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

May 2013

Declaration

This is to certify that:

(i) the thesis comprises only my original work towards the PhD except where indicated in the Preface;

(ii) due acknowledgement has been made in the text to all other material used;

(iii) the thesis is fewer than 100,000 words in length, exclusive of tables, maps, bibliographies and appendices.

Signed: ______________________________ Date: ____________________

Thesis advisor(s) Author
Timothy Baldwin **Fei Liu**
Maria Vasardani

Automatic Identification of Locative Expressions from Informal Text

Abstract

Language technology rules, OK.

Contents

Citations to Previously Published Work

Large portions of Chapter 1 have appeared in the following paper:

> Kim Smith (2005) LT Stuff, In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, Ann Arbor, USA, pp. 1–8.

Acknowledgments

I want to thank my Mum, and my Dad, and my Uncle Bruce, and my Aunty Gertrude, and ...

Dedicated to youse all.

Chapter 1

Introduction

1.1 Motivation

1.2 Research Question

1.3 Contribution

1.4 Definition of Locative Expression

1.5 Scope of the Thesis

1.6 Structure of the Thesis

Chapter 2

Background

2.1 Related Work

Chapter 3

Resources

In this chapter, we introduce resources used in this research. First, the corpus used in this research is introduced. The corpus was sourced from a web-hosted location-based mobile game project - *Tell Us Where*[1]. Next, we move on to the machine learning model. Lastly, external resources, such as gazetteers and dictionaries, are introduced as they provide additional features for the model to learn from, thereby, further improving the performance of our model.

3.1 Corpus

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

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 located by the on-board GPS, participants were asked to answer the question "Tell us where you are?" and submit descriptions about their locations through a web interface via their cellphones. 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.

[1] `http://telluswhere.net/`

In this research, the data was collected from 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 (See Section 3.1.2) combined with manual annotations (See 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 purpose of *part-of-speech tagging* (*POS tagging*) and *shallow parsing* (*chunking*). *POS tagging* is the process of identifying words as nouns, verbs, adjectives, adverbs and etc. according to their particular part of speech whereas *chunking* is the process of analysing and identifying the constituents of a sentence but not their internal structure. *OpenNLP*[2] 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 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 _ 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 by Igor Tytyk. Each place reference was annotated with its granularity level and identifiability, both of which were marked with the assist of external gazetteers, namely OpenStreetMap[3] and Google Maps.[4]

[2] `http://opennlp.apache.org/index.html`
[3] `http://www.openstreetmap.org/`
[4] `http://maps.google.com/`

Table 3.1: Detail Information of Annotation

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*

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 which can later be used to automatically re-annotate locative expressions according to the set of rules presented in Definition **??**.

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. 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 (See Table 3.3).

With the help of manual annotations, 3,279 place references were extracted. How-

Table 3.2: Possible Values of Identifiability

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 3.3: Possibile Values of Granularity Level

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*

ever, 218 place references were marked irrelevant. 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

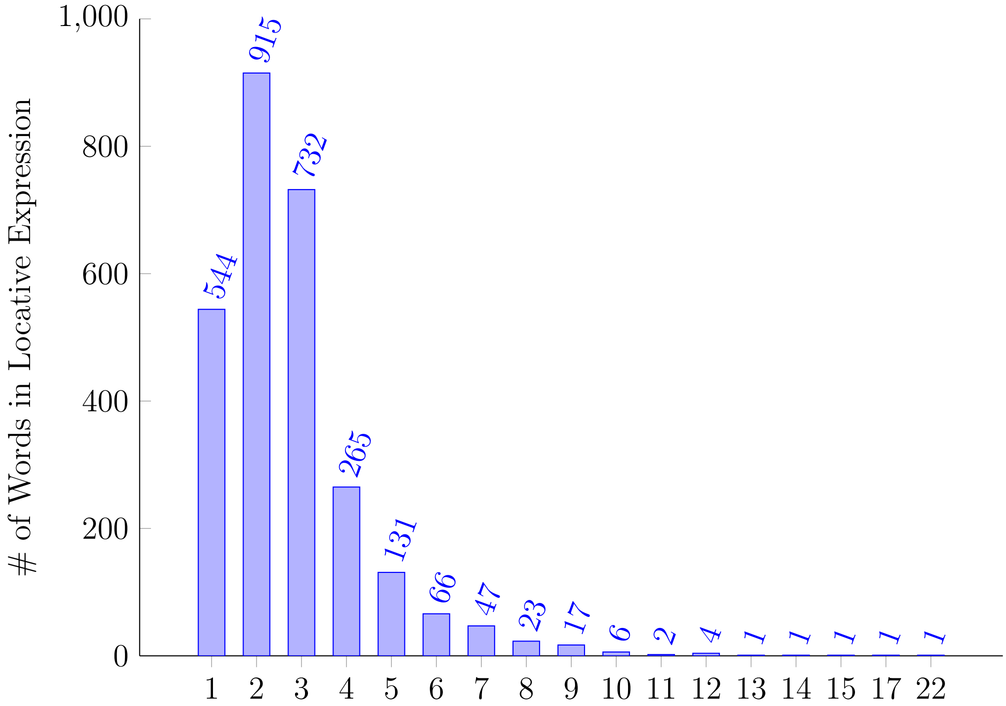
Due to the fact that a locative expression consists of at least one place reference, the process of identifying locative expressions can be interpreted as the task of expanding place references to locative expressions and concatenating multiple place references to one locative expression.

