Automatic keyword extraction from a real estate classifieds data set

استخراج الكلمات الرئيسية التلقائي من مجموعة بيانات المنتمين إلى
العقارات الإعلانات المبوبة

By

Dibin Devassy

Dissertation submitted in partial fulfillment of

MSc in Information Technology Management

Faculty of Engineering & IT

Dissertation Supervisor

Dr. Sherief Abdallah

November-2011

Abstract

In this age where information and internet technologies are developing at a radical pace, users have the access to large amount of documents and information online. With the increasing amount of information, it also becomes an area of interest on how to make online search easier. Keywords are considered to be a solution to this problem and are now widely used to search the information over the internet.

In this project we analyze a real estate classifieds data set, with an objective to find keywords that represent this data set. We begin with designing data cleansing algorithms to verify different attributes of the real estate classified. Further, we progress to extract the candidate keywords from the cleansed data set. Finally, we develop a method to automatically extract the keywords and also the key phrases that are formed along with the keywords.

خلاصة

في هذا العصر من سرعة تطوير تكنولوجيات المعلومات، يكون لدى المستخدمين الحصول على كمية كبيرة من الوثائق والمعلومات على شبكة الإنترنت. ومع ازدياد كمية المعلومات، فإنه يصبح مجالاً للاهتمام بشأن كيفية جعل الإنترنت البحث أسهل. الكلمات الأساسية تعتبر حلاً لهذه المشكلة، والآن تستخدم على نطاق واسع للبحث عن المعلومات على شبكة الإنترنت.

في هذا المشروع، نحن نحلل مجموعة بيانات إعلانات العقارية بهدف انتزاع الكلمات الرئيسية. نبدأ بتصميم البيانات التطهير خوارزميات للتحقق من سمات مختلفة من الإعلانات المبوبة العقارات. علاوة على ذلك، نحصل على مجموعة محتملة من الكلمات الرئيسية من مجموعة البيانات المطهرة. وأخيراً، علينا أن نطور طريقة لاستخراج تلقائياً الكلمات الرئيسية والعبارات الرئيسية التي تتشكل جنبا إلى جنب مع الكلمات الرئيسية.

Acknowledgement

I hereby thank my supervisor, Dr. Sherief Abdallah, for providing constant support throughout the course of this dissertation.

I also thank my family and friends for their patience and continuous encouragement.

Declaration

I declare that this dissertation is my own work and is completed in my own words. Any work from other sources is duly acknowledged, for which a list of references is included at the end.

Dibin Devassy

Contents

List of figures

List of Tables

Chapter 1: Introduction

With ever increasing amount of data, for a market or an industry to have a strong understanding of their data accumulated from day to day business activities holds key significance. Knowing their data also adds value to the strategic advantage for any business, thereby helping them to understand the market forces, increase profits and also serve their customers more intuitively. This need of gathering valuable information from data has added significance to the discipline recognized as “Data Mining”.

In this project, the focus is to analyze a data set obtained from online real estate classifieds, in order to find the keywords used to describe them. Finally, a method is developed to automatically extract the keywords and key phrases from this data set.

1.1 Objectives, Key Questions and Scope

Objectives:

- Refine the data set obtained on real estate classifieds.
- Analyze the refined data set to derive the keywords.
- Design a model that can automatically extract the keywords and key phrases based on the attribute ‘rent’ in real estate classifieds.

Key Questions:

- Can we discover keywords automatically from real estate classifieds and if so, what methods can be used?
- What are the prominent keywords used in online real estate classifieds in Dubai?

Scope:

The scope of this project is limited to real estate classifieds listed online in Dubai during the period 07-feb-2011 to 26-Apr-2011.

1.2 Proposed solution

The keyword extraction from the obtained data set will be carried out in two phases:

1. Data Cleansing

   In this phase we focus on cleansing the attributes related to a real estate classified. Here we will check the accuracy of information for each attribute of beds, location and rent in the real estate classified. This will be followed by correcting the incorrect values and find missing values of attributes, for the required records in dataset.
2. Keyword extraction

   The cleansed data set will be used to identify the candidate keywords based on the occurrence of the words in real estate classified dataset. Once the candidate keywords are obtained, we will formulate methods to calculate the word and phrase weights to extract the keywords from the data set.

1.3 Report Outline

The report is organized into seven chapters. Subsequent chapters are as follows:

Chapter2 establishes a background by giving an overview about data mining, data cleansing, interactive data mining and keyword extraction.

Chapter 3 elaborates the steps of data cleansing algorithms for each attribute of the dataset.

Chapter 4 identifies the candidate keywords and develops method to automatically extract keywords and key phrases.

Chapter 5 specifies the database and software tool used for implementation.

Chapter 6 illustrates the results of data cleansing and keyword / key phrases extraction.

Chapter 7 provides a recommendation, synthesizes the work done and concludes with a direction for future work.

Chapter 2: Background

Data mining can be defined as the analysis of the collected data sets, to discover relationships which aren't normally inferred and to sum up the data in simple ways so that it is both easy to comprehend and valuable to data owner (Hand, Mannila & Smith, 2001, p. 1). This is just one way of defining data mining and researchers have defined it in various ways.

As a scientific technique, data mining is a part of the larger process known as knowledge discovery in databases (KDD) (Fayyad, Piatetsky-Shapiro & Smyth, 1996). As defined by Fayyad, Piatetsky-Shapiro and Smyth (1996, p. 3), KDD "is the non trivial process of identifying valid, novel, potentially useful and ultimately understandable patterns in data." According to them KDD process, as depicted in figure 2.1, can be illustrated using following steps:

1. Understand the application domain and realize how KDD process can add value.
2. Choose the dataset on which the knowledge discovery is required to be carried out.
3. Preprocess the data to eliminate noise and inconsistency. This step involves formulating a data cleaning approach that can be used to fix issues like missing and incoherent data values.
4. Finding attributes which are relevant to the data and are required to accomplish the objective.
5. Choose an appropriate data mining method with respect to the task.
6. Select an algorithm for the purpose of extracting patterns from the data.
7. Look out for interesting patterns using data mining methods.
8. Evaluate the patterns obtained.
9. Document the discovered knowledge and communicate it to the data owners.

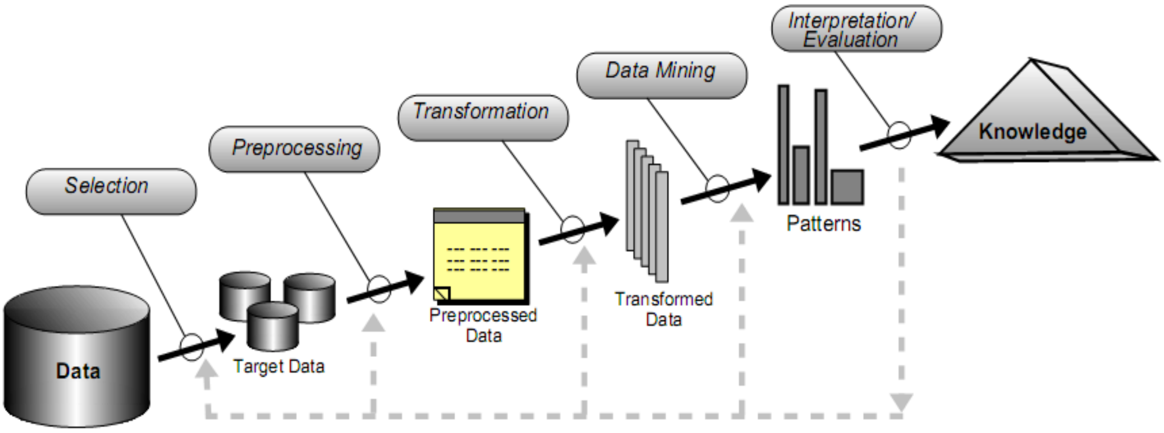


Figure 2.1: Overview of KDD process (Fayyad, Piatetsky-Shapiro and Smyth, 1996, p. 3)

2.1 Data Cleansing

Errors are an integral part of data entry (Maletic and Marcus, 2005, p. 21) and collecting error free data is a major challenge for the researcher. Hence, data cleansing plays an important role in dealing with the erroneous data. Data cleansing has variety of definitions. Chapman (2005, p. 1) state that data cleansing means:

1. Identify "inaccurate, incomplete or unreasonable data"
2. Enhance the data quality by rectifying "detected errors and omissions"

The cleansing of data can be considered to have three stages (Maletic and Marcus, 2000, p. 3):

1. Identify the error types
2. Locate the error instances
3. Fix the identified errors

As these steps for data cleansing are stated generically, it can be incorporated into a data cleansing algorithm intended to be used on any domain. As first step of the process, the error types are to be identified. This might be a point of concern in evolving databases because the error types identified by studying sample datasets may not represent all the kinds of error types that may be encountered in future. Therefore, as a proactive measure, the error types are required to be revisited and updated frequently.

Data cleansing is incomplete if they are not documented and documentation of errors are necessary to establish the quality of data being used (Chapman, 2005, p.55). Further, Chapman (2005, p.55) suggests that documenting the errors helps in avoiding the overwork of same data being repeatedly checked for the same type of error. Keeping a log of data errors can also provide information on what kind of errors are dealt for the data and also assist in identifying new errors.

2.2 Interactive Data Mining

Most of the data mining tasks give prime importance on “automation and efficiency”, whereas in “interactive data mining” the focus is on having “adaptive and effective communications between human users and computer systems” (Zhao and Yao, 2005). They also point out that human role has not been given enough emphasis in data mining.

According to Zhao and Yao (2005, n.d.), the processes involved in interactive data mining is the same as in knowledge discovery process (figure 2.1), but with an added factor of continuous user interaction. They classify the forms of interactions into four (figure 2.2):

1. “Information acquisition” - It is the phase where user interactively extracts the information from the system.

2. “Navigation” - involves movement from one level of information to another in the information hierarchy.

3. “Manipulation”- involves applying different approaches for E.g. in form of algorithms to solve a data mining problem.

4. “Evaluation and explanation”- It is the phase where user can decide on whether to continue looking for patterns, based on the results obtained from data mining.

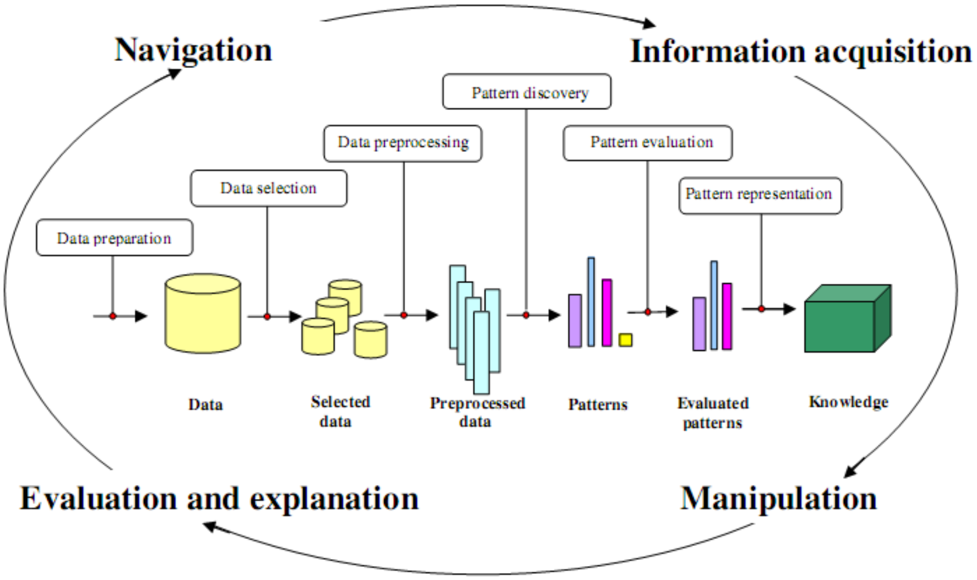


Figure 2.2: Interactive Data mining (Zhao and Yao, 2005, n.d.)

Interactive data mining also includes an approach known as “retrieval by content” (Hand, Mannila & Smith, 2001, p. 450). In this approach, a set of query patterns (for e.g. a set of words) are identified by the user and then they are searched in a database to find similar patterns.

In this project the essence of ‘interactive data mining’ and ‘retrieval by content’ will be applied for verifying the attributes in the given data set. This includes:

1. Manually identifying the patterns on how an attribute is described in the dataset.
2. Finding similar patterns for the attribute.
3. Verifying the actual values for the attributes given in the data set.

2.3 Keyword Extraction

Keyword extraction is a process of extracting set of words or phrases from a document that expresses the significance of the document (Hulth, 2003, p. 1).

Breaking up an array of characters into individual meaningful words or tokens will be a simple task if the words are uniformly separated either by a white-space or by a punctuation mark (Kaplan, 2005, p. 55). But in reality the case is different as the data extracted from any system might have irrelevant characters or any other domain specific issue.

Domain specific Approach:

Hou and Chan(2003), suggested a model that can be used in any domain to extract the keywords specific to that domain. The method considered analyzing the keyword correlation based on frequency and location of the keyword in a document. The aim of this research was to reduce the dependency on the domain based thesaurus or knowledge experts for extracting the keywords from a document.

Thesaurus based Approach:

An experiment done by Hulth(2003) tried to include linguistics into the process of keyword extraction. The terms were considered to be keywords based on document frequency, collection frequency, the location of the first occurrence of the term relative to the document and the significance of the term as part speech. The outcome of this experiment showed that considering linguistic aspect enhanced the automatic keyword extraction.

Statistics approach:

In statistics based approaches a common practice for finding the weight of a term in a document is by using term frequency (TF) and inverse document frequency (IDF) (Hand, Mannila & Smith, 2001, p. 463). Here term frequency (TF) depends on the frequency on the term in a document, whereas IDF helps in reducing the discrimination of a term if it occurs frequently in large number of documents.

In some cases position of the keyword can also be considered for extracting keywords, where the term in the title of a document is considered significant than the term in description (Viji, 2002).

Statistics approaches are also used in “automatic text summarization” where the weight of a word in a document is assigned based on how frequently they appear in that document. (McCargar, 2004, p. 22)

From the different approaches discussed here on keyword extraction, it seems that there is no universal approach to extract keywords. The decision to use an approach might be influenced by the kind of data and domain. Apparently, successful extraction of keywords may depend combining the various approaches to suite the data being experimented.

Chapter 3: Data Cleansing

The purpose of data cleansing is to check the accuracy of the information given against the attributes beds, location and rent of the real estate classified. The approach used here is to extract the information related to the attributes from the title and description of the classified and check if they match with actual values for the same attributes given in dataset.

The dataset used here is belongs to online real estate classifieds posted in Dubai. The source dataset file was extracted from MySQL database. Since this project is using Oracle database as the platform for implementation, as the first step, the MySQL statements are converted into Oracle SQL statements. As the data file was large enough to manually do this conversion, a small program is written which reads the MySQL data file and converts MySQL statements into Oracle SQL on the fly.

3.1 Tables

The dataset is divided in to 2 main tables namely, ADCORE and ADDETAIL.

ADCORE

This is the master table and stores the attributes TYPE, BEDS and RENT of a real estate classified.

Column	Description
ID	Unique identifier of a record
TYPE	Denotes if the property being advertised is an apartment or villa and whether it is for sale or for rent
BEDS	Denotes the number of bedrooms in the property
RENT	Denotes the rental price or sale price of the property

Table 3.1: Structure of ADCORE

ADDETAIL

This is the child table of master table ADCORE and stores the attribute LOCATION related to the real estate property advertisement. This table also includes the TITLE and DESCRIPTION given for a classified.

Column	Description
ID	Unique identifier of a record which connects to the master table ADCORE
LOCATION	Denotes the actual location of property being advertised.
TITLE	Denotes the title of the advertisement
DESCRIPTION	Denotes the description of the advertisement

Table 3.2: Structure of ADDETAIL

One more table namely, AD_CORE_DETAIL_LOG, is created for documenting the data cleansing part. This is a log table that stores the information on verification of the attributes given in ADCORE and ADDETAIL tables.

Column	Description
ID	Same ID as in ADCORE/ADDETAIL
LOCATION_MATCH	stores 'Y' if LOCATION matching is successful
LOCATION_CORRECTED	stores 'Y' if LOCATION information is updated for records having missing location in ADDETAIL table
LOC_MISMATCH_CORRECTED	stores 'Y' if LOCATION information is corrected in ADDETAIL table
BED_MATCH	stores 'Y' if BEDS (table: ADCORE) matching is successful
BED_CORRECTED	stores 'Y' if BEDS (table: ADCORE) information is corrected
BED_KEYWORD_MATCHED	Stores the keyword used for BEDS (table: ADCORE) matching
RENT_MATCH	stores 'Y' if RENT (table: ADCORE) matching is successful
RENT_CORRECTED	stores 'Y' if RENT (table: ADCORE) information is corrected
RENT_KEYWORD_MATCHED	Stores the keyword used for RENT matching

Table 3.3: Structure of AD_CORE_DETAIL_LOG

3.2 General flow

A common sampling method was used for cleansing all the attributes. To start with, a simple random sample of given dataset is taken, with 100 records (from ADDETAIL table). To store and work on this sample, a temporary table ADDETAIL_SAMPLE (same structure as table ADDETAIL) is created. In this temporary table, we manually search in columns TITLE and DESCRIPTION to identify patterns in which an attribute is described. Once a pattern is found for an attribute, it is added into a trainer set. Further the algorithms detailed in section 3.2.3 are executed to find records with same patterns and also to check if the value of the attribute extracted from the pattern matches with the actual value of the same attribute given in dataset. The pattern matching for all records is done using TITLE and DESCRIPTION columns of ADDETAIL table.

Also, when a match is found for an attribute using the pattern, it is logged into the log table AD_CORE_DETAIL_LOG for that particular record. As the next step, records for which patterns are identified are removed from the sample table and the process said above is repeated for the next identified pattern.

Once sample table is empty or the records in it do not give any useful information for the attribute being searched, next set of 100 records is populated into the sample table from the dataset (ADDETAIL table), for which an entry has not been made in the log table corresponding to the attribute.

This process of taking sample records and identifying the patterns is done for one attribute at a time. This means, for instance, we check for attribute ‘location’ and repeat the process of identifying patterns and populating the sample table with random set of records, until we find all possible patterns for this attribute. When no more patterns are found for attribute ‘location’, then we consider the next attribute and the sample table is repopulated with new sample records and the process is repeated for the new attribute.

