An Automatic Approach to Discover Lexical Semantic Differences in Varieties of English

by

Priyal Nagra

**Bachelor of Engineering and Technology in Computer Science,
Punjab Technical University, 2015**

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF**

Master of Computer Science

In the Graduate Academic Unit of your GAU

Supervisor:	Paul Cook, PhD, Faculty of Computer Science
Examining Board:	Virendrakumar C. Bhavsar, PhD, Faculty of Computer Science
Chair	David Bremner, PhD, Faculty of Computer Science
	Christine Horne, PhD, French, Department of Arts

This thesis is accepted

Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

May, 2017

Abstract

The English language is not uniform. Speakers of English in different parts of the world can use the same word, but with different meanings. Investigating lexical semantic differences in varieties of English such as American, Australian, British, Canadian is an interesting area of research in computational linguistics. We use corpora of varieties of English to detect words that changed their meaning from one variety to another. Methods of automatically identifying lexical variation used in this work are the distributional semantic models, measures of keywords, and word embedding models inspired by neural network language models. We determine whether word embedding models can detect lexical semantic differences between varieties of English better than distributional similarity approaches and approaches based on keywords. This study presents the first important step towards a robust application of word embeddings to variational linguistics. Our results indicate that word embeddings perform best among all other methods in 2 out of 3 cases.

Dedication

I'd like to dedicate my research to my parents and grand parents for supporting me throughout my life.

Acknowledgements

I'd like to thank my supervisor, Dr. Paul Cook, for his support throughout my Master's studies. Without his continuous feedback on my work, this thesis wouldn't even have been possible. He always encouraged me to contribute a good piece of work. I could not have imagined having a better supervisor and mentor for my Master's study.

Besides my supervisor, I would like to thank my thesis committee, Dr. Virendrakumar C. Bhavsar, Dr. David Bremner, Dr. Christine Horne, and Dr. Patricia Evans for their insightful comments and encouragement on my thesis greatly improved the quality of my work.

I'd like to thank my fellow lab mates, Milton King and Waseem Gharbieh for their help, cooperation, and friendship.

I'd like to thank the Natural Sciences and Engineering Research Council of Canada, the New Brunswick Innovation Foundation, and the University of New Brunswick for financially supporting this research.

Table of Contents

List of Tables

List of Figures

List of Symbols, Nomenclature or Abbreviations

AU	Australian English corpus
$NOT - AU$	non-Australian English corpus
CA	Canadian English corpus
$NOT - CA$	non-Canadian English corpus
US	American English corpus
$NOT - US$	non-American English corpus

Chapter 1

Introduction

The English language is far more than one language. The varieties of English include Canadian English, American English, Australian English, British English and many more. In the last few years, there has been interest in computational models and resources for studying similarities in language varieties and dialects. Measuring linguistic variation is one of the prominent topics in the growing field of natural language processing. One sub-field of natural language processing is concerned with quantifying linguistic differences and similarities among varieties and dialects. The task of computationally studying linguistic variation is quite hard but the increasing number of web corpora from different languages makes this type of research possible. A language like English is spoken and written differently in different parts of the world like the United States and the United Kingdom. These two countries share a common language but they use some of the same words differently;

many words are the same but some are used in different contexts and have different meanings. For example, the word *boot* in American English denotes a type of footware and a specific type of shoe, whereas the meaning of *boot* in British English can have the same meaning, but can also mean the trunk of a car. In this masters thesis, our main focus is on varietal differences, an area that is concerned with the semantic differences between varieties of the same language. Our study deals with detecting lexical variation in the meaning of the same word in varieties of English such as in the previous example with boot. Another interesting area of research in the study of varietal differences is detecting words unique to one language variety. For example, the word *toque* is unique to Canadian English and means a warm winter hat, traditionally knitted. This word is specific to Canadian English and is not widely used in other countries. Detecting unique words in language varieties is a much easier problem to solve because of their unique word forms. The notion of keywords in corpus linguistics could easily identify such unique words. Keywords (discussed in section 2.5) are defined as words that occur significantly more often in a particular corpus than in a reference corpus. Detecting lexical variation in the meaning of the same word in language varieties is a much harder problem to solve than identifying unique words because differences in meanings cannot be easily observed. Therefore, we present an automatic approach to discover lexical semantic differences of a word in varieties of English. Specifically, in this work we will determine whether recent advances in forming word embeddings inspired by neural network language models can

detect lexical semantic differences between varieties of English better than traditional count-based distributional similarity approaches and approaches based on keywords.

1.1 Objectives

Our objectives for this masters thesis are to answer the following research questions:

1) Do word embeddings inspired by neural network language models outperform traditional count-based distributional similarity models and measures of keywordness for identifying lexical semantic differences between varieties of English?

For this research question, we compare the performance of word embeddings with count-based distributional similarity models and traditional keywordness approaches to identify words that have a different meaning in one language variety than another. The measure of keywordness we consider include the measure proposed by Kilgarriff et al. 2009 [13], the chi-square statistic and log-likelihood ratio. Another way of detecting semantically similar and related words is count-based distributional semantic models. These models can capture a word's meanings from their usage in context by counting the co-occurrence of a given word with other words occurring in its context. Peirsman et al. 2010 [22] automatically detect lexical variation between Belgian Dutch and Netherlandic Dutch by using distributional models. Word

embeddings have proven to be one of the most powerful building blocks for advancements in natural language processing in recent years [9]. A word embedding model maps words from vocabulary to vectors. We used Word2Vec, a popular word embedding model given by Mikolov et al. (2013a). By representing a word by a vector, one can compute the semantic similarity between two words by obtaining the cosine similarity between their vectors. We trained Word2Vec on corpora of varieties of English to detect lexical semantic differences by comparing their word vectors based on cosine similarity (discussed in section 2.1.2).

2) Which of two approaches to applying Word2Vec to identify lexical semantic differences between varieties of English is most effective?

