Institut für Maschinelle Sprachverarbeitung
University of Stuttgart
Pfaffenwaldring 5B
D-70569 Stuttgart

Diploma Thesis

Predicting Sentiment about Places of Living

Feifei Liu

Course of Study: Informatik

Examiner: Prof. Dr. Sebastian Padó

Supervisor: Dr. Christian Scheible

Commenced: August 04, 2016

Completed: February 03, 2017

CR-Classification: I.2.7, I.5.4, J.4, J.5

Abstract

Nowadays studies about the quality of life in major cities are often published in the daily news. These contain ranked list according to the quality of living with indicators representing various aspects. Typical indicators are crime level, transport, health care etc. Along with the flourishing of different social medias, a huge amount of information could be collected from the Internet. Moreover, machine learning as a branch of artificial intelligence becomes more and more prominent. The recent advances in machine learning had found usage in a wide range of applications. One of such application is that of text categorization and sentiment analysis. Relying on these conditions, this thesis aims to create a classifier to predict the sentiment about places of living.

In this thesis a ranking list of cities of Mercer is taken use. As a result of the quality of living survey 230 cities of the world are ranked in the list. Text form information of microblogging is chosen as our testbed. Specifically, tweets, microblogging messages from the popular website Twitter, are studied. The tweets chosen for this study are those about cities living standard and contain rich sentiment information. Classification label is assigned to cities under study by their position in the ranking list. After sentiment related features are extracted, machine learning techniques are then applied on the collected tweets. As a result, a classifier with a strong baseline for predicting sentiment about places of living is trained using logistic regression model.

Contents

List of Figures

List of Tables

List of Algorithms

Listings

1 Introduction

1.1 Motivation

Usually, there are many reviews of the quality of life of cities could be seen in news or on the Internet. People concerns more and more about the quality of life in a city. Based on the rich information from these resources numerous studies about the quality of life in major cities are made every year. Regarding various aspects of daily life in a city such as the crime rate, health care, entertainment and satisfaction, public transport etc., and relying on analysis of this textual data through various process technologies many study results are published in a form of ranking list, of which city is good for living or not.

The information about the life of a city can be collected in a textual form from various of data sources, such as news, articles, Twitter, government reports, Wikipedia etc. The basic idea of this study is that based on the ranking list of the cities and the large collectible textual information about the quality of life in a city, is that possible to take use of machine learning algorithm making computer to predict the life quality of a city automatically?

On one hand, since the quality of life of a city is always combined with the satisfaction of people, whereas the microblogging is considered as online word-of-mouth branding, which contains rich sources of data about people's opinions on different aspects of life. Table 1.1 shows the tweets that the words like"love","good moment" and "fabulous show" are used to express the emotion like glorification and so on.

Table 1.1: Examples of Tweets

City	Tweet
Paris	"The next time I come to Paris it will be with my man. **love** boyfriend gay paris france @mateoitis https:// instagram.com/p/8RijRysUu4/"
Beijing	Moderate pollution (39) at 5AM. Really low for Beijing . **Good moment** to go running at Summer Palace http:// buff.ly/1Usm44V airpollution
London	"What a **fabulous show**it was...!!!! Soldout in three days times London . Great job done by kudosmusic and... http:// fb.me/4z6056iOs"

Moreover Twitter offers officially API which enables the user to collect information from tweets easily, collecting the textual data on Twitter could be seen as a collection of sentiment about a city held by its inhabitants. Thus textual information from Twitter are chosen as the testbed of this thesis.

On the other hand, since 1990's machine learning as a subfield of artificial intelligence developed rapidly. The classification technologies cooperated with natural language processing enable the computer to process textual data, explore the data and learn from the text.

Aggregating these features this thesis tends to train a classifier, which could automatically predict the sentiment of living in a city through applying machine learning technology on collected information from Twitter.

All sorts of the text form information of tweets are considered as study objects, in which the most important is the words appeared in tweets, URLs, hashtags and emoticons are also taken into account. A ranking list of cities as a result of a survey on quality of living in 2016 is exploited, cities on the top position of the ranking list are considered as positive sentiment for living and cities on the bottom of the ranking list are considered as negative sentiment for living. Through this way the ranking problem is casted into a two sentiment classification problems.

1.2 Objectives and Benefites

This project addresses the challenge of creating a classifier for predicting sentiment about places of living, associated with managing large volumes of textual data from the Internet. In particular, the task of this thesis focuses on the classification technologies of machine learning and the natural language processing of textual data. Therefore, main objectives of this thesis are listed as the following:

- collecting the necessary data from Twitter, analyzing the structure of text data, with respect of linguistic analysis processing the data and generating the dataset.
- extracting features and applying suitable classification technologies to training models.
- performing the trained classifier on the test data, comparing the results and evaluating the classifier with proper measure method.
- enhancing the performance of the trained classifier through analyzing feature weights and improving the features, based on the feature weights analyzing the aspects of living standard which impact the classification model.

This Study will deliver benefits on following aspects:

- collecting data of cities from Twitter and generating a dataset for training classification models, which are not provided or published by official organizations and also could be reused for further research.
- conducting an automatic classification system for predicting a place of living, based on microblogging information from a social platform, which is not presented in previous and ongoing works.

- contribution to a more economic classifier for automatically predicting the changes of quality of life, since the cost of conducting frequent surveys on qualifying the city life is expensive.
- investigating analysis about the information of the most and the least impact to predicting of the city of live through training classifier, which also has its realistic significance. From this side of view, it enables us to do further analyze on which social aspects may of most importance or influence more on the live in a city or of more interests of users.

1.3 Outline

This diploma thesis is structured in 8 chapters. As seen in figure 1.1 the remaining 7 chapters are presented as following:

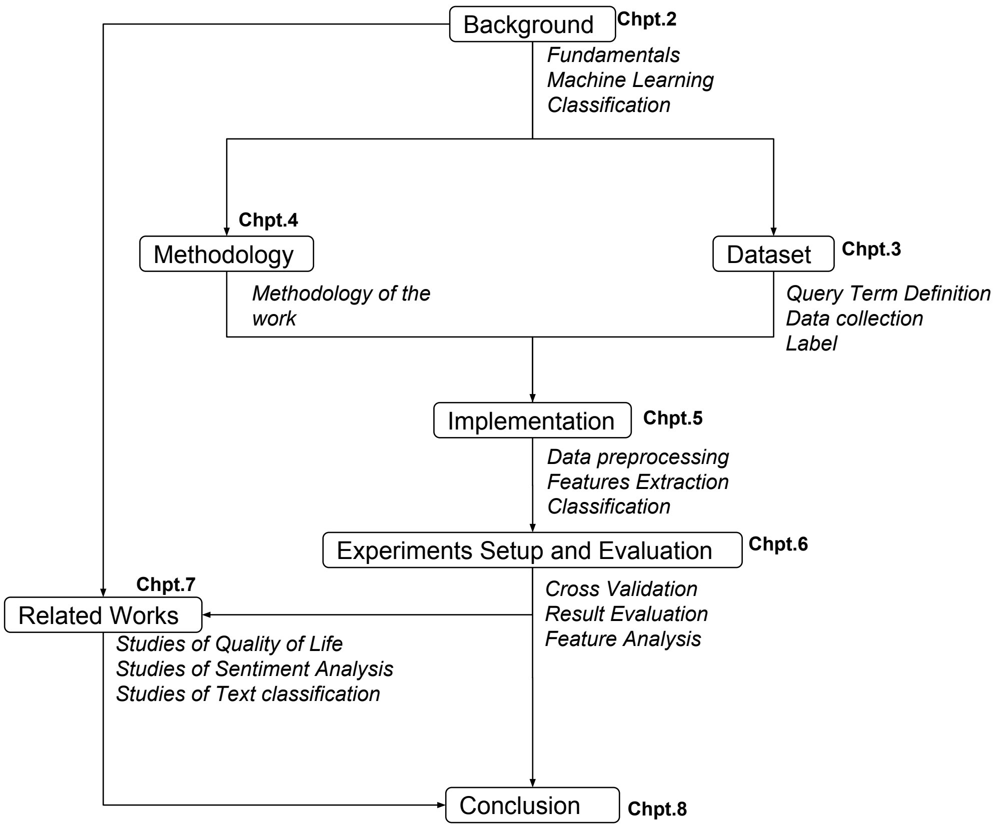


Figure 1.1: Thesis Layout

- **Chapter 2 – Background:** The fundamentals of machine learning will be explained, the algorithms of classification, which are ground stones of this thesis will be presented, and the principle of evaluating a classifier will be interpreted in this chapter, which related studies about classification could be seen in chapter 7.

- **Chapter 3 – Dataset:** Data collection process will be presented, particular on query terms definition, data selection and marking labels.
- **Chapter 4 – Methodology:** The workflow of machine learning process and the methodology specific for creating classification model for predicting will be presented.
- **Chapter 5 – Implementation:** Based on the specific methodology how the classification model is implemented is described step by step.
- **Chapter 6 – Experiments Setup and Evaluation:** First introduces the experiments setup used in this thesis and then the results of the performance of classifiers is compared, at last the improved features and enhanced results of classifiers are presented.
- **Chapter 7 – Related Works:** Introduces the previous works, which concerns more on the sentiment of microblogs and also related with the life quality of places.
- **Chapter 8 – Conclusion:** A summary of this thesis is concluded and the suggestions on how to further improve the results in the future work are presented.

2 Background

This chapter introduces the technical fundamentals about machine learning and classification. With respect to the characteristics of the thesis, the theoretical training models are explained and the TwitterAPI used for collecting data is introduced.

2.1 Machine Learning

Machine learning is a subfield of computer science that involved from the study of pattern recognition and computational learning theory in artificial intelligence.[1] There are sorts of definition of machine learning in which [MFH$^+$13] provided a more formal and widely quoted definition: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E."

The principle of machine learning is to explore the study and construction of algorithms that can learn from and make predictions on data. A model is built by executing the algorithms from an example training set of input observations in order to make data-driven predictions or decisions expressed as outputs. Generally, there are three types of tasks of machine learning, depending on the nature of the learning "signal" or "feedback" available to a learning system in [RNC$^+$03]:

- **Supervised learning:** Simply to say is the task of inferring a function from labeled training data.

  Specifically supervised learning is a type of machine learning which refers to give the algorithm a data set in which the "right answers" were given. The "right answers" are called labels of the data, which are marked by the specialist. [Anz12] In supervised learning there are two main terminologies which present two types of problems it solves:

  - **Regression:** The goal of regression is to predict the value of one or more continuous *target* variables t given the value of a D-dimensional vector $\mathbf{x}$ of input variables. The variables t is presented simply the vector of real numbers whose values wish to be predicted. [Anz12]
  - **Classification:** The goal in classification is to take an input vector $\mathbf{x}$ and to assign it to one of K discrete classes C_k where $k = 1, ..., K$. In the most common scenario, the classes are taken to be disjoint, so that each input is assigned to one and only one class. In contrast, there are many ways of using target values to represent class labels. For probabilistic models, in the case of

[1] `https://www.britannica.com/technology/machine-learning`

two-class problems is the binary representation of t the most convenient with $t \in \{0, 1\}$ where $t = 0$ represents class C_1 and $t = 1$ represents class C_2.

- **Ranking:** Learning to rank is a task to automatically construct a ranking model using training data, such that the model can sort new objects according to their degrees of relevance, preference, or importance. Ranking algorithms in machine learning are called "learning-to-rank"methods which specifically learn how to combine predefined features for ranking. [L+09]

- **Semi-supervised learning:** Compared to supervised learning semi-supervised learning model is trained through a given training set with some of the target outputs missing.

- **Unsupervised learning:** In contrast to supervised learning, unsupervised learning works with a dataset which no labels are given to the learning algorithm, leaving it on its own to find structure from the input. The most common algorithm to solve this problem is clustering.

2.2 Classification

2.2.1 Classifiers for text classification

[Seb02] introduced the main ideas underlying the Machine Learning approach to text classifications. In particular, many classifiers are theoretically presented: Probabilistic Classifiers in which Naïve Bayes approaches is one of the best-known method, Decision Tree Classifiers, Decision Rule Classifiers, On-Line Methods, The Rocchio Method, Neural Networks, Example-Based Classifiers which includes the k-NN method, Support Vector Machine.

The following three classifiers are most used algorithms for processing text:

- **k-NN Classifier:** This classifier is memory-based, and require no model to be fit. Given a query point x_0, the k training points $x_{(r)}$, $r = 1, ..., k$ closest are found in distance to x_0, and then classify using majority vote among the k neighbors. [JL10]

- **Näive Bayes:** This is a popular technique that especially appropriate when the dimension p of the feature space is high, making density estimation unattractive. The naive Bayes model assumes that given a class $G = j$, the features X_k are independent:[JL10]

$$f_j(X) = \prod_{k=1}^{p} f_{jk}(X_k) \tag{2.1}$$

  While this assumption is generally not true, it does simplify the estimation dramatically:

  - The individual class-conditional marginal densities f_{jk} can each be estimated separately using one-dimensional kernel density estimates. This is, in fact a generalization of the original naive Bayes procedures, which used univariate Gaussians to represent these marginals.
  - If a component X_j of X is discrete, then an appropriate histogram estimate can be used. This provides a seamless way of mixing variable types in a feature vector.

- **MaxEnt:** Stands for maximum entropy classifier, which prefers the most uniform models that also satisfy any given constraints.[NLM99]

2.2.2 Logistic Regression

Despite of the name Logistic Regression is a linear classification model rather than regression.[Anz12] It is also known as logit regression or maximum-entropy classification or the log-linear classifier in the literature. This model takes use of probabilities to describe the possible outcomes of a single trial which are modeled with a logistic function.

- **Logistic function:** Logistic function is a common "S" shape with formula:

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}} \tag{2.2}$$

  where e is the natural logarithm base, x_0 is the x-value for sigmoid's midpoint, L is the curve's maximum value and k is the steepness of the curve. x is a real number range from negative infinite to positive infinite.

- **Sigmoid function:** special case of logistic function which is also called standard logistic function, where $L = 1, k = 1, x_0 = 0$ with formula:

$$S(t) = \frac{1}{1 + e^{-t}} \tag{2.3}$$

  Since the sigmoid function has a standard "S" shaped curve, which means that the function can take any real input whearas the output always distributed between zero and one.

- **Logistic Regression:** The goal of logistic regression is to find the parameters that best fit the formula:

$$y = \begin{cases} 1 & \beta_0 + \beta_1 x + \varepsilon > 0 \\ 0 & else \end{cases} \tag{2.4}$$

  where ε represent an error distributed.

  For linear logistic regression with the real input t ($t \in R$), taking use of the standard logistic function (2.3) and assuming t is a linear function of a single explanatory variable x, which is expressed as:

$$t = \beta_0 + \beta_1 x \tag{2.5}$$

and then logistic function can now be written as a function of x:

$$F(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}} \tag{2.6}$$

where the terms are :

- $F(x)$ is the probability of the dependent variable equaling "1", which also means "success" rather than "unsuccess" or "0", since the distribution of probability of $P(Y_i|X)$ differs from X_i.
- β_0 is the intercept from the linear regression equation (the value of the criterion when the predictor is equal to zero)
- β_1 is the regression coefficient, which is multiplied by some value of the predictor.
- e denotes the exponential function.

Furthermore, as an extension the linear function t could be of multiple variables x with multiple factors β, where is presented as vector θ^T. Thus the function $F(x)$ could be presented as:

$$F(x) = \frac{1}{1 + e^{-(\theta^T x)}} \tag{2.7}$$

- **Cost function:** The cost function is used to measure the accuracy of hypothesis function h_θ. Ideally, the best situation is when the cost function equals to 0, which means the hypothesis of each input x is perfectly matched to each class label y.

$$Cost(h_\theta(x), y) = \begin{cases} -log(h_\theta(x)) & \textit{if } y = \textit{1} \\ -log(1 - h_\theta(x)) & \textit{if } y = \textit{0} \end{cases} \tag{2.8}$$

where $h_\theta(x)$ is the hypothesis Representation of logistic regression which is another form of (2.6), where θ is the collection of all parameters of the model such as β. F function is represented as a hypothesis function with h. In binary classification case, $Cost(h_\theta(x), y) = 0$ *if* $h_\theta(x) = y$, Thus the equation has:

- $Cost(h_\theta(x), y) \to \infty$ *if* $y = \textit{0}$ *and* $h_\theta(x) \to 1$
- $Cost(h_\theta(x), y) \to \infty$ *if* $y = \textit{1}$ *and* $h_\theta(x) \to 0$

which guarantees that the cost function of logistic regression is convex without waving, so that it is easy to reach its global optimal .

Until now the Logistic regression is introduced as linear using a linear function of input x like $\theta^T x$, whereas it can also be a nonlinear function.