Based on the definition of locative expression (See Section 1.4), 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. Prepositional phrases, namely *of*[5] and *and*, are considered semantic connectors that concatenate the two surrounding place references, thereby, constituting a larger locative expression.

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

Figure 3.1: Distribution of Word Count of Locative Expressions



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

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

(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.

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 Methodology

In this section, the mathematical model for learning (See Section 3.2.1) is explained. Next, we introduce the application, which is an implementation of the mathematical model, used in this research.

3.2.1 Conditional Random Fields

Conditional Random Fields (*CRFs*) are widely used for sequential labelling tasks (Kudo *et al.* 2004). In natural language processing tasks, the prediction of a label of a word relies not only on the text of a word but the 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 brought in 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 and 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 3.5 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).

$$f(s, i, l_i, l_{i-1}) \tag{3.5}$$

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, the Equation 3.6 is repeated until it reaches certain stopping conditions.

$$\lambda'_j = \lambda_j + \alpha[\sum_{i=1}^{n} f_j(s, i, l_i, l_{j-1}) - \sum_{l'} p(l'|s) \sum_{i=1}^{n} f_j(s, i, l'_i, l'_{i-1})] \tag{3.6}$$

In Equation 3.6, 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 3.7.

$$score(l|s) = \sum_{j=1}^{m}\sum_{i=1}^{n} \lambda_j f_j(s, i, l_i, l_{i-1}) \tag{3.7}$$

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 3.8 where the l' is all possible label sequences.

$$p(l|s) = \frac{\exp(score(l|s))}{\sum_{l'} \exp(score(l'|s))} \tag{3.8}$$

3.2.2 CRF++

CRF++[6] is an open-sourced, highly-customizable implementation of the CRF model written in C++. It can be applied to a wide variety of NLP tasks thanks to its generic design which allows feature sets to be redefined. Both training and testing functions are provided and can perform their designated tasks respectively with optimised memory usage and minimum time consumption. Considering the merits mentioned above, we adopt CRF++ as our machine learning model to which we feed our data.

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 its features as well. Sentences should be separated by empty lines. In this example, apart from the current word, one additional feature: POS tag, is listed in Table 3.4 as the second column and the last column is the correct label of the word. More features are allowd to 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, a word being inside a locative expression and a word being outside of a locative expression.

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

Table 3.4: Example Training Data for CRF++

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

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 to be defined. 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.

3.3 External Resources

Two types of external resources, gazetteers and dictionaries, are introduced as they provide data sources upon which we can improve the performance of our model.

Table 3.5: Core Components of Gazetteer

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	The category of a place.	*A* (country, state, region,)

Properties stored in external resources can be used as features to feed to *CRF++* with the aim of improving the performance of the system.

3.3.1 Gazetteers

A gazetteer is defined as 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*[7] and *VICNAMES*[8]. As stated by *Hill et al*, a gazetteer consists of three core components as shown in Table 3.5 (Hill 2000). In order to improve the performance of the learning model, we make use of the type of a place reference.

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, 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 while referring to the same place. To cope with such cases, alternative names are included as one feature of every entry as well.

In this research, we adopt *GeoNames* as an external gazetteer and extract useful information such as feature class and feature code[9]. 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.

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

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

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

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.

3.3.2 Dictionaries

Of all the nouns and verbs, some, such as the underlined words shown in Example 3.9 and Example 3.10, 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.

(3.9) 103 hephman street

(3.10) 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[10] 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.

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

Such inheritance relationships enable us to obtain a set of words that are geospatially related to locative expressions.

In this research, *Natural Language Toolkit*[11] (*nltk*) is used to access *WordNet*.

Lexical Normalisation Dictionary

To deal with non-standard words, an external lexical normalisation dictionary[12] is employed. Essentially, the dictionary provides mappings between typos, ad hoc abberviations, etc. and canonical spellings.

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

As shown in Example 3.11, the word *appartment* is misspelt and the canonical spelling is *apartment*. The dictionary enables us to transform mispelt words back to their canonical forms. (See Section 4.1.5)

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*[13] (*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. Named entity recognition is the task of identifying references to entities such as person, organisation and location names, and numeric expressions (e.g., time, date, money and percent expressions) from unstructured text (Nadeau and Sekine 2007). Equipped with well-engineered features and the training data which can be downloaded from the the website[14], the application takes text as input and identifies named entities in the input text, which is similar to our task of identifying locative expressions.

A place reference sometimes equals to a named entity as shown in Example 3.12. The underlined phrase *Bourke Street* is identified as a named entity by *StanfordNER*. It is also annotated as a place reference.

[11] `http://nltk.org/`
[12] `http://ww2.cs.mu.oz.au/~tim/etc/emnlp2012-lexnorm.tgz`
[13] `http://nlp.stanford.edu/software/CRF-NER.shtml`
[14] `http://www-nlp.stanford.edu/software/CRF-NER.shtml#Download`

(3.12) 570 Bourke Street, DES Building

In most cases, however, it is not always true that a place reference equals to a locative expression. As highlighted in Example 3.12, *Bourke Street, DES Building* should be recognised as one locative expression while *StanfordNER* only manages to identify *Bourke Street*.

