Spring 1-1-2020

Property Recommendation System With Geospatial Data Analytics Andnatural Language Processing For Urban Land Use

Sean K. Riehl
Cleveland State University

PROPERTY RECOMMENDATION SYSTEM WITH GEOSPATIAL DATA ANALYTICS ANDNATURAL LANGUAGE PROCESSING FOR URBAN LAND USE

SEAN K. RIEHL

Bachelor of Science in Computer Engineering

Cleveland State University

May 2016

submitted in partial fulfillment of requirements for the degree

MASTER OF SCIENCE IN SOFTWARE ENGINEERING

at the

CLEVELAND STATE UNIVERSITY

May 2020

We hereby approve this thesis for

SEAN K. RIEHL

Candidate for the Master of Science in Computer Engineering degree for the

Department of ELECTRICAL AND COMPUTER ENGINEERING

and the

CLEVELAND STATE UNIVERSITY'S

College of Graduate Studies by

_____

Committee Chairperson, Dr. Sunnie Chung

_____

Department/Date

_____

Committee Member, Dr. Yongjian Fu

_____

Department/Date

_____

Committee Member, Dr. Pong Chu

_____

Department/Date

Student's Date of Defense: May 7, 2020

DEDICATION

I would like to dedicate this work to my thesis advisor Dr. Sunnie Chung. Her knowledge, advice, recommendations, weekly meetings, check-ins, and continued faith in my abilities has led to the completion of this thesis and it would never have been possible without her. I would also like to dedicate this to my parents Ken and Kathy Riehl as without their continued love, support, and encouragement I would have never thought it would be possible for me to complete my thesis in big data analytics.

PROPERTY RECOMMENDATION SYSTEM

WITH GEOSPATIAL DATA ANALYTICS AND

NATURAL LANGUAGE PROCESSING FOR URBAN LAND USE

SEAN K. RIEHL

ABSTRACT

Recently Cuyahoga County has been tremendously improved as properties are being constructed, renovated, or altered for new land use transactions on a nearly daily basis. Most existing property recommendation systems for the area simply rely on surface-level information and user history data to produce recommendations while failing to prioritize factors according to their importance and utilizing the location based complex information efficiently. This is leading them to become stagnant and simplistic in their approach and their accuracy is worsening as there are too many factors to be considered and location based complex yet useful information such as land use aspects of neighboring areas or information about people who are living or working in the area are often hard to be discovered. To combat these issues, this thesis proposes a modern property recommendation system with new approaches: 1) Employing data analytic methods to discover complex location based geospatial knowledge from big data processing, 2) Collecting and deriving summary information on people demographic data in the neighbor, and 3) Adopting natural language processing techniques for a user given phrase query to generate accurate candidate sets. Our recommendation system consists of three key components: 1) Using derived geospatial knowledge as new features and viewpoints for a better overall understanding of neighbor for a given property. 2) Incorporating Hotspot Analysis and data analytic methods to identify which areas are the

most ideal for each type of properties based on current and history data. 3) Allowing a user query in a sentence or phrase through natural language text processing techniques to create accurate candidates to tailor recommendations to a given individual user to return the Top-*N* ranked results. The experimental results show the effectiveness of these new approaches.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1.Background

Recommendation systems can effectively aid users in filtering down potential products into only the most personalized results. However, as the volume and complexity of the data grows, so does the need to make the recommendation systems more sophisticated and adaptive. Today the traditional recommendation systems mainly follow one of two patterns: memory-based and model-based collaborative filtering algorithms that mainly rely on user purchase history data [38].

Memory-based collaborative filtering algorithms [38] are the more popular method of the two and are widely adopted in commercial systems. They are subdivided into two smaller types called user-based and item-based approaches. A user-based system [38] works by calculating the similarity between users based on given data, such as their recent purchase history. The methodology behind this is that if two users purchased the same product, then they are likely similar users who would purchase the same products. Conversely, an item-based system [38] works in reverse. It starts by comparing items based on which users have recently purchased them. In a comparable fashion, any time a user purchases the item, the similar items are recommended with the similarity being

calculated using either the cosine similarity function or the Pearson correlation coefficient.

The other approach is a model-based collaborative filtering algorithm [38]. It uses computed models prebuilt using Machine Learning algorithms like Bayesian networks [12], clustering models [3], and latent semantic models [37]. Their goal is to improve the core models through machine learning techniques by deriving training sets from the user history data or item input data. One way this tactic produces results is by attempting to plot and group the data into similar areas based on one or more dimensions. At its core, the methodologies behind this is that plotted points in similar area represent the same type of data so they should belong to the same group.

1.2.Motivation

Today the traditional recommendation systems mainly follow one of two patterns: memory-based and model-based collaborative filtering algorithms. Most of these recommendation system algorithms rely on past user purchase history data to tailor their results to the specific needs of the user. When new users are presented, the models must rely on popular trends to make educated guesses as to what the new user may enjoy while the system slowly builds up a personalized collection. This leads to poor recommendations for users who are either new or more unusual and not into the latest movements [38]. Moreover, these traditional approaches are not accurate to generate recommendations for complex items simply because they do not effectively consider complex characteristics of information, which are often hidden on each item as the volume and complexity of the data grows. Therefore, the motivation of this thesis is to create a recommendation system more sophisticated and adaptive using big data analytic

techniques that can effectively derive hidden knowledge on item data for consideration without relying on past user history data to generate worthwhile recommendations for all types of users.

1.3.Problem Statement

This thesis is to build an urban property recommendation system using big data collected from many different public government data resource sites of real estate property data and census data in Cuyahoga County. The recommendation system presented in this study is an item-to-item collaborative filtering algorithm which does not need to use user history data. Instead, the system uses derived information obtained from a user-given query sentence using natural language processing techniques to effectively consider complex geospatial information and location-based knowledge that is derived from various big data analytics methods.

An immediate challenge starts from the differences between the complex and various big data types coming from the collected raw data and the uniform structured data format required for data analytic algorithms when trying to integrate the information into a single collective. Given the mass amount of distributed and unstructured GPS and GeoJSON coordinate data, for example, deep learning algorithms cannot effectively and efficiently analyze and respond to complex end user requirements.

Furthermore, many of the key factors in determining outcomes are not laid out plainly but instead, need to be derived through preprocessing and big data analytic techniques. The natural language queries provided by users also fails to help simplify matters since the meaning of those queries are often difficult for computers to understand. These systems need to be able to recognize and react to meaningful patterns hidden

within the natural language data by creating correlations through text mining techniques. However, without a proper way for the user to visualize the results, the rest is pointless as there is no point in generating recommendations if there is no one to accept them. The development of an algorithm that incorporates big data preprocessing methods, text analysis strategies, and derived variables while overcoming said trials is needed.

1.4.Objectives

The objectives of this thesis are to develop a recommendation system with capabilities to effectively derive hidden knowledge on urban property data using big data analytic techniques without relying on past user history data to generate worthwhile recommendations for all types of users. More precisely, the objectives of this thesis is to develop methods to build a urban property recommendation system using data analytic techniques to derive geospatial information and location-based knowledge of neighbors of each property and employing natural language text analysis to generate accurate candidates to consider for a user-given query sentence to provide end users with accurate and meaningful results. It will be able to handle a vast amount of different data types, process all the information in real time, and calculate complex analysis in memory-efficient ways. Additionally, this system will be scalable to process with extended data sets over the entire US.

Here is the summary of contributions of this study in the literature recommendation system research:

- Employing big data preprocessing techniques to integrate geospatial data and geographical neighborhood demographics from heterogeneous and complex big data resources

- Applying big data analytic methods to discover comprehensive information of geospatial data to consider in the recommendation system
- Allow a user query in a phrase or sentence with Natural language processing techniques to derive user preference related features to be considered in the recommendation system

CHAPTER II

RELATED WORK

In this chapter, literature reviews on approaches for recommendation systems will be summarized and discussed. Starting with review of traditional algorithms, the chapter moves toward to introduce and discuss recent and new approaches in the literature.

Amazon is the go-to site for online shopping and their popularity has led to a website with millions of users and hundreds of millions of products. This meant a big data scale that the then-current recommendation algorithms could not handle since the process of comparing the products meant comparing hundreds of millions of entries. Therefore, after testing other options Amazon developed a new algorithm that would meet their needs called the item-to-item collaborative filtering algorithm [38]. It is capable of scaling upwards to handle the product corpus while being able to produce high-quality recommendations in real time thanks to a unique approach. Namely, it first takes the user's purchases and rated items and matches them with similar items. To determine which items are considered similar, it pairs together products that are frequently purchased together, calculates the cosine similarity, and then saves that information into an offline recommendation list. Doing this offline reduces the processing time and memory usage costs considerably.

When it comes to recommendations, it is common practice to ask friends or family members for suggestions so much so that we trust their opinions more so than ones created by computer algorithms. Because of this, Ma et al [39] decided to incorporate the interaction between users as an input factor in their matrix-based social recommendation SoRec system. Matrix factorization is the process of deriving two smaller matrices from a larger product one. This technique is used primarily because it can save memory since only the factors need to be stored. Additionally, missing values can be derived using similarity techniques. Moreover, by incorporating a social network graph by first transforming it into a matrix, SoRec was able to transpose and multiply their user-item matrix against it to create a significantly larger matrix. From that result relationships could be inferred that involved the social network between users as well as the relationships between users and products so recommendations could be provided.

In the industry, collaborative filtering and the matrix factorization methods are the most widely used do their versatility and speed when handling big data depositories. However, these papers differ from the approach in this thesis in that they rely on historic user data to provide recommendations using user-item relationships. Another difference is that their simple approach relies on user reviews to measure relevancy among items and users ignoring complex characteristics of items and location-based information which can be discovered from other related data.

YouTube is the world's largest platform for video content, and it is home to billions of users viewing millions of videos. To produce worthwhile recommendations, they created a recommendation system that can produce quick and highly relevant content for any user. The recommendation system behind YouTube [11] consists of two

deep neural networks, a candidate generation model and a ranking model. The candidate generation model has the job of sampling the millions of videos and narrowing them down to only hundreds by creating watch and search vectors like a bag of words text analysis model. From there, the vectors are concatenated together with other variables such as the user's geographical information, age, and gender and passed to the first neural network. Next the candidates are further reduced using the nearest neighbor algorithm and the remaining videos are passed on to the ranking model. This system's job is to maximize the watch time by using weighted logistic training and to return the highest ranked videos to the end user.

Social media has allowed users to generate short messages, such as tweets, to describe their feelings about locations or events. However, these user-generated short texts (UGSTs) are rarely geocoded. Therefore, by using text analysis in conjunction with geospatial information, Deng et al [14] set out to develop an algorithm that can accurately couple the UGSTs with a geographical location. They started by collecting UGSTs and breaking them down into tokens, stemming them, removing any stop words, and saving the token strings as vectors. The tokens were further preprocessed by being saved as entities as well since they contain more semantic information than the individual words. Next they gathered geotagged location information from Foursquare and built a probabilistic model from it and the vectors. The idea was to build up a depository of terms and entities associated with each location so that the UGST vectors could then be compared to it. For example, entities like "Big Apple" were associated strongly with New York City. Move over, the pairings with the highest weights indicated the strongest

couplings and from that they were able to make accurate deductions as to the locations of the UGSTs without geotags.

Text analysis is generally solely dependent on literal statistical information from texts to determine the importance and associated weight of each term in each natural language user query. Knowing this, Tan et al [55] developed a method to integrate user intentions into the formula when recommending phone apps by using an attention-based gated recurrent unit recurrent neural network (GRU-RNN). GRU-RNNs are a step above neural networks because they do not suffer from the vanishing gradient problem. As a neural network is running, the gradients, the values used to update the network weights, shrink as it back propagates though time. This causes the earliest terms in the input sequence to be assigned extremely small weights and thus they are deemed unimportant. GRU-RNNs solve this problem using internal gates which regulates the flow of information. They learn which data is important and filter out the rest. Tan et al were able to apply this concept to natural language text queries for phone apps to determine which words should hold the most weight and then they compared that to phone app reviews to provide more accurate recommendations.

CHAPTER III

DESCRIPTION AND PROCESSING OF BIG DATA

3.1.Big Data

Big data is defined based on its six properties. Variety refers to a variety of complex formats, which are often unstructured because they are stored in databases, log files, or web pages. Volume indicates the sheer size of the data, which is often in petabytes and terabytes. Velocity and Variability state the speed at which new data is generated and how it is constantly evolving with the number of inconsistencies in it. Complexity is about data transformation and the amount of work needed to clean and process the data. Lastly, Value refers to the amount of worthwhile information within the data [57].

Furthermore, it is also defined as a term for a collection of data too large and complex to easily process using traditional data preprocessing methods. The data sets are commonly associated with the challenges of capturing, curating, storing, searching, sharing, transferring, analyzing, and visualizing the data in a timely manner. In fact due to the data often coming from multiple different sources and in multiple different data formats, big data is often quite challenging to integrate into a singular model for further

analytic processing, especially when there is no straightforward connection between the datasets or when the data files are too large to open or even store on a single computer.

3.2. Collection and Description of Data

The massive raw data was collected from various government information sites and public repositories for the property recommendation system. The collected raw data is mainly categorized into two types of contents – land data and people data. The former is about the characteristics of every registered property and related code structures, referred to henceforth as Land Data, while the latter is about the demographic census information of the people either living or working within the neighboring areas, referred to henceforth as People Data. Given the time and resource constraints, a scope of this thesis is to build a property recommendation system which focuses on the data within Cuyahoga County. This recommendation system framework can be easily extended across the entire United States.

3.2.1. Land Data

The Land Data is split into three subcategories about the properties, sales, and characteristics of the spaces. The property data comes from the Cuyahoga County file transfer protocol (FTP) website in the form of Computer-Assisted Mass Appraisal (CAMA) system files and represents a status of basic ownership and land use information [13]. The data is stored in GeoJSON file format, which is a file format for encoding a variety of geographic data structures [22]. Figure 1-1 and 1-2 showcase an example of the format for one property from the property data. The sales data is sourced from the Maxine Goodman Levin College of Urban Affairs as well as from the Cuyahoga County FTP Department of Information Technology website and provides detailed information

on the sales of properties from 1976 onwards [23]. This data was sourced as comma-separated values (CSV) and came on a compact disc. The characteristics data comes from tax assessments every six years by the Cuyahoga County Fiscal Office and covers details of the structures such as the number of elevators, the number of bathrooms, and the amount of residential and commercial square footage [51]. For this thesis, based on the focus, only the characteristics data was primarily used so as such the following relevant sections will exclusively focus on it.


```
{
        "_id": {
                "$oid": "5cdfe38b5ba20a6bfb3b0871"
        },
        "type": "Feature",
        "id": 0,
        "properties": {
                "PARCELPIN": "20228021",
                "PARCEL_PK": "15253",
                "PARCEL_TYP": "LAND",
                "PARCEL_ID": "20228021",
                "BOOK_PAGE": "B 202 P 28",
                "PARCEL_YEA": 2017,
                "PARCEL_OWN": "VASIL JR., WILLIAM L.",
                "DEEDED_OWN": "VASIL JR., WILLIAM L.",
                "GRANTOR": "FEDERAL HOME LOAN MORTGAGE CORPORATION",
                "GRANTEE": "VASIL JR., WILLIAM L.",
                "TRANSFER_D": "2014/07/18",
                "SALES_AMOU": 0,
                "PAR_ADDR": "28326",
                "PAR_STREET": "WEST OAKLAND",
                "PAR_SUFFIX": "RD",
                "PAR_CITY": "BAY VILLAGE",
                "PAR_ZIP": "44140",
                "PAR_ADDR_A": "28326 WEST OAKLAND RD, BAY VILLAGE, OH, 44140",
                "MAIL_NAME": "VASIL JR., WILLIAM L.",
                "MAIL_ADDR_": "2231 HOLLY  LN",
                "MAIL_CITY": "AVON",
                "MAIL_STATE": "OH",
                "MAIL_ZIP": "44011",
                "MAIL_COUNT": "USA",
                "TAX_LUC": "5100",
                "TAX_LUC_DE": "1-FAMILY PLATTED LOT",
                "ZONING_USE": "1F-3",
                "PROPERTY_C": "R",
                "TAX_DISTRI": "050",
                "NEIGHBORHO": "03112",
```


Figure 1-1: Land Data for One Property in GeoJSON Format

```
                "ROAD_TYPE": "PV",
                "WATER": "MUN",
                "SEWER": "SNS",
                "GAS": "Y",
                "ELECTRICIT": "Y",
                "TAX_YEAR": 2016,
                "CERT1": 139,
                "CERT2": 50500,
                "CERT3": 160100,
                "CERT4": 210600,
                "CERT6": 0,
                "CERT7": 0,
                "CERT8": 0,
                "CERT10": 0,
                "CERT11": 0,
                "CERT12": 0,
                "GCERT1": 50500,
                "GCERT2": 160100,
                "GCERT3": 210600,
                "RES_BLDG_C": 1,
                "TOTAL_RES_": 2087,
                "TOTAL_RES1": 7,
                "COM_BLDG_C": 0,
                "TOTAL_COM_": 0,
                "COM_LIVING": 0,
                "TOTAL_LEGA": 75,
                "TOTAL_SQUA": 17250,
                "TOTAL_ACRE": 0.396,
                "OurCode": "R1",
                "SiteCat1": "Residential",
                "SiteCat2": "Single Family",
                "Descrip": "1-FAMILY PLATTED LOT",
                "SPA_NAME": "NULL",
                "PAR_CITY2": "NULL",
                "Units": 1,
                "Units2": 1,
                "PARCL_OWN2": "vasil jr., william l.",
                "PARCL_OWN3": "vasil jr., william l.",
                "MAIL2": "2231 holly ln avon",
                "PAREN2": "306066",
                "SPA_COD": "000000",
                "PARCELLOC": "39035bayage00000020228021"
        },
        "geometry": {
                "type": "Polygon",
                "coordinates": [
                        [
                                [-81.9332,41.4843],
                                [-81.9332,41.4837],
                                [-81.9335,41.4837],
                                [-81.9335,41.4843]
                        ]
                ]
        }
}
```


Figure 1-2: Land Data for One Property in GeoJSON Format (continued)

The land use data, which will be referred to as Land Use Data here after, is divided based on the characteristics of the particular property with the properties themselves being divided into parcels of land [20]. These parcels serve as the unique identifier for the roughly 550,000 properties across Cuyahoga County. Furthermore, the data is split across seven characteristic files based roughly on the taxable land use, such as residential, industrial, and commercial use, with each file corresponding to a different aspect and only containing information of a parcel when appropriate. For example, the Land Use Data file about residential land use does not contain any information about a commercial industry. The other files store data on the parcel record, historic changes, land description, and commercial and industrial usage. Across all the files every property is thoroughly detailed using hundreds of variables of various types such as integers, strings, years, dates, and other identifiers with each having a parcel variable column to connect them. Some of the variables of the Land Use Data from the residential data file are documented in Table 1.

Table 1. Partial Information from the Residential Land Use Data

Parcel	Link ID	Update Date	Occupied	Style	Stories	Quality	Year Built
101001	8097172	9/2/2004	1	RAN	1	B	1951
101004	8097174	2/8/2006	1	CAP	1.5	A+	1957
101005	8097175	1/19/2012	1	CAP	1.5	A+	1953
101006	8097176	2/8/2006	1	COL	2	AA	1928
101007	8097177	2/8/2006	1	COL	2	AA	1927
⋮							
99122088	8643109	11/16/2011	1	BUN	1.5	C	1900
99122088	8643109	11/16/2011	1	BUN	1.5	C	1900
99122089	8643110	7/12/2005	1	BUN	1.5	C	1900
99122089	8643110	7/12/2005	1	BUN	1.5	C	1900
99122089	8643110	7/12/2005	1	BUN	1.5	C	1900

3.2.2. People Data

The People Data is provided by the United States Census Bureau and is referred to as the Longitudinal Employer-Household Dynamics (LEHD) Origin-Destination Employment Statistics (LODES) [60, 61]. The files themselves are stored individually in CSV GZ file formats, which is a single file compression format based on the DEFLATE algorithm created for the GZIP project started by French software developer Jean-Loup Gailly and American software engineer Mark Adler in 1992 [68]. Examples of some of the source files can be seen below in Figure 2. These People Data files cover information about the people, their demographics, and their relationships to different areas. Despite what the name may imply, the data is not about the individual residents and workers, but rather about the groups of the individuals who reside or work in the census tracts across the county. Section 4.1. of this thesis further explains the Cuyahoga County census tracts in detail.