The Word2Vec approach is applicable to a single corpus, but to identify lexical semantic differences we need to compare vectors for 2 corpora of varieties of English. Work on identifying diachronic differences in word meaning has tried training Word2Vec on one time period, and then continued training it on another (Kim et al. 2014). Work on forming cross-lingual word embeddings has trained Word2Vec on two separate corpora of different languages, and then learned a transformation from one vector space to another. For this research question, we train Word2Vec to obtain word vectors in two different ways: first by training Word2Vec separately on one corpus, then on another corpus, and learn a transformation [29]; second by training Word2Vec on one corpus, and then continuing training on another. We compare these two different variations of Word2Vec models and examine which performs better

in identifying lexical semantic differences in varieties of English.

1.2 Contributions

This is the first attempt to apply the word embeddings paradigm to the identification of lexical semantic differences in varieties of English. We examine two variations of Word2Vec, one based on training Word2Vec separately on two corpora and learning a transformation, the other based on training Word2Vec on one corpus and then continuing training on the other. Our findings show that the latter approach performs better than the former in 2 out of 3 cases considered. We also apply several measures of keywordness based on the approach of Kilgarriff (2009), the chi-square statistic, and the log-likelihood ratio. We also apply a distributional semantic model for the task of identifying lexical semantic differences in varieties of English. Our work focuses on American, Australian, and Canadian English. For the first research question, our results are mixed with Word2Vec approaches, as well as other approaches (traditional count-based distributional similarity models and measures of keywordness), performing poorly on American English, while Word2Vec2 is best overall by a large margin in the case of Canadian English. For Australian English measures of keywordness and Word2Vec2 give good results overall.

Chapter 2

Background

In this chapter, we present the background to our work. We discuss different models used in this research thesis. Our work focuses on lexical semantic differences and similarities in language varieties of English. We discuss methods that can identify the semantic similarity of words used in a language. One of them is distributional similarity, i.e., one of the well known models that captures the meaning of a word from the context in which it occurs. Another research area discussed in this section is word embedding models, which are quite similar to distributional similarity. Both models use same kind of information, i.e., contextual information to find similarity between two words. This section also highlights computational work on lexical variation. We further study models that identify diachronic change in word meaning. Finally we describe the measures of keywordness, i.e, approaches to contrasting language corpora by comparing the frequencies of words. We

begin by discussing distributional similarity in detail.

2.1 Distributional Similarity

Distributional semantic models known as *word space* or *distributional similarity* models are generally used to represent the meaning of a word from its usage (distribution in text). A quite understandable definition of distributional similarity is that two words are distributionally similar if they appear in similar contexts. The meaning of a word can be predicted if it is repeatedly occurring in the same context of words, or with a similar set of words. Consider that you had never seen the word *gluttony*, but the following example of 3 sentences is given to you:

1. The man's gluttony caused him to gain four hundred pounds.
2. Adam uses a meal chart to avoid the temptation of gluttony.
3. An all-you-can-eat buffet encourages gluttony.

We could easily understand from these examples that, the word *gluttony* means excessive eating and drinking. We can acquire a sense of a word's meaning automatically by counting the words surrounding it. For the word *gluttony* in the example above the most common words seen are *weight* and *food*. These surrounding words are represented as a vector where each dimension represents the frequency of a word co-occurring with the target word. There are different variations of distributional similarity model: fixed

window, syntactic-based models and position-based distributional similarity models. In this work, we looked at the very commonly used fixed window distributional similarity model (which we later use in Section 4.1).

The definition of context is important in constructing a distributional similarity model (DSM). The context of a word means choosing a window n of words around a target word *w*. The context window can be seen as *-n/+n*, i.e, the n preceding words and *n* succeeding words around a target word. After choosing a window around a target word, we count the co-occurrences of context words with the target word and put them in a matrix. Each cell in the matrix represents the number of times the target word (word currently being analyzed) and the context word co-occur in a context window in a corpus. The context window considers one word at a time, until the whole corpus has been processed, this is also called a sliding context window. In distributional similarity models, the words in a corpus are represented as *vectors*. The row of a word-context matrix represents the vector of a target word. Consider a very small corpus consisting of the following (made up) sentences:

A *dog* bites a woman. The *dog* chased a little boy. Do not step on a *dog* tail. Do your *dog* bites hurt? The *cat* bites the mouse. The *cat* chased the rat. The *cat* tail looks long. She grabbed the *coffee* cup. He drinks *coffee* only. I ordered hot *coffee* and a muffin. I chipped the edge of the *tea* cup. Marta is always ready to drink *tea*. I put a cup of hot *tea* on your desk.

We show a sample distributional similarity model for four target words (dog, cat, coffee, and tea) and six context words (woman, bites, tail, drinks, chased, and cup) using a -1/+1 word context window in Table 2.1.

	woman	...	bites	tail	drinks	chased	cup
dog	0		2	1	0	1	0
cat	0		1	1	0	1	0
coffee	0		0	0	1	0	1
tea	0		0	0	1	0	1

Table 2.1: The co-occurrences of 4 sample words in a corpus with other words in their context, showing only six of the dimensions. Note that in a real application the vector of each target word would have more dimensions than shown in this example.

Table 2.1 shows that the vectors of the two words *dog* and *cat* are more similar (both *bites*, *chased* and *tail* tend to occur in their window); however, *tea* and *coffee* are more similar (because both tend to occur with *drinks* and *cup*). The count based co-occurrence matrix indicates the raw frequency of target words with the context words. The problem with raw frequency is that it is biased and not very discriminative. Words like *the*, *it*, *or*, and *of* that occur frequently in the context of many target words can give a false impression of similar vectors. Instead of just using raw frequency counts, we could add weighting to these counts. One of most commonly used weighting methods is *PPMI* or *positive point-wise mutual information*.

2.1.1 Weighting terms: Positive Point-Wise Mutual Information (PPMI)

A measure of association tells us how strongly associated two words are. *Point-wise mutual information* is one, of many, measures of strength of association. It was proposed by Church and Hanks (1989) [6], based on the notion of mutual information [12]. Given two events x and y, where x and y are the occurrence of two words, the mutual information is given as:

$$I(x, y) = log_2 \left(\frac{P(x, y)}{P(x)P(y)} \right) \quad (1)$$

We can apply this method to co-occurrence vectors of a word-context matrix by defining the *pointwise mutual information* association between a target word t and a context word c as:

$$PMI(t, c) = log_2 \left(\frac{P(t, c)}{P(t)P(c)} \right) \quad (2)$$

This equation can give positive as well as negative PMI values. A *PMI(t,c)* $= 0$ means that the particular values of t and c are statistically independent; *positive PMI* means the target word t and the context word c co-occur more frequently than expected by chance, and *negative PMI* means they co-occur less frequently than expected.