Thus, given the uniqueness of the task, it is highly unlikely that using *Stanford-NER* out-of-the-box is able to produce any competitive results.

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.

3.4.2 Unlock Text

Unlock Text[15], 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, which is the similar to our task of identifying locative expressions. Thus, we employ *Unlock Text* as the second baseline system.

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

Chapter 4

Methodology

In this chapter, we introduce the methods used to get the best performance out of the learning model. 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 without the assist 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 learning 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) <u>at home</u> <u>in bed</u>

(4.2) I'm <u>at home</u> <u>in Kensington</u>

Apart from the text of the current word, additional information about neighbouring words is taken into account as well to help the learning 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*.

Table 4.1: An Example of Word Feature

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

An example is presented in Table 4.1. In this particular example, we assume *home* is the current word.

Assume the current word is the ith word in a sentence, and we define templates shown in Table 4.2 for this feature.

The interpretation of templates defined in Table 4.2 of Example 4.1 is presented in Table 4.3 with *home* being the current word .

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 covered in the training corpus. Simply matching the text of words does not provide much useful information. In such cases, matching POS tags of words provides more general 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.

Table 4.2: Template Setup for 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 word and the $(n+1)$th word ($i-2 \leq n \leq i+1$)

Table 4.3: Example of Mapping between Template and Words

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*

(4.3) <u>On_IN the_DT train_NN</u> at_IN bentleigh_NN

(4.4) <u>In_IN the_DT car_NN</u> on_IN the_DT corner_NN of_IN Arden_NNP St_NNP and_CC Dryburgh_NNP St_NNP

Assume the current word is the ith word in a sentence, and we define templates shown in Table 4.4 for this feature.

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*. 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.

Table 4.4: Template Setup for POS Tag Feature

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 word 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 word and the $(n{+}1)$th word and the $(n{+}2)$th word $(i{-}2 \leq n \leq i)$

Most place references (2,584 out of 3,061, 84.4%) start with chunks, therefore, chunks have indications of boundaries of place references. To discriminate the beginning and the rest of a chunk, IOB tags are employed. Hence, the beginning of a chunk is marked *B-* + chunk tag (e.g., *B-NP*) and words in the rest of the chunk are tagged *I-* + chunk tag (e.g., *I-NP*).

Assume the current word is the ith word in a sentence, and we define templates shown in Table 4.6 for this feature.

4.1.4 Word Position

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

(4.5) [ADVP $\overbrace{\text{Approximate\_RB}}^{0}$ $\overbrace{\text{halfway\_RB}}^{1}$] [PP $\overbrace{\text{between\_IN}}^{0}$] [NP $\overbrace{\text{Lara\_NNP}}^{0}$ $\overbrace{\text{and\_CC}}^{1}$ $\overbrace{\text{Little\_NNP}}^{2}$ $\overbrace{\text{River\_NNP}}^{3}$] [VP $\overbrace{\text{...}}^{0}$]

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

Assume the current word is the ith word in a sentence, and we define templates shown in Table 4.7 for this feature.

Table 4.5: Example of Mapping between Template and POS Tags

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 4.6: Template Setup for Chunk Tag Feature

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 word and the $(n+1)$th word $(i-2 \leq n \leq i+1)$

4.1.5 Text Normalisation

In the filed 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 (A good reference here). A pair of examples is shown in Example 4.6 and Example 4.7. The underlined words, *walking* and *walked*, are eventually lemmatised to *walk*. Thus, lemmatisation essentially removes differences in morphology and increase the chance of matching of two words derived from the same origin. In this research, we adopt the text of lemmatised words as a feature.

Table 4.7: Template Setup for Word Position 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 4.8: Template Setup for Lemmatisation Feature

Template	Description
Windows of neighbouring lemmatised words	The lemmatised words 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 words of the nth word and the POS tag of the nth word $(i-2 \leq n \leq i+2)$

(4.6) walking down the street

(4.7) walked here from my house

To lemmatise words, we utilise the package *WordNetLemmatizer* included in *nltk*. It is essentially based on *WordNet*[1].

Assume the current word is the ith word in a sentence, and we define templates shown in Table 4.8 for this feature.

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.

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

Table 4.9: Template Setup for Lexical Normalisation Feature

Template	Description
Windows of neighbouring lexical normalised words	The lexical normalised words 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 ($i-2 \leq n \leq i+1$)

(4.8) on hollywood <u>avenu</u>

(4.9) 23 espirte <u>aven</u>

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, therefore, are used as the feature without any modification.

Assume the current word is the ith word in a sentence, and we define templates shown in Table 4.13 for this feature.

4.1.6 Chunk-Preceding Prepositional Word

According to the definition of locative expression, a locative expression may consist of not only one or multiple place reference(s) but prepositional words as well. In fact, 1,103 out of 2757 locative expressions start with a prepositional word, not to mention locative expressions preceded by lexical connectors (e.g., *of*, *and*). As can be drawn from the statistics, prepositional words play an essential role in determining the beginnings of locative expressions. Therefore, we employ prepositional words as a feature to feed to the learning model.

