

# CS 5316: ASSIGNMENT 4

18100276 - Muhammad Zain Qasmi

- It is mandatory to provide a single word document with all the solutions and clear screenshots of code along with descriptive comments. It is your duty to explain what you have done as clearly as possible in your solution document.
- Ambiguous answers and unrelated submission documents will be heavily penalized.
- You also need to provide your code which shall be used for plagiarism checks. ( make sure its clean and readable.
- You will use python 2/3 for this assignment. (NLTK has most packages that you would need)
- You will be marked on clarity and accurate solutions. If required, a viva might be arranged.
- Submission deadline: 16<sup>th</sup> April

## Part 1: POS tagging

### Question 1:

- a) Write down the name of 3 POS tag annotated datasets for the English language.

*“The Cambridge Analytica scandal is more than a “breach,” as Facebook executives have defined it. It exemplifies the possibility of using online data to algorithmically predict and influence human behavior in a manner that circumvents users’ awareness of such influence. Using an intermediary app, Cambridge Analytica was able to harvest large data volumes—over 50 million raw profiles—and use big data analytics to create psychographic profiles in order to subsequently target users with customized digital ads and other manipulative information. According to some observers, this massive data analytics tactic might have been used to purposively swing election campaigns around the world. The reports are still incomplete and more is likely to come to light in the next days.”*

- 1) Brown Corpus - American English
- 2) CPSAE - Corpus of Spoken Professional American English
- 3) COLT - Bergen Corpus of London Teenage Language

*Following has list of all corpus in varios English Languages*  
<http://www.corpora4learning.net/resources/corpora.html#AE>

- b) Use a pre-trained pos tagger to identify tags for the above text. Which tagger did you use? Comment on your result highlighting inaccuracies and strategies for improving performance. (you will be marked accordingly)

```
zainqasmi@zainqasmi:~/Desktop/NLP4$ python my1B.py
[('The', 'DT'), ('Cambridge', 'NNP'), ('Analytica', 'NNP'), ('scandal', 'NN'), ('is', 'VBZ'), ('more', 'JJR'), ('than', 'IN'), ('a', 'DT'), ('breach', 'NN'), ('"', '""'), ('"', '""'), ('as', 'IN'), ('Facebook', 'NNP'), ('executives', 'NNS'), ('have', 'VBP'), ('defined', 'VBN'), ('it', 'PRP'), ('.', '.'), ('It', 'PRP'), ('exemplifies', 'VBZ'), ('the', 'DT'), ('possibility', 'NN'), ('of', 'IN'), ('using', 'VBG'), ('online', 'JJ'), ('data', 'NNS'), ('to', 'TO'), ('algorithmically', 'RB'), ('predict', 'VB'), ('and', 'CC'), ('influence', 'VB'), ('human', 'JJ'), ('behavior', 'NN'), ('in', 'IN'), ('a', 'DT'), ('manner', 'NN'), ('that', 'WD'), ('circumvents', 'VBZ'), ('users', 'NNS'), ('"', 'POS'), ('awareness', 'NN'), ('of', 'IN'), ('such', 'JJ'), ('influence', 'NN'), ('.', '.'), ('Using', 'VBG'), ('an', 'DT'), ('intermediary', 'JJ'), ('app', 'NN'), ('"', '""'), ('Cambridge', 'NNP'), ('Analytica', 'NNP'), ('was', 'VBD'), ('able', 'JJ'), ('to', 'TO'), ('have', 'VB'), ('large', 'JJ'), ('data', 'NNS'), ('volumes-over', 'RB'), ('50', 'CD'), ('million', 'CD'), ('raw', 'JJ'), ('profiles-and', 'NN'), ('use', 'NN'), ('big', 'JJ'), ('data', 'NNS'), ('analytics', 'NNS'), ('to', 'TO'), ('create', 'VB'), ('psychographic', 'JJ'), ('profiles', 'NNS'), ('in', 'IN'), ('order', 'NN'), ('to', 'TO'), ('subsequently', 'RB'), ('target', 'VB'), ('users', 'NNS'), ('with', 'IN'), ('personalized', 'JJ'), ('digital', 'JJ'), ('advertising', 'NN')]
```

