$

For the impact of the bigram probabilities $P^I(w_n|"grateful")$ the high value of $\lambda_{grateful}$ states that the smoothed probability is very close to the native conditional probability $P(w_n|"grateful")$ and the probability of w_n has just a minor influence. As "grateful" only has a few possible successors it is uncertain that a successor appears in testing which was not respected during training. In general, this states, that the word w_n does only a major influence in combination with grateful which makes sense for combinations like *grateful for/that/because*.

The lower value of λ_{linear} appears because the probability that a bigram with "linear" as history does not occur in the training set is higher. Further, this means that the word w_n as successor of "linear" has a higher weighted meaning alone than as successor of "grateful".

Exercise 4 - Discounting 2/3

4.1

$P_{disc}(w_3|w_1w_2)$ is very close to the native probability $P(w_3|w_1w_2)$ when $N_{1+}(w_1w_2)$ is small. This happens when w_1w_2 often appear in combination with w_3 and therefore the probability that w_1w_2 has another successor than w_3 is low.

In contrast, $P_{disc}(w_3|w_1w_2)$ is heavily discounted when w_3 does not highly depend on w_1w_2. In this case many words could appear after w_1w_2 which leads to a high $N_{1+}(w_1w_2)$. ✓

4.2

Discarding V and N_{1+} will lead to the situation that the discounted estimates equal the native estimate. (obviously?!) ✓

↳ no smoothing

4.3

We distinguish between words w_{3in} which appear in a 3-gram as 3rd word after w_1w_2 and w_{3out} that never appear as last word of the 3-gram.

$$
\begin{aligned}
\sum_{w_3} \hat{P}(w_3|w_1,w_2) &= \sum_{w_{3in}} \hat{P}(w_{3in}|w_1,w_2) + \sum_{w_{3out}} \hat{P}(w_{3out}|w_1,w_2) \\
&= \sum_{w_{3in}} P_{disc}(w_{3in}|w_1,w_2) + \sum_{w_{3out}} \beta(w_1,w_2)\hat{P}(w_3|w_2) \checkmark \quad |\texttt{because } \underbrace{\sum_{w_{3out}} \hat{P}(w_3|w_2) = 1}_{\text{this does not hold!}} \\
&= \left(\sum_{w_{3in}} P_{disc}(w_{3in}|w_1,w_2)\right) + \beta(w_1,w_2) \quad \text{is not correct.} \\
1 &\stackrel{!}{=} \left(\sum_{w_{3in}} P_{disc}(w_{3in}|w_1,w_2)\right) + \beta(w_1,w_2) \quad |\texttt{applying definition} \\
&= \left(\sum_{w_{3in}} \frac{count(w_1,w_2,w_3)}{count(w_1,w_2) + N_{1+}}\right) + \beta(w_1,w_2) \quad N_{1+}(w_1,w_2)! \\
\Leftrightarrow \beta(w_1,w_2) &= 1 - \left(\sum_{w_{3in}} \frac{count(w_1,w_2,w_3)}{count(w_1,w_2) + N_{1+}}\right) \\
&= 1 - \sum_{w_{3in}} P_{disc}(w_{3in}|w_1,w_2)
\end{aligned}
$$

$$1 - \sum_{count(w_1,w_2,w_3)>0} \hat{P}(w_3|w_2)$$