Automatic Identification of Locative Expressions from Informal Text

A thesis presented
by

Fei Liu

to
The Department of Computing and Information Systems
in total fulfillment of the requirements
for the degree of
Master of Software Systems Engineering

The University of Melbourne
Melbourne, Australia
June 2013

Declaration

I certify that:

(i) this thesis does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any university; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person where due reference is not made in the text.

(ii) the thesis is fewer than 25,000 words in length (excluding text in images, table, bibliographies and appendices).

Signed: ______________________________ Date: ____________________

Thesis advisor(s)
Prof Timothy Baldwin
Dr Maria Vasardani

Author
Fei Liu

Automatic Identification of Locative Expressions from Informal Text

Abstract

Informal place descriptions that are rich in locative expressions can be found in various contexts. The ability to extract locative expressions from such informal place descriptions is at the centre of improving the quality of services, such as interpreting geographical queries and emergency calls. While much attention has been focused on the identification of formal place references (e.g., *Rathmines Road*) from natural language, people tend to make heavy use of informal place references (e.g., *my bedroom*).

This research addresses the problem by developing a model that is able to automatically identify locative expressions from informal text. Moreover, we study and discover insights of what aspects are helpful in the identification task.

Utilising an existing manually annotated corpus, we re-annotate locative expressions and use them as the gold standard. Having the gold standard ready, we take a machine learning approach to the identification task with well-reasoned features based on observation and intuition. Further, we study the impacts of various feature setups on the performance of the model and provide analyses of experiment results. With the best performing feature setup, the model is able to achieve significant increase in performance over the baseline systems.

Contents

List of Figures

List of Tables

Acknowledgments

I would like to express my gratitude to both my supervisors Timothy Baldwin and Maria Vasardani for their constant encouragement and inspiration during the completion of the thesis, for the weekly meetings, friendly approach to supervising and useful feedbacks on both the research methodologies and the thesis presentation.

I would like to thank my research coordinator Simon Milton for his excellent support and guidance.

I would also like to thank Igor and Ned for providing resources that play critical roles in this research.

Additionally, I would like to thank the University of Melbourne for giving me such an opportunity to study at a world-class university and spend wonderful three months on an exciting research project.

Last but not least, a special thank you to all colleague students at my lab for the great company, support and fun.

Dedicated to my parents, grandparents, and friends

Chapter 1

Introduction

Informal place descriptions can be found in various contexts, such as emergency calls, local web search and route directions (Winter *et al.*, 2011). Two examples are shown in Example (1.1) and Example (1.2). In both examples, the informal place references are underlined and the formal place references are double-underlined.

(1.1) I am in my bedroom at home, on Rathmines Road, Hawthorn East.

(1.2) corner of como parade east and parkers road, in the library building, next to the bee shop and across from the parkdale railway station

Such informal place descriptions are rich in locative expressions. In both examples, informal place references (e.g., *my bedroom*, *the library building*), mixed with formal ones (e.g., *Rathmines Road*, *parkers road*), are extensively used to describe locations. Eventually, locative expressions can be constructed based on these place references.

The ability to analyse the components of place descriptions is at the centre of the task of extracting locative expressions from place descriptions expressed in informal natural language.

In the task of analysing place descriptions, descriptions can be classified as formal or informal. Formal place descriptions tend to follow certain patterns and are structured. Example (1.3) and Example (1.4) are both written in a standard address format and they refer to the same places as Example (1.1) and Example (1.2) respectively. On the contrary, informal place descriptions are unstructured and often used in scenarios where people attempt to describe one's whereabouts (Example (1.1)) or giving directions (Example (1.2)).

(1.3) 192 Rathmines Road, Hawthorn East VIC 3123

(1.4) 96 Parkers Road, Parkdale VIC 3195

A place description, formal or informal, often consists of one or more place references which are key components of locative expressions. Place references can be further categorised as formal or informal. Formal place references can be represented in many forms, such as geographic coordinates (e.g., *latitude: -37.99370691, longitude: 145.0777912*) and geospatial named entities (e.g., *parkers road*). Informal place references, on the other hand, are mostly geospatial noun phrases (e.g., *my bedroom*, *home*). A locative expression is often comprised of one or more place references.

While much work has been focused on the identification of formal place references from natural language (Mikheev *et al.*, 1999; Zhou and Su, 2002; Ritter *et al.*, 2011), people tend to make heavy use of informal place references.

Identifying locative expressions is a difficult task as they are flexible and consist of both formal and informal place references. Adopting a named entity recogniser to analyse place descriptions works for formal place references (e.g., geospatial named entities). However, given that locative expressions are comprised of both informal and formal place references, we hypothesise that applying named entity recogniser to the task of extracting locative expressions from informal place descriptions is unlikely to work well. As can be seen in Example 1.5, *Stanford Named Entity Recognizer* (Section 3.4.1) is able to correctly identify one formal place reference *Hawthorn East* as double underlined. The other formal place reference *Rathmines Road* can only be recognised partially with the word *Rathmines* correctly spotted but *Road* left out. As for informal place references, *my bedroom* and *home*, none is identified by *Stanford Named Entity Recognizer*.

(1.5) I am in my bedroom at home, on Rathmines Road, Hawthorn East.

In this research, we aim to develop a system that is able to automatically identify locative expressions from informal text. A diagram of the input/output of the system is presented in Figure 1.1. The input place description is shown at the top of the figure and the output is at the bottom with locative expressions identified as underlined.

1.1 Motivation

Huge amounts of data are being generated daily by millions of users via communication channels such as social media sites and a fairly large proportion of them are geolocation-related messages (place descriptions). The inability to identify locative expressions hinders the improvement of the quality of services, such as interpreting geographical queries and emergency calls, and extracting geolocation information relating to appointments in emails (similar to *Gmail*'s *Google Calendar* integration[1] where users are able to add events to *Google Calendlar* held at a specific time recognised in the contents of emails in *Gmail*).

[1] http://gmailblog.blogspot.com.au/2013/05/add-events-to-google-calendar-from-gmail.html

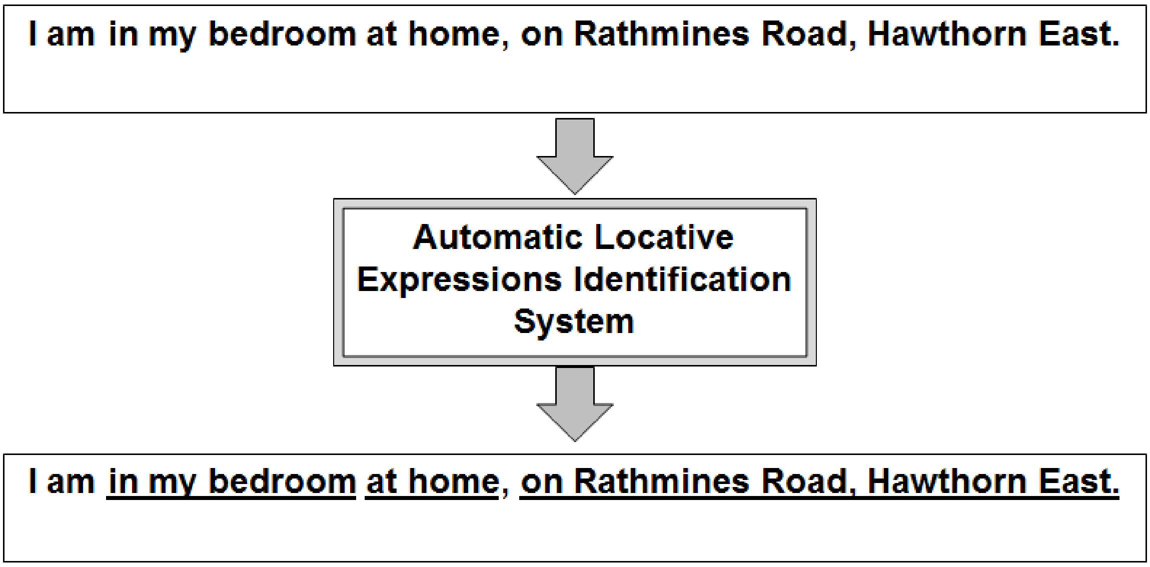
I am in my bedroom at home, on Rathmines Road, Hawthorn East.
Automatic Locative Expressions Identification System
I am in my bedroom at home, on Rathmines Road, Hawthorn East.

Figure 1.1: Diagram of the Input/Output of the System

Additionally, the ability to automatically identify locative expressions from informal text can be used to improve gazetteers. Moreover, it is also essential to research projects in the related field as it provides a means to automatically generate useful information, which, if done manually, is way too costly.

Ultimately, it enables us to gain a deeper understanding of how people describe locations using locative expressions in an informal way.

1.2 Research Question

In this research, we attempt to discover what aspects aid in the task of identifying locative expressions from informal text. Specifically, we try to investigate what the most discriminative feature setup is.

1.3 Contribution

In this research, we develop a system that is able to automatically identify locative expressions within informal text and discover insights of what aspects are helpful in the task of identifying locative expressions. Specifically, the mutual exclusiveness of the two gazetteers involved, *GeoNames* and *VICNAMES* (Section 3.3.1), is revealed in Section 5.2.1. Moreover, we uncover that *VICNAMES* performs better than *GeoNames*. Further, we identify features that have negative impacts on the model.

Additionally, the state-of-the-art performance in the identification task is achieved by our system, an 6.63 increase in $F_{\beta=1}$.

1.4 Structure of the Thesis

The thesis is structured as follows:

Chapter 2 presents an overview of the background knowledge of this research. We first explain geospatial language and geosparsing. Next we move on to the topic of natural language processing and machine learning both of which are involved in this research. Lastly, the machine learning model adopted in this research is studied.

Chapter 3 presents the resources involved in this research. The corpus from where the machine learning model learns is introduced. Next, an introduction of an implementation of the machine learning model used in this research is presented. Thirdly, two types of external resources, together with how they are used in this research, are explained. Lastly, in order to measure the performance of our machine learning model, the benchmark tools are described.

Chapter 4 presents the features we feed to our model. Two different feature setups are introduced with one utilising features automatically extracted from the corpus and the other making use of manually annotated information. Moreover, we explain the methodology adopted to evaluate our model. Lastly, three baseline systems are introduced, upon which we evaluate our model.

Chapter 5 presents the performance of the three baseline systems and our model. In order to understand the effectiveness of each feature explained in Chapter 4, a comparison based on feature ablation is provided. Further, we analyse locative expressions that are not identified by our model and investigate the reasons for such errors.

Chapter 6 summarises the research by providing an overview of what we have attempted and what we have achieved, and describes ideas for future work (Section 6.3).

Chapter 2

Background

In this chapter, we present background knowledge of this research. Specifically, the concept of geospatial language and geoparsing is explained in Section 2.1. Next, a definition of locative expression, along with the approach to interpreting locative expressions, is provided in Section 2.2. Thirdly, natural language processing technologies involved in this research are introduced in Section 2.3. Lastly, the introduction of the machine learning model adopted in this research is offered in Section 2.4.

2.1 Geospatial Language and Geoparsing

The definition of geospatial language is not limited to coordinate-based information, it can also be extended to natural language that contains geospatial information, such as spatial locations, orientation, movement and paths (Blaylock *et al.*, 2009). Geoparsing is the task of of locating geospatial language within natural language (Tytyk).

The importance of the ability to interpret spatial language is summarised by Zhang *et al.* (2010) into three points. First, given the volume of the data generated by human users, tools that can automatically analyze them are required in order to deal with what Miller (2010) called the "data avalanche". Next, the ability to automatically understand large amounts of text on the World-Wide-Web enables us to collect data and perform further analysis on it. Lastly, spoken language can be transcripted into written form since it is the most common way for humans to exchange thoughts and information.

2.2 Definition of Locative Expression

A locative expression is a human-generated expression used to describe geospatial location(s) expressed in geospatial language. Specifically, a locative expression may involve prepositions and geospatial place references (Herskovits, 1985).

An example is shown in Example (2.1). One locative expressions can be identified as underlined, consisting of four place references (*my bedroom*, *home*, *Rathmines Road* and *Hawthorn East*).

(2.1) I am in my bedroom at home, on Rathmines Road, Hawthorn East.

However, given the time and resources at our disposal, interpreting locative expressions as defined above was considered infeasible. Therefore, compromises on the interpretation of the definition of locative expressions had to be made. Specifically, place references framed by manual annotations (Section 3.1.3) are used to expand to locative expressions. The preposition preceding a place reference is counted as part of the very locative expression that contains the manual annotation. Further, neighbouring place references linked by either connective words such as *of*s and *and*s or punctuations such as commas are grouped into the same locative expressions.

Applied the interpretation described above, Example (2.1) is then translated into a place descritpion with three instread of one place descriptions as underlined in Example (2.2). The first and second locative expressions, *in my bedroom* and *at home*, both consist of prepositions (*in*, *at*) and geospatial noun phrases (*my bedroom*, *home*). The third one is formed by a preposition (*on*) and two geospatial named entities (*Rathmines Road*, *Hawthorn East*) connected by a comma. Even though compromised, the resulting three locative expressions in Example (2.2) approximate the one locative expressions in Example (2.1).

(2.2) I am in my bedroom at home, on Rathmines Road, Hawthorn East.

The process of breaking a place description into several pieces of locative expressions reflects how human interpret a place description. Geographic information systems, which process, store, manipulate, analyse and present geospatial information, can be used to process geospatial information that follows certain formats. When it comes to informal place descriptions, which may be generated by human and do not comply with any pre-defined format, the performance of even state-of-the-art geographic information systems suffer (Wu and Winter, 2011).

2.3 Natural Language Processing

In order to extract information from place descriptions, we adopt natural language processing methods to analyse them. Natural language processing methods, such as part-of-speech tagging (Section 2.3.1) and shallow parsing (Section 2.3.2), are widely used in many natural language processing applications (Munoz *et al.*, 2000; Jurafsky *et al.*, 1999). In this section, we introduce natural language processing methods involved in this research.

2.3.1 Part-of-speech Tagging

Part-of-speech tagging is the process of identifying words as nouns, verbs, adjectives, adverbs and etc. according to their particular part of speech (Jurafsky *et al.*, 1999). A complete list of types of POS tags is presented in Appendix A. Having place descriptions POS tagged is not only helpful to the shallow parsing, but to the task of identifying locative expressions as well since most locative expressions start with prepositions (e.g., *at*, *on*, *in*).

An example of a part-of-speech tagged place description is shown in Example (2.3). Each word in the place description is followed by a underscore and its part-of-speech tag.

(2.3) I_PRP am_VBP in_IN my_PRP$ bedroom_NN at_IN home_NN ,_, on_IN
Rathmines_NNP Road_NNP ,_, Hawthorn_NNP East_NNP ._.