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

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

If and only if the chunk is preceded by a *prepositional phrase* (*PP*) and the chunk itself is a *noun phrase* (*NP*), then every word in the chunk is assigned the text of the

Table 4.10: An Example of Chunk-Preceding Prepositional Word Feature

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

Table 4.11: Template Setup for Chunk-Preceding Propositional Words Feature

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

preceding *PP*. An example of the interpretation of this feature using Example **??** is presented in Table 4.10.

Assume the current word is the ith word in a sentence, and we define templates shown in Table 4.11 for this feature.

4.1.7 Automatic Geospatial Feature Class

Intuitively, some prepositional words tend to be used in conjunction with particular types of place references. The connection between prepositional words and place references' geospatial categories is supposed to have impacts on the process of determining the boundaries of locative expressions. Therefore, it is sensible to mark place references with their corresponding geospatial categories which can be used as a feature to improve the performance of the learning model.

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]

Table 4.12: An Example of Geospatial Feature Class Feature

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

Table 4.13: Template Setup for 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 ($i-2 \leq n \leq i+1$)

$$\text{[NP } \overbrace{\text{the\_DT You\_PRP Yangs\_NNP}}^{\text{T (mountain, hill, rock, \ldots)}} \text{] .\_.}$$

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 (See Section 3.3.1), place references can be assigned feature classes according to their geospatial categories. If and only if place references that can be found in gazetteers are feature classes assigned. An example of the interpretation of this feature using Example 4.11 is presented in Table 4.12.

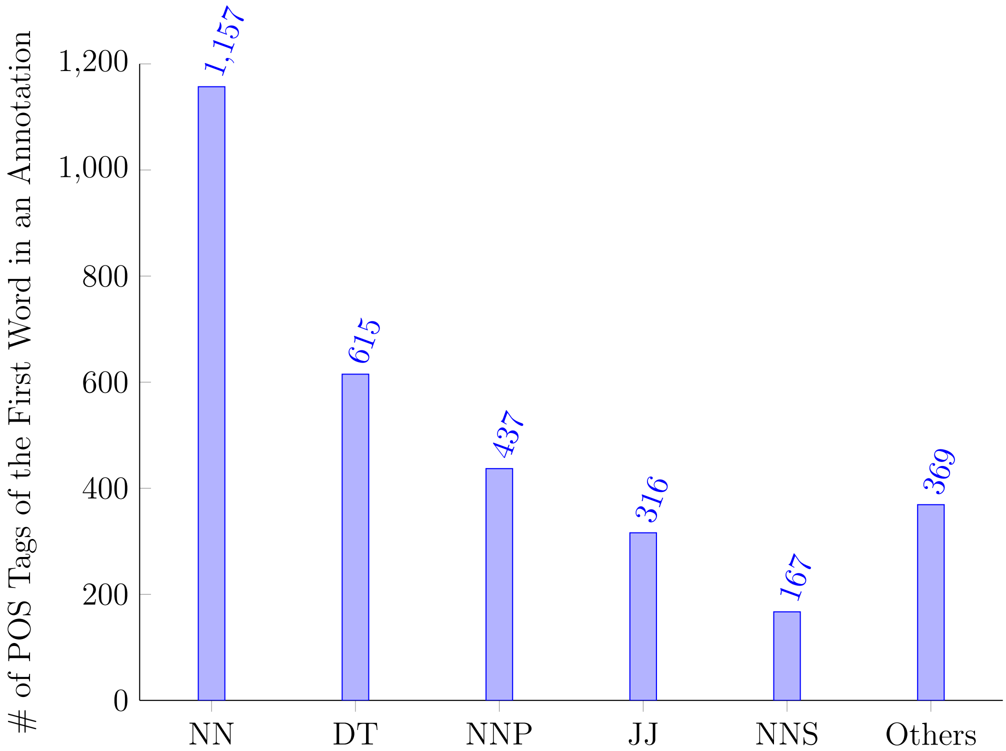
In the case where *GeoNames* is used, 1,311 matches can be found whereas in *VICNAMES* the number of matches is 861.

Assume the current word is the ith word in a sentence, and we define templates shown in Table 4.13 for this feature.

4.1.8 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 the POS tag of the first word in an annotation 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) the total number. Since locative expressions are expanded from place references and most place references are essentially derived from 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] ._.

Assume the current word is the ith word in a sentence, and we define templates shown in Table 4.15 for this feature.

