CS 1671/2071
Human Language Technologies

Session 20: Exam review

Michael Miller Yoder

March 31, 2025

University of Pittsburgh | School of Computing and Information

Course logistics

- In-person exam is **this Wed Apr 2**
  - Covers Modules 2-4
  - Paper exam: true/false and written questions
  - Some math calculations but no calculators or devices are permitted. It’s fine to leave things in fractional form.
  - One page of double-sided notes will be permitted
    - Some formulas will be provided with the exam
- Homework 3 has been released and is now **due Apr 14**
  - LLM prompting
  - Use class OpenAI API account. Copy the key in the Canvas announcement

Course logistics

- Project resources
  - Can also use class OpenAI API account for your projects
  - 5 TB class storage is available on CRCD at /ix/cs1671_2025s
  - To access the CRCD through the command line: ssh <pitt username>@h2p.crc.pitt.edu
  - Look into CRCD user manual for SLURM jobs for running Python scripts, otherwise use JupyterHub

Exam will cover Modules 2-4

MODULE 1	Prerequisite skills for NLP	text normalization, linear alg., prob., machine learning

	Approaches	How text is represented	NLP tasks
MODULE 2	statistical machine learning	n-grams	language modeling text classification
MODULE 3	neural networks	static word vectors	language modeling text classification
MODULE 4	transformers and LLMs	contextual word vectors	language modeling text classification sequence labeling

MODULE 5	NLP applications and ethics	machine translation, chatbots, information retrieval, bias

Overview: Exam review

- **Your questions: ask me anything**
- Go through high-level concepts from Modules 2-4
    - Feel free to ask questions throughout

Questions?

Module 2: N-grams and statistical NLP

Module 2 N-grams and statistical NLP: how text is represented

- n-grams
- term-document matrices
  - Possibly weighted with tf-idf
- term-term matrices
  - Possibly weighted with PPMI

Module 2 N-grams and statistical NLP: algorithms

- N-gram language modeling
- Logistic regression for text classification
    - Parameters (one for every feature) learned with stochastic gradient descent

Module 3: Neural networks and word2vec

Module 3 Neural networks and word2vec: how text is represented

- Dense word embeddings (vectors), learned with e.g. word2vec
- Word2vec
    - Logistic regression to classify words as occurring together or not
        - Positive examples: words that occur together in a corpus within a context window
        - Negative examples: random words with target word
        - Example from a part of a corpus: “the dog barked two times”. Target word is “dog”
            - Positive example: (dog, barked)
            - Negative example: (dog, interstellar)
    - From randomly initialized word vectors, moves vectors for words that co-occur together closer in vector space

Module 3 Neural networks and word2vec: algorithms

- Feedforward neural networks for text classification
  - Parameters learned from stochastic gradient descent

Module 4: Transformers and LLMs

Module 4 Transformers and LLMs: how text is represented

- Contextual word embeddings: different vector (embedding) for every token
- Embedding for each word type + embedding for position in the sentence

Module 4 Transformers and LLMs: algorithms

- Transformer models and self-attention
  - One output vector for every input token after many transformations
  - Output vectors incorporate information from other words in a sentence through self-attention
- Pretrained transformer-based models: LLMs
  - Decoder-only models trained on (causal, left-to-right) language modeling: GPT series models
  - Encoder-only models trained on masked language modeling: BERT family of models
  - Encoder-decoder models first encode an input sentence to a vector and then use that as input to start doing language modeling (decoding) to produce an output sentence
- Finetune pretrained models for specific tasks or prompt them with zero-shot, few-shot or chain-of-thought prompting

Questions?

Best of luck on the exam!