- c) Train a hmm using nltk on the penn-tree bank dataset and tag the above text. Comment on your result highlighting inaccuracies and strategies for improving performance.

```
zainqasmi@zainqasmi:~/Desktop/NLP4$ python my1C.py

=====

[('The', u'DT'), ('Cambridge', u'NNP'), ('Analytica', u'NNP'), ('scandal', u'NNP'), ('is', u'NNP'), ('more', u'NNP'), ('than', u'NNP'), ('a', u'NNP'), ('breach', u'NNP'), ('as', u'NNP'), ('Facebook', u'NNP'), ('executives', u'NNP'), ('have', u'NNP'), ('defined', u'NNP'), ('it', u'NNP'), ('It', u'NNP'), ('exemplifies', u'NNP'), ('the', u'NNP'), ('possibility', u'NNP'), ('of', u'NNP'), ('using', u'NNP'), ('online', u'NNP'), ('data', u'NNP'), ('to', u'NNP'), ('algorithmically', u'NNP'), ('predict', u'NNP'), ('and', u'NNP'), ('influence', u'NNP'), ('human', u'NNP'), ('behavior', u'NNP'), ('in', u'NNP'), ('a', u'NNP'), ('manner', u'NNP'), ('that', u'NNP'), ('circumvents', u'NNP'), ('users', u'NNP'), ('awareness', u'NNP'), ('of', u'NNP'), ('such', u'NNP'), ('influence', u'NNP'), ('Using', u'NNP'), ('an', u'NNP'), ('intermediary', u'NNP'), ('app', u'NNP'), ('Cambridge', u'NNP'), ('Analytica', u'NNP'), ('was', u'NNP'), ('able', u'NNP'), ('to', u'NNP'), ('have', u'NNP'), ('large', u'NNP'), ('data', u'NNP'), ('volumes', u'NNP'), ('over', u'NNP'), ('50', u'NNP'), ('million', u'NNP'), ('raw', u'NNP'), ('profiles', u'NNP'), ('and', u'NNP'), ('use', u'NNP'), ('big', u'NNP'), ('data', u'NNP'), ('analytics', u'NNP'), ('to', u'NNP'), ('create', u'NNP'), ('psychographic', u'NNP'), ('profiles', u'NNP'), ('in', u'NNP'), ('order', u'NNP'), ('to', u'NNP'), ('subsequently', u'NNP'), ('target', u'NNP'), ('users', u'NNP'), ('with', u'NNP')]
```

### Question 2:

- a) Write briefly on the evaluation problem, decoding problem and the learning problem in HMMs. You should describe what the problem is and what the solution is. Mention the

name of any algorithms used to solve the problem and provide an intuitive explanation of the algorithm. Use as less mathematical notation as possible and try writing in your own words.

Generally, the learning problem is how to adjust the HMM parameters, so that the given set of observations (called the training set) is represented by the model in the best way for the intended application. Thus it would be clear that the "quantity" we wish to optimize during the learning process can be different from application to application. In other words there may be several optimization criteria for learning, out of which a suitable one is selected depending on the application.

There are two main optimization criteria found in ASR literature; Maximum Likelihood (ML) and Maximum Mutual Information (MMI). The solutions to the learning problem under each of those criteria is described below.

REF:

<http://jedlik.phy.bme.hu/~gerjanos/HMM/node9.html#SECTION00243000000000000000>

- b) Consider a Hidden Markov Model with three hidden states: N (noun), V (verb) and O (other). Let all transitions between states be equally probable. Consider the following possible outputs:

N: *mimsy* | *borogoves*

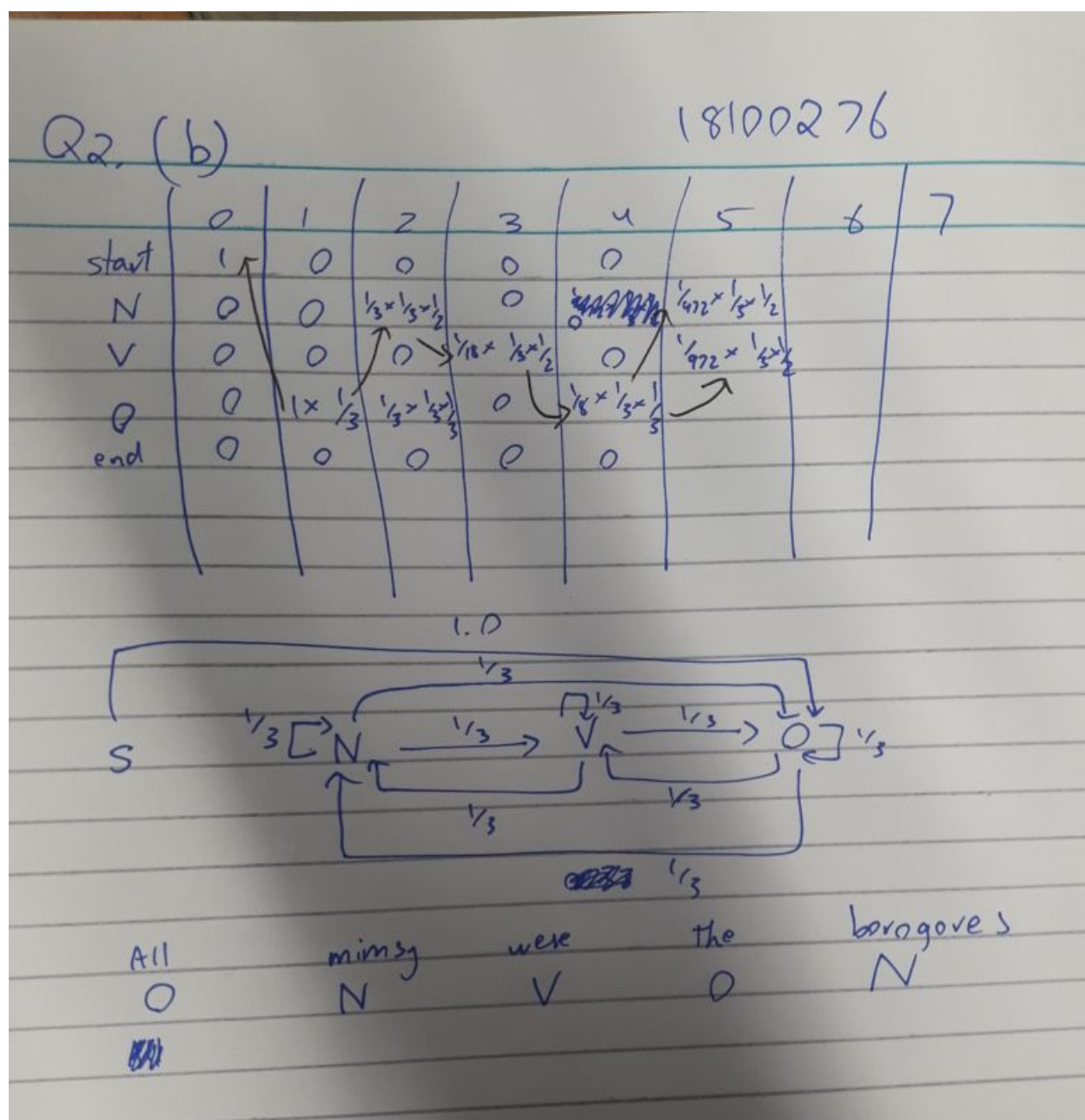
V: *were* | *borogoves*

O: *All* | *mimsy* | *the*

Let all these outputs be also equiprobable.

Consider the sentence "*All mimsy were the borogoves*". Find the most probable tag sequence





### Question 3:

- a) Explain how you would use a deep learning model for POS tagging. Also comment on the input representation to the model.

There are different approaches to automatic PoS tagging: rule-based approaches use linguistic knowledge to formulate simple rules that assign a part of speech to an ambiguous word using context information; statistical approaches use the statistics collected from ambiguously or unambiguously tagged texts to estimate the likelihood of each possible interpretation of a sentence or text portion so that the most likely disambiguation is chosen.

We will use a neural approach and refer to a text as unambiguously tagged or just tagged when each occurrence of each word has been assigned the correct PoS tags. Words receiving the same set of PoS tags are said to belong to the same ambiguity class; for example, the words *tailor* and *book* belong to the ambiguity class {noun, verb}.

### **Input Representation:**

We will train a single-layer perceptron to produce the PoS tag of a word as a unary vector. Input is a window of the  $p = 2$  or  $p = 3$  words before the current word, the current word, and the  $f = 1$  or  $f = 2$  words after it; the following words and the current word are represented by a vector in which the  $j$ th component is the frequency with which that word gets the  $j$ -th PoS tag in a large unambiguously tagged corpus; the previous words are represented by a linear combination of this vector and the output produced by the neural network for the corresponding word.

Recurrent neural networks (RNN) have been used since their inception for predictive tasks in the natural language processing arena; it predicts the next word in a synthetic corpus of simple, randomly generated two- or three-word grammatical sentences drawn from a small vocabulary of untagged nonambiguous words. After learning, the state representation the network developed after reading a word grouped words in categories that could be identified with classical tags such as noun, transitive verb, intransitive verb, etc. For the predictive task to succeed, the network had developed a way to obtain a syntactic representation of the portion of text seen up to that point.