Table 4.14: An Example of First POS Tag Feature

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

Table 4.15: Template Setup for First POS Tag Feature

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

4.1.9 Most Frequent POS Tag

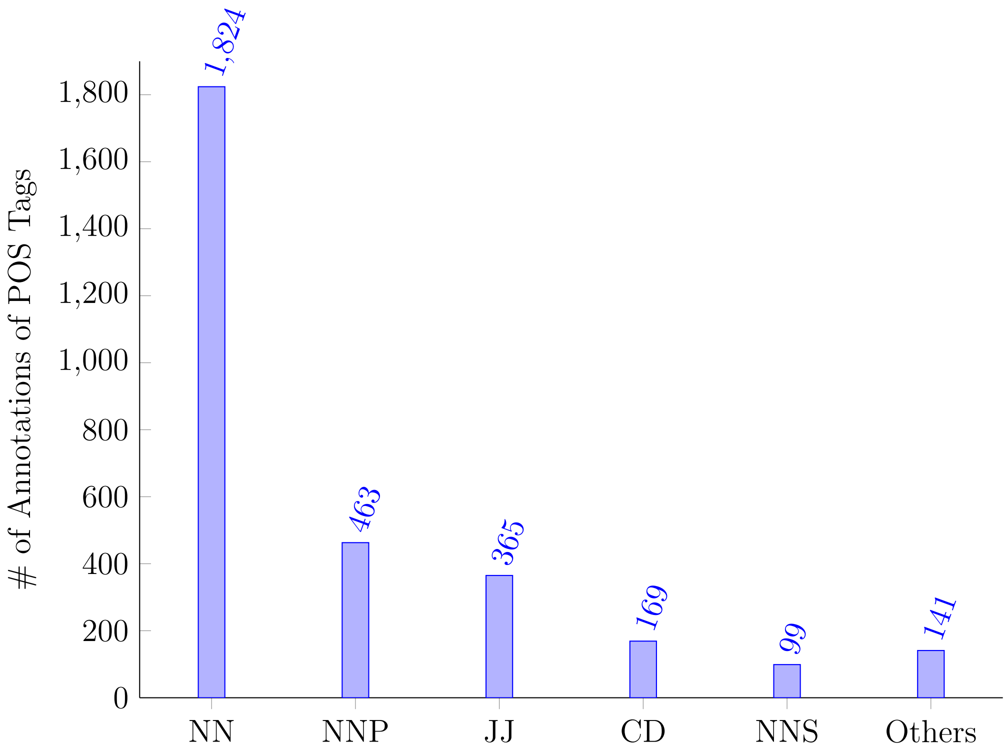
It can be observed from Figure 4.2 that the most frequent POS tag in more than 50% of the manual annotations is *NN*. The sum of the top five consumes 95% of all manual annotations. As mentioned in Section 4.1.8, according to the connection between manual annotations and locative expressions, it is suggested that the most frequent POS tag in a chunk has implications of a chunk being a 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] ._.

Assume the current word is the ith word in a sentence, and we define templates shown in Table 4.17 for this feature.

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



4.1.10 Locative Indicator and Motion Verb

As mentioned in Section 3.3.2, some nouns (*locative indicators*) and verbs (*motion verbs*) have strong indications of representing 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

Table 4.16: An Example of Most Frequent POS Tag Feature

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

to match against words in the corpus. Ultimately, each words 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.

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

(4.14) I am at Flinders Street Station in Melbourne.

Assume the current word is the ith word in a sentence, and we define templates for the locative indicator feature and motion verb feature in Table 4.19 and Table 4.20 respectively.

Further, motion verbs tend to be used in conjunction with particular prepositional words. As can be seen from Example 4.15, the place reference *collins street* is preceded by a motion verb *walking* and a prepositional word *down.* The combination of *walking* and *down* has clear implication of *down collins street* being a locative expression.

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

Assume the current word is the ith word in a sentence, and we define templates that reflect this feature in Table 4.21.

Table 4.17: Template Setup for First POS Tag Feature

Template	Description
Windows of neighbouring most frequent POS tags	The most frequent POS tag of 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 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 nth word and the number 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 contains the nth word and the number of words in the chunk ($i-1 \leq n \leq i+1$)

4.2 Gold Standard Setup

To help the learning model determine the sequence of labels of words, we make use of the manually annotated corpus to get the maximum performance. Features, together with their templates, are introduced in this section.

4.2.1 Identifiability

The identifiability of a place reference reflects the uniqueness of the place reference in question within Victoria. Three possible values can be assigned as shown in Table 3.2.

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 by any place reference, *None*s are assigned. Consequently, this feature does not only provide information about identifiabilities of words but potentially feeds knowledge on boundaries of place references as well.

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

An example is displayed in Example 4.16 where four place references were annotationed ("*my bedroom*", "*home*", "*Rathmines Road*", "*Hawthorn East*") with their

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

Word	Locative Indicator	Motion Verb
I	False	False
am	False	True
at	False	False
Flinders	False	False
Street	True	False
Station	True	False
in	False	False
Melbourne	False	False
.	.	

Table 4.19: Template Setup for Locative Indicator Feature

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 ($i-2 \leq n \leq i+1$)

respective identifiabilities hovering over. The identifiability feature for the sentence presented in Example 4.16 is translated as shown in Table 4.22.

Assume the current word is the ith word in a sentence, and we define templates shown in Table 4.23 for this feature.

4.2.2 Preceding Prepositional Word

According to the definition of locative expression, a locative expression may consist of not only one or multiple place reference(s) but prepositional words as well. In fact, 1,103 out of 2757 locative expressions start with a prepositional word, not to mention locative expressions preceded by lexical connectors (e.g., *of*, *and*). As can be drawn from the statistics, prepositional words play an essential role in determining the beginnings of locative expressions. Therefore, we employ prepositional words as a feature to feed to the learning model. Moreover, since most prepositional phrases are used in conjunction with either particular place references of particular granularity level or identifiability, it is advisable to take the combinations of preceding prepositional words and features mentioned above into account as well.

Table 4.20: Template Setup for Motion Verb 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 4.21: Template Setup for Motion Verb Combination Feature

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

(4.17) [ADVP Approximate_RB halfway_RB] [PP between_IN] [NP Lara_NNP and_CC Little_NNP River_NNP] [VP ._.]