2.3.2 Shallow Parsing

Structurally analysing place descriptions helps in terms of extracting constituents relevant to geospatial information. *Shallow parsing*, also known as chunking, is the process of partially analysing the syntactic structures and identifying the constituents of a sentence but not their internal structure (Abney, 1992; Munoz *et al.*, 2000). It is widely used in many language processing tasks (Munoz *et al.*, 2000). Sentences are dissembled into several non-overlapping chunks with each chunk assigned a type from the types shown in Appendix B. With the place descriptions shallow parsed, we hypothesise that the model is able to figure out the potential connection between the chunks and locative expressions.

An example of a shallow parsed place description is shown in Example (2.4). Each chunk is surrounded by a pair of [and].

(2.4) [NP I] [VP am] [PP in] [NP my bedroom] [PP at] [NP home] , [PP on]
[NP Rathmines Road] , [NP Hawthorn East] .

2.3.3 Named Entity Recognition

Named entities are phrases that represents names of persons, locations and organisations, expressions of times, quantities, amount of money and percentages (Tjong Kim Sang and De Meulder, 2003). Given the concept of named entity, named entity recognition is the process of identifying such entities from unstructured data.

Named entity recognition can be used to locate formal place references within natural language. However, the primary concern of this research is to identify not only formal place reference but informal ones as well. Therefore, we assume that the result would be unbalanced with high precision and low recall.

In this research, named entity recognition is used as part of the evaluation methodology.

2.4 Machine Learning

The primary goal of this research is to develop a system that is able to automatically identify locative expressions from informal text. The identification task can be realised based on machine learning methods. In this section, we introduce the machine learning methodology employed in this research.

2.4.1 Conditional Random Fields

Conditional Random Fields (*CRFs*) are widely used for sequential labelling tasks (Kudo *et al.*, 2004; Finkel *et al.*, 2008; Sarawagi and Cohen, 2004). In natural language processing tasks, the prediction of a label of a word relies not only on the text of a word but contextual information as well. In this research, the word itself, together with neighbouring words, plays an essential role in the task of identifying locative expressions from informal text. Since *CRFs* take context into account and have been proven to perform well in such tasks, they are used to predict the label of a single word with regard to contextual information.

In order to understand *CRFs*, three primary concepts are explained: what a feature function is, how the weight for each feature function is determined, how the probability of a sequence of labels given a sequence of words (a sentence) is calculated.

Feature Functions

A feature function takes the form shown in Equation 2.5 where s is the observation sequence (a sentence), l is a particular label sequence and i is the position of a word in the observation sequence s. Hence, l_i is the label of the ith word (current word) in the observation sequence s, and l_{i-1} is the label of the $(i-1)$th word (previous word).

$$f(s, i, l_i, l_{i-1}) \tag{2.5}$$

The output of a feature function is a real-valued number which is usually either 0 or 1.

Learning Weights

In order to learn the optimal weight for a particular feature function, Equation 2.6 is repeated until the updates on λ_j reach convergence (Sutton and McCallum, 2010).

$$\lambda'_j = \lambda_j + \alpha[\sum_{i=1}^{n} f_j(s, i, l_i, l_{j-1}) - \sum_{l'} p(l'|s) \sum_{i=1}^{n} f_j(s, i, l'_i, l'_{i-1})] \tag{2.6}$$

In Equation 2.6, λ'_j is the next weight for feature function $f_j(s, i, l_i, l_{i-1})$ while λ_j is the current weight. α is the learning rate which can be adjusted.

Probabilities

Using a set of feature functions, the score of a label sequence l given a particular observation sequence s, can be calculated as shown in Equation 2.7.

$$score(l|s) = \sum_{j=1}^{m} \sum_{i=1}^{n} \lambda_j f_j(s, i, l_i, l_{i-1}) \tag{2.7}$$

For each feature function f_j, a weight value λ_j is assigned. A large and positive λ_j suggests that the feature defined by function f_j has strong indications of the current word's label being l_i.

The probability of the label sequence l being the correct label sequence of the observation sequence s is calculated as shown in Equation 2.8, where the l' is all possible label sequences.

$$p(l|s) = \frac{\exp(score(l|s))}{\sum_{l'} \exp(score(l'|s))} \tag{2.8}$$

In order to determine the label sequence, we use *maximum a posteriori estimation* (*MAP estimation*) as shown in Equation 2.9. For every obervation sequence s, the *MAP* label sequence $\hat{l}_{MAP}(s)$ is assigned.

$$\hat{l}_{MAP}(s) = \arg\max_{x} p(l|s) \tag{2.9}$$

2.5 Chapter Summary

In this chapter, we explain critical background knowledge involved in this research.

In Section 2.1, the concept of geospatial language is introduced, which is the language that conatines geospatial information. Therefore, geoparsing is the process of locating geospatial language within natural language. Further, we discuss the important of the ability to interpret spatial language. Three key points summarised by Zhang *et al.* (2010) are presented.

In Section 2.2, the definition of locative expression is provided. Expressions used to describe geospatial location(s) are locative expressions. However, due to time and resources constraints, a compromised version of how we translate locative expressions using manual annotations (Section 3.1.3) is described.

In Section 2.3, natural language processing technologies relevant to this research are introduced. Part-of-speech tagging and shallow parsing, together with the reasons why they are adopted, are presented. Next, the concept of named entity recognition is explained as it is employed as port of the evaluation methodology.

In Section 2.4, we study the machine learning model adopted in this research. Specifically, we discuss how to use the model in our particular task with respect to the three primary aspects of the model: function feature, weight for each feature function and the probability of a sequence of labels given a sequence of words. Lastly, we present how to determine the label sequence for a given word sequence using *maximun a posteriori estimation.*

Chapter 3

Resources

In this chapter, we introduce resources used in this research. First, the corpus is introduced in Section 3.1. Specifically, not only do we introduce the dataset and how we preprocess the data, but we also explain what manual annotations are and how manual annotations are used to automatically re-annotate locative expressions. Next, we move on to the machine learning application. Thirdly, two types of external resources, gazetteers and dictionaries, are introduced in Section 3.3. Lastly, the benchmark tools involved in this research are introduced in Section 3.4.

3.1 Corpus

In this section, the corpus involved in this research is introduced (Section 3.1.1) followed by the introduction of the preprocessing of the corpus (Section 3.1.2) and a section dedicated to manual annotations (Section 3.1.3). The mechanism of automatic re-annotation of locative expressions is described in Section 3.1.4.

3.1.1 Tell Us Where Dataset

The *Tell Us Where* dataset was collected from a location-based mobile game. The locations of participants needed to be verified. Once confirmed their locations, participants were asked to answer the question "Tell us where you are?" and submit natural language descriptions about their locations through a web interface via their mobile phones. If not correctly located, participants could re-locate themselves. Therefore, the data is likely to be rich in locative expressions, which makes it an appropriate dataset for this research.

The collected data is primarily used to support academic projects aimed at discovering how people describe locations in Victoria, Australia, which may ultimately enable the development of better web searching, mapping and navigation systems, and even emergency services.

In this research, the data was collected as part of the *Tell Us Where* project. Ultimately, 1,858 place descriptions were collected by the game and will be used as the original corpus of this research.

An example of the raw data collected from *Tell Us Where* is presented in Example (3.1).

(3.1) optus oval watching the footy

In this research, the corpus used for the model to learn from is the preprocessed version of the raw data (Section 3.1.2) combined with manual annotations (Section 3.1.3), such as granularity levels and toponym ambiguities of place references within Victoria, Australia.

3.1.2 Data Preprocessing

Previous to being fed to the machine learning model, the raw data was preprocessed for the purposes of *part-of-speech tagging* (*POS tagging*, Section 2.3.1) and *shallow parsing* (*chunking*, Section 2.3.2). *OpenNLP*[1] was used for this purpose. An example of the outcome of Example (3.1) from *OpenNLP* is presented in Example (3.2).

(3.2) [NP optus_NN oval_NN] [VP watching_VBG] [NP the_DT footy_NN]

As can be seen, a place description can be divided into several chunks. Each chunk starts with the type of the chunk (chunk tag, e.g., *NP* (noun phrase), *PP* (prepositional phrase) and etc.) and consists of one or more word(s). Each word is followed by a underscore and its part-of-speech tag (POS tag, e.g., *IN* (conjunction, subordinating or preposition), *NNP* (noun, proper singular) and etc.). In some cases, a chunk does not have a chunk tag (e.g., *and_CC* (*CC* = conjunction, coordinating)). Such chunks are recognised as chunks that contain only one word and have no chunk tag.

3.1.3 Manual Annotation

The annotations were marked manually. Each place reference was annotated with its granularity level and identifiability, both of which were marked with the assist of external gazetteers, namely OpenStreetMap[2] and Google Maps.[3]

Each annotation clearly defines the boundary of a place reference. Manual annotations are vital to this research as they provide a means to locate place references

[1] `http://opennlp.apache.org/index.html`

[2] `http://www.openstreetmap.org/`

[3] `http://maps.google.com/`

Attribute	Description	Example
Start Position	The character offset of the start of a place reference in the chunked corpus.	*67288*
End Position	The character offset of the start of a place reference in the chunked corpus.	*67310*
Identifiability	The uniqueness of a place reference within Victoria, Australia. Possible values for this attribute are shown in Table 3.2.	*yes_unamb*
Granularity Level	The zoom (granularity) level of a place reference. Possible values for this attribute are shown in Table 3.3.	*1*
Normalisation Flag	A flag of whether the place reference is a vernacular/misspelt name of the canonical name/spelling.	*True*, *False*
Canonical Name/Spelling	The canonical name spelling of the place reference.	*Princes Par*

Table 3.1: Detail Information of Annotation

which can later be used to automatically re-annotate locative expressions according to the definition of locative expression presented in Section 2.2.

Several attributes are contained in an annotaion: start position, end position, identifiability, granularity level, normalisation flag and canoncial name/spelling.

The start position and end position are the character offsets of the start and the end of a place reference in the chunked corpus respectively.

Identifiability is the uniqueness of a place reference within Victoria, Australia. Granularity level is the zoom (granularity) level of a place reference. A normalisation flag represents whether a place reference is a vernacular/misspelt name of the canonical name/spelling. Canonical name/spelling stands for the canonical name/spelling of the place reference. Attributes contained in an annotation is presented in Table 3.1.

Three different values can be assigned to the identifiability of an annotation as shown in Table 3.2.

The value of granularity level ranges from 1 to 7 with each value representing a specifically defined level of geospatial granularity (Table 3.3).

With the help of manual annotations, 3,279 place references were extracted. However, 218 place references were marked irrelevant due to the fact that they either do not contain any geospatial information (e.g., *Economics class*) or are located outside

Identifiability	Description	Example
yes_unamb	identifiable non-ambiguous	*Carlton*, *Parkville*
yes_amb	identifiable ambiguous	*Swanston Street*, *Grattan St*
no	non-identifiable	*home*, *the station*

Table 3.2: Possible Values of Identifiability

Granularity Level	Description	Example
1	Furniture	*my bed*, *windows*
2	Room	*back porch*, *my bedroom*
3	Building	*the church*, *Swan St Optometrist*
4	Street	*Bell St*, *Tobruk Avenue*
5	District	*Templestowe*, *Parkville*
6	City	*Melbourne*, *Mornington*
7	Country	*australia*, *Victoria*

Table 3.3: Possibile Values of Granularity Level

of Victoria, Australia (e.g., *in wagga wagga*). Therefore, only 3,061 place references were actually valid and can be used as seeds to be expanded to locative expressions.

3.1.4 Automatic Re-annotation of Locative Expressions

Due to the fact that a locative expression consists of at least one place reference, the process of identifying locative expressions can be interpreted as the task of expanding place references to locative expressions and concatenating multiple place references to one locative expression.

Based on the definition of locative expression (Section 2.2), a set of rules can be derived to identify locative expressions using place references:

1. A locative expression contains at least one place reference.
2. A prepositional phrase is considered a part of a locative expression if it precedes a place reference.
3. The preposition *of*[4] and conjunction are considered to be semantic connectors that concatenate two surrounding place references, thereby constituting a larger locative expression.

[4] *of*s that are essentially tagged as particles (*PRT*) are excluded.

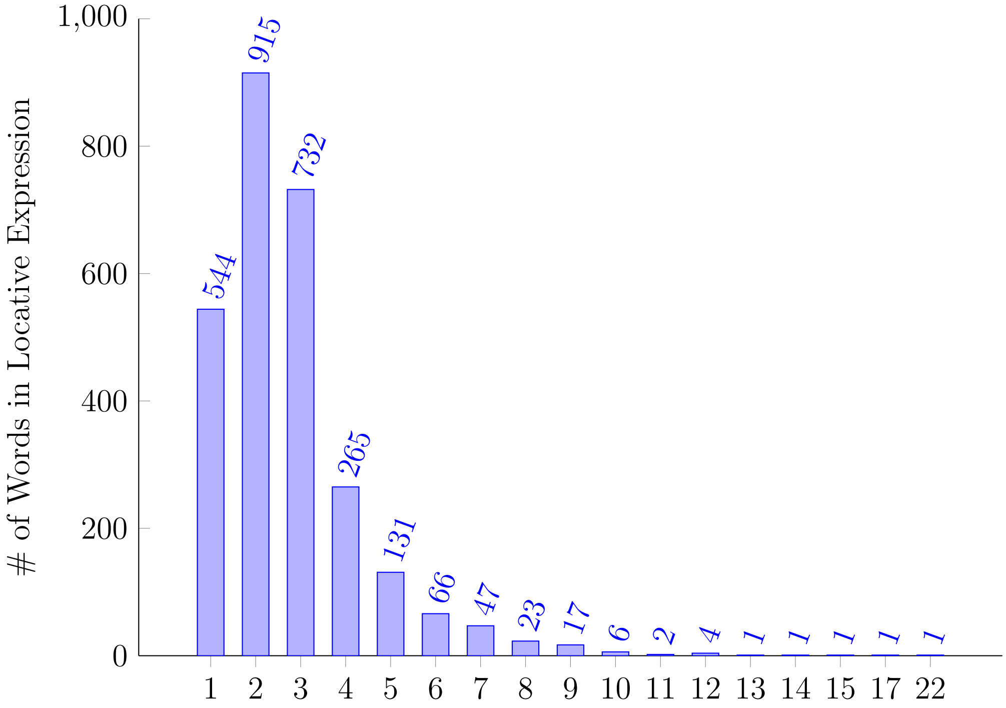

Figure 3.1: Distribution of Word Count of Locative Expressions

4. Punctuation, namely commas and the possessive apostrophe, are considered to be semantic connectors that concatenate two place references, thereby constituting a larger locative expression.

(3.3) I am in my bedroom at home, on Rathmines Road, Hawthorn East.

(3.4) I am in my bedroom at home, on Rathmines Road, Hawthorn East.

As underlined in Example (3.3), four place references are identified. The output of the automatic re-annotation task is shown in Example (3.4). Three locative expressions are re-annotated as highlighted. The third one, *on Rathmines Road, Hawthorn East* contains two place references, *Rathmines Road* and *Hawthorn East*, connected by a comma, all within a prepositional phrase headed by *on*.

