

LIN 127

Text Processing and Corpus Linguistics

Week 10

Python for Text 2 (and Beyond)

Notes from Assignment 6

- Run POS taggers (and other models) on full sentences -
What tag is “run” if we have:
 - Just “run”
 - Verb
 - “I went on a run”
 - Noun

Notes from Assignment 6

- Careful with negative indexing!
- In `left_adjectives`:

```
for idx, token in enumerate(doc):  
    if token.text == target_word and  
        doc[idx - 1].tag_ == 'JJ':  
        adj_counts[doc[idx - 1].text] += 1
```

Classification!

Is this spam?

YOUU Have Been PAID 📦 Check Yourr Account-Now 📦 💰 \$1000.00 💰 User ID#372-75784 ⌵ Spam x



CashAP 💰 uqaxeuhkmhorygq@extentor.help via pm.mtasv.net

📧 to contigome ▾

Sat, Mar 18, 11:08 PM



-THIS MESSAGE WAS SENT FROM A TRUSTED SENDER.

CONGRATULATIONS ****@gmail.com !

A.balance..OF **\$1000.00** Is AVAILABLE FOR..your ***CashApp*.Accountt**

Thiss.TRANSACTION.may.Only.appearr. On.your.ACC0UNTT..after.
VALIDATE.your.Info.

03/2023

FUNDING.For: **** EMAIL: ****@gmail.com

Balance Amount: \$1000.00

PAYOUT:

\$1000.000

Confirm Here

Memo PAYOUT

SIGNATURE ****

Is this spam?

URGENTLY



Samir Khuller <drwhitneywhitaker@gmail.com>

To: ○ Rob Voigt



Mon 4/3/2023 10:34 AM

Hello,

Are you in the office ?

Samir Khuller

Chair, Department of Computer Science

Office: Mudd Room 3017

Phone: 847-491-2748

Email: samir.khuller@northwestern.edu

Classification is the task of assigning labels

Which is spam?

Congrats Andy Spellman!!!! You
have won the sweepstakes!!!
Click here to receive your FREE
\$50 Costco gift card!

Andy Spellman,

Thank you for your purchase of a
\$50 Costco giftcard. Your order
details are listed below.

Classification is the task of assigning labels

Basic approach: rule-based!

Rules based on combinations of words or other features

- spam: black-list-address OR
 (“dollars” AND “you have been selected”)

