FNLP Tutorial 5

1 Lexical semantics

1. Consider the words "pie" (dish baked in pastry-lined pan often with a pastry top), "relationship" (human relationship), "universe" (everything that exists anywhere) and "garden" (ground where plants are cultivated). At first glance, these words seem to have very little in common. Use Wordnet (http://wordnetweb.princeton.edu/perl/webwn) to follow the chain of hypernyms for these words, to figure out which one is the odd one out. Report the chain of hypernyms and use them to motivate your answer. You follow the chain of hypernyms by clicking 'S' next to a word sense, and looking at its 'direct hypernym'.

   Solution The four words lead to the following hypernym chains. The words "garden", "pie" and "universe" have in common that they are physical entities, while "relationship" is an abstract entity. Therefore, "relationship" could be viewed as the odd one out.

   (a) garden → plot, plot of land, plot of ground, patch → tract, piece of land, piece of ground, parcel of land, parcel → geographical area, geographic area, geographical region, geographic region → region → object, physical object → **physical entity** → entity

   (b) pie → pastry → baked goods → food, solid food → solid → matter → **physical entity** → entity

   (c) relationship, human relationship → relation → abstraction, **abstract entity** → entity

   (d) universe, existence, creation, world, cosmos, macrocosm → natural object → whole, unit → object, physical object → **physical entity** → entity

2. Describe the relationship between the following word pairs. Where there are multiple plausible word senses for the relation, we disambiguate this. How far do you get with the main relationships in WordNet (hyponym/hypernym, meronym, antonym)? Characterise the relationships that are not covered by WordNet and give them a name; try to invent relationships that could potentially be used for many other word pairs.

   - bike - vehicle
   - bike - sports
   - book - novel
   - book - library (a room where books are kept)
   - fish (the flesh of fish used as food) - chips (a thin crisp slice of potato fried in deep fat)
   - knife - drawer
   - knife - cut

   Solution Sample solution (there are other plausible ones!):

   - bike - vehicle. Hyponymy: a bike is a vehicle
   - bike - sports. Not captured by WordNet. Bikes are used for sports (among other things); we introduce the "is-used-for" relation.

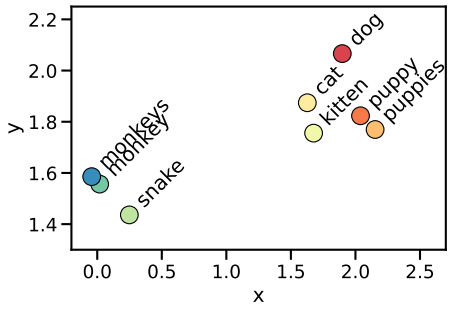
- book - novel. Hypernymy: a novel is a certain type of book.
- book - library. Not captured by WordNet. Books can often be found in libraries; we introduce the "can-be-found-in" relation.
- fish - chips. Not captured by WordNet. Fish and cheaps are often eaten together ("fish and chips" is arguably one dish); we introduce the "are-often-eaten-together" relation.
- knife - drawer. Not captured by WordNet. Knives are often kept in drawers; we can re-use the "can-be-found-in" relation.
- knife - cut. Not captured by WordNet. Knives can be used for making cuts, we can re-use the "is-used-for" relation.

As you can see, there are many relationships that are not captured by WordNet that encode a lot of world knowledge, which could be useful for drawing inferences in NLP applications. However, it would be an incredible effort to write down all "relevant" relations for all conceivable applications.

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.

   Solution According to the distributional hypothesis words that occur in similar contexts tend to have similar meanings. This is reflected in the point in the vector space we associate with a word: other points in its neighbourhood will occur in similar contexts. The contexts come from the training corpus. For arbitrary content from the web concerning animals, we expect pets like dogs and cats to appear in similar contexts. The same might hold for wildlife like a monkey and a snake, but it seems less likely that a cat and a snake appear in similar contexts.

   The question suggests the vectors could have reflected the mammal vs. reptile difference, but this difference is not as prominent from text as the very different habitats of pets and wildlife.

Were the vectors trained on a specialised corpus about biological evolution or the anatomy of animals, they probably *would* reflect that difference more.

