Toy Search Engine Design Document

# Overview:

The problem we are tying to tackle is making a toy search engine (TSE) for a restricted domain. This TSE will have a simple page ranking system based on the key words in the word(s) or phrase entered in the search box. This ranking system will rank pages based on criteria such as: the number of words contained in a particular page that match the current query, the proximity of words on the page and the in-degree (number of pages pointing to the page at hand).

In addition, if we do not find an exact match for a keyword/ phrase: we will then look into the synonyms of the word/phrase and then run the search, ranking these results comparatively lower than the results obtained using exact matches.

These operations require a database as these cannot fit in memory.

In later phases, we will implement some extra features such as a “trending” feature that provides the user with auto-completion for the queries he enters in the search box based on the trending/ most popular searches. This will require modification of the current Databases to include a store for a timestamp, a record of the search as well as a counter. Another feature would be an open chat that enables all users of the engine to chat in real time. This feature will require a separate database.

# Design Overview:

For the primary phase, we make use of two databases (DBs): WordFromIndexedPage and IndexedPage. These are used to store information about the pages visited during the search process. These DBs are used by the indexer and crawler when a search is launched and further operations are conducted to rank pages as required.

# Design Details:

The fields of the DBs are as follows:

1. Word from Indexed Page:

*Requires:* word, indexed page,

*Optional/ Generated:* \_unique\_ID (generated upon construction/saving), offsets\_in\_indexedPage

1. Indexed Page:

*Requires:* URL

*Optional/ Generated:* raw\_html, text\_content, raw\_html\_hash, original\_page,

Indegree (a metric)

The indexer, given a URL populates the fields of the IndexedPage DB. It is the job of the crawler(s) to assign indexers to pages they visit so that data about these pages can be collected.