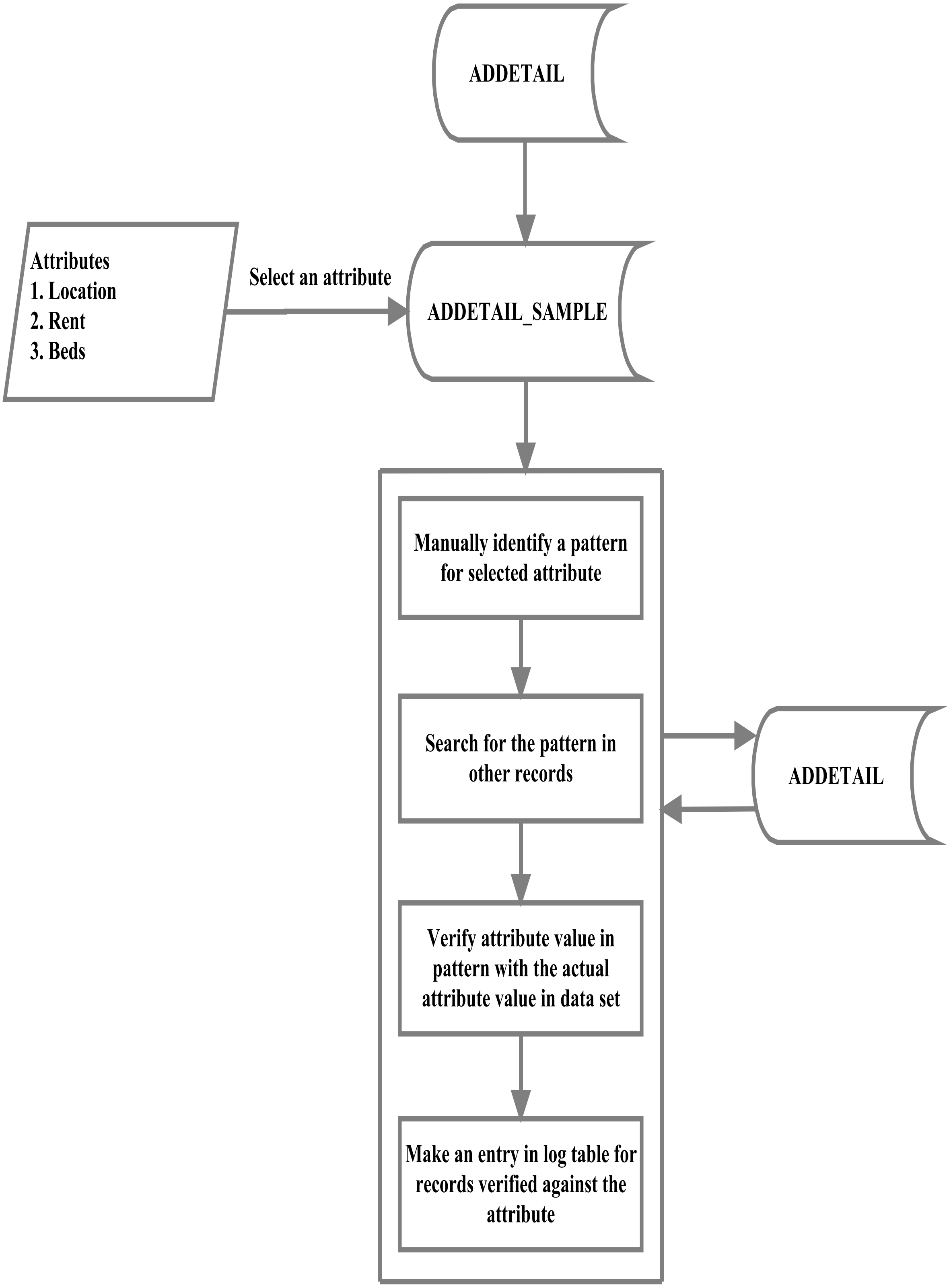

ADDETAIL
Attributes
1. Location
2. Rent
3. Beds
Select an attribute
ADDETAIL_SAMPLE
Manually identify a pattern for selected attribute
Search for the pattern in other records
ADDETAIL
Verify attribute value in pattern with the actual attribute value in data set
Make an entry in log table for records verified against the attribute

Figure 3.1: General flow of data cleansing

3.3 Algorithms

The algorithms used for data cleansing are categorized as per the attributes – location, beds and rent.

Attribute: Location

Set up:

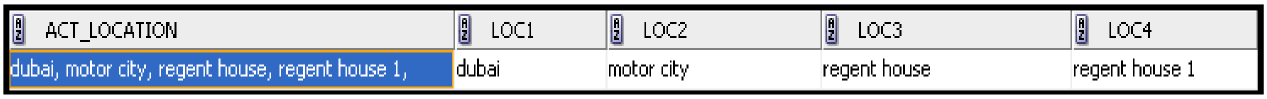
Location information (table: ADDETAIL, Column: Location) in the dataset is given as comma-separated word segments:

‘city, area, sub area 1, sub area 2’

E.g. “dubai, arabian ranches, palmera, palmera 3”.

Considering the above format, there are 1328 distinct locations in dataset. Although above format is considered as generic, but not all of the locations in dataset had details as per this format. A reason might be that for an online property classified, it may not be necessary to have all the information of ‘city’, ‘area’, ‘sub area 1’ and ‘sub area 2’.

Further, a set up table namely ‘ADLOCATION’ is created to store all the distinct locations in dataset. ADLOCATION table contains the individual location segments that are separated by comma for a given location information. This means that if we have a location given as ‘dubai, motor city, regent house, regent house 1’, the individual location segments separated by comma(,) will be extracted and stored in separate columns of ADLOCATION table as shown below:

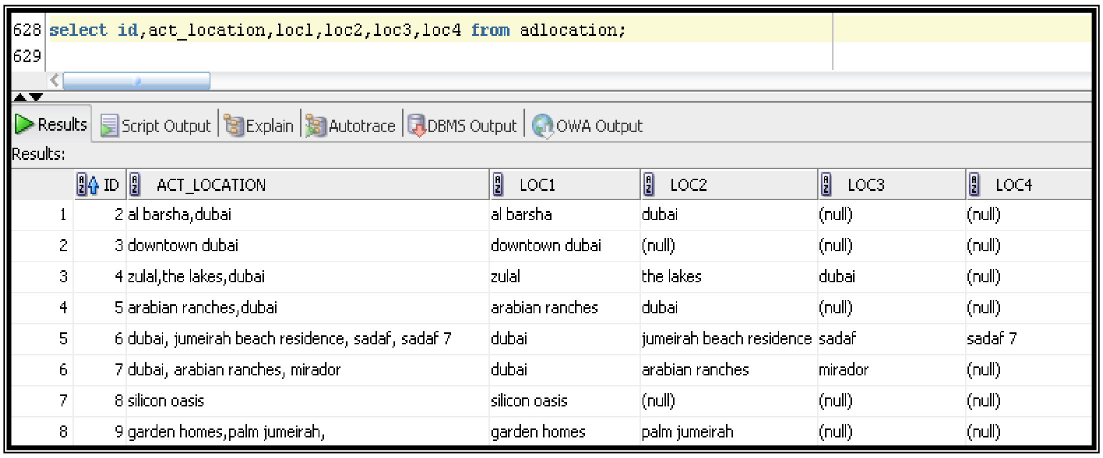


ACT_LOCATION	LOC1	LOC2	LOC3	LOC4
dubai, motor city, regent house, regent house 1,	dubai	motor city	regent house	regent house 1

There are two reasons why the separated location words were stored in table:

a) In the method for matching ‘location’ data, individual location segments that are separated by comma are searched in ‘TITLE’ or ‘DESCRIPTION’ columns in ‘ADDETAIL’ table. Having the individual word segments stored in ADLOCATION table will help is eliminating the overhead of extracting the individual location segments each time during the process of matching.

b) For the classifieds with missing location, the location could be identified by finding the best match from table ‘ADLOCATION’.



```
select id,act_location,loc1,loc2,loc3,loc4 from adlocation;
```

Results:

	ID	ACT_LOCATION	LOC1	LOC2	LOC3	LOC4
1	2	al barsha,dubai	al barsha	dubai	(null)	(null)
2	3	downtown dubai	downtown dubai	(null)	(null)	(null)
3	4	zulal,the lakes,dubai	zulal	the lakes	dubai	(null)
4	5	arabian ranches,dubai	arabian ranches	dubai	(null)	(null)
5	6	dubai, jumeirah beach residence, sadaf, sadaf 7	dubai	jumeirah beach residence	sadaf	sadaf 7
6	7	dubai, arabian ranches, mirador	dubai	arabian ranches	mirador	(null)
7	8	silicon oasis	silicon oasis	(null)	(null)	(null)
8	9	garden homes,palm jumeirah,	garden homes	palm jumeirah	(null)	(null)

Figure 3.2: A snapshot of ADLOCATION table

Assumption:

1. In some cases location names are given as acronyms or short forms and also some classifieds had partial names of the location. For such cases, the matching on location data is done as shown in table 3.4:

Acronym / short form		Location Full name
jlt	**Is equal to**	jumeirah lake towers
jbr	**Is equal to**	jumeirah beach residence
downtown	**Is equal to**	downtown burj dubai
difc	**Is equal to**	dubai international financial centre
marina	**Is equal to**	dubai marina
silicon oasis	**Is equal to**	dubai silicon oasis
sports city	**Is equal to**	dubai sports city
jvc	**Is equal to**	jumeirah village circle
jvt	**Is equal to**	jumeirah village triangle
downtown burj khalifa	**Is equal to**	downtown burj dubai
Al (name) E.g. Al barsha	**Is equal to**	(name) E.g. barsha

Table 3.4: Acronym matching for attribute ‘location’

2. The location segment ‘Dubai’ is avoided from comparison on location since the whole dataset belongs to Dubai real estate property advertisements.

Algorithm Steps:

Data cleansing for attribute ‘location’ is divided into three parts.

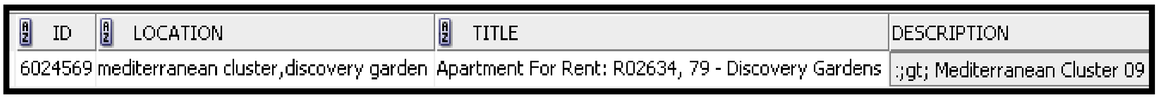
PART 1:

The first step was to find the perfect match for attribute ‘location’.

The steps followed are:

1. Get the actual location value from dataset (table: ADDETAIL, column: LOCATION)

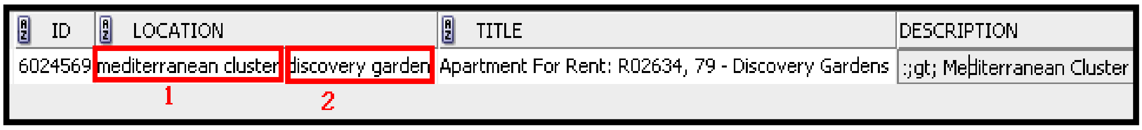
   E.g.



ID	LOCATION	TITLE	DESCRIPTION
6024569	mediterranean cluster,discovery garden	Apartment For Rent: R02634, 79 - Discovery Gardens	:;gt; Mediterranean Cluster 09

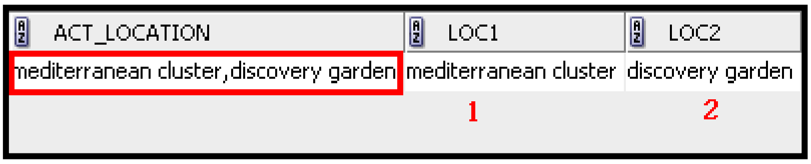
2. Get the individual location segments for the location value obtained in step 1.

   E.g. we have two location segments here.



ID	LOCATION	TITLE	DESCRIPTION
6024569	mediterranean cluster discovery garden	Apartment For Rent: R02634, 79 - Discovery Gardens	:;gt; Mediterranean Cluster

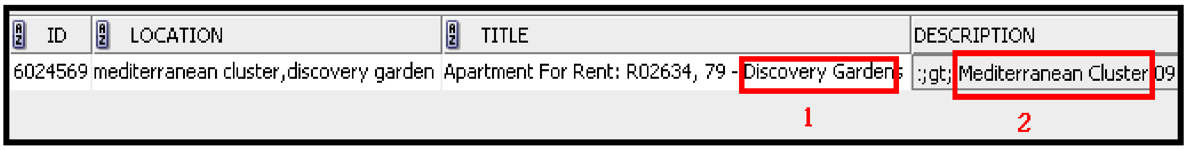
Note: Here the individual location segments are actually obtained from table ADLOCATION for the particular location.



ACT_LOCATION	LOC1	LOC2
mediterranean cluster,discovery garden	mediterranean cluster	discovery garden

3. Check if all the location segments found in step 2 are present either in TITLE or DESCRIPTION columns of the same record. If all word segments found then the particular record is considered as perfectly matched for location attribute.

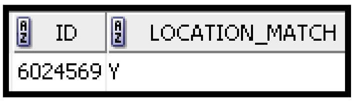
   E.g. the location segments found in step 2 is present in columns TITLE and DESCRIPTION combined.



ID	LOCATION	TITLE	DESCRIPTION
6024569	mediterranean cluster,discovery garden	Apartment For Rent: R02634, 79 - Discovery Gardens	:;gt; Mediterranean Cluster 09

4. Create a log for this matching in log table (AD_CORE_DETAIL_LOG) to state that the particular record is processed for ‘LOCATION’ matching. A separate column (LOCATION_MATCH) in log table will say if matching was successful (‘Y’) or not (‘N’).

   In step 3, we found the example to be a perfect match. Hence, the column LOCATION_MATCH is updated as ‘Y’ against the same ID



ID	LOCATION_MATCH
6024569	Y

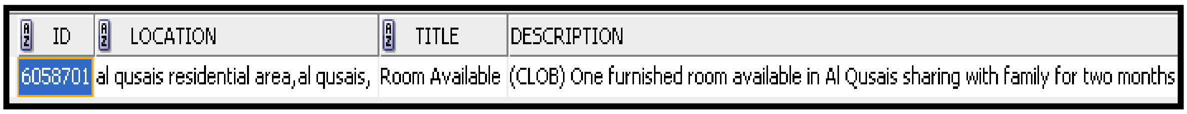
PART 2:

If perfect match doesn’t exist, try to find partial match for location.

The steps followed are:

1. Get the actual location value from dataset (table: ADDETAIL, Column: Location)

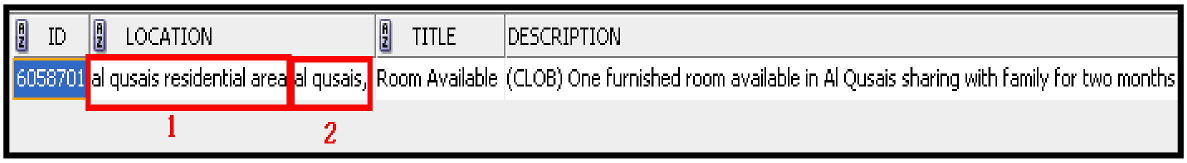
   E.g.



ID	LOCATION	TITLE	DESCRIPTION
6058701	al qusais residential area,al qusais,	Room Available	(CLOB) One furnished room available in Al Qusais sharing with family for two months

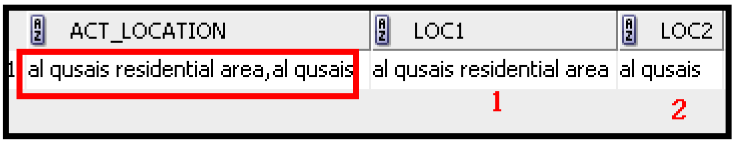
2. Get the individual location segments for the location value obtained in step 1.

   E.g. we have two word segments here.



ID	LOCATION	TITLE	DESCRIPTION
6058701	al qusais residential area al qusais,	Room Available	(CLOB) One furnished room available in Al Qusais sharing with family for two months

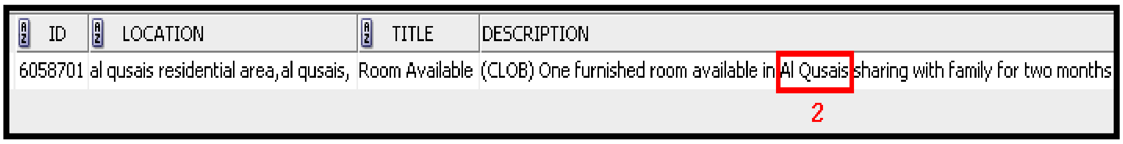
Note: Here the individual location segments are actually obtained from table ADLOCATION for the particular location.



	ACT_LOCATION	LOC1	LOC2
1	al qusais residential area,al qusais	al qusais residential area	al qusais

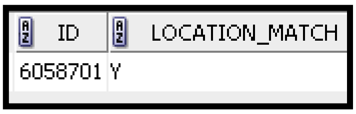
3. Check if at least a single location segment derived in step 2 is present either in TITLE or DESCRIPTION columns. If matching is found for at least one location segment but not for all, then it is considered as a partial match.

E.g. In this case only one location segment from step 2 above has found a match. Hence it is a partial matching.



ID	LOCATION	TITLE	DESCRIPTION
6058701	al qusais residential area,al qusais,	Room Available	(CLOB) One furnished room available in Al Qusais sharing with family for two months

4. Create log for this matching in a log table (AD_CORE_DETAIL_LOG) to state that the particular record is processed for ‘LOCATION’ matching.



ID	LOCATION_MATCH
6058701	Y

PART 3:

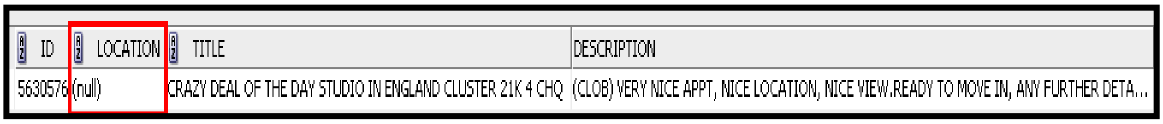
Find a location for records which have the location value missing or mismatched.

The steps followed are:

1. Get the record having missing or mismatched location value (table: ADDETAIL, Column: Location)

   E.g.

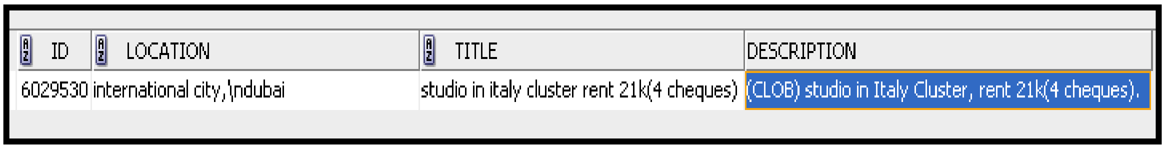
   - Missing location



ID	LOCATION	TITLE	DESCRIPTION
5630576	(null)	CRAZY DEAL OF THE DAY STUDIO IN ENGLAND CLUSTER 21K 4 CHQ	(CLOB) VERY NICE APPT, NICE LOCATION, NICE VIEW.READY TO MOVE IN, ANY FURTHER DETA...

- Location mismatch

  In below snapshot location is mismatched since the location segment ‘international city’ is not found in either TITLE or DESCRIPTION columns



ID	LOCATION	TITLE	DESCRIPTION
6029530	international city,\ndubai	studio in italy cluster rent 21k(4 cheques)	(CLOB) studio in Italy Cluster, rent 21k(4 cheques).