- **Regularization:** Fitting the training data too well can lead to overfitting problem, which degrades the risk on future predictions.[JL10] Regularization is a method to reduces overfitting by adding a complexity penalty to the cost function. Using $\lambda\|\theta\|^2$ as a penalty the equation (2.8) can be presented as:

$$Cost(h_\theta(x), y) = \begin{cases} -log(h_\theta(x)) + \lambda\|\theta\|^2 & \textit{if } y = \textit{1} \\ -log(1 - h_\theta(x)) + \lambda\|\theta\|^2 & \textit{if } y = \textit{0} \end{cases} \tag{2.9}$$

where $\|\theta\|^2 = \theta^T\theta$ and λ is a parameter to tune the regularization strength, the bigger the value of λ the regularization degree is higher. Regularization may cause worse predicting result when choosing an inappropriate λ.

2.2.3 Support Vector Machine

Support Vector Machine is a very useful classification model. Comparing to the regular linear classification models, it has the advantage of dealing with infinite features dataset. The most used area of SVM is in Information Retrieval.

The support vector classifier is to find linear boundaries in the input feature space. As with other linear methods, its procedure can be more flexible by enlarging the feature space using basis expansions such as polynomials or splines. Generally linear boundaries in the enlarged space achieve better training-class separation, and translate to nonlinear boundaries in the original space. Once the basis functions $h_m(x), m = 1, ..., M$ are selected, We fit the support vector classifier using input features $h(x_i) = (h_1(x_i), h_2(x_i), ..., h_M(x_i)), i = 1, ..., N$, and produce the (nonlinear) function $f\hat{(x)} = h(x)^T\hat{\beta} + \hat{\beta_0}$, where the classifier is $\hat{G}(x) = sign(f\hat{(x)})$.

The support vector machine classifier is an extension of support vector classifier, where the dimension of the enlarged space is allowed to get very large, infinite in some cases. [JL10]

- **Kernel Function:** Used for computes inner products in the transformed space, K should be asymmetric positive (semi-) definite function.

$$K(x, x') = \langle h(x), h(x') \rangle \tag{2.10}$$

- **SVM polynomial kernel:** One of the popular choices for SVM, the kernel function K is a dth-Degree polynomial function:

$$K(x, x') = (1 + \langle x, x' \rangle)^d \tag{2.11}$$

- **SVM rbf kernel:** Uses radial basis function as kernel function K:

$$K(x, x') = exp(-\gamma\|(x - x')\|^2) \tag{2.12}$$

2.3 Evaluation of Classifiers

Evaluating classifiers is the end phase of the whole process of supervised machine learning, which tends to measure the performance achieved by a learning algorithm. In a real-world application of supervised learning, the purpose is to predict examples without known labels through applying the learning models trained with labeled examples. In contrast, the research needs a set of data with labels for evaluating a classifier experimentally. In this section the main concepts and evaluating methods are introduced:

- **The goal of Evaluation:** Evaluation of classifiers is typically conducted experimentally rather than analytically. The experimental evaluation of a classifier usually measures its effectiveness rather tan its efficiency, which means that the ability of a classifier to take the right classification decisions. Furthermore, there are also some issues need to be considered, the efficiency of an algorithm, robustness of a classifier and also the scalability. [Seb02]

- **Data set for Evaluation:** From section 2.3 introduced there are usually Training set with marked labels used for training and Test set also with labels independent on the training set used for the test. Sometimes the training set and test set are given already. Other times there is only one database contains all training data. In this case, the whole dataset needs to be divided into two datasets, experimentally take a certain percentage of data for training set (usually 70%) , and the left for testing (usually 30%).[Elk]

  Besides these two datasets, validation sets are usually used in some studies. During training models there are always some choices for the user to manipulate the settings of the parameters of an algorithm, then the results are compared and the best algorithm with optimized performance is picked for applying on the validation set. Because the performance which trained on training set is not always consistent with the result of applying on the validation set. A set of labeled examples which are used to pick settings of an algorithm is called a validation set. In this case, an independent test set is also necessary for evaluating classifier.

- **Cross Validation:** For test dataset, the most studies take use of a certain percentage (usually 70%) of the dataset to training models, and the left (30%) are used as test dataset to examine the performance of classifier experimentally. This approach is called train-and-test. An alternative way is the *k-fold* cross-validation, which is shown in algorithm 1.

  The whole dataset M is splitted into k equal parts, each time one part M_i is using as test data M_{test}, the left $M \setminus M_i$ then is used to train classifier, the test dataset M_i is rotated iteratively as $i = 1, 2...k$.

  If there are n labeled examples available, the largest possible number of folds is $k = n$. The special case is called leave-one-out cross-validation(LOOCV). However, the time complexity of cross-validation is k times that of running the training algorithm once, so often LOOCV is computationally infeasible. Experimentally the most common choice for k is 10. [Elk]

Algorithm 1 Cross Validation

```
procedure CrossValidation($M$, $S_M$, $k$)
    shuffling the rows of matrix $M$
    $a := 0$
    $b := S_M/k$
    for $i = 0$ to $i = k - 1$ do
        $M_i := M_{[a:b]}$
        $a := b$
        $b := b + b$
        $M_{test} := M_i$
        $M_{train} := M \setminus M_i$
    end for
end procedure
```

Table 2.1: The Utility Matrix of Two-Class Classifier

		Expert Prediction	
		positive	negative
Prediction of Classifier	positive	tp	fp
	negative	fn	tn

- **Evaluating methods:** There are different methods of measuring classifier's performance. In [Elk] a measuring method for two possible classes based on four basic numbers is introduced. The four basic numbers are obtained from applying the classifier to the test set, which called true positives tp, false positives fp, true negatives tn and false negatives fn. The sum of the four entries $tp+fp+tn+fn = n$, the number of whole test examples.

  A utility table is conducted based on these four basic numbers as table 2.3, where the four numbers in this context have the meanings:

  - True positive tp is the number of examples that are predicted as positive by the classifier, which is consistent with annotation of the expert.
  - False positives fp is the number of examples that are predicted as positive by the classifier, whereas they are not annotated in the positive class by the expert, which means inconsistent with the real annotation.
  - True negative tn is the number of examples that are predicted as negative by the classifier, which is consistent with annotation from expert.
  - False negative fn is similar to fp, is the number of examples that are predicted as negative by the classifier, which are inconsistent with real annotation of the expert.

Depending on the application, many different summary statistics are computed from these entries. In particular:

- calculating accuracy of the classifier on the whole evaluation dataset:

$$accuracy = \frac{N(correct\ classifications)}{N(all\ classifications)} \tag{2.13}$$

  here equals to

$$accuracy = \frac{(tp + tn)}{n} \tag{2.14}$$

  where $n = tp + tn + fp + fn$.

- calculating the precision of the classifier :

$$precision = \frac{tp}{(tp + fp)} \tag{2.15}$$

- calculating the recall of the classifier:

$$recall = \frac{tp}{(tp + fn)} \tag{2.16}$$

- further more measuring the accuracy across the classifier's decision:

$$decision = \frac{N(retrieved\ documents)}{N(all\ documents)} \tag{2.17}$$

  This function is used in [PP10] for document classification, where the classes are predefined and only retrieved documents are classified. From this point of view this function is similar like *accuracy*.

- Examine the impact of the dataset size on the performance of the classification system using F-measure [MS99]:

$$F = (1 + \beta^2)\frac{precision \cdot recall}{\beta^2 \cdot recall + recall} \tag{2.18}$$

  where β could be set different depending on which results people concerned more.

2.4 Twitter APIs

Many studies about sentiment analysis have chosen tweets from the platform of Twitter as research subjects, because the tweets are known of the following characteristics. [GBH09],[PP10],[JYZ+11],[MFH+13], [DWT+14] [VZ15]

- constraints of 140 characters
- expressions are flexible and sometimes without aspects.

- could be posted everywhere and at any time.
- rich of all sorts of information.
- easy to collect by using TwitterAPI.

This section introduces the Twitter API. For purpose of research, Twitter offers APIs to access their data for developers. There are Twitter Libraries for different programming languages, which enable us easily to collect data which are needed from Twitter.

2.4.1 OAuth

Before starting to gather the data from Twitter, the first requirement is to get authentication and authorization.

OAuth is an open protocol to allow secure authorization in a simple and standard method from web, mobile and desktop applications.[2] The OAuth 2.0 authorization framework makes it possible that the third-party application to obtain limited access to an HTTP service.

Since OAuth is secure for third-part and compatible with many libraries, Twitter chooses OAuth to send authorized requests to the Twitter API to access its API.

There are two models for authentication in Twitter:

- **User authentication:** It is the common way of authentication. The user signed request both identifies the applications identity and the users access token, which represents the identity accompanying permissions of the end-user.
- **Application-only authentication:** A manner of authentication where the application makes API requests on its own behalf without a user context. Not all API methods support this model since some methods require user context. The methods which support application-only authentication has two rate limits, one is per user and the other is per app.

2.4.2 Objects

There are four main "Objects" may be encountered in the API: Tweets, Users, Entities and Places.[3]

- **Tweets:** Tweets are messages posted on a user's page on Twitter, it is the fundamentals of Twitter and could be updated to change one status, embedded, replied, liked, disliked and deleted by user.
- **Users:** Users could be an individual, an organization, or even an automated system. They could tweet, create lists, follow someone or be followed by someone, mention or be mentioned by any user, moreover each user has a timeline.

[2] `https://oauth.net/`

[3] `https://dev.twitter.com/overview/api`

- **Entities:** Entities provide metadata and additional contextual information about content posted on Twitter, like hashtags with "#", media like photos and videos, user mentions with "@", and the URLs.
- **Places:** Places are specific, named locations with corresponding geo-coordinates. This information is not necessary required fields when tweeting, but it could be embedded with a tweet.

Each of the "Objects" contains its own fields, which are of different data types. Searching through setting these fields it is easy to get the needed information.

2.4.3 Connecting APIs

There are two types of APIs offered by Twitter, one is REST APIs, another is Streaming APIs.

- **REST APIs:** REST APIs allows user programmatic access to read and write Twitter data. It identifies Twitter applications and users using OAuth. All the responses are in JSON format.
- **Streaming API:** Streaming API offers samples of the public data flowing through Twitter, it enables monitoring or processing Tweets in real-time. Connecting to a streaming endpoint Twitter allows for each account only one standard connection, if the connecting request is sent again from the same account, the old connection will be released Once the connection is set up between applications and the streaming endpoint, a feed of Tweets delivered without REST API rate limits.

The Twitter API website offers a list of Twitter libraries of different platforms such as C, C++, ASP, .NET, Java, Javascript, Objective-C, Perl, PHP, Qt on HTML and so on, which support Twitter API. Official libraries are built and maintained by Twitter in Java. There are also a huge amount of libraries working with Python.

2.4.4 Limitations

As previously introduced through the object fields the needed information could be obtained by setting the query term when connecting Twitter APIs. For example, the timestamp of a user's tweet when is published is recorded in Twitter, then this tweet could be gathered by searching the user at this time point. From this point of view, it seems to be able to get all the needed useful information. Indeed there is a restriction not only for connecting REST APIs but also Streaming API.

Twitter official API denied accessing the tweets over one week. It means that the needed old data could not be gathered but only be collected from now on. That means if some needed tweets of one query term are of a period of time, the tweets could be only collected day by day or once a week since now, which costs lots of time.

As a solution, there is an application called "GetOldTweets" offered in GitHub, which enables us to bypass the problem programmatically. It has two versions one is built in Java and one is in Python.

The working principle of “GetOldTweets-python” is similar to the scenario of Twitter searching on browser. When browsing on Twitter, the page of Twitter is scrollable and the tweets of the page will be loaded continually along with scrolling down the websites. These tweets are shown up through calls to a JSON provider at the back-end [4]

The “GetOldTweets-python” has three main Components:

- **Tweet:** A class model contains specific members which enable us to construct tweet.
- **TweetManager:** A class with `getTweets()` method, which manages construct collected tweets in tweet class model.
- **TwitterCriteria:** A criteria use for accessing specific informations form tweets, in which `setUsername()` query via username is enabled, `setSince()` and `setUntil()` pairs for setting query time, `setQuerySearch()` for matching any query terms and `setMaxTweets()` setting the maximum numbers of Tweets need to be retrieved.

Otherwise “GetOldTweets” also has some weakness, it is unstable and handles errors by running out of work. So it is essential to supervise it for getting good results and dataset construction still involves a bit more manual labor.

[4] `https://github.com/drat/GetOldTweets-python`

3 Dataset

In this chapter, specifications about dataset are introduced: definition of query terms, the process of collecting data from Twitter, the problems we meet during this process and marking classification labels.

3.1 Query Terms

In this thesis, a ranking list from Mercer[1] is exploited. This list is a result of a scientific survey made by Mercer in 2016. 230 cities from different countries are evaluated for their quality of life. In figure 3.1 the distribution of all cities on ranking list is shown on the world map.

Since the ranking list has only 230 cities, although the ranking list is regarding various aspects of living in a city, for the original database, all information of a city are concerned, thus tweets need to be collected for each city separately and in this thesis they are stored separately in CSV files.

In section2.2.2 the tool for collecting tweets from Twitter is introduced. Taking use of the class `TwitterCriteria` any tweets with the predefined query terms and appropriate parameters can be accessed. The details about query processing are as following:

- City name: The exact "city name" in ranking list is taken use as our main query terms.
- Period: In order to make the original data are collected under as much as the same conditions, we have chosen the same period settings for each city, which is from 01.09.2015 to 31.08.2016. Because we don't know how many tweets will be collected for a year, the collection is processed month by month periodically.
- Query Form: Class `TwitterCriteria` offers these five methods for us. We use the definition in list 3.1 for collecting the tweets.

Listing 3.1: GetOldTweets Settings for Collecting Tweets

```
for index in range(230):
  n = Citylist[index]
  times = ["2015-09-01","2015-10-01","2015-11-01","2015-12-01","
       2016-01-01","2016-02-01","2016-03-01","2016-04-01","2016-05-01
       ","2016-06-01","2016-07-01","2016-08-01","2016-09-01"]
  tweetCriteria = got3.manager.TweetCriteria().setQuerySearch('#'+ n
        +" lang:en").setSince(times[i]).setUntil(times[i+1]).
       setMaxTweets(100000)
```

[1] https://www.imercer.com/content/mobility/quality-of-living-city-rankings.html

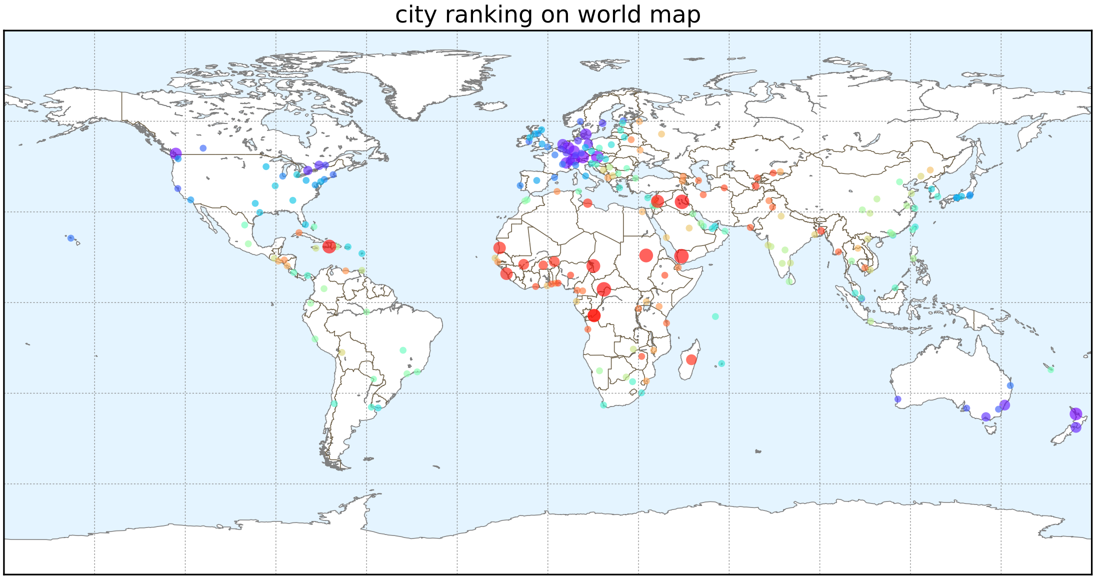


Figure 3.1: Distribution of Cities on Ranking List. The blue points show the cities on top of the ranking list, which means the place is good for living, the bluer the better, whereas the red points present the cities on the bottom of the ranking list, the redder the worse of the place for living. The cities with positive sentiment for living are most in Europe and north American, whereas the cities with positive sentiment for living are most in west Africa.

In list 3.1 the collection criteria is set, taking city Vienna as an example, the settings are as following:

– `n` presents one city name in `Citylist`, which is a list of 230 city names. Here the city name "*Vienna*" is given, where `n` is `Vienna`.

– `setQuerySearch()` is for setting the query term. "`#Vienna lang:en`" is the query term where using hashtag "#" plus city name "Vienna" like "`#Vienna`", and "`lang:en`" means that only tweets in English are searched.
Then `setQuerySearch(#Vienna lang:en)` is set.