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*):

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

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

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

Solution

- $\vec{b} - \vec{a} + \vec{a}* = \begin{bmatrix}2.04\\1.82\end{bmatrix} - \begin{bmatrix}1.9\\2.07\end{bmatrix} + \begin{bmatrix}1.63\\1.87\end{bmatrix} = \begin{bmatrix}1.77\\1.62\end{bmatrix}$

word	**distance**
dog	$\sqrt{(1.9-1.77)^2+(2.07-1.62)^2} \approx 0.47$
puppy	$\sqrt{(2.04-1.77)^2+(1.82-1.62)^2} \approx 0.34$
puppies	$\sqrt{(2.15-1.77)^2+(1.77-1.62)^2} \approx 0.41$
cat	$\sqrt{(1.63-1.77)^2+(1.87-1.62)^2} \approx 0.29$
kitten	$\sqrt{(1.68-1.77)^2+(1.76-1.62)^2} \approx 0.17$
snake	$\sqrt{(0.25-1.77)^2+(1.44-1.62)^2} \approx 1.53$
monkey	$\sqrt{(0.02-1.77)^2+(1.56-1.62)^2} \approx 1.75$
monkeys	$\sqrt{(-0.04-1.77)^2+(1.59-1.62)^2} \approx 1.81$

The word vector for 'kitten' has the minimum distance, hence the vectors suggest dog:puppy::cat:kitten, which makes perfect sense!

- $\vec{b} - \vec{a} + \vec{a}* = \begin{bmatrix}2.15\\1.77\end{bmatrix} - \begin{bmatrix}2.04\\1.82\end{bmatrix} + \begin{bmatrix}0.02\\1.56\end{bmatrix} = \begin{bmatrix}0.13\\1.51\end{bmatrix}$

word	**distance**
dog	$\sqrt{(1.9-0.13)^2+(2.07-1.51)^2} \approx 1.85$
puppy	$\sqrt{(2.04-0.13)^2+(1.82-1.51)^2} \approx 1.93$
puppies	$\sqrt{(2.15-0.13)^2+(1.77-1.51)^2} \approx 2.04$
cat	$\sqrt{(1.63-0.13)^2+(1.87-1.51)^2} \approx 1.54$
kitten	$\sqrt{(1.68-0.13)^2+(1.76-1.51)^2} \approx 1.57$
snake	$\sqrt{(0.25-0.13)^2+(1.44-1.51)^2} \approx 0.14$
monkey	$\sqrt{(0.02-0.13)^2+(1.56-1.51)^2} \approx 0.12$
monkeys	$\sqrt{(-0.04-0.13)^2+(1.59-1.51)^2} \approx 0.19$

Dependent on whether we exclude a, b and a* as potential answers or not, 'monkey' or 'snake' is closest, which, unfortunately, is not 'monkeys'. The vector $\overrightarrow{\text{monkeys}}$ is close to $\overrightarrow{\text{monkey}}$ in the vector space, but the relationship is not analogical to the one between $\overrightarrow{\text{puppies}}$ and $\overrightarrow{\text{puppy}}$.

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?

Solution We collected the similarities below. For both the Euclidean distance and the cosine similarity, the word most similar to 'monkey' is 'monkeys', but the dot product suggests 'dog' is the most similar to 'monkey'. This is due to the impact of the vectors' norm on the computation of the dot product (Equation 2), since there is no length normalisation.

$$dot(\vec{u}, \vec{v}) = \sum_{i=1}^{d} \vec{u}_i \vec{v}_i \tag{2}$$

word	**Euclidean dist.**	**cosine dist.**	**dot product**
monkey	0.00	0.00	2.43
dog	1.95	0.25	3.27
puppy	2.04	0.32	2.88
puppies	2.14	0.35	2.80
cat	1.64	0.24	2.95
kitten	1.67	0.27	2.78
snake	0.26	0.01	2.25
monkeys	0.07	0.00	2.48

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

Solution The cosine similarity represents the cosine of the angle between two input vectors, and is the dot product normalised by the norms of the two vectors (Equation 3). The relation between the cosine similarity and the Euclidean distance may not be clear directly. After all, two vectors in the same direction can be very distant as per the Euclidean distance, but still very similar according to the cosine similarity. However, for normalised vectors, they are related, as is illustrated in Equations 4 and 5. If $\vec{u}$ and $\vec{v}$ have length 1, $ED(\vec{u}, \vec{v})^2 = 2 - 2dot(\vec{u}, \vec{v})$.

$$cos\theta_{u,v} = \frac{dot(\vec{u}, \vec{v})}{\|\vec{u}\|\|\vec{v}\|} \tag{3}$$

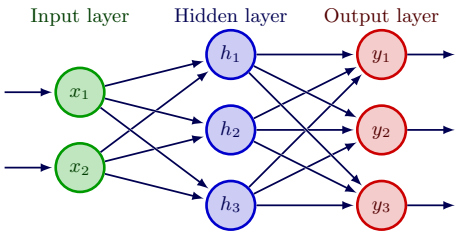
$$ED(\vec{u}, \vec{v}) = \sqrt{\sum_{i=1}^{d} (\vec{v}_i - \vec{u}_i)^2} = \sqrt{\sum_{i=1}^{d} (\vec{v}_i \vec{v}_i - 2\vec{v}_i \vec{u}_i + \vec{u}_i \vec{u}_i)} = \sqrt{\sum_{i=1}^{d} \vec{v}_i \vec{v}_i + \sum_{i=1}^{d} \vec{u}_i \vec{u}_i - 2\sum_{i=1}^{d} \vec{v}_i \vec{u}_i} \tag{4}$$

$$ED(\vec{u}, \vec{v})^2 = \sum_{i=1}^{d} \vec{v}_i \vec{v}_i + \sum_{i=1}^{d} \vec{u}_i \vec{u}_i - 2\sum_{i=1}^{d} \vec{v}_i \vec{u}_i \tag{5}$$