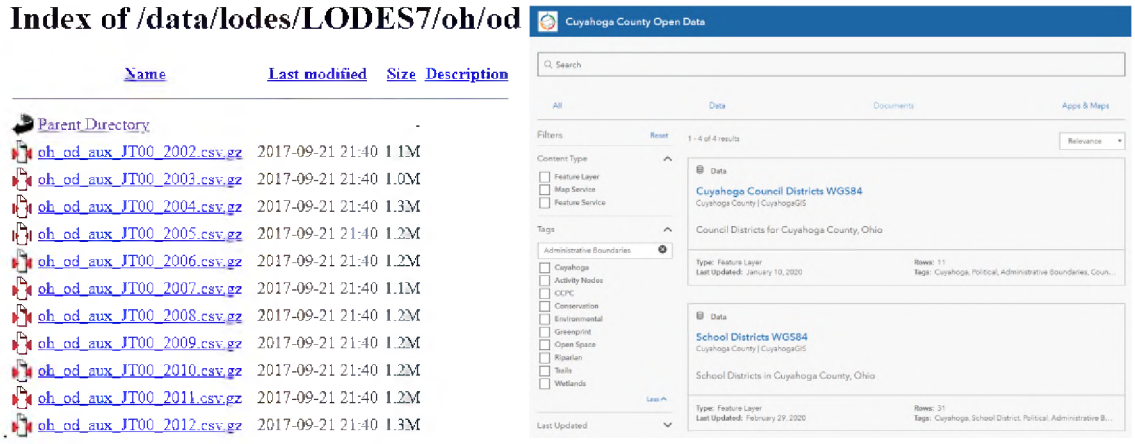


Figure 2-1: People Data Sources

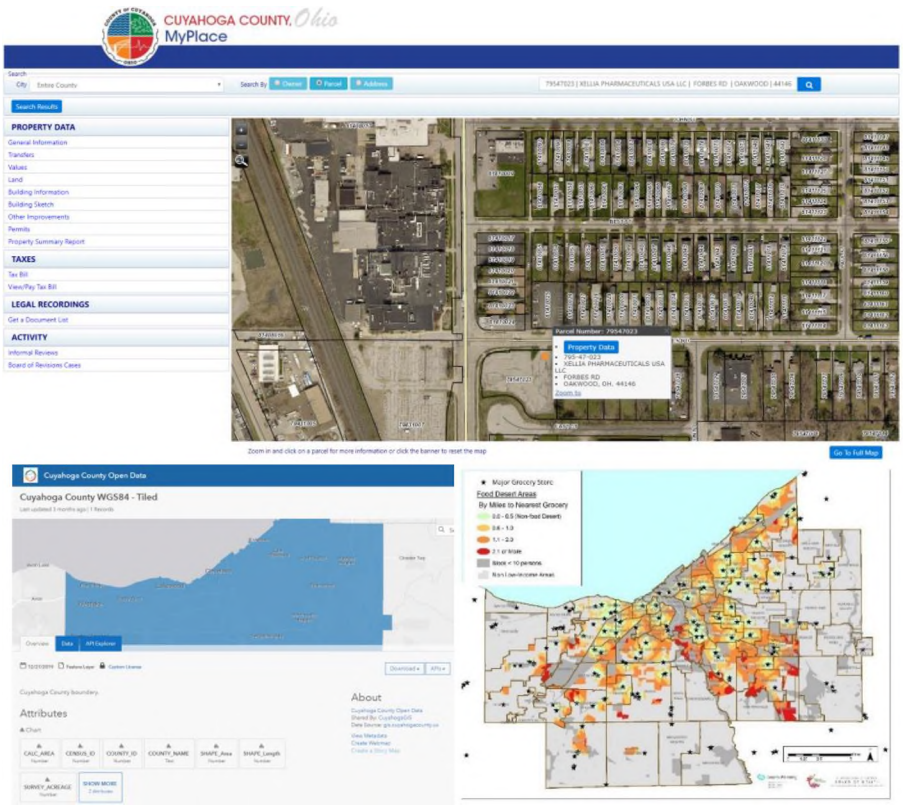


Figure 2-2: People Data Sources (continued)

The People Data is split into three subcategories named Resident Area Characteristic data (RAC), Workplace Area Characteristic data (WAC) and Origin-Destination data (OD) [61]. The subcategories each consist of thousands of individual files referring to a combination of state, subcategory, segment of the workforce, job type, and year with the relevant areas having a unique identifier in the form of a geocode. The workforce segments are divided based on age, earnings, and job sector, the job type specifies whether it is a primary job, private job, federal job, or combination of the three, and the geocode is a concatenation of the state code, county code, tract code, and sometimes the block code.

Moreover, RAC refers to occupational information for residents in an area regardless of whether the resident works in that same area or if they even work in the

county. Conversely, WAC refers to occupational information for workers in an area regardless of whether the worker lives in that same area or if they even live in the same county. A partial documentation of the WAC data can be seen in Tables 2-1 and 2-2. Finally, OD is the intersection of the RAC and the WAC datasets and contains both geocodes from the other subcategories. These three subcategories are mainly comprised of information that refers to the number of jobs in different NAICS code classifications. NAICS is an abbreviation for the North American Industry Classification System which is the standard used by federal statistical agencies in classifying business establishments for collecting, analyzing, and publishing statistical data related to the United States business economy according to the 2017 North American Industry Classification System [62]. Also included in the People Data files are the ages, genders, races, ethnicities, education levels, and income levels of the residents and workers.

Table 2-1. Partial Documentation of the WAC Data

Workplace Area Characteristics (WAC) File Structure			
Pos	**Variable**	**Type**	**Explanation**
1	w_geocode	Char15	Workplace Census Block Code
2	C000	Num	Total number of jobs
3	CA01	Num	Number of jobs for workers age 29 or younger
4	CA02	Num	Number of jobs for workers age 30 to 54
5	CA03	Num	Number of jobs for workers age 55 or older
6	CE01	Num	Number of jobs with earnings $1250/month or less
7	CE02	Num	Number of jobs with earnings $1251/month to $3333/month
8	CE03	Num	Number of jobs with earnings greater than $3333/month
9	CNS01	Num	Number of jobs in NAICS sector 11 (Agriculture, Forestry, Fishing and Hunting)
10	CNS02	Num	Number of jobs in NAICS sector 21 (Mining, Quarrying, and Oil and Gas Extraction)
11	CNS03	Num	Number of jobs in NAICS sector 22 (Utilities)
⋮			
26	CNS18	Num	Number of jobs in NAICS sector 72 (Accommodation and Food Services)
27	CNS19	Num	Number of jobs in NAICS sector 81 (Other Services [Except Public Administration])
28	CNS20	Num	Number of jobs in NAICS sector 92 (Public Administration)
29	CR01	Num	Number of jobs for workers with Race: White, Alone

Table 2-2. Partial Documentation of the WAC Data (continued)

Workplace Area Characteristics (WAC) File Structure			
Pos	**Variable**	**Type**	**Explanation**
30	CR02	Num	Number of jobs for workers with Race: Black or African American Alone
31	CR03	Num	Number of jobs for workers with Race: American Indian or Alaska Native Alone
32	CR04	Num	Number of jobs for workers with Race: Asian Alone
33	CR05	Num	Number of jobs for workers with Race: Native Hawaiian or Other Pacific Islander Alone
34	CR07	Num	Number of jobs for workers with Race: Two or More Race Groups
35	CT01	Num	Number of jobs for workers with Ethnicity: Not Hispanic or Latino
36	CT02	Num	Number of jobs for workers with Ethnicity: Hispanic or Latino
37	CD01	Num	Number of jobs for workers with Educational Attainment: Less than high school
38	CD02	Num	Number of jobs for workers with Educational Attainment: High school or equivalent, no college
39	CD03	Num	Number of jobs for workers with Educational Attainment: Some college or Associate degree
40	CD04	Num	Number of jobs for workers with Educational Attainment: Bachelor's degree or advanced degree
41	CS01	Num	Number of jobs for workers with Sex: Male
42	CS02	Num	Number of jobs for workers with Sex: Female
43	CFA01	Num	Number of jobs for workers at firms with Firm Age: 0-1 Years
44	CFA02	Num	Number of jobs for workers at firms with Firm Age: 2-3 Years
45	CFA03	Num	Number of jobs for workers at firms with Firm Age: 4-5 Years
46	CFA04	Num	Number of jobs for workers at firms with Firm Age: 6-10 Years
47	CFA05	Num	Number of jobs for workers at firms with Firm Age: 11+ Years
48	CFS01	Num	Number of jobs for workers at firms with Firm Size: 0-19 Employees
49	CFS02	Num	Number of jobs for workers at firms with Firm Size: 20-49 Employees
50	CFS03	Num	Number of jobs for workers at firms with Firm Size: 50-249 Employees
51	CFS04	Num	Number of jobs for workers at firms with Firm Size: 250-499 Employees
52	CFS05	Num	Number of jobs for workers at firms with Firm Size: 500+ Employees
53	Create Date	Char8	Date on which data was created, formatted as YYYYMMDD

3.3. Data Preprocessing

3.3.1. Land Data Preprocessing

The original seven Land Use Data files have a total of 306 different variables across them, so the first steps in preprocessing is to identify each variable. Using a characteristics appraisal inventory file [20], most of the variables can be accurately identified. The remaining ones have their meanings derived based on the file they are present in, their names, the variables they are nearby in the Excel sheets, and the values present in the columns. The next step is to cut down the 306 variables to only 190 to help with a data volume issue and because not every variable is useful for this thesis work. The final preliminary step is to look through each of the remaining variables individually and clean them up as needed. For example, the city variable from the parcel file is corrected so that no row has a missing or incorrect value. After the Land Data is cleaned, some of the variables are chosen to be further preprocessed. One of them is the lot size variable, which indicates the square footage of the parcel, and it is normalized into a weight variable for use in the ranking function.

3.3.2. People Data Preprocessing

At first the People Data consists of three folders, one for each subcategory, containing about 1,500 CSV files from the Government Data Warehouse. Each of these files only contains the combination of state, subcategory, segment of the workforce, job type, and year within the file names themselves, so each row of each file is concatenated with that information to merge them into three distinct database tables. From there the People Data variables are selected to serve as future weights in the ranking function. Some of the variables chosen are the income variables and the populations, but only for

the year 2015 to prevent repetition of the data and because the year 2015 is the most recent present year. The former variables indicate the number of people in the given tract who have monthly incomes of under $1,250, between $1,251 and $3333, and over $3333. The latter is not a variable present in the data, but one derived by calculating how many people live in each tract. All these variables are then normalized into their respective weights.

3.3.3. Integrating of Heterogenous Data Sources

With the input data preprocessed the only step that remains is to integrate the two datasets into one database, as pictured in the big data collection and geospatial information integration phase of the framework in Chapter 4. Because the properties in the Land Use Data are identified using parcel numbers while the residents and workers of the People Data are identified using geocodes, a third data source is needed to integrate the two. The parcel numbers are eight-digit identification numbers from the Cuyahoga County fiscal office and are created from the concatenation of the book number, page number, and circle number on the map [43]. However, the geocodes are fifteen-digit identification numbers that serve as the basic census geographical hierarchy and are the concatenation of the Federal Information Processing System (FIPS) state code and county code, tract code, and block code [2,7, 18]. Therefore, the only common ground between the data sets is that they both refer to Cuyahoga County. To overcome this issue, every parcel number is manually matched with its corresponding geocode using the Cuyahoga County census tract maps [63, 64, 65] and using the Cuyahoga County web mapping application website [67]. Aligning the results with the census tract map means the two datasets can be integrated. Figure 3 shows the integration process with a marked progress

census tract map on the left and the Cuyahoga County web mapping application website on the right.

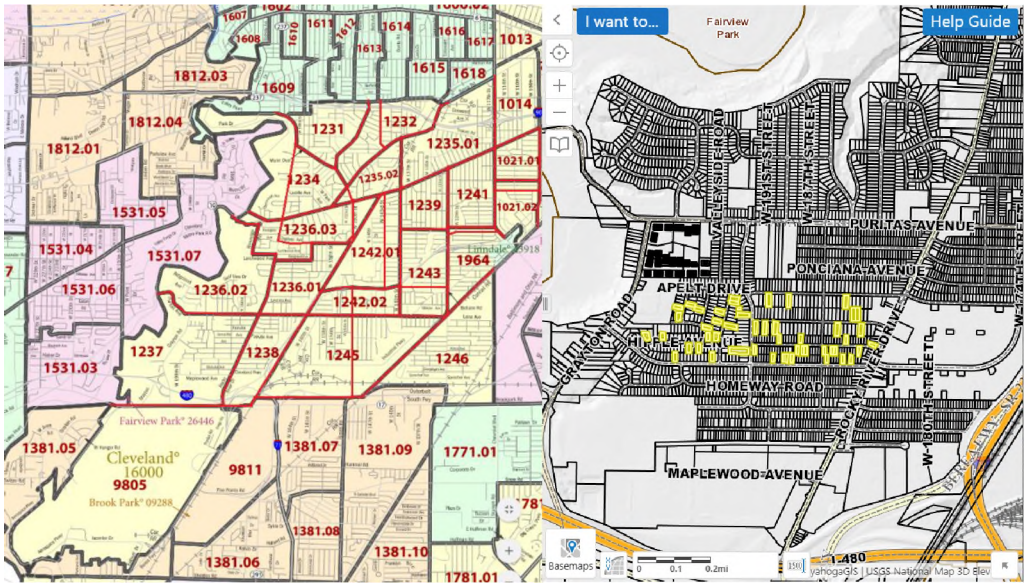


Figure 3: Integration Progress with the Census Tract Map (left) and the County Website (right)

Twenty of the sixty cities in the county are small enough to be contained within a single tract area, so for those it is little effort to integrate them. As for the other forty, months of work is required to correlate the roughly 550,000 properties to their relative geocodes because the parcel numbers do not often align with the tract lines. The information is stored in an Excel file called the Tracts file, pictured in Figure 4, which contains lists of every tract and in which city it belongs. The smallest cities only have one, but the largest, Cleveland, has 177 tracts alone. Also in the file are lists of the parcel number ranges which relate to cities. For example, the city of Bay Village only contains parcel numbers that begin with a string from "201" to "204", such as "20101001". Cleveland is unique in this regard in the sense that it contains two parcel ranges "001-029" and "101-144" which correspond to the West and East halves of Cleveland, respectively. Lastly, the Tracts file also contains a master list of every parcel number in

the county, the city in which that parcel is located, the determined tract number, and the geocode which is created by concatenating the FIPS state code and county code with the tract code.

	A	B
1	city	tracts
2	Bay Village	130103, 130104, 130105, 130106
3	Beachwood	131102, 131103, 131104
4	Bedford	132100, 132200, 132301, 132302
5	Bedford Heights	133103, 133104, 195600
6	Bentleyville	195800 (Shared with Solon)
7	Berea	134100, 134203 (Shared with Olmsted Township), 134204, 134205, 134206, 134300
8	Bratenahl	192800
9	Brecksville	135103, 135104, 135105, 135106
10	Broadview Heights	136101, 136102, 136103

	A	B
1	city	parcels
2	BAY VILLAGE	201-204
3	BEACHWOOD	741-742
4	BEDFORD	811-814
5	BEDFORD HEIGHTS	791-792
6	BENTLEYVILLE	941-941
7	BEREA	361-364
8	BRATENAHL	631-631
9	BRECKSVILLE	601-606
10	BROADVIEW HEIGHTS	581-585

	A	B	C	D
1	parcel	city	tract	geocode
2	10101001	CLEVELAND	107101	39035107101
3	10101002	CLEVELAND	107101	39035107101
4	10101003	CLEVELAND	107101	39035107101
5	10101004	CLEVELAND	107101	39035107101
6	10101005	CLEVELAND	107101	39035107101
7	10101006	CLEVELAND	107101	39035107101
8	10101007	CLEVELAND	107101	39035107101
9	10101008	CLEVELAND	107101	39035107101
10	10101009	CLEVELAND	107101	39035107101

Figure 4: Consolidated Tracts by City in Cuyahoga County

The finished Tracts file allows for a new variable to be derived from the preprocessed data. Present in the Land Data is a land use code (LUC) variable which indicates the main use of the property such as a restaurant or bank [20]. By creating a matrix between the tract and LUC information, it became known through Hotspot Analysis how many of each type of property exist in each tract. These counts are further processed using prediction accuracy index (PAI) techniques [9, 28] and normalized to create Hotspot Analysis weights, arguably the most significant weights used in the ranking function. Figure 5 shows some of the possible LUC values alongside their corresponding four-digit codes while Section 4.2. goes into further detail about hotspot

PAI analysis. Additionally, Chapter 6 demonstrates some experiments using Hotspot Analysis based on various LUC property types.

	A	B	C
1	LUC COUNTS		
2	LUC_Meaning	LUC	Count
3	NULL (Missing LUC)	0000	27288
4	Agricultural vacant land	1000	1
5	Nurseries	1080	34
6	Greenhouses, vegetables, floriculture	1090	9
7	Agricultural vacant land (CAUV)	1100	6
8	Cash grain/general farm (CAUV)	1110	23
9	Livestock farms (not dairy or poultry) (CAUV)	1120	64
10	Fruit and nut farms (CAUV)	1150	3
11	Vegetable farms (CAUV)	1160	7
12	Timber (CAUV)	1210	22
13	Other agricultural use (CAUV)	1990	7
14	Oil and gas rights-working interest	2400	452
15	Oil and gas rights-separate royalty interest	2500	363
16	Industrial vacant land	3000	1117
17	Loose material and storage yard	3010	26
18	Equipment and machinery storage yard	3020	42
19	Salvage yard, scrap metals, etc.	3030	121
20	Vehicle recycling yard	3040	33
21	Billboard sites	3050	6
22	Land fill	3060	28
23	Recreational vehicle storage yard	3070	2
24	Food and drink processing plants and storage	3100	67
25	Foundries and heavy manufacturing plants	3200	50

Figure 5: Land Use Codes

CHAPTER IV

OVERVIEW OF THE FRAMEWORK

4.1. Architecture

This chapter is dedicated to explaining the overall architecture of the property recommendation system. The framework is divided into six phases of data processing pipelining; the first phase is big data collection and geospatial information integration that is followed by the second phase for big data preprocessing and deriving geospatial information with a variety of big data analytic methods to derive hidden relevant features and to discover location-based complex information in the surrounding areas. The third and fourth phases are natural language processing phases to analyze a user-given query sentence to derive and determine relevant factors and weights of features to generate a candidate set in the following fifth phase. Finally, in the sixth phase, a well-defined system ranking model is used to calculate a score for each candidate in the set to identify the Top-*N* recommendation list to display. Figure 6 provides an overview of the architecture of the framework with tasks of each phase of the data processing pipelining. The tasks in each phase will be described in a corresponding subsection throughout this chapter.