  Defining the query term in the form with hashtag enlarges the collection range of all tweets referred to the city, but not only the city name showed up in tweets. Since English is the worldwide languages comparing to other languages, using this language as searching language has two advantages, one is the guarantee to get tweets of the query term as most as possible, another is text form in English is easy to dealing in natural language processing.

– `setSince()` and `setUntil()` are for setting collecting period. From `times` list the timestamp is taken used for presenting period of collection, which is set month by month iteratively.

– `setMaxTweets(100000)` means that each time maximum 100,000 tweets are allowed to be by collected.

During the process of collecting tweets some limitations are encountered:

- Before using this application to collect tweets a connection must be set up with Twitter server first, sometimes the connection is limited or interrupted during the process, so it makes the collecting progress slowly.
- Some cities' mother tongue is English, such as "Chicago", "London", "New York City", so the tweets of these cities are much more than other cities, in this case the maximum tweets setting has priority than time setting, so we increased the limitation of maximum settings.
- Some city names has more than two words, like "New York City", there is space between two words, but inhabit people sometimes forget to type the space while tweeting, due to this reason we have collected for these cities twice by using "NewYorkCity".
- Some cities with English name, but people used to tweet with city name in mother language. For example "Ashkhabad"is English name of the city, but people living there often use "Ashgabat". City like "Bandar Seri Begawan", formerly called Brunei Town and people in the city used to tweet with name "BSB" or others. These problems cause the collecting work difficult and the less than 200 of tweets have been collected over one year for these cities.

3.2 Data Selection

The last section describes the collection of tweets using GetOldTweets in Python. Through this method tweets for each of 230 cities between the period from 01.09.2015 to 31.08.2016 are all collected. As a result, the city with most tweets is Toronto, with 1,259,569 tweets, and the city with least tweets is Jilin with 154 tweets.

Figure 3.2 shows the distribution of the number of all cities related with the log_e number of collected tweets. From this histogram, it could be find that the most density part is distributed in the interval from 8 to 12, which means round 154 to 15,000 tweets, and there are round 30 cities stand between 12 and 13 corresponding from 15,000 tweets to 40,000 tweets, and round 12 cities are with more than 40,000 tweets.

As introduced in section 2.4 GetOldTweets enables us to collect tweets with these 10 fields: username, date, retweets, favorites, text, geo, mentions, hashtags, id and permalink. Thus the structure of collected tweets which stored in a CSV file are shown in table 3.1.

Based on the collected tweets the work turns to focusing on data selection. In order to select the most proper data for training our classifier, the most same amount of tweets is needed to be chosen as the training data,

We have compared the cities with most tweets and least tweets. Table3.3 and table3.2 show the ranking of the top 10 cities with most tweets and the last 10 cities with least collected tweets.

Table 3.1: Examples of the Structure of Collected Tweets

username	date	retweets	favorite	text	geo	mention	hashtags	id	permalink
mylocalescorts	2015-09-05 01:59	0	0	"Duo and Solo meets with Courtesan Annab http://www.my-local-escorts.co.uk/escort/Catherine-Can-Aberdeen–Edinb-4553 #Escorts #Aberdeen ,Edinb pic.twitter.com/mzNO1MWrc1"			#Escorts #Aberdeen	"639950982009462785"	`https://twitter.com/mylocalescorts/status/639950982009462785`
CumbernauldDeal	2015-09-05 01:48	0	0	"#WetWetWet 's 2016 Big Picture Tour: Ticket from 48 (Excl. Fees) at Choice of Location #Aberdeen #Glasgow - http://tidd.ly/f067d286?kgiR"			#WetWetWet #Aberdeen #Glasgow	"639948273428598784"	`https://twitter.com/CumbernauldDeal/status/639948273428598784`
iamnursecamila	2015-09-05 01:42	0	1	"RT http://twitter.com/JoinBAYADA/status/63994612 5684813824 #Nursing #Job in #Aberdeen , NC: Licensed Practical Nurse (LPN) Home Care at BAYADA Home Health Care "			#Nursing #Job #Aberdeen	"639946652585992192"	`https://twitter.com/iamnursecamila/status/639946652585992192`
Aberdeen_News_	2015-09-05 01:39	0	0	"#Aberdeen #News Perryman murder suspect ordered to remain in jail without bond despite health concerns: An Abe... http:// tinyurl.com/p95kkx2"			#Aberdeen #News	"639946022341316608"	`https://twitter.com/Aberdeen_News_/status/639946022341316608`

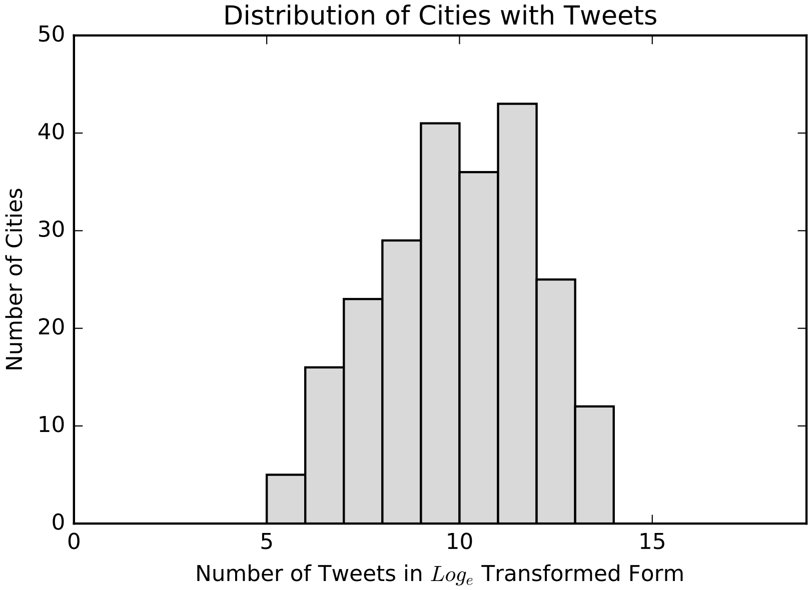


Figure 3.2: Distribution of Cities with Tweets

Table 3.2: City List of Least Tweets

	City Name	**Tweets No.**
1	Jilin	154
2	Bandar Seri Begawan	169
3	Pointe-a-Pitre	204
4	Nouakchott	377
5	Ashkhabad	388
6	Port Louis	429
7	Niamey	432
8	Banjul	448
9	Phnom Penh	449
10	Antananarivo	583

Table 3.3: City List of Most Tweets

	City Name	**Tweets No.**
1	Toronto	1,259,569
2	Chicago	903,891
3	Paris	857,485
4	Boston	838,170
5	Houston	719,694
6	Detroit	669,845
7	Miami	652,859
8	Dallas	597,571
9	Los Angeles	581,043
10	Atlanta	574,897

The difference between the number of tweets in city Toronto and in Jilin is more than 1,259,000 tweets. Facing this unbalance of data we have compared the distribution of all cities and made the decision that choosing 5,000 tweets from each city files randomly as our basic data set.

After random selection of data the distribution of selected tweets is shown in figure 3.3,

Table 3.4: City List of Least Selected Tweets

City Name	Tweets No.
Jilin	152
Bandar Seri Begawan	167
Pointe-a-Pitre	202
Nouakchott	375
Ashkhabad	386
Port Louis	427
Niamey	430
Banjul	446
Phnom Penh	447
Antananarivo	581
Lome	646
Shenyang	652
Noumea	707
Cotonou	745
Conakry	844
Dushanbe	853
San Salvador	1019
Tegucigalpa	1043
Port of Spain	1068
Libreville	1087
Blantyre	1090
Asuncion	1153
Nurnberg	1179
Gaborone	1300
Managua	1519
Yaounde	1624
Guatemala City	1681
Tashkent	1681
N'Djamena	1720
Bishkek	1729
Douala	1729
Brazzaville	1732
Manaus	1779
Qingdao	1897
Vientiane	1978
Luanda	2036
Taichung	2126
Maputo	2216
Bangui	2241
Ouagadougou	2320
Khartoum	2390
Jeddah	2402
Johor Bahru	2484
Abidjan	2908
Dares Salaam	3100
Kuwait City	3285
Lusaka	3294
Windhoek	3328
Algiers	3729
Kinshasa	3881
Tirana	4141
Almaty	4276
Montevideo	4665
Skopje	4668
La Paz	4701
Chengdu	4779
Ho Chi Minh City	4889

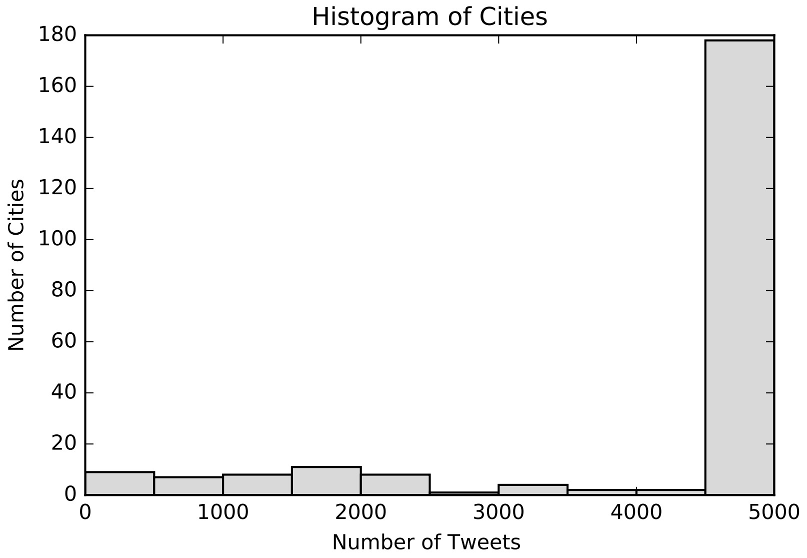


Figure 3.3: Distribution of Cities of Selected Tweets

in which more than 160 cities are with 5,000 tweets, whereas round 60 cities with less than 5,000 tweets as shown in table 3.4.

3.3 Classification Labels

In this study, the aim is to predict sentiment of places for living through building a classifier based on a ranking list of cities. Before machine learning, the dataset needs to be labeled first. The step of defining labels is described as following:

- First, binary classification is chosen in the thesis. The sentiment analysis is regarding positive sentiment, negative sentiment and in some studies also have neutral sentiment. In this thesis, the ranking list of Mercer is taken use for labeling. It contains 230 cities, in which each city has a ranking number, the first city is at the first place good for living and so on until the 230th city is at the first place not good for living. In this case, it is really hard to set a line to define which cities are of neutral sentiment. Moreover, for two-class classification problem, it is relatively easy to learning by using binary class classification.
- Second, the binary “1” is defined for positive sentiment and “0” is for the negative sentiment of living in a place. From the ranking list the first 115 cities are marked for positive sentiment, and the left 115 cities are marked for negative sentiment. Table 3.5 shows a part of the ranking list with labels.

Table 3.5: Ranking List with label

Ranking	**City Name**	**Country Name**	**Label**
1	Vienna	Austria	1
2	Zurich	Switzerland	1
3	Auckland	New Zealand	1
4	Munich	Germany	1
5	Vancouver	Canada	1
6	Dusseldorf	Germany	1
7	Frankfurt	Germany	1
8	Geneva	Switzerland	1
9	Copenhagen	Denmark	1
10	Sydney	Australia	1
11	Amsterdam	Netherlands	1
12	Wellington	New Zealand	1
13	Berlin	Germany	1
14	Bern	Switzerland	1
15	Toronto	Canada	1
15	Melbourne	Australia	1
⋮	⋮	⋮	1
115	Sofia	Bulgaria	1
116	Rabat	Morocco	0
⋮	⋮	⋮	0
215	Ouagadougou	Burkina Faso	0
216	Tripoli	Libya	0
217	Niamey	Niger	0
218	Antananarivo	Madagascar	0
219	Bamako	Mali	0
220	Nouakchott	Mauritania	0
221	Conakry	Guinea	0
222	Kinshasa	Congo, Democratic Republic of	0
223	Brazzaville	Congo, Republic of	0
224	Damascus	Syria	0
225	N'Djamena	Chad	0
226	Khartoum	Sudan	0
227	Port-au-Prince	Haiti	0
228	Sana'a	Yemen	0

Table 3.5: Ranking List with label

Ranking	**City Name**	**Country Name**	**Label**
229	Bangui	Central African Republic	0
230	Baghdad	Iraq	0

4 Methodology

This chapter introduces the methodology of the thesis. An overview of a general machine learning process is described first and the concrete approaches for this thesis are presented specifically.

4.1 Workflow

The main concepts of machine learning and classification are introduced in chapter 2, now the aim is to present the learning process.

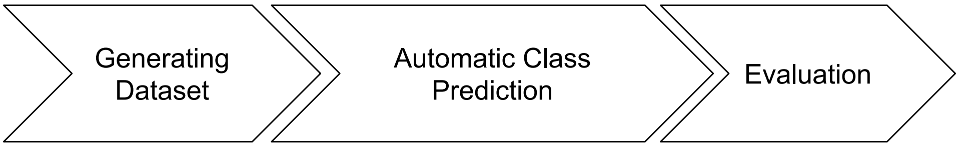


Figure 4.1: General Flowchart of Machine Learning

Figure 4.1 shows a general flowchart of a machine learning process. This flowchart contains three main steps:

- **Generating Dataset:** In this thesis the basic idea is making use one of the ranking list. We take use of a certain percentage (usually 70%) of the ranking information of the cities contributing to training data set. These cities are selected randomly. Regarding the distribution of the cities in the ranking list, one possible way to choose the training dataset and test dataset is that partitioning the cities of the list one by one to different data sets, which means that the first three good city to the training set, and the fourth good city to test set and so on.

  Based on the list of cities which are already selected as a training set, the training data of each city is tend to be collect through the Twitter API. After finishing collection of the data, the data for classification needs to be processed for preparing to train, where the most important are referred as cleaning data:

  - dealing with the stopwords, punctuations, emoticons.
  - dealing with the multimedia data, like URL, Graphics and so on.
  - dealing with redundant data, like Retweets.

- **Automatic Class Prediction:** With the goal of categorizing whether the quality of life in a city is good for living or not, one possible method is to create two classes namely positive sentiments and negative sentiments. Furthermore, in order to create a ranker a class of neutral sentiments could be added.

  In the feature extraction phase which machine learning model is of the best result and the features for improving the model need to be analyzed. In the context of text classification problems, through analyzing the relation between words, the n-gram method, bag of words method, term frequencyinverse document frequency method could be chosen to implement.

  In the classification phase based on the existed studies, linear classifier such as logistic regression classifier, naive bayens, SVM and the maximum entropy all have their own advantages for text classification, in addition nonlinear classifier such as boosting also have good performance. So to categorize the data a range of classifiers could be chosen to training model so that a proper classifier is obtained.

- **Evaluation:** At first the data of the cities which are selected as test set is introduced. In the thesis, this data is used for evaluating the classifier. Actually this test data set is already labeled, but we consider them as “unlabeled” first. After applying the classifier on this data, the result will be taken to compare with these already known labels, so that we could easily evaluate the accuracy of the classifier.

  There are different ways of measuring classifier’s performance, which based on the four basic numbers obtained from applying the classifier to the test set. (section 2.3) In this thesis the accuracy and the variance of each classifier have been compared.

4.2 Machine Learning Approaches

In section 4.1, the workflow of a general machine learning process has been discussed. Through combining the specifications of the dataset, the working process for generating a classifier to predict sentiment of a living place is improved.

As shown above figure 4.2 presents an overview of the predicting system. Matching to procedures shown in figure4.1, the three phases can be divided as flowing:

- Tweets about city information are collected and stored in CSV files. The details of collection process is already described in chapter3. Thus the processes of generating a dataset from Tweets are as following:
  - reading the CSV files from dataset to Dataset 1, where the information of tweets are selected and structured and be loaded in a dictionary form.
  - linguistic processing the data from Dataset 1 and then loading the tweets into Dataset 2, in which the tweets need to be tokenized.
- For automatic predicting phase, it contains the stages of preparing data, extracting features, training models and predicting data.

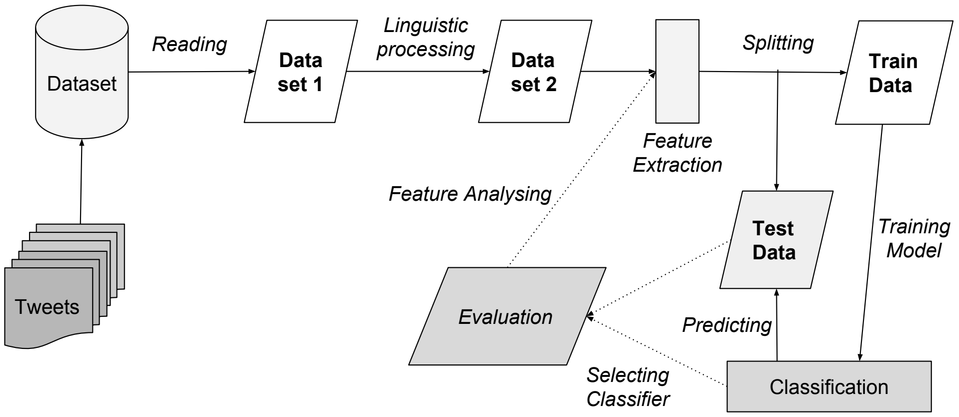


Figure 4.2: System Overview

- preparing data stage takes charge of cleaning data, which means the redundant data, useful less information will be deleted, and on the linguistic level, the corpus will be generated. This process is combined with feature extraction.