2. Find the best match for location from the set up table ADLOCATION. To find the best match, the below method is used:

For each i^{th} record R_i, with missing or mismatched location value in ADDETAIL

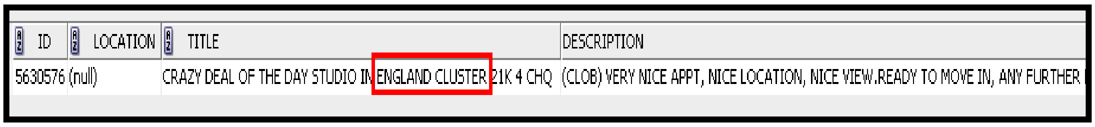
- Search the location segments(*) from each j^{th} record R_j of ADLOCATION in TITLE and DESCRIPTION columns of record R_i
- Count the location segment matches for each record R_j
- Select the value of ACT_LOCATION say R_j(act), which gives maximum count for location segment matches
- Add or replace the value R_j(act) in LOCATION column of record R_i

(*) Location segments points to values in columns LOC1, LOC2, LOC3 and LOC4 in ADLOCATION table.

E.g.

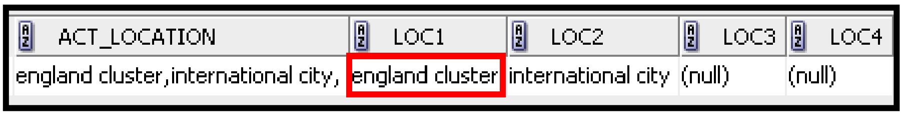
- Finding match for missing location

ADDETAIL



ID	LOCATION	TITLE	DESCRIPTION
5630576	(null)	CRAZY DEAL OF THE DAY STUDIO IN ENGLAND CLUSTER 21K 4 CHQ	(CLOB) VERY NICE APPT, NICE LOCATION, NICE VIEW.READY TO MOVE IN, ANY FURTHER

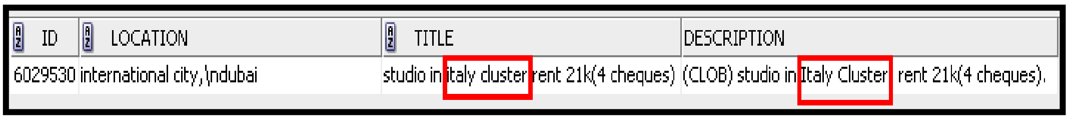
ADLOCATION



ACT_LOCATION	LOC1	LOC2	LOC3	LOC4
england cluster,international city,	england cluster	international city	(null)	(null)

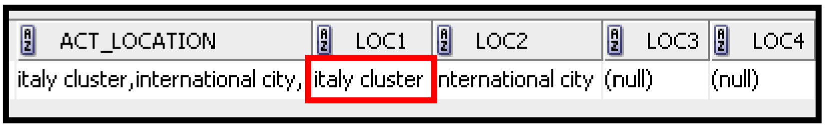
- Finding match for location mismatch

ADDETAIL



ID	LOCATION	TITLE	DESCRIPTION
6029530	international city,\ndubai	studio in italy cluster rent 21k(4 cheques)	(CLOB) studio in Italy Cluster rent 21k(4 cheques).

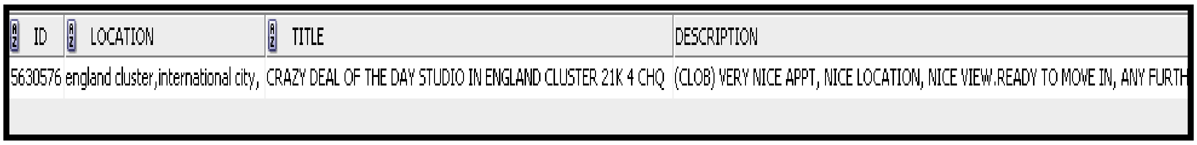
ADLOCATION



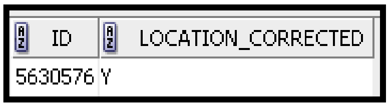
ACT_LOCATION	LOC1	LOC2	LOC3	LOC4
italy cluster,international city,	italy cluster	nternational city	(null)	(null)

3. When location value is updated for the records that were having missing location information in ADDETAIL table, LOCATION_CORRECTED column in log table AD_CORE_DETAIL_LOG is updated as ‘Y’. When location value is corrected as result of location mismatch, LOC_MISMATCH_CORRECTED column in log table AD_CORE_DETAIL_LOG is updated as ‘Y’.

E.g. Missing location corrected

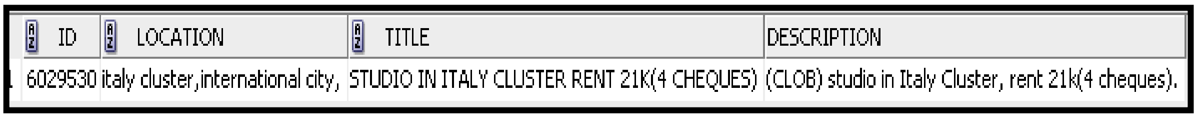


ID	LOCATION	TITLE	DESCRIPTION
5630576	england cluster,international city,	CRAZY DEAL OF THE DAY STUDIO IN ENGLAND CLUSTER 21K 4 CHQ	(CLOB) VERY NICE APPT, NICE LOCATION, NICE VIEW.READY TO MOVE IN, ANY FURTH

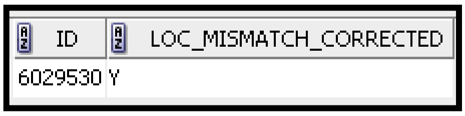


ID	LOCATION_CORRECTED
5630576	Y

- Location mismatch corrected



	ID	LOCATION	TITLE	DESCRIPTION
1	6029530	italy cluster,international city,	STUDIO IN ITALY CLUSTER RENT 21K(4 CHEQUES)	(CLOB) studio in Italy Cluster, rent 21k(4 cheques).



ID	LOC_MISMATCH_CORRECTED
6029530	Y

Figure 3.3 below summarizes the steps performed to data cleanse the data set for attribute ‘location’.

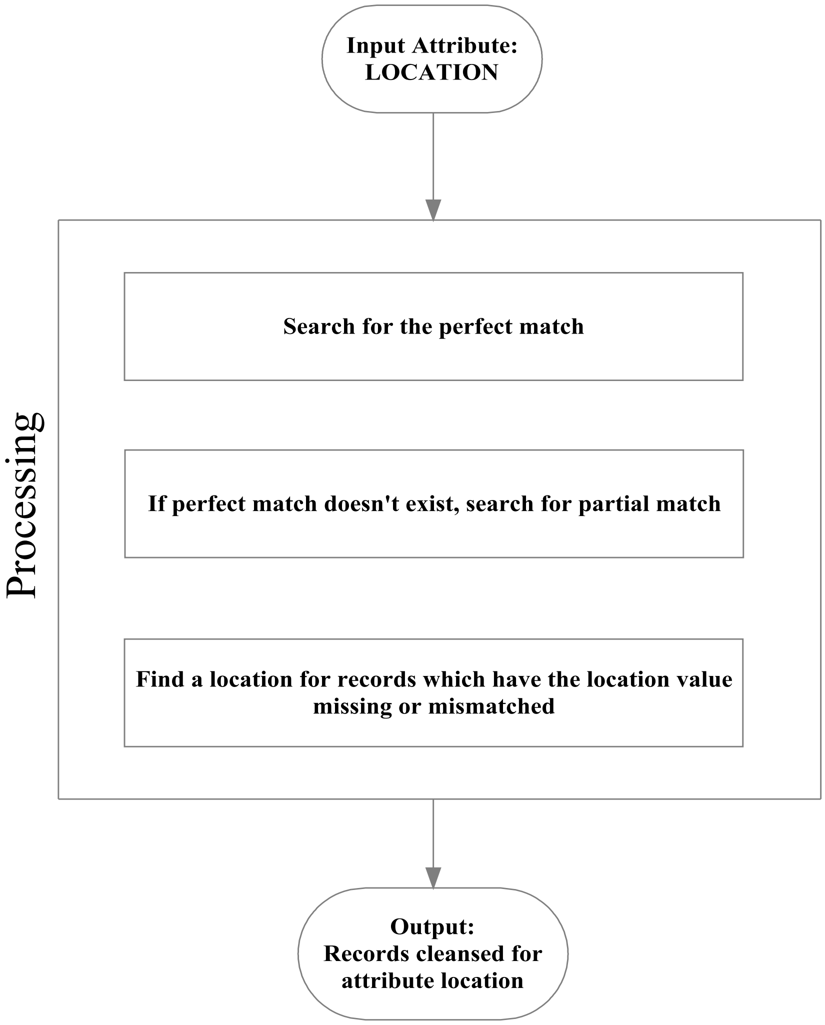

Input Attribute:
LOCATION
Processing
Search for the perfect match
If perfect match doesn't exist, search for partial match
Find a location for records which have the location value missing or mismatched
Output:
Records cleansed for attribute location

Figure 3.3: Data cleansing summarized for attribute ‘location’

Attribute: Beds

Data cleansing for attribute ‘Bed’ consists of three parts.

Algorithm Steps:

PART 1:

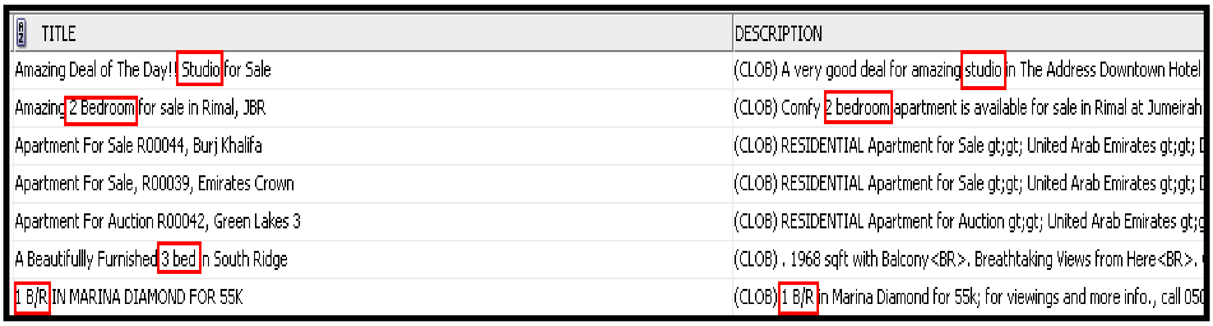
In the first part, we try to find the pattern used to denote information on number of ‘beds’.

The steps involved are:

1. Manually identify pattern keywords for attribute ‘beds’ from the sample dataset.

E.g. STUDIO is a keyword, which implies that there are no bedrooms in the apartment.

Another example is “2 bedrooms”. In this case ‘bedroom’ is a keyword.



TITLE	DESCRIPTION
Amazing Deal of The Day!! Studio for Sale	(CLOB) A very good deal for amazing studio in The Address Downtown Hotel
Amazing 2 Bedroom for sale in Rimal, JBR	(CLOB) Comfy 2 bedroom apartment is available for sale in Rimal at Jumeirah
Apartment For Sale R00044, Burj Khalifa	(CLOB) RESIDENTIAL Apartment for Sale gt;gt; United Arab Emirates gt;gt;
Apartment For Sale, R00039, Emirates Crown	(CLOB) RESIDENTIAL Apartment for Sale gt;gt; United Arab Emirates gt;gt;
Apartment For Auction R00042, Green Lakes 3	(CLOB) RESIDENTIAL Apartment for Auction gt;gt; United Arab Emirates gt;
A Beautifully Furnished 3 bed in South Ridge	(CLOB) . 1968 sqft with Balcony<BR>. Breathtaking Views from Here<BR>.
1 B/R IN MARINA DIAMOND FOR 55K	(CLOB) 1 B/R in Marina Diamond for 55k; for viewings and more info., call 05

2. Next is to derive the ways in which the information on number of ‘beds’ is used along with the identified keywords.

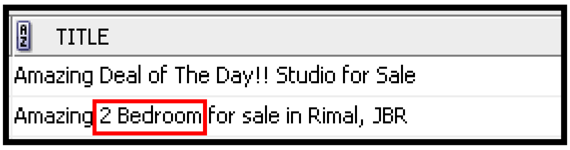
   E.g. let’s suppose if identified keyword is ‘bedroom’. This keyword becomes relevant information in our case when prefixed with a number or a meaningful word, i.e. for example ‘2 bedroom’ or ‘two bedroom’ or ‘single bedroom’

   So basically focus here is to derive the patterns that include the keyword to represent the information on ‘beds’.

   The patterns that were considered to be valid as to find information on ‘beds’ are:

   i. [digit] [keyword]

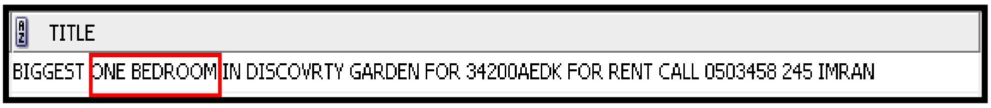
   For E.g. “5 Bedroom”



TITLE
Amazing Deal of The Day!! Studio for Sale
Amazing 2 Bedroom for sale in Rimal, JBR

   ii. [number in words] [keyword]

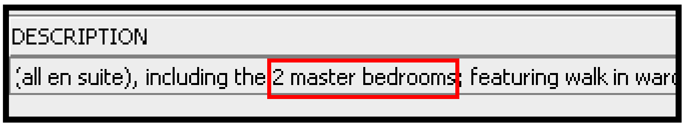
   For E.g. “One Bedroom”



TITLE
BIGGEST ONE BEDROOM IN DISCOVRTY GARDEN FOR 34200AEDK FOR RENT CALL 0503458 245 IMRAN

   iii. [digit] [qualifier] [keyword]

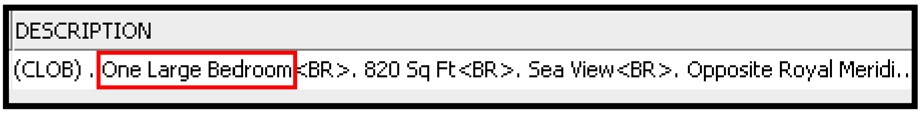
   For E.g. “2 master Bedrooms”



DESCRIPTION
(all en suite), including the 2 master bedrooms; featuring walk in ward

iv. [number in words] [qualifier] [keyword]

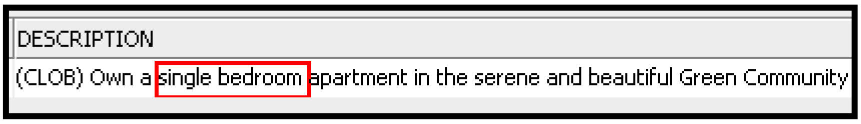
For E.g. “One large Bedroom”



DESCRIPTION
(CLOB) . One Large Bedroom<BR>. 820 Sq Ft<BR>. Sea View<BR>. Opposite Royal Meridi..

v. [meaningful qualifier] [keyword]

For E.g. ‘single bedroom’



DESCRIPTION
(CLOB) Own a single bedroom apartment in the serene and beautiful Green Community

Couples of cases are handled separately:

- STUDIO – was considered as case with zero bedroom
- BHK – if this keyword is given without any preceding number say just BHK instead of 2 BHK, it was considered advertisement for single bedroom.

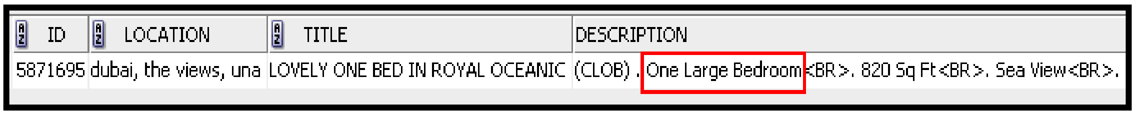
3. Check if the patterns derived in step 2 are found in the TILTE and DESCRIPTION columns of ADDETAIL table. If found, then:

   i. Update the log table AD_CORE_DETAIL_LOG for column BED_MATCH (stores ‘Y’ when matched) and column ‘BED_KEYWORD_MATCHED’, which stores the information on which keyword is matched against a particular record in dataset.

   E.g. let’s say the pattern we are searching is:

   [number in words] [qualifier] [keyword]

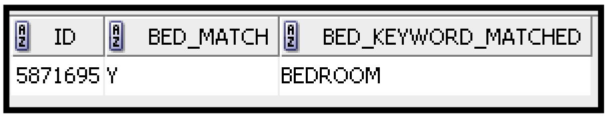
ADDETAIL table:



ID	LOCATION	TITLE	DESCRIPTION
5871695	dubai, the views, una	LOVELY ONE BED IN ROYAL OCEANIC	(CLOB) . One Large Bedroom<BR>. 820 Sq Ft<BR>. Sea View<BR>.

So we have a match for this pattern against ID 5871695. The log table is update for same ID.

AD_CORE_DETAIL_LOG table:



ID	BED_MATCH	BED_KEYWORD_MATCHED
5871695	Y	BEDROOM

The pattern keywords identified in the dataset for attribute ‘beds’ are:

1. BEDROOM
2. BD
3. B/R
4. BR
5. BED
6. –BR
7. EN-SUITE BEDROOMS
8. BHK
9. –BED
10. -B/R
11. B.H
12. B/D
13. B-R
14. /BR
15. .B/R
16. /-BR
17. -B/D
18. B / R
19. /BED
20. B.R
21. B/.R

Problems encountered:

1. Some property classifieds had the information on studio and bedroom apartments put together. Due to the ambiguity in these cases on deciding the number of beds, such records are not considered for data cleansing on attribute ‘beds’.

PART 2:

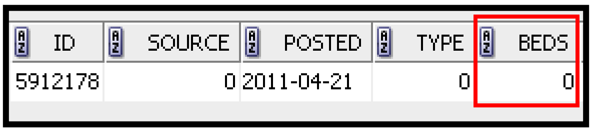
In the second part we try to match the information on ‘beds’ collected from TITLE or DESCRIPTION columns to the actual ‘beds’ information given in BEDS column, ADORE table.

The steps involved are:

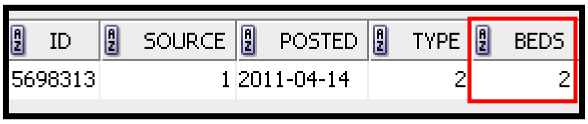
1. Get the actual number of beds from the dataset (table: ADCORE, column: BEDS), corresponding to the records in ADDETAIL for which the pattern keywords representing information on ‘beds’ are identified.

   E.g. consider below cases

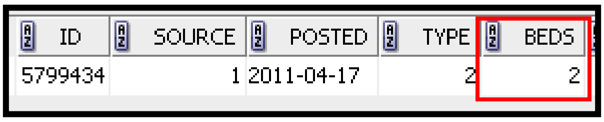
   ADCORE table



ID	SOURCE	POSTED	TYPE	BEDS
5912178	0	2011-04-21	0	0



ID	SOURCE	POSTED	TYPE	BEDS
5698313	1	2011-04-14	2	2

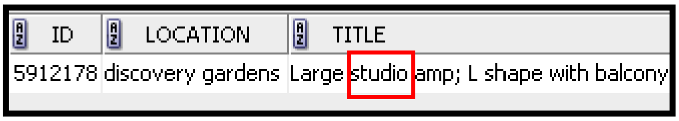


ID	SOURCE	POSTED	TYPE	BEDS
5799434	1	2011-04-17	2	2

2. If the keyword is STUDIO and actual value of number of beds (step 1) is 0 then the record is considered as matched for BED attribute and the same is logged in AD_CORE_DETAIL_LOG table for column BED_MATCH(‘Y’ for successful match).