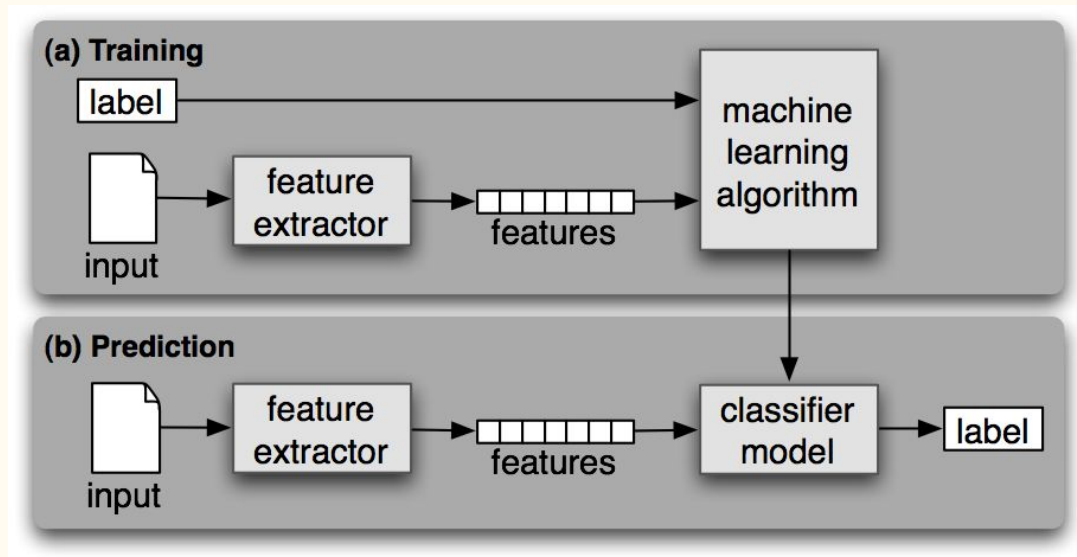
Accuracy can be high

If rules carefully refined by expert

But building and maintaining these rules is expensive

Classification is the task of assigning labels

- ML approach: use known input-label pairs to train an algorithm to decide which category a previously unseen input belongs to



Features are leveraged to make predictions

- Features can take many forms:
 - Counts of particular words
 - Counts of n -grams
 - multi-word phrases of length n :
e.g. trigrams are three-word phrases (“so it goes”)
 - Numerical values (e.g., average concreteness)
 - Word vector dimensions
- Each is part of a mathematical representation of a document

Features are leveraged to make predictions

- “Learning” is most frequently the process of assigning numerical weights to each feature

NLTK movie review classification example:

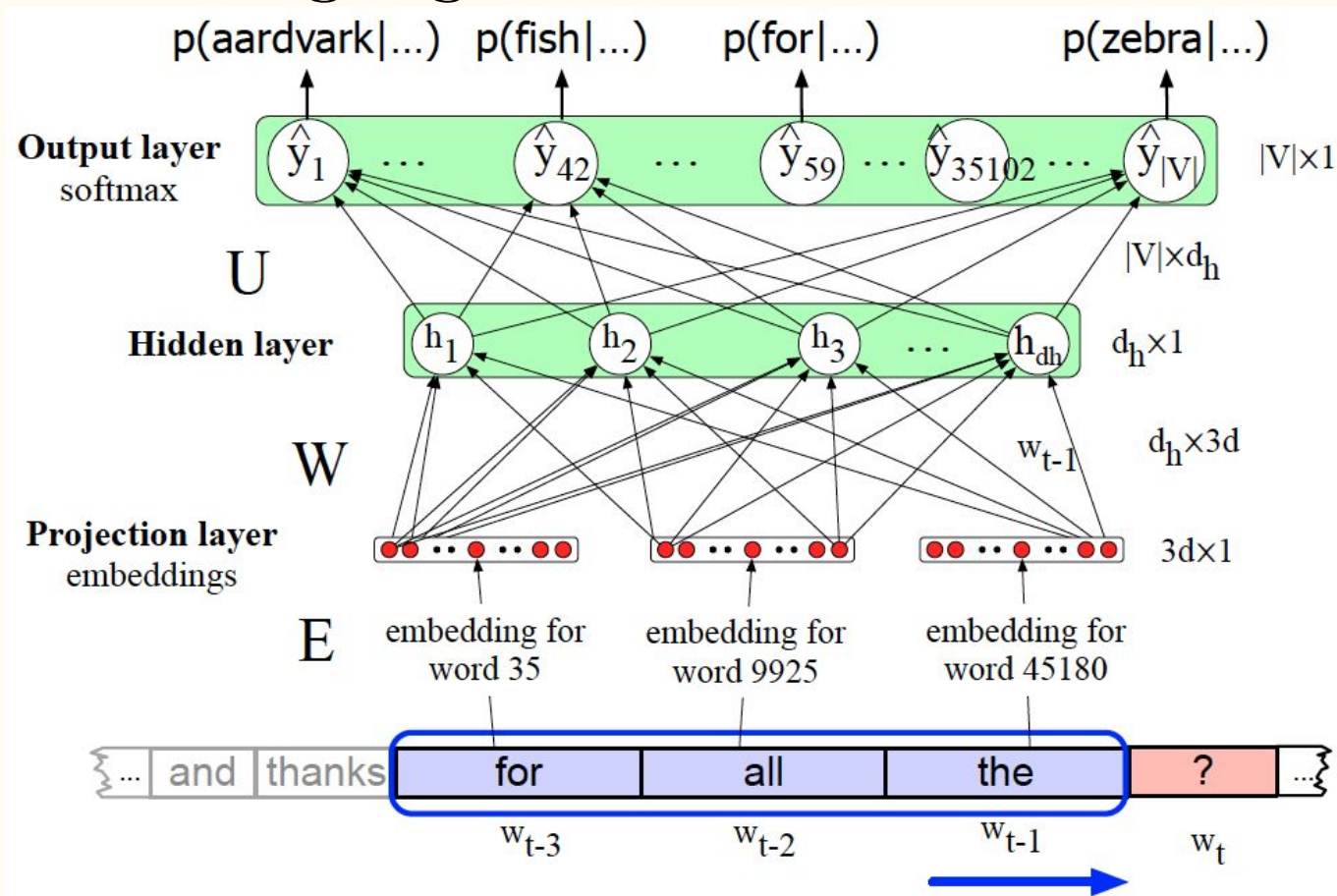
```
>>> print(nltk.classify.accuracy(classifier, test_set)) ❶  
0.81  
>>> classifier.show_most_informative_features(5) ❷  
Most Informative Features  
contains(outstanding) = True          pos : neg      =      11.1 : 1.0  
contains(seagal) = True              neg : pos      =       7.7 : 1.0  
contains(wonderfully) = True         pos : neg      =       6.8 : 1.0  
contains(damon) = True               pos : neg      =       5.9 : 1.0  
contains(wasted) = True              neg : pos      =       5.8 : 1.0
```

