Natural Language Processing
Portfolio II

Leon F.A. Wetzel
Information Science
Faculty of Arts - University of Groningen
`l.f.a.wetzel@student.rug.nl`

March 21, 2021

Abstract

In this document, you can find the results and explanations for the assignments of the second part of the portfolio for the course Natural Language Processing, taught at the University of Groningen. The corresponding Python code can be found at https://github.com/leonwetzel/natural-language-processing[1]. Note that version control of Jupyter notebooks is done via `jupytext`, so do not forget to convert the relevant Python scripts to notebooks yourself!

1 Week 5 - Neural Language Models

1.1 Neurons

Consider the basic neuron y with a sigmoid activation σ (g in the image).

$$y = \sigma(x * w + b) \qquad \sigma = s(z) = \frac{1}{1+e^{-z}} \qquad y = \sigma(x * w + b) = \frac{1}{1+e^{-(x*w+b)}}$$

Given the weight and bias values $x = [0.3, 0.9, 0.1]$, $w = [-0.2, 0.8, -0.6]$, $b = 0.4$ compute the value of y and provide intermediate calculations.

$$\begin{aligned} y &= \frac{1}{1+e^{-(w*x+b)}} = \frac{1}{1+e^{-(0.3*-0.2+0.9*0.8+0.1*-0.6+0.4)}} \\ &= \frac{1}{1+e^{-0.06+0.72+-0.06+0.4}} = \frac{1}{1+e^{-1}} \\ &= \frac{1}{1+0.36787944117} = 0.73105857863 \end{aligned} \tag{1}$$

1.2 Relu

Assume the same basic neuron as in 5.1, but now with ReLU activation, where

$$\sigma = \text{relu}(z) = max(z, 0)$$

Give the updated formula for computing y, as well as the new value of y (with intermediate calculations).

$$y = \sigma(x * w + b) \qquad \sigma = \text{relu}(z) = max(z, \theta) \qquad y = \sigma(x * w + b) = max(x * w + b, 0)$$

[1] All code will be published after the course has been completed

$$
\begin{aligned}
y = max(x * w + b, 0) &= max(0.3 * -0.2 + 0.9 * 0.8 + 0.1 * -0.6 + 0.4, 0) \\
&= max(-0.06 + 0.72 + -0.06 + 0.4, 0) \\
&= max(1, 0) = 1
\end{aligned}
\tag{2}
$$

1.3 Model size

The feed-forward neural network below with a vocabulary of 50.000 words represented by one-hot vectors, a context window of 3 words, a projection layer with $d = 100$, a hidden layer $d_h = 500$, like the one below, has a total of $P = E + W + U$ trainable parameters. Which of the following has a bigger impact on the number of parameters:

1. increasing context-size from 3 to 4, or
2. increasing vocabulary from 50.000 to 51.000 words

Give the calculations of P to motivate your answer.

$$
\left.
\begin{aligned}
E &= d * |V| = 100 * 50.000 = 5.000.000 \\
W &= d_h * 3d = 500 * 3 * 100 = 150.000 \\
U &= |V| * d_h = 50.000 * 500 = 25.000.000 \\
P &= E + W + U = 5.000.000 + 150.000 + 25.000.000 = 30.150.000
\end{aligned}
\right\} \quad \text{CS = 3} \tag{3}
$$

$$
\left.
\begin{aligned}
E &= d * |V| = 100 * 50.000 = 5.000.000 \\
W &= d_h * 4d = 500 * 4 * 100 = 200.000 \\
U &= |V| * d_h = 50.000 * 500 = 25.000.000 \\
P &= E + W + U = 5.000.000 + 200.000 + 25.000.000 = 30.200.000
\end{aligned}
\right\} \quad \text{CS = 4} \tag{4}
$$

$$
\left.
\begin{aligned}
E &= d * |V| = 100 * 51.000 = 5.100.000 \\
W &= d_h * 3d = 500 * 3 * 100 = 150.000 \\
U &= |V| * d_h = 51.000 * 500 = 25.500.000 \\
P &= E + W + U = 5.100.000 + 150.000 + 25.500.000 = 30.750.000
\end{aligned}
\right\} \quad \text{V = 51.000} \tag{5}
$$

Increasing the vocabulary size leads to a higher amount of trainable parameters, contrary to increasing the context-size. The size of vocabulary affects both E and U (opposed to only W when changing the context size), which leads to higher values in the summation.