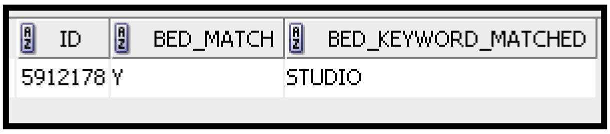
   E.g.

   ADDETAIL table



ID	LOCATION	TITLE
5912178	discovery gardens	Large studio amp; L shape with balcony

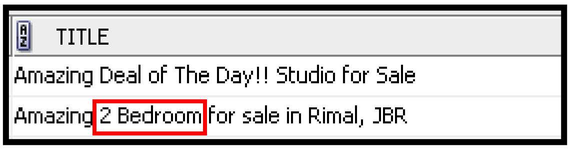
   AD_CORE_DETAIL_LOG table



ID	BED_MATCH	BED_KEYWORD_MATCHED
5912178	Y	STUDIO

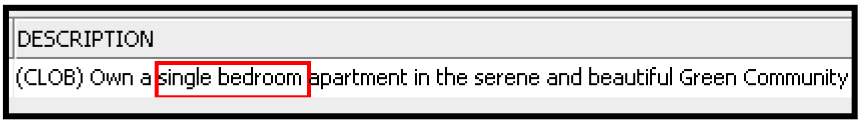
3. For other keywords there are two cases:

   - Number beds is given as a number like 1, 2, 3 …



TITLE
Amazing Deal of The Day!! Studio for Sale
Amazing 2 Bedroom for sale in Rimal, JBR

   - Number of beds is given as a word like one, two, double, single.



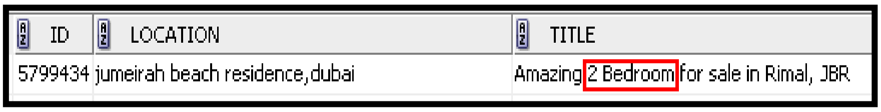
DESCRIPTION
(CLOB) Own a single bedroom apartment in the serene and beautiful Green Community

For the case where 'beds' value is given as **number**, method used for matching is:

i. Find the information on beds from TITLE or DESCRIPTION.

E.g. suppose 'bedroom' is the keyword identified. We have information based on this keyword as '2 bedroom'

ADDETAIL table

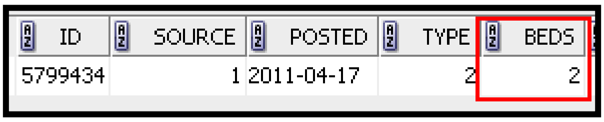


ID	LOCATION	TITLE
5799434	jumeirah beach residence,dubai	Amazing 2 Bedroom for sale in Rimal, JBR

ii. For the same ID, get the actual value on number of beds and join it with the keyword for 'beds' obtained from column BED_KEYWORD_MATCHED in AD_CORE_DETAIL_LOG table

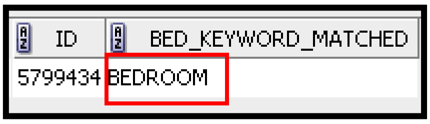
E.g.

From step 1, ADCORE table



ID	SOURCE	POSTED	TYPE	BEDS
5799434	1	2011-04-17	2	2

AD_CORE_DETAIL_LOG table



ID	BED_KEYWORD_MATCHED
5799434	BEDROOM

At the end of this step we have the information on beds framed as:

‘**2 bedroom**’

Where ‘**2**’ is from ADCORE table→column: BEDS

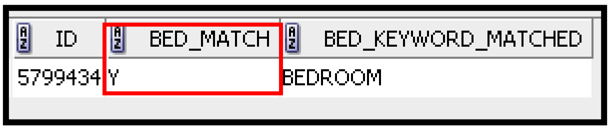
&

‘**bedroom**’ is from AD_CORE_DETAIL_LOG table →
column: BED_KEYWORD_MATCHED

iii. Compare the value in steps i & ii above. If they match, create a log in AD_CORE_DETAIL_LOG table, BED_MATCH column as ‘Y’.

From the example in seen I & ii above, we have a matching, so same is logged.

AD_CORE_DETAIL_LOG



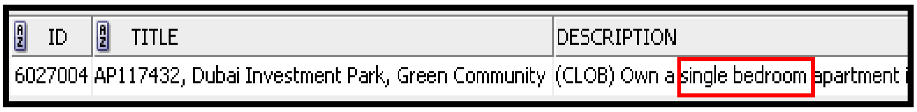
ID	BED_MATCH	BED_KEYWORD_MATCHED
5799434	Y	BEDROOM

For the case where ‘bed’ value is given as **word,** method used for matching is:

i. Find the information on beds from TITLE or DESCRIPTION.

E.g. suppose ‘bedroom’ is the keyword identified. We have information based on this keyword as ‘single bedroom’

ADDETAIL table

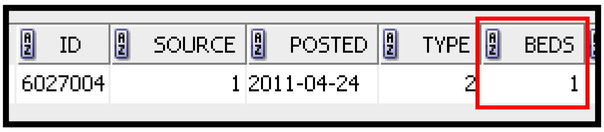


ID	TITLE	DESCRIPTION
6027004	AP117432, Dubai Investment Park, Green Community	(CLOB) Own a single bedroom apartment i

ii. For the same ID, get the actual value on number of beds, decode this actual value into words and join it with the keyword obtained from column BED_KEYWORD_MATCHED in AD_CORE_DETAIL_LOG table

E.g.

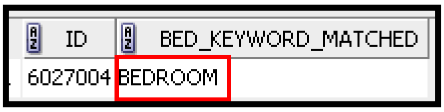
From step 1, ADCORE table



ID	SOURCE	POSTED	TYPE	BEDS
6027004	1	2011-04-24	2	1

Here value of BEDS is 1; so this value is decoded in words as either ‘one’ or ‘single’

AD_CORE_DETAIL_LOG table



ID	BED_KEYWORD_MATCHED
6027004	BEDROOM

At the end of this step we have two possibilities for the information on beds:

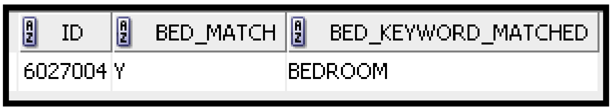
- ‘**one bedroom**’
- ‘**single bedroom**’

Where **one**/**single** is the decoded value for BEDS in ADCORE table→
column: BEDS
&
‘**bedroom**’ is from AD_CORE_DETAIL_LOG table →
column: BED_KEYWORD_MATCHED

iii. Compare the value in i & ii. If they match, create a log in AD_CORE_DETAIL_LOG table, BED_MATCH column as ‘Y’.

E.g. from (i) we have bed information as ‘single bedroom’ which matches with one of the possible derived information in (ii). Hence we have a match in this case.

AD_CORE_DETAIL_LOG



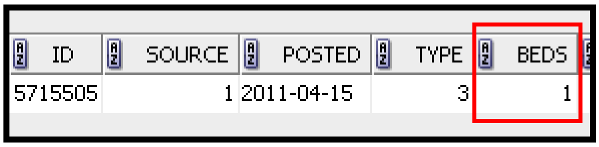
ID	BED_MATCH	BED_KEYWORD_MATCHED
6027004	Y	BEDROOM

PART 3:

This part deals with correcting the information on beds, in case of a mismatch. If there is a mismatch in the value of beds between the one in dataset when compared to the value on ‘beds’ derived in PART 2 above, the value derived in PART 2 is considered to be the right one and the same is corrected in the original dataset. A log for this correction is created in the log table AD_CORE_DETAIL_LOG, column BED_CORRECTED. A value ‘Y’ in column BED_CORRECTED suggests that the particular record in dataset is corrected for the information on ‘beds’.

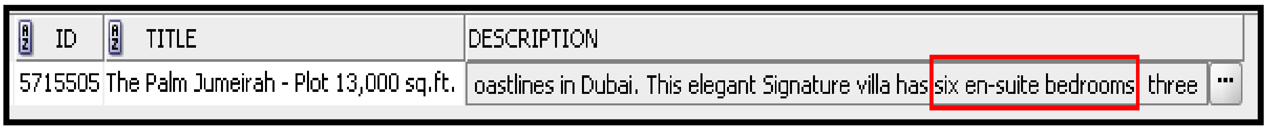
Example:

A record ADCORE table has BEDS as 1



ID	SOURCE	POSTED	TYPE	BEDS
5715505	1	2011-04-15	3	1

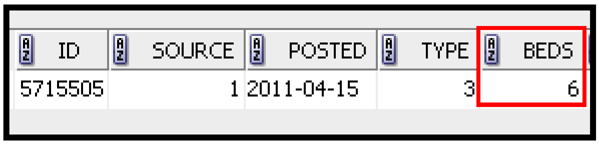
DESCRIPTION in ADDETAIL table says as 6 bedrooms for same record



ID	TITLE	DESCRIPTION
5715505	The Palm Jumeirah - Plot 13,000 sq.ft.	oastlines in Dubai. This elegant Signature villa has six en-suite bedrooms three

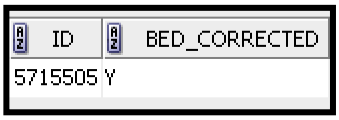
Hence, for this record, ADCORE table is corrected with the right value. Also, AD_CORE_DETAIL_LOG table records the log for this correction in column BED_CORRECTED.

ADCORE table:



ID	SOURCE	POSTED	TYPE	BEDS
5715505	1	2011-04-15	3	6

AD_CORE_DETAIL_LOG table:



ID	BED_CORRECTED
5715505	Y

Figure 3.4 below summarizes the steps performed to data cleanse the data set for attribute ‘beds’.

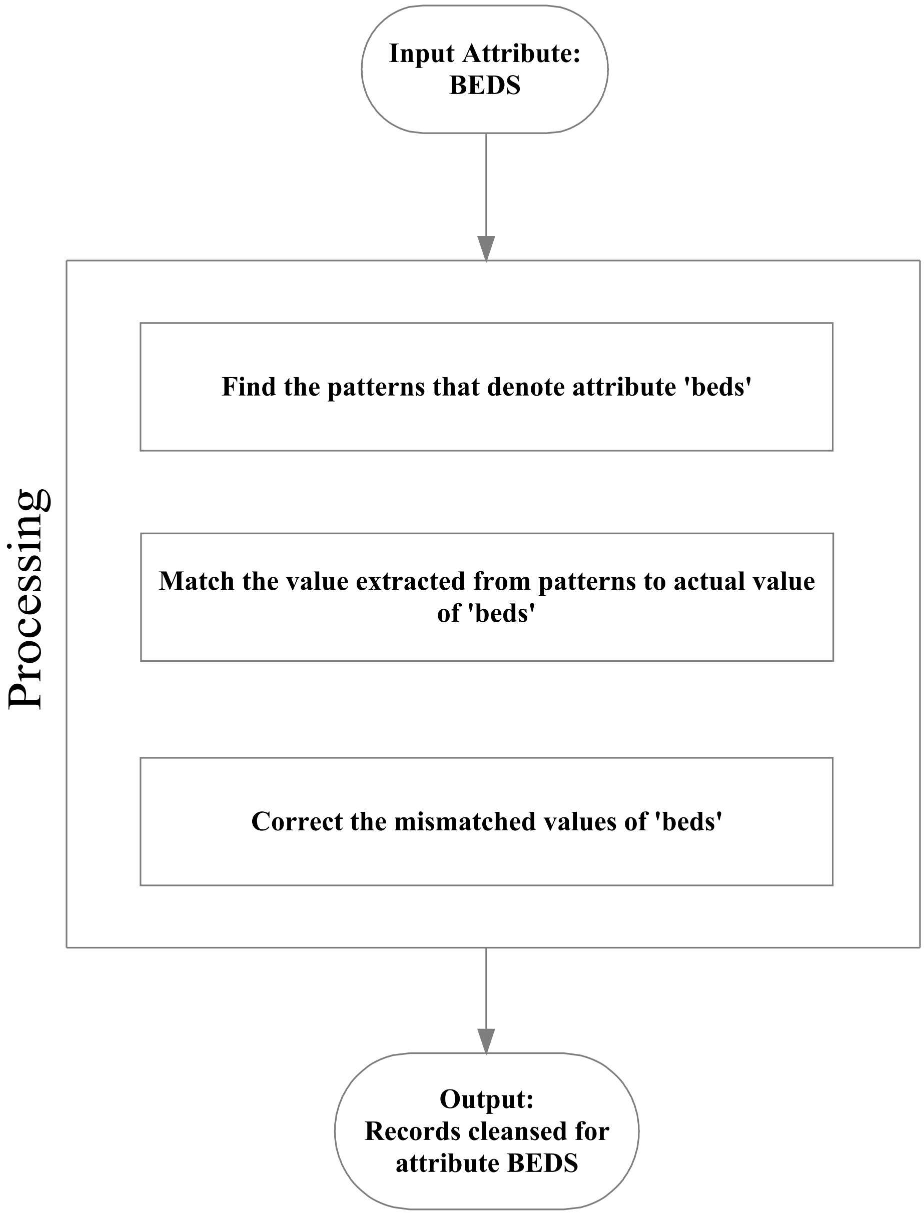

Input Attribute:
BEDS
Processing
Find the patterns that denote attribute 'beds'
Match the value extracted from patterns to actual value of 'beds'
Correct the mismatched values of 'beds'
Output:
Records cleansed for attribute BEDS

Figure 3.4: Data cleansing summarized for attribute ‘beds’

Attribute: Rent

Data cleansing for attribute ‘rent’ consists of following parts:

1. Find the patterns in which rent is given in a property classifieds.
2. Extract the rent value from the pattern found.
3. Match the extracted rent value with actual rent given in dataset.
4. Correct the mismatched rent.

PART 1: Find the patterns in which rent is given in a property classifieds

To find the patterns in which the attribute ‘rent’ is represented in a classified, the data under TITLE or DESCRIPTION columns (ADDETAIL table) are studied.

The meaningful patterns identified are:

1. AED [number] [K | M | MIL | MLN | MILLION | BILLION]

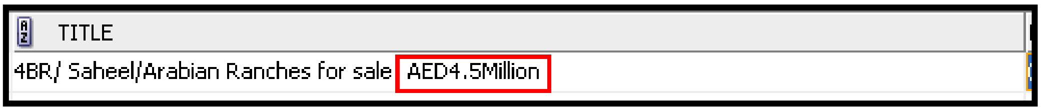
   ‘|’ represents OR.

   ‘K’ =1000

   ‘M’ or ‘MIL’ or ‘MLN’ = million

   E.g. AED 50K

   AED 1.2 MILLION

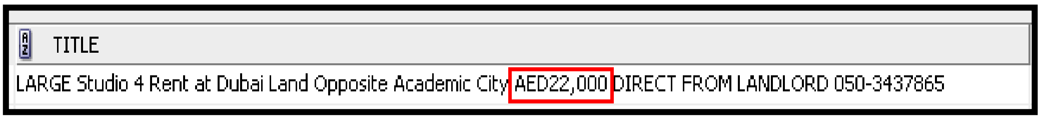


TITLE

4BR/ Saheel/Arabian Ranches for sale AED4.5Million

2. AED [number]

   E.g. AED 50,000

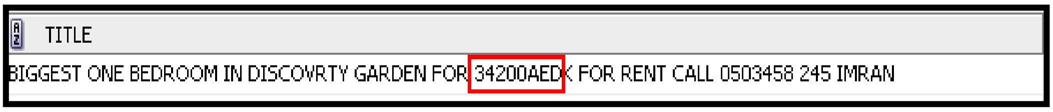


TITLE

LARGE Studio 4 Rent at Dubai Land Opposite Academic City AED22,000 DIRECT FROM LANDLORD 050-3437865

3. [number] AED

   E.g. 50,000 AED

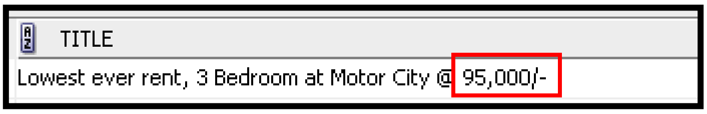


TITLE

BIGGEST ONE BEDROOM IN DISCOVRTY GARDEN FOR 34200AEDK FOR RENT CALL 0503458 245 IMRAN

4. [number] [/-]

   E.g. 50000/-

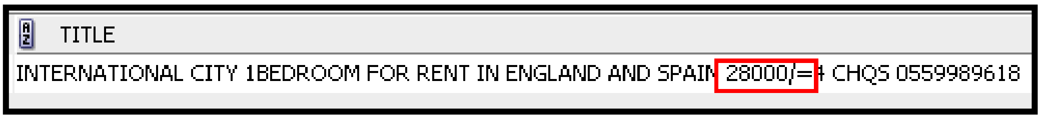


TITLE

Lowest ever rent, 3 Bedroom at Motor City @ 95,000/-

5. [number] [/=]

   E.g. 50000/=



TITLE

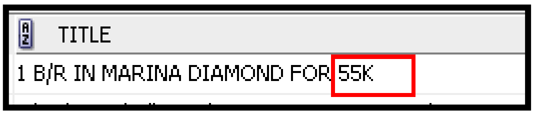
INTERNATIONAL CITY 1BEDROOM FOR RENT IN ENGLAND AND SPAIN 28000/=4 CHQS 0559989618

6. [number] [K | M | MIL | MLN | MILLION | BILLION]

   ‘K’ =1000

   ‘M’ or ‘MIL’ or ‘MLN’ = million

   E.g. 5 MLN, 35K



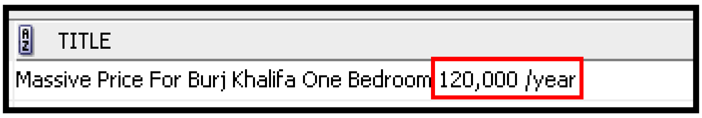
TITLE

1 B/R IN MARINA DIAMOND FOR 55K

7. [number] [/Y | /YR | /YEAR]

   ‘/Y’ & /YR denotes per year

   E.g. 75,000/Y



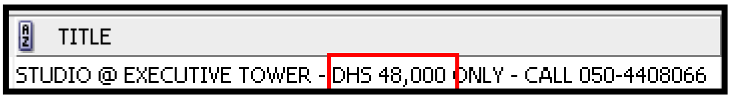
TITLE

Massive Price For Burj Khalifa One Bedroom 120,000 /year

8. [DHS | DHR | DHM | DHMS | DIRHAM] [number]

   ‘DHS’ or ‘DHR’ or ‘DHM’ or ‘DHMS’ = DIRHAMS

   E.g. DHS 80,000

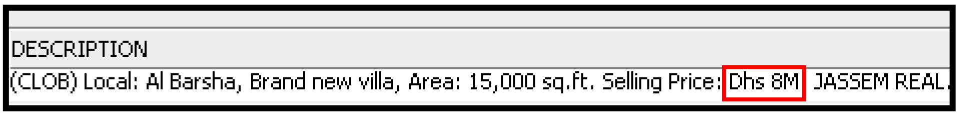


TITLE

STUDIO @ EXECUTIVE TOWER - DHS 48,000 ONLY - CALL 050-4408066

9. [DHS | DHR | DHM | DHMS | DIRHAM] [number] [K | M | MIL | MLN | MILLION | BILLION]

   E.g. DHR 3.2 MILLION

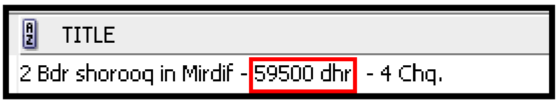


DESCRIPTION

(CLOB) Local: Al Barsha, Brand new villa, Area: 15,000 sq.ft. Selling Price: Dhs 8M JASSEM REAL.