### **Training:**

In our approach, RRN will be trained in two phases. First, the training text is ambiguously tagged using a lexicon analyser, assigning each word its ambiguity class, that is, a set of possible PoS tags.

In the first phase, the RRN is trained to predict the ambiguity class of the next word from the ambiguity class of the current word and from the classes of preceding words in the sentence. In this way, the SRN will learn to develop in its state a syntactic representation of each particular ambiguity class in its previous context which allows it to make its best prediction about the ambiguity class of the following word.

In the second phase, after successful training, each word in the text is tagged with the hidden state vector computed for it by the network; then, for each word a perceptron is trained to predict its part of speech from the state vector assigned to the word which is  $f$  positions to the right of it. The number  $f$  represents the amount of right (forward) context needed to disambiguate a word. An output unit is assigned to each PoS in this second phase.

Thereafter, the combination of the state part of the RRN and the perceptron can be used to determine the PoS tag of words in new sentences.

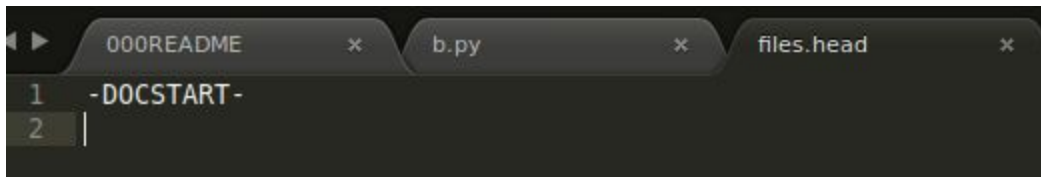
<http://www.dlsi.ua.es/~japerez/pub/pdf/ijcnn2001.pdf>

## Part 2: Entity Recognition

### Question 4:

- a) Study the CONLL-2003 NER shared-task dataset. Provide a screenshot of the 'head' of the file and explain how it can be interpreted.

Content inside files.head:



```
1 -DOCSTART-
2 |
```

The only plausible interpretation is this is the head of the file and marks the start of document.

I could not get access to the whole document as that requires access to RCV1 for English corpus which requires institution's access (LUMS) that was not granted in time. Please refer to the email below.

request for Reuters corpus

0 1 ✓



Ellis, Angela G (Fed) <angela.ellis@nist.gov>

Today, 5:16 PM



Greetings Muhammad,  
Thank you for submitting your individual agreement. Please remember to submit your organization agreement also. I will send the data requested as soon as I receive the signed form. Thank you.

Peace be with you,  
Angela

...



Muhammad Zain Qasmi

Today, 4:46 PM

reuters-request@nist.gov ✓



Download Save to OneDrive - Higher Education Commission

Name: Muhammad Zain Qasmi  
Postal address: Room 422, Hostel M5, Lahore University of Management Sciences,  
Opposite Sector U, DHA, Lahore Cantt, Lahore  
Requesting for: RCV1

Safety & Peace,  
Muhammad Zain Qasmi

“Atlas Honda is expected to achieve sales of 1.1 million units by end of its financial year ending March 31, while it aims to hit sales of 1.3m bikes in its next financial year, a Honda dealer said.”

- b) Use a pre-trained NER model to identify the named entities in the above sentence. (use NLTK)

**ANSWER:**

```

zainqasmi@zainqasmi:~/Desktop/NLP4/q4/b$ python b.py
NLTK :: ['Atlas Honda', 'Honda', '']
SpaCy:: [Atlas Honda, 1.1 million, year ending March 31, 1.3m, its next financial year, Honda]

```

```

000README x b.py x b2.py x
1 ##### NLTK #####
2
3 from nltk.tree import Tree
4 from nltk import ne_chunk, pos_tag, word_tokenize
5
6 txt = """Atlas Honda is expected to achieve sales of 1.1 million units by end of its
7 financial year ending March 31, while it aims to hit sales of 1.3m bikes in its next
8 financial year, a Honda dealer said."""
9
10 def id_entities(myStr):
11     lump1 = word_tokenize(myStr)
12     lump2 = pos_tag(lump1)
13     lump3 = ne_chunk(lump2)
14     block = []
15     currBlock = []
16     for one in lump3:
17         if type(one) == Tree:
18             currBlock.append(" ".join([token for token, pos in one.leaves()]))
19         elif currBlock:
20             entity = " ".join(currBlock)
21             if entity not in block:
22                 block.append(entity)
23                 currBlock = []
24         else:
25             continue
26     if block:
27         entity = " ".join(currBlock)
28         if entity not in block:
29             block.append(entity)
30     return block
31
32 print "NLTK :: ", id_entities(txt)
33
34 ##### SpaCy #####
35
36 import spacy
37 nlp = spacy.load('en')
38 doc = nlp(txt.decode('utf8'))
39 print "SpaCy:: " , list(doc.ents)