1.4 Architectures

Provide at least one example of an NLP application for each of the following task formulations. Try to come up with examples that are different from those mentioned in class.

Task formulation	**Example(s)**
One-to-many	Poem generation
Many-to-one	Language detection
Many-to-many (seq2seq)	Document summarization
Many-to-many (sequence labelling)	Semantic tagging

Table 1: Examples per task formulation for RNN's

1.5 Probing

Think of a grammatical phenomenon in a language of your choice, and come up with at least 10 example sentences to probe whether the model makes the correct predictions. Think of cases where the context makes it clear that the mask has to be plural or singular, that a verb has to have a particular form (like plural or singular, or participle or infinitive), that a specific (personal, possessive, reflexive) pronoun has to be used, that an adjective or noun has to have a specific inflection (like in German and more generally in languages with a rich case and/or gender marking system). There is a host of literature on this, see for instance Marvin and Linzen (for English) and Sahin et al (for multilingual probes).

1. De minister-president heeft [MASK] handtekening gezet onder de bepaling.
   (a) de;
   (b) een;
   (c) geen;
   (d) zijn;
   (e) deze;
2. De ontwikkelaars onthouden [MASK] van commentaar op hun code.
   (a) software;
   (b) ook;
   (c) gebruik;
   (d) code;
   (e) niet;
3. Het ijsje begon met [MASK] toen de zon doorbrak.
   (a) ij;
   (b) ,;
   (c) zon;
   (d) :;
   (e) water;
4. De universiteit investeerde [MASK] in onderzoeken naar kunstmatige intelligentie.
   (a) ook;
   (b) verder;
   (c) ##n;
   (d) vooral;
   (e) zich;
5. Het zwaard van Damocles hing boven [MASK] hoofd.
   (a) zijn;
   (b) het;
   (c) de;
   (d) haar;
   (e) hun;
6. De kaas uit Duitsland was niet [MASK] dan de kaas uit Nederland.

    (a) groter;
    (b) meer;
    (c) kleiner;
    (d) anders;
    (e) beter;

7. De diplomaten dronken uit [MASK] bekers terwijl het ongeval plaatsvond.
    (a) de;
    (b) in;
    (c) het;
    (d) uit;
    (e) en;

8. Het weer was onstuimig, het [MASK] namelijk vrij hard.
    (a) was;
    (b) weer;
    (c) is;
    (d) kwam;
    (e) had;

9. De scholen sloten hun deuren [MASK].
    (a) in;
    (b) aan;
    (c) op;
    (d) ’;
    (e) .;

10. Jan had gisteren [MASK] voet gestoten tegen de tafelpoot.
    (a) en;
    (b) met;
    (c) van;
    (d) aan;
    (e) de;

> Give at least ten example sentences with a [MASK] and a list of targets that illustrate a specific grammatical phenomenon in a language of your choice. Describe what the grammatical phenomenon is you are investigating. Use the probe function for testing. Try to include both easy sentences (where the model should do well) as well as hard sentences (where there are words in the context that might lead to confusion, or where the clue words are far away from the mask). For languages other than Dutch or English, make sure to include enough explanation so that examples and tests are clear to a non-native speaker.
> Describe how well the model did on your probe sentences. Where there any cases where the model made the wrong decision?

Please see the notebook for the sentences that were tested. The model did quite well on the provided probe sentences. There is a noticeable bias present in the model when it comes to gender and pronoun usage; him/his often comes on tops when such cases are masked. Although the model performs well in most cases, it did make a slight error in the sentence `The dog [MASK] hunting for food in the evenings.`, where *are* apparently has a higher score than *was*.