It is more relevant to use *positive PMI* values than *negative PMI*, as we are more interested in looking at words that co-occur frequently. Hence, the

negative PMI values could be replaced with 0. The positive pointwise mutual information is given as:

$$PPMI(t,c) = max\left(log_2 \frac{P(t,c)}{P(t)P(c)}, 0\right) \quad (3)$$

For example, we could compute *PPMI(t = dog, c = bites)* as follows:

$$P(t = dog, c = bites) = \frac{2}{11} = .18$$

$$P(t = dog) = \frac{4}{11} = .36$$

$$P(c = bites) = \frac{3}{11} = .27$$

$$PPMI(dog, bites) = log_2(\frac{.18}{(.36 * .27)}) = .88$$

Table 2.2 shows the *PPMI* values for each cell in Table 2.1.

	woman	.....	bites	tail	drinks	chased	cup
dog	0		0.88	0.59	0	0.59	0
cat	0		0.36	1.18	0	1.18	0
coffee	0		0	0	1.58	0	1.58
tea	0		0	0	1.58	0	1.58

Table 2.2: Replacing the counts in Table 2.1 with positive point-wise mutual information of a target word with context words

PPMI is a popular method to measure the association between two words (the context word and target word in our case). Furthermore, if we want to find how similar the two vectors are, it could be calculated by cosine similarity metric, discussed in the next section.

2.1.2 Measuring similarity: Cosine

The *cosine similarity* is a measure that calculates the cosine of the angle between vectors of two words, i.e, rows of a co-occurrence matrix such as Table 2.2. The cosine measure considers the difference in direction of two vectors in context space as opposed to the distance. Cosine is a widely used measure for vector similarity. The cosine of two word vectors can be calculated by using the following formula:

$$cosine(\overrightarrow{v}, \overrightarrow{w}) = \frac{\overrightarrow{v}.\overrightarrow{w}}{|\overrightarrow{v}||\overrightarrow{w}|} \quad (4)$$

If vectors point in same direction, the value of cosine is 1. If vectors point in an orthogonal direction, the value is 0. Vectors pointing towards opposite directions have a cosine similarity value of -1. However, if all the values of the vectors are positive (as is the case for $PPMI$), then cosine will be between 0 and 1.

Let us see from two words *coffee* or *cat* which is closer in meaning to *dog*, based on the PPMI values from Table 2.2

$$\cos(dog, cat) = \frac{0.88\times0.36+0.59\times1.18+0.59\times1.18}{(\sqrt{(0.88)^2+(0.59)^2+(0.59)^2}).(\sqrt{(0.36)^2+(1.18)^2+(1.18)^2})} = 0.82$$

$$\cos(dog, coffee) = \frac{0+0+0}{(\sqrt{(0.88)^2+(0.59)^2+(0.59)^2}).(\sqrt{(1.58)^2+(1.58)^2})} = 0$$

The *cosine similarity* value indicates that *dog* is closer to *cat* than it is to *coffee*. In this thesis, the cosine similarity measure is used in a number of experiments on different semantic similarity tasks.

2.2 Word Embeddings

Word embeddings are an unsupervised learning approach that has been successful in performing many NLP tasks and is frequently used by researchers [29]. Both word embeddings and distributional semantic models are based on the concept of representing words as vectors. Word2Vec proposed by Mikolov et al. 2013 [19] is built on a neural network architecture and it predicts words rather than using the co-occurrence counts. As discussed in section 2.1, DSM's are the traditional count-based models as they *count* co-occurrence of the target words with the context words. Neural network word embedding models in contrast are predictive models, as they try to *predict* surrounding words from the context in which they occur. The term *word embedding* was first introduced by Bengio et al. 2003 [2]. Mikolov et al. 2013 created an excellent toolkit called *Word2Vec* that enables training a word embedding model on a corpus [19]. Pennington et al. 2010 [23] released GloVe, that captures statistical information by training only on the non-zero elements in a word-context co-occurrence matrix instead of the entire sparse matrix in a large corpus [23]. We discuss the widely used skip-gram and continuous bag of words (example given in section 2.2.2).

2.2.1 Word Embedding models: Skip-gram and CBOW

We look at methods for generating word embeddings that follow the architecture of the neural network models. Generally, these models consider a

word and try to predict context words. This prediction process can be used to learn embeddings for each target word. Words with similar meanings are found close to each other in the text. The neural network models learn embeddings by taking a word vector and slowly shifting it towards a word embedding that is similar to the word embeddings of neighboring words, and dissimilar to the embeddings of words that do not occur nearby [12]. The architecture of CBOW and Skipgram is shown in Figure 2.1. We begin by discussing the skip-gram model.