Ultimately, 2,757 locative expressions were identified. The number of words contained in a locative expression ranges from 1 to 22 (mean: 2.74, standard deviation: 0.18). The distribution of word count of locative expressions is displayed in Figure 3.1.

3.2 Machine Learning Application

In Section 2.4.1, the machine learning model used in this research is explained. In this section, we introduce the the specific implementation of *CRF* we use.

3.2.1 CRF++

CRF++[5] is an open-source, highly-customisable implementation of the CRF model written in C++. It can be applied to a wide variety of NLP tasks thanks to its generic design and the use of feature templates. Both training and testing functions are provided and can perform their designated tasks respectively with optimised memory usage and minimal time consumption. Considering the merits mentioned above, we adopt CRF++ as our CRF toolkit.

Training Data

An example of the training data of CRF++ is presented in Table 3.4. Each line of the input file represents not only the word itself but also its features. Sentences are separated by empty lines. In this example, apart from the current word, one additional feature (POS tag) is listed as the second column and the last column is the correct label of the word. More features can be inserted into the feature table as long as the last column remains the correct label of the current word. The correct label column is IOB encoded with *B-NP*, *I-NP* and *O* representing the beginning of a locative expression, the inside of a locative expression and a word being outside of a locative expression, respectively.

Testing Data

Based on the features of a word in the testing data and the knowledge obtained from the training data, *CRF++* predicts the sequence of labels of words and appends it as the last column.

Feature Template

A feature is represented as a column in the training and testing data of *CRF++*. To make a feature visible to *CRF++*, however, a set of feature templates is required. Feature templates are defined in a similar fashion to two-dimensional coordinates (x, y) where the x coordinate is the relative position to the current word and the y coordinate corresponds to the absolute position of a column which represents a feature.

Eventually, *CRF++* generates $L \times N$ features where L is the number of output classes (in this case $L = 3$ (*B-NP*, *I-NP*, *O*)) and N is the number of distinct features.

[5] `http://crfpp.googlecode.com/svn/trunk/doc/index.html`

Word	POS Tag	Label
Off	IN	B-NP
Rathdowne	NNP	I-NP
St	NNP	I-NP
,	,	O
behind	IN	B-NP
the	DT	I-NP
Kent	NNP	I-NP
Hotel	NNP	I-NP
Parked	VBN	O
on	IN	B-NP
road	NN	I-NP
outside	IN	B-NP
primary	JJ	I-NP
school	NN	I-NP

Table 3.4: An Example Training Data of *CRF++*

3.3 External Resources

Two types of external resources, gazetteers and dictionaries, are introduced as they provide data sources for feature extraction.

3.3.1 Gazetteers

A gazetteer is a geospatial dictionary of geographic names Hill (2000). It provides mappings between actual place references and information about places. In this research, we use two gazetteers: *GeoNames*[6] and *VICNAMES*[7].

As stated by Hill (2000), a gazetteer consists of three core components as shown in Table 3.5. In this research, we make use of categories of place references.

GeoNames

GeoNames is a geographical database with eight million place references across all countries. Its data can be accessed through various web services. Even though users are able to edit and improve the *GeoNames* database through a Wiki interface,

[6] `http://www.geonames.org/`

[7] `http://services.land.vic.gov.au/vicnames/`

Component	Description	Example
Name	The name of a place, including aliases.	*Parkville*
Location	The coordinates of a place.	*Lat: -37.78333*, *Long: 144.95*
Type	Chosen from a type scheme of categories for places/features	The category of a place: *A* (country, state, region,)

Table 3.5: Core Components of Gazetteer

most data is provided by official public sources. Having said that, however, it is not guaranteed each source is of the same quality.

112,858 place references across Australia are stored in the *GeoNames* database. Apart from place references, other features are also included in the database, such as latitude, longitude, elevation, population, administrative subdivision etc. In some cases, people may use various aliases to refer to the same place. To cope with such cases, alternative names are included as one feature of every entry.

In this research, we adopt *GeoNames* as an external gazetteer and extract useful information such as feature class and feature code.[8] The general geospatial category of a place reference is represented by its feature class property whereas its detailed geospatial category information is stored as the attribute feature code.

VICNAMES

VICNAMES consists of more than 200,000 places located in Victoria, Australia. A wide variety of places are included in the database ranging from landscape features (e.g., *mountains* and *rivers*) to bounded localities (e.g., *suburbs*, *towns*, *cities* and *regions*). Physical infrastructure such as roads, reserves and schools are stored as well.

Entries included in the *VICNAMES* database are created and maintained by the state government of Victoria. A total of 43,863 entries are included in the VICNAMES database. Compare with GeoNames the data of which is collected from a range of official public sources, the quality of the *VICNAMES* database is supposed to be better than *GeoNames* thanks to governmental maintenance and a localised coverage.

Since no web service is provided, data stored in the *VICNAMES* database can only be accessed by downloading and parsing it locally.

In this research, *VICNAMES* is used in a similar fashion to *GeoNames* except for feature class since it is not provided in the *VICNAMES* database.

[8] `http://www.geonames.org/export/codes.html`

3.3.2 Dictionaries

Of all the nouns and verbs, some, such as the underlined words shown in Example (3.5) and Example (3.6), have strong indications of representing places or are frequently used in conjunction with locative expressions. Hence, for the purpose of identifying such nouns and verbs, we employ external dictionaries.

(3.5) 103 hephman street

(3.6) visiting rocky point for the day

Additionally, it is not uncommon that the quality of status messages varies and non-standard words, such as typos, ad hoc abberviations, etc., exist (Han *et al.*, 2013). Therefore, we adopt an external dictionary for the purpose of lexical normalisation.

WordNet

WordNet[9] is a lexical database for English with four types of words (nouns, verbs, adjectives and adverbs) grouped hierarchically into a network structure according to their conceptual relations. Tow types of relationships, hypernym relation and hyponym relation, exist among words. For a word A, a word B is a hypernym of A if B is a supertype of A. Correspondingly, a word C is considered a hyponym of A if C is a subtype of A (Snow *et al.*, 2004). For instance, *dog* and *cat* are both hyponyms of the word *animal* and *animal* is a hypernym of both *dog* and *cat.* Hence, for any word, "is-a" relationships exist between the word itself and all its hyponyms. Such inheritance relationships enable us to obtain a set of words that are geospatially related to locative expressions.

In this research, *Natural Language Toolkit*[10] (*nltk*) is used to access *WordNet.*

Lexical Normalisation Dictionary

To deal with non-standard words, an external lexical normalisation dictionary[11] is employed. Essentially, the dictionary provides mappings between typos, ad hoc abberviations, etc. and canonical spellings.

As shown in Example (3.7), the word *appartment* is misspelt and the canonical spelling is *apartment.* The dictionary enables us to transform mispelt words back to their canonical forms. (Section 4.1.5)

(3.7) In my $\overbrace{\text{appartment}}^{\text{apartment}}$ overlooking the sports oval on liardet street.

[9] http://wordnet.princeton.edu/

[10] http://nltk.org/

[11] http://ww2.cs.mu.oz.au/~tim/etc/emnlp2012-lexnorm.tgz

3.4 Benchmark Tools

In this section, we introduce two benchmark tools, *StanfordNER* and *Unlock Text*, upon which we build two baseline systems.

3.4.1 StanfordNER

The *Stanford Named Recognizer*[12] (*StanfordNER*), developed by the Stanford Natural Language Processing Group at Stanford University, is a Java implementation of a named entity recogniser based on the conditional random field sequence model. It was developed to tackle the problem of named entity recognition (Section 2.3.3). Equipped with well-engineered features and the training data which can be downloaded from the the website[13], the application takes text as input and identifies formal named entities in the input text. In this research, we use *StanfordNER* as one of our baseline systems (Section 4.4.1).

Apart from the well-engineered features and the provided training data, a general implementation of linear chain Conditional Random Field sequence models is provided by StanfordNER as well, which allows us to retrain the model using our particular training data and therefore is employed as one of the baseline systems in this research (Section 4.4.3).

3.4.2 Unlock Text

Unlock Text[14], developed by the Language Technology group at the School of Informatics at the University of Edinburgh, is a geoparser based on *GeoNames*. A geoparser identifies place references in natural language using gazetteers. It is able to identify possible place references from informal text. Thus, we employ *Unlock Text* as the one of the baseline systems (Section 4.4.2).

3.5 Chapter Summary

In this chapter, we present resources involved in this research.

In Section 3.1, the corpus is introduced. First, we present the source of the corpus, which is the *Tell Us Where* game. Next, we introduce the preprocessing scheme (part-of-speech tagging and shallow parsing) for the corpus. Lastly, we study manual annotations and discuss how we use them as the gold standard data to re-annotate locative expressions.

[12] `http://nlp.stanford.edu/software/CRF-NER.shtml`

[13] `http://www-nlp.stanford.edu/software/CRF-NER.shtml#Download`

[14] `http://unlock.edina.ac.uk/texts/introduction`

In Section 3.2, the machine learning application, *CRF++* is studied. Further, we explain key concepts, such as training data, testing data and feature template.

In Section 3.3, we present two types of external resources, gazetteers and dictionaries. For both external gazetteers, *GeoNames* and *VICNAMES*, the mapping between a place reference and the category of that very place reference is used. Further, we adopt two external dictionaries, *WordNet* and *Lexical Normalisation Dictionary. WordNet* is used to identifying locative indicators and motion verbs whereas *Lexical Normalisation Dictionary* is employed to recover non-standard words to their canonical form.

In Section 3.4, we present two benchmark tools, *StanfordNER* and *Unlock Text*, both of which are used as baseline systems in this research.

Chapter 4

Methodology

In this chapter, we introduce the classifier setup. Two sets of features are defined for both the automatic identification setup (Section 4.1) and the gold standard setup (Section 4.2), which makes use of the manual annotations. Moreover, for each feature, a set of templates is defined. Features in both setups are explained in this chapter.

4.1 Automatic Identification Setup

In this section, we introduce features that can be extracted automatically from the corpus without the use of manual annotations.

4.1.1 Word

The text of a word is used as a feature as it provides the most basic information of a word. If the model has seen a word A in the training corpus before, then the probability of the label of a word B that has the same text being the same the label of A is relatively higher. As underlined in Example (4.1) and Example (4.2), *at home* in both examples are locative expressions with exactly the same words.

(4.1) at home in bed

(4.2) I'm at home in Kensington

Apart from the text of the current word, additional information about neighbouring words is taken into account as well to help the model make a more informed prediction of the label of the current word. We adopt the same examples (Example (4.1) and Example (4.2)). The last words in both examples, although different in text, are identified as parts of locative expressions as they share the same sequence of previous words *at home in.*

Word	POS Tag	Chunk Tag	Label
I	PRP	B-NP	O
am	VBP	B-VP	O
in	IN	B-PP	B-NP
my	PRP$	B-NP	I-NP
bedroom	NN	I-NP	I-NP
at	IN	B-PP	B-NP
home	NN	B-NP	I-NP << current word
,	,	O	O
on	IN	B-PP	B-NP
Rathmines	NNP	B-NP	I-NP
Road	NNP	I-NP	I-NP
,	,	O	I-NP
Hawthorn	NNP	B-NP	I-NP
East	NNP	I-NP	I-NP
.	.	O	O

Table 4.1: An Example of Word Feature

Template	Description
Windows of neighbouring words	The text of the nth word $(i-3 \leq n \leq i+3)$
Combinations of two immediate neighbouring words	The combination of the text of the nth and the $(n+1)$th word $(i-2 \leq n \leq i+1)$

Table 4.2: Template Setup for Word Feature

An example is presented in Table 4.1 with *home* as the current word. The interpretation of templates defined in Table 4.2 of Example (4.1) is presented in Table 4.3.

4.1.2 POS Tag

In addition to the text of a word, the POS tag of a word is also used as a feature as it provides information about the grammatical role the word plays in a sentence. Since the vocabulary in the training corpus is limited, chances are text of words in the testing corpus may not be contained in the training corpus. Lexical features do not generalise well. In such cases, matching POS tags of words provides more general

Template	Word
$i-3$	my
$i-2$	bedroom
$i-1$	at
i	home
$i+1$	,
$i+2$	on
$i+3$	Rathmines
$[i-2/i-1]$	bedroom/at
$[i-1/i]$	at/home
$[i/i+1]$	home/,
$[i+1/i+2]$	,/on

Table 4.3: Example of Mapping between Template and Words

information than matching text of words in the process of determining the labels of words. In Example (4.3) and Example (4.4), each word is followed by a underscore and its POS tag and the underlined phrases are locative expressions. Despite differences in actual words, *On the train* and *In the car* are both identified as locative expressions due to the fact that they share the same sequences of POS tags.

(4.3) On_IN the_DT train_NN at_IN bentleigh_NN

(4.4) In_IN the_DT car_NN on_IN the_DT corner_NN of_IN Arden_NNP St_NNP and_CC Dryburgh_NNP St_NNP

We define templates for this feature in Table 4.4 with the current word as the ith word in a sentence.

We adopt the same example as displayed in Table 4.1. Again, we assume *home* is the current word, therefore the POS tag of the current is *NN*.[1] Neighbouring POS tags that will be used to predict the label of *home* are listed in Table 4.5.

4.1.3 Chunk Tag

Similar to POS tags, chunk tags also provide grammatical information about the constituents (e.g., noun groups, verb groups, prepositional groups, etc.) of a sentence.

Most place references (2,584 out of 3,061, 84.4%) start at chunk boundaries, and therefore, chunks are good indications of boundaries of place references. To discriminate the beginning and the inside of a chunk, IOB tags are employed. That is, the

[1] noun, singular or mass

Template	Description
Windows of neighbouring POS tags	The POS tag of the nth word ($i-2 \leq n \leq i+2$)
Combinations of two immediate neighbouring POS tags	The combination of the POS tags of the nth and the $(n+1)$th word ($i-2 \leq n \leq i+1$)
Combinations of three immediate neighbouring POS tags	The combination of the POS tags of the nth and the $(n+1)$th and the $(n+2)$th word ($i-2 \leq n \leq i$)

Table 4.4: Template Setup for POS Tag Feature

Template	POS Tag
$i-2$	NN
$i-1$	IN
i	NN
$i+1$	,
$i+2$	IN
$[i-2/i-1]$	NN/IN
$[i-1/i]$	IN/NN
$[i/i+1]$	NN/,
$[i+1/i+2]$	,/IN
$[i-2/i-1/i]$	NN/IN/NN
$[i-1/i/i+1]$	IN/NN/,
$[i/i+1/i+2]$	NN/,/IN

Table 4.5: Example of Mapping between Template and POS Tags

beginning of a chunk is prefixed with the *B-* tag (e.g., *B-NP*) and rest of the words in the chunk are prefixed with the *I-* tag (e.g., *I-NP*).