10. [number] [DHS | DHR | DHM | DHMS | DIRHAM]

    E.g. 50000 DHM

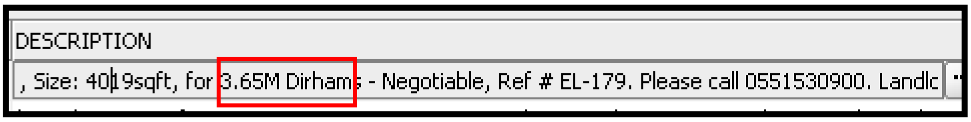


TITLE

2 Bdr shorooq in Mirdif - 59500 dhr - 4 Chq.

11. [number] [K | M | MIL | MLN | MILLION | BILLION] [DHS | DHR | DHM | DHMS | DIRHAM]

    E.g. 3.2 MILLION DHMS

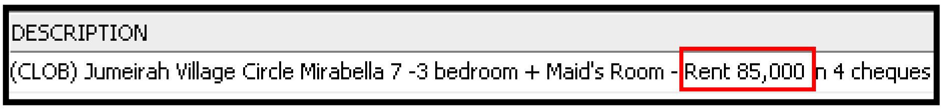


DESCRIPTION

, Size: 4019sqft, for 3.65M Dirhams - Negotiable, Ref # EL-179. Please call 0551530900. Landlc

12. [RENT] [number]

    E.g. RENT 50,000

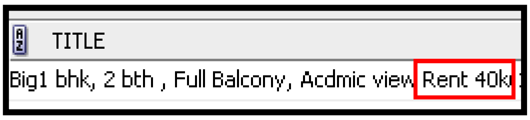


DESCRIPTION

(CLOB) Jumeirah Village Circle Mirabella 7 -3 bedroom + Maid's Room - Rent 85,000 in 4 cheques

13. [RENT] [number] [K]

    E.g. RENT 50K



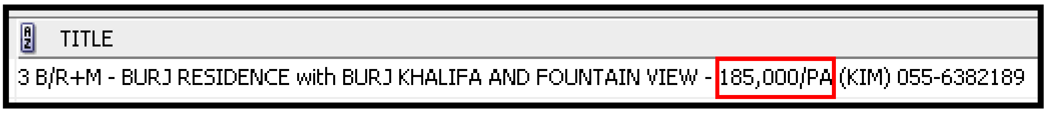
TITLE

Big1 bhk, 2 bth , Full Balcony, Acdmic view Rent 40k

14. [number] [/PA | PER YEAR]

    /PA = per year

    E.g. 75000/PA



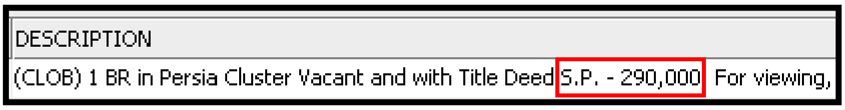
TITLE

3 B/R+M - BURJ RESIDENCE with BURJ KHALIFA AND FOUNTAIN VIEW - 185,000/PA (KIM) 055-6382189

15. [SELLING PRICE. | SELLING PRICE | S.P | S.P.] [number]

    S.P or S.P. = selling price

    E.g. S.P. 1,450,000

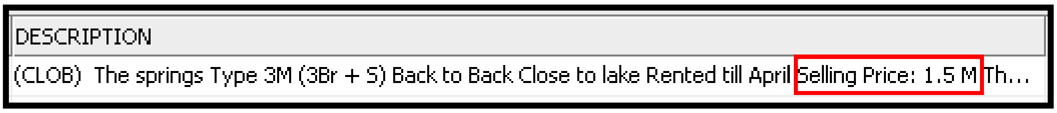


DESCRIPTION

(CLOB) 1 BR in Persia Cluster Vacant and with Title Deed S.P. - 290,000 For viewing,

16. [SELLING PRICE. | SELLING PRICE | S.P | S.P.] [number] [K | M | MIL | MLN | MILLION | BILLION]

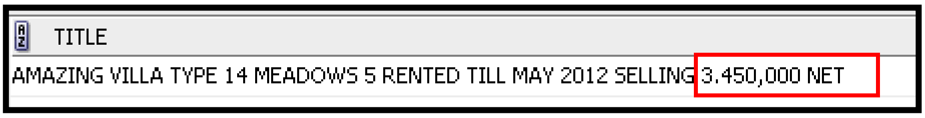
    E.g. S.P. 1.5 MILLION



DESCRIPTION
(CLOB) The springs Type 3M (3Br + S) Back to Back Close to lake Rented till April Selling Price: 1.5 M Th...

17. [number] [NET]

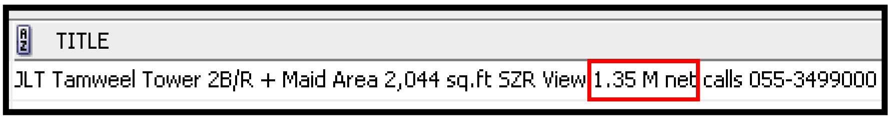
    E.g. 1,350,000 NET



TITLE
AMAZING VILLA TYPE 14 MEADOWS 5 RENTED TILL MAY 2012 SELLING 3.450,000 NET

18. [number] [K | M | MIL | MLN | MILLION | BILLION] [NET]

    E.g. 1.6 MILLION NET

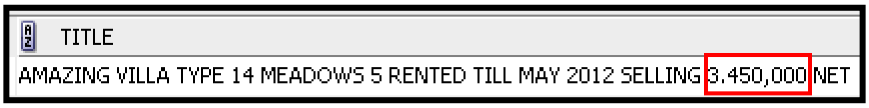


TITLE
JLT Tamweel Tower 2B/R + Maid Area 2,044 sq.ft SZR View 1.35 M net calls 055-3499000

PART 2: Extract the rent value from the patterns found.

In all the above said 18 patterns the segment of interest is [**number**]. [**number**] holds the value for rent. Further, in this part the value in [**number**] is extracted.

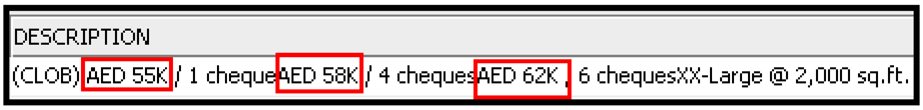
E.g. consider pattern no: 18 above; **[number]** is the one highlighted below



TITLE
AMAZING VILLA TYPE 14 MEADOWS 5 RENTED TILL MAY 2012 SELLING 3.450,000 NET

A point considered while extracting this number is:

If multiple occurrences of the same ‘rent’ patterns are found in TITLE or DESCRIPTION columns, but with different rent values, then such cases are considered to be ambiguous and are avoided.

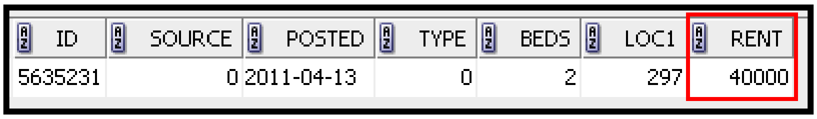


DESCRIPTION
(CLOB) AED 55K / 1 cheque AED 58K / 4 cheques AED 62K / 6 chequesXX-Large @ 2,000 sq.ft.

PART 3: Match the extracted rent value with actual rent given in dataset

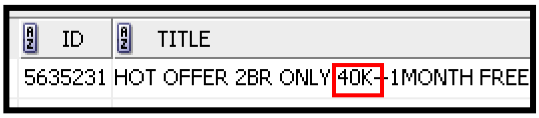
The extracted rent value is now matched with actual rent value (table: ADCORE, column: RENT). For every matched record on attribute 'rent', column RENT_MATCHED was updated with 'Y' in log table AD_CORE_DETAIL_LOG.

ADCORE table: ID 5635231



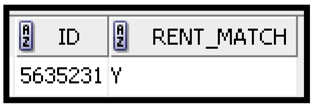
ID	SOURCE	POSTED	TYPE	BEDS	LOC1	RENT
5635231	0	2011-04-13	0	2	297	40000

ADDETAIL table: ID 5635231



ID	TITLE
5635231	HOT OFFER 2BR ONLY 40K+1MONTH FREE

AD_CORE_DETAIL_LOG table: ID 5635231



ID	RENT_MATCH
5635231	Y

Also, while matching for 'rent', following conversions are done:

1. If rent pattern has any of **[K | M | MIL | MLN | MILLION | BILLION],** then the [number] extracted was multiplied by:

   1000 for K

   1000000 for M or MIL or MLN or MILLION

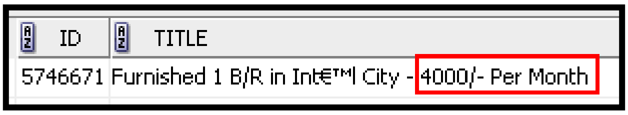
   1000000000 for BILLION

   In the above example shown, RENT in ADCORE table was 40000 but in TITLE of ADDETAIL table it is 40K. So here for comparison of actual RENT in ADCORE table, 40K was converted as 40 multiplied by 1000 i.e. 40000.

2. If rent pattern has any of **[PER MONTH | /MONTH | P.M | P/M]**, that denote monthly rent, then the [number] was multiplied by 12. This was done assuming that column RENT in table ADCORE holds the annual rent.

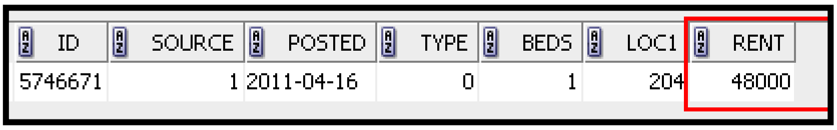
For e.g. in a case ADDETAIL table rent is given as "4000/- per month". So while comparing with the actual rent given in table ADCORE, we multiply 4000 by 12= 48000

ADDETAIL table:



ID	TITLE
5746671	Furnished 1 B/R in Int€™l City - 4000/- Per Month

ADCORE table:



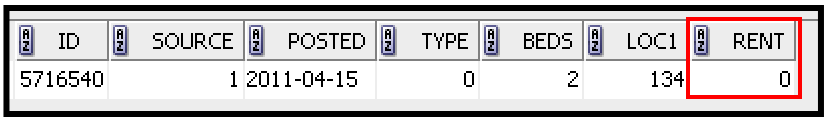
ID	SOURCE	POSTED	TYPE	BEDS	LOC1	RENT
5746671	1	2011-04-16	0	1	204	48000

PART 4: Correct the mismatched rent.

If matching of 'rent' is unsuccessful, the value for rent extracted from 'rent' pattern is used to correct the rent value in dataset (column: RENT, table ADCORE). For each such correction, column RENT_CORRECTED was updated with 'Y' in log table AD_CORE_DETAIL_LOG.

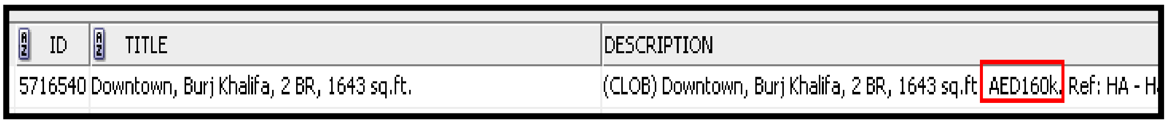
An example:

ADCORE table prior to rent correction: "0"



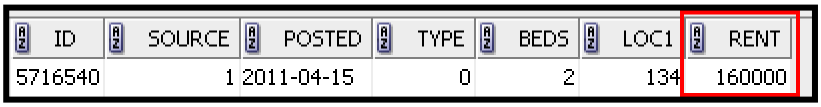
ID	SOURCE	POSTED	TYPE	BEDS	LOC1	RENT
5716540	1	2011-04-15	0	2	134	0

Value got from DESCRIPTION, ADDETAIL table: 'AED 160K', which after conversion becomes 160000



ID	TITLE	DESCRIPTION
5716540	Downtown, Burj Khalifa, 2 BR, 1643 sq.ft.	(CLOB) Downtown, Burj Khalifa, 2 BR, 1643 sq.ft AED160k. Ref: HA - H

ADCORE table after rent correction: "160000"



ID	SOURCE	POSTED	TYPE	BEDS	LOC1	RENT
5716540	1	2011-04-15	0	2	134	160000

Figure 3.5 below summarizes the steps performed to data cleanse the data set for attribute 'rent'.

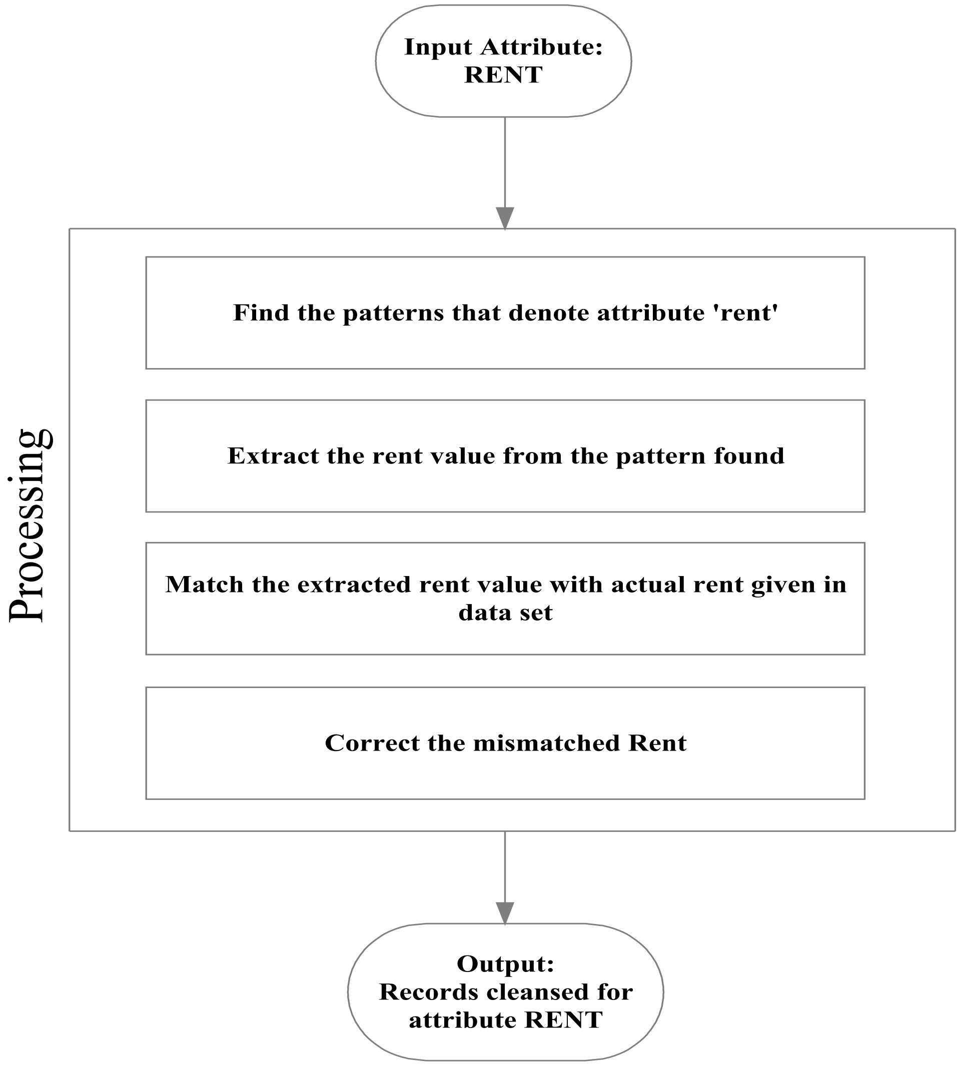

Input Attribute:
RENT
Processing
Find the patterns that denote attribute 'rent'
Extract the rent value from the pattern found
Match the extracted rent value with actual rent given in data set
Correct the mismatched Rent
Output:
Records cleansed for attribute RENT

Figure 3.5: Data cleansing summarized for attribute 'rent'

Chapter 4: Keyword Extraction

Once the data cleansing is completed, the next task is to extract the keywords. For extracting the keywords from the given dataset, we will consider only those records which have been verified for all the three attributes location, beds and rent.

4.1 Tables

The database table used in keyword extraction phase is:

1. AD_KEYWORD_GENERAL

   The purpose of having this table was to store the candidate keywords and use them for later analysis for finding the keywords and key phrases.

Column	Description
ID	Unique identifier of a record
AREA	Name of Area/place
TYPE	Denotes if the property being advertised is an apartment or villa and whether it is for sale or for rent
RENT	Denotes the rental price or sale price of the property
BEDS	Denotes the number of bedrooms in the property
KEYWD	individual words in the classified
WORD_POSITION	Position number of the word in classified
AVG_RENT	average rent grouped by area, type and beds
ASSOC_RENT_LEVEL	specifies if the rent is above average or below average in case of average method OR specifies if the rent is above median or below median in case of median method
MEDIAN_RENT	median rent grouped by area, type and beds

Table 4.1: Structure of table AD_KEYWORD_GENERAL

4.2 Candidate Keywords

Methods Employed:

To find the candidate keywords, we will split the dataset using attribute rent as the divider. Here we will use two methods to split the data set, one with average rent and other with median rent.

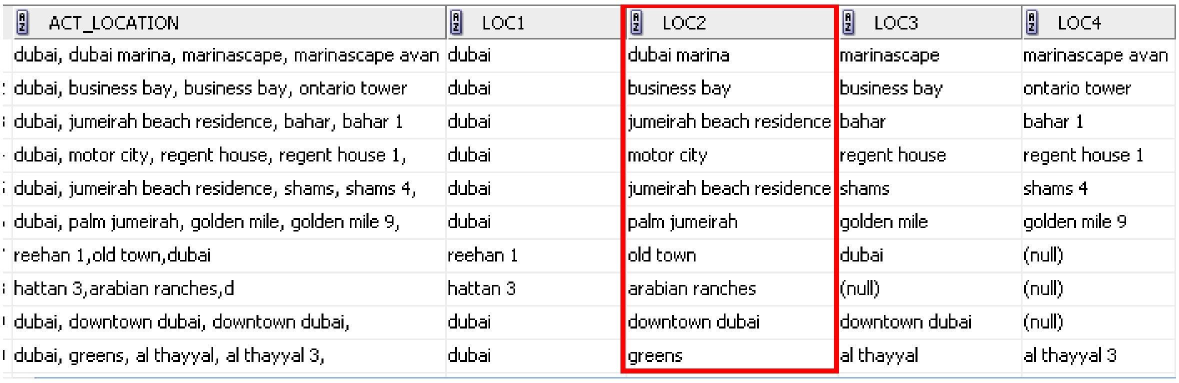
1. Average Rent method:

Steps followed to find the candidate keywords using average method are:

1. Get the distinct Area from the cleaned dataset.

   The area is obtained from LOCATION column in ADDETAIL table. As we already have the distinct LOCATION value split up and stored in set up table ADLOCATION table for data cleansing, the same is used for identifying the distinct 'area' given in the cleaned dataset.