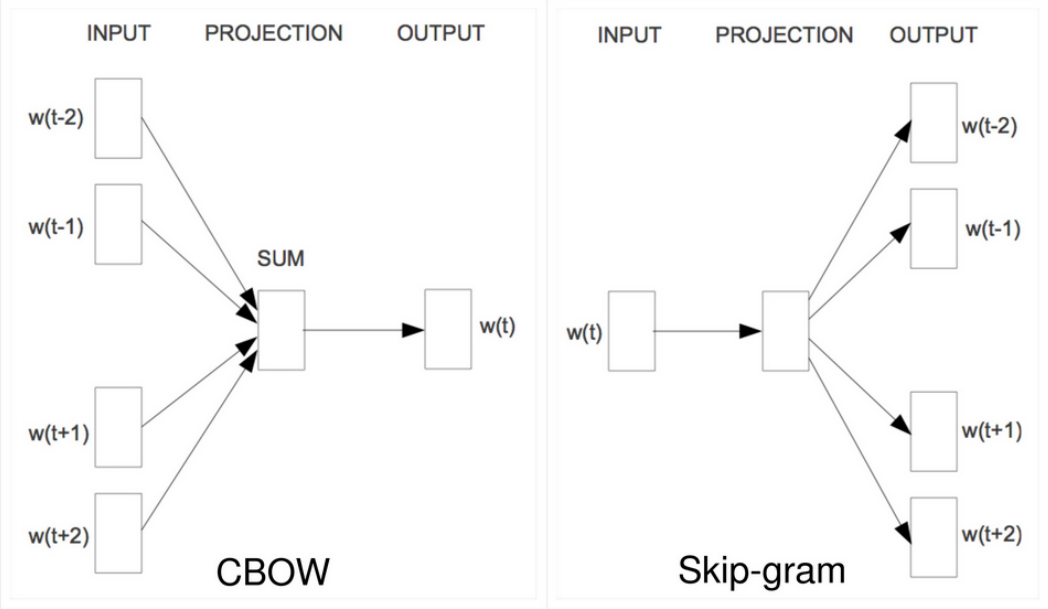

Figure 2.1: Word2Vec Architecture

2.2.1.1 Continuous Skip-gram Model

In this section, we first look at the skip-gram model inspired by neural network language models. The model architecture is shown in Figure 2.1 (right panel). In this model, the vector representation of the target word is fed at the input, the projection layer remains unchanged in the projection phase, and the model predicts the vector representation of the context words at output layer. The skip-gram model learns embeddings which are represented in two matrices, the word matrix W and the context matrix C. Each row of the word matrix W is of the dimensionality $1 \times d$ and represents a vector embedding v_i for word i in the vocabulary. Each column i of the context matrix C is of the dimensionality $d \times 1$ and represents a vector embedding c_i for word i in the vocabulary [12]. d is the dimensionality of the word embeddings and is a parameter of the model that must be set. Typically d is set between 100 and 1000.

Consider a target word w whose index in the vocabulary is j, we can call it w_j. The skip-gram model predicts neighboring words in a context window of 2N words around the current word (N is another model parameter that must be set. Typical values are 2-10). If N=2 and the current word is w_t, the model predicts these 4 words $(w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2})$. To predict w_{t+2}, whose index in the vocabulary is k, the task is to compute $P(w_k|w_j)$ where the *context vector* is w_k and the *target vector* is w_j [12].

Skip-gram computes the probability $p(w_k|w_j)$ by taking the dot product between the word vector v_j and the context vector c_k, and turning this dot

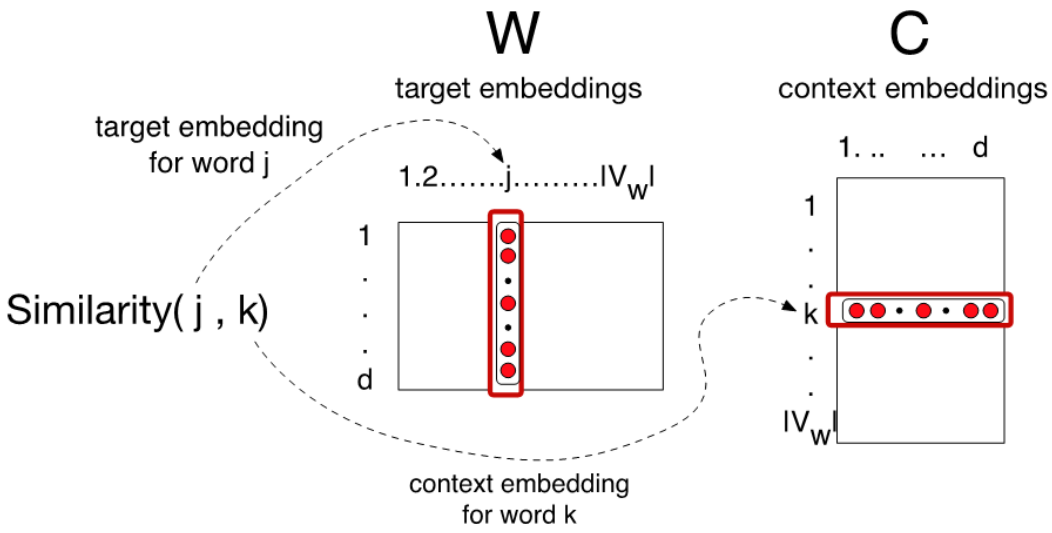

Figure 2.2: Shows that the similarity requires selecting a target vector v_j from W, and a context vector c_k from C

product $v_j \cdot c_k$ into a probability by passing it through a softmax function [12].

$$p(w_j, w_k) = \frac{exp(c_k \cdot v_j)}{\sum_{j \epsilon v} exp(c_k \cdot v_j)} \quad (5)$$

Figure 2.2 illustrates this process. In the process of learning, the model starts with initial random embeddings, and then iteratively makes the embedding of a target word more like the embeddings of neighbouring word and less like the embeddings of other words. The problem with the softmax function is that the denominator has to be computed over every word in the vocabulary. Therefore, negative sampling is used to maximise the softmax function, which involves selecting a small number of negative words (i.e., words that do not

occur in the context of the target word).

2.2.1.2 Continuous Bag-of-Words Model

In contrast to skip-gram, the continuous bag-of-words (CBOW) model predicts the target word given its context. The model architecture is shown in Figure 2.1 (left panel). Like skip-gram, it is based on a predictive model, but this time predicting the current word w_t from the context window of 2N words around it, e.g., for N=2 the context is $(w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2})$. CBOW starts with the context word vectors and sums them at the projection layer to give an output of a target word vector at the output layer [19].

2.2.2 Properties of Embeddings

A well trained set of word vectors by *Word2Vec* will place semantically similar words close to each other in the vector space. The words like *wheat*, *oats*, *rice*, *corn*, and *barley* might cluster in one corner, while *important*, *substantial*, *vital*, and *essential* might be together in another end. A visualisation of these words in the vector space projected down to two dimensions is shown in the Figure 2.3.

Furthermore, word analogies can be captured by word embeddings. For example, the vector operation $vec(king) - vec(man) + vec(woman)$ gives a vector close to that of $vec(queen)$ [19], corresponding to the analogy that man is to king as woman is to what?