3 Feed-forward neural networks

Consider a two-layer neural network with the topology visualised below, with the corresponding weights and bias values in the table. The hidden layer is followed by a non-linear function: the ReLU. The output layer is followed by a non-linear function too: the softmax. Read up on those functions and how to work with feedforward neural networks in sections 7.1 to 7.3 from J&M (only available in the 3rd edition!).

The network can be used for simple classification for three output classes. An input (consisting of features x_1 and x_2) belongs to one of the three classes, and you will classify an example input.



$w_{x_1,h_1} = 0.23$	$b_{h_1} = 0.13$	$w_{h_1,y_1} = 0.02$	$b_{y_1} = 0.31$
$w_{x_1,h_2} = -2.90$	$b_{h_2} = 0.90$	$w_{h_1,y_2} = -0.13$	$b_{y_2} = -0.18$
$w_{x_1,h_3} = -0.45$	$b_{h_3} = 0.24$	$w_{h_1,y_3} = 0.47$	$b_{y_3} = -0.56$
$w_{x_2,h_1} = 1.10$		$w_{h_2,y_1} = -0.88$	
$w_{x_2,h_2} = 0.05$		$w_{h_2,y_2} = 1.74$	
$w_{x_2,h_3} = -0.01$		$w_{h_2,y_3} = -0.09$	
		$w_{h_3,y_1} = -0.55$	
		$w_{h_3,y_2} = 0.72$	
		$w_{h_3,y_3} = -2.22$	

1. Compute the class an input with $x_1 = 1.50$, $x_2 = 3.11$ would belong to. Show the intermediate computations, not just the final class.

   Solution Firstly, let's compute the activation of the nodes in the hidden layers. This requires combining the inputs with the weights, adding the bias, and applying the ReLU function:
   $a_{h_1} = \text{ReLU}(1.50 \cdot 0.23 + 3.11 \cdot 1.10 + 0.13) = \text{ReLU}(3.896) = 3.896$
   $a_{h_2} = \text{ReLU}(1.50 \cdot -2.90 + 3.11 \cdot 0.05 + 0.90) = \text{ReLU}(-3.2945) = 0$
   $a_{h_3} = \text{ReLU}(1.50 \cdot -0.45 + 3.11 \cdot -0.01 + 0.24) = \text{ReLU}(-0.4661) = 0$

   Secondly, let's compute the activation of the output nodes **before** computing the softmax. We need all three outputs to be able to apply the softmax:
   $a_{y_1} = 3.896 \cdot 0.02 + 0 + 0 + 0.31 = 0.38792$
   $a_{y_2} = 3.896 \cdot -0.13 + 0 + 0 + -0.18 = -0.68648$
   $a_{y_3} = 3.896 \cdot 0.47 + 0 + 0 + -0.56 = 1.27112$
   Now, as a final step, we can apply the softmax to turn the outputs into probabilities:
   $p_1 = \frac{\exp(0.38792)}{\exp(0.38792)+\exp(-0.68648)+\exp(1.27112)} \approx 0.266$
   $p_2 = \frac{\exp(-0.68648)}{\exp(0.38792)+\exp(-0.68648)+\exp(1.27112)} \approx 0.091$
   $p_3 = \frac{\exp(1.27112)}{\exp(0.38792)+\exp(-0.68648)+\exp(1.27112)} \approx 0.643$

   This input is categorised with class 3!

2. Now imagine that you want to perform classification, but one input can belong to multiple classes. For example, when classifying a sentence with an emotion, that sentence can capture both anger and despair. To enable multi-class classification in this network, what adaptation would you make to its structure or the non-linear functions it uses?

   Solution A rather complicated solution would be creating output classes that represent multiple classes. For example, for 3 output classes, we would create separate output nodes for classes (1,), (2,), (3,), (1, 2), (1, 3), (2, 3), (1, 2, 3). This is the *wrong* solution due to the complexity it adds to the network.
   The most straightforward solution is replacing the non-linear function of the output layer with a sigmoid function, that computes a value between 0 and 1 for each class, and by thresholding that value, one would know whether an input belongs to that class or not.