Deeksha Koul

in ⌂ Ⓢ deekshakoul317

Email: deekshakoul0245689@gmail.com

Mobile : 6006970893

EDUCATION

- **Indian Institute of Science** — Bengaluru, KA
  Master of Technology in Computer Science and Engineering; CGPA: 8.2/10.0 — *Aug. 2019 – Jul. 2021*

- **Dhirubhai Ambani Institute of Information and Communication Technology** — GandhiNagar, GJ
  B.Tech Honors in ICT with Minors in Computational Science; CGPA: 8.55/10.0 — *Aug. 2013 – May. 2017*

WORK EXPERIENCE

Machine Learning Engineer 2 — *January 2023 - Present*
Moj Livestream Team, Sharechat, Bangalore

- **Ranker model for live feed:** Developed and implemented a multi-gate mixture of expert model to enhance the feed for the moj-livestream. Achieved a 0.42% relative lift in DAU(retention) and a 1.5% relative lift in engagement metrics.
- Worked on **Go implementation** of XGBoost model inferencing that reduced service's latency by 20ms, improving the overall user experience.
- Developed a machine learning-based **notification delivery system** for Sharechat chatrooms, leveraging user, user-creator, and creator history. This system resulted in a 0.02% gain in Daily Active Users (DAU) and a 17% increase in Click-Through Rate (CTR).
- Tech used: Airflow, OpenLens(for Kubernetes pod monitoring), pytorch, polars, BigQuery, eppo(for experiment monitoring)

Research Engineer — *January 2022 - December 2022*
R&D Team, Raxter Inc., Gujarat

- **Developed advanced search experience for researchers:**
  Utilized weak supervision techniques to generate one-liner TLDR summaries based on abstract content. Leveraged the TextRank model to extract key concepts from abstracts, enhancing content understanding. With key concepts and TLDR, build a complete end-to-end pipeline that provide users to explore papers based on "Similar Problem" or "Similar Methodology".
- **Paraphrase generation with diversity in mind:**
  Implemented classical NLP techniques like active-passive, synonym, negation to generate paraphrases for a sentence. Developing encoder-decoder based model to paraphrase sentence with lower iThenticate score and faster inference time compared to classical.
- Tech used: spacy, weak supervision, Snorkel, PyTextRank, Rubrix, pytorch, ElasticSearch(Kibana), Python, FastApi, Git, Redis

Trainee Software Engineer — *August 2021 - December 2021*
Search and Recommendation Team, Monster India, Bangalore.

- **Auto-Suggestions on Search:** Utilized semantic and lexical text matching techniques to align manually entered designations with system-owned designations. Implemented Word2Vec algorithm on job-related data to identify similar terms for a given word or set of terms. Target metric achieved: increase in CTR for system owned tags.
- Tech Used: fuzzywuzzy, fuzzymatcher, dedupe, Word2Vec, Flask, Uvicorn, Nginx, Docker

Application Engineer — *June 2017 - July 2019*
HCM Team, Oracle India Pvt. Ltd., Gujarat.

- Contributed to implement new features in the Absence module of HCM cloud application using Oracle Application Development Framework (MVC), Java and SQL. Used various concepts related to Java and algorithms for the development of features.

- Analyzed and resolved various recurring issues causing data corruption. Worked on a self-service module using Java and ADF to facilitate users in correcting data corruptions from UI. PL/SQL procedures were used to maintain data integrity within database tables.

M.Tech. Research Project

Prof. Shirish Shevade.
Personalized News Recommendation System.

- Re-implemented the famous Neural Collaborative Filtering paper(https://arxiv.org/abs/1708.05031) on curated Microsoft news dataset and evaluated our recommendations based on hit ratio and normalized discounted cumulative gain(NDCG).
- Modified NCF by adding textual information, available per news, in the embedding process of both user and item. Different models like auto-encoders, Google's universal sentence encoder were tried to represent the news - text as a vector. In addition to embedding process, several techniques were implemented for modeling the interaction function as well.
- Challenges faced included setting thresholds for sparse data to make it computationally possible to run models and in addition to news-item ids making use of news-text as well to enhance their representations.
- Our "text inclusion" model beat state of the art NCF results on our dataset by 3% - 4%.

Skills

Programming: Python, Pytorch, SQL, Java, Go(beginner)
Frameworks: scikit-Learn, spacy, BigQuery, FastAPI, Benthos, PyTextRank, Hugging Face Transformers, Docker, redis, polars, dask, Kibana(Elasticsearch)
experimentation/monitoring: OpenLens (for Kubernetes pod), prometheus + Grafana dashbaord, EPPO(for A/B testing) and superset(for online monitoring)
Version control: Git, Airflow, Jenkins