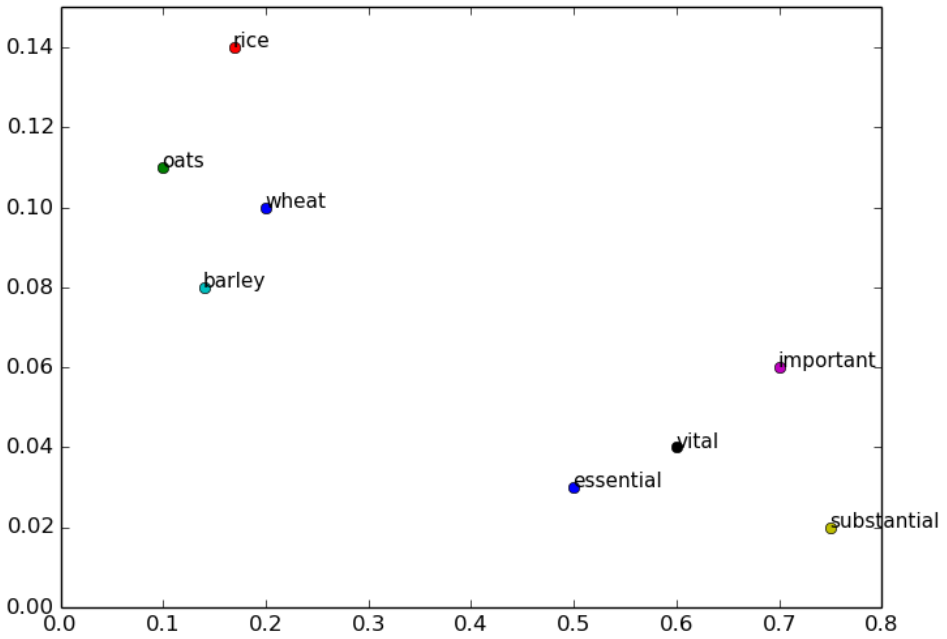

Figure 2.3: A two-dimensional visualisation of the word embedding space for a sample of words

2.3 Lexical Variation

In this section, we discuss the small number of previous studies on identifying lexical semantic differences between varieties of a language.

The study of Peirsman et al. 2010 [22] is similar to our work in this research thesis, as the authors identify differences between language varieties of Dutch (Netherlandic and Belgium Dutch) and we are trying to identify lexical variation in varieties of English. Peirsman et al. 2010 [22] address the problem

of automatic recognition of synonymy across two language varieties. The authors represent words as vectors using syntax-based and word-based distributional models. Their syntax-based distributional model looks at syntactic relations for nouns, verbs, and adjectives. The co-occurrence frequencies of their distributional models are weighted with PPMI. The authors measure similarity between context vectors using Lin's metric (Lin 1998), weighted Jaccard (Grefenstette 1994) and cosine similarity. They compile a list of 4000 words of Belgium Dutch with their synonyms in Netherlandic Dutch. Their experiment shows that distributional models are able to identify a correct Netherlandic synonym directly for about 25 percent of the Belgian words. The authors show some examples of Belgian markers whose Netherlandic synonym were correctly identified by the distributional similarity model. For example, a *microwave oven* is called *microgolf* in Belgium, but *magnetron* in the Netherlands. Similarly, the word *schoonbroer* is a typically Belgian word for *brother-in-law*, with *zwager* as its Netherlandic Dutch synonym.

Peirsman et al. 2010 [22] also address the problem of automatically identifying words that are typical of one language variety compared to another. For this task, the authors use a distributional similarity model and a frequency-based approach for identifying lexical variation between languages varieties. Their model addresses two type of markers. First, the markers that distinguish themselves in terms of higher frequency in a specific variety. Second, the markers with a different meaning in one variety, compared to another. They compared the frequencies of all words in the Belgian corpus with their

frequencies in the Netherlandic corpus, both on the basis of a chi-square (discussed in section 2.5.2) and a log-likelihood test (discussed in section 2.5.2), then sorted the words according to their *keyness* values from higher to lower. Their results show that a distributional approach was able to identify markers more successfully than the traditional frequency-based keyword approaches [22].

Tan et al. 2014 [29], use a method inspired by Mikolov et al. (2013a) [18], to learn word embeddings for a corpus of tweets and compare them to word embeddings for *Wikipedia*. To do so, they learn a transformation from one vector space to another learned from corpora of two varieties of English. The authors train Word2Vec on a twitter corpus to learn embeddings, then they trained Word2Vec on *Wikipedia* to learn word embeddings separately for that corpus. The authors demonstrated that there exists a linear transformation relationship between the vectors for the most frequent words from each corpus. Using the frequent terms from both corpora, they learned a linear projection matrix that maps the source (Twitter) to the target (Wikipedia) vector spaces.

Tan et al.(2014) [29] learn the transformation from one vector space to another by taking $a \in R^{1\times d}$ and $b \in R^{1\times d}$, where a and b are the source and target word vector representations with dimension d. The authors construct two different matrices A and B, where $A = [a_1^T, a_2^T, ..., a_v^T]$ is a source matrix and $B = [b_1^T, b_2^T, ..., b_v^T]$ is a target matrix, where T is the transpose, composed of vector pairs $\{a_i, b_i\}_{i=1}^{v}$, where v is the size of the vocabulary in common

between the source and target corpora. The authors order these vectors according to the frequency in the target corpus such that a_i and b_i are the i-th most common word in the target corpus. The authors used these vectors to learn the transformation matrix $M \in R^{d \times d}$. After the transformation matrix M is obtained, they transform a_i to $a_i^{'} = a_i M$ in order to approximate b_i. The transformation is represented by Lan et al. 2014 [29] as:

$$AM = B \quad (6)$$

The transformation matrix M can be approximately computed by using stochastic gradient descent:

$$min_M \sum_{i=1}^{n} \parallel a_i M - b_i \parallel^2 \quad (7)$$

where, the training process is limited to the top n terms.