2 Week 6 - Dependency Parsing

2.1 Evaluation

We are given the following sentence: `I hate to put a little pressure on you`

Our total amount of nodes (N) is 8, so our formulas would look like...

$$UAS = \frac{\text{\# of nodes with correct parent}}{8} \tag{6}$$

$$LAS = \frac{\text{\# of nodes with correct parent and edge label}}{8} \tag{7}$$

For the example on the upper side, the scores are as follows:

$$UAS = \frac{8}{8} = 1 \tag{8}$$

$$LAS = \frac{8}{8} = 1 \tag{9}$$

For the example on the lower side, the scores are as follows:

$$UAS = \frac{6}{8} = 0.75 \tag{10}$$

$$LAS = \frac{3}{8} = 0.375 \tag{11}$$

2.2 Transition-based Parsing

Describe the states and actions that a transition-based parser has to go through to produce the gold standard (top) analysis of the sentence in question 6.1. A state consists of a stack, input buffer, and set of dependency relations. An action is either SHIFT, RIGHTARC, or LEFTARC, where you can assume that the RIGHTARC and LEFTARC actions also produce the correctly labeled dependency relations.

We are given the following sentence: `I hate to put a little pressure on you`

We assume that the upper example from exercise 2.1 is the gold standard.

Step	Stack	Word List	Action	Relation Added
0	[root]	[I, hate, to, put, a, little, pressure, on, you]	SHIFT	
1	[root, I]	[hate, to, put, a, little, pressure, on, you]	SHIFT	
2	[root, I, hate]	[to, put, a, little, pressure, on, you]	LEFTARC	(I ← hate)
3	[root, hate]	[to, put, a, little, pressure, on, you]	SHIFT	
4	[root, hate, to]	[put, a, little, pressure, on, you]	SHIFT	
5	[root, hate, to, put]	[a, little, pressure, on, you]	LEFTARC	(to ← put)
6	[root, hate, put]	[a, little, pressure, on, you]	SHIFT	
7	[root, hate, put, a]	[little, pressure, on, you]	SHIFT	
8	[root, hate, put, a, little]	[pressure, on, you]	SHIFT	
9	[root, hate, put, a, little, pressure]	[on, you]	LEFTARC	(little ← pressure)
10	[root, hate, put, a, pressure]	[on, you]	LEFTARC	(a ← pressure)
11	[root, hate, put, pressure]	[on, you]	RIGHTARC	(put → pressure)
12	[root, hate, put]	[on, you]	SHIFT	
13	[root, hate, put, on]	[you]	SHIFT	
13	[root, hate, put, on, you]	[]	LEFTARC	(on ← you)
14	[root, hate, put, you]	[]	RIGHTARC	(put → you)
15	[root, hate, put]	[]	RIGHTARC	(hate → put)
15	[root, hate]	[]	RIGHTARC	(root → hate)
16	[root]	[]	Done	

2.3 Crossing arcs

It is not possible to produce the analysis for the sentence below using a transition-based parser. Give the state and input buffer for the point where the problem arises and explain why there is no sequence of actions that leads to a correct parse.

We are given the following sentence: `Who did Kim talk to`

Step	Stack	Word List	Action	Relation Added
0	[root]	[Who, did, Kim, talk, to]	SHIFT	
1	[root, Who]	[did, Kim, talk, to]	SHIFT	
2	[root, Who, did]	[Kim, talk, to]	SHIFT	
3	[root, Who, did, Kim]	[talk, to]	SHIFT	
4	[root, Who, did, Kim, talk]	[to]	LEFTARC	(Kim ← talk)
5	[root, Who, did, talk]	[to]	LEFTARC	(did ← talk)
6	[root, Who, talk]	[to]	LEFTARC	(Who ← talk)
7	**[root, talk]**	**[to]**	**SHIFT (?)**	
?	[root]	[]	Done	

As we can see in the table above, the problem arises near step 7. As a result of the previous LEFTARC action, we removed the word **Who** which would be needed for a later RIGHTARC operation. The remaining word **talk** cannot be coupled with **to**, as there is no sensible relation between these words. We cannot couple **root** and **talk** at this moment, as **to** is still part of either the stack or word list.

2.4 Tiny-dependency parser

Improve the oracle of the dependency parser in the notebook and train and evaluate it on the data. See notebook for details.

3 Week 7 - Word Sense Disambiguation

See the attached notebook for the results of the various experiments.