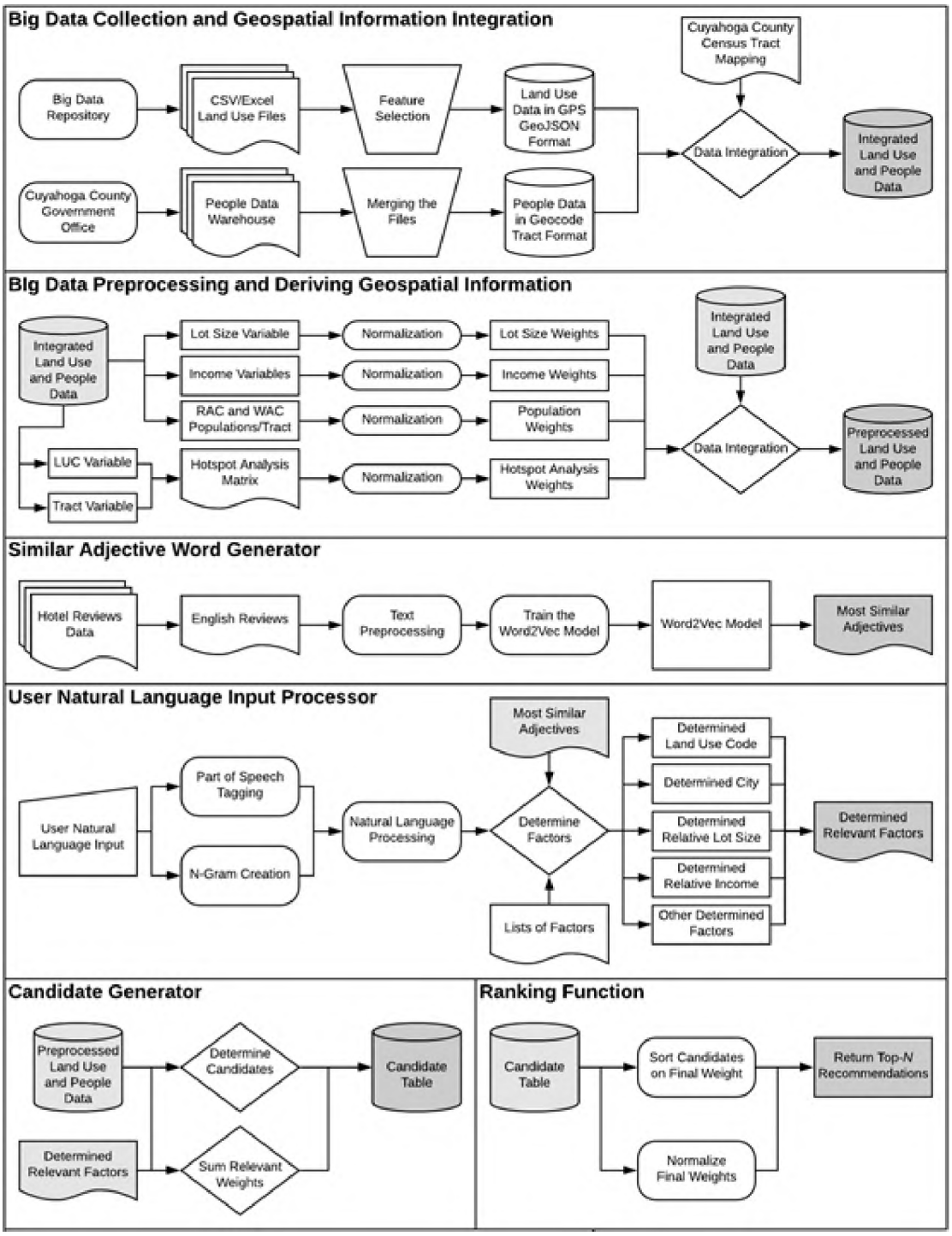


Figure 6: Framework Overview for the Property Recommendation System

4.2. Big Data Collection and Geospatial Information Integration

The first stage in the process is about collecting and integrating the big data into a single usable database. After collecting all the Land Use Data files from the different

sites and big data repository, extensive preprocessing is done for data integration. Different file formats, such as GeoJSON and CSV, are converted into a unified format. Then the manual preliminary feature selection begins from this early stage. Each variable from each file is looked through and analyzed to see if it is a good fit for this recommendation system. Generally, the ones selected are features that an average user would look for in a property, such as the location and house style. The ones that are selected are then subset into a Land Use Data database for future use and preprocessing. Additionally, from the Cuyahoga County government office comes the People Data warehouse files. These roughly 1,500 files are first merged into three main files to prevent the database from having to store a ridiculous number of tables. Once the data is joined, the information is moved to a People Data database just like with the Land Data.

With the two databases created, the next step is to integrate all the data using the Cuyahoga County census tract maps [63, 64, 65]. The Land Data has a unique identifier for the data in the form of a parcel number, and the People Data has the same as a geocode. The tract maps aid in aligning and integrating these two variables since the information covers the same area, namely the county. Subsequently after the work is finished, the combined information is stored in a new database called the integrated Land Use and People Data database for use in the next section about preprocessing the data and deriving the geospatial information.

4.3. Big Data Preprocessing and Deriving Geospatial Information

This phase is the most complicated component of the property recommendation system. This second stage is divided into two subcomponents; one is the big data integration part, and the other is the big data analytic part which derives the information

from the complex geospatial maps and discovers the location-based hidden knowledge from the surrounding areas of each property by applying big data analytic techniques.

In the big data integration subcomponent, the integrated data in the consolidated database from the previous stage is processed into a finalized format for completion with normalized weights and preprocessed derived factors for the next phase of processing. First, from the database, several variables are pulled for preprocessing. The lot size and income are the first two which are normalized into weights for use later in the ranking function. Doing this allows the variables to hold the same amount of significance so one variable that happens to have values in the thousands does not completely overshadow a different variable that is usually in the single digits when the two are summed together. Moreover, from the People Data, the RAC and WAC populations are calculated with respect to the tract areas by dividing the total populations. The tract populations are then normalized like the previous variables for the same reason.

In the next subcomponent, Deriving Geospatial Information, two other variables, the LUC and tract values, are also pulled from the database, but these are used for data analytic methods to create a knowledge base in a matrix form that is derived from one of the data analytic methods, Hot Spot Analysis [9]. The Hot Spot matrix leads to the Prediction Accuracy Index (PAI) [9] analysis to obtain weights that represent the concentration intensity of each type of property and in which tract they are located. The final PAI values are also normalized into weights in the same scale as the other variables. The detailed data analytic methodologies employed in this phase will be described in great deal in the following Chapter 5 Methodologies.

With the weights preprocessed and normalized, the remaining step is to incorporate those values into the integrated Land Use Data and People Data database. This allows for the creation of the final preprocessed version of the database which can easily queried to create candidates with all the relevant weights alongside them based on the text analysis of the user natural language input. The finalized database is pictured in Figure 7.

parcel	geocode	tract	city	LUC	LUC_meaning	beds	baths	lotsize	lotsize_weight
64832002	39035196200	196200	EUCLID	4080	Apartments 40 or more units (garden)	0	0	451870	0.01382
95619001	39035184108	184108	SOLON	1110	Cash grain/general farm (CAUV)	2	1	991426	0.03033
10708065	39035111800	111800	CLEVELAND	5990	Other residential structures	0	0	3255	0.00010
48619004	39035175202	175202	NORTH ROY...	1150	Fruit and nut farms (CAUV)	3	2	1083337	0.03314
10712134	39035118602	118602	CLEVELAND	5990	Other residential structures	0	0	3500	0.00011
95505004	39035195800	195800	SOLON	1110	Cash grain/general farm (CAUV)	3	1	1794672	0.05490
21213006	39035189110	189110	WESTLAKE	4720	Home garden center	4	1	11220	0.00034
73625099	39035183605	183605	SHAKER HEI...	5990	Other residential structures	0	0	6500	0.00020
78203033	39035171103	171103	MAPLE HEIG...	4095	Day care centers	3	1	11200	0.00034
10424001	39035108400	108400	CLEVELAND	4590	Franchise auto service center	0	0	25330	0.00077
36423001	39035134300	134300	BEREA	4080	Apartments 40 or more units (garden)	4	2	60210	0.00184
10428003	39035108301	108301	CLEVELAND	4800	Commercial warehouse (under 75,000 sq. ft.)	0	0	8515	0.00026

Figure 7: Preprocessed Land Use and People Data Database

4.4. Similar Adjectives Generator

In this phase, the Similar Adjective Generator, as shown in Figure 6, employees Natural Language Processing (NLP) technique Word2Vec [45, 46, 47] to create a language knowledge base to generate a similar context words list for a given term in a user query string. The generated lists are used for the fourth phase to analyze a user query string in a phrase or sentence given as an input to the system. In this third phase, correct word embeddings are generated to obtain a synonym list. A Word2Vec model [45, 46, 47] is created, trained, and used to produce the most similar adjectives which are needed for the next phase. The Word2Vec process will be described in detail in Chapter 5.

4.5. Natural Language Query Analyzer

The fourth phase is the Natural Language Query Analyzer which uses the language knowledge base from the previous phase in conjunction with other NLP

methods to derive relevant factors to analyze a user-given query sentence. This phase uses Part-of-Speech tagging (POS) [49], N-gram creation, and Named Entity Recognition (NER) techniques [4] to determine the relevance of specific derived weights. Moreover, this phase is about taking in the user input and processing it through text analysis techniques to determine a set of derived relevant factors with which to create candidates for the following phase. The NLP methodologies – POS and NER processes – will be discussed in detail in the following Chapter 5.

4.6. Candidate Generator

The Candidate Generator phase, the second-to-last phase, deals with filtering down the potential properties into the candidates that fit the needs of the user. This phase starts with searching from the preprocessed integrated database in the second phase and combining it with the determined relevant weights from the previous phase to produce a new candidate set. This candidate set is then used in the final phase to generate a score for each of the candidates and to generate the Top-*N* recommendations.

4.7. Ranking Function

The sixth phase of the framework ranks each candidate with the system ranking function to generate a score to produce the recommendation list. Each of the candidates and their weights are both sorted and normalized into their final weights before being returned as the Top-*N* recommendations to the user. The final subsection of Chapter 5 details the algorithms of how the normalization is performed and how each of the individual weights as well as the final score are calculated.

CHAPTER V

METHODOLOGY

5.1. Geospatial Information

The geospatial information is a structured encoding scheme, called a geocode, that is used in geographical maps for the Census Bureau [7, 8]. It is also used in the Land Use Data based on its geographical location and its proximity to other nearby areas. These areas are divided up based on cities, census tracts, census blocks, neighborhoods, and property lines. In this recommendation system the geospatial information is used to integrate the Land Use Data and the People Data. More importantly, it is also used to derive geospatial knowledge using data analytic methods for additional key factors.

Census Blocks are a geographical unit of areas for statistics bounded by visible features, such as streets, roads, streams, and railroad tracks, and by nonvisible boundaries, such as selected property lines and city, township, school district, and county limits and short line-of-sight extensions of streets and roads [7]. They are the smallest unit of tabulation geography defined by the Census Bureau – there were a total of 11,166,336 defined for the 2010 census, covering the U.S. and its territories – but are diverse in size. While the largest block is over 8,500 square miles in Alaska, half the blocks are smaller than a tenth of a square mile (6.4 acres) [7]. However, these units only

make up the final four digits of geocodes and are the lowest on the geographical hierarchy.

The next unit size up are the Census Tracts. Census tracts are small, relatively permanent statistical subdivisions of a county each uniquely numbered in each county with a numeric code [8]. Each census tract averages about 4,000 inhabitants with the minimum and maximum being, 1,200 and 8,000 individuals, respectively. Each tract number also has a unique four- or six-digit identification code. The discrepancies come from updates to the census tracts by the Participant Statistical Areas Program (PSAP) [8]. PSAP is a program offered once every ten years for local involvement in delineating statistical areas. They split or merge current census tracts depending on population changes over the past decade. Ideally census tracts are relatively permanent to allow data from different decades to be easily compared. However, when a tract has a population over 8,000 it is split into two or more tracts with each tract being given a unique extension to its existing numeric code. Minor revisions are also sometimes allowed. Figures 8-1 and 8-2 showcase an example of a census tract from 1970 being split and renamed several times [8].

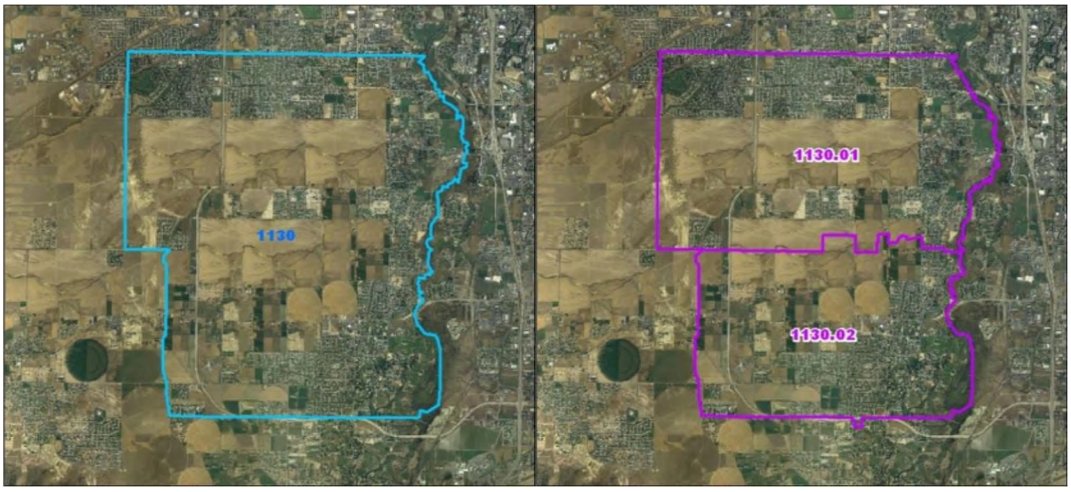


Figure 8-1: Splitting a Census Tract

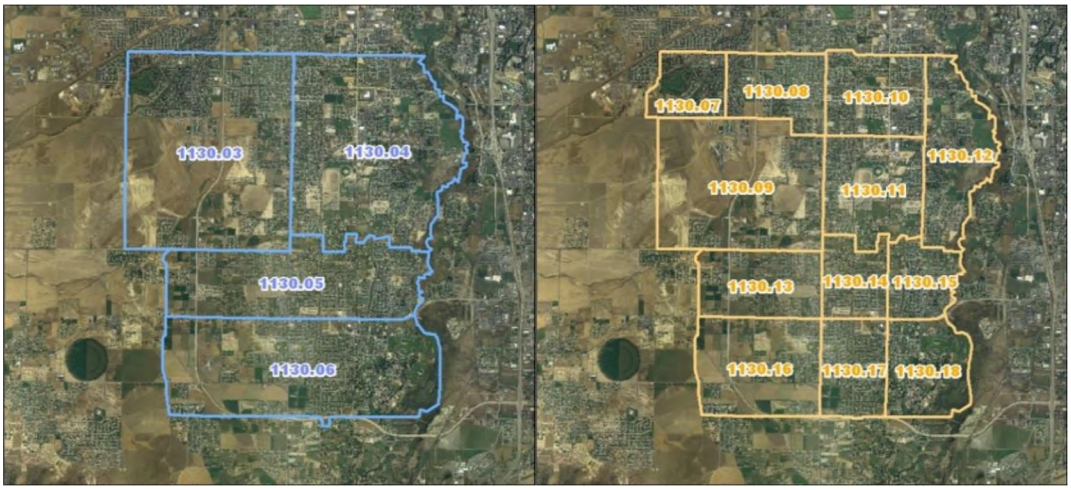


Figure 8-2: Splitting a Census Tract (continued)

The remaining parts of the geospatial information are the county and state codes. In a similar fashion to the previous parts, each has a unique numeric identifier determined by their FIPS codes with Ohio having a state code of 39 and Cuyahoga County having a county code of 035 [2]. Combining the state code, the county code, the census tract code, and the census block code creates the geocode of the property. Figure 9 showcases the layout of the geocodes in more detail and Figure 10 displays the relationships between the counties, tracts, and blocks [1, 7].

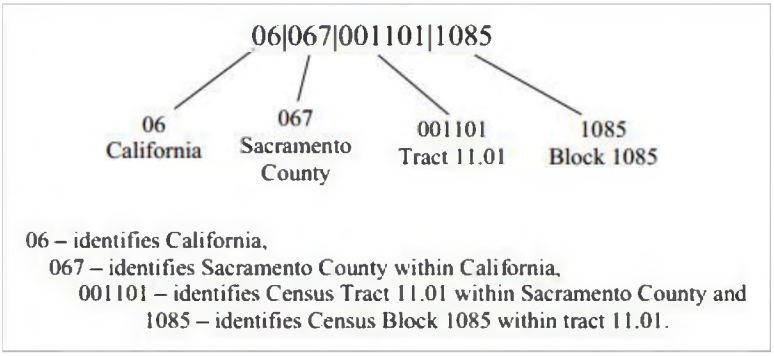


Figure 9: Breakdown of a Geocode

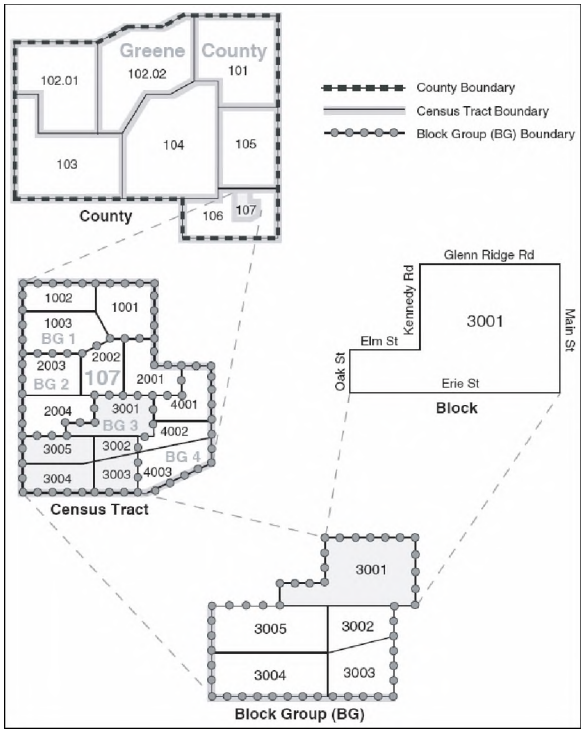


Figure 10: County-Tract-Block Group-Block Relationship

5.2. Hotspot Analysis

In the second phase of the recommendation system architecture, a data analytic method called Hotspot Analysis [9, 28] is used to derive geospatial knowledge from the data using the LUC and tract variables to create a matrix. Hotspot Analysis is the process of identifying locations that are statistically significant hot spots or cold spots in the data by aggregating points of occurrence into polygons or by converging points that are in proximity to one another based on calculated distance [9, 28, 69]. The hot spots indicate areas with a high frequency of cases while the cold areas indicate areas with a low frequency. This technique is often used to identify high crime locations, such as by The U.S. National Institute of Justice who described criminal hotspots in a report as, "an area that has a greater than average number of criminal or disorder events, or an area where people have a higher than average risk of victimization" [42]. Generally, though, the polygon maps will be in the shape of administration boundaries or custom square grids,

examples of which can be seen in Figure 11 with the hot spots in red and the cold spots in blue [69].

