**ITCS 6112**

**Software System Design and Implementation**

**Smart data summarization and contextualization**

# **Abstract or Problem Description**

With advent of big data we have huge amount of information which lies unprocessed and unsearchable which is increasingly becoming difficult to access due to inherent aspects of Big Data which is volume, variety, veracity and velocity. However, in reality, a lot of this data comprises ‘noise’ or irrelevant information which lies around which is not harnessed in anyway.

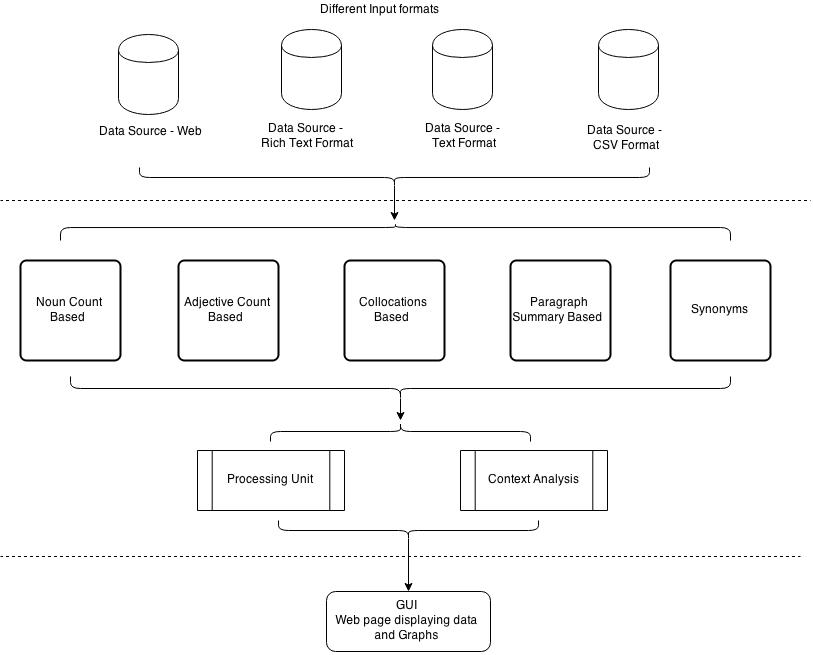
The purpose of smart data (veracity and value) is to filter out the noise and hold the valuable data, which can be effectively used by the enterprise to solve business problems. Our proposal is to create Smart data from Big Data which actually converts unrelated, non-synthesized data into connected, searchable, Index able snapshot information about context and contents of the Big Data source.

# **Introduction: Explain the abstract in detail including what has been achieved**

Over the last three months we have designed, implemented and tested project for converting any data to smart data. For Smart data conversion there are no formulas, we need to understand the clues in the questions around the data which we are provided. Analyzing data qualitatively enables one to creatively-driven rather than mundane data-driven. And this is where big data becomes smart data. Smart data conversion is important for segmenting, contextualizing Big Data sources which are not inherently searchable. Hence those big data sources ae not indexed by search engines. When data is mined for nouns, collocations, synonyms, adjectives, contextualized it is easy to search, relate and retrieve results.

Our project has achieved several goals: smart data conversion process, wide applicability, scalability, performance. Algorithm is scalable to be used in Hadoop or any related platforms. We have ran project on various input files ranging from ancient English literature to real time twitter feeds. We were successful in parsing those varying data and do data mining and produce meaningful smart data. This is along with synonyms and context output.

# **Methods used and Technical documentation (see below)**



Architecture Diagram

For creating smart data we have learned what all components make the smart data. We have identified 3 aspects to make any data to smart data. These are following:

1. **Adding meaning**

The first step is to add meaning to your data by richly describing both the entities within your data and the relationships between them. Equally important to describing the meaning of data is *where* the meaning is described. Dumb data often has its meaning recorded via data dictionaries, service interface documents, relational database catalogs, or other out-of-band mechanisms. To make data smarter, don’t rely on the meaning of the data to be hardcoded within software; instead, link the meaning of the data directly to the data itself.

There are several ways to describe data’s meaning, and the richer the description you choose, the smarter your data becomes. Data’s meaning can include:

Controlled vocabularies that describe the acceptable values of an attribute

Taxonomies that capture hierarchical relationships within the data

Schemas that communicate data types, cardinalities, and min/max value ranges

Ontologies that declaratively represent rich semantics of data such as transitive relationships, type intersections, or context-sensitive value restrictions

There are two benefits of adding meaning to your data. First, software can respond appropriately (for example, by performing data validation or automatically choosing the right UI widget) to the meaning of different data sets without having to be customized for each. Second, rich data descriptions attached to data can empower business experts to manipulate data themselves without relying on scarce IT or data science personnel for every new dashboard, visualization, or analysis.

For synonym identification we use a large lexical database of English called Wordnet.