<https://www.nltk.org/book/ch06.html>

But, hand-engineered features
are sort of out of date

- Neural networks / LLMs start from an abstract feature set induced from data (like word embeddings)
- ... and induce intermediary features from the data
- Key task is Language Modeling: given some context, predict the next word or a masked-out word

Neural Language Model



Why Neural LMs work better than N-gram LMs

Training data:

We've seen: I have to make sure that the cat gets fed.

Never seen: dog gets fed

Test data:

I forgot to make sure that the dog gets ____

N-gram LM can't predict "fed"!

Neural LM can use similarity of "cat" and "dog" embeddings to generalize and predict "fed" after dog

Where To Go From Here

Congratulations!

*You are all
officially
computational
linguists!*

Programming and text processing are very useful

- The skills you've learned are broadly applicable to linguistic and non-linguistic applications
- Try out your new computational tools and thinking in other parts of your life!

Other things you are now well-equipped to start learning

- Version control (git, see [these lectures](#))
- Data science (see e.g. [pandas](#) and [numpy](#))
- Machine learning (see e.g. [scikit-learn](#), [pyTorch](#))
- Web scraping (see e.g. [BeautifulSoup](#), [scrapy](#))
- Dynamic web programming (see e.g. [Flask](#) or [Django](#))
- App development (see e.g. [Kivy](#), [Python recommendations](#))
- Game programming (see e.g. [pygame](#) or [Godot](#))

Natural Language Processing (NLP) and Computational Linguistics (CL)

- NLP = more engineering, everything is a “task”,
focus on system performance
- CL = computational social science, using and developing
NLP tools for social, linguistic, humanistic questions
- No need, of course, to strictly pick a camp!

AI and LLMs

- Modern “neural networks” - I recommend this book:

<https://d2l.ai/>

- and these more advanced lectures

(Stanford CS224N):

<https://www.youtube.com/watch?v=DzpHeXVSC5I&list=PLoROMvodv4rOaMFbaqxPDoLWjDaRAAdP9D>



AI and LLMs

- Or more practically:

https://huggingface.co/docs/transformers/en/tasks/sequence_classification

Closing out the Class!

Final submission dates:

All assignments, final project - 3/17 (reviewable on 3/18)

Final self-evaluation - 3/17

(absolute final 3/19 at 10am)

Walkthrough of Final Self-Evaluation

Thank you!

It's been a privilege and
a joy to teach this class.
