Neural networks
Natural language processing - language modeling
**Topics:** language modeling

- A language model is a probabilistic model that assigns probabilities to any sequence of words

\[ p(w_1, \ldots, w_T) \]

- language modeling is the task of learning a language model that assigns high probabilities to well formed sentences
- plays a crucial role in speech recognition and machine translation systems

"a person smart"

"une personne intelligente"

?"a smart person"
Topics: language modeling

• An assumption frequently made is the $n^{th}$ order Markov assumption

$$p(w_1, ..., w_T) = \prod_{t=1}^{T} p(w_t | w_{t-(n-1)}, ..., w_{t-1})$$

- the $t^{th}$ word was generated based only on the $n-1$ previous words
- we will refer to $w_{t-(n-1)}, ..., w_{t-1}$ as the context
Topics: $n$-gram model

- An $n$-gram is a sequence of $n$ words
  - unigrams ($n=1$): “is”, “a”, “sequence”, etc.
  - bigrams ($n=2$): [“is”, “a”], [“a”, “sequence”], etc.
  - trigrams ($n=3$): [“is”, “a”, “sequence”], [“a”, “sequence”, “of”], etc.

- $n$-gram models estimate the conditional from $n$-grams counts

$$p(w_t | w_{t-(n-1)}, \ldots, w_{t-1}) = \frac{\text{count}(w_{t-(n-1)}, \ldots, w_{t-1}, w_t)}{\text{count}(w_{t-(n-1)}, \ldots, w_{t-1}, \cdot)}$$

- the counts are obtained from a training corpus (a data set of word text)
**Topics: n-gram model**

- **Issue: data sparsity**
  - we want $n$ to be large, for the model to be realistic
  - however, for large values of $n$, it is likely that a given $n$-gram will not have been observed in the training corpora
  - smoothing the counts can help
    - combine $\text{count}(w_1, w_2, w_3, w_4)$, $\text{count}(w_2, w_3, w_4)$, $\text{count}(w_3, w_4)$, and $\text{count}(w_4)$ to estimate $p(w_4 | w_1, w_2, w_3)$
  - this only partly solves the problem