We define templates for this feature in Table 4.6 and assume the current word is the ith word in a sentence.

4.1.4 Word Position

Even though 84.4% of the locative expressions start with the beginnings of chunks, it is unclear to the model where the beginning of the remaining 15.5% are. It is difficult for the model to identify the start of locative expressions without the help of additional features. To cope with such difficulty, the position of a token in a given

Template	Description
Windows of neighbouring chunk tags	The chunk tag of the nth word $(i-2 \leq n \leq i+2)$
Combinations of two immediate neighbouring chunk tags	The combination of the chunk tags of the nth and the $(n+1)$th word $(i-2 \leq n \leq i+1)$

Table 4.6: Template Setup for Chunk Tag Feature

Template	Description
Windows of neighbouring token positions	The token position of the nth word $(i-2 \leq n \leq i+2)$
Combinations of two immediate neighbouring token positions	The combination of the token positions of the nth word and the $(n+1)$th word $(i-1 \leq n \leq i)$

Table 4.7: Template Setup for Word Position Feature

chunk is likely to be instructive in determining the start of a locative expression.

$$\text{(4.5) [ADVP } \overbrace{\text{Approximate\_RB}}^{0}\ \overbrace{\text{halfway\_RB}}^{1} \text{] [PP } \overbrace{\text{between\_IN}}^{0} \text{] [NP}$$

$$\overbrace{\text{Lara\_NNP}}^{0}\ \overbrace{\text{and\_CC}}^{1}\ \overbrace{\text{Little\_NNP}}^{2}\ \overbrace{\text{River\_NNP}}^{3} \text{] [VP } \overbrace{\text{.\_.}}^{0} \text{]}$$

As displayed in Example (4.5), each word is marked with its position in the enclosing chunk starting from 0.

We define templates for this feature in Table 4.7 and assume the current word is the ith word in a sentence.

4.1.5 Text Normalisation

In the field of natural language processing, text normalisation is an important problem (Sproat *et al.*, 2001). In this research, we use two methods, lemmatisation and lexical normalisation, to address this problem.

Lemmatisation

Lemmatisation is the process of converting a word to its dictionary form. It has been proven beneficial to the processing of natural language (Toutanova and

Template	Description
Windows of neighbouring lemmatised words	The lemmatised word of the nth word ($i-2 \leq n \leq i+2$)
Combinations of two immediate neighbouring lemmatised words	The combination of the lemmatised words of the nth word and the $(n+1)$th word ($i-2 \leq n \leq i+1$)
Combinations of lemmatised words and POS tags	The combination of the lemmatised version of the nth word and the POS tag of the nth word ($i-2 \leq n \leq i+2$)

Table 4.8: Template Setup for Lemmatisation Feature

Cherry, 2009). A pair of examples is shown in Example (4.6) and Example (4.7). The underlined words, *walking* and *walked*, are lemmatised to *walk*. Thus, lemmatisation essentially removes differences in inflectional morphology and increases the chance of matching two words derived from the same lemma. In this research, we adopt the text of lemmatised words as a feature.

(4.6) walking down the street ⇒ walk down the street

(4.7) walked here from my house ⇒ walk here from my house

To lemmatise words, we utilise the package *WordNetLemmatizer* included in *nltk*, which is based on a *WordNet*[2] library.

We define templates for this feature in Table 4.8 and assume the current word is the ith word in a sentence.

Lexical Normalisation

As mentioned in Section 3.3.2, to transform non-standard words back to their canonical form, an external dictionary is employed for the purpose of lexical normalisation. As shown in Example (4.8) and Example (4.9), both underlined words are misspelt. By applying the dictionary, we are able to transform them back to the canonical form *avenue*. Thus, lexical normalisation helps in terms of reducing the number of misspelt words.

(4.8) on hollywood avenu ⇒ on hollywood avenue

(4.9) 23 espirte aven ⇒ 23 espirte avenue

[2] `http://wordnet.princeton.edu/`

Template	Description
Windows of neighbouring lexical normalised words	The lexical normalised word of the nth word $(i-2 \leq n \leq i+2)$
Combinations of two immediate neighbouring lexical normalised words	The combination of the lexical normalised words of the nth word and the $(n+1)$th word $(i-2 \leq n \leq i+1)$

Table 4.9: Template Setup for Lexical Normalisation Feature

To transform a word back to its canonical form, the external dictionary introduced in Section 3.3.2 is adopted. For words that exist in the dictionary, the canonical form extracted from the dictionary is used as the lexical normalisation feature of the word. For words that are not stored in the dictionary, it is assumed that they are in their correct forms, and therefore are used as the feature without any modification.

We define templates for this feature in Table 4.13 and assume the current word is the ith word in a sentence.

4.1.6 Chunk-Preceding Preposition

According to the definition of locative expression, a locative expression may consist of not only one or multiple place reference(s) but also prepositions. In fact, 1,103 out of 2,757 locative expressions start with a preposition, not to mention locative expressions linked by lexical connectors (e.g., *of*, *and*). As can be drawn from the statistics, prepositions play essential roles in determining the beginnings of locative expressions. Therefore, we employ prepositions as a feature to feed to the model.

As displayed in Example (4.10), the chunk *Lara and Little River* is preceded by the preposition *between* and is therefore assigned *between* as its chunk-preceding preposition.

If and only if the chunk is preceded by a *prepositional phrase* (*PP*) and the chunk itself is a *noun phrase* (*NP*) is every word in the chunk assigned the text of the preceding *PP*. An example of the interpretation of this feature using Example (4.10) is presented in Table 4.10.

(4.10) [ADVP Approximate_RB halfway_RB] [PP between_IN] [NP
$\overbrace{\text{Lara\_NNP and\_CC Little\_NNP River\_NNP}}^{\text{between}}$] [VP ._.]

We define templates for this feature in Table 4.11 and assume the current word is the ith word in a sentence.

Word	Preposition
Approximate	None
halfway	None
between	None
Lara	between
and	between
Little	between
River	between
.	None

Table 4.10: An Example of Chunk-Preceding Preposition Feature

Template	Description
Windows of neighbouring chunk-preceding propositions	The chunk-preceding preposition of the nth word $(i - 2 \leq n \leq i + 2)$
Combinations of two immediate neighbouring chunk-preceding prepositions	The combination of the chunk-preceding preposition of the nth word and the $(n+1)$th word $(i-2 \leq n \leq i+1)$

Table 4.11: Template Setup for Chunk-Preceding Prepositions Feature

4.1.7 Automatic Geospatial Feature Class

Intuitively, some prepositions tend to be used in conjunction with particular types of place references. The connection between prepositions and a place reference's geospatial category is likely to be informative in determining the boundaries of locative expressions. Therefore, it is sensible to mark place references with their corresponding geospatial categories.

As displayed in Example (4.11), *the_DT You_PRP Yangs_NNP* is identified as a place reference that has a feature class of *T (mountain, hill, rock, . . .)* by *GeoNames*.

(4.11) [ADVP Almost_RB] [PP at_IN] [NP the_DT foot_NN] [PP of_IN]
[NP $\overbrace{\text{the\_DT You\_PRP Yangs\_NNP}}^{\text{T (mountain, hill, rock, \dots)}}$] ._.

Since no information regarding the boundaries of place references is provided, we adopt Algorithm 1 to assign the geospatial feature class to chunks in a sentence.

With the help of external gazetteers (Section 3.3.1), place references can be assigned feature classes according to their geospatial categories. Feature classes are

Algorithm 1 Search Geospatial Feature Class

```
function SEARCH_FEATURE_CLASS($sentence, dictionary$)
    $i \leftarrow 0$
    $query \leftarrow sentence.chunks[i]$
    while $i < sentence.chunks.length$ do
        if $sentence.chunks[i]$ is not a $NP$ chunk then
            $i \leftarrow i + 1$
            continue
        end if
        $j \leftarrow i + 1$
        while $j < sentence.chunks.length$ do
            if $sentence.chunks[j]$ is a $NP$ chunk or($sentence.chunks[j+1]$ is a $NP$ chunk and($sentence.chunks[j]$ is a $PP$ chunk or$sentence.chunks[j] =$ “$and''$ or$sentence.chunks[j] =$ “$,''$)) then
                $query \leftarrow query + sentence.chunks[j]$
            else
                break
            end if
            $j \leftarrow j + 1$
        end while
        if $query$ in $dictionary$ then
            $fc \leftarrow dictionary.get(query)$
            $k \leftarrow i$
            while $k < j$ do
                $sentence.chunks[k] \leftarrow fc$
                $k \leftarrow k + 1$
            end while
            $i \leftarrow j$
        else
            $query \leftarrow sentence.chunks[i]$
            if $sentence.chunks[i]$ in $dictionary$ then
                $fc \leftarrow dictionary.get(query)$
                $sentence.chunks[i] \leftarrow fc$
            else
                for each $word$ in $chunks[i]$ do
                    if $word$ in $dictionary$ then
                        $fc \leftarrow dictionary.get(query)$
                        $word \leftarrow fc$
                    else
                        $word \leftarrow O$
                    end if
                end for
            end if
            $i \leftarrow i + 1$
        end if
    end while
end function
```

Word	Geospatial Feature Class
Almost	None
at	None
the	None
foot	None
of	None
the	T
You	T
Yangs	T
.	None

Table 4.12: An Example of Geospatial Feature Class Feature

Template	Description
Windows of neighbouring geospatial feature classes	The geospatial feature classes of the nth word ($i-2 \leq n \leq i+2$)
Combinations of two immediate neighbouring geospatial feature classes	The combination of the geospatial feature classes of the nth word and the $(n+1)$th word ($i-2 \leq n \leq i+1$)

Table 4.13: Template Setup for Geospatial Feature Class Feature

assigned if and only if place references that are found in gazetteers. An example of the interpretation of this feature using Example (4.11) is presented in Table 4.12.

In the case where *GeoNames* is used, 1,311 matches can be found whereas in *VICNAMES* the number of matches is 861.

We define templates for this feature in Table 4.13 and assume the current word is the ith word in a sentence.

4.1.8 POS Tag within Chunk

The POS tag within chunk feature aims at capturing common patterns of POS tags within chunks that constitute locative expressions. Inspired by observation, we derive two sub-features, the first POS tag feature and the most frequent POS tag feature.

First POS Tag

As mentioned in Section 4.1.3, chunks are important in the process of identifying locative expressions.

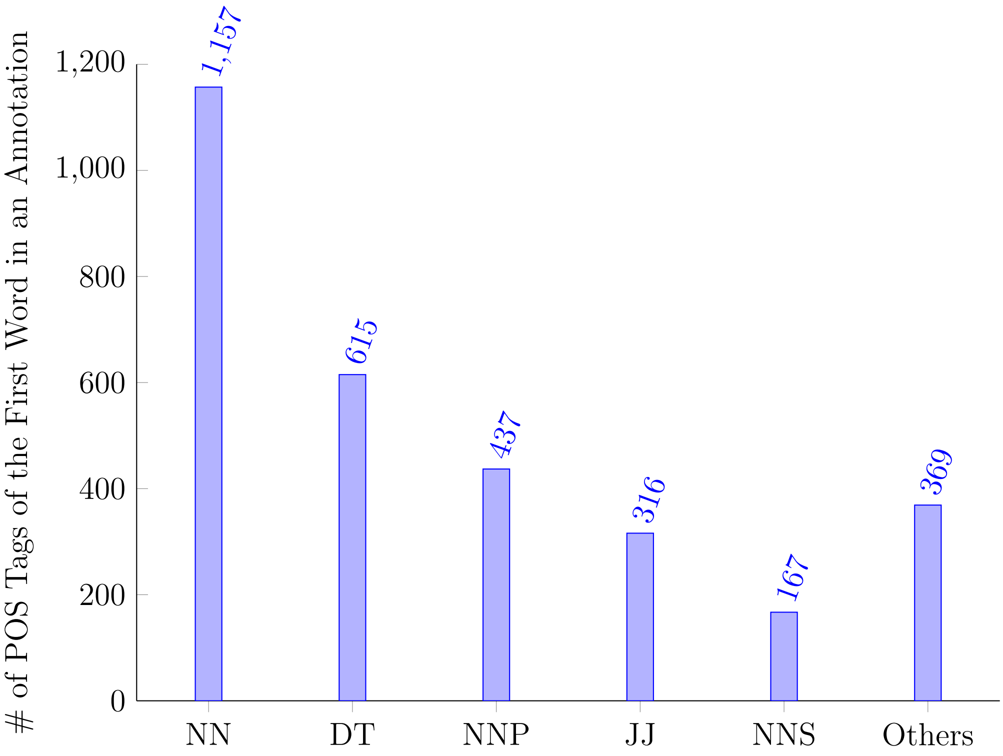

Figure 4.1: Distribution of the POS Tag of the First Word in an Annotation

It can be observed from Figure 4.1 that the majority of POS tags of the first word in a locative expression are: *NN*s, *DT*s, *NNP*s, *JJ*s and *NNS*s. These five types of POS tags account for over 80% (2,692/3,061) of the total number. Since locative expressions are expanded from place references and most place references are essentially comprised of one or more chunks, it is implied that POS tags of the first words in annotations are of importance to our task.

To interpret this as a feature, we simply assign each word in a chunk the POS tag of the first word in that chunk. The example shown in Example (4.12) is interpreted as presented in Table 4.14.

(4.12) [ADVP Almost_RB] [PP at_IN] [NP the_DT foot_NN] [PP of_IN] [NP the_DT You_PRP Yangs_NNP] ._.

We define templates for this feature in Table 4.15 and assume the current word is the ith word in a sentence.

Word	First POS Tag
Almost	RB
at	IN
the	DT
foot	DT
of	IN
the	DT
You	DT
Yangs	DT
.	.

Table 4.14: An Example of First POS Tag Feature

Template	Description
Windows of neighbouring first POS tags	The first POS tag of the chunk that encloses the nth word ($i-2 \leq n \leq i+2$)
Combinations of two immediate neighbouring first POS tags	The combination of the first POS tag of the chunk that encloses the nth and the first POS tag of the chunk that encloses the $(n+1)$th word ($i-1 \leq n \leq i$)
Combinations of words and first POS tags	The combination of the nth word and the first POS tag of the chunk that encloses the nth word ($i-1 \leq n \leq i+1$)

Table 4.15: Template Setup for First POS Tag Feature

Most Frequent POS Tag

It can be observed from Figure 4.2 that the most frequent POS tag in more than 50% of the manual annotations is *NN*. The sum of the top five POS tags consumes 95% of all manual annotations. As mentioned in the previous section, according to the connection between manual annotations and locative expressions, it is suggested that the most frequent POS tag in a chunk is indicative of whether a chunk is part of a locative expression.

To interpret this as a feature, we simply assign each word in a chunk the most frequent POS tag of that chunk. The example shown in Example (4.13) is interpreted as presented in Table 4.16.