Once the transformation matrix M is generated, the authors find the cosine similarity metric between $\left(a_i^{'}, b_i\right)$ to determine the extent to which the meaning of these words differs between the two corpora. Tan et al. (2014) [29] used the 1000 top most frequent common words from a Twitter and Wikipedia corpus to learn the transformation matrix M. The authors suggested that the transformation can be either from Twitter to Wikipedia or in the opposite direction, but the results in both cases were similar. The authors found interesting differences in the meanings of some word, for example, *bc* means roughly *because* in Twitter but *bce* (Before Common Era) in Wikipedia, and

ill means *ll* or *will* (as a non-standard spelling of I'll) in Twitter and means *unwell* or *sick* in Wikipedia.

2.4 Diachronic change

Another area of interest for researchers is detecting changes in language across time. *Diachronic change* is defined as a change of the meaning of a word over time. In the past century, word meaning and usages have changed in many languages. For example, the English word *silly* has changed its meaning from *worthy* or *blessed* to *foolish*. The study of diachronic semantic change is an area of interest to historical linguistics. With the changes in the meaning of words lexicons and dictionaries must be updated too. Developing automatic methods for identifying changes in word meaning can, therefore, be useful for both theoretical linguistics and a variety of NLP applications which depend on lexical information. Various researchers have studied shifts in word meaning and approaches to detecting novel word senses. We look at approaches used by the researchers in detecting specific types of semantic change, and semantic change more generally, in the following sub-sections, and discuss how these approaches are related to the methods used in this thesis.

2.4.1 Models that capture specific types of semantic change

Methods for semantic change detection were introduced by Sagi et al. (2011) and Cook and Stevenson (2010) [8] focusing on specific types of semantic change. Cook and Stevenson (2010) [8] consider two types of semantic changes, amelioration and pejoration, which focus on a word changing to have a more positive or negative meaning, respectively. The authors used three British English corpora from different time periods to determine the semantic orientation of words in each time period and used this to identify ameliorations and pejorations. The semantic orientation of a word is the extent to which the word has a positive or negative meaning. The authors determined the semantic orientation of a word by computing its association using a PMI-based method with positive and negative sets of seed words. They select as seed words only those words which have either positive or negative senses, for example, positive seeds include good, nice, and excellent and negative seeds include bad, nasty, and poor. The authors constructed artificial words to evaluate the ability of their model to identify ameliorations and pejorations in any given pair of corpora. We also construct some pseudo words (artificial words) for evaluation (see Section 5.2).

In earlier work, Sagi et al. 2011 [25] use latent semantic analysis (LSA) distributed representation to identify semantic change across time. LSA builds a representations of a word based on its co-occurrence patterns with other

words in a corpus similarly to a distributional similarity model. The authors examine two of the traditional categories of semantic change: narrowing and broadening. In the case of narrowing, a general meaning of a word becomes more specific, whereas, in broadening, a restricted meaning of a word becomes less restricted. They looked at some examples, such as how in old English, the word *deer* meant any animal, whereas, in modern English, the word *deer* is used for an animal family. The authors demonstrated that narrowing and broadening can be captured through LSA.

Xu et al. 2015 [30] propose computational models to study two laws of semantic change: the law of differentiation that near-synonyms of a word change their meaning over time, and the law of parallel change that states that related words undergo a parallel change in meaning. They studied these laws by using the Google Million corpus (Michel et al. 2011 [17]). The authors use a distributional similarity model to capture the meaning of a word during a given decade by representing the word as a vector. To compute the similarity between two vectors, they use Jensen-Shannon divergence. The authors identify semantic change for a word by comparing the word's top nearest neighbors in the 1890s and again in the 1990s from the Google Million corpus [17]. Their results show that synonyms of a word may become more different in meaning over time, and that parallel change is more common than differentiation for these periods of time [30].

More recently, Hamilton et al. 2016 [11] focus on capturing semantic change over time by using word embeddings. The authors suggest that there is a

relationship between semantic change and polysemy. They use word embedding models for capturing semantic change over time and compare them with more conventional distributional similarity approaches. Their analysis is based on data from 4 different languages: English, German, French, and Chinese. They propose two laws of semantic change: the law of conformity (frequent words change more slowly) and the law of innovation (polysemous words change more quickly). They capture the semantic change in two different ways: first by comparing pair-wise word similarities over time, and second by capturing semantic shift in word embeddings over time. Their analysis shows that polysemous words tend to change their meaning faster than the higher frequency words in corpora.

2.4.2 More-general models of semantic change

Distributional similarity models are used by many past researchers in the detection of semantic change of words. Gulordava and Baroni 2011 [10] also capture semantic change using distributional similarity models. The authors compare a frequency-based technique with the distributional similarity approaches for detection of the semantic change of words. They use the Google Books N-gram data (Mitchel et al. 2010) for 7 different languages from the 1960s and 1990s for evaluation. The authors compute cosine similarity between word vectors. The authors compute distributional similarity for the words in the 1960s and 1990s corpora. To evaluate their methods, the authors create a reference categorization using human judgments for a set of

randomly selected mid-frequency words. The human raters annotated the words as almost no change, somewhat changed, or changed significantly. According to the human raters, words like *sleep* and *parent* did not change at all, whereas words like *virus* and *virtual* acquired a new sense across time. The authors took the average of the human judgments as a reference value to compare with distributional similarity scores. Their results show that the distributional similarity-based measure produces good results for the words that are popular nowadays, while the frequency-based measure performs better for the words popular in the 1960s [10].

Cook et al. 2014 [7] introduce different methods to automatically identify new word senses. The authors use two different corpora— a reference corpus in which the new word senses are not seen, and a focus corpus in which the new word senses appear. In this paper, the authors propose a word sense induction (WSI) method to detect novel senses of a word. The authors extract usages of a target word from both corpora, and then apply a WSI system to the instances of the target word. The WSI model is based on a Hierarchical Dirichlet Process. The WSI model induces the senses of a target word w from a given set of usages of w. HDP is run on those usages to induce topics; these topics are then interpreted as representing the senses of w (one topic per sense). The authors compute $Novelty_{ratio}$ (which is the ratio of the relative frequency of an induced sense in the focus and reference corpora), $Novelty_{diff}$ (based on the difference in relative frequency of an induced sense in the focus and reference corpus), $Novelty_{LLR}$ (based on the log-likelihood

ratio of an induced sense in the two corpora) and $Relevence_{manual}$ (based on the association of an induced sense with manually identified keywords for the corpora) measures. The authors use the relevance and novelty measures to rank each sense of each word, and then computed the rank sums based on the novelty and relevance. They rank words with a novel sense, and distractors (words that haven't changed in meaning) by the various Novelty, Relevance and Rank Sum methods and show that these methods are able to identify senses that are in the focus corpus but not in reference corpus.

