

AKANKSHA DARA

Stony Brook University | Apple | BITS Pilani

📍 Stony Brook, New York

✉ akankshadara7@gmail.com

☎ +1-631-710-1468

in akanksha-dara

🔗 AkankshaDara

EDUCATION

MS in Computer Science

Stony Brook University (GPA: 3.92/4.00)

📅 2020-2021

📍 Stony Brook, NY

- Graduate Research Assistant at NLP Lab (Advisor: Prof. Ritwik Bannerjee).
- Graduate Teaching Assistant for the course CSE-216 (Programming Abstractions).

B.E.(Hons.) in Computer Science

Birla Institute of Technology and Science, Pilani (GPA: 7.9/10.0)

📅 2014-2018

📍 Pilani, India

- Teaching Assistant for the course CS-F111 (Computer Programming)
- Undergraduate Research Thesis at the University of Melbourne, Australia
Fall 2017 – (Grade: 10/10)

EXPERIENCE

SOFTWARE DEVELOPMENT ENGINEER

🍏 Apple Inc.

📅 October 2018 – January 2020

📍 Hyderabad, India

- **Java Development:** Implemented a **Solr cluster** solution to facilitate efficient management of the test cases. Delivered business logic in back-end APIs and an interactive web application.
- Automated the process of publishing code coverage reports using JaCoCo (Java Code Coverage) as an agent across all the servers by generating the build files dynamically; analysed the coverage for reflection classes in the source code. Used these results to identify redundancy in code execution.

RESEARCH ASSISTANT

Memory and Language Lab, University of Melbourne

(Advisor: Prof. Simon Dennis)

📅 July 2017 - December 2017

📍 Melbourne, Australia

- Built a computational model for language processing that characterized sentence processing and learning as an interaction of three lexical, syntactic, and relational memory systems that operate on distributed instance based knowledge representations.
- Conducted several experiments to optimize the model with different input feature sets, objective functions, corpora and tokenization schemes. This helped in gaining insight into the operation of the model, in particular the impact of interference on memory based algorithms (used the stochastic gradient descent algorithm for learning).
- Achieved a perplexity of 42 on a vocabulary size of 65,536.

KEY COURSES

- Machine Learning
- Data Science Fundamentals*
- Data Visualization
- Natural Language Processing*
- Probability & Statistics
- Data Structures & Algorithms

ACADEMIC PROJECTS

Semantic Textual Similarity (STS) of Clinical Notes (Master's Thesis)

- Developed an ensemble model for computing the semantic similarity between paired text snippets of medical records using the state-of-the-art parallel deep learning architectures with traditional machine learning algorithms.
- The similarity features were based on token-based metrics, sequence-based metrics, transformer-based BERT embeddings, and a novel least common ancestor (LCA) approach based on the medical terms queryable on the MeSH (Medical Subject Headings) database.
- The model achieves a score of 0.875 on the test dataset.

CNN with word2vec for Sentence Classification

- Implemented a Convolutional Neural Network trained on top of pre-trained word vectors (word2vec) for sentence-level classification tasks. Compared the classification results on multiple benchmarks.
- Further modified the model to use both task-specific and static vectors. Achieved an accuracy of 89% on the Subj dataset.

Sarcasm Detection: Behavioral Modelling Approach

- Implemented sarcasm detection as a binary classification problem using the behavior modelling approach. Identified sarcasm as three different forms of expressions.
- This was achieved by leveraging users' historical information of past tweets and by identifying sarcasm as a contrast of sentiments, and as a function of familiarity; made some more enhancements to handle the use of emoticons and complex hashtags to improve the results.

Text Summarization using Audio Retrieval

- A textual summary generator based on the LexRank algorithm that takes into account the frequencies of words spoken as appearing in the audio waveform.

Compiler Construction

- Implemented a compiler for the language ERPLAG. The project was done in a pipelined manner with its various stages being: Lexer - Parser - Abstract Syntax Tree generation - Type-checking, Semantic checking and Assembly Code Generation.

SKILLS

- Over 50,000 lines: C++
- Over 10,000 lines: python, java, C
- **Machine Learning:** sklearn, PyTorch, pandas, NLTK
- **Search and Databases:** MySQL, Solr
- **Operating system :** Windows, Linux, macOS
- **Web:** HTML/CSS/JavaScript, Spring, Flask, D3.js