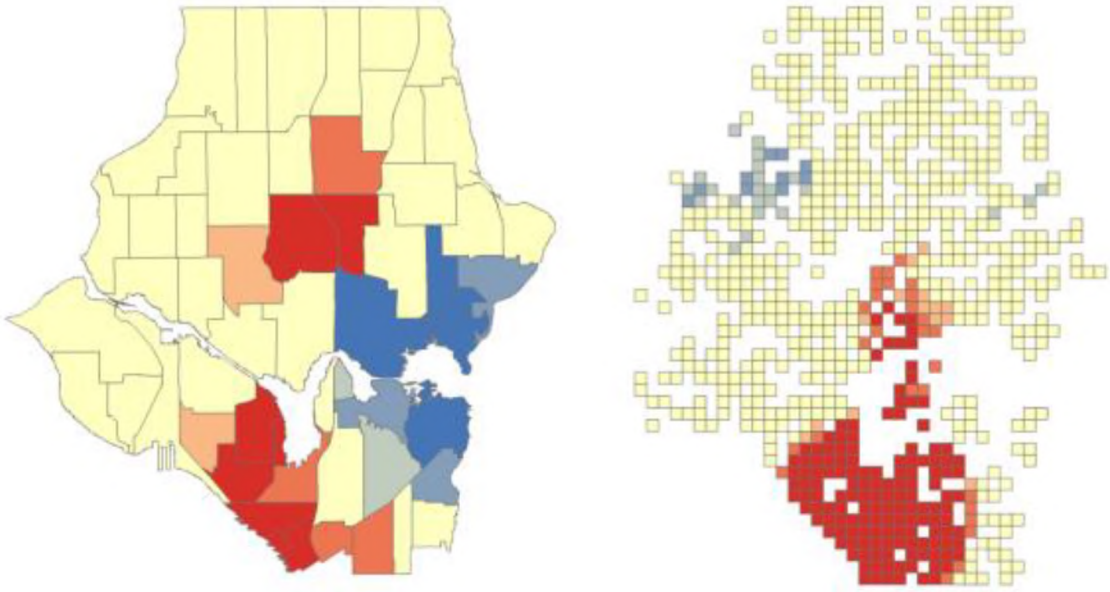


Figure 11: Hotspot Analysis Map Examples

One reliable and well-known method to calculate whether an area is a hotspot or not is the Predictive Accuracy Index (PAI) measure [9, 28]. It was proposed by Chainey et al [9] in 2008 to define a measure for testing forecasting accuracy. It measures the hit rate against the areas where targets are predicted to occur with respect to the size of the study area". The criminology field, specifically those working on forecasting and prediction, have principally relied on this measure since its inception [9]. Hotspot PAI analysis works by measuring how many instances of a data point appears in a specified area versus how many instances of that same type exist in the entire area. The magnitude of the resulting answer indicates the frequency, so by comparing the PAI value for one area against all other areas it is possible to identify the locations with the highest concentration of the given data type. Furthermore, by normalizing all the values, PAI

weights can be created to be used in the ranking function in the final phase. The PAI analysis formula is shown in (1) with n and N indicating the number of nodes in the given area and the entire area, respectively, and a and A indicating the size of the selected area and entire area, also respectively [9, 28].

$$\text{Prediction Accuracy Index (PAI)} = \frac{n/N}{a/A} \tag{1}$$

In addition to the PAI measure, Hunt [28] proposed the Prediction Efficiency Index (PEI*) as a complimentary measure. Hunt sought to define a measure for testing forecasting efficiency by measuring how well a forecast does compared to how well it could have done. Despite it being only recently introduced, it is the only known plausible alternative at this time. The formula for it is shown in (2) with n^* indicating the maximum obtainable n value for the area a [28].

$$\text{Prediction Efficiency Index (PEI}^*) = \frac{\text{PAI}}{\frac{n^*/N}{a/A}} = \frac{n^*}{n} \tag{2}$$

5.2.1. Hotspot Analysis by Site Category

To better explain PAI analysis, below is an example showcasing the hotspot technique over the Downtown Cleveland area. The value being analyzed was the SiteCat1 variable which specifies the main site category type for each property and comes from the Land Use Data. There are nine possible values, as well as a tenth serving as a combination of the other nine, which include commercial, government, and residential spaces as well as many others listed in the map key. The map in Figure 12 shows colored points specifying each property and its type.

Table 3 lists each possible SiteCat1 value, the number of points of that type in the downtown area, n, the number of points of that type in all of Cuyahoga County, N, and the PAI value. The area variables a and A represented the square mileage of the downtown area, 77.7 mi^2, and the entire county, 457 mi^2, respectively. Not surprising given the area in question, the SiteCat1 types with the highest PAI values are the commercial and utility types. This makes sense since the downtown area is a public hub of stores, businesses, restaurants, and utility sites, such as power stations and water processing facilities.

Figure 12: Hotspot PAI Analysis of the Downtown Cleveland Area

Table 3. The Hotspot PAI Analysis Results of Downtown, Cleveland

SiteCat1	n	N	PAI
Agricultural	0	157	0
Commercial	1314	23337	0.3311658
Government	335	28673	0.06871742
Industrial	187	6679	0.1646741
Institutional	99	9258	0.06289458
Mixed	28	2804	0.05873206
Other	1	246	0.02390893
Residential	869	456027	0.0112079
Utility	57	886	0.3783871
Everything	2890	528067	0.03218874

5.2.2. Hotspot Analysis by Land Use Code

For this thesis hotspot PAI analysis is also performed on the property LUC values to discover the most ideal and relevant areas for each type of establishment. To that end, a table is created with every LUC in Cuyahoga County listed with the number of cases for each type. Next, a matrix is created between the LUC information and all 446 tract numbers with the values being the number of the given type of property being in the given tract. The total land area of the tracts as well as the county is also recorded from the United States Boundary website [59] pictured in Figure 13.

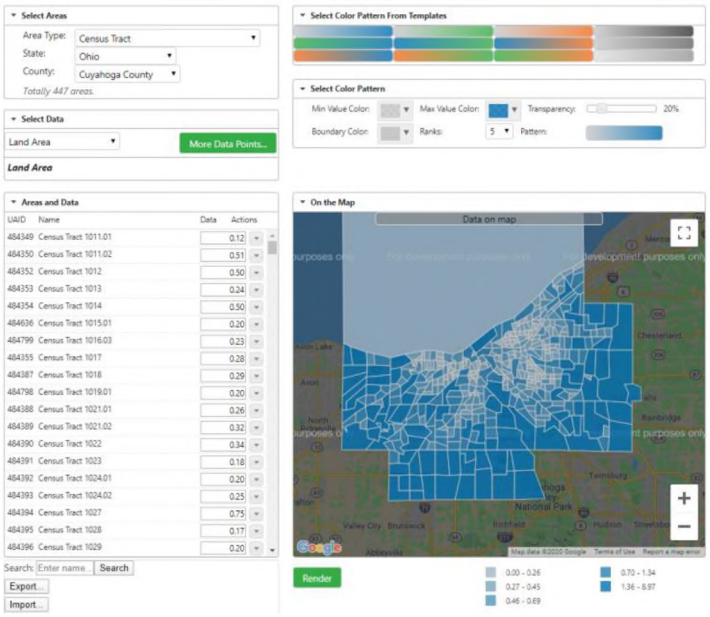


Figure 13: United States Boundary Website

From there a second matrix is created to calculate the PAI values by using the first matrix. To do that, the count of a given type of LUC in each tract from the first matrix is divided by the total count of the given LUC across the county. Then the resulting value is divided by the area of the tract over the area of the entire county. The final value is stored in the new matrix in the corresponding location and this process is repeated across every tract for every type of LUC. Some of the results can be seen in

Figure 14 where a higher PAI value indicates a stronger concentration of the type of property, also known as a hotspot, with the lower values indicating a cold spot. Any zeros in the second matrix indicate that there are no properties of the given type in the given tract. These are the values normalized and used in the ranking function phase of the architecture.

PAI ANALYSIS	TRACT						
LUC_Meaning	135104	135105	135106	136101	136102	136103	137101
NULL (Missing LUC)	0.340313279	0.116116031	0.239917239	0.09105372	0.083614581	0.127027403	0.460731824
Agricultural vacant land	0	0	0	0	0	0	0
Nurseries	0	0	0	0	4.970968794	0	0
Greenhouses, vegetables, floriculture	0	0	0	0	0	0	0
Agricultural vacant land (CAUV)	0	0	0	0	14.08441158	0	0
Cash grain/general farm (CAUV)	5.176403986	0	0	0	14.6967773	0	0
Livestock farms (not dairy or poultry) (CAUV)	0	4.715140264	0	2.588201993	1.320413586	0	0
Fruit and nut farms (CAUV)	0	0	0	0	0	0	0
Vegetable farms (CAUV)	0	0	0	0	0	0	0
Timber (CAUV)	0	0	0	0	3.841203159	0	0
Other agricultural use (CAUV)	0	0	0	0	12.07235279	0	0
Oil and gas rights-working interest	1.317005439	1.6690762	1.338876287	10.62766128	3.739224314	7.04705861	0
Oil and gas rights-separate royalty interest	0	1.246979243	0	78.9437258	3.026402489	10.32335275	0
Industrial vacant land	0	0.742940632	0.147759125	0	0.151309704	0.251614396	4.297573859
Loose material and storage yard	0	0	0	0	0	0	0
Equipment and machinery storage yard	0	0	0	0	0	0	5.442619048
Salvage yard, scrap metals, etc.	0	0	0	0	0	0	0
Vehicle recycling yard	0	0	0	0	0	0	0
Billboard sites	0	0	0	0	0	0	0
Land fill	0	0	0	0	0	13.28348946	0
Recreational vehicle storage yard	0	0	0	0	0	0	0
Food and drink processing plants and storage	0	0	0	0	0	0	0
Foundries and heavy manufacturing plants	0	0	0	0	0	0	4.5718
Manufacturing and assembly, medium	0	0.589392533	0.429809717	0	0	0	3.57171875
Manufacturing and assembly, light	0	0.037910675	0.248814469	0	0	0.141232803	5.858336583
Small shops (machine, tool and die, etc.)	0	0.205006098	0.074749516	0	0	0.254577245	5.590516304
Mines and quarries	0	0	0	0	0	0	0
Grain elevators	0	0	0	0	0	0	0
Contract and construction service facilities	0	0	0	0	0.488476702	0	3.963988439
Bulk oil storage facilities	0	0	15.00426649	0	0	0	0
Research and development facilities	0	0	0	0	0	0	0
Transportation facilities	0	0	3.667709587	0	0	0	15.23933333
Communication facilities	0	0	1.375391095	0	2.112661738	0	0
Utility service facilities	0	0.867152232	0.632363722	0	0	0	0
Other industrial structures	0	0	0.294201304	0	0	1.801972473	8.55694492
NULL (Missing LUC meaning)	0	0	0	0	0	0	0

Figure 14: LUC PAI Analysis Matrix

5.3. K-Means Clustering Analysis

Used for experiments detailed in Section 6.3., K-means clustering (KMC) is a clustering algorithm designed to try and group data points on a graph [25, 54]. Its goal is to try to minimize the distance between points in a cluster and maximize the distance

between clusters. It is used to create correlations among what would otherwise be arbitrary points based on latent factors. KMC works by first plotting all observations on a map and randomly grouping them into *K* groups where *K* is a predetermined integer. Next, the centroids of each group are plotted, and every observation is regrouped to the closest one. After that the centroids are moved to the new centers of all the points within a given cluster. This process repeats until the results converge with the accuracy of the clustering being measured using a sum of squares approach to determine the variance within the clusters. Figure 15 showcases an example of this algorithm with the objective function for it is shown in (3) [54]. In it, *k* represents the number of clusters, *n* is the number of cases, and $||x_i^{(j)}-c_j||^2$ is the measured distance between a data point $x_i^{(j)}$ and a cluster centroid c_j, an indicator of the *n* data points from their respective cluster centers [3]. The experiments for the KMC analysis are described in detail in Section 6.3.

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} ||x_i^{(j)} - c_j||^2 \quad (3)$$

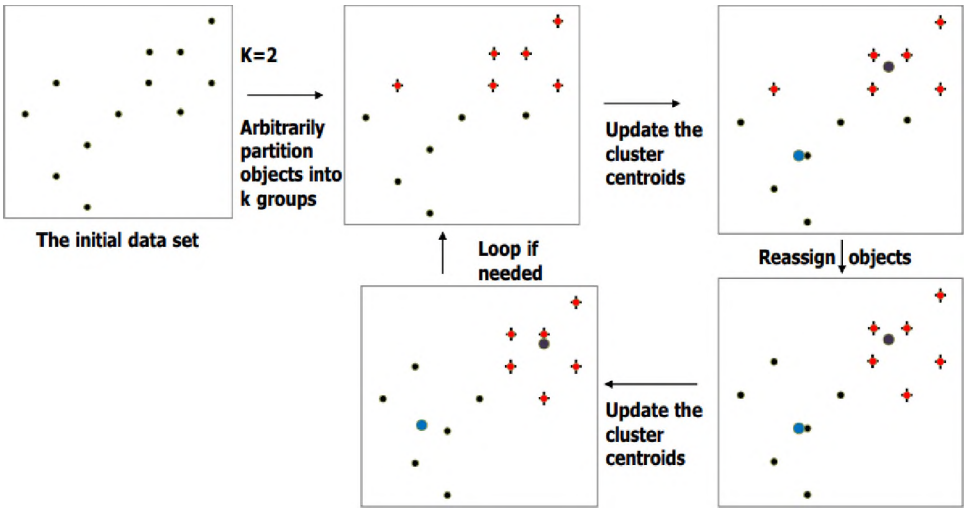


Figure 15: K-Means Clustering Example

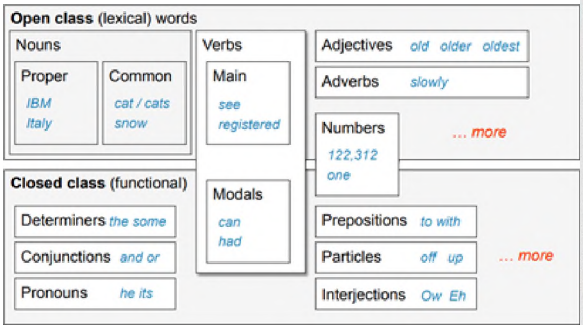
5.4. Natural Language Processing

Natural Language Processing (NLP) techniques are widely employed to process and analyze unstructured texts in a natural language. NLP involves 'understanding' complete human utterances, at least to the extent of being able to give useful responses to them [5]. Natural language processing is broadly defined as the automatic manipulation of natural language, such as speech and text, by software [6]. It is all about understandings what a user means by using Part-of-Speech tagging, Named Entity Recognitions, Chunking, Semantic Role Labeling, and other text analysis techniques to provide them with meaningful results. This section goes into further detail about three natural language text analysis techniques used in this recommendation system.

In the third and fourth phases, the NLP methods are used to analyze the natural language input from a user to derive relevant features. The system starts with using POS tagging and *N*-gram creation to mark each term in the user input and to create unigrams, bigrams, and trigrams from it, respectively. That information is then combined with the most similar adjectives generated from the Word2Vec embeddings [45, 46, 47] and lists of potential factors for each variable to determine the relevant features. These features include the LUC, city, and relevant lot size and income. The relative lot size and income factors indicate whether the user is looking for a larger or smaller property and a higher or lower income neighborhood, respectively. There are related factors as well, such as the number of bedrooms and bathrooms for residential homes. Each of these processes are detailed further below in the following subsections.

5.4.1. Part-of-Speech Tagging

One such technique is Part-of-Speech tagging (POS) [49] which is used in the fourth phase of the architecture to label each word with a grammatical tag that indicates its syntactic role in the sentence [10]. In schools English is commonly said to have only nine parts of speech: noun, verb, article, adjective, preposition, pronoun, adverb, conjunction, and interjection. However, that is not the case because of subcategories, such as plural, possessive, and singular forms for nouns alone [49]. Figure 16 demonstrates many different parts of speech with examples [49]. The open class groups are for classes that still allow new words to be added, while the closed classes are finite. Nevertheless, words often belong to more than one category and this is the core reason behind why Part-of-Speech tagging is difficult to perfect. For example, the word "back" can be an adjective, "The back door", a noun, "On my back", an adverb, "Win the voters back", or a verb base form, "Promised to back the bill" [49].



Tag	Description	Example	Tag	Description	Example
CC	coordin. conjunction	*and, but, or*	SYM	symbol	*+,%, &*
CD	cardinal number	*one, two*	TO	"to"	*to*
DT	determiner	*a, the*	UH	interjection	*ah, oops*
EX	existential 'there'	*there*	VB	verb base form	*eat*
FW	foreign word	*mea culpa*	VBD	verb past tense	*ate*
IN	preposition/sub-conj	*of, in, by*	VBG	verb gerund	*eating*
JJ	adjective	*yellow*	VBN	verb past participle	*eaten*
JJR	adj., comparative	*bigger*	VBP	verb non-3sg pres	*eat*
JJS	adj., superlative	*wildest*	VBZ	verb 3sg pres	*eats*
LS	list item marker	*1, 2, One*	WDT	wh-determiner	*which, that*
MD	modal	*can, should*	WP	wh-pronoun	*what, who*
NN	noun, sing. or mass	*llama*	WP$	possessive wh-	*whose*
NNS	noun, plural	*llamas*	WRB	wh-adverb	*how, where*
NNP	proper noun, sing.	*IBM*	$	dollar sign	*$*
NNPS	proper noun, plural	*Carolinas*	#	pound sign	*#*
PDT	predeterminer	*all, both*	"	left quote	' or "
POS	possessive ending	*'s*	"	right quote	' or "
PRP	personal pronoun	*I, you, he*	(	left parenthesis	[, (, {, <
PRP$	possessive pronoun	*your, one's*	)	right parenthesis	],), }, >
RB	adverb	*quickly, never*	,	comma	,
RBR	adverb, comparative	*faster*	.	sentence-final punc	. ! ?
RBS	adverb, superlative	*fastest*	:	mid-sentence punc	: ; ... – -
RP	particle	*up, off*			

Figure 16: English Parts of Speech (left) and the Penn Treebank Tagset (right)

To clear the confusion the Penn Treebank Tagset was created which defined 36 parts of speech with a unique tag and definition, also pictured in Figure 16 [49, 52]. Using this, and by assigning every word its most common tag, leads to a baseline

accuracy of about 90%. This is because most words are unambiguous, such as articles and punctuation marks, but 11% of words and 40% of word tokens are ambiguous and these ones tend to be very common words, such as "that" which can be a preposition, determiner, or an adverb. Therefore, to accurately assign each word a tag, the knowledge of the neighboring words and the probabilities of each tag for a given word must be considered. The capitalization, prefixes, suffixes, word shapes, and other indicators also aid in this matter.

The concept to take away from this is that POS tagging is a sequence classification problem. Every natural language input is a just a sequence of observations in a sliding window that need to be classified. By independently assigning each word a classifier while also considering the nearby words and their features, tags become more accurate. This concept can be improved upon by increasing the window size to observe more words and by looking at the token tags instead of just the words themselves. This concept is called forward classification, but it also works in reverse. Backward classification can sometimes help make a word less ambiguous due to word ordering. Some examples of forward classification can be seen in Figure 17 [49].

