FNLP Tutorial 3

Question 1. A bigram language model

We are given the following corpus, modified from J&M (Chapter 3 in the 3rd edition):

<s> language is awesome </s>
<s> language is awesome </s>
<s> language and students </s>
<s> awesome students like language </s>

1. Using a bigram language model without smoothing, generate 4 sentences, starting from '<s>'. To generate sentences manually, randomly choose a real number between 0 and 1, and use the cumulative probabilities of words in the vocabulary to select a word from the language model. For example, if there are two words in the vocabulary with non-zero conditional probabilities $P(\text{one}|\text{<s>}) = 0.25$ and $P(\text{two}|\text{<s>}) = 0.75$, then choosing a random number between 0 and 0.25 would result in 'one', whereas a random number between 0.25 and 1 would result in 'two'.

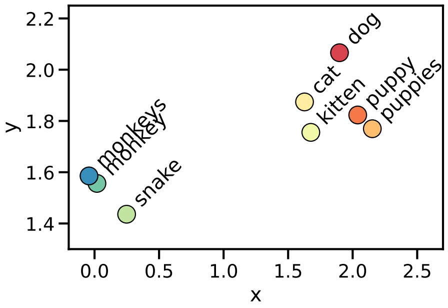
2. How does the lack of smoothing impact the novelty of the sentences generated?

3. How does the nature of the corpus impact the lengths of the generated sentences?

1 Question 2. Word vectors

Consider the following two-dimensional word vectors that encode animals. They are projections of 300-dimensional word vectors, that were trained using data from all over the web.

$$\overrightarrow{\text{dog}}=\begin{bmatrix}1.9\\2.07\end{bmatrix}, \overrightarrow{\text{puppy}}=\begin{bmatrix}2.04\\1.82\end{bmatrix}, \overrightarrow{\text{puppies}}=\begin{bmatrix}2.15\\1.77\end{bmatrix}, \overrightarrow{\text{cat}}=\begin{bmatrix}1.63\\1.87\end{bmatrix}$$
$$\overrightarrow{\text{kitten}}=\begin{bmatrix}1.68\\1.76\end{bmatrix}, \overrightarrow{\text{snake}}=\begin{bmatrix}0.25\\1.44\end{bmatrix}, \overrightarrow{\text{monkey}}=\begin{bmatrix}0.02\\1.56\end{bmatrix}, \overrightarrow{\text{monkeys}}=\begin{bmatrix}-0.04\\1.59\end{bmatrix}$$



1. Visually inspect the word vectors. Distances in the vector space capture word similarities (albeit strongly simplified when using only two dimensions). Why would 'snake' (a reptile) be closer to 'monkey' (a mammal) compared to the remaining animals (that are also mammals)? Please refer to the distributional hypothesis and the way word vectors are constructed in your answer.

2. Word vectors can capture relationships between words, as discussed in J&M Section 6.10, 3rd edition. An example of such a relationship is an analogy, such as Edinburgh is to Scotland as Canberra is to ...? When we have word vectors, we can use the parallelogram method to solve analogies: by subtracting $\overrightarrow{\text{edinburgh}}$ from $\overrightarrow{\text{scotland}}$ and adding $\overrightarrow{\text{canberra}}$. You would pick the word for which the word vector has the smallest distance to the resulting vector. J&M represent the procedure as follows for the analogy a:b::a*:b* (a is to b as a* is to b*):

$$\overrightarrow{\text{b}}* = \text{argmin distance}_{\overrightarrow{\text{x}}}(\overrightarrow{\text{x}}, \overrightarrow{\text{b}} - \overrightarrow{\text{a}} + \overrightarrow{\text{a}}*) \tag{1}$$

   Using Euclidean distances, compute $\overrightarrow{\text{b}}*$ for:

   - $\overrightarrow{\text{a}} = \overrightarrow{\text{dog}}$, $\overrightarrow{\text{b}} = \overrightarrow{\text{puppy}}$ and $\overrightarrow{\text{a}}* = \overrightarrow{\text{cat}}$
   - $\overrightarrow{\text{a}} = \overrightarrow{\text{puppy}}$, $\overrightarrow{\text{b}} = \overrightarrow{\text{puppies}}$ and $\overrightarrow{\text{a}}* = \overrightarrow{\text{monkey}}$

3. Retrieve the word that is the most similar to 'monkey' using cosine similarity, euclidean distance and the dot product. Where do the different metrics disagree, and how does this relate to their definition?

4. Explain how Euclidean distance, the dot product and the cosine similarity are related.

Question 3. Model design

The next problem looks at how to apply neural models to a classical problem in NLP: part-of-speech (POS) tagging. POS tags are labels assigned to each word in a sentence to indicate their grammatical role in the sentence. For example:

$i =$	1	2	3	4	5	6	7	8	9	10
$x_i =$	Each	day	starts	with	one	or	two	lectures	by	researchers
$y_i =$	DT	NN	VBZ	IN	CD	CC	JJR	NNS	IN	NNS

Common categories are Nouns (N), Pronouns (PRP), Verbs (VB), Adjectives (JJ), Adverbs (RB), Prepositions (IN), Determiners (DT), Conjunctions (CC) and Interjections(UH). There are also more fine-grained categories: word *day* in the sentence is a singular noun (NN). However, the same word can have different PoS tags, depending on the context.

Given an input sentence $x = x_1 \dots x_{|x|}$, we want to predict the corresponding tag sequence $y = y_1 \dots y_{|x|}$. Let x_i denote the ith word of x, y_i denote the ith word of y, and $|x|$ denote the length of x. Note that $|y| = |x|$. We have access to many training examples like this, and our goal is to model the conditional probability of the tag sequence given the sentence, that is: $P(y \mid x)$. There are many possible choices here. To simplify the problem, let's *assume* that each element of y is conditionally independent of each other. That is, we want to model:

$$P(y \mid x) = \prod_{i=1}^{|y|} P(y_i \mid x)$$

1. Design a feedforward neural network to model $P(y_i \mid x)$: clearly define the probability being computed and how it is estimated, and specify the input and output vocabulary, as well as the objective function used for training. Identify any independence assumptions you make. Draw a diagram that illustrates how the model computes probabilities for the tag of the word "with": What is the input, and how is the output distribution computed from the input? Write out the basic equations of the model, and explain your choices.

2. Design an RNN to model $P(y_i \mid x)$: clearly define the probability being computed and how it is estimated, and specify the input and output vocabulary, as well as the objective function used for training. Identify any independence assumptions you make. Draw a diagram that illustrates how the model computes probabilities for the tag of the word "with": What is the input, and how is the output distribution computed from the input? Write out the basic equations of the model, and explain your choices.

3. Can you model $P(y_i \mid x)$ without independence assumptions, using multiple RNNs?

For each question, the goal is to design a *simple* model for the distribution. You solution should only use architectures that we discussed in the first four weeks of the course. If you are aware of other architectures, you should not use them here.