   From visualizing the actual location (column ACT_LOCATION) and its individual location segments in ADLOCATION table, it is identified that the value in column LOC2 mostly signifies the area names. Hence the distinct value of this column is considered to denote the 'area'.



ACT_LOCATION	LOC1	LOC2	LOC3	LOC4
dubai, dubai marina, marinascape, marinascape avan	dubai	dubai marina	marinascape	marinascape avan
dubai, business bay, business bay, ontario tower	dubai	business bay	business bay	ontario tower
dubai, jumeirah beach residence, bahar, bahar 1	dubai	jumeirah beach residence	bahar	bahar 1
dubai, motor city, regent house, regent house 1,	dubai	motor city	regent house	regent house 1
dubai, jumeirah beach residence, shams, shams 4,	dubai	jumeirah beach residence	shams	shams 4
dubai, palm jumeirah, golden mile, golden mile 9,	dubai	palm jumeirah	golden mile	golden mile 9
reehan 1,old town,dubai	reehan 1	old town	dubai	(null)
hattan 3,arabian ranches,d	hattan 3	arabian ranches	(null)	(null)
dubai, downtown dubai, downtown dubai,	dubai	downtown dubai	downtown dubai	(null)
dubai, greens, al thayyal, al thayyal 3,	dubai	greens	al thayyal	al thayyal 3

   A problem with finding the distinct area was with incomplete words which denote the same 'area'.

   For instance we have areas like: **Palm Jum, Palm Jumeir, Palm Jumeirah.** Although, the above three names stand for same area, the first two have truncated values in given dataset. To handle such cases, a small algorithm was written to translate the incomplete area names to a more complete one. This will help in segregating only the correct area names when average rent is found, where grouping the area names is one of the criteria.

The steps followed in this translation are:

i. Order the list of distinct areas in ascending.

Area
…..
Palm Jum
Palm Jumeir
Palm Jumeirah
……….

ii. For each area, loop through the subsequent records to find if the same string is present in any of the records below it. If any subsequent matching is found, take the value of the last matched strings as the most complete area name.

That means, say we have **Palm Jum** and below that we have **Palm Jumeir** and **Palm Jumeirah** in order. Since we have ordered them in ascending, the most complete name will come at last. Now we try to find if **Palm Jum** is present in any of the subsequent area names. As a result, we can find that string **Pam Jum** present in both **Palm Jumeir** and **Palm Jumeirah**, but since the last match will be **Palm Jumeirah**, it will be considered to be the most complete name for **Palm Jum**.

2. Find the average rent for an area.

   The average rent is calculated by grouping area, type (rent/sale) and beds.

   The average is only considered for those records for which the data cleansing was possible for all three attributes of LOCATION, BEDS and RENT.

3. Compare the rent value of each record in cleansed dataset with the average rent as calculated in step 2 above.

   If the rent was above the average it was stored with value ‘ABV_AVG’ in column ASSOC_RENT_LEVEL of AD_KEYWORD_GENERAL table and ‘BEL_AVG’ if below average.

4. Split the words in each classified.

   We consider two columns in ADDETAIL table to split the words in a classified, i.e. TITLE and DESCRIPTION.

   Further, we spilt the sentences in columns TITLE and DESCRIPTION to individual words. Here the special characters and numbers are avoided. Only English alphabets are considered

   An example for splitting sentences is as follows:

   a. Suppose TITLE says:

      Al Barsha, 5 B/R, european kitchen,   private pool !!!!

   b. Remove special characters and numbers to keep only english alphabets.

      Al Barsha BR european kitchen   private pool

   c. Adjust the spacing between words to make it to single spacing

      Al Barsha BR european kitchen private pool

   d. Split the sentence into units of words and store them in table AD_KEYWORD_GENERAL

Al
Barsha
BR
european
kitchen
private
pool

2. Median Rent method:

A second method was used with median rent instead of average rent. The steps used for the approach is the same as in ‘average rent method’ except for the reason that here ‘median’ is used instead of ‘average’, to split the records in data set.

The motive behind using median approach is to check if ‘average rent method’ is skewed due to any possible outliers, as median helps reducing the attachment to any outliers. Outlier here will be for example a classified for a furnished apartment where rent is quoted higher than normal.

Some additional criteria used for candidate keyword extraction:

1. Remove the instances of html tags. This was done by repeated searching of the all html tags in the cleaned dataset.

2. Thesaurus approach

   Initially separated out words from the classifieds had lot of meaningless words. To fix this problem and filter out the meaningful words, a setup was created for English dictionary. This was done by creating an Oracle control file that had around 70,000+ English words. The source for this set up was obtained from:

   http://asktom.oracle.com/pls/asktom/f?p=100:11:0::::P11_QUESTION_ID:854500633682#2748709100346763819

   A disadvantage with this approach is that we could lose some nouns like name of place that were local to the Dubai real estate property classifieds.

3. A criterion was added to filter out the only words that are of length greater than 2 letters.

4. Only records above or below the median rent were considered, thereby ignoring the words in records which had their RENT value equal to the corresponding median value.

5. Remove the common words like the, here, where etc. A list was created for these so called stop words.

   Sources of stop words are:

   http://www.ranks.nl/resources/stopwords.html

   http://www.codingforums.com/showthread.php?t=172760

Comparison of Average and Median methods:

The ‘average rent method’ shows a large number classifieds being below average. Whereas, ‘median rent method’ depicted a better distribution of classifieds under above and below median category. Therefore we will use the candidate keywords obtained by using ‘median rent method’ for further analysis.

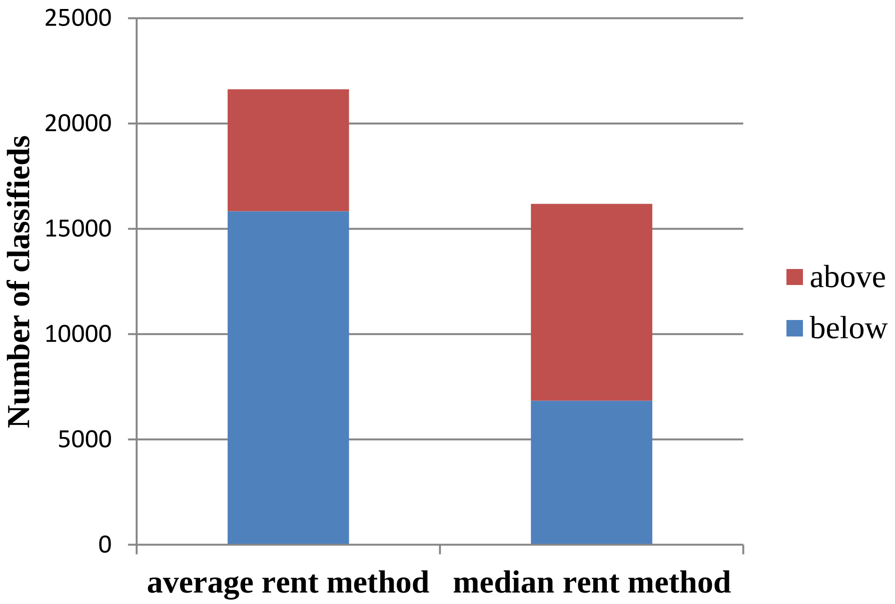


_Figure 4.1: Comparison of average and median methods

4.3 Keywords

From initial analysis we got a list of candidate keywords. Further, the occurrence of each candidate keyword is calculated for above and below median cases. A snapshot of the candidate keywords generated is given below (table 4.2), where the candidate keywords, their occurrences in data set and their respective percentage in the data set in enumerated.

Below Median case				Above Median case		
Candidate Keyword	occurrence	% in total classifieds		Candidate Keyword	occurrence	% in total classifieds
view	2923	21.06		view	3671	26.45
rent	2772	19.97		rent	2465	17.76
marina	1800	12.97		marina	1794	12.93
pool	1542	11.11		bedroom	1635	11.78
bedroom	1536	11.07		floor	1420	10.23
floor	1339	9.65		room	1417	10.21
type	1288	9.28		pool	1408	10.14
room	1283	9.24		property	1408	10.14
viewing	1275	9.19		type	1321	9.52
apartment	1222	8.8		apartment	1242	8.95

Table 4.2: Top candidate keywords extracted using ‘median rent method’

As observed in table 4.2, most of the top occurring candidate keywords in above and below median cases are same. Since the occurrence may not always convey the significance of a candidate keyword, we will devise a method to find the keywords that are actually significant to a particular case of above or below median.

To find out the relevance of a candidate keyword to above /below median case we will consider 3 factors:

1. **Word weight in a classified**

   This function gives the weight score for a word (Chen et al., 2008, p. 32).

   Word weight $W\ w$ is calculated as:

   $$\boldsymbol{W\ w = tf * itf}$$

   Where tf is the frequency of a word in a classified

And

itf* =log *(N/n)

Where N = total number of classifieds in the dataset

n = number of classifieds in which the word appear

2. **Position weight for word** Wp

This function adds weight to the word based on its position (Viji, n.d.). In our data set, the positions are TITLE and DESCRIPTION columns in ADDETAIL table. This function allows adding more value to a word, if it is from TITLE rather than DESCRIPTION. It is based on the assumption that words in TITLE are carefully selected. Wp also helps in balancing out the weights of words by giving more importance to words in TITLE; especially, when we have the same word existing in both TITLE and DESCRIPTION.

For instance we have an example described in table below.

Classified	Word	Position	Case
1.	‘word1’	TITLE	Below median
2.	‘word1’	DESCRIPTION	Above median

Here the same ‘word1’ occur in both above and below median cases, one in TITLE of below median case and other in DESCRIPTION of above median case. Without any measure like word position weight Wp, ‘word1’ has same weight in both the cases. But when we introduce word position weight Wp, it adds more weight to word ‘word1’ in classified no.1, thereby increasing the chances of ‘word1’ to become a keyword from below medina case.

Word position weight Wp is calculated as:

Wp*=*W(w)**0.75** → when word is positioned in only TITLE or both in
TITLE& DESCRIPTION of the classified

Wp*=*W(w)**0.25** → when word is positioned in only DESCRIPTION of the classified

Assumption:

The motive for selecting values 0.75 for TITLE and .25 for DESCRITON is just to give higher weight for TITLE than DESCRITION. Per se, it could be any value where TITLE is given higher weight than DESCRIPTION.

3. **Average word weight** in the above/below median case

This function helps in gauging the actual relevance of a word to above or below median case, by measuring its average weight.

Average word weight in above median case is given as:

$$\boldsymbol{Wwa} = \frac{\sum_{\mathbf{i=1}}^{\mathbf{n}}(\boldsymbol{Ww_i a} + \boldsymbol{Wp_i a})}{\mathbf{N}}$$

Where,

$Ww_i a$ = weight of the word in i^{th} classified that belongs to above median case

$Wp_i a$ = word position weight in i^{th} classified that belongs to above median case

N = total number of classifieds in above median case where the word appears

Word weight in below median case is given as:

$$\boldsymbol{Wwb} = \frac{\sum_{\mathbf{i=1}}^{\mathbf{n}}(\boldsymbol{Ww_i b} + \boldsymbol{Wp_i b})}{\mathbf{N}}$$

Where,

Ww_ib = weight weight of the word in i^{th} classified that belongs to below median case

Wp_ib = word position weight in i^{th} classified that belongs to below median case

N = total number of classifieds in below median case where the word appears.

As a next step, we assign the average word weights calculated to the candidate keywords identified in above and below medina cases.

Further, to find the keywords belonging to above and below median cases, the following check is performed:

1. If the average weight of a candidate keyword word is greater in above median case, when compared to the same in below median case, then this candidate keyword is considered to be the keyword for above median case.

2. If the average weight of a candidate keyword word is greater in below median case, when compared to the same in above median case, then this candidate keyword is considered to be the keyword for below median case.

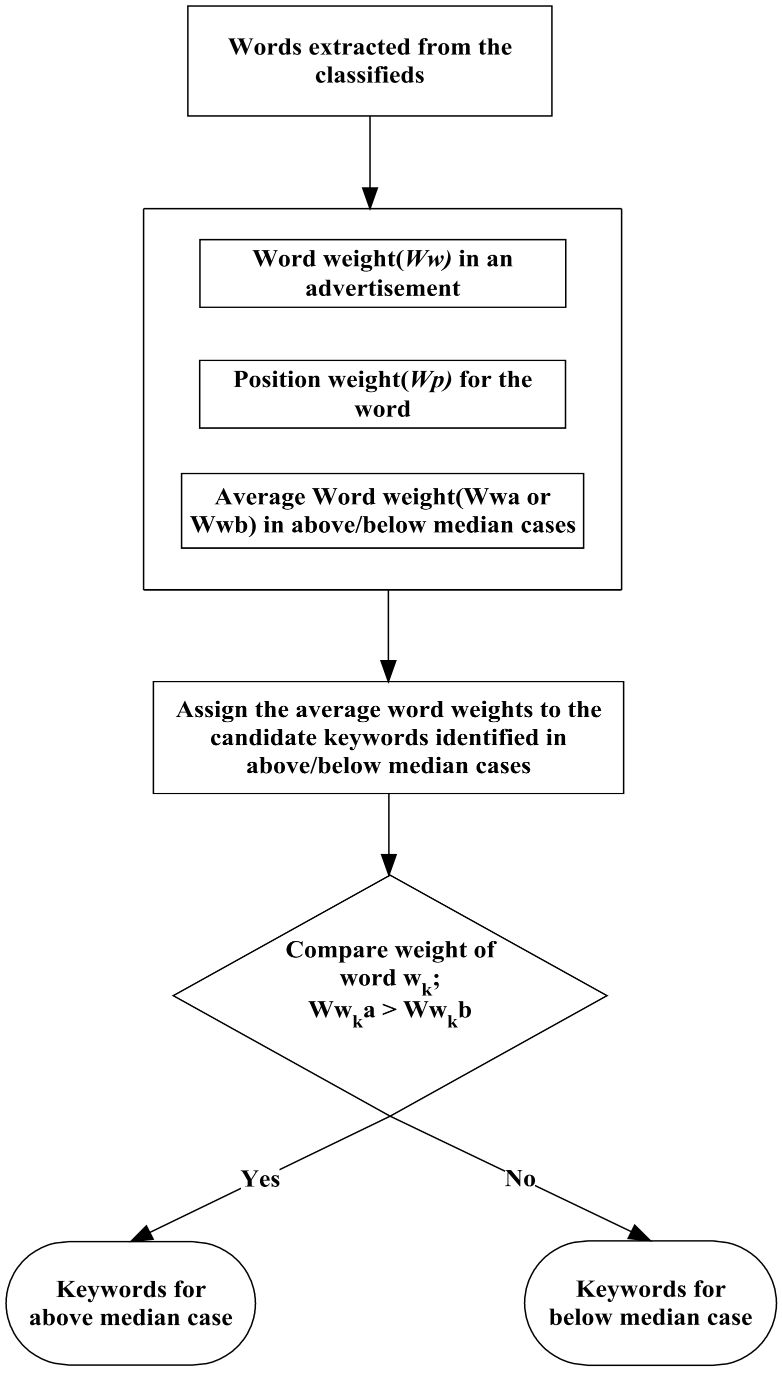
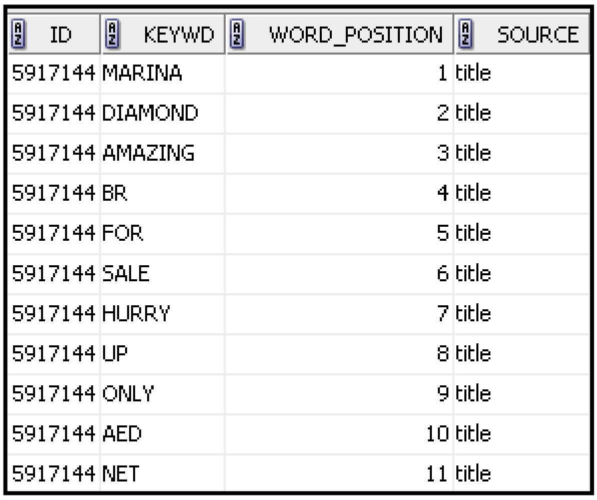

Words extracted from the classifieds
Word weight(*Ww)* in an advertisement
Position weight(*Wp)* for the word
Average Word weight(Wwa or Wwb) in above/below median cases
Assign the average word weights to the candidate keywords identified in above/below median cases
Compare weight of word w_k; $Ww_ka > Ww_kb$
Yes
No
Keywords for above median case
Keywords for below median case

Figure 4.2: A flow diagram on keyword generation

4.4 Key Phrases

The objective here is to extract the phrases from the classifieds, formed along with the keywords identified.

As a pre requisite to generate phrases, we use the word positions that were previously stored (table: AD_KEYWORD_GENERAL) while splitting up the sentences in the classified to extract the candidate keywords. These word positions represent the position of a word in either TITLE or DECRIPTION of a particular classified.



ID	KEYWD	WORD_POSITION	SOURCE
5917144	MARINA	1	title
5917144	DIAMOND	2	title
5917144	AMAZING	3	title
5917144	BR	4	title
5917144	FOR	5	title
5917144	SALE	6	title
5917144	HURRY	7	title
5917144	UP	8	title
5917144	ONLY	9	title
5917144	AED	10	title
5917144	NET	11	title

To find out the relevance of a phrase and to generate ‘top-k’ phrases in above /below median case we will consider below factors:

1. **Phrase weight** – is the weight of a phrase formed by the keyword and is represented as Pt

Pt = Pw + Pp

Where Pw = average weight of words in a phrase

$$\boldsymbol{Pw = \frac{\sum_{i=1}^{n} Ww_i}{N}}$$

Where,

Ww_i = average weight of i^{th} word in the phrase.

Ww_i can be Wwa or Wwb depending on above or below median case.

N= total number of words in the phrase

Pp = weight as per the position of the phrase; position weight is assigned based on same assumption made for calculating the position weight of word.

Pp=Pw*.75 → if when word position in only TITLE or both in TITLE& DESCRIPTION of the classified

Pp=Pw*.25 → if phrase position is only in DESCRIPTION of the classified

2. **Relevance of phrase Rp** – means how significant a phrase is to either above median case or below median case. This function is calculated by grouping the same phrases and calculating the sum their weights. This means higher occurrence of a phrase in the dataset will give a higher value for the measure Rp and thereby indicate significance of the phrase generated.

$$Rp = \sum_{\mathrm{i=1}}^{\mathrm{n}} (Pt_i)$$

Where,

Pt_i = weight of the phrase in i^{th} classified in above/below median case

Rp is a measure that is also used to generate 'top-k' key phrases.

Considering the above measures for a phrase, we can have an algorithm to generate 'top-k phrases based on the keywords identified.

The steps followed are:

1. Select a keyword for which the phrases are to be extracted from dataset.
   If the keyword is from, say above median case then the phrases will be extracted only from classifieds categorized under above median case.

   To begin with we need additional input criteria.

   →Maximum width of the phrase to be formed

Here we will be using keyword-centric approach, where the phrases will be generated by extracting words to either sides of the keyword. Hence, the value of maximum width in this context would represent the maximum number of words that are added to either sides of the keyword.