- extracting features process is important for training data, there are many methods of natural language processing method for extracting features, the most common way is to calculate the occurrence of the words which means using bag-of-words method or n-gram method.

- training models is the key step for creating a classifier. Based on different data specifications various of algorithms could be applied to the training data. In this thesis SVM classifier and logistic regression classifier are chosen for training models.

- predicting data is through fitting the already trained model to the test dataset to see the predicting result. Based on the results the baseline could be defined by comparing the performance of each classifier.

- Evaluation phase is not only the phase which aims to evaluate the classifier, but also to improve the performance of the classifier. To improve the performance of classifier this phase are not executed singly but with three other steps together, from the executing level these steps sometimes are not executed repeatedly:

  - performing classifier to predict the class of test data, evaluating the performance of the classifier through parameters like accuracy, deviation of the classifier etc.

  - electing useful feature information, adding or deleting some features are necessary, in another word it is a stage of feature engineering, the correlations between each features are observed and the feature weights need to be analyzed, some features are improved and collected based on common knowledge.

  - retraining the classification model with new features.

There are many program languages and corresponding libraries could be used to implement the system, but considering the convenient and flexibility for processing data of text form and languages, Python has been chosen as the main program language, in additions there are also much more specific packages available in Python for the realm of machine learning.

5 Implementation

Specific methodology of creating the classifier of the study have been presented in chapter 4, this chapter aims to introduce the implementation details and the problems during building this classifier for predicting the sentiment of a place of living.

5.1 Preprocessing

In data collecting process the database has been already constructed, in which the form is as introduced in CSV files. Now the need is to generate the dataset of the study. The methodology of generating dataset has been presented in section 4.2 and now the implementation of generating dataset in the study will be introduced.

As table 3.1 shown in section 3.1 all collected tweets are stored with these information in a CSV file. The city list contains 230 cities, which means there are 230 CSV files in the database. Now how to generate data set from these CSV files preparing for automatic predicting, the processes are shown in figure 5.1.

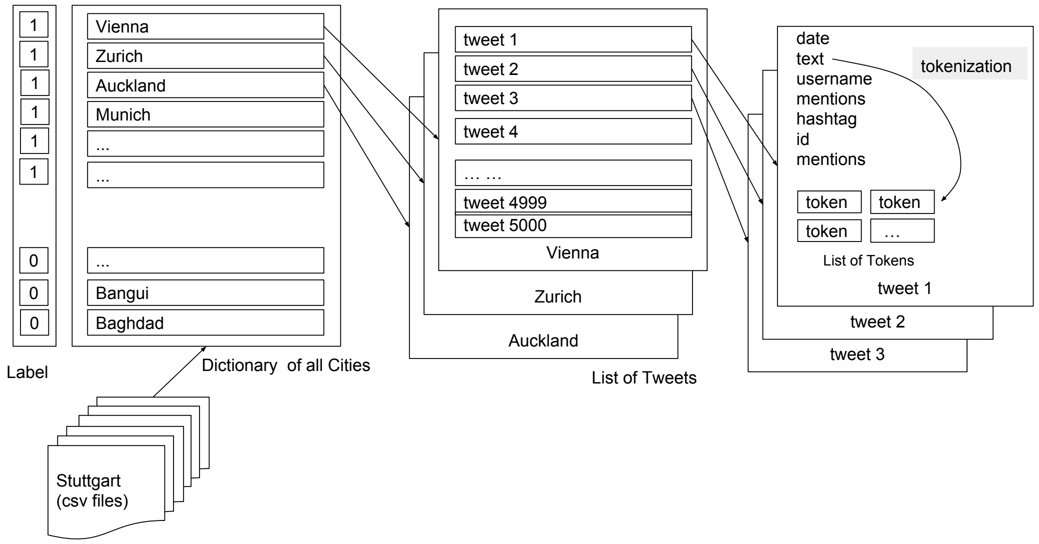


Figure 5.1: Overview of Data Processing

- **Loading all tweets from csv files into a organized form:** At first with aim to load all information of all tweets of all city files into one processable data form a tweet class is designed. The diagram of Class `Tweet` is shown in figure 5.2. It contains:

  - Class members: `date`, `text`, `username`, `mentions`, `id`, `hashtags`, `links` and list of `tokens` and `tags`.
  - Class methods: `load_tweets()` which used to load the tweets from CSV files, `load_tweet_tokens()` used to load the tokens of `text` of `tweet` into list `tokens` .

- **Tokenization the text of each tweet:** Since this work focuses on analyzing the tweets with sentiment and the tweet has its own characteristics, the tokenization process is implemented separately by using Tweet NLP[1].

  Tweet NLP is a specific tool for natural language processing of tweets, in which the emoticons could be recognized so that the sentiment information will not be lost, whereas they usually be recognized as normal punctuations by general tokenization. Furthermore through applying Tweet NLP for tokenization process, the POS tags are also generated for each token.

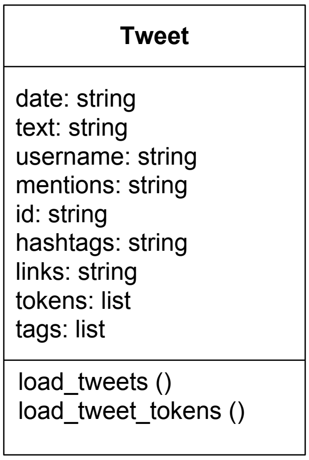


Figure 5.2: Class Diagram of Tweet

  Table 5.1 shows the structure of tokens of tweets. The abbreviations of Part of Speech tags of Tweet NLP which particularly mark for tweets are as shown in table 5.2

- **Loading all tokens into a list:** Object `Tweet` contains two members `tokens` and `tags`, using `load_tweet_tokens()` method the tokens of each tweet could be loaded into a list, thus for city with 5,000 tweets means 5,000 lists of tokens are loads for the city.

- **Generating dataset:** Aims to prepare training data for next step, as introduced in chapter4, the training dataset is a dataset with labels. How the labels for each city marked is already described in section3.3, the labels of all 230 cities are presented as a vector. Then We need to create a matrix to match the tweets of all cities with the vector of city labels, these are done with the extraction of features.

[1] `http://www.ark.cs.cmu.edu/TweetNLP/`

Table 5.1: Example of Tokenization Tweet

Text of Tweet	**Tokens**	**POS tags**
#receptionist #jobs Clerical Receptionist: Before applying for this job, it is important that you re... http:// bit.ly/1R8n7l0 #Aberdeen	#receptionist	#
	#jobs	#
	Clerical	A
	Receptionist	N
	:	,
	Before	P
	applying	V
	for	P
	this	D
	job	N
	,	,
	it	O
	is	V
	important	A
	that	P
	you	O
	re	V
	...	,
	http://	U
	bit.ly/1R8n7l0	U
	#Aberdeen	#

5.2 Automatic Class Prediction

Automatic class prediction phase always cooperates with Evaluation phase together. This part of processing includes many possibilities, the executed processes in this work will be introduced. The whole Evaluation phase will be introduced separately in chapter 6 with analysis.

Table 5.2: Twitter Part-of-Speech Tagging

Type	Tag	Annotation
Nominal	N	common noun
	O	pronoun (personal/WH; not possessive)
	ˆ	proper noun
	S	nominal + possessive
	Z	proper noun + possessive
Other open-class words	V	verb incl. copula, auxiliaries
	A	adjective
	R	adverb
	!	interjection
Other closed-class words	D	determiner
	P	pre- or postposition, or subordinating conjunction
	&	coordinating conjunction
	T	verb particle
	X	existential there, predeterminers
Twitter/online-specific	#	hashtag (indicates topic/category for tweet)
	@	at-mention (indicates another user as a recipient of a tweet)
	˜	discourse marker, indications of continuation of a message across multiple tweets
	U	URL or email address
	E	emoticon
Miscellaneous	$	numeral
	,	punctuation
	G	other abbreviations, foreign words, possessive endings, symbols, garbage
Other compounds	L	nominal + verbal (e.g. im), verbal + nominal (lets, lemme)
	M	proper noun + verbal
	Y	X + verbal

Table 5.3: Principle of Creating Matrix

	feature 1	feature 2	feature 3	feature 4	feature 5	...	feature $k-1$	feature k
tweet1	1	2	3	4	3	...	0	0
tweet2	0	15	0	1	0	...	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
tweet$n$	1	0	3	4	3	...	0	0

5.2.1 Feature Extraction

In section 4.2 the process of automatic predicting is described. Feature extraction is one of the most important parts of the whole classification processing. In this process extracting the features from the text and preparing the data for training is needed.

Since how the features will impact on classifying is unknown, at the beginning a feature list is generated with the aim to keep the original information as most as possible, thus the bag-of-words model is used, choosing all tokens of the dataset as features, which is of 1,943,828 dimensions. Then the count of occurrence of each token in a city file composes the original matrix for predicting like table 5.3.

For generating the data matrix `DictVectorizer` which imported from library `sklearn.feature_extraction` is exploited, in which a sparse matrix for each city is created. In order to match the matrix with the labels, the matrix of each city have been zipped into one vector using `sparse.vstack()` method.

Furthermore, the classifier has been improved by using improved features base on analysis of feature weights of logistic regression model. For the improvement, these features have been used for first step:

- Cityname: The city names of Mercer ranking list is taken use again. Since the tweets are collected city by city and using the query term of the city name, so for each city the city name is a domain feature impact the classification, so they need to be removed.
- Stopwords and Punctuations: There is an existed English stopwords list from library NLTK[2], which contains 127 stopwords. Before using this word list it need to be download first and then is useful by importing `wordpunct_tokenize` from library `nltk.tokenize`.

[2] `www.nltk.org`

Table 5.4: Sentiment Wordlist Examples

Sentiment	Word Examples
positive	wonder,warm,welcome,trendy,succes
negative	threaten, thumb-down,trash,unclean,unlucky

Meanwhiles all punctuations are removed by checking POS tags of each token. As described in section 5.1, the tokenization processing has already tagged all tokens with Part of Speech Tags, in which the tag with "." means that the token is a punctuation.

- Lowercase: Since the tokenization processing is not case sensitive, there are many features of same words but in different case. To solve this problem all tokens are converted into lowercase.

For feature engineering, in order to see how features impact the classification, the following attempts for improving features are presented as second step:

- URLs: All appeared URLs in tweets are considered as a whole entity. Since the links are tokenized by tagging with "U", all URLs are removed through taking use of the POS tags.
- Hashtags: Since the query terms are combined with hashtags, with the aim to see the influence of hashtags, POS tag "#" is taken used to remove all tokens with Hashtag.
- Country names: Since the country names often appear on both top and last 100 feature weight list, in order to avoid over weighted on these features the country names need to be removed. Removing the tokens with country name We exploit country list of ISO 3166[3], in which contains 249 countries.

5.2.2 Sentiment Analysis

In order to further improve the model, sentiment word list[4] is imported for sentiment analysis. This sentiment word list consists of two parts, one is word list of positive sentiment, which contains 2,006 words, the other is word list of negative sentiment, which contains 4,783 words.

The occurrence of positive words and the occurrence of negative words are counted for each city file, so that feature space can be extended with *count(positive words)* and *count(negative words)*, furthermore the combination of these two features are extended: *count(positive words − negative words)* and *count(positive words/negative words)*. The reason why a combination of the two counts is added as extended feature is to solve the problem that the same feature weight of p and n working on different tweets may

[3] http://www.iso.org/iso/home/standards/country_codes.htm

[4] https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html

Table 5.5: Sentiment Features Examples

Tweet	counts of p-words	counts of n-words	p-n	p/n
We **love** this **pretty** blue embroidered dress worn by WendyNguyen in Paris !	2	0	2	none
The Bia in light natural tan, looking beautiful on a photoshoot in Stockholm! stockholm	0	0	0	none
The **frozen** Northeast China.. travel jilin dongbei china	0	1	-1	0
No reason to be **bored** this week in Shenzhen	0	1	-1	0

influence to the classification differently. Since the logistic regression classifier is linear, each feature is added together for training models, in this case, the p/n can keep the relation between positive counts p and negative counts n the same.

Table 5.5 shows examples of counting the sentiment words of a single tweet, where four tweets of different cities are shown up. Since tweet has limited of 140 characters, each tweet is short and sometimes contains only one sentiment, which may cause the problems:

- the value of $p - n$ may be as same as p or $-n$, which in context of using linear logistic regression classifier contributes almost the same as feature p or n.
- no value of p/n is calculated or $p/n = 0$, thus the feature of sentiment words is counted in city.

5.2.3 Logistic Regression

Nowadays there are many libraries offering tools for classification. Scikit-learn is a particular library for machine learning in Python, it offers simple and efficient tools for data mining and data analysis, especially it is accessible to everybody, and it is reusable in various contexts, open source and very popular. Thus the library of scikit-learn.org[5] is taken use for the thesis.

Logistic regression classifier is taken use for this work since it has outperform result for text classification. To avoid overfitting there are also L_1 and L_2 regularization method for turning the parameters. The class `LogisticRegression` is first imported from library `sklearn.linear_model`.

For training the classification model, these steps are executed:

[5] `http://scikit-learn.org`

- Creating an object of the class: `logistic = linear_model.LogisticRegression(C=1e5)`, where parameter `C=100000.0` is set for L_2 regularization.
- Creating the model by training each fold of training dataset: `logistic.fit(city_X_train, city_y_train)`, where `city_X_train` is the training dataset of each fold, and `city_y_trai` is the corresponding label set. By learning these data the logistic regression model is trained.
- Using the classifier to predict data: `logistic.predict(city_X_test)`, where `city_X_test` is each fold of test dataset. By predicting the test data the performance of the classifier could be measured.

As a result the classifier is trained as the baseline of the study.

6 Experiments Setup and Evaluation

In this chapter, the experimental setup of the system is described. The evaluation of the classifier is tested, especially for logistic regression model and SVM model. On the basis of the baseline, attempts to enhance the performance of the classification are done, while features of the training dataset are improved through the analysis of feature weights of each model.

6.1 Cross-validation

In chapter 3 how data in the thesis are collected from Twitter is presented. Since there are only 230 labeled examples available for both training and testing classifier, the dilemma that all the examples are needed for training dataset, on the other hand, all examples are necessary for the testing dataset is faced. The solution is to use cross-validation, which is described in section 2.3.

In this thesis Mercer ranking list has 230 cities, so the Matrix M contains 230 rows, the study does 10 splits, where $k = 10$ leading to fold size 23 ($230/10 = 23$) rows, while the left 207 rows are used to train the classifier. Since the ranking list of Mercer is in an order and so is the whole dataset, to ensure the labeled examples are randomly distributed in each training set and testing set, the order of the matrix have been messed up before applying the algorithm.

Table A.1 shows the city list for testing the classifier of each fold. As Using cross-validation each fold devotes for testing data, and the left 90% of data devote to training classifier.

6.2 Evaluation

In the thesis, average accuracy score is used as an evaluation method of the classifier. For each fold dataset of applying a classifier is scored and the performance of this classifier by calculating average value of these 10 scores is measured.

6.2.1 Baseline

The logistic regression model and SVM are introduced in chapter 2, both of them show the advantages of handling text classification. Since many studies present the SVM and logistic regression have the outperform of other classifiers in text categorization, which is introduced in section 7.3. In the thesis, these two types of classifiers are chosen for training the models firstly.

Table 6.1: Logistic Regression vs SVM Classifiers

	Logistic	**SVM**	**SVM**	**SVM**
	Regression	**linear**	**polynomial**	**rbf**
Average Score	**0.795652**	0.7391303	0.7260868	0.6086954
Variance	**0.009849**	0.011342	0.003800	0.004915

Table 6.1 shows the comparison of the results of applying logistic regression classifier and the SVM classifier to training data. The average score reflects the accuracy of a classifier. We calculate the accuracy for each classifier 10 times, and then the mean value as the average score is gained.

The results of all SVM classifiers are worse than logistic regression classifier with 0.795652. Based on the analysis of SVM in existed studies, the reason may be caused by the high dimensions of extracted features (1,943,828) than the labeled examples (230).

At the end, the result of applying logistic regression classifier on original dataset have been chosen as Baseline.