As displayed in Example 4.17, two place references were annotated as underlined. The first place reference *Lara_NNP* is preceded by a prepositional phrase (hence the chunk tag *PP*) and is therefore assigned *between* as its precdeing prepositional word. The second one *Little_NNP River_NNP*, even though located inside the chunk, has *and_CC* as its predecessor which is assigned to both *Little* and *River*. For words that is neither annotated as place references nor included in annotations not preceded by prepositional phrases, *None* is assigned. An example of the interpretation of this feature using Example 4.17 is presented in Table 4.24.

Assume the current word is the ith word in a sentence, and we define templates shown in Table 4.25 for this feature.

Table 4.22: An Example of Identifiability Feature

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

4.2.3 Geospatial Feature Class

Intuitively, some prepositional words tend to be used in conjunction with particular types of place references. The connection between prepositional words and place references' geospatial categories is supposed to have impacts on determining the boundaries of locative expressions. Therefore, it is sensible to mark place references with their corresponding geospatial categories which can be used as a feature to improve the performance of the learning model.

As displayed in Example 4.18, *Lara_NNP* and *Little_NNP River_NNP* are identified as place references by *GeoNames*.

(4.18) [ADVP Approximate_RB halfway_RB] [PP between_IN] [NP $\overbrace{\text{Lara\_NNP}}^{\text{S (spot, building, farm)}}$ and_CC $\overbrace{\text{Little\_NNP River\_NNP}}^{\text{H (stream, lake, \ldots)}}$] [VP ._.]

With the help of external gazetteers (See Section 3.3.1), annotated place references can be assigned feature classes according to their geospatial categories. For place references that are cannot be found in gazetteers, *None* is assigned. An example of the interpretation of this feature using Example 4.18 is presented in Table 4.26.

In the case where *GeoNames* is used, 607 manual annotations can be found.

Assume the current word is the ith word in a sentence, and we define templates shown in Table 4.27 for this feature.

Table 4.23: Template Setup for Identifiability Feature

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 $(i-2 \leq n \leq i+1)$

Table 4.24: An Example of Preceding Prepositional Word Feature

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

4.3 Evaluation Methodology

In this section, we introduce the methods that are used to evaluate the learning model. First, we explain the methodology we use to assess the correctness of each prediction (See Section 4.3.1). Next, we move on to the introduction of the methodology employed to evaluate the performance of the learning model (See Section 4.3.2). Lastly, we introduce two baseline systems (See Section 4.4).

4.3.1 Locative-expression-span-level Evaluation

In the task of identifying locative expressions, the primary concern is the performance of the learning model on locative-expression-span-level rather than on word-level. Therefore, a locative expression is considered incorrectly predicted if the label of one word in it is assigned a wrong label.

To explain locative-expression-spal-level evaluation, we adopt *positive predicitive value* and *negative predicitive value* (See Table 4.28).

An example is shown in Table 4.29. The first two words are correctly predicted as not locative expression, therefore, *FN*. The three-word phrase *at the end* is not a locative expression but predicted as one, therefore is considered *TN*. The phrase *on*

Table 4.25: Template Setup for Preceding Propositional Words Feature

Template	Description
Windows of neighbouring preceding propositional words	The preceding prepositional word of the nth word $(i - 2 \leq n \leq i + 2)$
Combinations of two immediate neighbouring preceding prepositional words	The combination of the preceding prepositional word of the nth word and the $(n+1)$th word $(i-2 \leq n \leq i+1)$
Combinations of words and preceding prepositional words	The combination of the text of the nth word and the preceding prepositional word of the nth word $(i - 1 \leq n \leq i + 1)$
Combinations of granularity levels and preceding prepositional words	The combination of the granularity level of the nth word and the preceding prepositional word of the nth word $(i - 1 \leq n \leq i + 1)$
Combinations of identifiabilities and preceding prepositional words	The combination of the identifiability of the nth word and the preceding prepositional word of the nth word $(i - 1 \leq n \leq i + 1)$

Malibu Mews is correctly identified as locative expression, hence, *TP*. The last phrase *in Chadstone* is incorrectly rejected but actually is a locative expression, thus, *FP*.

To evaluate the prediction result, precision, recall and $F_{\beta=1}$ are adopted. Precision represents the percentage that of all the predicted locative expressions how many of them actually are locative expressions. Recall stands for the percentage that of all the actual locative expressions how many of them are correctly predicted as locative expressions. $F_{\beta=1}$ is the mean of precision and recall.

They are calculated as shown in Equations 4.19, 4.20 and 4.21.

$$precision = \frac{TP}{TP + TN} \tag{4.19}$$

$$recall = \frac{TP}{TP + FP} \tag{4.20}$$

$$F_{\beta=1} = (1 + \beta^2) \times \frac{precision \times recall}{(\beta^2 \times precision) + recall} \tag{4.21}$$

If we apply Equations 4.19, 4.20 and 4.21 to the example shown in Table 4.29 the result is shown in Table 4.30.

