Recitation notes on Kneser-Ney

Maria Ryskina

September 7, 2018

1 Notation

- $V$ – corpus vocabulary
- $c(x)$ – count of n-gram $x$ in the corpus
- $N_{1+}(\bullet w) \triangleq |\{u : c(u, w) > 0\}|$ – number of unique bigrams in the corpus ending in $w$
- $N_{1+}(w\bullet) \triangleq |\{u : c(w, u) > 0\}|$ – number of unique bigrams in the corpus starting with $w$
- $N_{1+}(\bullet w\bullet) \triangleq |\{(u, v) : c(u, w, v) > 0\}|$ – number of unique trigrams in the corpus with $w$ in the middle
- $1[\cdot]$ – indicator function

2 Conditional n-gram probabilities

$$P(w|\text{prev}_{k-1}) = \frac{\max(c'(\text{prev}_{k-1}, w) - d, 0)}{\sum_{v \in V} c'(\text{prev}_{k-1}, v)} + \alpha(\text{prev}_{k-1})P(w|\text{prev}_{k-2})$$ (1)

Highest order (trigram):

$$P(w_3|w_1w_2) = \frac{\max(c'(w_1w_2w_3) - d, 0)}{\sum_{v \in V} c'(w_1w_2v)} + \alpha(w_1w_2)P(w_3|w_2)$$ (2)

Lower order (bigram and unigram):

$$P(w_3|w_2) = \frac{\max(c'(w_2w_3) - d, 0)}{\sum_{v \in V} c'(w_2v)} + \alpha(w_2)P(w_3)$$ (3)

$$P(w_3) = \frac{c'(w_3)}{\sum_{v \in V} c'(v)}$$ (4)

Remembering the definition of $c'(x)$:
• if $x$ is a trigram, $c'(x) = c(x)$ (count of the trigram in the corpus)
• if $x$ is a bigram or a unigram: $c'(x) = N_{1+}(\bullet x)$ (number of unique words preceding $x$ in the corpus)

We substitute it in Equations 2-4:

$$P(w_3|w_1w_2) = \frac{\max(c(w_1w_2w_3) - d, 0)}{\sum_{v \in V} c(w_1w_2v)} + \alpha(w_1w_2)P(w_3|w_2) = \frac{\max(c(w_1w_2w_3) - d, 0)}{c(w_1w_2)} + \alpha(w_1w_2)P(w_3|w_2)$$

(5)

$$P(w_3|w_2) = \frac{\max(N_{1+}(\bullet w_2w_3) - d, 0)}{\sum_{v \in V} N_{1+}(\bullet w_2v)} + \alpha(w_2)P(w_3) = \frac{\max(N_{1+}(\bullet w_2w_3) - d, 0)}{N_{1+}(\bullet w_2\bullet)} + \alpha(w_2)P(w_3)$$

(6)

$$P(w_3) = \frac{N_{1+}(\bullet w_3)}{\sum_{v \in V} N_{1+}(\bullet v)} = \frac{N_{1+}(\bullet w_3)}{N_{1+}(\bullet \bullet)}$$

(7)

Here $N_{1+}(\bullet \bullet)$ is the number of all unique bigrams.

### 3 Computing $\alpha$

To compute $\alpha$, we sum over both sides of Equations 5-6 and use the fact that $\sum_{w \in V} P(w_3 = w|\ldots) = 1$. For the trigram case:

$$\sum_{w \in V} P(w_3 = w|w_1w_2) = \sum_{w \in V} \frac{\max(c(w_1w_2w) - d, 0)}{c(w_1w_2)} + \alpha(w_1w_2) \sum_{w \in V} P(w_3 = w|w_2)$$

$$1 = \sum_{w \in V} \frac{\max(c(w_1w_2w) - d, 0)}{c(w_1w_2)} + \alpha(w_1w_2)$$

(8)

Since $0 < d < 1$, we can rewrite this equation as:

$$1 = \sum_{w \in V} \frac{c(w_1w_2w) - d \cdot \sum_{w \in V} 1[c(w_1w_2w) > 0]}{c(w_1w_2)} + \alpha(w_1w_2) = \frac{\sum_{w \in V} c(w_1w_2w) - d \cdot N_{1+}(w_1w_2\bullet)}{c(w_1w_2)} + \alpha(w_1w_2)$$

(9)
Finally,

\[ \alpha(w_1w_2) = d \cdot \frac{N_{1+}(w_1w_2\bullet)}{c(w_1w_2)} \]  

(10)

Now, doing the same for the bigram case:

\[ 1 = \sum_{w \in V} \max \left( \frac{N_{1+}(\bullet w_2 w) - d, 0}{N_{1+}(\bullet w_2 \bullet)} \right) + \alpha(w_2) = \]
\[ = \sum_{w \in V} N_{1+}(\bullet w_2 w) - d \cdot \sum_{w \in V} \mathbb{1}[N_{1+}(\bullet w_2 w) > 0] \]
\[ + \alpha(w_2) \]  

(11)

Indicator \( \mathbb{1}[N_{1+}(\bullet w_2 w) > 0] \) is equal to 1 for every \( w \) for which \( w_2w \) occurs in at least one context. That is equivalent to saying bigram \( w_2w \) occurs at least once, so we can replace \( \mathbb{1}[N_{1+}(\bullet w_2 w) > 0] \) with \( \mathbb{1}[c(w_2w) > 0] \):

\[ 1 = 1 - d \cdot \sum_{w \in V} \mathbb{1}[c(w_2w) > 0] \]
\[ + \alpha(w_2) = 1 - d \cdot \frac{N_{1+}(w_2 \bullet)}{N_{1+}(\bullet w_2 \bullet)} + \alpha(w_2) \]  

(12)

Finally,

\[ \alpha(w_2) = d \cdot \frac{N_{1+}(w_2 \bullet)}{N_{1+}(\bullet w_2 \bullet)} \]  

(13)

4 Edge cases

- Our derivation until now assumed that \( c(w_1w_2) > 0 \), otherwise the denominators turn into 0. If the context \( w_1w_2 \) has never occurred before, fully back off to lower order until you get to a context with non-zero count.

- If \( w_3 \) is a word that has not been seen before, you can return a zero probability or back off to a uniform model and return \( \frac{1}{|V|} \). See Implementation tips for practical advice.

5 Implementation tips

- In your hashmap structures, you might want to store tables for values used for computing \( \alpha \) and \( P \) in addition to count tables:
  - for every occurring unigram \( w \) you would store \( N_{1+}(\bullet w), N_{1+}(w\bullet), N_{1+}(w\bullet \bullet) \)
  - for every occurring bigram \( vw \) you would store \( N_{1+}(vw\bullet) \) and \( N_{1+}(\bullet vw) \)

- To account for unknown words in translation, you can return a very small constant instead of a zero probability in case of a unigram not seen before.