6.2.2 Improving Baseline

Based on the baseline the features and the performance of the classifier have been improved. Relying on the analysis of features introduced in section 5.2.1 the following steps are then processed on the original dataset:

- Removing City names
- Removing Stopwords
- Converting all words to lowercase

These steps are first implemented individually, then the compositions of each condition are further implemented.

The table 6.2.2 shows the improved average score of the classifier for combining with each condition. For individual conditions, the result of logistic regression classifier working with out city name improved the baseline, which gets the best with 0.834782, whereas the classifier without stop words shows worst result at 0.769565 which decreases the baseline. It seems that the Stop words plays an insignificant role.

For the combination of the conditions, the best improved result of 0.873913 are gained when the classifier work with the combination of the dataset without city names, stop words and also converted into lowercase, and the deviation of this combination is also relatively stable. In contrast the classifier works without stop words and with all tokens in lowercase most unstable and its average score is the worst at 0.743478 of all combinations

Table 6.2: Comparision of the Performance of the Classifiers

	Logistic Regression Classifier in Conditions	Avarage Score	Variance
0	**Baseline**	**0.795652**	**0.009849**
1	No Cityname	0.834782	0.005974
2	In Lowercase	0.791304	0.007486
3	No Stopwords	0.769565	0.010605
4	In Lowercase and No Cityname	0.839130	0.004934
5	No Cityname and Stopwords	0.847826	0.008034
6	In Lowercase and No Stopwords	0.743478	0.010605
7	**No Cityname and No Stopwords in Lowercase**	**0.873913**	**0.006975**

which are also under baseline. Both features of removing stop words and using lowercase combining with removing city names have increased the performance of singly using the feature without city name.

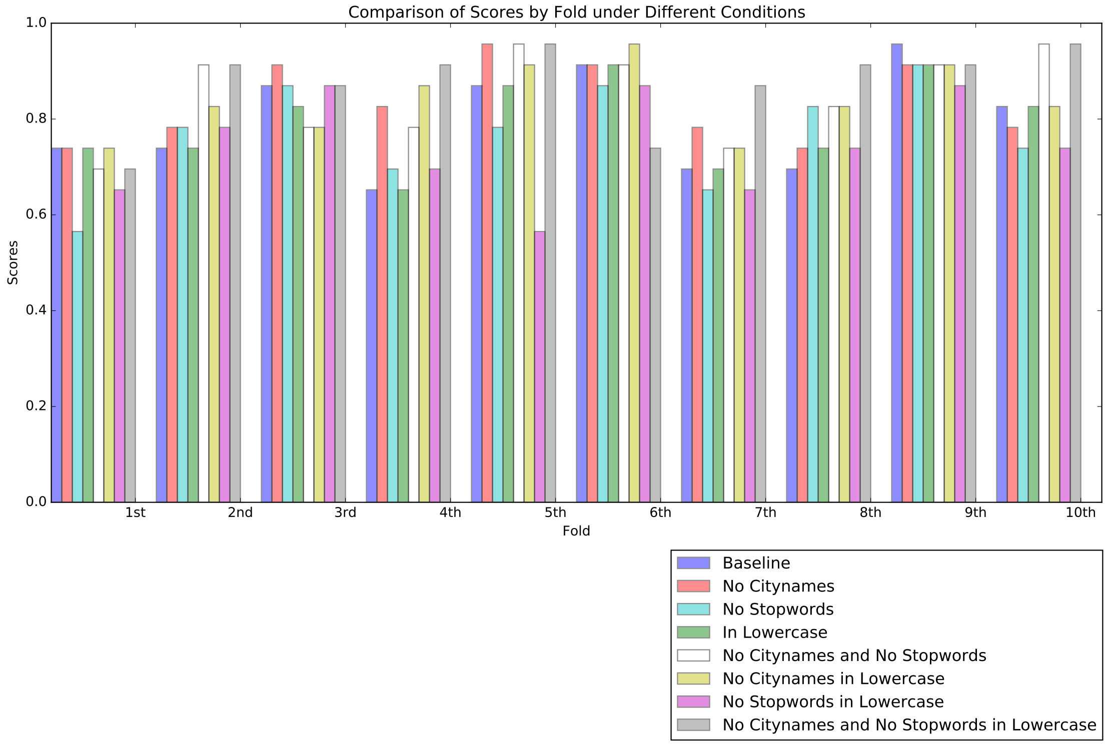


Figure 6.1: Scores of Logistic Regression Classifiers by Fold under Different Conditions

Figure 6.1 shows the accuracy scores of logistic regression classifier for each fold cooperated with all conditions showed in the legend box. From figure 6.1 it could be seen that the accuracy scores of different classifiers applying on the same fold of dataset have a similar trend. For example, all classifiers work well for the 9th fold dataset with the score round 0.9 and obviously work badly for the data of 1st fold.

6.2.3 Sentiment Analysis Result

In section 5.2.2 the sentiment analysis is discussed. By extending the sentiment analysis features with logistic regression, the following results are obtained.

- Table 6.3 shows the result of using single features of sentiment word list applying to the original dataset. This attempt uses only the $p - n$ as combination features, it could be seen that average scores of using p, n and $p - n$ as a single feature with logistic regression classifier has decreased the baseline score, where only using the counts of negative words as one feature with the worst result at 0.539130. Using a combination of $p - n$ has the highest scores but is only slightly increased the score of other combinations to 0.713043.

  Table 6.3: Result Table of LR Model with Sentiment Analysis Feature

Baseline	p-words	n-words	p-n words	average scores
√				0.795652
	√			0.630435
		√		0.539130
			√	**0.713043**
	√	√		0.708696
	√	√	√	0.708696

  Specially the result of using p and n is as same as the composition of p, n, $p - n$, which is consistent with the analysis that $p - n$ may not a proper combination way of the two basic features in logistic regression model. In this case, the feature combination of p/n is extended instead of $p - n$ for further improvement.

- Table 6.4 shows the result of improving the extra features of sentiment analysis on the improved baseline. The best result is using p/n at 0.865217 which also declined the improved baseline slightly. Combining all possible features with max abs scale the result is stable at 0.856522.

6.2.4 Feature Engineering

Based on the best result by improving the baseline, many further attempts are made with the goal of further improving the classifier, as described in section 5.2.1.

Table 6.4: Result Table of Using Sentiment Analysis Based On Improved Baseline

nocn_sw_lc	p-words	n-words	p-n words	p/n words	maxabs scale	average scores
√						0.873913
√	√					0.847826
√		√				0.847826
√			√			0.856522
√				√		**0.865217**
√	√	√				0.847826
√	√	√	√			0.843478
√					√	0.856522
√	√				√	0.856522
√		√			√	0.856522
√			√		√	0.856522
√				√	√	0.856522

Table 6.5 shows part of the results of the accuracy score of the logistic regression classifier under variance conditions and the combinations of them.

"**B**" is the abbreviation of baseline and this row shows the score of baseline gained through the original data set. The first combination with "**no cn sw lw**" is the best result of improved baseline, in which the tokens are without city names, without stop words and in lower case. The abbreviations of each condition are shown in table6.6.

Table 6.5: Best Result under Conditions Based on Best Baseline

	no cn sw lc	**no url**	**no hash-tag**	**no country name**	**p-words**	**n-words**	**p/n words**	**maxabs scale**	**average scores**
B									0.795652
0	√								**0.873913**
1	√	√							0.821739
2	√	√						√	0.821739
3	√	√			√	√	√	√	**0.873913**
4	√	√			√	√	√		0.830435

Table 6.5: Best Result under Conditions Based on Best Baseline

	no cn sw lc	**no url**	**no hashtag**	**no country name**	**p-words**	**n-words**	**p/n words**	**maxabs scale**	**average scores**
5	√	√			√			√	**0.873913**
6	√	√			√				0.830435
7	√	√				√		√	**0.873913**
8	√	√				√			0.830435
9	√	√					√	√	**0.873913**
10	√	√					√		0.834783
11	√	√			√	√		√	**0.873913**
12	√	√			√	√			0.830435
13	√	√				√	√	√	**0.873913**
14	√	√				√	√		0.830435
15	√			√					0.856522
16	√			√				√	0.856522
17	√			√	√	√	√	√	0.856522
18	√			√	√	√	√		0.847826
19	√			√	√			√	0.856522
20	√			√	√				0.865217
21	√			√		√		√	0.856522
22	√			√		√			0.856522
23	√			√			√	√	0.856522
24	√			√			√		0.856522

From table A.6 it is trivial to find that all combinations have better results than baseline. But no better score are obtained than the already improved best result. In spite of the influence of sentiment analysis features with scaling, the highest score is as same as the best results at 0.873913, which with combinations of the improved baseline without URLs. Since the analysis in section 5.2.2 the improved baseline with sentiment features with scaling shows the stable result of 0.856522, from this point of view the feature without URLs has improved the performance slightly. Comparing the results of combining the features singly with improved baseline and keep the matrix scaling, removing the

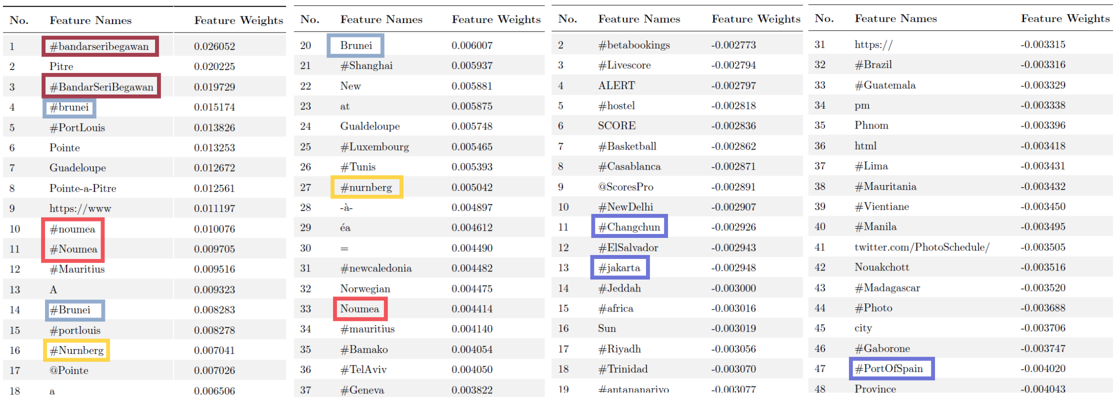
Table 6.6: Conditions for Further Improvement LR Model

Abbreviation	Condition Name
no url	removing the tokens with url
no hashtag	removing all the tokens with hashtags.
no country name	removing all possible country name from a country name list.
p-words	adding counts of occurence of positive words.
n-words	adding counts of occurence of negative words.
p/n-words	adding percentage of counts of occurence of positive words to negative words.

country names and removing hashtags both have the same score at 0.856522, which is better than just removing URLs at 0.821739.

Feature Weights Analysis

After training logistic regression model on the original dataset, the feature weights are analyzed based on tables A.2 and A.3 in Appendix. These two tables present the top 100 features of positive weights and top 100 features of negative weights after applying the classifier on *10th*-fold of the training set.



No.	Feature Names	Feature Weights
1	#bandarseribegawan	0.026052
2	Pitre	0.020225
3	#BandarSeriBegawan	0.019729
4	#brunei	0.015174
5	#PortLouis	0.013826
6	Pointe	0.013253
7	Guadeloupe	0.012672
8	Pointe-a-Pitre	0.012561
9	https://www	0.011197
10	#noumea	0.010076
11	#Noumea	0.009705
12	#Mauritius	0.009516
13	A	0.009323
14	#Brunei	0.008283
15	#portlouis	0.008278
16	#Nurnberg	0.007041
17	@Pointe	0.007026
18	a	0.006506

No.	Feature Names	Feature Weights
20	Brunei	0.006007
21	#Shanghai	0.005937
22	New	0.005881
23	at	0.005875
24	Gualdeloupe	0.005748
25	#Luxembourg	0.005465
26	#Tunis	0.005393
27	#nurnberg	0.005042
28	-à-	0.004897
29	éa	0.004612
30	=	0.004490
31	#newcaledonia	0.004482
32	Norwegian	0.004475
33	Noumea	0.004414
34	#mauritius	0.004140
35	#Bamako	0.004054
36	#TelAviv	0.004050
37	#Geneva	0.003822

No.	Feature Names	Feature Weights
2	#betabookings	-0.002773
3	#Livescore	-0.002794
4	ALERT	-0.002797
5	#hostel	-0.002818
6	SCORE	-0.002836
7	#Basketball	-0.002862
8	#Casablanca	-0.002871
9	@ScoresPro	-0.002891
10	#NewDelhi	-0.002907
11	#Changchun	-0.002926
12	#ElSalvador	-0.002943
13	#jakarta	-0.002948
14	#Jeddah	-0.003000
15	#africa	-0.003016
16	Sun	-0.003019
17	#Riyadh	-0.003056
18	#Trinidad	-0.003070
19	#antananarivo	-0.003077

No.	Feature Names	Feature Weights
31	https://	-0.003315
32	#Brazil	-0.003316
33	#Guatemala	-0.003329
34	pm	-0.003338
35	Phnom	-0.003396
36	html	-0.003418
37	#Lima	-0.003431
38	#Mauritania	-0.003432
39	#Vientiane	-0.003450
40	#Manila	-0.003495
41	twitter.com/PhotoSchedule/	-0.003505
42	Nouakchott	-0.003516
43	#Madagascar	-0.003520
44	#Photo	-0.003688
45	city	-0.003706
46	#Gaborone	-0.003747
47	#PortOfSpain	-0.004020
48	Province	-0.004043

Figure 6.2: Top Features Weights of LR Models

Figure 6.2 shows a snapshot of tables A.2 and A.3, on which some significants are marked:

- The city name “#bandarseribegawan” and “#BandarSeriBegawan” both have high positive feature weights, but they are only different with capitals. Similar to this characteristics there are “#brunei”, “#Brunei” and “Brunei”, “#Nurnberg” and “#nurnberg”,“#noumea”,“#Noumea”and “Noumea”.

- The city names is a significant characteristic dominating in the top features list. “#Changchun”,“#jakarta” and “#PortOfSpain” are all of the ranking list.

Table 6.7 shows the analysis of the feature weights. The feature characteristics of these 200 features are summarized as following: First more than half of the samples are city names some are with hashtags some not, and without considering the Capital form; Second 10% of the samples has shown up the same features but in different forms, some with capital and some in lower case; Third the frequency that the country name is shown up for 5% of all samples.

Table 6.7: Comparison the Feature Weights of Logistic Regression Model

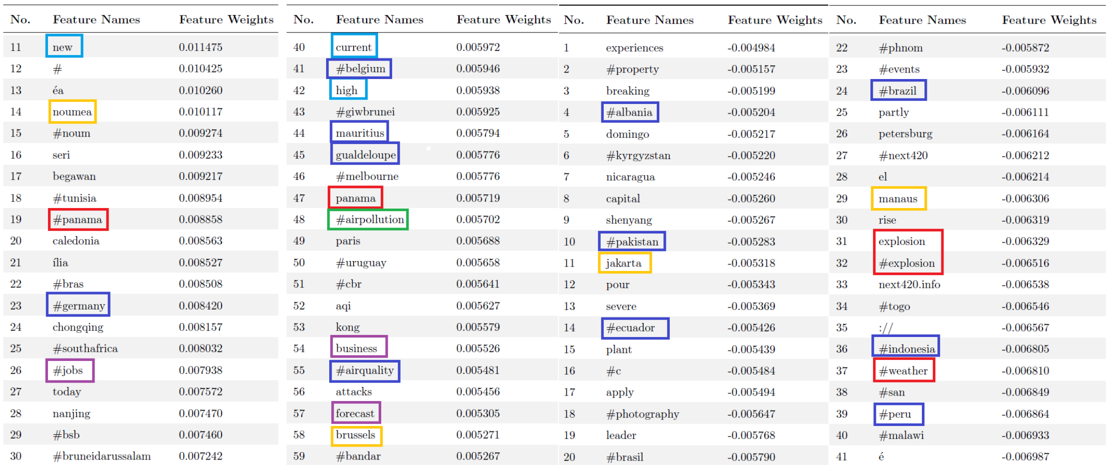
Feature characteristics	Top 100 features with positive weights	Top 100 features with negative weights
City names	54	52
Stop words	5	2
punctuations	2	3
lower cases	10	12
links	2	2
special charactor	4	2
Country names	5	4

Relying on the comparison of the features, the feature extraction process has been improved by removing the features of city names, removing the stop words using English stopwords list extension with punctuations together, while all the tokens are used in lower cases. After these works, the feature space is reduced from 1,943,828 to 1,802,722. The performance of the classifier on training this dataset has been obviously improved.

Further analysis is made based on tables A.4 and A.5, where the snapshot are shown in figure 6.3 These two tables show the top 100 and last 100 feature weights of the improved logistic regression model. The comparison of these features is shown in table 6.8.