WordNet is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets). Our top frequency words are passed to synsets which returns the list of synonyms which are appended to the result.

**Step 2: Add context**

The Sisyphean pursuit of a generally unreachable “single version of the truth” belies the importance of context in making data smart enough to be discovered and understood by business decision makers. The lack of context makes data unreliable and hard to trust and decreases the chance that decision makers will rely on it.

Just as meaning is traditionally captured separately from data, so, too, is contextual metadata usually divorced from the data it describes. To make data smarter, you must treat metadata as data. This means directly capturing and maintaining simple metadata such as the author or creation time of a piece of data, but it also means linking data to its full lineage, including the source of the data (e.g., a particular enterprise database, document, or social media posting) and any transformations on the data. Context can also include probability, confidence, and reliability metadata. Finally, data’s context might involve domain-specific attributes that limit the scope in which a particular piece of data is true, such as the time period for which a company had a particular ticker symbol or the position a patient was in when a blood pressure reading was obtained.

By representing contextual metadata alongside the data itself, users can query, search, and visualize both at once. There’s no need to create separate, time-consuming data-load processes that select data from a particular time period or author. There’s no need to login to separate applications to verify the trustworthiness of data within a business intelligence dashboard.

In the project we have made first step move towards contextualizing we have attempted to achieve polarity check by creating list of positive and negative words. We found Cornell professor Bo Pang’s page on movie review data and selected his sentence polarity dataset v1.0, which has 10,663 sentences from movie reviews, each classified as either positive or negative. Samples from these files were added and used for finding context.

The input file is analyzed and text mined for adjectives which uses Wordnet and then it is counted to identify which are the top adjectives and once those are identified are added to the list of contextualized list of arrays. These arrays are analyzed commonly by data scientist and categorized. After that it is run on a classification algorithm

For contextualizing we identified key and top adjectives used in the file and those words are used for finding the mood and context of the sentence. This is done by identifying which word belong in what group and once the data mined context is selected as group of contexts. And the result is appended along with other elements.

A data scientist is needed to understand the context and categorize it.

**Step 3: Add graphs**

One of the hallmarks of the burgeoning NoSQL ecosystem is its move away from the rigid tabular data structures of the relational database world. Perhaps the most flexible and expressive of the NoSQL data models is the graph. A graph is a web-like network model; it represents data as a collection of entities linked together via explicit relationships. Graphs have no up front constraining structure, though their structure can be described after the fact. This means that graphs are fundamentally flexible: new attributes or new relationships simply add new edges and nodes to the graph without requiring the rework that a more rigid model (such as the relational model) might require.

Furthermore, graphs act as a lingua franca -- a least common denominator -- for representing information from other data models. For example, relational data easily fits into a graph model (by representing tables as nodes and both columns and foreign keys as edges within the graph). So, too, does data from unstructured text fit into a graph -- as text analytics processes uncover new insights within text, those insights grow the graph by adding new edges and nodes as necessary, even though we might not have known up front what we’d discover in the text. Finally, graphs are ideal for linking data together with its description and metadata -- the meaning and context that we’ve already established as key parts of embracing smart data.

**4. Analyze top nouns**

We get input from user, optionally collects and analyze the top nouns from the input file. This is based on the NLTK library and top numbered ones based on the frequency are found out. This gives an insight into key actors in the data set. The top noun list is added to the output and also saved in file.

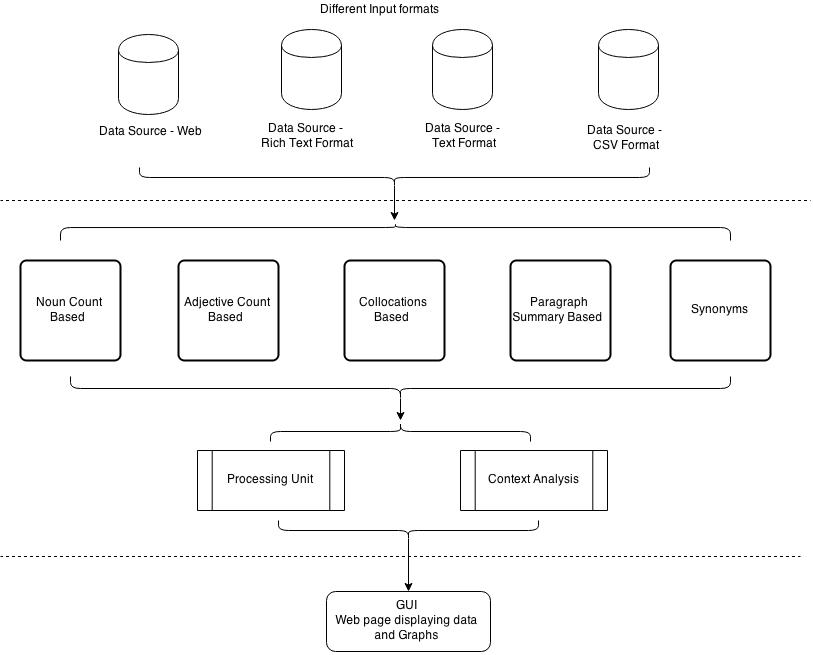
**5. Derive top collocations**

We get input from user, optionally collects and analyze the top collocations from the input file. This is based on the NLTK package and top numbered ones based on the frequency are found out. This gives an insight into key word groups in the data set.