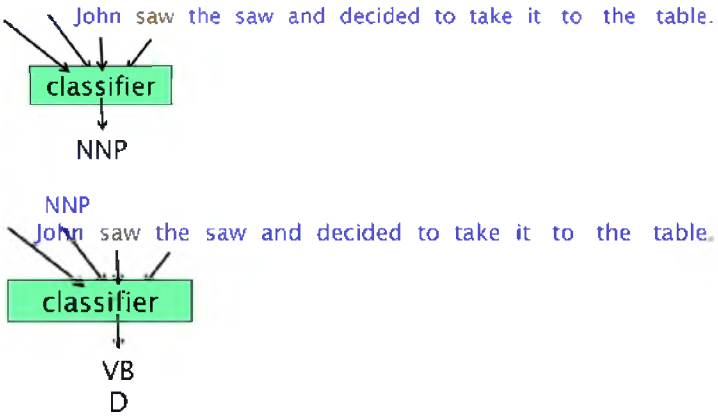


Figure 17-1: Forward Classification Example

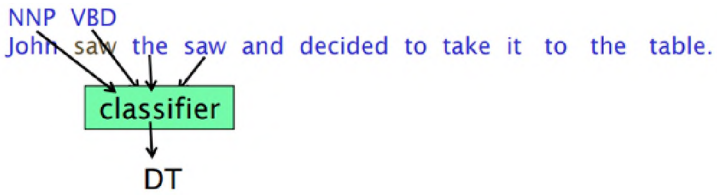


Figure 17-2: Forward Classification Example (continued)

A more advanced concept is the Maximum Entropy Markov Model (MEMM) which is a sequence version of the logistic regression, also known as maximum entropy, classifier [30]. It determines the best tag sequence by calculating the probability of a given word having a specific tag based on the word itself, the previous word, and the previous tags. Instead of each word being treated as conditionally independent, a Markov chain, a sequence of possible events, is created. The equation for MEMM is shown in (4) with $\hat{T}$, t_i, and w_i representing the tag sequence, the tag at position i, and the word at position i, respectively. Figure 18 shows an example of the MEMM model [30, 49].

$$\hat{T} = \underset{T}{\operatorname{argmax}}\, P(T|W) = \underset{T}{\operatorname{argmax}} \prod_i P(t_i|w_i, t_{i-1}) \tag{4}$$

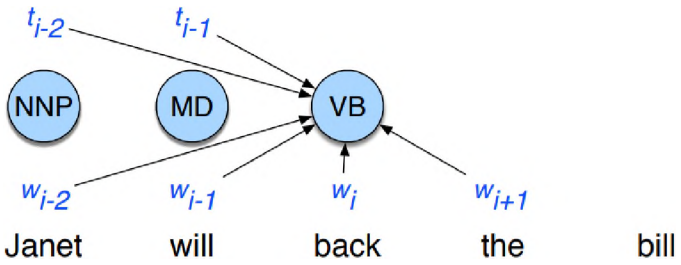


Figure 18: Maximum Entropy Markov Model Example

From these concepts new POS models were developed such as the Stanford CoreNLP model pictured in Figure 19 [35, 41]. It is an integrated NLP toolkit with a broad range of grammatical analysis tools for use in breaking down natural language user input across six languages. The model can provide the base forms of words and their

parts of speech, it can indicate whether each word is a name of a company, person, etc., and it can normalize dates, times, and numeric quantities. One example of the model being tested can be seen in Figure 19. It uses the natural language input of "Alexander Nwala has been a Computer Science PhD Student at Old Dominion University in Norfolk Virginia since May 2014, under the supervision of Dr. Michael Nelson". There each word is assigned a Part-of-Speech tag based on nearby words and probabilities, such as "Alexander" being tagged "NNP" to indicate that it is a singular proper noun, or "been" being tagged as a "VBN" meaning a past participle verb. Their research has led to the model achieving a 97% accuracy.

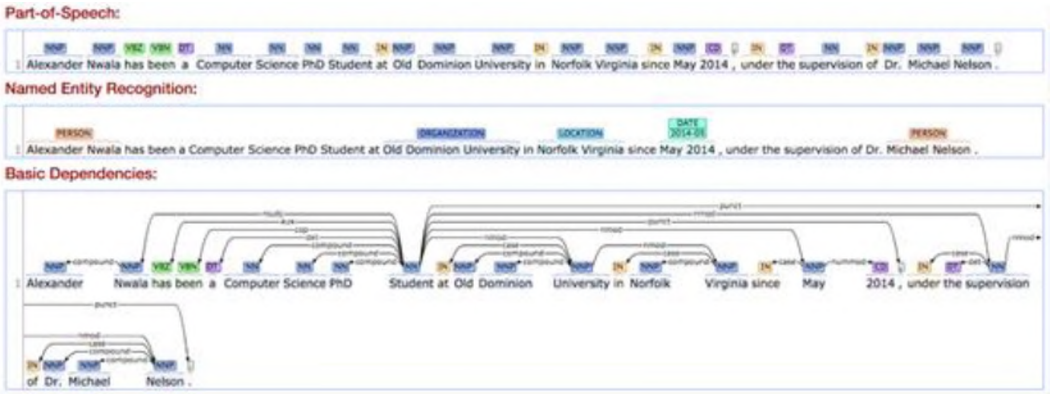


Figure 19: Text Analysis using the Stanford CoreNLP

5.4.2. Named Entity Recognition

Named Entity Recognition (NER) [29] is another common text analysis practice that aims for finding and classifying names of persons, dates, locations, and organizations in natural language text, an example of which can be seen in Figure 19 [29]. It is used for a variety of reasons such as attributing sentiment analysis with companies and products, question and answering systems, and web page tags for quicker searching. Unlike POS tagging, NER is evaluated based on entities, such as the N-grams used in the User Natural Language Input Processor portion, instead of individual tokens. More

specifically, NER models are evaluated using precision, *P*, which is the percentage of selected items that are correct, recall, *R*, which is the percentage of correct items that are selected, and the F-measure, which is the precision-recall tradeoff. F-measure is also a weighted harmonic mean, which can be seen in (5), with a very conservative average, so generally a balanced F_1-measure is used with β, a parameter that controls a balance between *P* and *R*, being set to one and α being set to one-half. Doing this reduces the formula significantly to just $F = 2PR/(P+R)$ [53].

$$\mathrm{F-measure} = \frac{1}{\alpha\frac{1}{\mathrm{P}}+(1-\alpha)\frac{1}{\mathrm{R}}} = \frac{(\beta^2+1)\mathrm{PR}}{\beta^2\mathrm{P}+\mathrm{R}} \text{ where } \alpha = \frac{1}{\beta^2+1} \tag{5}$$

Regardless, many advanced NER models have been developed such as the Stanford NER one which uses a conditional random field (CRF) [19]. A CRF is a sequence modeling algorithm that assumes features are codependent while also considering future observations and learning new patterns, somewhat like a MEMM [4]. The goal of each model is to not just memorize given answers for an unlabeled training set, but to come up with its own patterns that can be generalized across new examples. Figure 20 showcases an overview of a NER model that starts with the training data being provided with the natural language text and the proper labels for each entity [24]. From there an error gradient based on the loss function is fine-tuned by calculating the difference between the training examples and the expected outputs. If there is a large difference, then the gradient becomes more significant and the model is updated accordingly. This learning process repeats until a working model is successfully trained and produced.

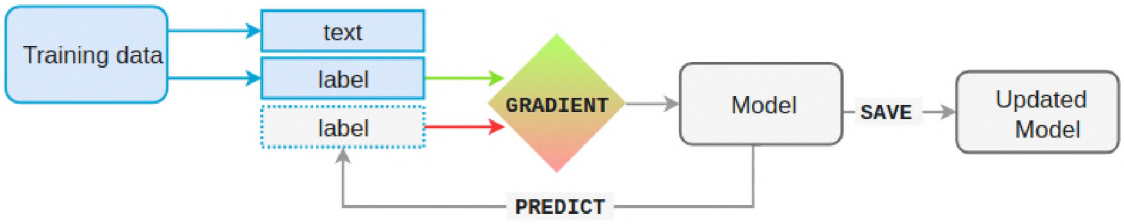


Figure 20: NER Model Overview

5.4.3. Word2Vec Model

The third and final text analysis technique from the NLP literature is the Word2Vec model [45, 46, 47]. It is used to generate the most similar adjectives in the third phase of this recommendation system. Word2Vec is an NLP algorithm with a neural network architecture to generate word embeddings by training a text corpus in a skip-gram model [45, 46, 47] that can be used to find highly comparable words in relationships [44, 45, 46, 47]. The overview of the model is that it is a simple trained neural network with a single hidden layer that is designed to perform a task so the hidden layer weight values can be determined, as that is what is truly most important. The task is for the network to select a random word within a predetermined window size around the input word and to calculate the probability of each word being selected. The output probabilities indicate how likely it is that two given words are closely related. So, for example, if the input word was “British”, then the output probabilities for words like “Columbia” and “Parliament” would be much higher than other words like “Bicycle” or “Computer”. This works because given enough examples, the first two words would appear as a pair much more frequently than the latter two words.

To train the model, the input source texts are split up into predetermined window sizes. Next, the input word is set as the first word in the sequence and training samples

are created by combining the input word with the other words in the window. This allows the network to learn the frequencies of the pairs based on how many times they show up meaning pairs like ("British", "Columbia") are going to appear far more often than ("British", "Bicycle"). After the model has been trained on every possible pair, it will be possible to pass new input words to it and have it print out a list of the most similar words. Figure 21 showcases an example of input sentences, the windows, and the determined training sample pairs [44].

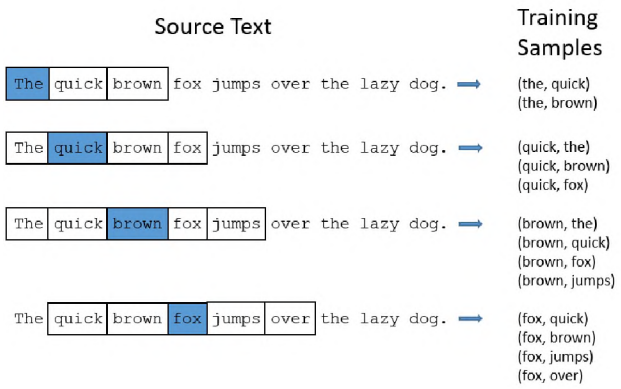


Figure 21: Creating the Word2Vec Pairs

On the input end, the natural language sentences are not simply being fed into the neural network. Instead the entire input text corpus is broken down into the individual words and then transformed into a one-hot encoded vector. If there are 10,000 words, then each vector will be 10,000 bits long with only one bit being a one and the rest being zeros. After each vector passes through the network, the output will be a single vector of the same length with values indicating the likelihood that a randomly selected nearby word is that vocabulary word. This means that when the trained network is given an input word to evaluate, it will return a vector with the floating-point probabilities of every word, so the word "British" would return a high probability for the words "Columbia"

and “Parliament”. Figure 22 shows an overview of the network with the input vectors, hidden neurons, and output layer [44].

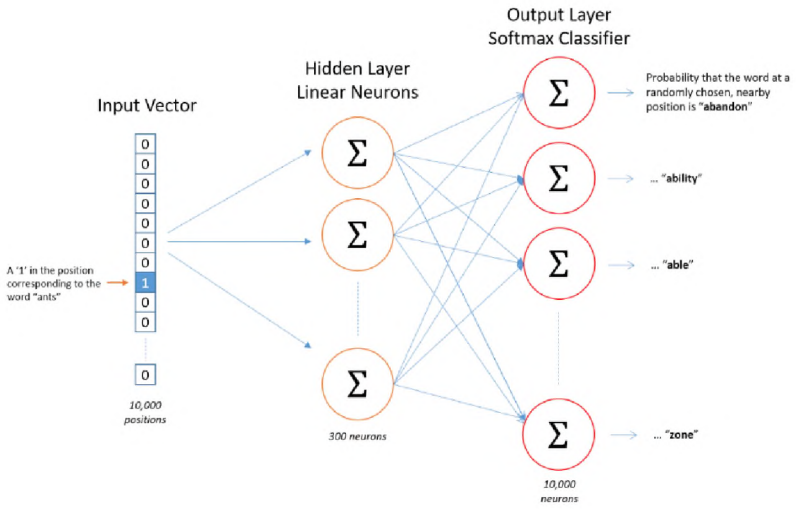


Figure 22: Word2Vec Neural Network Overview

The next part, the hidden layer, is where the weight matrix is created. If the input vector is 10,000 bits long and there are 300 hidden neurons, then a matrix of size 10,000 by 300 is created with the rows indicating the word vectors. Furthermore, by taking one of the one-hot encoded vectors and multiplying it by the weight matrix, the output left is simply one selected row that has the probability weights for the input word. This concept is shown in Figure 23 with an example [44].

$$[0 \quad 0 \quad 0 \quad 1 \quad 0] \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} = [10 \quad 12 \quad 19]$$

Figure 23: Matrix Multiplication

Moreover, a skip-gram model works by predicting the context of words by using a radius of neighboring output context words [40]. As the center word at position t shifts down the input sentence, the windows of neighboring words, m, move with it. The idea is to maximize the probability of any context word given the current center word. Figure 24

showcases an example of part of an input sentence with labels explaining each part [40]. To calculate the probability of an output context word, the weight of it, w_{t+j}, must be taken with respect to the center word weight, w_t, where j is an integer indicating the number of spaces to the left or right of the center word. So, for example, to calculate the probability of the word "turning" being near the word "banking", the formula $p(w_{t-2}|w_t)$ must be evaluated.

Figure 24: Skip-gram Example

By expanding on this idea, it is possible to create an objective function to maximize the probability of any context word given the current center word. Shown in (6), alongside the negative log likelihood loss (7), θ represents the variables being optimized and T represents the final possible position of the word at position t [40].

$$J'(\theta) = \prod_{t=1}^{T} \prod_{\substack{-m \le j \le m \\ j \neq 0}} p(w_{t+j}|w_t;\ \theta) \tag{6}$$

$$J(\theta) = -\frac{1}{T}\sum_{t=1}^{T} \sum_{\substack{-m \le j \le m \\ j \neq 0}} \log p(w_{t+j}|w_t) \tag{7}$$

Lastly, there is the output layer which uses a softmax regression, which is a way in which each output neuron will produce a value from zero to one and the sum of all the values will equal one. Basically, this step takes the input vector for a single word, v_c, the output weights from the matrix for the other words, u_o, and calculates the softmax between them to decide the probabilities of the nearby words. It decides which words are most likely to appear in the vicinity of the input word based on the weight matrix. Figure

25 demonstrates an example of this concept with the equation for it being shown in (8), o being the nearby outside words and c being the center word [44].

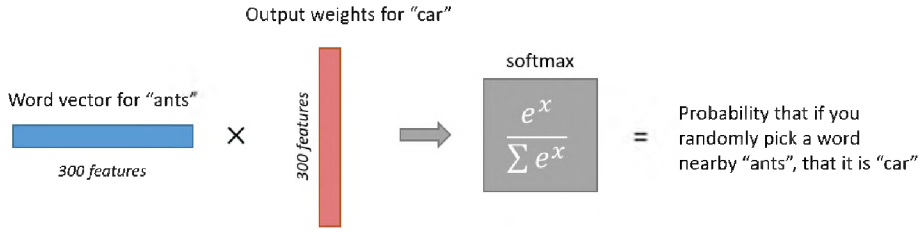


Figure 25: Word2Vec Output Layer Calculations

$$p(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w=1}^{v} \exp(u_w^T v_c)} \tag{8}$$

The great part about this system is that it does not simply need to be used to calculate which words are most likely to be near each other. Pictured in Figure 26, the Word2Vec model can be used to determine which two words are synonymous. The word weight vectors for the synonymous words will be very similar meaning the words are more likely to appear around the same words. This also works for stemming words because terms like "Bicycle" and "Bicycles" are also both likely to appear around the same words. Regardless, when plotted, the synonyms will cluster near each other and the antonyms will be further out.

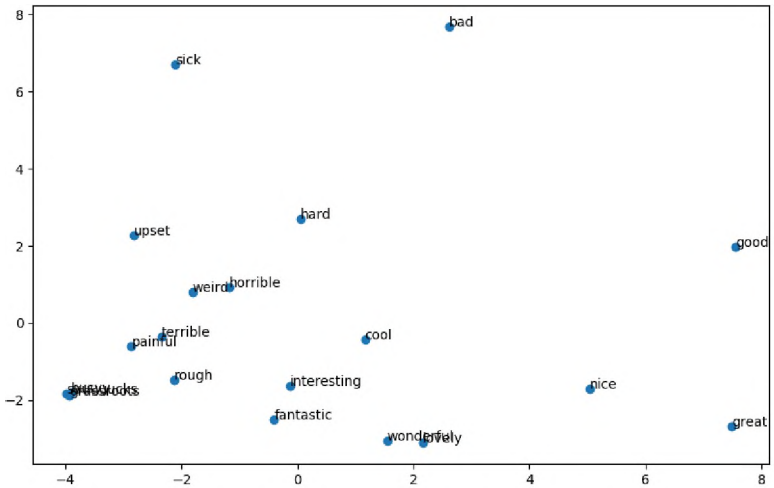


Figure 26: Word2Vec Associations between Similar Words

Additionally, it can also be used to determine which words are similar based on context, a concept of which is used on user reviews in the recommendation system architecture [16]. The Word2Vec model is implemented as the similar adjectives generator by training it with the TripAdvisor hotel review data obtained in a JSON file format [48]. Hotel review data is chosen for three reasons. First, despite the large diversity of businesses within the Land Data, few have conventional review datasets in a natural language format. For instance, there are no datasets or websites for reviewing banks or parks, and while there is information about residential properties, most of it is basic information about the property, such as the number of bedrooms and bathrooms, instead of a written user review. Second, hotels vary widely in style which somewhat mimics the variety of properties across the county. Third, the dataset consists of over 870,000 reviews which serves as a more than adequate basis for training the Word2Vec model. Moreover, the data is transformed from a JSON file format to a CSV one for ease of access. From there the English reviews can be subset away from the French reviews leaving over 770,000 in total. Figure 27 shows a section from the English hotel reviews CSV file.

	ratings.service	ratings.cleanliness	ratings.overall	ratings.value	ratings.location	ratings.sleep_quality	ratings.rooms	text	id
1	5	5	5	5	5	5	5	Stayed in a king suite for 11 nights and yes it cots us ...	147643103
2	5	5	5	4	5	5	5	Wonderful boutique hotel located next to Times Squa...	147722090
3	4	5	4	4	5	4	4	This is a great property in Midtown. We two different ...	147697954
4	4	5	4	3	5	5	5	I have stayed at each of the US Andaz properties, and ...	147612823
5	5	5	5	4	5	5	5	My husband and I stayed at The Chatwal for 9 nights i...	147769675
6	4	5	4	4	5	5	4	This hotel is a nice stay for NYC because the rooms ar...	147759101
7	5	5	5	5	4	5	5	I've stayed at 4 and 5 star hotels all over Manhattan. ...	147735177
8	4	5	4	4	5	5	4	Hotel was very very good. High quality finish through...	147758332
9	4	4	4	4	5	4	4	We chose the San Carlos based on reviews in T.A and ...	147640456
10	5	5	5	5	5	5	5	On every visit to NYC, the Hotel Beacon is the place w...	147639004

combinedReview.0.hotel_class	combinedReview.0.address.region	combinedReview.0.address.street-address	combinedReview.0.address.postal-code	combinedReview.0.address.locality	combinedReview.0.name
3.0	NY	2130 Broadway at 75th Street	10023	New York City	Hotel Beacon
5.0	NY	130 West 44th Street	10036	New York City	The Chatwal
4.0	NY	485 5th Avenue	10017	New York City	Andaz 5th Avenue
4.0	NY	485 5th Avenue	10017	New York City	Andaz 5th Avenue
5.0	NY	130 West 44th Street	10036	New York City	The Chatwal
4.0	NY	851 Avenue of the Americas (Sixth Avenue)	10001	New York City	Eventi – a Kimpton Hotel
4.0	NY	851 Avenue of the Americas (Sixth Avenue)	10001	New York City	Eventi – a Kimpton Hotel
4.0	NY	150 East 50th Street	10022	New York City	San Carlos Hotel
4.0	NY	150 East 50th Street	10022	New York City	San Carlos Hotel
3.0	NY	2130 Broadway at 75th Street	10023	New York City	Hotel Beacon

Figure 27: English Hotel Reviews CSV File

Since the Word2Vec model only requires the review text, the preprocessing only involves separating and cleaning the text into a usable format. Typos, grammatical mistakes, punctuation errors, and other minor issues are resolved. Afterwards the script is written to generate the Word2Vec model and the review data is passed to it to train it. A short while passes before it is ready to be used. The hotel review data produces a vocabulary that contains 317,843 unique terms and is given various common adjectives to describe size and quality. The model returns the top ten most common words in the vocabulary which are used in the user natural language input processor during the text analysis section to better filter out nonideal recommendations. Part of the Word2Vec model output can be seen in Figure 28. These are the outputs for the words, "Upscale", "Low-end", "Big", and "Small" with the corresponding similarity weights next to them.