Figure 6.3 shows a snapshot of tables A.4 and A.5, on which some significants are marked:

- The number of city names are reduced obviously.
- More features are country names like “#brazil”, “#indonesia”, “#peru”, “#albania”, “#ecuador”, “#belgium”, “#pakistan” and so on.
- “#explosion”and “explosion” both have high negative feature weights, but the difference is only a hashtag “#”. Similarly is “#panama” and “panama”.



No.	Feature Names	Feature Weights	No.	Feature Names	Feature Weights	No.	Feature Names	Feature Weights	No.	Feature Names	Feature Weights
11	new	0.011475	40	current	0.005972	1	experiences	-0.004984	22	#phnom	-0.005872
12	#	0.010425	41	#belgium	0.005946	2	#property	-0.005157	23	#events	-0.005932
13	éa	0.010260	42	high	0.005938	3	breaking	-0.005199	24	#brazil	-0.006096
14	noumea	0.010117	43	#giwbrunei	0.005925	4	#albania	-0.005204	25	partly	-0.006111
15	#noum	0.009274	44	mauritius	0.005794	5	domingo	-0.005217	26	petersburg	-0.006164
16	seri	0.009233	45	gualdeloupe	0.005776	6	#kyrgyzstan	-0.005220	27	#next420	-0.006212
17	begawan	0.009217	46	#melbourne	0.005776	7	nicaragua	-0.005246	28	el	-0.006214
18	#tunisia	0.008954	47	panama	0.005719	8	capital	-0.005260	29	manaus	-0.006306
19	#panama	0.008858	48	#airpollution	0.005702	9	shenyang	-0.005267	30	rise	-0.006319
20	caledonia	0.008563	49	paris	0.005688	10	#pakistan	-0.005283	31	explosion	-0.006329
21	ília	0.008527	50	#uruguay	0.005658	11	jakarta	-0.005318	32	#explosion	-0.006516
22	#bras	0.008508	51	#cbr	0.005641	12	pour	-0.005343	33	next420.info	-0.006538
23	#germany	0.008420	52	aqi	0.005627	13	severe	-0.005369	34	#togo	-0.006546
24	chongqing	0.008157	53	kong	0.005579	14	#ecuador	-0.005426	35	://	-0.006567
25	#southafrica	0.008032	54	business	0.005526	15	plant	-0.005439	36	#indonesia	-0.006805
26	#jobs	0.007938	55	#airquality	0.005481	16	#c	-0.005484	37	#weather	-0.006810
27	today	0.007572	56	attacks	0.005456	17	apply	-0.005494	38	#san	-0.006849
28	nanjing	0.007470	57	forecast	0.005305	18	#photography	-0.005647	39	#peru	-0.006864
29	#bsb	0.007460	58	brussels	0.005271	19	leader	-0.005768	40	#malawi	-0.006933
30	#bruneidarussalam	0.007242	59	#bandar	0.005267	20	#brasil	-0.005790	41	é	-0.006987

Figure 6.3: Top Features Weights of Improved LR Models

- Some interesting features are shown up such as "#jobs" "#airquality", "#airpollution" with positive feature weights, and "explosion" with negative feature weights. Further more some adjectives are shown up in the top feature list.
- Some city names still are shown up but without hashhtag, like "noumea","brussels" and "jakarta".

Table 6.8: Comparison the Feature Weights of Best Improved LR Model

Feature characteristics	Top 100 features with positive weights	Top 100 features with negative weights
City names	16	12
City names with #	4	1
Stop words	0	0
punctuations	0	1
lower cases	0	0
links	2	2
special charactor	8	3
Country names	12	24

In spite of the reduced features space, the conditions for analyzing feature weights are all the same, these features are analyzed based on the *10th*-fold training dataset and the top 100 features with positive weights and 100 with negative weights.

From table 6.8 it is shown that the city names are still in the top feature list but not the unique dominate characteristic anymore, whereas more country names appeared in the list. The stop words, punctuations are almost removed.

Correlation between features and living standard sentiment

By analyzing the features of the top 100 positive feature weights and top 100 negative feature weights of applying improved classifier on 1*st* fold dataset, table 6.11 is constructed.

Despite the summarized characteristics of these features (country names, city names, links and special characters) 30 features are selected and shown in table 6.11. These features include nouns, adjectives, adverbs and some with hashtags.

Based on table 6.11 an analysis of the aspects of the living standard which impact model to associate positive sentiment or negative sentiment of living is provided. The following issues are summarized:

Table 6.9: Interference Features

Sentiment	Feature Name
positive	attacks pollution bad #airpollution
negative	good

- As shown in table 6.9, some of the features that indicated positive sentiment in natural language is associated with negative weight, this is probably due to noise in the collected data, but further study or larger dataset would be required for more in-depth investigation.
- A categorization exercises are performed to group most positive features and most negative features from table 6.11 into various aspects of the living standard. This is shown in table 6.10. This illustrates how the model learns the impact of features on the classification.

Table 6.10: Categorization of Features Based on Living Standard Aspects

Aspects Category	Positive Feature	Negative Feature
time and date	today weekend christmas day tomorrow current night pm	time minutes utc
weather	pollution #airpollution #airquality forecast weather	wind #sunrise #weather sun
job	#jobs president business	leader president
war related	attacks	alert explosion #explosion

- Based on table 6.10 the model tends to associate positive sentiment with features about time and date, job and business. In contrast, war-related features such as “explosion” are associated with negative sentiment.

Table 6.11: Comparison of Feature Sentiments

Feature name	Feature weights	Feature name	Feature weights
new	0.008840	plant	-0.004589
#jobs	0.006836	partly	-0.004659
today	0.006827	rejoint	-0.004737
#airpollution	0.006681	#dating	-0.004764
see	0.006052	youth	-0.004851
#airquality	0.006017	capital	-0.004879
forecast	0.005615	html	-0.005018
high	0.005585	explosion	-0.005139
attacks	0.005504	president	-0.005181
current	0.005432	#events	-0.005270
weather	0.005081	#explosion	-0.005300
great	0.004966	breaking	-0.005438
feeling	0.004845	leader	-0.005644
moderate	0.004790	rise	-0.006154
live	0.004612	wind	-0.006481
pollution	0.004354	#weather	-0.006658
tomorrow	0.004327	#sunrise	-0.006914
business	0.004236	sun	-0.007436
day	0.004210	alert	-0.008705
little	0.004053	score	-0.008823
cn	0.004040	minutes	-0.008987
bad	0.004039	good	-0.009042
view	0.004004	utc	-0.009114
like	0.003945	#livescore	-0.009133
weekend	0.003811	city	-0.010295
starting	0.003765	#photo	-0.010361
feel	0.003750	time	-0.010927
christmas	0.003709	pm	-0.011716
night	0.003701	local	-0.014346
best	0.003588	province	-0.021025

7 Related Works

This chapter is structured in three parts: first, researches relating to the study of quality of life are presented, in which the factors which impact the quality of life are studied in many studies, and then studies about the quality of life are introduced. Second, the studies related to text categorization and sentiment analysis applying to the textual data of various media resources which motivate this thesis are presented. At last a short overview of the related studies is summarized.

7.1 Researches about Quality of Life

7.1.1 Studies Dedicating to Quality of Life

Since the definition of quality of life are various and is always determined by many factors, different country succeeds at individual characteristics, and with various of objects, the researches of the quality of life are always aspect-related.

- The Organization for Economic Cooperation and Development (OECD)[1] releases its Better Life Index of countries with the best quality of life annually. To do this, the OECD studied 34 countries across different parameters of well-being, including work-life balance, financial wealth, and quality of the environment.
- In Eurostat[2] the Quality of Life indicators which used to measuring quality of life are presented in 8+1 dimensions, namely material living conditions, productive or main activity, Health, Education, Leisure and social interactions, Economic and physical safety, Governance and basic rights, Natural and living environment and Overall experience of life.
- Mercer[3] releases a report about Mercers Quality of Living Rankings cover 230 prevalent cities. Living conditions are analyzed according to 39 factors, grouped in 10 categories, which are similar as the Eurostat, which contains aspects of Political and social environment, Economic environment, Socio-cultural environment, Medical and health considerations, Schools and education, Public services and transportation, Recreation, Consumer goods, Housing, Natural environment. The scores attributed to each factor, which is weighted to reflect their importances to expatriates, permit objective city-to-city comparisons. The result is a quality of living index that compares relative differences between any two locations evaluated.

[1] http://oecdbetterlifeindex.org/#/55555555555

[2] http://ec.europa.eu/eurostat/statistics-explained/index.php/Quality_of_life_indicators

[3] https://www.imercer.com/content/mobility/quality-of-living-city-rankings.html

- Numbeo[4] also offers a result of Quality of Life ranking, it contains much quality of life indices, each index presented an aspect of life. The Quality of Life Index is an estimation of overall quality of life by using empirical formula which takes into account purchasing power index, pollution index, house price to income ratio (lower is better), cost of living index (lower is better), safety index, health care index, traffic commute time index (lower is better) and climate index.

7.1.2 Studies of Predicting Quality of Life

As studies shown in section 7.1.1 the quality of living in a city is always with respect of various of social aspects. Considering these factors [MFH+13] investigated on how geographic place correlates with and potentially influences societal levels of happiness. In this work, the happiness of different urban of the United States is leveled and analyzed with similarities in word expression, demographics, message length associated with education levels and obesity rates.

They combined a geo-tagged data set which consists of over 80 million words generated from over 10 million tweets posted in Twitter in 2011, which covered approximately 1% of the whole messages of the year, and characteristics of all 50 states and close to 400 urban populations. From the geographic level since that urban area boundaries often agglomerate small towns together, particularly when there are small towns close to larger urban or cities, the more details about these cities are also described in the thesis.

The method they used to measure sentiment is using LabMT word list [DHK+11], these individual words are scored for their average happiness independently by users of Amazon's Mechanical Turk service on a scale from 1 the saddest to 9 the happiest [KDH+12]. In order to measure the overall average happiness of people located in cities, they calculate the average happiness for a given text T containing N unique words based on word frequency distributions by:

$$h_{avg}(T) = \frac{\sum_{i=1}^{N} h_{avg}(w_i) f_i}{\sum_{i=1}^{N} f_i} = \sum_{i=1}^{N} h_{avg}(w_i) p_i \tag{7.1}$$

where f_i is the frequency of the ith word w_i in T for which happiness score is $h_{avg}(w_i)$ of the word, and $p_i = f_i / \sum_{i=1}^{N} f_i$ is the normalized frequency. More important to be noticed is that this thesis is implemented without respect on dependency between each words.

The variation of happiness across different cities is then analyzed on how individual word usage correlates with happiness and various social and economic factors.

A most recent related work presented a study of targeted aspect-based sentiment analysis dataset for urban neighborhoods.[SBLR16] Different to [MFH+13] this study gathered test data from a question answering (QA) platform where is far less constrained than review specific platforms. Only the QA data about urban neighborhoods which discussed by users are collected, because the content of these sentences is sometimes referring to more than one location, the sentiment analysis on this data are more complicated than

[4] `http://www.numbeo.com/quality-of-life/rankingscurrent.jsp`

other studies. The label of the sentiment of each sentence is annotated both towards the target (locations) and aspects (safety or transit-location), while SentiHood dataset is generated with 5215 sentences, in which 3862 sentences containing a single location and 1353 sentences containing multiple locations.

The main task of this study is a three-class classification for each aspect. They provide a list of tuples $\{l, a, p\}_{t=0}^{T}$ for interpreting the label of each given sentence s, where p is the polarity expressed for the aspect a of location l, which may be "Positive","Negative" and "None". Each sentence can have maximum T number of labels.

During aspect-based sentiment analysis, the four aspects namely Price, Safety Transit, General are chosen for features. For classification logistic regression combined with three feature selection methods are trained, as well as LSTM models a choice of neural networks is used to training model. Both of them developed strong baselines:

- For sentiment predicting result, the Logistic Regression classifier with n-gram and POS tags has gained the highest score with 0.875 for accuracy and the 0.905 for AUC scores, while the LSTM methods also have higher accuracy round 0.820 and round 0.840 for AUC.
- For aspect-based sentiment predicting result by using average AUC scores for each aspect, the logistic regression classifier with n-gram and POS tags has reached highest scores with 0.940 for "Price", 0.960 for "Safety", 0.879 for "Transit" and LSTM-Final gained 0.869 for "General".
- For target sentiment predicting result by using average AUC scores for predicting sentence with one location or multi locations, the logistic regression classifier with n-gram and POS tags has the highest score at 0.916 for "Single Location" and 0.907 for "Multi".

7.2 Researches about Text Categorization

At the early stage, researchers attempt to applying various algorithms to text categorization which is now considered as classification technologies in Machine Learning:

- With the rapid growth of online information, people concerns more on benefits delivered from these information. Manually analyzing and categorizing this information is much more difficult, thus building the classifier from examples for text categorization is advantageous.

  Support Vector Machines, which is a new learning technique introduced in many studies, is selected in [Joa98] as examine objects for text categorization. The study was done on two datasets, one is Reuter corpus generated on 9,603 training documents and 3,299 test documents of Reuters-21578 dataset, the other one is from the Ohsumed corpus, which 10,000 training documents in contract with 10,000 testing documents

  The performance of SVMs polynomial and SVMs RBF kernels both showed better results compared with four other conventional learning methods, namely Bayes, Rocchio, C4.5 and k-NN, in which k-NN classifier always outperformed. The analysis based on the result shows that the advantage of SVMs applying on higher dimensional feature spaces.

- In [NLM99] the performance of using Maximum Entropy for Text Classification is shown as a competitive algorithm. Maximum entropy it is a general technique for estimating probability distributions from data, which has been widely used for a variety of natural language tasks. In this study, the task of text classification is document based. Each document is represented by a set of word count features, maximum entropy estimates the conditional distribution of the class label given a document.

  The datasets of this work are three datasets from previous studies [5], one is the WebKB dataset contains web pages gathered from university computer science departments, in which 4,199 pages of student, faculty, course and project four most populous categories results in 23,830 words, the second data set is Industry Sector, in which 6,440 web pages of company [6]classified in 71 classes results in 29,964 vocabularies are selected, the third is Newsgroups dataset contains about 20,000 articles results in 57,040 words after removing redundant and meaningless text.

  This thesis using Improved Iterative Scaling to calculate the parameters of the maximum entropy classifier given a set of constraints. The Performance of applying maximum entropy compared with Scaled Naive Bayes and Regular Naive Bayes algorithm to these three datasets are different, on WebKB it showed lower error but for the other two datasets where maximum entropy performs worse than scaled naive Bayes. At the end of the work, the reasons of the results is explained that maximum entropy may be sensitive to poor feature selection

7.3 Researches about Sentiment Analysis

Along with the flourishing of artificial intelligence, applying the technology of machine learning as well as natural language processing technologies to analyze the sentiment of text information of different text resources like the Internet, newspaper, product reviews and so on, becomes available and mature. The development from studies for text categorization using Machine Learning technologies to analyze the social network information of textual form and to further extension for the sentiment analyze is presented as following.

Early researches about sentiment came up in [DC01] and [MYTF02]. In this period the work for sentiment analysis focuses on identifying the overall sentiment or polarity of a given text. [SBLR16] [Ana03] first mentioned the term sentiment analysis. Based on text categorization technologies it becomes a popular research realm and attracted more attentions. Until now the most studies for sentiment analysis are categorized into two fields: one is targeted sentiment analysis, which analyses opinion polarities towards

[5] http://www.cs.cmu.edu/~TextLearning

[6] www.marketguide.com

certain target of the given sentence like a tweet, the other one is aspect-based sentiment analysis, which takes the aspects of one sentence into account to deciding the polarities, which is also very practice due to its contribution to reviews of a product in some aspect.

The testbed of sentiment analysis comes out not only from social media but also news, financial reports etc. from where it originates, thus it is more meaningful to society, many researchers devote to improving the technical approaches for analyzing the sentiment of text in the last decade:

- [GBH09] represented results of machine learning algorithms for classifying the sentiment of Twitter messages using distant supervision. In contrast to other studies, this is the first study building classifier based on tweets since tweets are abundantly available and can be obtained by automated means.

  In this work, all the tweets are collected for the first time. The researchers have scraped this information by their own with queries through Twitter API to access them. Because the Twitter messages contain more or less emoticons, which expressed positive or negative emotion, the tweets are all gathered with emoticons from the period between April 6, 2009 to June 25, 2009. At the post-processing stage considering emoticons will impact classifier, this study labeled them as noisy and stripped them off. Moreover, any tweets contain both positive and negative emoticons are removed, as well as Retweets are all removed. At the end, a training dataset is generated with total 1,600,000 tweets, in which 800,000 tweets which with positive emoticons are labeled as positive, and 800,000 tweets which contain negative emoticons are labeled as negative. Test data is collected regardless of emoticons by using selected query terms of seven domains, which contains 177 negative tweets and 182 positive tweets labeled manually.

  Associating Unigrams, the combination of Unigrams and Bigrams and Parts of speech tags as features with machine learning algorithms like Naïve Bayes, Maximum Entropy and SVM, the performance of the classifier has above 80% accuracy when trained with emoticon data. As the main contribution of this work, the emoticon is taken into account for text categorization.