(4.13) [PP Just_RB near_IN] [NP the_DT Theatre_NNP Royal_NNP] and_CC [NP the_DT supermarket_NN] ._.

We define templates for this feature in Table 4.17 and assume the current word is

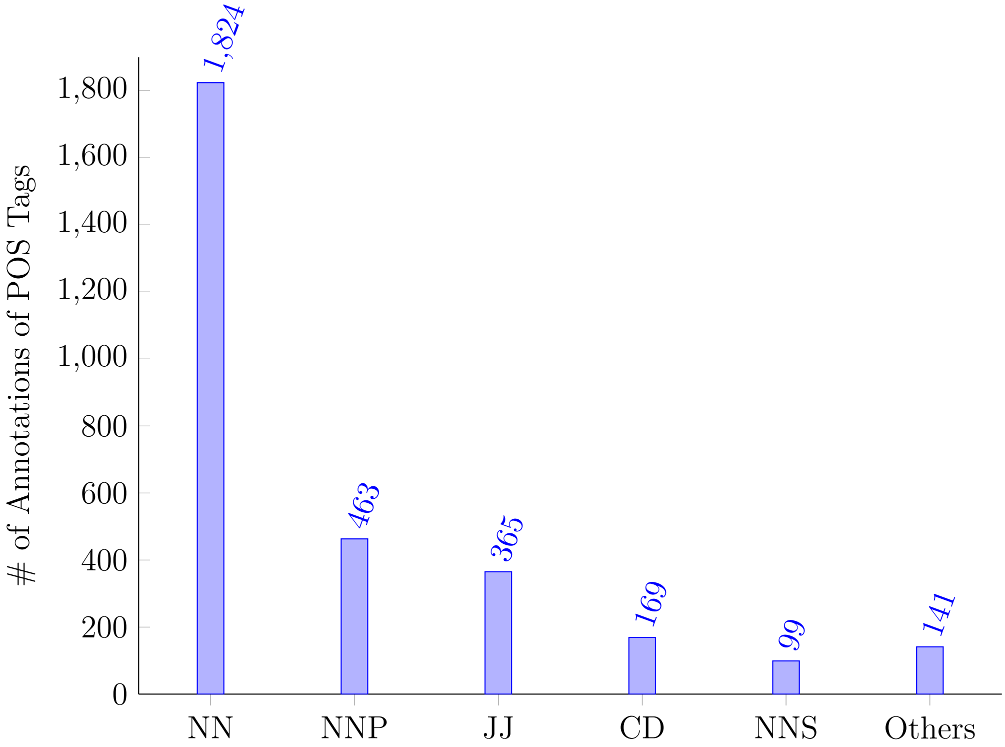

Figure 4.2: Distribution of the Most Frequent POS Tags in Manual Annotations

Word	Most Frequent POS Tag
Just	RB
near	RB
the	NNP
Theatre	NNP
Royal	NNP
and	CC
the	DT
supermarket	DT
.	.

Table 4.16: An Example of Most Frequent POS Tag Feature

the *i*th word in a sentence.

Template	Description
Windows of neighbouring most frequent POS tags	The most frequent POS tag of the chunk that encloses the nth word $(i - 2 \leq n \leq i + 2)$
Combinations of two immediate neighbouring most frequent POS tags	The combination of the most frequent POS tag of the chunk that encloses the nth and $(n+1)$th word $(i-1 \leq n \leq i)$
Combinations of the most frequent POS tags and the counts of the most frequent POS tags	The combination of the most frequent POS tag of the chunk that encloses the nth word and the number of appearances of that very POS tag in the chunk $(i-1 \leq n \leq i + 1)$
Combinations of the most frequent POS tags and the numbers of words contained in the chunk	The combination of the most frequent POS tag of the chunk that encloses the nth word and the number of words in the chunk $(i - 1 \leq n \leq i + 1)$

Table 4.17: Template Setup for First POS Tag Feature

4.1.9 Locative Indicator and Motion Verb

As mentioned in Section 3.3.2, some nouns (*locative indicators*) and verbs (*motion verbs*) are strongly indicative of places or are frequently used in conjunction with locative expressions. To identify such words, we adopt an external dictionary (*WordNet*).

The process of choosing such words, however, cannot be done automatically and, therefore, is unavoidably subjective. To select locative indicators and motion verbs, we take the following steps:

- Collect words that are locative indicators and motion verbs from the corpus.
- Derive a set of hyponyms for each collected word.
- Remove words whose most common senses are not relevant to locative indicators and motion verbs.

Once the sets of locative indicators and motion verbs are selected, we use them to match against words in the corpus. Ultimately, each word is assigned two binary features:

- A flag of whether the word is a locative indicator.
- A flag of whether the word is a motion verb.

Word	Locative Indicator	Motion Verb
walking	False	True
to	False	False
john	False	False
cain	False	False
memorial	False	False
park	True	False

Table 4.18: An Example of Locative Indicator and Motion Verb Features

Template	Description
Windows of neighbouring locative indicator flags	The locative indicator flag of the nth word ($i-2 \leq n \leq i+2$)
Combinations of two immediate neighbouring locative indicator flags	The combination of locative indicator flags of the nth and $(n+1)$th word ($i-2 \leq n \leq i+1$)

Table 4.19: Template Setup for Locative Indicator Feature

Template	Description
Windows of neighbouring motion verb flags	The motion verb flag of the nth word ($i-2 \leq n \leq i+2$)
Combinations of two immediate neighbouring motion verb flags	The combination of motion verb flags of the nth and $(n+1)$th word ($i-2 \leq n \leq i+1$)

Table 4.20: Template Setup for Motion Verb Feature

An example is shown in Example (4.14). The interpretation of Example (4.14) is presented in Table 4.18.

(4.14) walking to john cain memorial park

We define templates for the locative indicator feature in Table 4.19 and the motion verb feature in Table 4.19 and assume the current word is the ith word in a sentence.

4.1.10 Motion Verb / Lemmatisation

Motion verbs tend to be used in conjunction with particular prepositions. As can be seen from Example (4.15), the place reference *collins street* is preceded by the

Template	Description
Combinations of POS tags, motion verb flags and chunk-preceding prepositions	The combination of the POS tag, motion verb flag and chunk-preceding preposition of the nth word ($i-1 \leq n \leq i+1$)
Combinations of the POS tag and the motion verb flag of the current word and chunk-preceding preposition of other words	The combination of the POS tag and motion verb flag of the ith word and chunk-preceding preposition of the nth word ($i-3 \leq n \leq i+3$)

Table 4.21: Template Setup for Motion Verb / Lemmatisation Feature

motion verb *walking* and preposition *down*. The combination of *walking* and *down* has the clear implication of *down collins street* being a locative expression.

(4.15) $$\overbrace{\text{[VP walking\_VBG]}}^{\text{motion verb}}\ \underbrace{\overbrace{\text{[PP down\_IN]}}^{\text{preposition}}\ \overbrace{\text{[NP collins\_NNS street\_NN]}}^{\text{place reference}}}_{\text{locative expression}}$$

We define templates for this feature in Table 4.21 and assume the current word is the ith word in a sentence.

4.2 Gold Standard Setup

In order to maximise the performance of our model, we make use of manual annotations. We introduce one feature and its templates in this section.

4.2.1 Manual Annotation

Four types of information about one place reference can be extracted from manual annotations: identifiability, granularity level, normalisation flag and canonical name (Section 3.1.3). From these four types of information, it is possible to retrieve the boundaries of place references since every place reference has an corresponding manual annotation.

As these four types of information are similar and can be grouped by place references, we use identifiability to illustrate how each type of information in the group is interpreted. The rest three types are omitted due to the similarity.

Word	Identifiability
I	None
am	None
in	None
my	B-no
bedroom	I-no
at	None
home	B-no
,	None
on	None
Rathmines	B-yes_unamb
Road	I-yes_unamb
,	None
Hawthorn	B-yes_unamb
East	I-yes_unamb
.	None

Table 4.22: An Example of Identifiability Feature

To interpret identifiability as a feature, we assign the identifiability of a place reference to each word within that place reference. For words that are not contained in any place reference, *None* is assigned. Consequently, this feature not only provides information about the identifiability of words, but potentially provides insights into boundaries of place references as well.

An example is displayed in Example (4.16) where four place references were annotationed (*my bedroom*, *home*, *Rathmines Road*, *Hawthorn East*) with their respective identifiabilities rendered on top of each. The identifiability feature for the sentence presented in Example (4.16) is translated as shown in Table 4.22. (Similar to the chunk tag feature described in Section 4.1.3, we adopt IOB tags.)

(4.16) I am in $\overbrace{\underline{\text{my bedroom}}}^{\text{no}}$ at $\overbrace{\underline{\text{home}}}^{\text{no}}$, on $\overbrace{\underline{\text{Rathmines Road}}}^{\text{yes\_unamb}}$, $\overbrace{\underline{\text{Hawthorn East}}}^{\text{yes\_unamb}}$.

We define templates for this feature in Table 4.23 and assume the current word is the ith word in a sentence.

4.3 Evaluation Methodology

In this section, we introduce the methods that are used to evaluate the classifier. First, we explain the methodology we use to assess the correctness of each prediction

Template	Description
Windows of neighbouring identifiabilities	The identifiability of the nth word $(i-2 \leq n \leq i+2)$
Combinations of two immediate neighbouring identifiabilities	The combination of the identifiabilities of the nth word and the $(n+1)$th word $(i-1 \leq n \leq i)$

Table 4.23: Template Setup for Identifiability Feature

(Section 4.3.1). Next, we move on to the introduction of the methodology employed to evaluate the performance of the model (Section 4.3.2). Lastly, we introduce three baseline systems (Section 4.4).

4.3.1 Full Span Locative Expression Evaluation

In the task of identifying locative expressions, the primary concern is the performance of the model at identifying the full span of locative expressions rather than identifying rather than identifying component words (possibly missing some component words). Therefore, a locative expression is considered incorrectly predicted if the label of one word in it is assigned the wrong label.

To illustrate, see the example in Table 4.24. The first two words are correctly predicted to not be locative expression, and are therefore, *true negatives.* The three-word phrase *at the end* is not a locative expression but predicted as one, therefore is considered to be a *false positive.* The phrase *on Malibu Mews* is correctly identified as locative expression, hence, *true positive.* The last phrase *in Chadstone* is incorrectly rejected but actually is a locative expression, thus, *false positive.*

To evaluate the prediction result, precision, recall and F$_{\beta=1}$ are adopted. Precision represents the fraction of retrieved locative expressions that are correct whereas recall stands for the fraction of relevant locative expressions that are retrieved. F$_{\beta=1}$ is the harmonic mean of precision and recall.

They are calculated as shown in Equations 4.17, 4.18 and 4.19.

$$precision = \frac{TP}{TP + TN} \tag{4.17}$$

$$recall = \frac{TP}{TP + FP} \tag{4.18}$$

$$F_{\beta=1} = (1 + \beta^2) \times \frac{precision \times recall}{(\beta^2 \times precision) + recall} \tag{4.19}$$

Word	Correct Label	Predicted Label
I	O	O
am	O	O
at	O	B-NP
the	O	I-NP
end	O	I-NP
of	B-NP	I-NP
the	I-NP	I-NP
court	I-NP	I-NP
on	B-NP	B-NP
Malibu	I-NP	I-NP
Mews	I-NP	I-NP
in	B-NP	O
Chadstone	I-NP	O
.	O	O

Table 4.24: An Example of Full Span of Locative Expressions Evaluation

Precision	Recall	$F_{\beta=1}$
50.00%	33.33%	40.00

Table 4.25: Evaluation Result of Table 4.24

If we apply Equations 4.17, 4.18 and 4.19 to the example shown in Table 4.24, we obtain results shown in Table 4.25.

For evaluation purposes, we employ the *conlleval*[3] Perl script.

4.3.2 10-Fold Cross-Validation

In this research, we employ 10-fold cross-validation to evaluate the performance of the model as 10-fold cross-validation has been proven more effective than the more expensive hold-one-out cross-validation (Kohavi *et al.*, 1995). Specifically, we split the collected place descriptions into 10 mutually exclusive subsets of equal length. To evaluate the performance of a model, one subset is held out at a time as the testing document and the rest is used to train the model. Next, the trained model is applied to the held out testing subset. Lastly, the accuracy is calculated as the total number of correct predictions across the entire dataset.

[3] `http://www.cnts.ua.ac.be/conll2000/chunking/output.html`

Iteration									
1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9	9	9
10	10	10	10	10	10	10	10	10	10

Table 4.26: Example of 10-fold Cross-validation

An example of 10-fold cross-validation is shown in Table 4.26. Each highlighted cell represents the held out subsets of the current iteration.

4.4 Baseline Systems

In this section, we introduce three baseline systems with which we benchmark our purposed method.

4.4.1 StanfordNER

StanfordNER (Section 3.4.1) is used out-of-the-box. We apply the same set of rules described in Section 3.1.4 to entities recognised by *StanfordNER*.

A place reference sometimes equals to a geospatial named entity as shown in Example (4.20). The underlined pharse *Bourke Street* is identified as a named entity by *StanfordNER*.

(4.20) 570 Bourke Street, DES Building

In most cases, however, it is not always true that a place reference is equal to a locative expression. As underlined in Example (4.21), even though the formal place references, *Rathmines Road* and *Hawthorn East*, can be recognised by *StanfordNER*, informal place references, *my bedroom* and *home*, are not identified. In this example, the locative expression *on Rathmines Road, Hawthorn East* is correctly recognised while the other two informal-place-reference-based locative expressions, *in my bedroom* and *at home*, cannot be recognised.

Word	Correct Label	*StanfordNER*	*Unlock Text*
on	O	O	O
the	O	O	O
south	O	O	O
side	O	O	O
of	B-NP	B-NP	O
Albert	I-NP	I-NP	O
Park	I-NP	I-NP	B-NP
Lake	I-NP	I-NP	I-NP
.	O	O	O

Table 4.27: Comparison between *StanfordNER* and *Unlock Text*

(4.21) I am in my bedroom at home, on Rathmines Road, Hawthorn East.

Given the fact that *StanfordNER* only works well on locative expressions based on formal place references, we hypothesise that it will not be able to generate competitive results. Therefore, we utilise another functionality provided by *StanfordNER* — retrainability.

4.4.2 Unlock Text

Unlock Text (Section 3.4.2) is adopted and used in conjunction with the set of rules described in Section 3.1.4 to identify locative expressions.

Similar to *StanfordNER*, *Unlock Text* is designed to spot formal place references from natural language rather than informal ones. However, the accuracy of *Unlock Text* is inferior to *StanfordNER*. A comparison is given in Table 4.27. In *StanfordNER*, *of Albert Park Lake* is correctly recognised as a locative expression where *Unlock Text* is only able to spot *Park Lake*.