To detect changes in language across time, Kim et al. (2014) [14] trained Word2Vec on corpora of Google Books N-grams to obtain word vectors from 1900 to 2009. The authors consecutively trained the model by initializing word vectors for subsequent years with the word vectors obtained from previous years. The authors identified words that changed significantly from 1900 to 2009 by comparing the vectors for the same word in different years based on cosine similarity. The authors show that the words *gay* and *cell* have changed their meaning from 1900 to 2000 [14]. We use a method based on the same approach to training Word2Vec used by Kim et al. (2014) to find varietal differences using Word2Vec.

Kulkarni et al. (2015) [15] used a similar approach based on Word2Vec to capture shifts in the meaning of words such as *gay* over the last century. The authors use corpora of movie reviews from Amazon and the Google Book N-grams. The authors propose three different methods: distributional similarity models to detect words that shifted in meaning over a time period;

a deep neural language model trained on a corpus for each time period from which they then learned a linear transformation from one vector space to another, which is similar to the approach used by Tan et al. 2014 [29]; and frequency based methods that can capture linguistic shift, as changes in frequency which can correspond to words acquiring or losing senses.

2.5 Keywordness

Keywords are words that are much more frequent to one corpus than another. There has been a large amount of research done to define various ways in which keywords of corpora can be identified. In this section, we discuss various measures for *keywordness*.

2.5.1 Pearson's chi-square statistics

The χ^2 statistic is a test for dependence which assumes the occurrences of a word are randomly distributed in corpora and has been widely used for identifying keywords [16]. The chi-square test can be applied to the simplest 2×2 contingency table shown in Table 2.3. It compares the observed frequencies in the corpora with the frequencies expected for independence. The value c corresponds to the number of words in corpus C1, and d corresponds to the number of words in corpus C2. The values a and b are called the observed values (O) and represent the frequency of word w in corpus C1 and C2. The chi-square statistic is computed by the following equation:

	C1	C2	
w	a	b	a+b
not w	c-a	d-b	c+d-a-b
	c	d	c+d

Table 2.3: Contingency table

$$\chi^2 = \sum_i \frac{(O_i - E_i)^2}{E_i} \quad (8)$$

where O_i is the observed frequency in the table, and E_i is the expected frequency in each cell [16]. The expected frequency for each cell is equal to the row total times the column total divided by the total sum for all cells. For example, the expected frequency of cell E_{11} in Table 2.3 is $\frac{(a+b)c}{c+d}$.
We can assign a positive and negative polarity to the signed root chi-square values of words in one corpus vs. the other using the method given below:

$$\sqrt{\chi^2}\,(polarity) = \begin{cases} positive & if \frac{a}{a+c} > \frac{b}{b+d} \\ \\ negative & if \frac{a}{a+c} < \frac{b}{b+d} \end{cases} \quad (9)$$

2.5.2 Log Likelihood Ratio

Log Likelihood ratio (LLR) is another approach to finding keywords of one corpus vs another. This approach is similar to *Chi-square*, as it uses the same contingency table to calculate the log-likelihood ratio (LLR) scores [16].
We can calculate the log-likelihood values according to this formula [16]:

$$-2 \ \ln \ \lambda = 2 \sum_i O_i \ \ln \left(\frac{O_i}{E_i} \right) \quad (10)$$

Similarly, we can also assign the signed root LLR values with negative or positive polarity as assigned to chi-square values as in Equation 9.

2.5.3 Relative Frequencies

Kilgarriff (2009) [13] presents a simple method for identifying keywords based on *relative frequencies* in one corpus vs. another. This method includes a parameter which allows focusing either on higher, or lower, frequency words. The keywordness score of a word is calculated according to the following formula:

$$\frac{fpm_{focus} + \alpha}{fpm_{reference} + \alpha} \quad (11)$$

where fpm_{focus} is the frequency (per million) of the word in the focus corpus, $fpm_{reference}$ is the frequency (per million) of the word in the reference corpus, α is the smoothing parameter. The α parameter could be any value 1, 10, 100, 1000 or more [13]. Kilgarriff's keywordness approach is very simple to calculate and easily interpretable. Higher values of α emphasize higher frequency keywords, while lower values of α emphasize lower frequency keywords.

Chapter 3

Corpora

In this chapter, we describe the corpora representing national varieties of English that we use in this study. The corpora used in our study are based on ENCOW14AX created by Schafer and Bildhauer (2014). ENCOW14AX is an English corpus crawled from the web in the years 2012 and 2014. Schafer et al. 2014 [26] introduce the COW14 (Corpora from the Web) web corpus creation and query architecture which was used to built the corpus. COW14 chains of wrapped annotation tools are available for Dutch, English, French, German, Spanish, and Swedish. We are only using the English web corpus of the COW14 chain.

The crawler used to build ENCOW14AX is the Heritrix web crawler. The authors collected data for 6 different languages (Dutch, English, French, German, Spanish, and Swedish) using the crawler. The authors used the rule-based Ucto tokenizer for tokenization and heuristic sentence splitting.

The pre-processing of corpora includes a tool texrex that performs HTML stripping, crawler and HTML metadata extraction, boilerplate detection, in-document paragraph deduplication, language detection, text quality assessment, near-duplicate document detection, conversion to UTF-8, some UTF-8 normalizations, and geolocation lookup based on server IP addresses [28]. This geolocation information is crucial for our work because it allows us to create sub-corpora from ENCOW14AX corresponding to specific countries. The authors used the TreeTagger (Schmid 1995 [27]) for part-of-speech tagging and chunking, dependency-parsed the corpus with the Malt Parser (Nivre et al. 2007 [21]), and morphological analysis is performed using Mate tools (Bjrkelund et al. 2010 [4]). The authors used the Map-Reduce framework Hadoop to efficiently create the corpora.