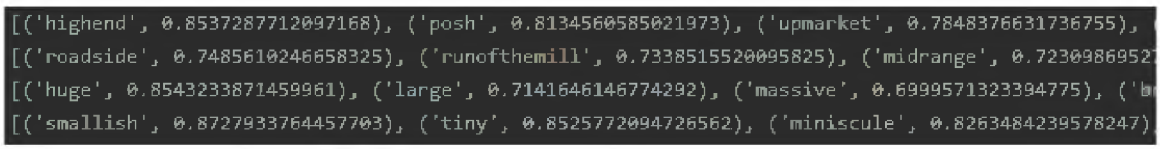


Figure 28: Word2Vec Model Similar Adjectives Output

5.5. Defining Feature Sets by Category and Method

This section covers the feature sets, how they are selected or derived, how they are preprocessed, and how they are incorporated. Appearing mainly in the second phase, the following factors are selected based on the unique information they represented and the relevance to the property recommendation system. The features come from the Land Use Data, derived geospatial information, Hotspot Analysis calculations, and the People Data. They are incorporated using natural language processing. This is done so the system can provide more accurate and user-specific recommendations based on their individual needs.

5.5.1. Land Use Characteristic Feature Set

The Land Use Data provides the LUCs and the city variables. The LUCs determine how the given property is designed, whether it be for a bank, gas station, or multi-family home, and the city determines the general area in the county. The first is selected because it provides a direct description of how the property is primarily used. Other variables are highly correlated with specific types of properties, such as the variable indicating how many safes there are being tied to banks, but many more are vague, so the LUC is the optimal choice. Similarly, the city variable is also straightforward. It is chosen over other variables describing the location, such as the full address, because when a user is looking to start a business or find a home, they start with the location. In large a city will hold properties of all shapes and sizes, but the cities vary widely in terms of attractions and proximity to points of interest, so this variable takes precedence.

5.5.2. Derived Geospatial Information Feature Set

Other key factors also come from the Land Data, but the geospatial information must be derived from them first for them to be made useful. The property lot size, which indicates the square footage, and the tract number, which is the way in which cities are subdivided, are two of these variables. The first on its own is useful, but by dividing the values with respect to the LUC types, it is possible to see the average sizes for each property type. Normalizing all of them together leads to recommendation issues since factories, malls, and parks tend to be far larger than car washes, gas stations, and single-family homes. Therefore, the relevant LUC average lot sizes are calculated to be used when appropriate. The tract numbers, on the other hand, are methodically determined

using the census tract maps and the Cuyahoga County websites. This is done because some cities are so much larger than others that it is impossible to generalize them into a single description. For example, the neighborhoods of Cleveland, such as Little Italy, Midtown, and The Flats, are all quite unique despite their proximity. Therefore, including the tracts is needed to account for these minute differences to tailor recommendations more closely.

Another important geospatial feature to be derived is a distance to the nearest property of the same land use type from a given candidate location. To identify the nearest distance, each distance needs to be derived from the GPS location of each candidate position to the location of each property of the same land use type in each target track.

5.5.3. Hotspot Analysis Feature Set

The third category includes the Hotspot Analysis factors which provided the PAI values, a numerical reflection of the most and least concentrated areas for different types of properties. This information is vital as it shows exactly where the best areas currently exist for every type of business. The reason there are so many of one type of property in one specific area is not pure random chance, but rather because of other latent factors, such as nearby residential areas, relative distance to popular tourist attractions, ease of access, and history. The PAI values show firsthand which areas work and which areas do not according to current information, so by incorporating these values into the system, they can set it apart from other systems.

5.5.4. People Demographic Feature Set

From the People Data, the populations and average incomes for every tract is calculated and used. The population count, which is split across the residential people, RAC, and working people, WAC, demographics, is quite useful for businesses that rely heavily on foot traffic. Some businesses such as farms are fine without it, though, so using the variables can further tailor suggestions to worthwhile areas based on the needs of the user. Additionally, the average incomes of the people living in each area is also an aid in determining which areas are best because higher end establishments, such as nice restaurants and luxury hotels, should be built in neighborhoods that can afford those amenities. Conversely, areas with a lower average income are more suited for fast food chains and motels. Simply put, a business will fail if there are no potential customers within a reasonable distance of it.

5.5.5. Natural Language Processing Incorporation

Finally, the NLP portion is where everything is tied together. From the user input, their unique requirements are derived which indicate which variables matter and which can be ignored. For example, when a user is looking for a property, they are likely to give the name of an area, so the natural language processor in phase four was made to recognize those entities and act accordingly. If the user specifies the name of a city, then that value is saved as a factor. Other times the user may specify a quality they are looking for in a property, so the system is also programmed to handle that. Regardless if the user prefers a larger or smaller property, the system pulls the information and determines the factors accordingly. The user may even simply list off what they want, such as the

number of bedrooms and bathrooms, so the model is also prepared to handle that and generate factors to further filter the potential candidates.

5.6. Ranking Model

The following section covers the final two phases of the system, the candidate generator and the ranking function. Here the process of selecting the candidates based on the derived and normalized weights as well as the algorithm to determine said weights are listed in detail. It is at these steps that all the heterogenous data preprocessing, geospatial data analysis, and NLP analytics come to a single point to produce the Top-*N* recommendations.

5.6.1. Candidate Generation

After the user inputs their natural language query, the next step is to generate the candidates from the large database of potential properties. To that end using natural language text processing, the LUC, city, adjectival descriptors, and the other factors, the candidates are determined. This is done through data pipelining and ends with an SQL query being created and used to generate a table of potential candidates.

Moreover, starting from the preprocessed Land Use Data and People Data from the second phase in conjunction with the determined relevant factors from the fourth phase, the candidate generator phase takes the data and determines the properties that fit the needs of the user. It works by selecting tuples from the finalized input data and narrowing the numbers down based on the determined relevant factors leaving only the potential candidates. Each of these candidates have all the qualities the user is looking for, so, for example, if they want a home in Strongsville, then only properties in Strongsville will be considered. Each relevant factor also has with it a relevant weight

which is a direct reflection of the accuracy of the factor. This means that if the user is looking for a small property, then the smallest properties will be given the highest weights and vice versa. This is to make sure the model does not simply single out the absolute smallest property as the most ideal recommendation, but rather to have the model treat all factors as relevant. One property might be larger in lot size, but if it more than makes up for it through the other factors, then it will be considered as a potential recommendation. Regardless, at this phase the candidates are selected based on the determined factors, the relevant weights are summed together, and the information is saved into a candidate table for the ranking function.

To be more specific, the first step is to determine the LUC from a potential list using POS tagging. The user input is scanned for any tokens that have a direct object tag as that is tag that determines the base type of property, such as an office building. From there up to two of the neighboring words before the direct object are analyzed to see if they are tagged as a compound or adjectival modifier. This is to differentiate single or double story office buildings from ones with three or more floors, which each have a different LUC code. Regardless, if the tokens are found, then the terms are concatenated together into a single string. The final step is to calculate the similarity between the final string and the list of potential LUCs. The one with the highest match is returned and saved for future use.

Next the city is derived using a tactic very similar to NER, more specifically by studying *N*-grams, which are strings of tokens of *N*-length. Given the names of the cities of Cuyahoga County, only unigrams, bigrams, and trigrams needed to be considered, *N*-grams of length one, two, and three, respectively. This is done by dividing the user input

into the corresponding lengths of terms and then searching through a list of potential city names for a match. To prevent cities such as "Cleveland" being named as the final city when the correct answer is "East Cleveland", the unigrams are searched first followed by the bigrams and then the trigrams. Doing this forces the model to find the unigram "Cleveland" first, but then the city value is overwritten after it finds the bigram "East Cleveland". In the event a city is not specified, the model opts to create recommendations on which city and tracts are the most ideal instead of individual properties.

Afterwards the adjectival descriptors are determined, which are values related to the quality and size of a given property. This is the section that separates the restaurants into the high-end ones and the low-end ones, and it is done through text analysis via the Word2Vec model. The Word2Vec model generates lists of positive and negative adjectives relating to quality and size separately to speed up the calculations by not having to retrain the model each time. Moreover, the user input text is searched for lemmas, also known as the dictionary or root form of a word, that are adjectives. If they relate to the quality or size, then they determine whether, for example, larger or smaller lot sizes are considered.

The last step is to determine the other miscellaneous factors, like the number of beds and baths for when a user is searching for a house or apartment. Like the first step, POS tagging is used only this time it searches for nouns. If the particular words are found, then the numeric tokens before them are saved as the corresponding value. This approach works well because when users are searching for quantities of an item, they put the quantity before the noun.

With the variables are analyzed and determined, the only remaining step is to generate the SQL query from the results and to process it to generate the candidate table. This table only contains properties that match all the user requirements and is used later to generate the Top-*N* recommendations based on the scoring function.

5.6.2. Scoring Function

A ranking model is one of the fundamental problems in all areas of information retrieval. Given a query and collection of documents that match the query, the problem is to sort the documents in the collection according to some measure so that the best results appear first in the results list when shown to the user [50].

The final stage of the recommendation system, the ranking model, is where the candidates are finalized and returned using the scoring function. It starts with the candidate table from the previous phase that contains all the information of the potential recommendations as well as the sum of their relevant weights. This information is sorted on those weights so that the most ideal properties appear first in the list. Additionally, the weights are normalized to give the user a better understanding of their meanings. Instead of having weights that range from one obscure number to another, all the finalized normal weights are bound between zero and one. After that is finished, the Top-*N* recommendations from the candidate list with the highest weights are returned to the user for evaluation.

The scoring function, precisely, aims to rank each of the candidates in the table using weights related to five of the variables, specifically the PAI LUC value, the lot size, the average income, the RAC population, which is the population of the residents in the area, and the WAC population, which is the population of the workers in the area. These

weights are preprocessed normalized values and only the relevant ones are selected which is entirely dependent on the natural language user input. Meaning if the user does not specify anything about the size of a property, then the associated weight, in this case the lot size weight, will not be included. Furthermore, if a user specifies that they want a higher quality area, then only the candidates with an above average income level will be considered. The logic here is that higher quality areas demand higher average incomes since quality is not cheap. Conversely, if a user wants a lower income neighborhood, then only the areas with an average income below the median will be considered and the weights will be subtracted from one so that the cheapest neighborhoods appear first.

These concepts are carried throughout every variable and the individual weights are summed together to create each candidate weight, W_i, as shown in Equation (9). The weights w_{PAI}, w_{Size}, w_{Income}, w_{RAC}, and w_{WAC} refer to the PAI analysis, lot size, average tract income, RAC population, and WAC population factors, respectively, with C referring to the total number of candidates. The w_{PAI} value is calculated by normalizing the count of a given LUC type in a specified tract, $n_{LUC \in T}$, and first dividing it by the total number of instances of that LUC, n_{LUC}. Next, that value is divided by the area of the tract, a_T, over the area of the county, which is the summation of all the tract areas. The lot size weight, w_{Size}, is the most straightforward since it comes simply from normalizing the lot size variable of each candidate, C_{Size}. Moreover, w_{Income} comes from the average income variables of the People Data which are recorded at a block level, B_{Income}, so the values of the blocks, B, belonging to a tract, T, were summed together into tract levels. From there the values are divided by the number of blocks within a tract, $n_{B \in T}$, and normalized into a weight format. In a similar fashion, the w_{RAC} and w_{WAC} variables are also grouped into

$$W_i = \sum_{i=1}^{C} w_{PAI} + w_{Size} + w_{Income} + w_{RAC} + w_{WAC} \tag{9}$$

$$\text{where } w_{PAI} = \frac{\frac{n_{LUC \epsilon T}/n_{LUC}}{a_T/\sum_{i=1}^{T} a_T} - \min\left(\frac{n_{LUC \epsilon T}/n_{LUC}}{a_T/\sum_{i=1}^{T} a_T}\right)}{\max\left(\frac{n_{LUC \epsilon T}/n_{LUC}}{a_T/\sum_{i=1}^{T} a_T}\right) - \min\left(\frac{n_{LUC \epsilon T}/n_{LUC}}{a_T/\sum_{i=1}^{T} a_T}\right)}$$

$$w_{Size} = \frac{C_{Size} - \min(C_{Size})}{\max(C_{Size}) - \min(C_{Size})}$$

$$w_{Income} = \frac{\frac{\sum_{i=1}^{B \epsilon T} B_{Income}}{n_{B \epsilon T}} - \min\left(\frac{\sum_{i=1}^{B \epsilon T} B_{Income}}{n_{B \epsilon T}}\right)}{\max\left(\frac{\sum_{i=1}^{B \epsilon T} B_{Income}}{n_{B \epsilon T}}\right) - \min\left(\frac{\sum_{i=1}^{B \epsilon T} B_{Income}}{n_{B \epsilon T}}\right)}$$

$$w_{RAC} = \frac{\sum_{i=1}^{B \epsilon T} B_{RAC} - \min(\sum_{i=1}^{B \epsilon T} B_{RAC})}{\max(\sum_{i=1}^{B \epsilon T} B_{RAC}) - \min(\sum_{i=1}^{B \epsilon T} B_{RAC})}$$

$$w_{WAC} = \frac{\sum_{i=1}^{B \epsilon T} B_{WAC} - \min(\sum_{i=1}^{B \epsilon T} B_{WAC})}{\max(\sum_{i=1}^{B \epsilon T} B_{WAC}) - \min(\sum_{i=1}^{B \epsilon T} B_{WAC})}$$

$$\widehat{W_i} = \frac{W_i - \min(W_i)}{\max(w_i) - \min(W_I)} \tag{10}$$

tract levels, but these values, the B_{RAC} and B_{WAC} ones which represent the respective populations of the given blocks, are added together and normalized to determine the relative population weights. It is important to note that the weight inversions and omissions are absent for simplicity. Namely, in Equation (8) it is assumed that the user provides positive specifications for all five weights. From there the candidate weights are then normalized again into their final values, $\widehat{W_i}$, using Equation (10) to convert them into forms more akin to percentages which are easier for users to understand.

CHAPTER VI

EXPERIMENTS

This section covers the five experiments performed with the property recommendation system. In order they are about analyzing the utilities of each city in Cuyahoga County, performing Hotspot Analysis on the LUC variable, performing K-means clustering also on the LUC variable, testing the recommendation system with six different natural language input queries, and evaluating the system by comparing to real estate websites. There is a different subsection covering each experiment which explains the premise, set up, and results.

6.1. Utility Map Hotspot Analysis

The first experiment done is to see if there is any meaningful information that can be derived from the utility variables in the Land Data. In total four are listed, the electricity, gas, sewer, and water utilities of each property. To carry out the experiment, the GeoJSON files are subset into cities and converted into shapefiles, a geospatial file format. From there the shapefile polygons are plotted on a map with their color and tooltip information indicating the type of utility. This process is repeated for all sixty cities across all four utilities with the results for the City of Lakewood being shown in Figure 29. The results from this work show that hotspot maps indicate that most

properties have the same types of utilities with exceptions being made for places such as parks or storage facilities.

Figure 29: Lakewood Electricity (top left), Gas (top right), Sewer (bottom left), and Water (bottom right) Utilities

Additionally, the SiteCat1 variable, which indicates the primary usage of the property, is also mapped out for each city. This one provides more useful derived information because it showcases how cities are commonly laid out. For example, Figure 30 demonstrates how commercial properties are almost entirely built near both residential properties and main city streets. Conversely, government properties appear to be built more randomly, perhaps to cover more ground, while the industrial properties are heavily grouped to one corner, likely because of the noise and pollution they emit. Perhaps while not too surprising, this pattern does repeat throughout each of the cities.

Figure 30: Lakewood Property Site Categorization

6.2. Land Use Code by Hotspot Analysis

Next the LUC variable from the Land Use Data is investigated further by using Hotspot Analysis on a few select property types. The ones chosen are office buildings, restaurants, single family dwellings, and warehouses because they represent the different categories of commercial, residential, and industrial types. Moreover, each LUC hotspot PAI analysis value is calculated for each type and for each tract. The results are then mapped out with the blue areas indicating cold spots and the red areas indicating hotspots, which can be seen in Figure 31. From this some interesting observations are drawn such as the fact that most residential areas are located just outside Cleveland city limits and that most warehouses are found on the east side of the county. Additionally, the office buildings and restaurants both seem to occupy the same areas, so that may indicate that those areas are ideal for commercial settings of all kinds.

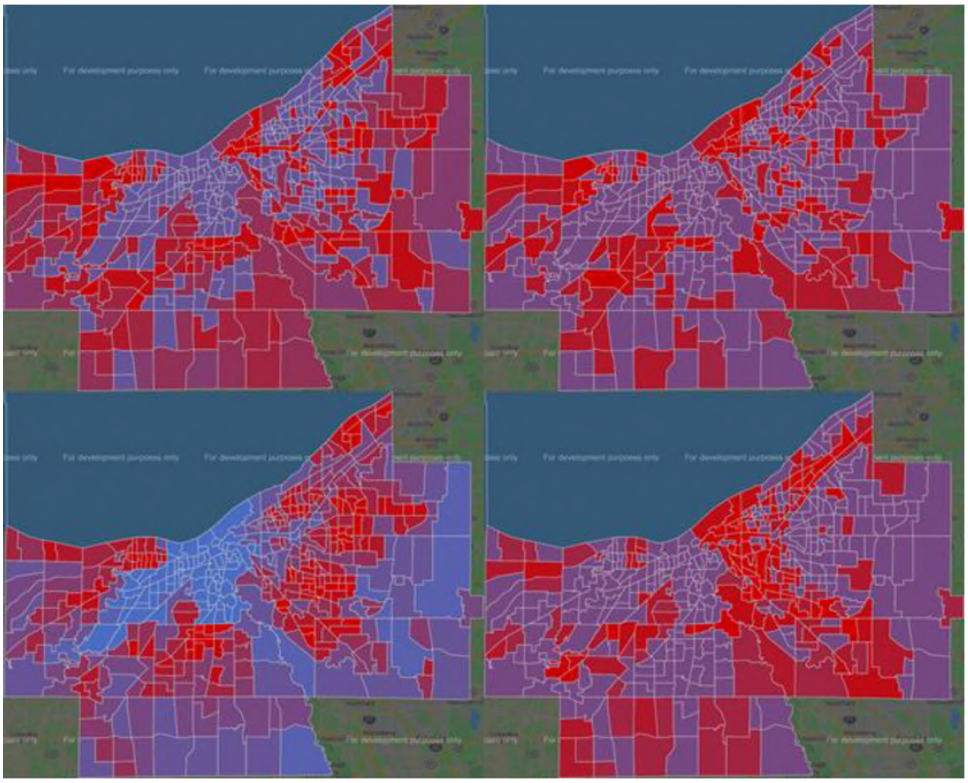


Figure 31: Hotspot Analysis of Cuyahoga County Office Buildings (top left), Restaurants (top right), Single Family Dwellings (bottom left), and Warehouses (bottom right)

6.3. Land Use Code by K-Means Clustering

The third experiment done also used the LUC, but this time it is analyzed using KMC. It is performed twice, once on all commercial properties and once on all government properties, with a wide variety in the number of clusters. The graphs and maps for both tests can be seen below in Figure 32 and Table 4 for the commercial ones and Figure 33 and Table 5 for the government ones. This analysis is done to observe how the properties in Cuyahoga County are laid out and to see if there are any notable patterns that would otherwise be difficult to detect. The results reinforced the findings from the utility map Hotspot Analysis, namely that the commercial properties are always located off main city streets and that the government properties are more random and spread out. The former observation is so true that the commercial map looks just like a street map.