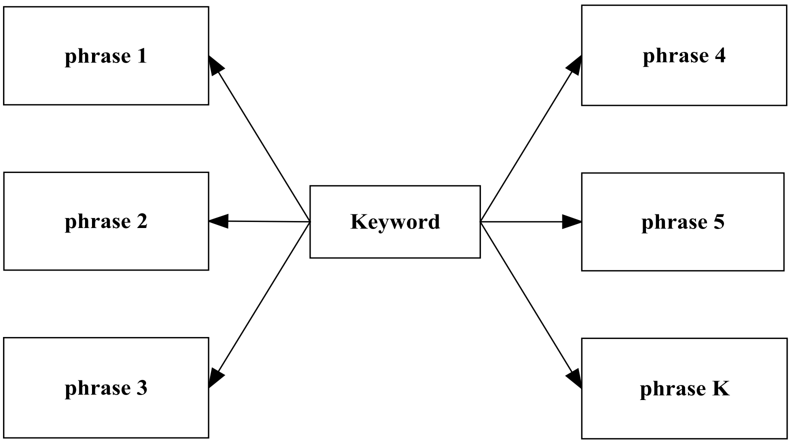


Figure 4.3: Keywords part of phrases

Considering keyword centric approach, let's suppose the input criteria for maximum width of the phrase is given as 2 for a keyword 'pool'. Therefore the maximum length of phrase formed will have 5 words, i.e. 2 words to the right and 2 words to the left of keyword 'pool'.

words	nice	swimming	**pool**	view	and	…
word position	3	4	5	6	7	…

2. For all the records in dataset which has the keyword chosen in step 1, generate all the possible combinations of phrases, from a phrase width of 1 to a maximum given phrase width. Considering the same example as in step 1, following combinations or rather phrases are generated for a maximum width of 2:

words	**pool**	view
word position	5	6

words	swimming	**pool**
word position	4	5

words	**pool**	view	and
word position	5	6	7

words	swimming	**pool**	view
word position	4	5	6

words	nice	swimming	**pool**
word position	3	4	5

words	nice	swimming	**pool**	view
word position	3	4	5	6

words	swimming	**pool**	view	and
word position	4	5	6	7

words	nice	swimming	**pool**	view	and
word position	3	4	5	6	7

As observed, with a max phrase width 2 we could generate 8 combinations.

Similarly,

Max phrase width of 1 implies 3 combinations

Max phrase width of 2 implies 8 combinations

Max phrase width of 3 implies 15 combinations

Max phrase width of 4 implies 24 combinations

Hence, in general,

For max phrase width ‘n’ we can generate ‘n*(n+2)’ combinations or phrases.

3. Group the same phrases and sum up their relevance measure Rp.

   A higher Rp with respect to a keyword will also imply higher percentage use of keyword as part of phrase, which could also add value to the keyword being used to generate the phrase.

4. Rank the phrases in order of sum of Rp, obtained from step 3 and list out the 'top-k' phrases generated for a threshold value 'k'. Here 'k' is a numeric input which can be assigned values ranging from 1 to 'k'.

   Additional feature:

   'top-k' phrases can also be generated specific to phrase word lengths.

   For e.g. we can extract from the dataset 'top-k' phrases having only 3 words or top-k phrases with 2 words only.

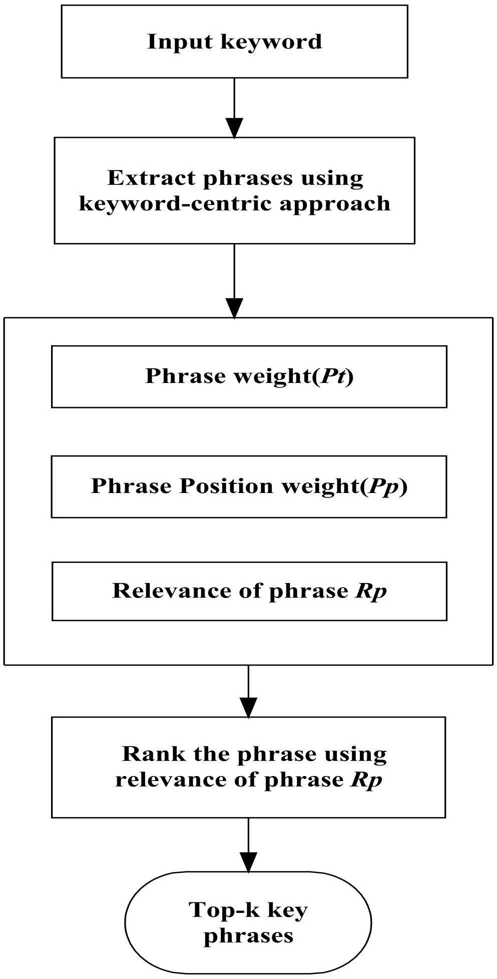

Input keyword
Extract phrases using keyword-centric approach
Phrase weight(*Pt*)
Phrase Position weight(*Pp*)
Relevance of phrase *Rp*
Rank the phrase using relevance of phrase *Rp*
Top-k key phrases

Figure 4.4: A flow diagram on key phrases generation based on keywords

Chapter 5: Database and Software

All the algorithms designed for data cleansing, keyword extraction and key phrases generation is implemented in Oracle database, version 11g.The programming language used for implementing the algorithms is known as PLSQL, which is a procedural language integrated with Oracle databases.

The interface tool used to interact with the Oracle database is Oracle SQL Developer. This is a free tool provided by Oracle Corporation for manipulating the Oracle databases. All the tables and programs for this implementation was created using this tool

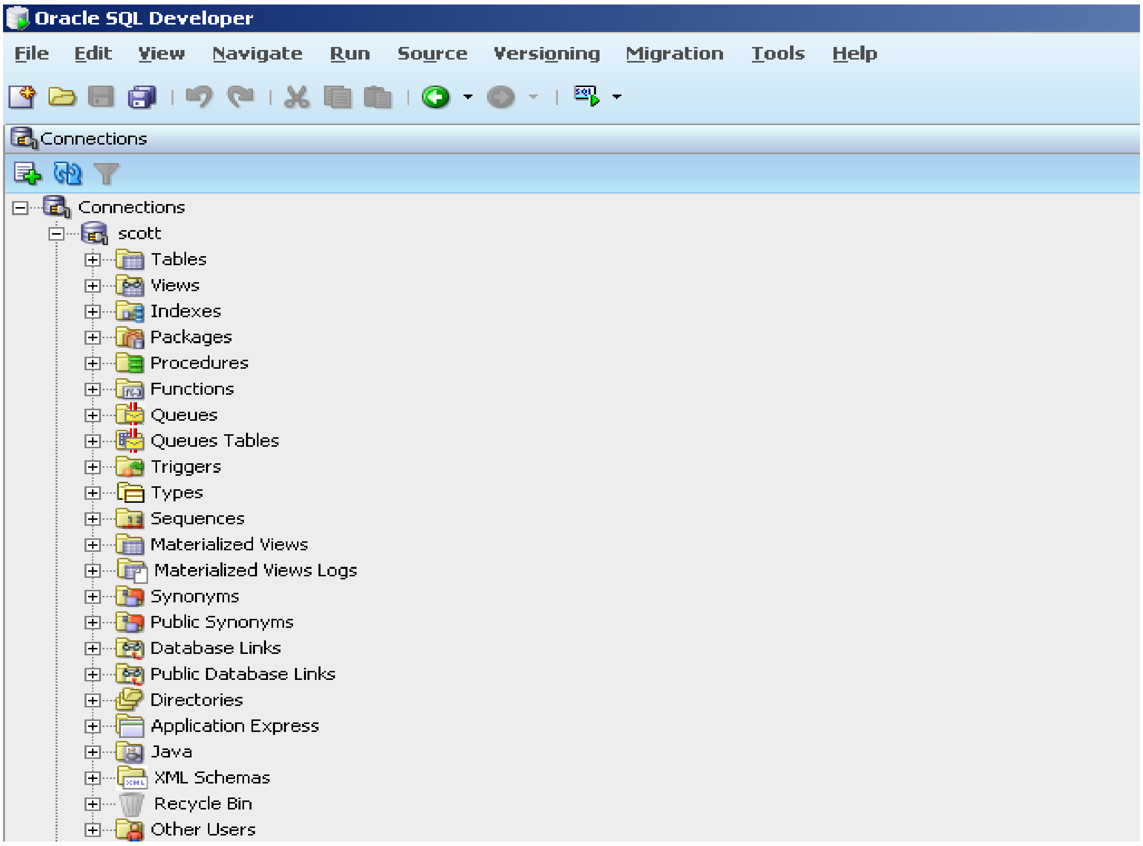


Figure 5.1: Tool used - Oracle SQL Developer

Chapter 6: Results

The algorithm designed to cleanse the attributes gave the results as detailed in table 6.1. Out of the data cleansing algorithms, the algorithm for attribute ‘location’ was the most efficient and the algorithm for ‘rent’ was the least efficient. A reason for the least efficiency for algorithm on attribute ‘rent’ is due to the fact the many of the TITLE and DESCRIPTION didn’t have the rent in them. As this data set had separate fields for entering rent value, the user who posted the classifieds didn’t always mentioned the rent in title or description of the classified.

	Total classifieds	Classifieds cleansed	Algorithm Efficiency (%)
Attribute: location	46475	43435	93.46
Attribute: beds	46475	40856	87.91
Attribute: rent	46475	21109	45.42

Table 6.1: Statistics of data cleansing

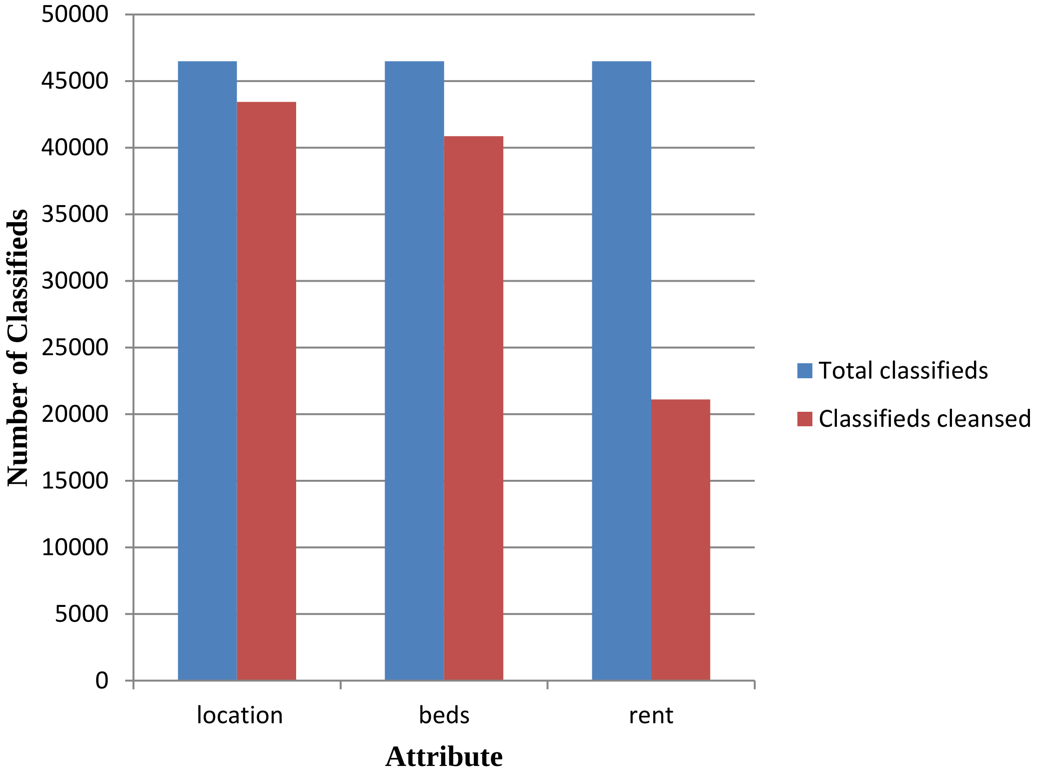


Figure 6.1: Graphical representation of statistics on data cleansing

Our focus for keyword extraction was to have a cleansed dataset that includes classifieds cleansed for all the three attributes of ‘location’, ‘beds’ and ‘rent’. Since only 45% of the classifieds were cleansed for attribute rent, the overall classifieds that became part of the cleansed data set were low in number compared to the total classifieds in original dataset. An analysis for the same is shown in figure 6.2

Total classifieds	Classifieds cleansed for location, beds and rent
46475	17733

Table 6.2: Statistics for classifieds cleansed for location, beds & rent

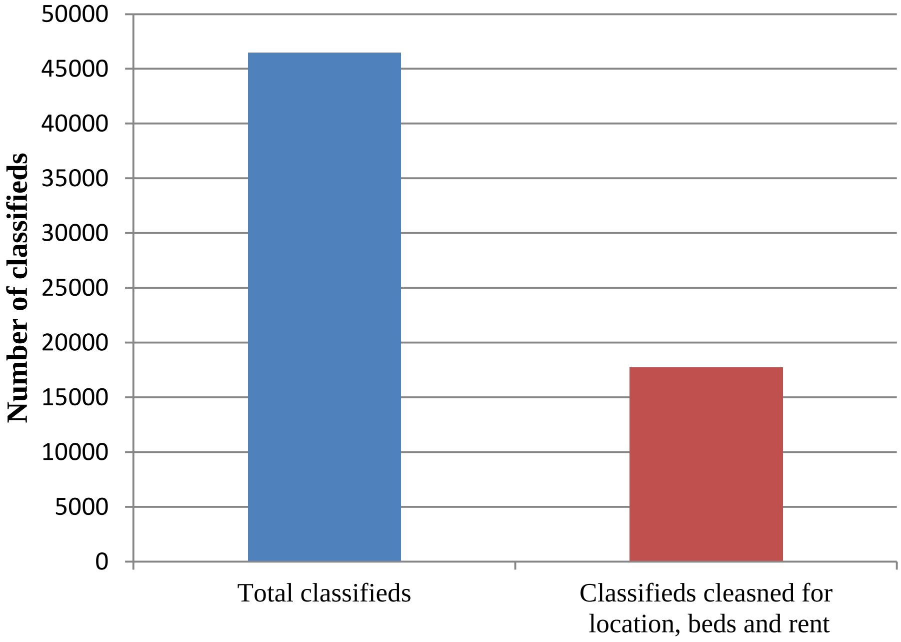


Figure 6.2: Graphical statistics for classifieds cleansed for location, beds & rent

Comparative keyword weights of top 20 keywords in above and below median cases are depicted in figure 6.3 and 6.4.

In figure 6.3, we have the keywords extracted from the cleaned dataset that are relevant to classifieds in above median case.

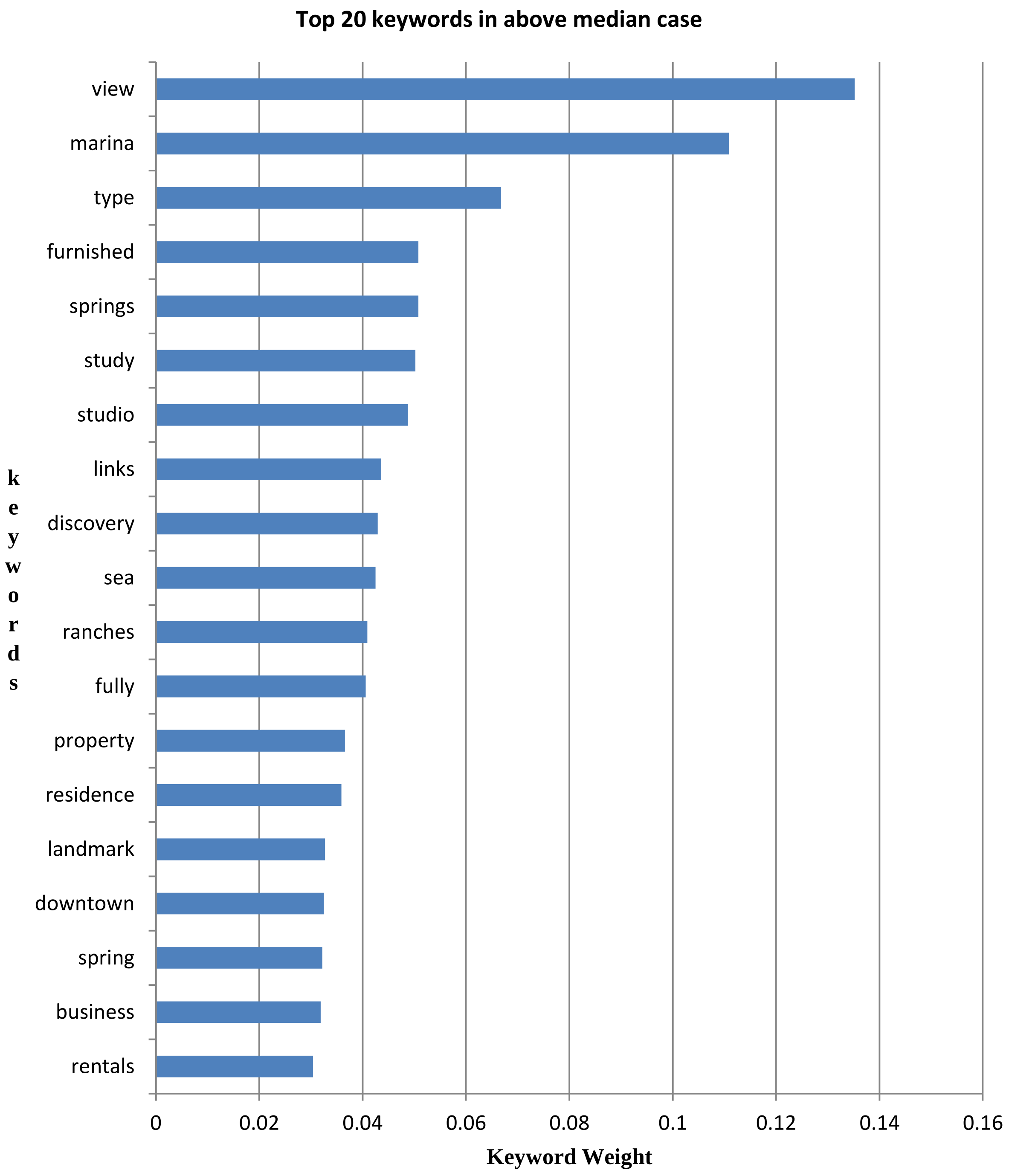


Figure 6.3: Top 20 keywords and their weights in above median case

In figure 6.4, we have the keywords extracted from the cleaned dataset that are relevant to classifieds in below median case.

