COMP3420 Lesson 13

Greg Baker

2023-05-29

MACQUARIE University

Today’s half-session

- A checklist of learning outcomes for the course
- Walk-through of a sample exam

Week 7

Memorize

Things you should **memorize** (because you will use them in real life, too):

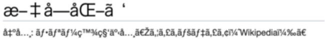
- Regular expressions before PCRE (i.e. before Perl)
    - `* [] ? . ^ $ [^A-Z] [a-z] abc \`
- The names of code pages for languages that you speak:
    - e.g. CP1252, ISO-8859-1 (Western European languages)

In 5 years' time, email me to say thanks when you realise you are using these things all the time.

Be able to define

The golden rule of text *When you have a stream of bytes, and you don't know how it was encoded, you have a useless stream of bytes.*

Mojibake What it is (and what it looks like)



æ–‡å—åŒ–ã ‘

å‡ºå…¸: ãƒ•ãƒªãƒ¼ç™¾ç§‘äº‹å…¸ã€Žã‚¦ã‚£ã‚ãƒšãƒ‡ã‚£ã‚¢ï¼ˆWikipediaï¼‰ã€

Fixed-width encoding vs varying width encoding

Python code you should be able to explain

re.compile()	Regex compile
x.decode('ascii')	Take a binary and interpret it as ASCII
x.encode('utf-8')	Take a string and make it UTF-8
open('filename', encoding='utf-8')	Read a file with an encoding
except UnicodeDecodeError:	Even when you aren't using Unicode.

Terms to know: Unicode Transformation Formats

UTF-32 Fixed length encoding, 4 bytes for every character (space inefficient); just store the Unicode code point.

UTF-16 Varying length encoding, mostly used in Microsoft Windows and JavaScript. Incompatible with ASCII.

UTF-8 Varying length encoding, compatible with ASCII, most widely used.

Manipulate: UTF-8 ⇔ Unicode points

(Don't need to memorise this, just understand how it works)

First code point	Last code point	Byte 1	Byte 2	Byte 3	Byte 4
U+0000	U+007F	0xxxxxxx			
U+0080	U+07FF	110xxxxx	10xxxxxx		
U+0800	U+FFFF	1110xxxx	10xxxxxx	10xxxxxx	
U+10000	U+10FFFF	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

Week 8

Define

Vocabulary Number of distinct words in a corpus

Corpus size Number of words in a corpus

Happax legomonon A word that appears only once in a corpus — typically 40%-60% of words

Bigram Two word sequences

Stop word Words that get ignored

Analyse and Interpret Parameters

Zipf's Law $f(r) = \frac{C}{r^s}$

Heap's Law, Herdan's Law $V = kN^\beta$

Search term hit rate estimation $\frac{N}{V} = \frac{N}{kN^\beta} = \frac{N^{1-\beta}}{k}$

Term usefulness estimate for a classifier $\frac{C}{V} = \frac{N}{kN^\beta L} = \frac{N^{1-\beta}}{kL}$

(The last two are not in the class textbooks.)

- No need to memorise these; in real life you can look them up!
- Be able to explain the kinds of corpora that would lead to different values for s and β, and the implications for search, classification and author identification.

Python code you should be able to explain

nltk.sent_tokenize(x) Break document into sentences.

nltk.word_tokenize Break sentence or document into words using rules typical of English-language texts.

tfidf = TfidfVectorizer(); vecs = tfidf.fit_transform(x) `sklearn`'s vectorizer, creating TFIDF vectors

sklearn.metrics.pairwise.cosine_similarity(x,y) A function to measure similarity between two vectors

Manipulate

Be able to perform one iteration of Byte-pair encoding by hand. (Not really a useful skill, but if you can do it, you will have an intuition for how and why it works.)

Repeat:

- Choose the two symbols that are most frequently adjacent in the training corpus (say 'A', 'B')
- Add a new merged symbol 'AB' to the vocabulary
- Replace every adjacent 'A' 'B' in the corpus with 'AB'.

Until *k* merges have been done, or the vocabulary is the target size.