- [PP10] is an improvement of [GBH09], it presented an improved method for an automatic collection of a corpus that can be used to train a sentiment classifier. The corpus is collected of 300,000 text posts from Twitter. These collected corpus are divided into three classes: positive sentiments, negative sentiments, and a set with no sentiments, in which the tweets of negative and positive sentiment are collected in the same manner of [GBH09], which use positive emoticon and negative emoticon as query term, for the objects without sentiment are collected by querying 44 newspapers' names. Test dataset is selected as the same way as [GBH09], which contains total 216 samples included with 108 positive posts, 75 negative posts and 33 neutral posts.

  Before training classifier the corpus is analyzed for its frequency distributions of words, the result is consistent with Zipf's law, furthermore they used Tree Tagger for tag all the posts in the corpus in order to observe the distributions of tags of three datasets. They also used a function to calculate the P value for further comparison of positive and negative posts:

$$p_{1,2}^{T} = \frac{N_1^T - N_2^T}{N_1^T + N_2^T} \tag{7.2}$$

where N_1^T and N_2^T are numbers of tag T occurrences in the first and second sets respectively. Based on Parts of speech tags, the analysis focus on analyzing the distribution of words with different POS-tags of the two datasets which one contains sentiment and the other has no sentiment.

During the feature selection processing, the URL in the text is replaced as “URL”, text is split into segments by spaces and punctuation marks, stopwords are removed and N-gram is constructed, especially for negation words. Next SVMs classifier, CRF classifier and Naïve Bayes classifier are built and applied to test dataset, in which the multinomial Naïve Bayes classifier associated with bigram and POS-tags as features gained the best result. At the end they evaluated the performance of the classifier and proved that their technique is efficient and better than previously proposed methods, while the accuracy of the classifier in [GBH09] has obviously worse results when using it applying to the three-class dataset.

7.4 Summary

Previous the related studies of text categorization, sentiment analysis and predicting quality of lifes have been introduced. A brief comparison of the technologies these existed works used are shown in table 7.1.

From table 7.1 the trends of development of machine learning and the changes of research object could be found obviously.

- At the 1990s the studies are more about the improvement of the classifiers, researchers concerned more on the efficiency of a classifier for categorizing the text, SVM are introduced as a new outperform algorithm, Maximum Entropy is examined versus many conventional algorithms like k-NN, Naive Bayes etc. The Object of the research is on how to categorize documents into different classes.
- At the 2000s, the research object changed and the purpose is subdivided. They turn to analyze the text from social media, which may contain opinions of users, in this case, the categorization task becomes an analyze of positive, negative and even neutral opinion of an entity, which also called sentiment. The text formed data of social media resources contains not only products reviews, QA platform, but also microblogs like tweets. According to different content and characteristics of these text form, target sentiment analyze and aspect-based sentiment analyze are shown up for a different purpose. Target sentiment analysis is an analysis of a target of a sentence, which is good for analyzing short simple sentence with a single subject. Aspect-based sentiment analysis is good for analyzing the reviews of a product. The customers who have reviews of one article are always written around different aspects, from this side of view aspect-based sentiment analyze is meaningful for the special area.

Table 7.1: Comparison of Related Works

Name	**Contributions**	**Datasets**	**Machine Learning Technology**	**Others**
Text Categorization with Support Vector Machines: Learning with Many Relevant Features [Joa98]	Analysis of Performance of SVMs to other conventional algorithms	Reuters Corpus, Ohsumed Corpus	SVMs polynomial kernels, SVMs RBF kernels, Bayes, Rocchio,C4.5, k-NN	
Using Maximum Entropy for Test Classification [NLM99]	Provement of Maximum Entropy as a competitive algorithm for text categorization	WebKB, Industry Sector, News-groups	Maximum Entropy, Scaled Naive Bayes, Regular Naive Bayes	
Twitter Sentiment Classification using Distant Supervision [GBH09]	Classifying the sentiment of Twitter messages by using Distant Supervision	Tweets	Naive Bayes, Maximum Entropy, SVM	Emoticons as noisy labeled
Twitter as Corpus for Sentiment Analysis and Opinion Mining [PP10]	Sentiment analysis based on linguistic analysis	Tweets	Multinomial Naive Bayes, SVM, CRF	introduces methods dealing with emoticons
The Geography of Happiness: Connecting Twitter Sentiment and Expression, Demographics, and Objective Characteristics of Place [MFH^{+}13]	Analysis of happiness of urbans based on Tweets, correlated with social factors	Tweets	None	Word list with score of happiness
SentiHood: Targeted Aspect Based Sentiment Analysis Dataset for Urban Neighborhoods [SBLR16]	Improvement of Targeted as well as Aspect based Sentiment Analysis on reviews of Urban Neighborhoods of QA platform	SentiHood dataset collected from QA platform	Logistic Regression, Long Short-Term Memory	

- From the 2010s the research became more diversified, the sentiment analyze are applying to wider fields, researchers didn't only satisfy with the task of sentiment analyze itself but also concerned on the relation between classification result and various factors, thus they are doing more on feature analysis and text processing, moreover they tends to analyse complicated text towards both target and aspects.

8 Conclusions

8.1 Summary

Along with the development of artificial intelligence, machine learning becomes more and more popular. A main application of machine learning is to teaching a computer to categorize things, which means classifying things through learning from proper data. This diploma thesis described a study of creating a classifier based on the data of Twitter, which could predict the sentiment of a living place. In order to implement this work the following issues are presented in the study:

- In the preparing stage the related works about sentiment analysis, text classification and quality of life are studied and analyzed. (chapter 7) Based on studying related fundamentals of machine learning, classification and evaluation method, general workflow of machine learning and the methodology of this thesis is defined specifically. (Chapter 4)

- In the data collecting stage, based on the ranking list of living quality in 230 cities offered by Mercer, more than 100,000 tweets of each city of most recent period from 01.08.2015 to 31.08.2016 from Twitter have been collected as raw data, which guarantees that the tweets are not only enough for learning, but also in real-time status and reflect the most sentiment of a city. comparing to the unbalance of the amount of collected tweets for each city 5,000 tweets of each city are selected randomly as the original dataset. (chapter 3)

- Based on this methodology, data is preprocessed and the tweets are tokenized by taking use of a natural language processing tool TweetNLP. Then logistic regression models and SVM models are trained. The extracted features are using bag of words method. The evaluation is implemented by calculating the accuracy of each classifier. Comparing to SVM classifier the best result of logistic regression classifier with 0.795652 has been chosen as baseline.

  Improving the baseline through removing city names, most stopwords and punctuations, tokens are also transformed into lowercase to avoid repeated features, where the data matrix is also scaled. As a result of using this method, a strong baseline has been improved with accuracy score 0.8793913.

  Further attempts have been implemented for improving the classifier and A sentiment analysis word list is used to generate three extra features to improving the models are also described in this section. Unfortunately, the effect of using these three features individually declined the baseline. (chapter 5)

Table 8.1: Analysis of URLs

URL	Feature Weights	Categorization	Tokenizations
`https://www`	0.012809	unknown	none
`getluckyhotels.com/travel-all-hot`	0.004395	advertisement	get lucky hotels tranvel all hot
`map-game.com/nouakchott`	-0.001414	game	map game nouakchott
`twitter.com/photoschedule/`	-0.010277	unknown	photo schedule
`btc2bid.com/Deals/China/Bu`	-0.001099	unknown	btc2bid Deals China Bu

- Cross-validation method was used to split the dataset into training part and testing part to prepare the data for training model, the features are analyzed by comparing the feature weights before and after improving the classifiers.

  In analysis of features stage, the feature weight of each classifier has been compared. Besides of the most impacting features like city names, stop words and punctuations the country names, hashtags, URLs were found that influenced the classifier to varying degrees. (chapter 6)

8.2 Future Work

In section 8.1, the whole work of this thesis have been summarized. From the anthropological and sociological dimensions the predicting of a city life is more meaningful. During the implementation procedure some aspects could be improved in the future:

- **Generating Dataset:** In data collecting phase tweets have been chosen as data set, considering the characters of these short text and uncontrolled, it contains more individual sentiments of users, sometimes they are without specific aspects and could not reflect the life of a city completely, combining the data with other data source such as data from government are more reliable.

  The query terms used for collecting city data are city names, from this point of view, the query terms may be more detailed with objects, such as collecting data from users who live in the city.

  In this thesis tweets for each city have been gathered, the sentiment of all tweets of each city are considered as a whole, another choice is that consider sentiment of each tweet of a city and label them individually.

- **Automatic Class Prediction:** logistic regression model and SVM model have been used to train data. For SVM model since high dimension features space versus little samples, its performance is not better than logistic regression model. In order to improve the performance, the logistic regression classifier regularization may be attempted by tuning the parameters.

Another improvement is to train a ranker, the similar working principle of classifying based on probability theory could be used to make grading the quality of living a place.

- **Feature extraction:** Feature extraction stage is more important for improving the performance of a classifier through analyzing the features. By now the analysis of features presented in section 6.2 is based on natural language processing, one improvement could be done is to take a look at the URLs.

  Instead of removing all URLs, We could analyze the information contained in the URLs by comparing them and further tokenization. As shown in table 8.1 the URLs have high feature weights, which means that they impact more to classification. Some of them have no special meanings, but some like "urlgetluckyhotels.com" may contain more information, from which it is possible to extract extra features and its influence to classification is unknown.

Appendix

A.1 Overview of City List for Cross-Validation

A.2 Feature Weights of Logistic Regression Model

Table A.2: First 100 Features of LR Model on 10th Fold Dataset

No.	Feature Names	Feature Weights
1	#bandarseribegawan	0.026052
2	Pitre	0.020225
3	#BandarSeriBegawan	0.019729
4	#brunei	0.015174
5	#PortLouis	0.013826
6	Pointe	0.013253
7	Guadeloupe	0.012672
8	Pointe-a-Pitre	0.012561
9	https://www	0.011197
10	#noumea	0.010076
11	#Noumea	0.009705
12	#Mauritius	0.009516
13	A	0.009323
14	#Brunei	0.008283
15	#portlouis	0.008278
16	#Nurnberg	0.007041
17	@Pointe	0.007026
18	a	0.006506
19	to	0.006333

Table A.2 – continued from previous page

No.	Feature Names	Feature Weights
20	Brunei	0.006007
21	#Shanghai	0.005937
22	New	0.005881
23	at	0.005875
24	Gualdeloupe	0.005748
25	#Luxembourg	0.005465
26	#Tunis	0.005393
27	#nurnberg	0.005042
28	-à-	0.004897
29	éa	0.004612
30	=	0.004490
31	#newcaledonia	0.004482
32	Norwegian	0.004475
33	Noumea	0.004414
34	#mauritius	0.004140
35	#Bamako	0.004054
36	#TelAviv	0.004050
37	#Geneva	0.003822
38	#Baghdad	0.003821
39	#Noum	0.003818
40	Caledonia	0.003777
41	Paris	0.003776
42	#Damascus	0.003771
43	Flight	0.003694
44	#Montevideo	0.003554
45	#Helsinki	0.003522
46	.	0.003474
47	#Job	0.003464
48	#Ljubljana	0.003437

Table A.2 – continued from previous page

No.	Feature Names	Feature Weights
49	#jobs	0.003418
50	Sanaa	0.003387
51	#Athens	0.003367
52	#PanamaCity	0.003329
53	#Brussels	0.003248
54	http://	0.003243
55	#Bucharest	0.003158
56	Seri	0.003151
57	Begawan	0.003149
58	#Victoria	0.003109
59	from	0.003040
60	#Tripoli	0.002938
61	Air	0.002925
62	#Nairobi	0.002909
63	Louis	0.002894
64	#Karachi	0.002837
65	#Belfast	0.002815
66	@	0.002787
67	#Vienna	0.002770
68	PITRE	0.002768
69	POINTE	0.002768
70	Pointe-A-Pitre	0.002768
71	#Kolkata	0.002762
72	#Warsaw	0.002749
73	#Djibouti	0.002748
74	hotel	0.002738
75	#AddisAbaba	0.002731
76	¿	0.002721
77	#Brasilia	0.002709

Table A.2 – continued from previous page

No.	Feature Names	Feature Weights
78	#Johannesburg	0.002685
79	#giwbrunei	0.002577
80	#bruneidarussalam	0.002576
81	nonstop	0.002561
82	#Perth	0.002555
83	#Vilnius	0.002533
84	#Singapore	0.002513
85	#HongKong	0.002490
86	#NewCaledonia	0.002461
87	Mauritius	0.002404
88	#Berlin	0.002373
89	’	0.002339
90	#Skopje	0.002326
91	Bandar_Seri_Begawan&country	0.002290
92	Brunei&source	0.002290
93	#Syria	0.002275
94	#mosque	0.002263
95	#Dallas	0.002256
96	#Monterrey	0.002227
97	#victoria	0.002132
98	#Minneapolis	0.002115
99	#Nassau	0.002106
100	#Panama	0.002099

Table A.3: Last 100 Features of LR Model on 10th Fold Dataset

No.	Feature Names	Feature Weights
1	#Quito	-0.002770
2	#betabookings	-0.002773
3	#Livescore	-0.002794
4	ALERT	-0.002797
5	#hostel	-0.002818
6	SCORE	-0.002836
7	#Basketball	-0.002862
8	#Casablanca	-0.002871
9	@ScoresPro	-0.002891
10	#NewDelhi	-0.002907
11	#Changchun	-0.002926
12	#ElSalvador	-0.002943
13	#jakarta	-0.002948
14	#Jeddah	-0.003000
15	#africa	-0.003016
16	Sun	-0.003019
17	#Riyadh	-0.003056
18	#Trinidad	-0.003070
19	#antananarivo	-0.003077
20	#Windhoek	-0.003082
21	Xian	-0.003083
22	#myunjobs	-0.003088
23	..	-0.003098
24	@http	-0.003103
25	Ashgabat	-0.003106
26	#Bogota	-0.003119
27	#saopaulo	-0.003131
28	minutes	-0.003144

Table A.3 – continued from previous page

No.	Feature Names	Feature Weights
29	30	-0.003229
30	UTC	-0.003276
31	https://	-0.003315
32	#Brazil	-0.003316
33	#Guatemala	-0.003329
34	pm	-0.003338
35	Phnom	-0.003396
36	html	-0.003418
37	#Lima	-0.003431
38	#Mauritania	-0.003432
39	#Vientiane	-0.003450
40	#Manila	-0.003495
41	twitter.com/PhotoSchedule/	-0.003505
42	Nouakchott	-0.003516
43	#Madagascar	-0.003520
44	#Photo	-0.003688
45	city	-0.003706
46	#Gaborone	-0.003747
47	#PortOfSpain	-0.004020
48	Province	-0.004043
49	#Jakarta	-0.004135
50	#SaoPaulo	-0.004156
51	#Lom	-0.004165
52	#Gambia	-0.004186
53	#Benin	-0.004209
54	China	-0.004262
55	#Qingdao	-0.004280
56	#Douala	-0.004341
57	Good	-0.004376

Table A.3 – continued from previous page

No.	Feature Names	Feature Weights
58	#Phnom	-0.004438
59	#Togo	-0.004446
60	#Bangkok	-0.004521
61	#Turkmenistan	-0.004575
62	local	-0.004753
63	#riodejaneiro	-0.004755
64	Salvador	-0.004768
65	#cotonou	-0.004864
66	#Africa	-0.005043
67	#PortofSpain	-0.005181
68	,	-0.005455
69	#Niger	-0.005463
70	Spain	-0.005482
71	City	-0.005612
72	#Dushanbe	-0.005681
73	#Conakry	-0.005754
74	#Blantyre	-0.005764
75	#manaus	-0.005930
76	#santodomingo	-0.005964
77	#SantoDomingo	-0.006029
78	:	-0.006221
79	#portofspain	-0.006239
80	#china	-0.006603
81	#Managua	-0.006713
82	é	-0.006867
83	#sansalvador	-0.006877
84	#Lome	-0.006900
85	;	-0.007047
86	Penh	-0.007194

Table A.3 – continued from previous page

No.	Feature Names	Feature Weights
87	#Manaus	-0.007366
88	#jilin	-0.008189
89	#Antananarivo	-0.009048
90	of	-0.010064
91	#Shenyang	-0.010533
92	#Niamey	-0.011569
93	#SanSalvador	-0.011580
94	#Cotonou	-0.011818
95	#Ashgabat	-0.012155
96	#Nouakchott	-0.012605
97	#China	-0.012883
98	#Banjul	-0.013017
99	in	-0.013555
100	#Jilin	-0.032227

A.3 Feature Weights of Best Improved Logistic Regression Model

Table A.4: First 100 Features of Improved LR Model on 10th Fold Dataset

No.	Feature Names	Feature Weights
1	#brunei	0.055305
2	#mauritius	0.028817
3	pitre	0.024389
4	shanghai	0.022281
5	pointe	0.017676
6	pointe-a-pitre	0.017329