4.4.3 Re-trained StanfordNER

As mentioned in Section 4.4.1, using *StanfordNER* out-of-the-box is highly unlikely to perform well in the task of identifying locative expression, especially for informal-place-reference-based locative expressions. Therefore, we re-train *StanfordNER* using the same corpus we feed to our model.

A comparison between re-trained *StanfordNER* and the other two baseline systems is provided in Table 4.28. In *StanfordNER*, only one out of five locative expressions (*in Abbotsford*) can be identified whereas *Unlock Text* is able to recognise only two (*Trenerry Crescent*, *in Abbotsford*). In the case where re-trained *StanfordNER* is

Word	Correct Label	*StanfordNER*	*Unlock Text*	Re-trained *StanfordNER*
I	O	O	O	O
am	O	O	O	O
in	B-NP	O	O	B-NP
apartment	I-NP	O	O	I-NP
22	B-NP	O	O	I-NP
of	I-NP	O	O	I-NP
the	I-NP	O	O	I-NP
Byfass	I-NP	O	O	I-NP
apartment	I-NP	O	O	I-NP
in	B-NP	O	O	B-NP
8	I-NP	O	O	I-NP
Trenerry	B-NP	O	B-NP	B-NP
Crescent	I-NP	O	I-NP	I-NP
in	B-NP	B-NP	B-NP	B-NP
Abbotsford	I-NP	I-NP	I-NP	I-NP
.	O	O	O	O

Table 4.28: Comparison between *StanfordNER*, *Unlock Text* and Re-trained *StanfordNER*

used, all locative expressions can be identified. Therefore, we hypothesise that re-trained *StanfordNER* is able to generate more competitive results than the other two baseline systems.

4.5 Chapter Summary

In this chapter, we present methodologies used in this research.

In Section 4.1, we explain features that can be extracted from the corpus automatically without the use of manual annotations. Specifically, ten features are introduced with similar features grouped into one feature. For each feature, its template setup is also defined in its respective subsection.

In Section 4.2, we further introduce the gold standard setup which exploits manual annotations. With the aid of such information, high performance is assumed.

In Section 4.3, the evaluation methodology is introduced. First, we discuss full span locative expression evaluation. Essentially, it evaluates the model by the ability of identifying the full span of locative expressions rather than identifying component words. In order to measure the performance, we adopt precision, recall and

$F_{\beta=1}$. Moreover, we employ 10-fold cross-validation to evaluate the performance of the model.

In Section 4.4, we study the three baseline systems, *StanfordNER*, *Unlock Text* and re-trained *StanfordNER*, used to benchmark our purposed method.

Chapter 5

Experiments

In this chapter, we present our experimental results. First, in Section 5.1 we reveal the performance of baseline systems. Next, in Section 5.2, we show how the model performs in the automatic identification setup (Section 4.1). Lastly, in Section 5.3 we present the performance of the model in the gold standard setup.

Since it is not guaranteed that every feature is contributive, we adopt *feature ablation* to test the effectiveness of our features (Chapter 4), for instance, the POS tag feature (Section 4.1.2) and the chunk tag feature (Section 4.1.3). In *feature ablation*, we remove one feature at a time and monitor how the performance ($\mathrm{F}_{\beta=1}$) changes. A feature is considered unproductive if, by removing it, the performance of the model increases. The removing-and-monitoring process continues until no unproductive feature can be found.

5.1 Performance of Baseline Systems

In this section, we examine the performance of baseline systems mentioned in Section 4.4. As can be observed from Figure 5.1, the precision and recall of both *StanfordNER* and *Unlock Text* are unbalanced with precision much higher than recall. This indicates that when the systems detect a locative expression it is often correct, but also that their coverage is very low. Such low performance is not all that surprising since both *StanfordNER* and *Unlock Text* aim at spotting *geospatial named entities* rather than *geospatial noun phrases.* 30.2% (922 out of 3,061) of the place references in the manual annotations are *geospatial noun phrases*. Neither *StanfordNER* nor *Unlock Text* is able to identify many locative expressions that contain such place references. In fact, only 3 out of 922 *geospatial noun phrases* can be identified by *Unlock Text.* Further, even though the remaining 69.8% (2,139 out of 3,061) of the place references in the manual annotations are *geospatial named entities*, only few can be picked up by *StanfordNER* and *Unlock Text.* In *Unlock Text*'s case, only 113 out of 2,139 *geospatial named entities* can be identified. Given the small percentage of

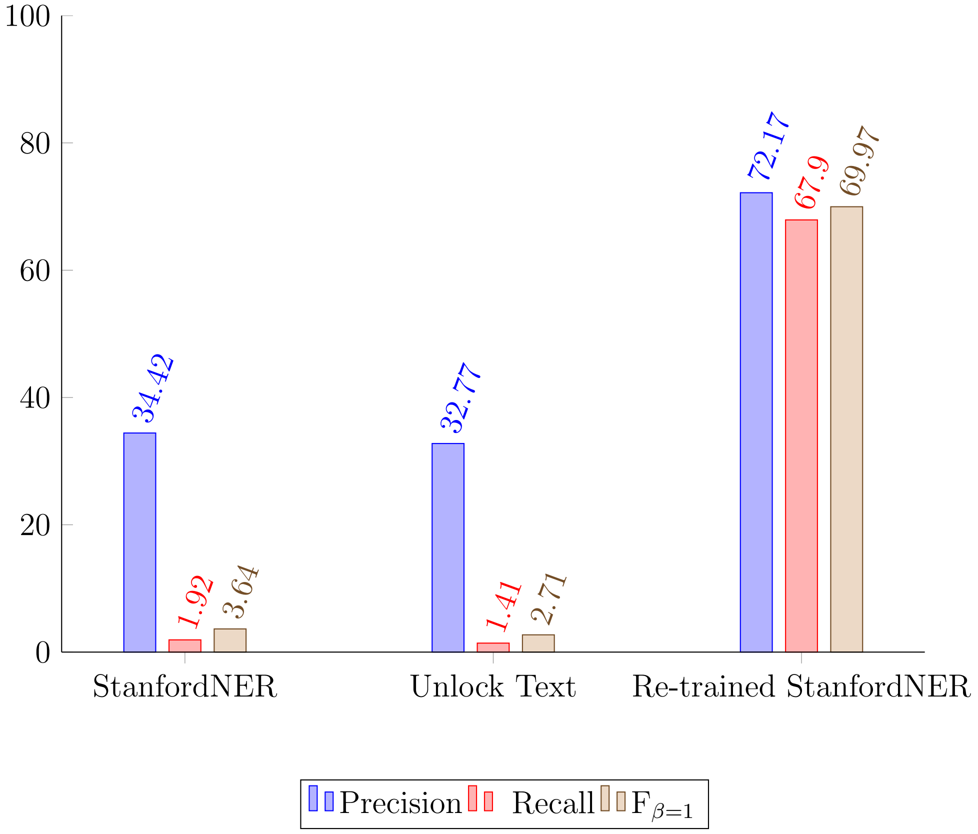

Figure 5.1: Performance of Baseline Systems

place references that can be spotted, the low performance is not all that surprising.

Re-trained *StanfordNER*, on the other hand, performs significantly better than the two baseline systems mentioned above. The precision and recall of re-trained *StanfordNER* are balanced. The more competitive performance of re-trained *StanfordNER* is not all that surprising as it is re-trained on the exact same data as we use to train our model, therefore, re-trained *StanfordNER* aiming at identifying locative expressions rather than *geospatial named entities*.

An example is shown in Table 5.1. *StanfordNER* is able to identify *Rathmines Road* and *Hawthorn East* as two separate place references. Applying the same set of rules described in Section 3.1.4, the preceding preposition *on* is included into the locative expression as well as the connective comma which concatenates the two place references together. Having said that, however, the named entity recogniser is still unable to recognise *in my bedroom* and *at home* as locative expressions as they are not formal place references (geospatial named entities) but informal ones (geospatial

Word	Correct Label	StanfordNER	Unlock Text	Re-trained StanfordNER
I	O	O	O	O
am	O	O	O	O
in	B-NP	O	O	B-NP
my	I-NP	O	O	I-NP
bedroom	I-NP	O	O	I-NP
at	B-NP	O	O	B-NP
home	I-NP	O	O	I-NP
,	O	O	O	O
on	B-NP	B-NP	O	B-NP
Rathmines	I-NP	I-NP	O	I-NP
Road	I-NP	I-NP	O	I-NP
,	I-NP	I-NP	O	I-NP
Hawthorn	I-NP	I-NP	O	I-NP
East	I-NP	I-NP	O	I-NP
.	O	O	O	O

Table 5.1: Example of Identification Results of Three Baseline Systems

noun phrases).

As for *Unlock Text*, no locative expression can be identified. The geoparser fails to recognise any place reference in this particular place description.

In the case where re-trained *StanfordNER* is used, all locative expressions are correctly identified. The learner is re-trained on the same corpus using features such as the word feature, the word character ngram feature[1] and the word shape feature. Therefore, the re-trained *StanfordNER* aims at identifying locative expressions rather than named entities. As shown in Figure 5.1, a significant improvement in precision, recall as well as $F_{\beta=1}$ can be observed. In this particular example (Table 5.1), re-trained *StanfordNER* is able to assigned the correct label to every word.

5.2 Performance of Automatic Identification Setup

The performance of our model is presented in Table 5.2 (for evaluation purposes, we employ the *conlleval*[2] Perl script.). The highest $F_{\beta=1}$ is achieved by having all features but the word feature, the chunk-preceding preposition feature and the auto-

[1] only words with character length greater than 6 are eligible and are used as prefixes and suffixes

[2] `http://www.cnts.ua.ac.be/conll2000/chunking/output.html`

Automatic Identification Setup			
Feature Set	Precision	Recall	$F_{\beta=1}$
All	76.57%	75.15%	75.86
—Word	76.81%	75.55%	76.17
—POS Tag	76.24%	75.08%	75.66
—Word Position	76.42%	75.12%	75.76
—Chunk Tag	76.30%	74.97%	75.63
—Text Normalisation	75.92%	74.79%	75.35
—POS Tag within Chunk	76.58%	75.08%	75.82
—Chunk-preceding Preposition	76.81%	75.34%	76.07
—Automatic Geospatial Feature Class (*GeoNames* + *VICNAMES*)	76.12%	74.90%	75.50
—Automatic Geospatial Feature Class (*GeoNames*)	76.89%	75.66%	76.27
—Automatic Geospatial Feature Class (*VICNAMES*)	76.41%	75.19%	75.80
—Locative Indicator & Motion Verb	75.62%	74.46%	75.04
—Motion Verb / Lemmatisation	76.36%	74.86%	75.60
—(Word, Text Normalisation, Motion Verb / Lemmatisation)	71.21%	69.71%	70.45
—(Word, Chunk-preceding Preposition)	76.93%	75.70%	76.31
—(Word, *GeoNames*, Chunk-preceding Preposition)	77.15%	76.06%	76.60
—(Word, *VICNAMES*, Chunk-preceding Preposition)	76.84%	75.59%	76.21
Baseline Systems			
Feature Set	Precision	Recall	$F_{\beta=1}$
StanfordNER	34.42%	1.92%	3.64
Unlock Text	32.77%	1.41%	2.71
Re-trained StanfordNER	72.17%	67.90%	69.97

Table 5.2: Performance of Automatic Identification Setup

matic geospatial feature class (*GeoNames*) feature turned on.

With all features described in Section 4.1 turned on, the model is able to achieve 75.86 in $F_{\beta=1}$, a 5.89 increase over the best performing baseline system (re-trained *StanfordNER*). Further, we use feature ablation to figure out the best performing feature setup which consists of all features but the word feature, the chunk-preceding preposition feature and the automatic geospatial feature class (*GeoNames*). With the best performing feature setup, the model is further improved to 76.60 in $F_{\beta=1}$, a 6.63 increase over the re-trained *StanfordNER*. To further investigate the impacts of these features, we present statistics in Table 5.2. Details are explained in Section 5.2.1.

5.2.1 Studying the Performance Increase

In this section, we investigate reasons for the performance gain as the statistics presented in Table 5.2. First, we examine the reason for the performance boost when the automatic geospatial feature class feature (*GeoNames*) is eliminated from the feature setup. Next, we discuss the connection between the performance increase and the word feature. Lastly, we study the reason why the chunk-preceding preposition feature has negative impacts on the performance.

The Automatic Geospatial Feature Class Feature

Two gazetteers (*GeoNames* and *VICNAMES*) can be used in place of the automatic geospatial feature class feature. As can be observed, the automatic geospatial feature class (*GeoNames + VICNAMES*) feature is contributive to the model as the performance declines to 75.50 if the feature is eliminated entirely from the setup. Eliminating the automatic geospatial feature class (*GeoNames + VICNAMES*) feature from the setup damages recall more than precision. This is not all that surprising since the connection between place references recognised by external gazetteers and locative expressions is strong. With the aid of external gazetteers, the model is likely to be able to identify more locative expressions.

If only one gazetteer is used, however, the model generates better results than having both *GeoNames* and *VICNAMES* eliminated. Particularly, the model is able to achieve 76.27 in $F_{\beta=1}$ when the automatic geospatial feature class (*GeoNames*) is not used. Given the fact described above, it is suggested that the two gazetteers are mutually exclusive since if one of them is eliminated from the setup, the total performance increases. Table 5.6 shows some statistics of *GeoNames* and *VICNAMES*. Out of all the matched name references, only 455 name references are matched by both *GeoNames* and *VICNAMES*. The rest of the name references that are matched by one gazetteer but not the other have a negative impact on the performance of our model. Therefore, it is advisable that only one of the gazetteer is used.

Comparing with having the automatic geospatial feature class (*GeoNames + VICNAMES*) turned off, two increases in $F_{\beta=1}$ are achieved by having either the automatic geospatial feature class *GeoNames* or *VICNAMES* feature removed. To select the better one out of the two, we simply compare the performance of the model with one gazetteer turned on and the other off. As can be observed from Table 5.2, eliminating the automatic geospatial feature class (*GeoNames*) feature generates better results than having the automatic geospatial feature class (*VICNAMES*) removed.

Comparisons between the two gazetteers are presented in Table 5.3 and Table 5.4. In Table 5.3, the place reference *warana* is recognised by *GeoNames*. Therefore, in the setup where *GeoNames* is used, *warana drive* is identified as a locative expression. In contrast, *VICNAMES* is unable to recognise *warana*, which leads to the failure of the identification of the locative expression *warana drive*. In the second example, shown

Word	*GeoNames*	*VICNAMES*	Correct Label	Using *GeoNames*	Using *VICNAMES*
cursing	None	None	O	O	O
up	None	None	O	O	O
warana	B-S	O	B-NP	B-NP	O
drive	O	O	I-NP	I-NP	O

Table 5.3: Comparison between *GeoNames* and *VICNAMES* #1

Word	*GeoNames*	*VICNAMES*	Correct Label	Using *GeoNames*	Using *VICNAMES*
walking	None	None	O	O	O
to	None	None	B-NP	B-NP	B-NP
john	B-T	B-PARK	I-NP	I-NP	I-NP
cain	None	I-PARK	I-NP	B-NP	I-NP
memorial	B-L	I-PARK	I-NP	I-NP	I-NP
park	I-L	I-PARK	I-NP	I-NP	I-NP

Table 5.4: Comparison between *GeoNames* and *VICNAMES* #2

in Table 5.4, *john cain memorial park* is recognised as a single place reference by *VICNAMES* whereas in *GeoNames* it is recognised as two separate place references rather than one. Hence, *john cain memorial park* is correctly correctly identified in the setup where *VICNAMES* is used but incorrectly identified when *GeoNames* is used.