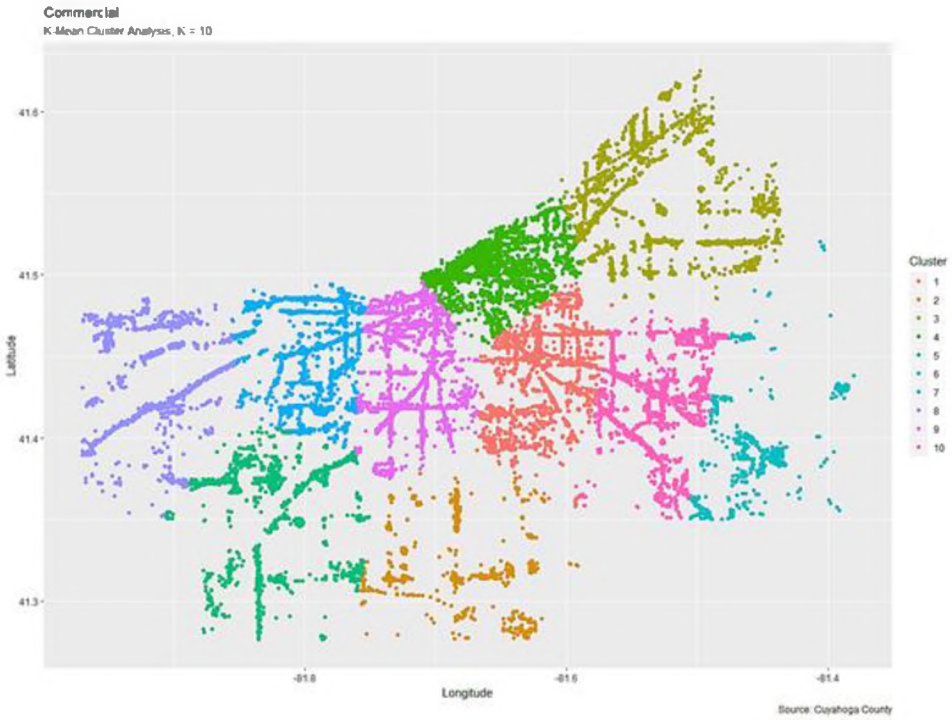


Figure 32: K-Means Clustering on Commercial Properties

Table 4. K-Means Clustering Results for Commercial Properties

Commercial (N = 23337)	K = 2	K = 3	K = 4	K = 5	K = 7	K = 10
N in Cluster 1	9826	6196	5952	3299	5083	2584
N in Cluster 2	13511	7556	5942	5513	4646	766
N in Cluster 3	-	9585	4037	4080	3215	3214
N in Cluster 4	-	-	7406	7051	2081	4260
N in Cluster 5	-	-	-	3394	2477	1651
N in Cluster 6	-	-	-	-	3247	815
N in Cluster 7	-	-	-	-	2588	2622
N in Cluster 8	-	-	-	-	-	1539
N in Cluster 9	-	-	-	-	-	3287
N in Cluster 10	-	-	-	-	-	2599
Accuracy	56.2%	70.3%	75.7%	82.3%	87.5%	91.5%

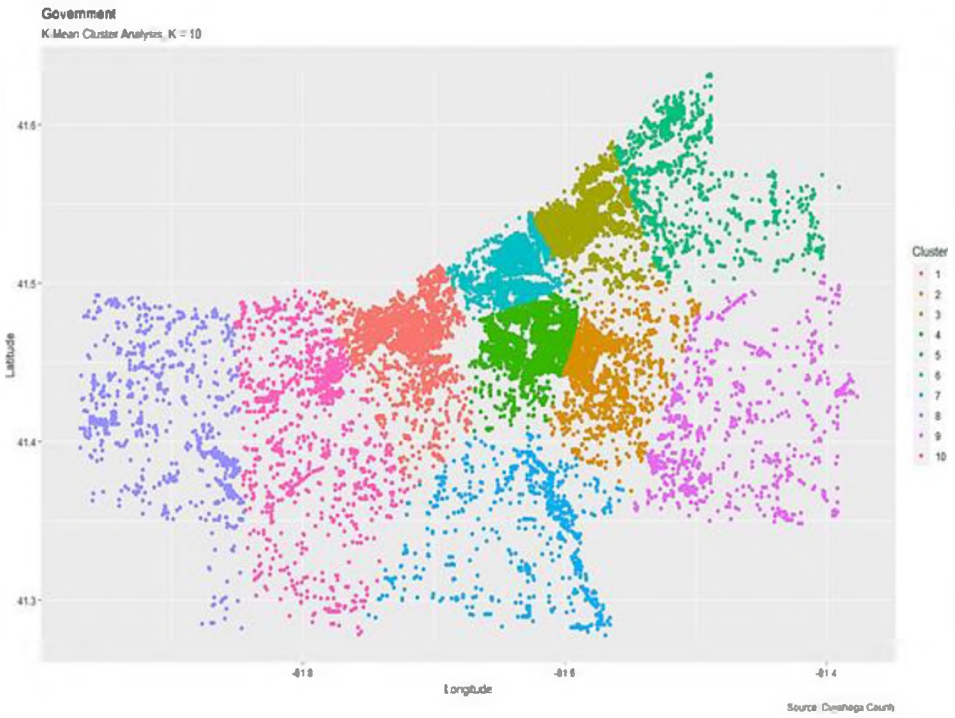


Figure 33: K-Means Clustering on Government Properties

Table 5. K-Means Clustering Results for Government Properties

Government (N = 28673)	K = 2	K = 3	K = 4	K = 5	K = 7	K = 10
N in Cluster 1	6637	5230	16374	5350	2295	3453
N in Cluster 2	22036	21123	4686	1481	3436	2600
N in Cluster 3	-	2320	2262	3791	840	4200
N in Cluster 4	-	-	5351	6267	2244	5787
N in Cluster 5	-	-	-	11784	7837	1805
N in Cluster 6	-	-	-	-	7579	5647
N in Cluster 7	-	-	-	-	4442	775
N in Cluster 8	-	-	-	-	-	1413
N in Cluster 9	-	-	-	-	-	1136
N in Cluster 10	-	-	-	-	-	1857
Accuracy	51.8%	61.2%	71.9%	77.5%	86.3%	90.9%

6.4. Ranking Generation with Examples

To showcase the results of the recommendation system, nine natural language user input tests are created. The nine examples can be split into three sets of three where every example in each set is more complex than the last. Namely, the examples start out simple and progressively become more complex as new factors and weights are added.

The first three examples are about creating recommendations for a user looking for a place to open an Italian restaurant, the second group of three are about a user looking to start a detached retail store business with no location in mind, and the third group of three are about finding the most ideal two-family home when given several requirements.

These are the first three examples tested from the first group: 1-1) "I want to open an Italian restaurant in Strongsville." 1-2) "I want to open a high-end Italian restaurant in Strongsville." 1-3) "I want to open a small high-end Italian restaurant in Strongsville." All three ask the system to produce recommendations for a property within Strongsville that is ideal for a restaurant setting. Additionally, the second and third examples add conditions further narrowing down the potential lots based on its quality and size. The scoring function is tailored for each example and they are shown in order in (11), (12), and (13). More information on each equation can be found in Section 5.6.2. Furthermore, the determined Top-*N* recommendations can be seen in Tables 6, 7, and 8.

$$W_i = \sum_{i=1}^{C} w_{PAI} + w_{RAC} + w_{WAC} \tag{11}$$

Table 6. Recommendations for Example 1-1

Parcel	Tract	City	LUC	Lot Size	Weight
39617020	186201	Strongsville	Restaurant	56323	1.000000000
39701005	186201	Strongsville	Restaurant	24825	1.000000000
39624004	186201	Strongsville	Restaurant	103498	1.000000000
39236013	186103	Strongsville	Restaurant	44431	0.859437572
39212018	186103	Strongsville	Restaurant	50000	0.859437572
39236012	186103	Strongsville	Restaurant	38364	0.859437572

$$W_i = \sum_{i=1}^{C} w_{PAI} + w_{Income} + w_{RAC} + w_{WAC} \tag{12}$$

Table 7. Recommendations for Example 1-2

Parcel	Tract	City	LUC	Lot Size	Weight
39617020	186201	Strongsville	Restaurant	56323	1.000000000
39701005	186201	Strongsville	Restaurant	24825	1.000000000
39624004	186201	Strongsville	Restaurant	103498	1.000000000
39236013	186103	Strongsville	Restaurant	44431	0.725550066
39212018	186103	Strongsville	Restaurant	50000	0.725550066
39236012	186103	Strongsville	Restaurant	38364	0.725550066

$$W_i = \sum_{i=1}^{C} w_{PAI} + (1 - w_{Size}) + w_{Income} + w_{RAC} + w_{WAC} \quad (13)$$

Table 8. Recommendations for Example 1-3

Parcel	Tract	City	LUC	Lot Size	Weight
39701005	186201	Strongsville	Restaurant	24825	1.000000000
39617020	186201	Strongsville	Restaurant	56323	0.998529389
39236012	186103	Strongsville	Restaurant	38364	0.725363440
39236013	186103	Strongsville	Restaurant	44431	0.725072382
39212018	186103	Strongsville	Restaurant	50000	0.724811961

The second group of three examples are as follows: 2-1) "What is a good area to open a detached retail store?" 2-2) "What is a good area to open a small detached retail store?" 2-3) "What is a good area to open a cheap small detached retail store?" All three are about a user asking for recommendations for a detached retail store, but because the user never specifies a city, the system returns location suggestions instead of property ones. The tailored scoring functions for each example can be seen in (14), (15), and (16) with the results stored in Tables 9, 10, and 11.

$$W_i = \sum_{i=1}^{C} w_{PAI} + w_{RAC} + w_{WAC} \quad (14)$$

Table 9. Recommendations for Example 2-1

Tract	City	LUC	Weight
117700	Cleveland	Detached Retail	1.000000000
177104	Parma	Detached Retail	0.881180840
181100	Rocky River	Detached Retail	0.742612839
177303	Parma	Detached Retail	0.679629896
107701	Cleveland	Detached Retail	0.664700810
190504	Olmsted Township	Detached Retail	0.645604183

$$W_i = \sum_{i=1}^{C} w_{PAI} + (1 - w_{Size}) + w_{RAC} + w_{WAC} \quad (15)$$

Table 10. Recommendations for Example 2-2

Tract	City	LUC	Weight
117700	Cleveland	Detached Retail	1.000000000
177104	Parma	Detached Retail	0.881171483
181100	Rocky River	Detached Retail	0.742622960
177303	Parma	Detached Retail	0.679643846
107701	Cleveland	Detached Retail	0.664710141
152400	Euclid	Detached Retail	0.631073381

$$W_i = \sum_{i=1}^{C} w_{PAI} + (1 - w_{Size}) + (1 - w_{Income}) + w_{RAC} + w_{WAC} \quad (16)$$

Table 11. Recommendations for Example 2-3

Tract	City	LUC	Weight
119402	Cleveland	Detached Retail	1.000000000
152400	Euclid	Detached Retail	0.948754797
121700	Cleveland	Detached Retail	0.864447901
141602	Cleveland Heights	Detached Retail	0.827700475
119502	Cleveland	Detached Retail	0.800953556
115800	Cleveland	Detached Retail	0.781096252

The final three examples are as follows: 3-1) "I need a two family house with 2 bedrooms and 1 bathroom in Cleveland Heights." 3-2) "I need an upscale two family house with 2 bedrooms and 1 bathroom in Cleveland Heights." 3-3) "I need a small upscale two family house with 2 bedrooms and 1 bathroom in Cleveland Heights." These three are about a user looking for a specific type of home in Cleveland Heights with the additional factors of bedroom and bathroom counts added. The tailored scoring functions for each example are shown in (17), (18), and (19) with the results stored in Tables 12, 13, and 14.

$$W_i = \sum_{i=1}^{C} w_{PAI} + w_{RAC} + w_{WAC} \quad (17)$$

Table 12. Recommendations for Example 3-1

Parcel	Tract	City	LUC	Lot Size	Weight
68604031	141400	Cleveland Heights	Two family dwelling	45097	1.000000000
68604032	141400	Cleveland Heights	Two family dwelling	59000	1.000000000
68606001	141400	Cleveland Heights	Two family dwelling	250652	1.000000000
68621005	141400	Cleveland Heights	Two family dwelling	125197	1.000000000
68622027	141400	Cleveland Heights	Two family dwelling	37100	1.000000000
68512040	141200	Cleveland Heights	Two family dwelling	25200	0.807267386

$$W_i = \sum_{i=1}^{C} w_{PAI} + w_{Income} + w_{RAC} + w_{WAC} \quad (18)$$

Table 13. Recommendations for Example 3-2

Parcel	Tract	City	LUC	Lot Size	Weight
68604031	141400	Cleveland Heights	Two family dwelling	45097	1.000000000
68604032	141400	Cleveland Heights	Two family dwelling	59000	1.000000000
68606001	141400	Cleveland Heights	Two family dwelling	250652	1.000000000
68621005	141400	Cleveland Heights	Two family dwelling	125197	1.000000000
68622027	141400	Cleveland Heights	Two family dwelling	37100	1.000000000
68512040	141200	Cleveland Heights	Two family dwelling	25200	0.965027212

$$W_i = \sum_{i=1}^{C} w_{PAI} + (1 - w_{Size}) + w_{Income} + w_{RAC} + w_{WAC} \quad (19)$$

Table 14. Recommendations for Example 3-3

Parcel	Tract	City	LUC	Lot Size	Weight
68622027	141400	Cleveland Heights	Two family dwelling	37100	1.000000000
68604031	141400	Cleveland Heights	Two family dwelling	45097	0.997495241
68604032	141400	Cleveland Heights	Two family dwelling	59000	0.993287246
68512040	141400	Cleveland Heights	Two family dwelling	25200	0.968840798

6.5. Evaluation of the Property Recommendation System

The final experiment is about trying to evaluate the property recommendation system results by comparing them with real-life real estate recommendation systems. To that end, the sites LoopNet for commercial real estate and Howard Hanna for residential real estate were chosen because they are the leading real estate companies with recommendation systems. The evaluations are carried out by using some of the examples from the previous subsection. For each example, the website search parameters are matched as closely as possible to provide the most level comparisons.

On the LoopNet website, example 1-3 was used, namely "I want to open a small high-end Italian restaurant in Strongsville". The search parameters set on the website were Strongsville for the city, restaurant for the property type, and under 60,000 square feet for the lot size. All other parameters were kept at their default values. Pictured in Figure 34 are the parameters set and the results, which was only one former Arby's fast food restaurant building which is in Middleburg Heights, a city to the north of

Strongsville. Changing the search from properties for sale to properties for lease, four results appeared, however none were within Strongsville.

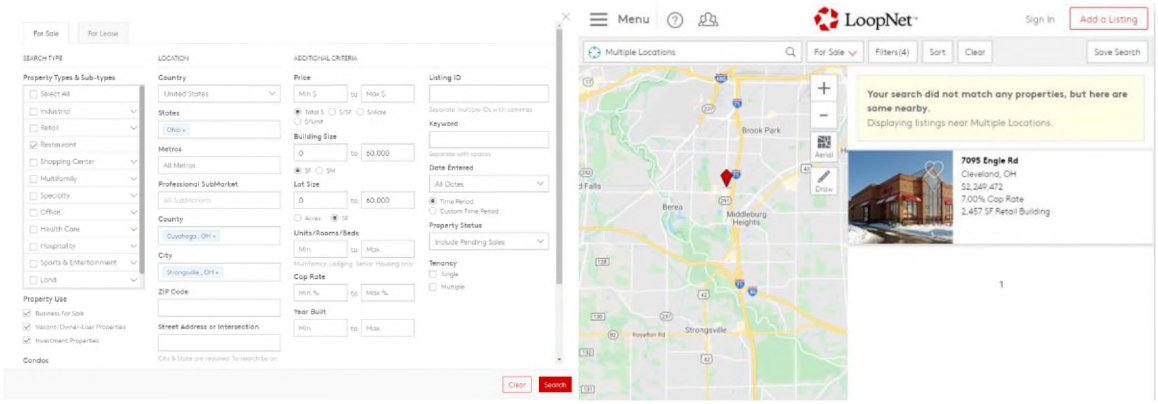


Figure 34: LoopNet Property Search Parameters and Results

As for the Howard Hanna website, example 3-3 was used, namely "I need a small upscale two-family house with 2 bedrooms and 1 bathroom in Cleveland Heights". The parameters this time were Cleveland Heights for the city, multi-family home for the property type, less than one acre for the lot size, two or more beds, and one or more full baths, as seen in Figure 35 alongside the results. Contingent and pending listings were also included in the search. The website returned two recommendations, only one of which was in Cleveland Heights, the other being in Cleveland. Both properties also included eight bedrooms and four bathrooms.

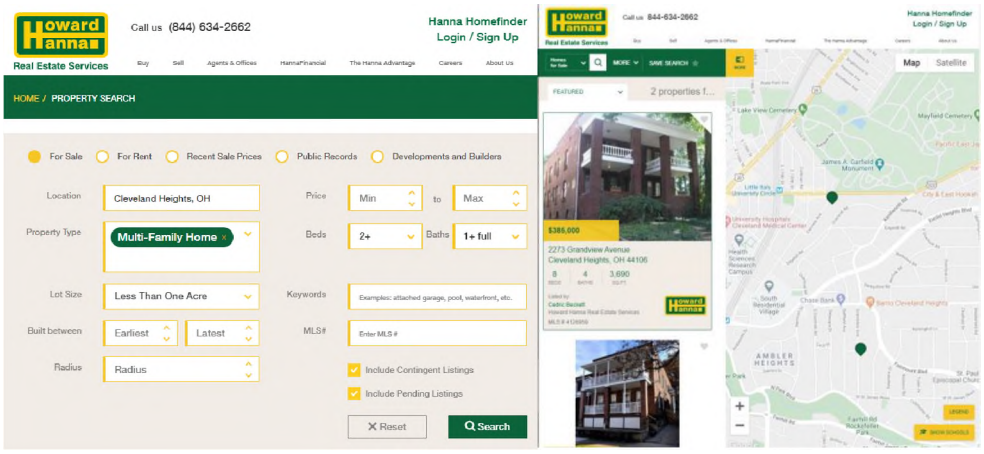


Figure 35: Howard Hanna Property Search Parameters and Results

Despite these comparison efforts, it was not possible to find any significant search results from the real estate company websites with which to make comparisons. The reason being that those websites do not return any meaningful search results are namely, first because of the current COVID-19 emergency pandemic situation, there are far fewer listings than there would be otherwise. And second, the real estate websites try only to find empty properties from the current listings while the property recommendation system finds all of the existing matches, ranks them, and suggests the best properties. After all, it is an item-to-item collaborative filtering-based recommendation system. Evidence that this system provides superior recommendations can be found within the visualizations. The recommended properties for the query "high-end Italian restaurant in Euclid" shown below in Figure 36 consists of locations along the major commercial streets of Euclid. This figure indicates that the system considers geospatial knowledge when identifying the properties on a major street in commercial areas. More information about the visualization process is present in the next chapter.