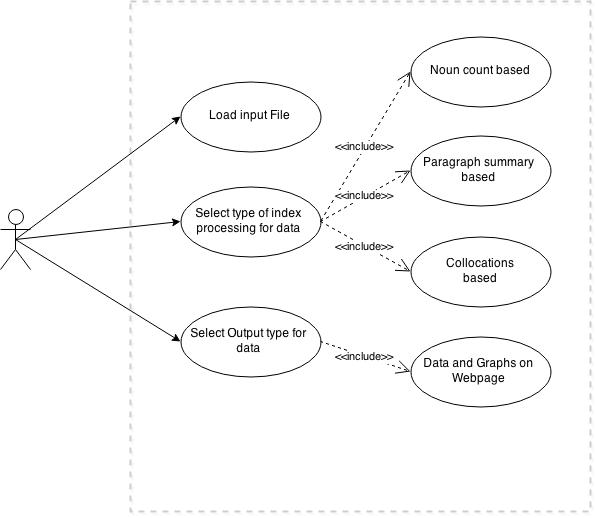
**6. Give snapshot view – Paragraph Summary**

For publishing the snapshot of the data source we use reduction algorithm which does summarization of text by extracting important sentences. We have given configurability to the user where the amount of reduction can be configured.

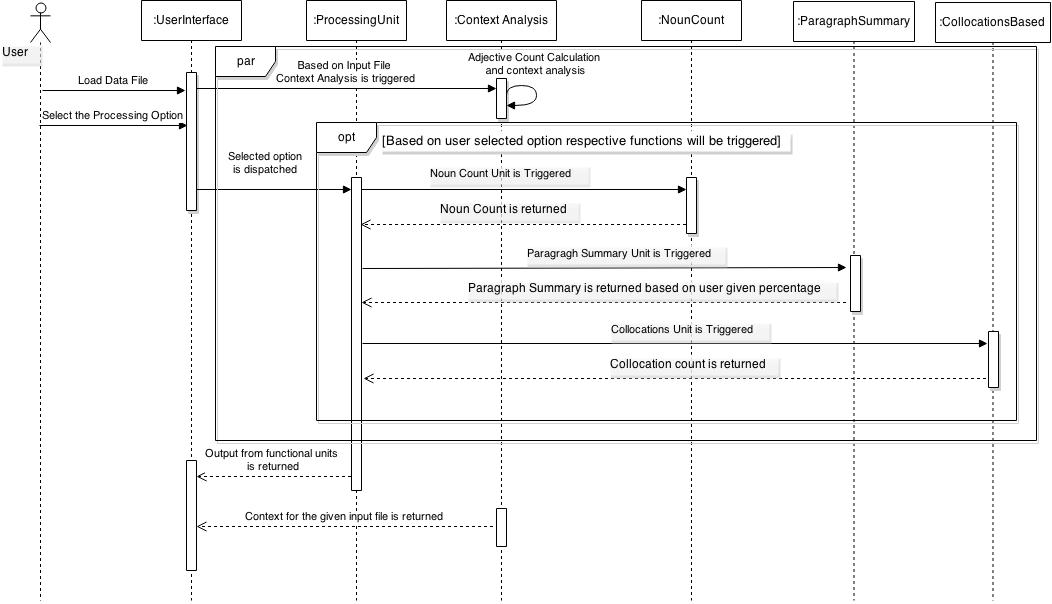
# **Sample runs with proper output and or diagrams**



Software Architecture Diagram



Use Case Diagram



Sequence Diagram

# **Results**

We have successfully ran the project on various files which are distinctly different in subjects and concluded the efficacy of the project. For sample run we used relevant data of top feeds from twitter and analyzed it. We were able to successfully create smart data and display results to the users based on the selection.

We selected currently top discussed topic “NetNeutrality”, “ObamaTownHall”, “TaxDay”, “MyChoice” and did analysis on it and was successfully able to categorize it and display context, its overall sentiment and other snapshot view of discussion in high level view.

# **Conclusions**

We have described Smart Data Summarization and contextualization, an approach to make any data source into Smart data which can be searchable. The various versions of summary we provided helps user to categorize, contextualize and understand overall view of the data present in the data source. The key feature of it is the variety of input file it can handle and produce result.