However, considering the low coverage of both gazetteers, a fairly large proportion of locative expressions cannot be identified just using external gazetteers. Hence, the differences of the abilities of both gazetteers to identify place references play a minor role.

To understand the reason why *VICNAMES* performs better than *GeoNames*, we investigate some statistics of *GeoNames* and *VICNAMES*. As presented in Table 5.6, 1,311 place references can be assigned geospatial feature class by *GeoNames* whereas in *VICNAMES* the number drops down to 861. Despite the low coverage, the accuracy of matched place references in *VICNAMES* is higher than *GeoNames*. 39.9% (524 out of 1,311) of the matched place references are actually locative expressions in *GeoNames* while the percentage in *VICNAMES* rises up to 56.3% (485 out of 861). In *GeoNames*, more than 60% of the matched place references are irrelevant to the process of identifying locative expressions from informal text. The confidence of the model predicting a place reference recognised by *VICNAMES* is higher than one

Word	*GeoNames*	*VICNAMES*	Correct Label	Using *GeoNames*	Using *VICNAMES*
working	None	None	O	O	O
at	None	None	B-NP	O	B-NP
citiport	O	O	I-NP	O	I-NP

Table 5.5: Comparison between *GeoNames* and *VICNAMES* #3

recognised by *GeoNames*.

The likelihood of a place reference recognised by *VICNAMES* being part of a locative expression is higher than a place reference recognised by *GeoNames*. The model, therefore, is more confident about predicting a place reference recognised by *VICNAMES* as part of a locative expression than a place reference recognised by *GeoNames*. That is to say, for recognised place references, the weight for the feature function of the automatic geospatial feature class (*VICNAMES*) feature in *CRF* is higher than the weight for the automatic geospatial feature class (*GeoNames*) feature.

For non-recognised place references, *VICNAMES* tends to be more aggressive than *GeoNames* in terms of identifying those place reference as locative expressions. An example is shown in Table 5.5. Even though *citiport* is recognised neither by *GeoNames* nor by *VICNAMES*, in the setup where *VICNAMES* is used, it is still correctly identified as part of a locative expression.

For place references that are not recognised, *O*s (outside) are assigned. Since the number of matched place references in *VICNAMES* (861) is lower than *GeoNames* (1,311), the number of words assigned *O*s is higher in *VICNAMES* than in *GeoNames*. While more place references are assigned *O*s in *VICNAMES* than in *GeoNames*, the number of manually annotated place references that are not matched by *GeoNames* is similar to that of *VICNAMES*. Given the facts mentioned above, it is suggested that the indicativeness of a place reference with the automatic geospatial feature class (*VICNAMES*) feature assigned *O* is less than a place reference with the automatic geospatial feature class (*GeoNames*) feature assigned the same value. Therefore, the weight assigned to the feature function of the automatic geospatial feature class (*VICNAMES*) feature with the value *O* in *CRF* is likely to be lower than the counterpart for *GeoNames*. In the setup where *VICNAMES* is used, the label of a non-recognised place reference is less dependent on the automatic geospatial feature class (*VICNAMES*) feature.

Further, we examine the effectiveness of the automatic geospatial feature class based on either *GeoNames* or *VICNAMES* in the case where the other two highlighted features, namely the word feature and the chunk-preceding preposition feature, are removed. Under such conditions, the performance of the setup with *VICNAMES*

Gazetteer	# of matched place references	# of matched manual annotations
GeoNames	1,311	524
VICNAMES	861	485

Table 5.6: Number of Matched Place References of *GeoNames* and *VICNAMES*

Word	Lemmatisation	Lexical Normalisation	Correct Label	All	—Word
viewing	view	viewing	O	O	O
the	the	the	B-NP	O	B-NP
kinglake	kinglake	kinglake	I-NP	B-NP	I-NP
national	national	national	I-NP	I-NP	I-NP
park	park	park	I-NP	I-NP	I-NP

Table 5.7: Comparison between Word, Lemmatisation and Lexical Normalisation Features

($F_{\beta=1}$ = 76.60, the best performing automatic identification setup) is still superior to the setup with *GeoNames* ($F_{\beta=1}$ = 76.21).

The Word Feature

By eliminating the word feature from the automatic identification setup, the model is able to achieve a $F_{\beta=1}$ of 76.17. Since the canonical versions of words are already provided by the text normalisation feature, the word feature is then considered redundant. The performance gain can be primarily attributed to the removal of the redundant feature.

An example is shown in Table 5.7. In this example, the first word can be lemmatised to *view* whereas the other words are all in their canonical forms. As can be observed, the lexical normalisation feature provides exactly the same information as the word feature since no non-standard word exists in this example. Considering the fact that in cases where non-standard words exist, the model benefits from the lexical normalisation feature as non-standard words can be recovered to their standard forms. Therefore, by eliminating the word feature, we essentially reduce the redundancy of the training data without sacrificing useful information.

Further, we examine whether lexical features are effective in the task of identifying locative expressions by having all lexical features (Word, Text Normalisation, Motion Verb / Lemmatisation) excluded from the feature setup. The performance declines dramatically to 70.45 in $F_{\beta=1}$. The performance drop is justified since no information

Word	Chunk Tag	Chuck-preceding Preposition	Correct Label	All	Without Chunk-Preceding Preposition
At	B-PP	None	B-NP	B-NP	B-NP
the	B-NP	At	I-NP	I-NP	I-NP
tram	I-NP	At	I-NP	I-NP	I-NP
stop	I-NP	At	I-NP	I-NP	I-NP
lonsdale	I-NP	At	B-NP	I-NP	B-NP
st	I-NP	At	I-NP	I-NP	I-NP

Table 5.8: Comparison between Using and Not Using the Chunk-preceding Preposition Feature

on the text of a word, normalised or not, is provided to the model. It can be concluded that the lexical features are of importance even though they are less generalisable than features such as the POS tag feature and the chunk tag feature.

The Chunk-preceding Preposition Feature

Having the chunk-preceding preposition feature removed from the automatic identification setup boosts the performance up to 76.07 in $F_{\beta=1}$. As templates of windows of neighbouring words, POS tags and chunk tags defined in Section 4.1 are fed to the model, the model already has information on chunk-preceding prepositions. The chunk-preceding preposition feature only provides redundant information and therefore can be eliminated from the feature setup.

An example is shown in Table 5.8. In this example, the *NP* (*noun phrase*) chunk *the tram stop lonsdale st* is preceded by a *PP* (*prepositional phrase*) *At* and therefore every word in the chunk is assigned *At* as its chunk-preceding preposition feature. Even though shallow parsed into the same chunk, *the tram stop* and *lonsdale st* are two separate place references. The feature setup with all features on is unable to classify *lonsdale st* as a separate locative expression. In the case where the chunk-preceding preposition feature is eliminated, the model correctly recognises *lonsdale st* as a separate locative expression.

In the case where a *NP* chunk is preceded by a *PP* chunk, the chunk-preceding preposition feature provides information about not only the preceding preposition but also the boundary of the chunk. However, the model already takes templates of windows of neighbouring words, POS tags and chunk tags into account. Not to mention the fact that chunk tag is also used as a feature and fed to the model. Therefore, information provided by the chunk-preceding preposition feature is considered redundant as it suggests that words in the same chunk are more likely to end up in one locative expression. With the chunk-preceding preposition feature on, the model receives more information on chunk boundaries than it needs and is more likely to classify words shallow parsed into the same chunk as one locative expression.

5.2.2 Error Analysis

In this section, we present the error analysis based on the best automatic identification setup, which is achieved by eliminating the following three features:

- Word
- Chunk-preceding Preposition
- Automatic Geospatial Feature Class (*GeoNames*)

Locations Outside of Victoria

In the corpus, only place references that reside within Victoria, Australia were manually annotated. Place references situated elsewhere (e.g., *On the moon*, *Unit in Lusty St, Wolli Creek.*), along with place descriptions that have nothing to do with locations (e.g., *Economics class*, *No, thanks*), are annotated irrelevant and not used to expand to locative expressions. Therefore, expressions that refer to locations positioned outside of Victoria, Australia are not re-annotated as locative expressions.

On the other hand, the model is not able to distinguish place references located outside of Victoria, Australia. As shown in Table 5.9, the place reference (*Lusty St, Wolli Creek*) is located outside of Victoria, Australia and was therefore not manually annotated as a place reference. However, apart from the fact that it is not in Victoria, Australia, the place references in this place description are eligible for being part of a locative expression. In the learning phase, feeding such locative expressions confuses the model since they are not classified as locative expressions while they should have been. The performance of the model may somehow be affected since it penalises itself by decreasing the weights on feature functions that represent patterns of genuine locative expressions.

Since our interest lies in the identification of locative expressions regardless of their locations, the example shown in Table 5.9 should be counted as a *true positive* while it is now treated as a *false positive*. The performance is damaged by such correctly identified out-of-Victoria locative expressions.

Currently, no information on the location of a place reference is fed to the model. Therefore, it is highly likely that the model is able to distinguish and identify them as locative expressions. However, external gazetteers, such as *GeoNames*, are able to provide such information. Such additional information may enable the model to identify place descriptions more accurately. Alternatively, a more feasible solution to this problem is modifying the manually annotations so as to include out-of-Victoria place references.

Word	Correct Label	Predict Label
Unit	O	O
in	O	B-NP
Lusty	O	I-NP
St	O	I-NP
,	O	I-NP
Wolli	O	I-NP
Creek	O	I-NP
.	O	O

Table 5.9: Example of Locative Expression Outside of Victoria

Errors at the Word *The*

The model tends to get confused at the word *the*. An example is presented in Table 5.10. The possible reason the phrase *at the* is also classified as part of the locative expression is because the word *the* is preprocessed as part of the *NP* chunk *the golden beach*. Hence, the phrase *the golden beach* is recognised as a place reference and *at the golden beach* is then recognised as a locative expression.

In the second example shown in Table 5.11, even though the phrase *the malvern golf club* is recognised as one *NP* chunk, the word *the* is excluded from the locative expression.

The primary reason lies in the manual annotations. Example (5.1) and Example (5.2) show how the two place descriptions in Table 5.10 and Table 5.11 were manually annotated. Manually annotated place references are underlined. In the first example, the word *the* is included in the manual annotation as part of the place reference whereas in the second example it is excluded. Such inconsistency is unavoidable since the annotation process was done manually and therefore is subjective.

(5.1) at the golden beach surfing up the waves

(5.2) about to go to the malvern golf club for some drinks

Fed with such inconsistent data, the model may not be able to figure out a clear pattern to apply when it comes across *NP* chunks that start with the word *the*. The model is then forced to rely on other features which may not turn out very useful.

Errors at Postal Codes

In some place descriptions, postal codes are placed at sentence ends. However, some postal codes in the corpus are not recognised correctly. Although the model

Word	Chunk Tag	Correct Label	Predicted Label
at	B-PP	O	B-NP
the	B-NP	O	I-NP
golden	I-NP	B-NP	I-NP
beach	I-NP	I-NP	I-NP
surfing	B-VP	O	O
up	B-PRT	O	O
the	B-NP	O	O
waves	I-NP	O	O

Table 5.10: Prediction Error #1 at the Word *the*

Word	Chunk Tag	Correct Label	Predicted Label
about	B-PP	O	O
to	B-VP	O	O
go	I-VP	O	O
to	B-PP	B-NP	O
the	B-NP	I-NP	O
malvern	I-NP	I-NP	B-NP
golf	I-NP	I-NP	I-NP
club	I-NP	I-NP	I-NP
for	B-PP	O	O
some	B-NP	O	O
drinks	I-NP	O	O

Table 5.11: Prediction Error #2 at the Word *the*

is able to recognise some of the postal codes as separate locative expressions, it tends to make mistakes when it comes to the pattern where two place references are concatenated by a comma and the postal code is placed at the end of the sentence (an example is shown in Table 5.12). In cases like this, postal codes are likely to be identified as parts of the concatenated locative expressions.

In the example shown in Table 5.12, the preceding word of the postal code, namely *melbourne*, is recognised as a *location* by *VICNAMES*. The postal code is preprocessed as part of the same chunk that contains the preceding word. Given the fact that the postal code is included in the same chunk with the preceding word, the model is unable to distinguish the postal code from the chunk and tends to categorise them as one single place reference.

On the other hand, in cases where postal codes are not in the same chunks as

Word	Chunk Tag	Correct Label	Predicted Label
8	B-NP	B-NP	B-NP
exhibition	I-NP	B-NP	B-NP
street	I-NP	I-NP	I-NP
,	O	I-NP	I-NP
melbourne	B-NP	I-NP	I-NP
3000	I-NP	B-NP	I-NP

Table 5.12: Prediction Error of Postal Code

Word	Chunk Tag	Correct Label	Predicted Label
16	B-NP	B-NP	B-NP
Oliver	I-NP	B-NP	B-NP
Road	I-NP	I-NP	I-NP
Templestowe	I-NP	B-NP	B-NP
3106	B-NP	B-NP	B-NP

Table 5.13: Example of a Correctly Classified Postal Code

the preceding place references, the model is able to classify postal codes as separate locative expressions (Table 5.13).

As explained above, whether the model is able to identify postal codes as separate locative expressions largely depends on the preprocessing of the corpus. A postal code is likely to be recognised if it is not in the same chunk as its preceding word.

In order to cope with postal codes, the help of extra features is required. We may devise a feature that represents if the end of a sentence is a number. Additionally, the model may benefit from another feature that checks if the number at the end of the sentence is a four-digit number.

Errors at Sentences Ends

As shown in Table 5.14 and Table 5.15, the model is not good at identifying one-word locative expressions situated at ends of sentences. Such locative expressions tend to be identified as parts of the preceding locative expressions.

The last words, namely *mall* and *junction*, in both examples shown in Table 5.14 and Table 5.15 are not identified as separate locative expressions. Same as errors at postal codes, the reason for errors at sentences ends is that words are shallow parsed into the same chunks as their preceding word.