Table 4.26: An Example of Geospatial Feature Class Feature

Word	Geospatial Feature Class
Approximate	None
halfway	None
between	None
Lara	B-S
and	None
Little	B-H
River	B-H
.	None

Table 4.27: Template Setup for 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 ($i-1 \leq n \leq i$)
Combinations geospatial feature classes and preceding prepositional words	The combination of the geospatial feature classes of the nth word and the preceding prepositional word of the nth word ($i-1 \leq n \leq i+1$)

For evaluation purposes, we employ a Perl script *conlleval*.[2]

4.3.2 10-Fold Cross-Validation

In this research, we employ 10-fold cross-validation to evaluate the performance of the learning model as 10-fold cross-validation has been proven more effective than more expensive hold-one-out cross-validation (Kohavi 1995). Specifically, we split the collected place descriptions into 10 mutually exclusive subsets of equal length. To evaluate the performance of a learning 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 perform the prediction on the held out testing subset. Lastly, the accuracy is calculated as the total number of correct predictions.

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

Table 4.28: Locative-expression-span-level Evaluation

		Actual	
		Locative Expression	Not Locative Expression
Predicted	Locative Expression	True Positive (TP, correctly identified as locative expression)	True Negative (TN, incorrectly identified as locative expression)
	Not Locative Expression	False Positive (FP, incorrectly identified as not locative expression)	False Negative (FN, correctly identified as not locative expression)

An example of 10-fold cross-validation is shown in Table 4.31. Each highlighted cell represents the held out subsets of the current iteration.

4.4 Baseline Systems

In this section, we introduce two baseline systems with which we benchmark our learning model.

4.4.1 StanfordNER

StanfordNER is used out-of-the-box as it is able to identify place references which is similar to locative expressions.

Additionally, Since the *StanfordNER* can be re-trained, it is used as a baseline system in this research. To train *StanfordNER*, we adopt 10-fold cross-validation and feed the same corpus. For feature set, we use the sample one shown on the FAQs page.[3]

4.4.2 Unlock Text

Unlock Text is adopted and used in conjunction with the set of rules described in Section 3.1.4 to re-annotate locative expressions.

[3]`http://www-nlp.stanford.edu/software/crf-faq.shtml#b`

Table 4.29: An Example of Locative-expression-span-level Evaluation

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 4.30: Evaluation Result of Table 4.29

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

Table 4.31: Example of 10-fold Cross-validation

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

Algorithm 1 Search Geospatial Feature Class

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

Chapter 5

Results

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

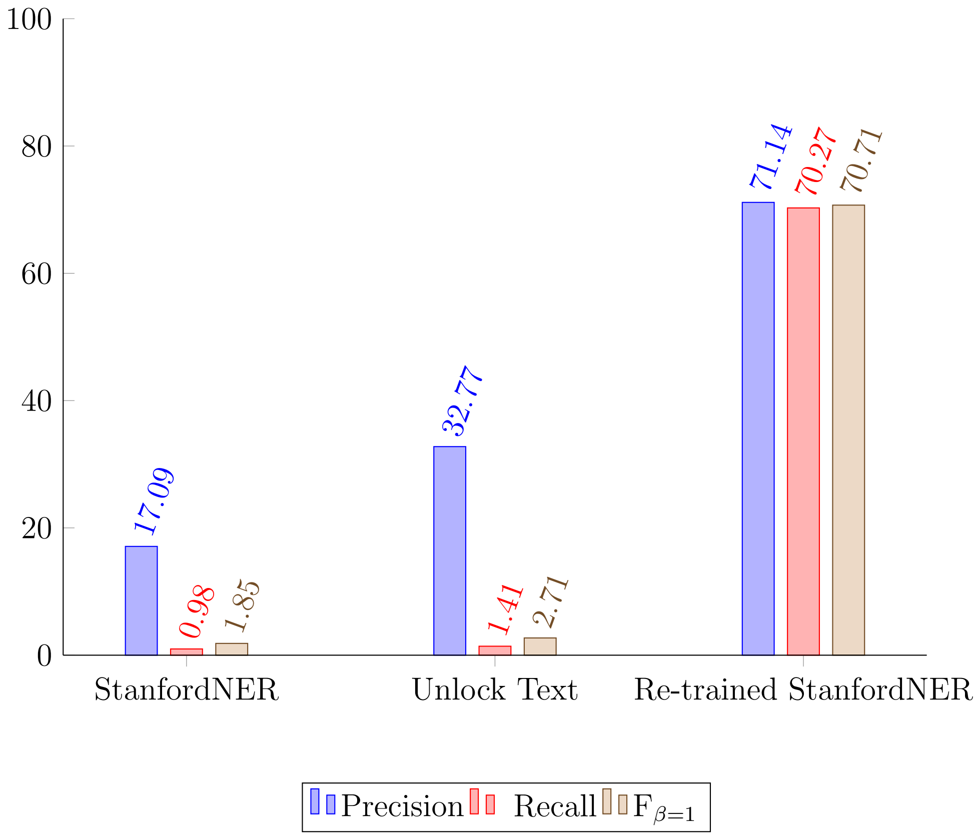
Since it is not guaranteed that every feature is contributive, we adopt *feature ablation* to verify our assumptions made in Chapter 4. In *feature ablation*, we remove one feature at a time and monitor how the performance ($F_{\beta=1}$) changes. A feature is considered unproductive if, by removing it, the performance of the learning model increases. The removing-and-monitoring process continues until no unproductive can be found.

