# Language models

- A language model learns a model from a corpus of text such that, in test mode, given a sequence of words the model predicts the next word.
- Language models are closely related to the tagging problem and similar algorithms can be used.

Let $\mathcal{V}$ be a finite vocabulary, $\mathbf{L} = (w_1, \ldots, w_n), \; w_i \in \mathcal{V}$ be finite sequences of words (that is sentences) from $\mathcal{V}$. $\mathbf{L}$ can have sentences of arbitrary length and is therefore potentially infinite. For a technical reason we assume each sequence ends with $STOP \notin \mathcal{V}$. So, actual sequence length is always one more than the sentence.

**Definition 1 (Language model)**

*A language model is a function* $p(w_1, \ldots, w_n)$ *where* $w_i \in \mathcal{V}$ *such that:*

1. $p(w_1, \ldots, w_n) \geq 0.$
2. $\sum_{(w_1 \ldots, w_n \in \mathbf{L})} p(w_1, \ldots, w_n) = 1$

*p, therefore, is a probability distribution over elements in* $\mathbf{L}$*.*

# How to learn $p$?

- The main question is given a finite corpus $\mathcal{C} \subset \mathbf{L}$ how can we learn $p$.
- A naive count based method will not work since sentences almost never repeat.
- Two ways to learn models:
  1. Learn Markov models.
  2. Learn using a recurrent network.

# Markov model

Let $X_1, \ldots, X_n$ be a sequence of random variables with values chosen from $\mathcal{V}$. Temporarily, assume, that $n$ is a fixed positive integer.

We want to model the probability of $(w_1, \ldots, w_n)$, $n \geq 1$ and $w_i \in \mathcal{V}$. That is model the joint probability $P(X_1 = w_1, \ldots, X_n = w_n)$

There are $|\mathcal{V}|^n$ parameters in the model. So, intractable for moderate values of $n$ and $|\mathcal{V}|$.

A more compact model will result if we assume that the model is a Markov process of low order 1 or 2.

Assuming it is a first order Markov process:

$P(X_1 = w_1, \ldots, X_n = w_n) = P(X_1 = w_1) \prod_{i=2}^{n} P(X_i = w_i)|X_1 = w_1, \ldots, X_{i-1} = w_{i-1})$, using chain rule, no approximation

$P(X_1 = w_1, \ldots, X_n = w_n) = P(X_1 = w_1) \prod_{i=2}^{n} P(X_i = w_i)|X_{i-1} = w_{i-1})$, approximation using first order Markov process assumption.

So, the value of $X_i$ depends only on $w_{i-1}$ that is $X_i$ is conditionally independent of $X_1, \ldots, X_{i-2}$ given $X_{i-1}$. So, we can write assuming $w_{-1}, w_0 = START$

$P(X_1 = w_1, \ldots, X_n = w_n) = \prod_{i=1}^{n} P(X_i = w_i)|X_{i-1} = w_{i-1})$

If we make a second order Markov assumption we get:

$P(X_1 = w_1, \ldots, X_n = w_n) = \prod_{i=1}^{n} P(X_i = w_i)|X_{i-2} = w_{i-2}, X_{i-1} = w_{i-1})$

# If $n$ is not fixed

Assume each sequence is ended with $STOP$. So for a second order model:

$P(X_1 = w_1, \dots, X_n = w_n, STOP) = \prod_{i=1}^{n+1} P(X_i = w_i)|X_{i-2} = w_{i-2}, X_{i-1} = w_{i-1})$

$n \geq 1$, $w_i \in \mathcal{V}$, $i = 1..n$. So, the process that generates sentences is:

1. $i = 1$, $w_0 = w_{-1} = START$.
2. Generate $w_i$ from $P(X_i = w_i)|X_{i-2} = w_{i-2}, X_{i-1} = w_{i-1})$.
3. If $w_i = STOP$ return $(w_1, \dots, w_n, STOP)$ else $i = i + 1$ go to step 2.

So, with $STOP$ we can generate variable length sequences.