```

- c) Provide a brief summary (6 lines) of what a bidirectional LSTM-CRF model (paper <https://arxiv.org/pdf/1508.01991.pdf>) is.

## ANSWER:

A recurrent neural network (RNN) is a class of artificial neural network where connections between units form a directed graph along a sequence. This allows it to exhibit dynamic temporal behavior for a time sequence. RNN's are applicable to tasks such as unsegmented,



connected handwriting recognition or speech recognition.

LSTMs are the same as RNNs, except that the hidden layer updates are replaced by purpose-built memory cells so the storage can be under direct control by the neural network. . The storage can also be replaced by another network or graph, if that incorporates time delays or has feedback loops. Such types of controlled states are what makes LSTMs and it is why LSTMs are better at finding and exploiting long range dependencies in the data.

Conditional random fields (CRFs) fall into the sequence modeling family and are a class of statistical modeling method often applied in pattern recognition and machine learning and used for structured prediction.

Combining a bidirectional LSTM network and a CRF network to form a BI-LSTM-CRF network helps with the use of future input features in addition to the past input features and sentence level tag information used in a LSTM-CRF model. The extra features are shown to boost tagging accuracy.

- d) Provide a detailed outline (point wise) of how to use such a model for NER tagging. More specifically, mention the dataset to use, the input representation of words, the ideal number of hyper parameters, the cost function and the overall architecture of the deep learning model.

## ANSWER:

### Dataset to use:

#### CoNLL2003

In the CoNLL2003 task, the entities are LOC, PER, ORG and MISC for locations, persons, organizations and miscellaneous. The no-entity tag is O. Because some entities (like New York) have multiple words, we use a tagging scheme to distinguish between the beginning or the inside of an entity.

### Input representation of words:

For our implementation, we are assuming that the data is stored in a .txt file with one word and its entity per line in the following manner

EU B-ORG

rejects O

German B-MISC

call O

to O

boycott O

British B-MISC

## Ideal number of hyper parameters:

I would suggest trying to understand what kind of mistake the model makes then try to fine-tune it. Finally, re-train on the whole dataset and try different set of hyperparameters.

## Cost function:

At this stage, each word is associated to a vector that captures information from the meaning of the word, its characters and its context. And we have two options here:

**softmax:** normalize the scores into a vector

**linear-chain CRF:** here we make use of neighbouring tagging decisions unlike softmax.

For instance, in New York, the fact that we are tagging York as a location should help us to decide that New corresponds to the beginning of a location.

## Overall architecture:

Like most of the NLP systems, ours is gonna rely on a recurrent neural network at some point. But before delving into the details of our model, I will break it into 3 pieces:

Word Representation: I will need to use a dense representation  $w \in \mathbb{R}^n$  for each word. The first thing I can do is load some pre-trained word embeddings ([GloVe](#), [Word2Vec](#), [Senna](#), etc.). I am also going to extract some meaning from the characters. A lot of entities don't even have a pretrained word vector, and the fact that the word starts with a capital letter also helps.

Contextual Word Representation: for each word in its context, I need to get a meaningful representation  $h \in \mathbb{R}^k$ . So obviously, I am gonna use an LSTM here.

Decoding: Now once I have a vector representing each word, I can use it to make a prediction.

- e) Find a pre-trained deep learning model in python (any package but mention the package name in this document and provide screenshots) and use it to on the test sentence provided above.

## ANSWER:

**Package name: Sequence Tagging with Tensorflow**

**Model Used: bi-LSTM + CRF with character embeddings for NER and POS**

[https://github.com/guillaumegenthial/sequence\\_tagging](https://github.com/guillaumegenthial/sequence_tagging)

<https://guillaumegenthial.github.io/sequence-tagging-with-tensorflow.html>

Atlas Honda is expected to achieve sales of 1.1 million units by end of its financial year ending

Find Entities

[illegible][illegible][illegible]

# The End 😊