Table A.4 – continued from previous page

No.	Feature Names	Feature Weights
7	#newcaledonia	0.015670
8	brunei	0.014463
9	guadeloupe	0.013928
10	https://www	0.012809
11	new	0.011475
12	#	0.010425
13	éa	0.010260
14	noumea	0.010117
15	#noum	0.009274
16	seri	0.009233
17	begawan	0.009217
18	#tunisia	0.008954
19	#panama	0.008858
20	caledonia	0.008563
21	ília	0.008527
22	#bras	0.008508
23	#germany	0.008420
24	chongqing	0.008157
25	#southafrica	0.008032
26	#jobs	0.007938
27	today	0.007572
28	nanjing	0.007470
29	#bsb	0.007460
30	#bruneidarussalam	0.007242
31	#switzerland	0.007188
32	@pointe	0.007071
33	#beijing	0.006968
34	c	0.006592
35	nurnberg	0.006535

Table A.4 – continued from previous page

No.	Feature Names	Feature Weights
36	¿	0.006451
37	#slovenia	0.006439
38	see	0.006412
39	#australia	0.006330
40	current	0.005972
41	#belgium	0.005946
42	high	0.005938
43	#giwbrunei	0.005925
44	mauritius	0.005794
45	gualdeloupe	0.005776
46	#melbourne	0.005776
47	panama	0.005719
48	#airpollution	0.005702
49	paris	0.005688
50	#uruguay	0.005658
51	#cbr	0.005641
52	aqi	0.005627
53	kong	0.005579
54	business	0.005526
55	#airquality	0.005481
56	attacks	0.005456
57	forecast	0.005305
58	brussels	0.005271
59	#bandar	0.005267
60	bandar_seri_begawan&country	0.005267
61	brunei&source	0.005267
62	|	0.005249
63	-á-	0.005134
64	great	0.005115

Table A.4 – continued from previous page

No.	**Feature Names**	**Feature Weights**
65	weather	0.004864
66	hong	0.004813
67	feeling	0.004779
68	#mosque	0.004749
69	like	0.004691
70	#bern	0.004665
71	#nuremberg	0.004659
72	live	0.004605
73	geneva	0.004554
74	tomorrow	0.004538
75	norwegian	0.004508
76	getluckyhotels.com/travel-all-hot	0.004395
77	moderate	0.004343
78	truth	0.004291
79	malicious	0.004095
80	#london	0.004076
81	propaganda	0.004043
82	#nouvellecaledonie	0.003972
83	incident	0.003960
84	bandar	0.003954
85	week	0.003921
86	#taiwan	0.003916
87	#foxnews	0.003912
88	france	0.003909
89	the	0.003907
90	@	0.003905
91	it’s	0.003884
92	little	0.003870
93	weekend	0.003842

Table A.4 – continued from previous page

No.	Feature Names	Feature Weights
94	massacre	0.003819
95	#chongqingjobs	0.003814
96	#uae	0.003813
96	christmas	0.003812
98	#nbc	0.003805
99	#abc	0.003796
100	feel	0.003794

Table A.5: Last 100 Features of Improved LR Model on 10th Fold Dataset

No.	Feature Names	Feature Weights
1	experiences	-0.004984
2	#property	-0.005157
3	breaking	-0.005199
4	#albania	-0.005204
5	domingo	-0.005217
6	#kyrgyzstan	-0.005220
7	nicaragua	-0.005246
8	capital	-0.005260
9	shenyang	-0.005267
10	#pakistan	-0.005283
11	jakarta	-0.005318
12	pour	-0.005343
13	severe	-0.005369
14	#ecuador	-0.005426
15	plant	-0.005439
16	#c	-0.005484

Table A.5 – continued from previous page

No.	Feature Names	Feature Weights
17	apply	-0.005494
18	#photography	-0.005647
19	leader	-0.005768
20	#brasil	-0.005790
21	#turkmenistan	-0.005852
22	#phnom	-0.005872
23	#events	-0.005932
24	#brazil	-0.006096
25	partly	-0.006111
26	petersburg	-0.006164
27	#next420	-0.006212
28	el	-0.006214
29	manaus	-0.006306
30	rise	-0.006319
31	explosion	-0.006329
32	#explosion	-0.006516
33	next420.info	-0.006538
34	#togo	-0.006546
35	://	-0.006567
36	#indonesia	-0.006805
37	#weather	-0.006810
38	#san	-0.006849
39	#peru	-0.006864
40	#malawi	-0.006933
41	é	-0.006987
42	#lebanon	-0.007104
43	#india	-0.007117
44	#amazonas	-0.007175
45	#myunjobs	-0.007235

Table A.5 – continued from previous page

No.	Feature Names	Feature Weights
46	wind	-0.007311
47	#dominicanrepublic	-0.007330
48	ne	-0.007372
49	#thailand	-0.007492
50	@http	-0.007512
51	#vietnam	-0.007591
52	#portofspain	-0.007602
53	banjul	-0.007787
54	#sunrise	-0.007792
55	sun	-0.007810
56	penh	-0.007889
57	&	-0.007907
58	#laos	-0.007971
59	kingston	-0.007984
60	@pdchina	-0.008186
61	#philippines	-0.008317
62	#trinidad	-0.008709
63	#guinea	-0.008789
64	minutes	-0.009125
65	:	-0.009184
66	30	-0.009624
67	spain	-0.009633
68	utc	-0.009635
69	good	-0.009666
70	time	-0.009669
71	vs	-0.009822
72	gambia	-0.009839
73	#tajikistan	-0.010034
74	#photo	-0.010136

Table A.5 – continued from previous page

No.	Feature Names	Feature Weights
75	twitter.com/photoschedule/	-0.010277
76	alert	-0.010347
77	#livescore	-0.010606
78	score	-0.010714
79	(-)	-0.010749
80	https://	-0.010966
81	nouakchott	-0.011114
82	#basketball	-0.011362
83	@scorespro	-0.011650
84	#madagascar	-0.011878
85	#nicaragua	-0.011959
86	pm	-0.012412
87	#mauritania	-0.012992
88	city	-0.013353
89	#ashgabat	-0.014536
90	jilin	-0.015019
91	local	-0.015170
92	salvador	-0.016637
93	#elsalvador	-0.016684
94	#benin	-0.017346
95	#gambia	-0.018379
96	#changchun	-0.019460
97	#africa	-0.021010
98	#niger	-0.023081
99	province	-0.024933
100	#china	-0.030837

A.4 Result Table under All Conditions Based on Best Baseline

Table A.6: Result Table of All Conditions Based on Best Baseline

	no cn sw lc	no url	no hashtag	no country name	p-words	n-words	p/n words	maxabs scale	average scores
B									0.795652
0	√								0.873913
1	√	√							0.821739
2	√	√						√	0.821739
3	√	√			√	√	√	√	0.873913
4	√	√			√	√	√		0.830435
5	√	√			√			√	0.873913
6	√	√			√				0.830435
7	√	√				√		√	0.873913
8	√	√				√			0.830435
9	√	√					√	√	0.873913
10	√	√					√		0.834783
11	√	√			√	√		√	0.873913
12	√	√			√	√			0.830435
13	√	√				√	√	√	0.873913
14	√	√				√	√		0.830435
15	√			√					0.856522
16	√			√				√	0.856522
17	√			√	√	√	√	√	0.856522
18	√			√	√	√	√		0.847826
19	√			√	√			√	0.856522
20	√			√	√				0.865217
21	√			√		√		√	0.856522
22	√			√		√			0.856522
23	√			√			√	√	0.856522
24	√			√			√		0.856522
25	√			√	√	√		√	0.856522
26	√			√	√	√			0.852174
27	√			√		√	√	√	0.856522
28	√			√		√	√		0.856522
29	√		√						0.856522
30	√		√					√	0.856522

Table A.6: Result Table of All Conditions Based on Best Baseline

	no cn sw lc	**no url**	**no hashtag**	**no country name**	**p-words**	**n-words**	**p/n words**	**maxabs scale**	**average scores**
31	√		√		√	√	√	√	0.847826
32	√		√		√	√	√		0.847826
33	√		√		√			√	0.847826
34	√		√		√				0.847826
35	√		√			√		√	0.847826
36	√		√			√			0.843478
37	√		√				√	√	0.847826
38	√		√				√		0.856522
39	√		√		√	√		√	0.847826
40	√		√		√	√			0.847826
41	√		√			√	√	√	0.847826
42	√		√			√	√		0.847826

Table A.1: Random City Lists of Cross-Validation

k-th Fold	Testing City List
0	Bamako, Dallas, Lahore, Kinshasa, San Salvador, Cape Town, Abu Dhabi, St. Louis, Bishkek, Lima, Brazzaville, Havana, Sydney, Santo Domingo, Monterrey, Beirut, Djibouti, Bangui, Port of Spain, Luxembourg, Manila, Houston, Perth
1	Windhoek, Milan, Dar es Salaam, Tallinn, Hong Kong, Auckland, Kiev, Shanghai, Bangalore, Buenos Aires, Hyderabad, Damascus, Geneva, Paris, Athens, Blantyre, Singapore, Vancouver, Budapest, Bucharest, Almaty, Hamburg, Berlin
2	Ouagadougou, Miami, Calgary, New Delhi, N'Djamena, Nassau, Kingston, Philadelphia, Phnom Penh, Jakarta, Wroclaw, Xi'an, Johor Bahru, Chengdu, Maputo, Sana'a, Warsaw, Skopje, Libreville, Aberdeen, Hanoi, Birmingham, Cairo
3	Brisbane, Vientiane, Niamey, Manaus, Abidjan, Nouakchott, Quito, Tel Aviv, Munich, San Jose, Santiago, Douala, Abuja, Port-au-Prince, Saint Petersburg, Bandar Seri Begawan, Shenyang, Belgrade, Victoria, New York City, Sao Paulo, 'Colombo', Mumbai
4	Khartoum, Yerevan, Detroit, Guatemala City, Bern, Lyon, Vienna, Pointe-a-Pitre, Port Louis, Yokohama, Toronto, Prague, Stockholm, Seattle, Tehran, Dakar, Pittsburgh, Wellington, Gaborone, Lisbon, Addis Ababa, Los Angeles, Busan
5	Kuwait City, Nanjing, Banjul, Nairobi, Lome, Sarajevo, Vilnius, Taipei, Canberra, Panama City, Caracas, Kuala Lumpur, Montreal, Adelaide, Jeddah, Riga, Boston, Algiers, Yaounde, Tbilisi, Moscow, Minsk, Tirana
6	Cotonou, Minneapolis, Melbourne, Dushanbe, Beijing, Glasgow, Belfast, Bogota, Rio de Janeiro, Sofia, Ashkhabad, Stuttgart, Islamabad, La Paz, Zagreb, Tunis, Lusaka, Bangkok, Taichung, San Juan, Brussels, Honolulu, Pune
7	Qingdao, Riyadh, Durban, Amsterdam, Mexico City, Chicago, Tripoli, Noumea, Montevideo, Rabat, Karachi, Helsinki, Baku, Ljubljana, Oslo, Ottawa, Dhaka, Brasilia, Casablanca, Johannesburg, Nagoya, Jilin, Nurnberg
8	Luanda, Madrid, Managua, Accra, Tokyo, Amman, Frankfurt, Chongqing, Conakry, Shenzhen, Chennai, Dubai, Zurich, Edinburgh, Manama, Kobe, Muscat, Antananarivo, Bratislava, Baghdad, Yangon, Kigali, Kolkata
9	Kampala, Harare, Seoul, Asuncion, Atlanta, London, Tegucigalpa, Barcelona, Doha, Rome, Leipzig, Guangzhou, Lagos, San Francisco, Osaka, Ho Chi Minh City, Tashkent, Dusseldorf, Limassol, Dublin, Copenhagen, Washington, Istanbul

Bibliography

[Ana03] Sentiment Analyzer. Extracting sentiments about a given topic using natural language processing techniques; jeonghee yi et al; ibm. In *Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03) 0-7695-1978-4/03*, volume 17, 2003.

[Anz12] Yuichiro Anzai. *Pattern recognition and machine learning*. Elsevier, 2012.

[DC01] Sanjiv Das and Mike Chen. Yahoo! for amazon: Extracting market sentiment from stock message boards. In *Proceedings of the Asia Pacific finance association annual conference (APFA)*, volume 35, page 43. Bangkok, Thailand, 2001.

[DHK$^+$11] Peter Sheridan Dodds, Kameron Decker Harris, Isabel M Kloumann, Catherine A Bliss, and Christopher M Danforth. Temporal patterns of happiness and information in a global social network: Hedonometrics and twitter. *PloS one*, 6(12):e26752, 2011.

[DWT$^+$14] Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *ACL (2)*, pages 49–54, 2014.

[Elk] Charles Elkan. Evaluating classifiers. *University of San Diego, California, retrieved [01-11-2012] from http://cseweb.ucsd.edu/~elkan/250Bwinter2012/, volume=250, year=2012, publisher=Citeseer.*

[GBH09] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1:12, 2009.

[HBSK16] T Hercig, T Brychcın, L Svoboda, and M Konkol. Uwb at semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), Association for Computational Linguistics, San Diego, California*, pages 354–361, 2016.

[JL10] ZQ John Lu. The elements of statistical learning: data mining, inference, and prediction. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 173(3):693–694, 2010.

[JO11] Yohan Jo and Alice H Oh. Aspect and sentiment unification model for online review analysis. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 815–824. ACM, 2011.

[Joa98] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer, 1998.

[JYZ+11] Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 151–160. Association for Computational Linguistics, 2011.

[JZSC09a] Bernard J Jansen, Mimi Zhang, Kate Sobel, and Abdur Chowdury. Micro-blogging as online word of mouth branding. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*, pages 3859–3864. ACM, 2009.

[JZSC09b] Bernard J. Jansen, Mimi Zhang, Kate Sobel, and Abdur Chowdury. Micro-blogging as online word of mouth branding. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '09, pages 3859–3864, New York, NY, USA, 2009. ACM.

[KDH+12] Isabel M Kloumann, Christopher M Danforth, Kameron Decker Harris, Catherine A Bliss, and Peter Sheridan Dodds. Positivity of the english language. *PloS one*, 7(1):e29484, 2012.

[L+09] Tie-Yan Liu et al. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331, 2009.

[LOCT11] Bin Lu, Myle Ott, Claire Cardie, and Benjamin K Tsou. Multi-aspect sentiment analysis with topic models. In *2011 IEEE 11th International Conference on Data Mining Workshops*, pages 81–88. IEEE, 2011.

[MCM13] Ryszard S Michalski, Jaime G Carbonell, and Tom M Mitchell. *Machine learning: An artificial intelligence approach.* Springer Science & Business Media, 2013.

[MFH+13] Lewis Mitchell, Morgan R Frank, Kameron Decker Harris, Peter Sheridan Dodds, and Christopher M Danforth. The geography of happiness: Connecting twitter sentiment and expression, demographics, and objective characteristics of place. *PloS one*, 8(5):e64417, 2013.

[MS99] Christopher D Manning and Hinrich Schütze. *Foundations of statistical natural language processing*, volume 999. MIT Press, 1999.

[MYTF02] Satoshi Morinaga, Kenji Yamanishi, Kenji Tateishi, and Toshikazu Fukushima. Mining product reputations on the web. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 341–349. ACM, 2002.

[NLM99] Kamal Nigam, John Lafferty, and Andrew McCallum. Using maximum entropy for text classification. In *IJCAI-99 workshop on machine learning for information filtering*, volume 1, pages 61–67, 1999.

[NRR+16] Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. Semeval-2016 task 4: Sentiment analysis in twitter. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval 2016), San Diego, US (forthcoming)*, 2016.

[PGP+15] Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Association for Computational Linguistics, Denver, Colorado*, pages 486–495, 2015.

[PL08] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135, January 2008.

[Pow11] David Martin Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. 2011.

[PP10] Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10, pages 1320–1326, 2010.

[RNC+03] Stuart Jonathan Russell, Peter Norvig, John F Canny, Jitendra M Malik, and Douglas D Edwards. *Artificial intelligence: a modern approach*, volume 2. Prentice hall Upper Saddle River, 2003.

[SBLR16] Marzieh Saeidi, Guillaume Bouchard, Maria Liakata, and Sebastian Riedel. Sentihood: Targeted aspect based sentiment analysis dataset for urban neighbourhoods. *arXiv preprint arXiv:1610.03771*, 2016.

[Seb02] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.

[VZ15] Duy-Tin Vo and Yue Zhang. Target-dependent twitter sentiment classification with rich automatic features. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pages 1347–1353, 2015.

All links were last followed on February 2, 2017

Acknowledgement

I am heartily thankful to my supervisor Dr. Christian Scheible from the University of Stuttgart for his encouragement, guidance and support in all the phases of this diploma thesis. Special thanks to my family, friends for their moral support.

Feifei Liu

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

Stuttgart, February 2, 2017 ______________________

(Feifei Liu)