A Minor Project Synopsis on

**Fuzzy Object Look-Up Using Hash-Maps and kD trees.**

Submitted to Manipal University, Jaipur

Towards the partial fulfilment for the Award of the Degree of

**BACHELORS OF TECHNOLOGY**

In Computers Science and Engineering

2020-2021

By

Tarun Krishnan

Siddhant Kate



Under the guidance of

Ghanshyam Raghuwanshi

**Department of Computer Science and Engineering**

**School of Computing and Information Technology**

**Manipal University Jaipur**

**Jaipur, Rajasthan**

Introduction

In the ever evolving age of technology and software, there is always a need for optimisation and increasing efficiency at scale. While there already has been quite a bit of advancement made towards such endeavours, it will always make sense to add to this effort.

As such, we aim to research the benefits of existing systems for kD trees and hash maps, to further improve the efficacy of search, indexing and querying methods with respect to vector space traversal, fuzzy matching and approximate nearest-neighbour querying.

**Motivation**

K-nearest neighbours (KNN) is the simplest and most easily optimizable supervised algorithm in a regression algorithm. The ‘k’ in KNN is a hyperparameter and must be manually tuned for optimal results from dataset to dataset. Recently, for larger datasets, KNN was replaced with ANN (approximate nearest neighbours) for faster query times, at the cost of precision. KNNs and ANNs can be used and implemented in full text search engines with word2vec, doc2vec or similar vectorizers to improve result relevance at very little compute cost if implemented correctly.

Recent ANN methods such as ScaNN (Google) and Annoy (Spotify) aim to do exactly this. Our goal is to improve full text document search and matching using a novel data structure to store and query a text document dataset, and achieve high recall and precision scores for a wide range of queries.

**Project Objective**

We plan on researching the best ways to optimize the current algorithms that exist for KNN problems, as well as for any related systems that we come across. After we have done adequate research, we will work towards trying to better optimize said algorithms, and then after collecting data, publish our findings in a research paper.

**Methodology/ Planning of work:**

We aim to complete our research work as efficiently as possible, with the following phases that will cover specific areas of research and testing:

**Phase 1:**

Collecting existing material on the subject, and deciding what to use and what to discard. Our current reading list includes :

1. Current KNN, ANN methods

2. Fuzzy matching for text

a) tokenization - CBoW, BoW, skip-gram, bert, vector spaces

b) fuzzy matching at scale, tfidf, batch querying

3. askd trees and large cache/high latency avoidance for kD trees

5. Tree structuring choices

**Phase 2:**

Having finalized our starting material, we form a basic idea and work towards eliminating lossy, unoptimised methods and solving any arising problems. We will also find a viable dataset to work with and test the same against current ANN and full text search methods.

**Phase 3:**

Establish a working model, and further proof test it, as well as debug any issues in it.

**Phase 4:**

Test the current model against the same dataset defined in phase 2, until it no longer poses any issues in deployment.

**Phase 5:**

Record and publish findings in a paper.

**Bibliography/References:**

<https://www.researchgate.net/publication/340475371_A_new_hashing_based_nearest_neighbors_selection_technique_for_big_datasets>

<https://dl.acm.org/doi/abs/10.5555/569046.569059>

<https://www.cvl.isy.liu.se/en/education/graduate/VOR14/Lecture6.pdf>

<https://ai.googleblog.com/2020/07/announcing-scann-efficient-vector.html>

<https://github.com/spotify/annoy>

<http://proceedings.mlr.press/v32/gieseke14.pdf>

<https://ivpl.northwestern.edu/wp-content/uploads/2019/02/2013-ICASSP2.pdf>

**Signature of Guide**