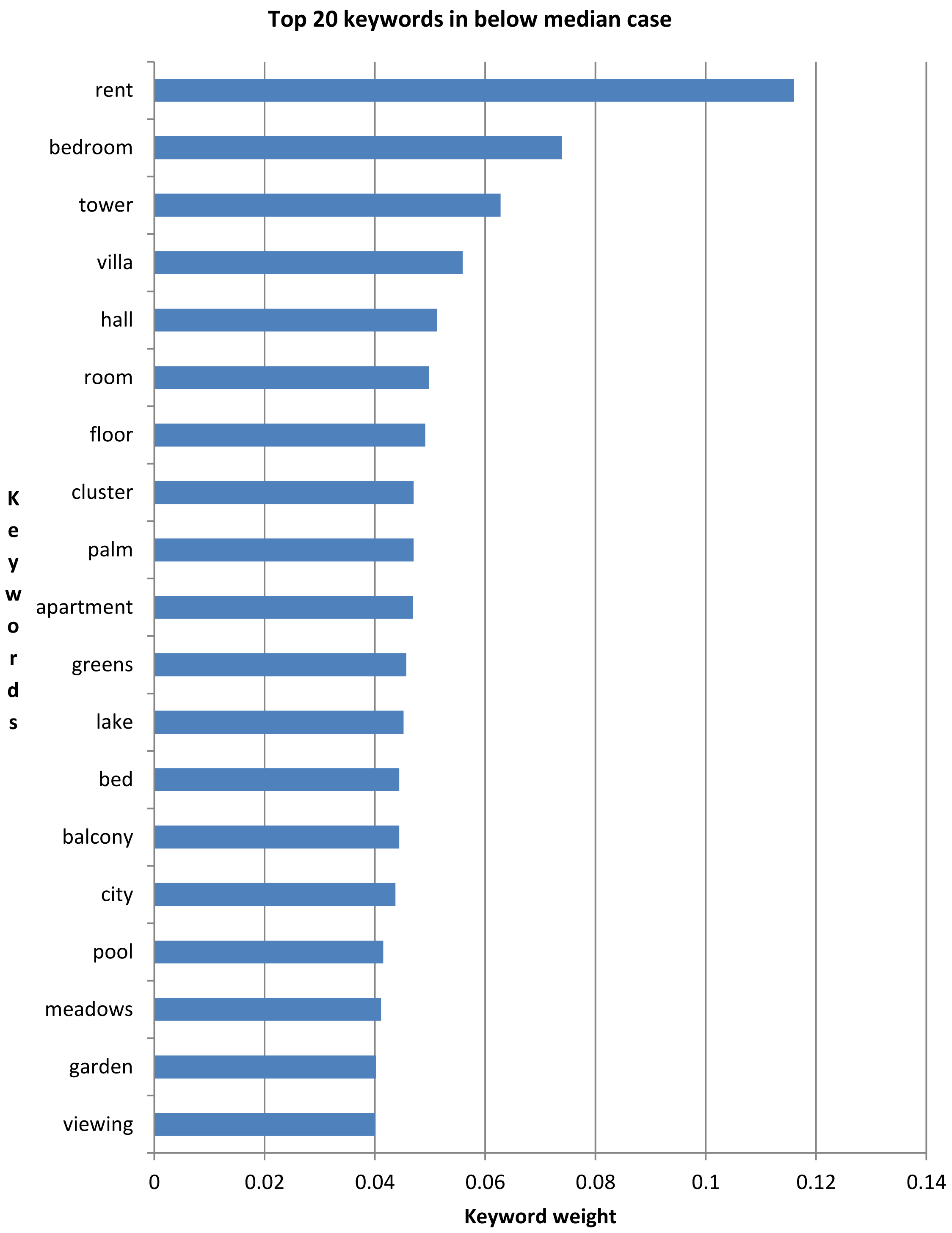


Figure 6.4: Top 20 keywords and their weights in below median case

For illustration of result on key phrase extraction, top-20 key phrases consisting of only three words extracted based on keyword 'view' is shown in figure 6.5. As from figure 6.3

above, ‘view’ is a keyword that belongs to above median case. Therefore, the top-20 key phrases for ‘view’ is extracted only from classifieds categorized as above median.

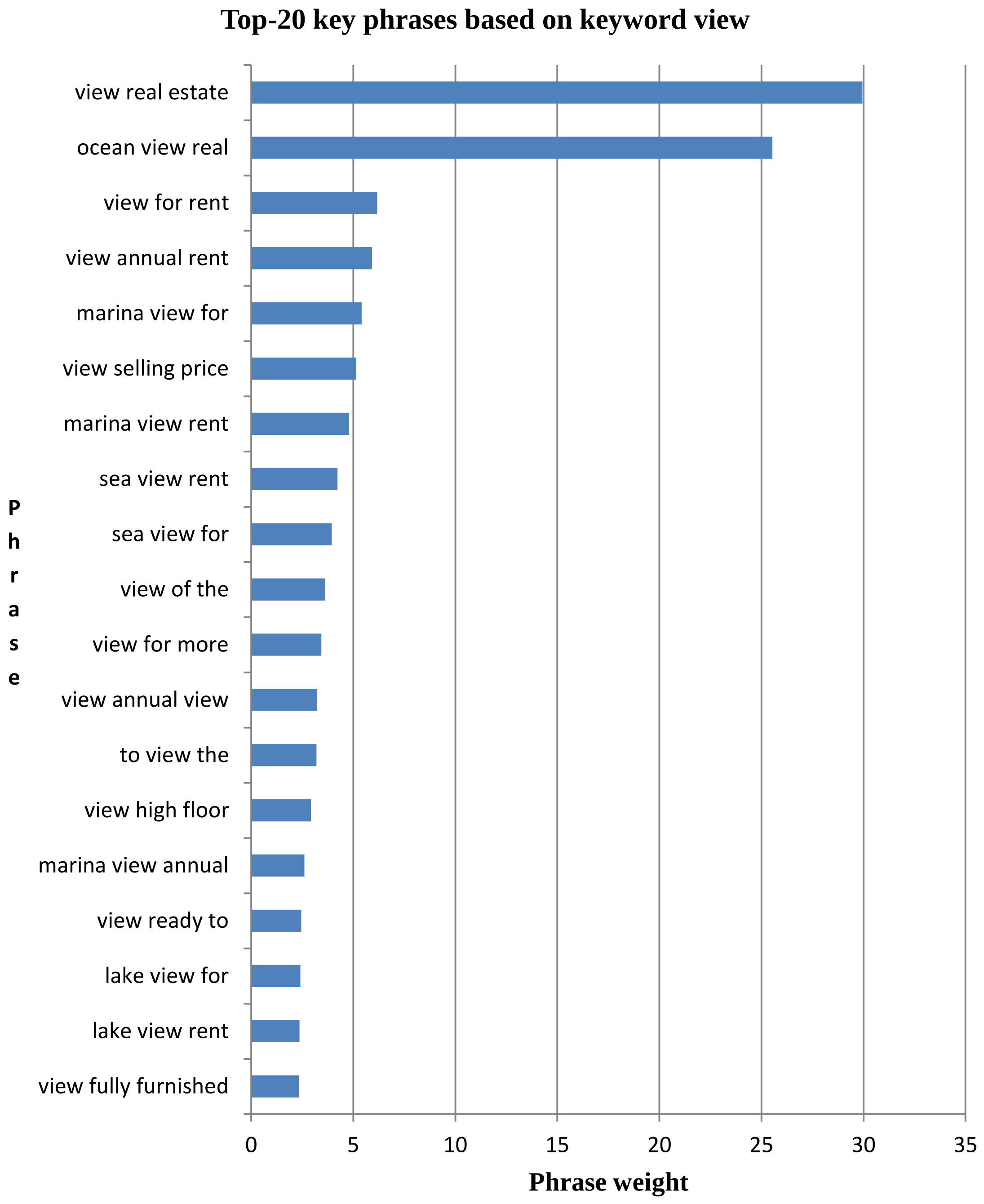


Figure 6.5: Top 20 key phrases based on keyword ‘view’

Observation:

In the set of keywords obtained a high prominence is seen towards keywords from a particular location.

For instance, the second keyword in the top keyword list (figure 6.3) is ‘marina’. As far as the dataset is concerned, ‘marina’ is name of location. From figure 6.6, it can be verified that a large number of candidate keywords are from one particular location i.e. ‘dubai marina’.

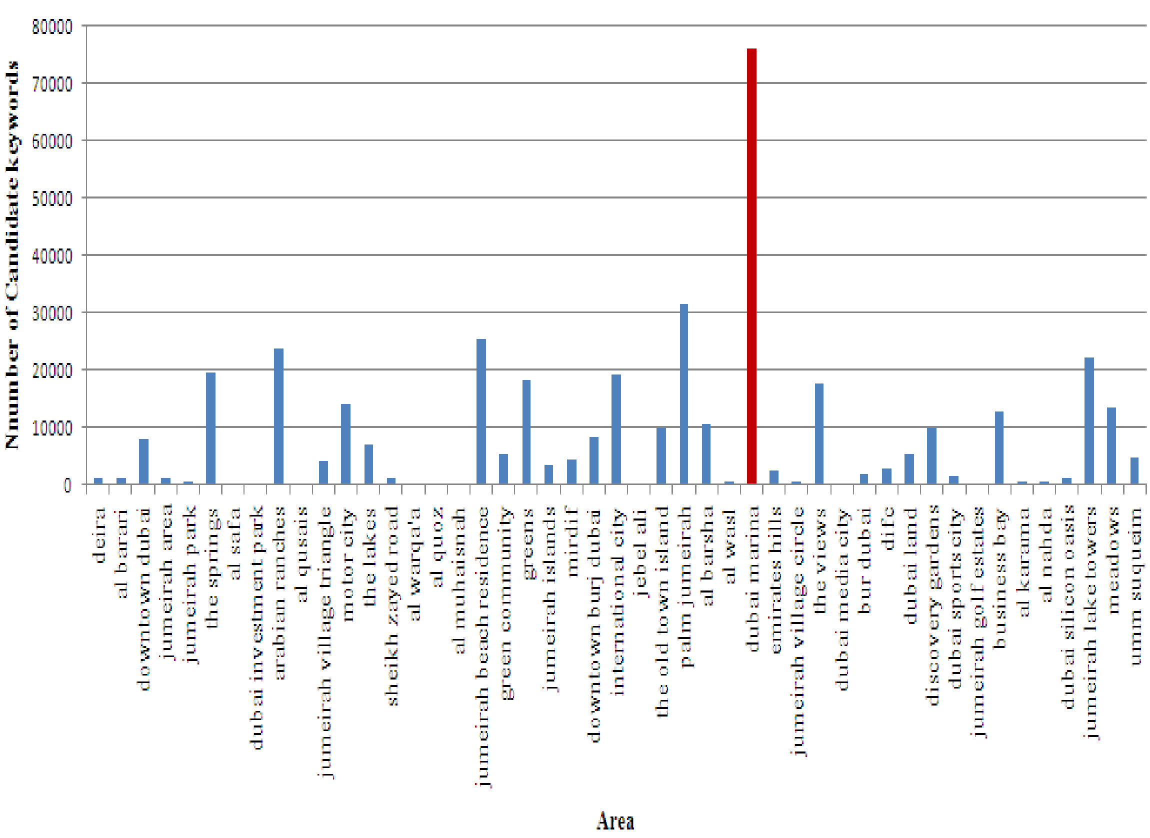


Figure 6.6: Area wise distribution of candidate keywords

Chapter 7: Recommendation

Taking into account the statistics observed in figure 6.6, there are two factors that have come into play for a keyword bias to attribute location:

1. The dataset considered had comparatively higher number of classifieds from a particular location.

2. On the other hand, the classifieds extracted after data cleansing had relatively higher number of classifieds for an area.

Therefore, focusing on a single data set to find the keywords might end up in extracting keywords that may be biased to a specific characteristic of the data set.

Considering a broader perspective, to overcome this situation a suggestion is to introduce a feed-forward approach (figure 7.1). A feed-forward approach could be used for keyword extraction by using multiple datasets belonging to different time frames. This means that the keyword extraction algorithm is fed with new instances of the data set, from the database, at periodic intervals e.g. monthly or quarterly basis. Further, it compares the current keywords with the newly generated keywords from new instance of dataset and does a keyword refresh to make available the new set of keywords. This will keep the keyword list constantly updated and also reduce potential bias to any specific characteristic of the dataset, as faced in this project.

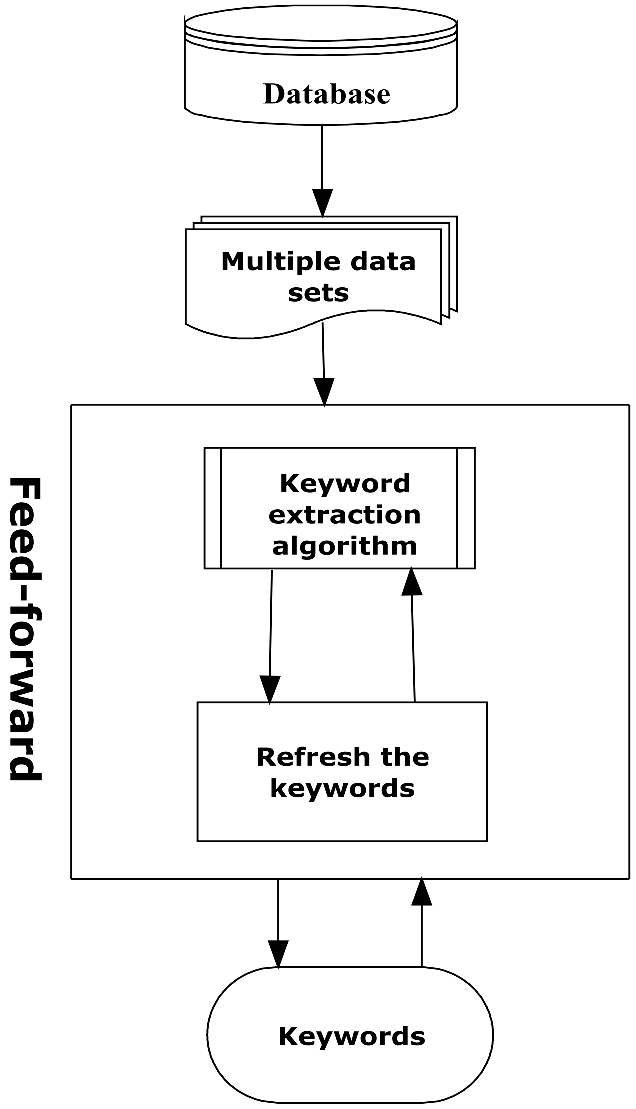


Figure 7.1: Concept of a feed-forward approach for keyword extraction

Chapter 8: Conclusion & Future Work

As the achievement of this project, we are able to automatically extract keywords from a given real estate classifieds data set, by dividing the data set into above and below median cases based on attribute ‘rent’.

While developing a design to extract the keywords from the real estate dataset, we integrated the principle of interactive data mining along with the keyword extraction methods. The principle of interactive data mining is applied in formulating data cleansing algorithms and the techniques of assigning weights to the words and phrases are applied to extract keywords and key phrases. As a result we are able to frame a method to generate keywords and key phrases, specific to the given dataset.

A disadvantage found in the keyword extraction method implemented is that, certain keywords extracted are biased towards a specific characteristic of the data set. Looking into the recommendations stated in previous section, it is also understood that keywords generated based on a single dataset can often have loop holes which can make the keyword generation biased. It is also comprehended that keywords are not static and they might evolve with the databases and therefore we cannot have keyword generation as a fixed process.

In this project we have focused on the attribute ‘rent’ as a medium to extract the keywords from real estate classifieds. In a broader view, this concept of keyword extraction based on an attribute of the data set can be applied to other domains, for example:

- extract keywords in car sale or rental classifieds based on attribute mileage
- extract keywords in job classifieds based on attribute salary.

References

Chapman, A.D. (2005). Principles and methods of data cleaning – primary species and species occurrence data. *Australian Biodiversity Information Services* [online]. July, pp. 1-72, [Accessed 5 September 2011]. Available at http://imsgbif.gbif.org/CMS_ORC/?doc_id=1262&download=1

Chen, Y.Y., Fong, O.M., Yong, S.P. and Iwan, K. (2000). Text summarization for oil and gas drilling topic. *World Academy of Science, Engineering and Technology* [online]. no. 42, pp. 31-34, [Accessed 22 October 2011]. Available at http://www.waset.org/journals/waset/v42/v42-6.pdf

Fayyad, U., Piatetsky-Shapiro, G. & Smyth, P. (1996). From data mining to knowledge discovery in databases. *American Association or Artificial Intelligence* [online]. September, pp. 37-54, [Accessed 7 August 2011]. Available at http://www.kdnuggets.com/gpspubs/aimag-kdd-overview-1996-Fayyad.pdf

Hand, D., Mannila, H. & Smyth, P. (2001). *Principles of Data Mining*. Massachusetts: The MIT Press.

Hou, J. & Chan, C. (2003). A document content extraction model using keyword correlation analysis. *International Journal of Electronic Business Management* [online]. vol. 1, no. 1, pp. 54-62, [Accessed 24 August 2011]. Available at http://ijebm.ie.nthu.edu.tw/IJEBM_Web/IJEBM_static/Paper-V1_%20N1/08.pdf

Hulth, A. (2003). Improved automatic keyword extraction given more linguistic knowledge. *Proceedings of the 2003 Conference on Empirical Methods in NLP* [online]. pp. 215-223, [Accessed 30 July 2011]. Available at http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.167.3848

Kaplan, R.M. (2005). A method for tokenizing text. *In Festschrift in Honor of Kimmo Koskenniemi's 60th anniversary* [online]. pp. 55-64, [Accessed 19 July 2011]. Available at http://cslipublications.stanford.edu/koskenniemi-festschrift/6-kaplan.pdf

Maletic, J.I., & Marcus, A. (2000). Data cleansing: beyond data integrity analysis. *IQ2000* [online]. June, pp. 1-10, [Accessed 30 September 2011]. Available at http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.37.5212

Maletic, J.I. & Marcus, A. (2005). Data cleaning – a prelude to knowledge discovery. *Data Mining and Knowledge Discovery Handbook* [online]. pp. 21-36, [Accessed 8 August 2011]. Available at http://www.cs.kent.edu/~jmaletic/papers/data-cleansing.pdf

McCargar, V. (2004). Statistical approach to automatic text summarization. *Bulletin of the American Society for Information Science and Technology* [online]. April, pp. 21-25, [Accessed 21 October 2011]. Available at https://asis.org/Bulletin/Apr-04/mcargar.html

Oracle. (2009). *Oracle® Database PL/SQL Language Reference 11g Release 1 (11.1) [online]*. [Accessed 01 October 2011]. Available at: http://download.oracle.com/docs/cd/B28359_01/appdev.111/b28370/toc.htm

Viji, S. (2002). Term and Document Correlation and Visualization for a set of Documents. *Text Information Retrieval, Mining, and Exploitation Fall 2002* [online]. September, [Accessed 28 August 2011]. Available at http://www.stanford.edu/class/archive/cs/cs276a/cs276a.1032/projects/reports/

Zhao, Y. & Yao, Y. (2005). On interactive data mining. *Proceedings of the Second Indian International Conference on Artificial Intelligence* [online]. pp. 2444-2454, [Accessed 30 July 2011]. Available at http://www.google.ae/url?sa=t&rct=j&q=interactive%20data%20mining&source=web&cd=3&ved=0CDIQFjAC&url=http%3A%2F%2Fciteseerx.ist.psu.edu%2Fviewdoc%2Fdownload%3Fdoi%3D10.1.1.91.9747%26rep%3Drep1%26type%3Dpdf&ei=VcPBTv2MMMu38gPjnrW0BA&usg=AFQjCNHX8pO9x0T3L_pUXzpC9JC6iwsYLQ