5.1 Performance of Baseline Systems

The performance of baseline systems mentioned in Section 4.4 is presented in Figure 5.1.

As can be drawn from Figure 4.4, the precision and recall of both *StanfordNER* and *Unlock Text* are unbalanced with precision way higher than recall. It is suggested that some locative expressions identified are correct but of all the locative expressions that should be identified only a few actually are identified. Such low performance is expected 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

Figure 5.1: Performance of Baseline Systems



out of 2,139 *geospatial named entities* can be identified. Given the small percentage of place references that can be spotted, the low performance is not unexpected.

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 competitive performance of *Re-trained StanfordNER* is expected as it is re-trained on the exact same data as we use to train our leanring model. Therefore, *Re-trained StanfordNER* aims at identifying locative expressions rather than *geospatial named entities* as it was originally set out to do.

5.2 Performance of Automatic Identification Setup

The performance of our learning model is presented in Table 5.1. The highest $F_{\beta=1}$ is achieved by the setup where all features but the most frequent POS tag feature are used. The most significant drop happens when motion verb is eliminated from the

Table 5.1: Performance of Automatic Identification Setup

Automatic Identification Setup			
Feature Set	Precision	Recall	$F_{\beta=1}$
All	92.33%	88.57%	90.41
—Word	92.12%	88.65%	90.35
—POS Tag	91.54%	87.52%	89.49
—Word Position	92.24%	88.43%	90.30
—First POS Tag	92.15%	88.14%	90.10
—Most Frequent POS Tag	92.46%	88.47%	90.42
—Lexical Normalisation	92.25%	88.54%	90.36
—Chunk Tag	92.31%	88.43%	90.33
—Automatic Geospatial Feature Class (*GeoNames*)	91.83%	88.10%	89.93
—Automatic Geospatial Feature Class (*VICNAMES*)	92.04%	88.10%	90.03
—Automatic Geospatial Feature Class	91.46%	87.45%	89.41
—Locative Indictor	92.01%	88.10%	90.01
—Motion Verb	76.06%	74.79%	75.42
—Lemmatisation	91.91%	88.14%	89.98
Baseline Systems			
Feature Set	Precision	Recall	$F_{\beta=1}$
StanfordNER	17.09%	0.98%	1.85
Unlock Text	32.77%	1.41%	2.71
Re-trained StanfordNER	71.14%	70.27%	70.71

feature set.

More specifically, a comparison of the effectiveness between the motion verb feature (See Table 4.20) and the motion verb combination feature (See Table 4.21) is shown in Table 5.2. As can be seen, removing the motion verb conbination feature is the primary reason of the drop in performance of our learning model. This is not unexpected since the motion verb combination feature aims at mimicing the set of rules (See Section 3.1.4) derived from the definition of locative expression. Therefore, it is supposed to be the most decisive feature as long as the set of motion verbs is correctly collected.

To illustrate the impact of the motion verb combination feature, we adopt two examples shown in Table 5.3 and Table 5.4. The highlighted rows in both tables are incorrectly classified by the feature setup where motion verb combination feature is not used. In first example shown in Table 5.3,

Table 5.2: Motion Verb Feature vs Motion Verb Combination Feature

Automatic Identification Setup			
Feature Set	Precision	Recall	$F_{\beta=1}$
—Motion Verb Feature	92.25%	88.50%	90.34
—Motion Verb Combination Feature	76.34%	75.12%	75.72

Table 5.3: Error Analysis #1 for Motion Verb Combination Feature

Word	POS Tag	Motion Verb	Correct Label	+MVCF	–MVCF
my	PRP$	False	B-NP	B-NP	B-NP
house	NN	False	I-NP	I-NP	I-NP
is	VBZ	True	O	O	O
at	IN	False	O	O	B-NP
the	DT	False	O	O	I-NP
top	NN	False	O	O	I-NP
of	IN	False	B-NP	B-NP	I-NP
a	DT	False	I-NP	I-NP	I-NP
hill	NN	False	I-NP	I-NP	I-NP
.	.	False	O	O	O

+MVCF represents motion verb combination feature on
–MVCF represents motion verb combination feature off

5.3 Performance of Gold Standard Setup

Table 5.4: Error Analysis #2 for Motion Verb Combination Feature

Word	POS Tag	Motion Verb	Correct Label	+MVCF	–MVCF
I	PRP	False	O	O	O
am	VBP	True	O	O	O
in	IN	False	B-NP	B-NP	O
the	DT	False	I-NP	I-NP	O
lobby	NN	False	I-NP	I-NP	O
of	IN	False	I-NP	I-NP	B-NP
the	DT	False	I-NP	I-NP	I-NP
University	NNP	False	I-NP	I-NP	I-NP
House	NNP	False	I-NP	I-NP	I-NP

+MVCF represents motion verb combination feature on
–MVCF represents motion verb combination feature off

Chapter 6

Conclusion

Bibliography

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

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

Kohavi, R. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. *INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE* 14.1137 – 1145.

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

Nadeau, David, and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes* 30.3–26.

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.

Appendix A

Stuff that didn't belong in the thesis

In this Appendix, I dump a whole bunch of stuff that didn't quite fit into the thesis proper.