Week 9

Python code you should be able to explain

Dense(1, activation='sigmoid') Also known as logistic regression.

Dense(10, activation='softmax') The last layer in a multi-class classification problem.

Dense(20, activation='relu') A layer which creates regions out of a dataset.

t = TextVectorization(); t.adapt(x) The Keras way of turning documents into vectors

Analyse and interpret

- `vocab = vectorizer.get_vocabulary()`
- `weights = model.get_weights()[0][:,0]`
- `print(zip(vocab,weights))`

Weights in a logistic regression describe the impact of a word in a document towards one classification or the other. (Big positive numbers = “strongly associated with that class”)

Be able to define

Data irrelevancy Nothing in X predicts y.

True positive, true negative, false positive, false negative What the system predicted vs what ground truth said

Precision, recall, accuracy Metrics for evaluating a classifier

Overfitting The model is memorizing the data rather than generalizing

Underfitting The model is unable to get good results because it isn't capture

Terms to Know

GDPR and PIPL EU and Chinese laws that have *extra-terrorality* and require explainability (even if your company is in Australia) if the model is used for something important that might affect someone's life.

F1 score A balanced score that captures both precision and recall. Trading off precision for recall will generally worsen F1 score

Be able to give examples of

- A classifier that would need explainability under GDPR or PIPL.
- Reasons you might trade off explainability vs accuracy
- An unbalanced data set
- Data sets that you would use each metric for
- Things you might do to reduce overfitting and underfitting

Week 10

Define

Embedding A mapping between language (usually words) and numeric vectors

Contextual embedding An embedding that distinguishes between different meanings for the same written word

Non-contextual embedding An embedding that uses the written form and doesn't distinguish meaning

Bag-of-words Vectorise by giving each word its own dimension

Hypernym / hyponym A hypernym is a more general concept; a hyponym is a specific example. "Animal" is a hypernym of "dog".

Null region The volume of linear space where all Relu functions are zero, and all points are indistinguishable.

Context drift Models perform worse over time; the future brings new words and the task may change

Python code you should be able to explain

- `s = wordnet.synsets(x)` What different meanings does the word `x` have?
- `s[0].definition()` For the first meaning, what is its definition?
- `s[0].lemmas()` What words express the first meaning?
- `s[0].hypernyms()[0]` What is the first hypernym of the first meaning?

Be able to give examples of

- Problems with the bag-of-words approach
- Ways to map words into numeric vectors, and whether they are
- Context and non-contextual embeddings

Python code you should be able to explain

```
c=Constant(embedding_matrix)]
embedding_layer = keras.layers.Embedding(
   input_dim=len(voc),
   output_dim=50,
   embeddings_initializer=c,
   trainable=False)
```

What does trainable=False mean? Why would we set this?

Week 11

Be able to define

Temperature A controllable parameter used in a sequence-to-sequence which increases the randomness of the selection from the probability distribution of next tokens.

Sequence-to-sequence model / text generation A model designed to predict the next character, word or token in a stream

AI alignment problem the challenge of designing artificial intelligence systems that reliably understand and execute human intentions and values, even when operating at superhuman levels of performance.

Misc

- Recognize architecture diagrams for RNNs, LSTM and Transformers
- Predict the results of changing the temperature of a model's output
- Follow Chollet's Guidance to identify the most likely-to-be-effective architecture for a classifer problem. i.e. Make a choice of classifier given the following guidance: $\frac{\text{Number of samples}}{\text{Mean sample length}}$
  - < 1500 Bag of words + logistic regression
  - $< 15,000$ Pretrained embeddings + LSTM or Transformer
  - $> 15,000$ Learn embeddings + Transformer

Week 12

Give examples of

- Differences between GPT-3.5 and GPT-4
- Prompt injection
- The purpose and effects of RLHF (Reinforcement learning from human feedback)
- Tasks that GPT-4 is unable to do

Define

Prompt injection If a prompt to a language model includes text from a third party, that third party can manipulate the language model to produce uncontrollable responses.

Context length The amount of input text the model can take into account when generating a response or prediction.

Hallucination The model generates output that is plausible but untrue.