Even though the last word *mall* in Table 5.14 is manually annotated as a separate place reference, in reality it is common sense that *street mall* is treated as an undivided

Word	Chunk Tag	Correct Label	Predicted Label
stanley	B-VP	B-NP	B-NP
street	B-NP	I-NP	I-NP
mall	I-NP	B-NP	I-NP

Table 5.14: Prediction Error #1 at the Sentence End

Word	Chunk Tag	Correct Label	Predicted Label
near	B-PP	B-NP	B-NP
camberwell	B-NP	I-NP	I-NP
junction	I-NP	B-NP	I-NP

Table 5.15: Prediction Error #2 at the Sentence End

Gold Standard Setup			
Feature Set	Precision	Recall	$F_{\beta=1}$
Best Performing Automatic Identification Setup +Manual Annotation	99.42%	99.60%	99.51

Table 5.16: Performance of Gold Standard Setup

place reference. The same is true for the last word *junction* in Table 5.15 since it is generally considered to be part of the place reference *camberwell junction*. Depending on how we define locative expression, it is not incorrect if we count the last words in both place descriptions as part of the preceding locative expressions rather than two separate ones. According to the definition of locative expression (Section 2.2), both *street mall* and *camberwell junction* can be considered as locative expressions. Therefore, we believe the model predicts correct labels in such cases.

5.3 Performance of Gold Standard Setup

In this section, we present the performance of our model when used on the gold standard setup. In addition to all features included in the best performing automatic identification setup, the gold standard setup also includes the features introduced in Section 4.2.

The performance of the gold standard setup is presented in Table 5.2 with highest $F_{\beta=1}$ being 99.51.

A comparison between the identification results of the gold standard setup and the automatic identification setup is presented in Table 5.17. Both setups are able

Word	Identifiability	Correct Label	Gold Standard Setup	Automatic Identification Setup
Outside	None	B-NP	B-NP	B-NP
the	B-no	I-NP	I-NP	I-NP
restaurant	I-no	I-NP	I-NP	I-NP
on	None	B-NP	B-NP	B-NP
warrigul	B-yes_unamb	I-NP	I-NP	I-NP
rd	I-yes_unamb	I-NP	I-NP	I-NP
with	None	B-NP	B-NP	O
the	B-no	I-NP	I-NP	O
big	I-no	I-NP	I-NP	O
fake	I-no	I-NP	I-NP	O
volcano	I-no	I-NP	I-NP	O

Table 5.17: Gold Standard Setup vs Automatic Identification Setup

to identify the first two locative expressions. The third locative expression is derived from the third place reference *the big fake volcano* with the preceding preposition *with*. Even though categorised as *non-identifiable*, the third place reference, together with its preceding preposition *with*, constitutes the third locative expression. However, it is not the identifiability of the place reference but the fact that the boundary of the place reference is clearly defined by its identifiability that helps the model locate locative expressions. From the identifiability feature, the model is able to figure out the start position and end position of a place reference. With the help of gold standard data (e.g., *identifiability*), the model can spot the third locative expression. Without such assist, the model with the automatic identification setup cannot locate the third locative expression. As can be observed in Table 5.17, the performance of the model declines significantly if the manual annotation feature is eliminated from the gold standard setup.

5.4 Chapter Summary

Within this chapter, we present the performance results of the baseline systems, the automatic identification setup and the gold standard setup.

Firstly, the performances of the baseline systems were presented. It can be draw by comparison that re-trained *StanfordNER* outperforms the other two baseline systems by a substantial margin in this particular task of automatically identifying locative expressions from informal text. We believe the reason for such performance difference is primarily due to the difference between the tasks that these baseline systems are designed to accomplish. A comparison of the identification results between the baseline systems over a particular place description is provided to illustrate the differences between the baseline systems.

Next, the performances of the automatic identification setup are presented. We study the impact of each single feature and attempt to identify unproductive features using *feature ablation*. With all features described in Chapter 4 turned on, the performance of our model comes out at 75.86 in $F_{\beta=1}$. By further examination and eliminating unproductive features from the automatic identification setup, the model is able to achieve 76.60 in $F_{\beta=1}$, a 6.63 increase over the best performing baseline system (re-trained *StanfordNER*). The connections between eliminating the unproductive features and the performance increases are discussed as well. Following this discussion, we analyse some of the common errors in the identification results and provide possible solutions to tackle those problems.

Lastly, the performances of the gold standard setup are presented. The gold standard setup is based on the best performing automatic identification setup. Similar to the automatic identification setup, we adopt *feature ablation* to figure out the best performing setup for the gold standard setup. The best performing gold standard setup is achieved by having all features turned on and the performance comes out at 99.51 in $F_{\beta=1}$. By further examination we discover that eliminating the the manual annotation feature from the setup has the most negative impact on performance, suggesting the manual annotation feature provide the most useful information to the model.

Chapter 6

Conclusion

6.1 Summary

In this thesis, we present our research methodology and results of the task of automatic identification of locative expressions from informal text.

In Chapter 2, we review background knowledge relevant to our research, such as natural language processing and machine learning. Further, the definition of locative expression is introduced.

Next, in Chapter 3, resources used in this research are introduced. First, we discuss the corpus, which was sourced from the *Tell Us Where* project. Following the introduction of the corpus, the preprocessing of the corpus, manual annotations and the method adopted to automatically re-annotate locative expressions are introduced. Next, we introduce the machine learning application we employ (*CRF++*). Then we move on to external resources which help the learning model gain additional information. Lastly, we introduce benchmark tools against which we evaluate our learning model.

In Chapter 4 we introduce the feature setups used in this research and the evaluation methodology. Two types of feature setups are introduced, the automatic identification setup and the gold standard setup. In the automatic identification setup, we reveal features that are extracted from the corpus automatically whereas in the gold standard setup section, features derived from manual annotations are explained. Following that, we present the evaluation methodology and the three baseline systems built on top of *StanfordNER* and *Unlock Text*.

In Chapter 5, the experiment results are presented. We examine the performance of the two different feature setups using the evaluation methodology introduced in Chapter 4. Before diving into the details of the performance of either the automatic identification setup or the gold standard setup, we first show the performance of the baseline systems. Due to the uniqueness of the problem, we discover that using *StandardNER* out-of-the-box and *Unlock Text* generates uncompetitive results ($F_{\beta=1}$ of *StanfordNER*: 3.64 and $F_{\beta=1}$ of *Unlock Text*: 2.71). Since *StanfordNER* can be re-trained using the same corpus as the one we feed to our learning model, we also examine the performance of the re-trained *StanfordNER*. The $F_{\beta=1}$ of re-trained *StanfordNER* comes out at 69.97, which is used in later sections as the baseline. With all features turned on, the automatic identification setup is able to achieve 75.84 in $F_{\beta=1}$, a 5.89 increase. After feature ablation, features marked as not contributive are eliminated from the feature setup. Having the unhelpful features removed from the feature set, the best performing automatic identification setup is able to achieve a further improvement, 76.60 in $F_{\beta=1}$, a 6.63 performance gain. Lastly, we examine the gold standard setup. With the assist of manual annotations, the learning model is able to achieve 99.51 in $F_{\beta=1}$.

Even though the result of the best performing automatic identification setup is not as high as the gold standard setup, it still well exceeds the baseline system with fairly balanced precision and recall. Therefore, it is safe to say that our learning model is able to make a positive impact on the task of automatic identification of locative expressions from informal text.

6.2 Conclusions

In this research, we develop a system that is able to identify locative expressions automatically within informal text.

Further, we also discover insights of what aspects are helpful in the identification task. Specifically, we discuss the mutual exclusiveness of the two gazetteers, *GeoNames* and *VICNAMES* (Section 5.2.1). Moreover, we study that *VICNAMES* performs better than *GeoNames* as the accuracy of *VICNAMES* is higher than *GeoNames* and therefore the possibility of a place reference identified by *VICNAMES* being part of a locative expression is higher than the possibility of a place reference identified by *GeoNames*. Furthermore, we find out that both the word feature and the chunk-preceding preposition feature provide no more than redundant information to the model, which, eventually, damages the performance of the model. Therefore, these two features are better left out of the feature setup.

Additionally, we identify some common errors and provide analysis of each type of errors. We discover that the model is able to make sensible prediction in most cases. In some cases, however, the reason of the error can be attributed to the inconsistency in the manual annotation scheme. Admittedly, such inconsistency is unavoidable since the corpus was annotated manually. Having said that, a better performance result can be expected if a finer-grained corpus is used.

Finally, our model is able to achieve state-of-the-art performance (76.60 in $F_{\beta=1}$, a 6.63 increase over the baseline system).

6.3 Future Work

As mentioned in Section 5.2.2, the learning model is not able to distinguish locative expressions within Victoria, Australia from ones located elsewhere. Since only place references in Victoria are candidates to locative expressions, the model's inability to discriminate place references based on their locations damages the performance. A possible solution to this problem is extracting the location of a place reference and use it as a feature to feed to the learning model. With such additional information, it is possible that the model is able to figure out the pattern that place references situated outside of Victoria, Australia are unlikely to be locative expressions.

Since the primary cause of the model's confusion at the word *the* is the inconsistency in the manual annotation scheme, a finer-grained corpus with more appropriate annotations is likely to further improve the performance.

When describing one's location in an informal way, people tend to use informal descriptions of locations and not pay attention to using the proper cases of words. Since most place descriptions in the corpus are not proper cased, using capitalisation of each words as a feature may not be as useful as it could be when applied to proper cased corpus. An interesting idea was brought up by Ritter *et al.* (2011)., where not only the capitalisation of words but whether the whole sentence is properly cased is taken into account as well (Ritter *et al.*, 2011). Applying such an idea to our learning model may help in terms of improving the performance.

Bibliography

Abney, Steven. 1992. *Parsing by chunks*. Springer.

Blaylock, Nate, Bradley Swain, and James Allen. 2009. Tesla: a tool for annotating geospatial language corpora. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, NAACL-Short '09, 45–48, Stroudsburg, PA, USA. Association for Computational Linguistics.

Finkel, Jenny Rose, Alex Kleeman, and Christopher D Manning. 2008. Efficient, feature-based, conditional random field parsing. *Proceedings of ACL-08: HLT* 959–967.

Han, Bo, Paul Cook, and Timothy Baldwin. 2013. Lexical normalization for social media text. *ACM Transactions on Intelligent Systems and Technology (TIST)* 4.5.

Herskovits, Annette. 1985. Semantics and pragmatics of locative expressions. *Cognitive Science* 9.341–378.

Hill, Linda L. 2000. Core elements of digital gazetteers: placenames, categories, and footprints. In *Research and Advanced Technology for Digital Libraries*, 280–290. Springer.

Jurafsky, Dan, James H Martin, and Andrew Kehler. 1999. *Speech and language processing: an introduction to natural language processing, computational linguistics and speech recognition*, volume 2. MIT Press.

Kohavi, Ron, and others. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International joint Conference on artificial intelligence*, volume 14, 1137–1145. Lawrence Erlbaum Associates Ltd.

Kudo, Taku, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying conditional random fields to japanese morphological analysis. In *Proceedings of EMNLP*, 230–237.

Mikheev, Andrei, Marc Moens, and Claire Grover. 1999. Named entity recognition without gazetteers. In *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*, EACL '99, 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.

Miller, Harvey J. 2010. The data avalanche is here: Shouldn't we be digging?. *Journal of Regional Science* 50.181 – 201.

Munoz, Marcia, Vasin Punyakanok, Dan Roth, and Dav Zimak. 2000. A learning approach to shallow parsing. *arXiv preprint cs/0008022* .

Ritter, Alan, Sam Clark, Oren Etzioni, and others. 2011. Named entity recognition in tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 1524–1534. Association for Computational Linguistics.

Sarawagi, Sunita, and William W Cohen. 2004. Semi-markov conditional random fields for information extraction. *Advances in Neural Information Processing Systems* 17.1185–1192.

Snow, Rion, Daniel Jurafsky, and Andrew Y Ng. 2004. Learning syntactic patterns for automatic hypernym discovery. *Advances in Neural Information Processing Systems 17* .

Sproat, Richard, Alan W Black, Stanley Chen, Shankar Kumar, Mari Ostendorf, and Christopher Richards. 2001. Normalization of non-standard words. *Computer Speech & Language* 15.287–333.

Sutton, Charles, and Andrew McCallum. 2010. An introduction to conditional random fields. *arXiv preprint arXiv:1011.4088* .

Tjong Kim Sang, Erik F, and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, 142–147. Association for Computational Linguistics.

Toutanova, Kristina, and Colin Cherry. 2009. A global model for joint lemmatization and part-of-speech prediction. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, ACL '09, 486–494, Stroudsburg, PA, USA. Association for Computational Linguistics.

Tytyk, Igor. Towards interpreting informal place descriptions. Master's thesis, The University of Melbourne.

Winter, Stephen, Allison Kealy, Matt Duckham, Abbas Rajabifard, Kai-Florian Richter, and Tim Baldwin. 2011. *Starting to talk about place.*.

Wu, Yunhui, and Stephan Winter. 2011. Interpreting destination descriptions in a cognitive way. In *Workshop on Computational Models for Spatial Language Interpretation and Generation (CoSLI-2)*, 8–15.

Zhang, Xiao, Prasenjit Mitra, Alexander Klippel, and Alan MacEachren. 2010. Automatic extraction of destinations, origins and route parts from human generated route directions. In *Geographic Information Science*, 279–294. Springer.

Zhou, GuoDong, and Jian Su. 2002. Named entity recognition using an hmm-based chunk tagger. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, 473–480, Stroudsburg, PA, USA. Association for Computational Linguistics.

Appendix A

Types of Part-of-speech Tags

Type	Description
Tag	Description
CC	conjunction, coordinating
CD	cardinal number
DT	determiner
EX	existential there
FW	foreign word
IN	conjunction, subordinating or preposition
JJ	adjective
JJR	adjective, comparative
JJS	adjective, superlative
LS	list item marker
MD	verb, modal auxillary
NN	noun, singular or mass
NNS	noun, plural
NNP	noun, proper singular
NNPS	noun, proper plural
PDT	predeterminer
PRP	pronoun, personal
PRP$	pronoun, possessive
RB	adverb
RBR	adverb, comparative
RBS	adverb, superlative
RP	adverb, particle
SYM	symbol
TO	infinitival to
UH	interjection
VB	verb, base form
VBZ	verb, 3rd person singular present
VBP	verb, non-3rd person singular present
VBD	verb, past tense
VBN	verb, past participle
VBG	verb, gerund or present participle
WDT	wh-determiner
WP	wh-pronoun, personal
WP$	wh-pronoun, possessive
WRB	wh-adverb
.	punctuation mark, sentence closer
,	punctuation mark, comma
:	punctuation mark, colon
(	contextual separator, left paren
)	contextual separator, right paren

Table A.1: Types of Part-of-speech Tag

Appendix B

Types of Chunk Tags

Type	Description
NP	Noun phrase
PP	Prepositional phrase
VP	Verb phrase
ADVP	Adverb phrase
ADJP	Adjective phrase
SBAR	Subordinating conjunction
PRT	Particle
INTJ	Interjection

Table B.1: Types of Chunk