Figure 36: Visualization of Commercial Recommendations in Euclid

CHAPTER VII

WEB APPLICATION

7.1. Property Recommendation System as a Web Application

Below is the architecture of the web application for the property recommendation system. The system uses a web application in which this system is integrated as an additional service to interact with the users, as pictured in Figure 37 which shows an overview of the data communication pipelining. It starts with the end user client inputting the natural language string which will be sent to the Node.js Server. That server then sends the data to the Python Property Recommendation Application Server which works with its backend RDBMS database to produce the Top-*N* recommendations. Those results are passed back to the Node.js server which sends it to the MongoDB server. That server then gathers the relevant data from the Preprocessed GeoJSON Relational Database and passes all the relevant information back to the first server which in turn sends it to the End User Client. There, on the web page using Leaflet Map API, an open-source JavaScript library, the results are visualized for the user.

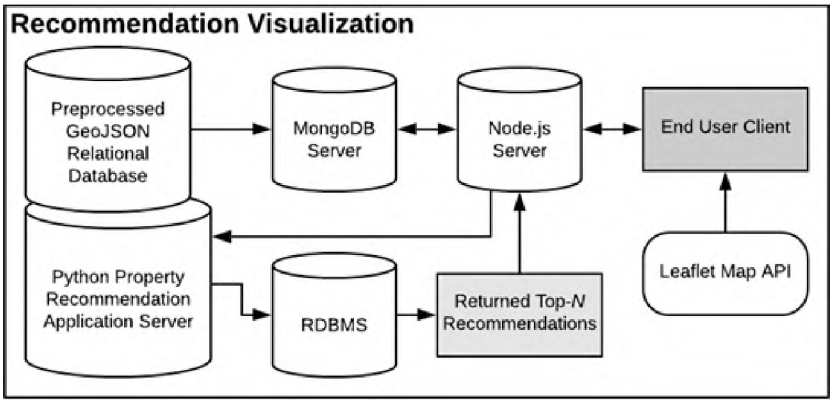


Figure 37: Framework Overview of the Web Application for the Property Recommendation System

7.2. User Interface of the Property Recommendation System

On the website users can input query specifications via the user interface when searching for a home. Pictured in Figure 38, there are five categories that can be specified besides the natural language input query. These are the type of house, the price range, the lot size, the number of bedrooms, and the number of bathrooms. The first allows the user to specify if they desire a single-family home, a two-family home, or a three-family home. Below that it is possible to indicate the ideal price range for a property and the area size in acres. The final two categories are for naming the number of bedrooms and bathrooms the user wants. After hitting the filter button, the results are displayed, an example of which can be seen in Figure 39.

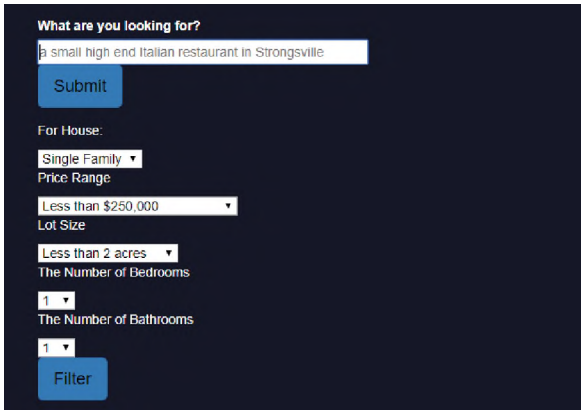


Figure 38: Website for the User Input Interface

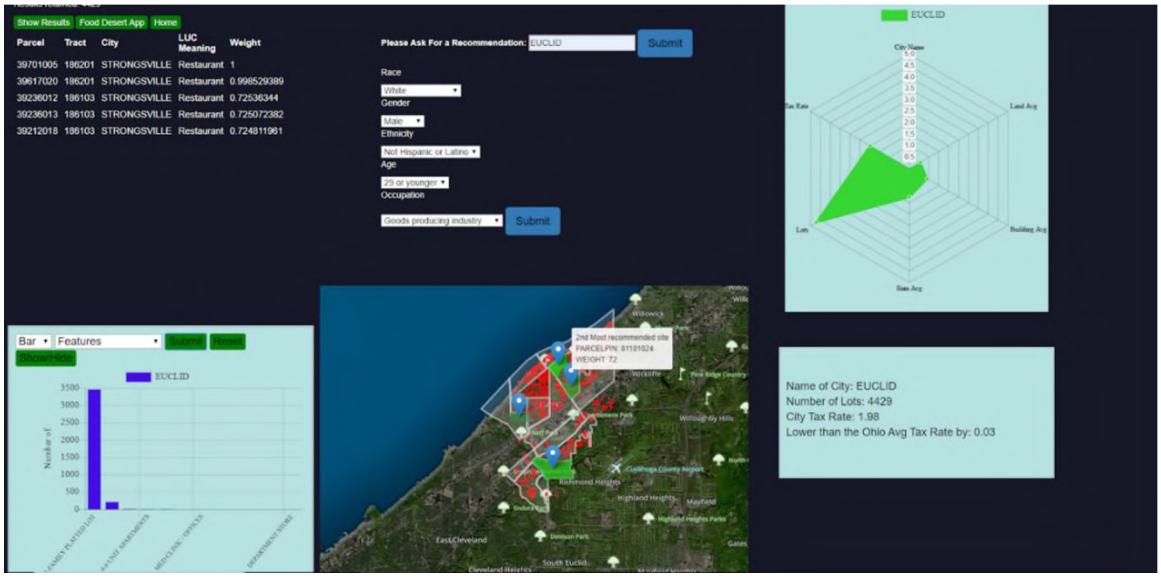


Figure 39: Property Recommendation System as a Web Service

CHAPTER VIII

CONCLUSION AND LIMITATION

The methodologies to build a property recommendation system that does not rely on user history data were studied in this thesis. This study explored the research in building a recommendation system using big data analytic methods to derive relevant key factors from complex structures of geospatial data with Hotspot Analysis and K-Means Clustering. The recommendation system also adopted text analysis methods using Natural Language Processing techniques Part-of-Speech tagging, Named Entity Recognition, and Word2Vec models to allow and analyze a user query in a natural language. A well-defined candidate ranking model was developed to score each candidate to return the final Top-*N* ranked property or area recommendations that are most likely to be in line with the needs of a user per given interest in a fixed set of locations.

The model uses a pipelined framework which includes a Word2Vec model to generate similar adjectives, a natural language processor to analyze the user input, a candidate generator to select potential properties, and a ranking function to score each candidate to generate accurate recommendations based on the relevant factors from the derived weights. The experiment results show that this framework offers effective methodologies that can be expanded upon to additional areas within the United States.

The challenges surrounding a lack of user history data were discussed and overcome through derived geospatial factors and an adaption of an item-to-item collaborative filtering model concept with natural language text processing analysis. Other limitations of this work were the lack of real-life examples and data to use to test and evaluate the property recommendation system and the lack of other similar recommendation systems with which to make comparisons.

CHAPTER IX

FUTURE WORK

The property recommendation system is built from data within the Land Use Data and People Data, but not all variables are considered. Expanding the model to include more factors may increase the refinement of the results, assuming each additional variable is unique enough to not produce bias within the system by having multiples of the same value. Second, incorporating long short-term memory units (LSTMs) or gated recurrent units (GRUs), both of which are artificial recurrent neural networks (RNNs), within the natural language text analysis phase could prove worthwhile in elevating the system to be able to handle more complex user inputs, such as paragraphs instead of single phrases or sentences. These RNNs would be able to generate a term list of only the important words and would automatically stem away the rest. Finally, expanding the concept to other counties and states would not only allow for a wider user base, but it would also potentially lead to additional variables which could be converted into weights. Regardless, Cuyahoga County is somewhat unique in Ohio because it is home to a major city. Analysis of smaller counties and their smaller populations could provide contrast and depth to the recommendations. Perhaps the Hotspot Analysis values discussed are relative and adding in other counties would scale everything.

REFERENCES

[1] "Analyzing Block Group Demographic-Economic Patterns." *ProximityOne*, proximityone.com/blockgroups.htm.

[2] "APPENDIX A FIPS State and County Codes." https://www.census.gov/prod/techdoc/cbp/cbp95/st-cnty.pdf.

[3] "A Tutorial on Clustering Algorithms." https://home.deib.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html.

[4] Bhavani, Durga. "Understanding Named Entity Recognition Pre-Trained Models." *V-Soft Consulting*, blog.vsoftconsulting.com/blog/understanding-named-entity-recognition-pre-trained-models.

[5] Bird, Steven, et al. *Natural Language Processing with Python*. 1st ed., O'Reilly, 2009.

[6] Brownlee, Jason. "What Is Natural Language Processing?" *Machine Learning Mastery*, 22 Sept. 2017, machinelearningmastery.com/natural-language-processing/.

[7] "Census Blocks and Block Codes." *ProximityOne*, proximityone.com/geo_blocks.htm.

[8] "Census Tracts." *United States Census Bureau*, https://www2.census.gov/geo/pdfs/education/CensusTracts.pdf.

[9] Chainey, Spencer, et al. "The Utility of Hotspot Mapping for Predicting Spatial Patterns of Crime." Security Journal, vol. 21, no. 1-2, 29 Jan. 2008, pp. 4–28., doi:10.1057/palgrave.sj.8350066.

[10] Collobert, Ronan, et al. “Natural Language Processing (Almost) from Scratch.” *Journal of Machine Learning Research*, Aug. 2011.

[11] Covington, Paul, et al. “Deep Neural Networks for YouTube Recommendations.” *Proceedings of the 10th ACM Conference on Recommender Systems – RecSys 16*, 2016, doi:10.1145/2959100.2959190.

[12] Cussens, James. “Bayesian Network Learning with Cutting Planes.”

[13] “Cuyahoga County Open Data.” *Cuyahoga County Open Data*, data-cuyahoga.opendata.arcgis.com/.

[14] Deng, Yao et al. “基于 LBSN 用户生成短文本的细粒度位置推测技术 (Fine-grained Geolocalisation of User Generated Short Text Based on LBSN).” 计算机科学 46 (2019): 316-321.

[15] Du, Jia, et al. “A Group Recommendation Approach Based on Neural Network Collaborative Filtering.” *2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW)*, 2019, doi:10.1109/icdew.2019.00-18.

[16] Dwarampudi, Mahidhar, and N. Reddy. “Twitter Sentiment Analysis Using Distributed Word and Sentence Representation.” 1 Apr. 2019.

[17] Elbadrawy, Asmaa, and George Karypis. “User-Specific Feature-Based Similarity Models for Top-N Recommendation of New Items.” *ACM Transactions on Intelligent Systems and Technology*, vol. 9, no. 4, 20 Mar. 2013, pp. 1–20., doi:10.1145/2700495.

[18] "Federal Information Processing System (FIPS) Codes for States and Counties." *Federal Communications Commission*, https://transition.fcc.gov/oet/info/maps/census/fips/fips.txt.

[19] Finkel, Jenny Rose, et al. "Incorporating Non-Local Information into Information Extraction Systems by Gibbs Sampling." *43rd Annual Meeting on Association for Computational Linguistics*, 2005, doi:10.3115/1219840.1219885.

[20] "FORMATS for CHARACTER FIELDS CHARACTERISTICS (APPRAISAL INVENTORY) FILE." 21 May 2008.

[21] Fu, Yanjie. "Research Statement." 2016, pp. 10–13.

[22] "GeoJSON." *GeoJSON*, geojson.org/.

[23] "GIS Data." *GIS Data – Cuyahoga County Information Services Center*, isc.cuyahogacounty.us/en-US/GIS-Data.aspx.

[24] Gupta, Mohan. "A Review of Named Entity Recognition (NER) Using Automatic Summarization of Resumes." *Medium*, Towards Data Science, 9 July 2018, towardsdatascience.com/a-review-of-named-entity-recognition-ner-using-automatic-summarization-of-resumes-5248a75de175.

[25] Han, Jiawei, et al. Data Mining: Concepts and Techniques. Morgan Kaufmann, 2012.

[26] Hore, Prodip, and Sayan Chatterjee. "A Comprehensive Guide to Attention Mechanism in Deep Learning for Everyone." 20 Nov. 2019.

[27] Hu, Jianfeng, and Bo Zhang. "Product Recommendation System." *CS224W Project Report*, 2012.

[28] Hunt, Joel M. "Do crime hot spots move? Exploring the effects of the modifiable areal unit problem and modifiable temporal unit problem on crime hot spot stability." (2016).

[29] "Information Extraction and Named Entity Recognition." *Stanford University*, https://web.stanford.edu/class/cs124/lec/Information_Extraction_and_Named_Entity_Recognition.pdf.

[30] Jurafsky, Daniel, and James H. Martin. "Part-of-Speech Tagging." *Speech and Language Processing*, 2 Oct. 2019.

[31] Kang, Wang-Cheng, and Julian McAuley. "Candidate Generation with Binary Codes for Large-Scale Top-N Recommendation." *Proceedings of the 28th ACM International Conference on Information and Knowledge Management – CIKM 19*, Nov. 2019, doi:10.1145/3357384.3357930.

[32] Keim, Daniel A., et al. "Pixel Based Visual Data Mining of Geo-Spatial Data." *Computers & Graphics*, vol. 28, no. 3, 2004, pp. 327–344., doi:10.1016/j.cag.2004.03.022.

[33] Koren, Yehuda. "Collaborative Filtering with Temporal Dynamics." *Communications of the ACM*, vol. 53, no. 4, 2010, pp. 89–97., doi:10.1145/1721654.1721677.

[34] Kou, Yufeng, et al. "Spatial Weighted Outlier Detection." *Proceedings of the 2006 SIAM International Conference on Data Mining*, 2006, doi:10.1137/1.9781611972764.71.

[35] Kristina Toutanova, et al. "Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network." *HLT-NAACL*, 2003, pp. 252-259.

[36] Li, Lei, et al. "Context-Aware Co-Attention Neural Network for Service Recommendations." *2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW)*, 2019, doi:10.1109/icdew.2019.00-11.

[37] Lifchitz, Alain, et al. "Effect of Tuned Parameters on an LSA Multiple Choice Questions Answering Model." *Behavior Research Methods*, vol. 41, no. 4, 2009, pp. 1201–1209., doi:10.3758/brm.41.4.1201.

[38] Linden, Greg, et al. "Amazon.com Recommendations: Item-to-Item Collaborative Filtering." *IEEE Internet Computing*, vol. 7, no. 1, 2003, pp. 76–80., doi:10.1109/mic.2003.1167344.

[39] Ma, Hao, et al. "SoRec: Social Recommendation Using Probabilistic Matrix Factorization." *Proceeding of the 17th ACM Conference on Information and Knowledge Mining – CIKM 08*, 2008, doi:10.1145/1458082.1458205.

[40] Manning, Christopher, and Socher Richard. "Natural Language Processing with Deep Learning." *Stanford University*, 2017.

[41] Manning, Christopher D., et al. 2014. "The Stanford CoreNLP Natural Language Processing Toolkit." *52nd Annual Meeting of the Association for Computational Linguistics*, pp. 55-60.

[42] "Mapping Crime: Understanding Hot Spots." *United States Department of Justice Office of Justice Programs*, Aug. 2005, https://www.ncjrs.gov/pdffiles1/nij/209393.pdf.

[43] "Maps & Digital Imaging Department." *Cuyahoga County Fiscal Officer*, https://fiscalofficer.cuyahogacounty.us/en-US/map-room-digital-imaging.aspx.

[44] McCormick, Chris. "Word2Vec Tutorial – The Skip-Gram Model." *Chris McCormick*, 19 Apr. 2016, mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/.

[45] Mikolov, Tomas, et al. "Distributed Representations of Words and Phrases and Their Compositionality." 16 Oct. 2013.

[46] Mikolov, Tomas, et al. "Efficient Estimation of Word Representations in Vector Space." 7 Sept. 2013.

[47] Mikolov, Tomas, et al. "Linguistic Regularities in Continuous Space Word Representations." Association for Computational Linguistics, June 2013, pp. 746–751., https://www.aclweb.org/anthology/N13-1090.

[48] Ott, Myle. "Hotel-Review Datasets." *Carnegie Mellon University*, www.cs.cmu.edu/~jiweil/html/hotel-review.html.

[49] "Part-of-speech tagging." *Stanford University*, https://web.stanford.edu/class/cs124/lec/postagging.pdf.

[50] Piccoli, Gabriele, and Federico Pigni. *Information Systems for Managers: With Cases*. 4.0 ed., Prospect Press, 2019.

[51] "Real Property Information." *Real Property Information – Delinquent Land Tax*, fiscalofficer.cuyahogacounty.us/en-US/REPI.aspx.

[52] Santorini, Beatrice. "Part-of-Speech Tagging Guidelines for the Penn Treebank Project." 15 Mar. 1991.

[53] Sasaki, Yutaka. "The Truth of the F-Measure." 26 Oct. 2007.

[54] Sun, Yizhou. "Vector Data: Clustering: Part I." 26 Apr. 2017, *Lecture Notes of UCLA, CA*, April 2017.

[55] Tan, Lianzhi, et al. "Attention Based Term Weighting for App Retrieval." *Thirty-Third AAAI Conference on Artificial Intelligence*, 2018.

[56] Tan, Pang-Ning, et al. Introduction to Data Mining. Pearson, 2015.

[57] Tian, Yuan. "Towards the Development of Best Data Security for Big Data." *Communications and Network*, vol. 09, no. 04, 30 Nov. 2017, pp. 291–301., doi:10.4236/cn.2017.94020.

[58] Tong, Yuzhen, et al. "Collaborative Generative Adversarial Network for Recommendation Systems." *2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW)*, 2019, doi:10.1109/icdew.2019.00-16.

[59] *United States Area Boundary, Data, Graphs, Tools and Services*, www.usboundary.com/.

[60] "United States Census Bureau Center for Economic Studies Publications and Reports Page." *United States Census Bureau*, lehd.ces.census.gov/data/.

[61] United States Census Bureau, "LEHD Origin-Destination Employment Statistics (LODES) Dataset Structure Format Version 7.4." *LEHD Origin-Destination Employment Statistics (LODES) Dataset Structure Format Version 7.4*, United States Census Bureau, 22 Aug. 2019.

[62] United States Census Bureau, "NORTH AMERICAN INDUSTRY CLASSIFICATION SYSTEM." *NORTH AMERICAN INDUSTRY*

CLASSIFICATION SYSTEM, Executive Office of the President Office of Management and Budget, 2017.

[63] United States Census Bureau, 2010, https://www2.census.gov/geo/maps/dc10map/tract/st39_oh/c39035_cuyahoga/DC10CT_C39035_001.pdf.

[64] United States Census Bureau, 2010, https://www2.census.gov/geo/maps/dc10map/tract/st39_oh/c39035_cuyahoga/DC10CT_C39035_002.pdf.

[65] United States Census Bureau, 2010, https://www2.census.gov/geo/maps/dc10map/tract/st39_oh/c39035_cuyahoga/DC10CT_C39035_003.pdf.

[66] Wang, Dawei, et al. "Crime Hotspot Mapping Using the Crime Related Factors—A Spatial Data Mining Approach." *Applied Intelligence*, vol. 39, no. 4, Aug. 2012, pp. 772–781., doi:10.1007/s10489-012-0400-x.

[67] "Web Mapping Application." *Cuyahoga County*, https://gis.cuyahogacounty.us/html5viewer/?viewer=cegis.

[68] "What Are GZ Files, Gzip Compressed Format." *What Are GZ Files, Gzip Compressed Format*, www.peazip.org/gzip-files.html.

[69] "What Is Hotspot Analysis?" *Geospatiality*, 21 Jan. 2016, https://glenbambrick.com/2016/01/21/what-is-hotspot-analysis/.

[70] Wu, Chao-Yuan, et al. "Recurrent Recommender Networks." *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining – WSDM 17*, Feb. 2017, doi:10.1145/3018661.3018689.