ENCOW14AX is available in the Colibri web interface. ENCOW14AX can also be accessed and downloaded from `webcorpora.org` (which requires registration but is available for research purposes). We registered at `webcorpora.org` to obtain the access and then downloaded the ENCOW14AX corpora. The ENCOW14AX corpus consists of 9,578,828,861 tokens and 425,374,806 sentences, making it one of the largest English corpora available.

ENCOW14AX is a very large web corpus which includes data crawled from 66 different countries based on the provided geolocation metadata. We do not use the corpus for all 66 countries provided in ENCOW14AX. For many countries the amount of data is insufficient for research on lexical semantic variation. Furthermore, as discussed in Section 1.1, we evaluate our methods

on their ability to identify Americanisms, Australianisms, and Canadianisms, i.e., words that have specialized meanings in the United States, Australia, and Canada respectively.

We therefore built subcorpora for Canadian English, American English, and Australian English from ENCOW14AX based on the country-level meta data provided. For Australia and Canada, we use all the data available. However, for the United States, we use a random sample of one billion tokens to keep the corpus size manageable. After building these subcorpora, we combined these subcorpora to build three new subcorpora, each representing NOT one of these varieties of English, to compare against. We construct a corpus of roughly 1 billion words (because bigger is better for building distributional similarity models, but this restriction keeps the corpus sizes manageable) for NOT each variety of English. Therefore, we built new subcorpora in such a way that the one language variety corpus could be compared with another non-language variety corpus, e.g., the Canadian English corpus is compared with non-Canadian English corpus, and the Australian English corpus is compared with non-Australian English corpus. We built an additional corpus for British English, and then combined the subcorpora for Canadian English, American English, British English and Australian English with each other such that we build new subcorpora for non-Canadian English, non-American English, and non-Australian English.

We construct these corpora to include as much data from each variety as possible, up to a total of roughly one billion tokens. The composition of these

corpora is shown in Table 3.1. We use the following terminology for these six corpora: American English: US, Canadian English: CA, Australian English: AU, non-American English: NOT-US, non-Canadian English: NOT-CA, non-Australian English: NOT-AU.

Language Variety	Tokens	#Tokens from each variety			
		AU	CA	GB	US
AU	57M	57M	0	0	0
NOT-AU	1B	0	219M	500M	500M
CA	219M	0	219M	0	0
NOT-CA	1B	57M	0	500M	500M
US	1B	0	0	0	1B
NOT-US	1B	57M	219M	750M	0

Table 3.1: Corpora size in terms of number of tokens and composition of each corpus

In our experiments, we compare the Australian English corpus AU with the non-Australian English corpora NOT-AU, the US corpus is compared with the NOT-US corpus, and CA is compared with NOT-CA.

Chapter 4

Models for identifying lexical semantic variation

Our work focuses on identifying lexical semantic differences in language varieties of English. We studied various approaches used in the past that try to capture lexical variation in Chapter 2. We studied various models such as *distributional similarity model*, *word-embedding model*, and *the methods of keywordness*. In this chapter, we propose methods based on each of these models to identify lexical semantic differences between varieties of English. First, we use count-based distributional similarity models to capture semantically different words based on the differences between their contexts. Next, we created word vectors using neural network inspired word embedding models to derive semantic differences between words. Finally, we explain how approaches to identifying keywords can be applied to identify words specific

to one corpus compared to another based on word frequencies, where words with very different frequencies can correspond to lexical semantic differences.

4.1 Distributional Similarity Models

We construct a traditional count-based distributional similarity model in which the meaning of a word can be represented from the context in which it occurs. There are different types of distributional similarity models: fixed window, syntactic-based models, and position-based distributional similarity models (see section 2.1). For this work, we focus on using the *fixed-window count based distributional similarity model* (DSM) rather than using syntactic-based or position-based DSMs. If we use syntactic-based models, then we have to use a parser which could give different results for different language varieties. Furthermore, the positional based models are more sparse than fixed window DSMs. We therefore used a typical DSM based on a fixed-window.

The distributional similarity models can be used to identify semantically similar words between a pair of corpora. We use the same technique on a pair of corpora of different language varieties of English to identify in particular semantically different words across varieties. DSM is built separately for each of two corpora, and then to determine how different a particular word is in one corpus vs. the other, we calculate the cosine for the vectors for that word from the DSM's for the two corpora.

Model We implement a count-based distributional similarity model as discussed in section 2.1 with *fixed-context window size* of -2/+2 words around the target word. The count-based distributional model takes into account each word in the corpus as a target word. The -2/+2 context window size is a common setting used in constructing a DSM.

After choosing a window -2/+2 words around a target word w, we count the co-occurrences of context words with the target word and put them in a *co-occurrence matrix*. Each cell in the matrix represents the number of times the target word (word currently being analyzed) and the context word co-occur in a corpus. The co-occurrence frequencies in the co-occurrence matrix are weighted with positive point-wise mutual information (as discussed in section 2.1.1) because positive point-wise mutual information works well for measuring semantic similarity in the co-occurrence matrix [12].

Stopwords There are some extremely common words in any corpus which provide very little information to a DSM, therefore, we exclude them from the vocabulary entirely. These words are called *stop words*. Some of the most common English stop words are *the*, *is*, *at*, *which*, *on* etc. We are not interested in these words, as these words are not helpful in detecting the semantic similarity between two words. For instance, the co-occurrence count of a target word with *the* in a distributional similarity model does not provide us much information. We obtained a list of stop words [1] and then we removed the occurrences of stop words from each corpus.

[1] `https://sites.google.com/site/kevinbouge/stopwords-lists`

The corpora then contain only the important words from which we construct a count-based distributional similarity model. After removing stop words, all other words are considered as context words in constructing the model.

Corpora We used six corpora; AU, NOT-AU, CA, NOT-CA, US, and NOT-US (see Chapter 3). We construct one DSM for each of these six corpora.

Similarity We use the resulting co-occurrence matrix, after re-weighting the frequencies using PPMI, for computing similarity. The similarity measure is the cosine similarity (discussed in section 2.1.2). We built a DSM separately for each of two corpora, then we calculate the cosine between the vectors of a word obtained from the DSMs of the two corpora to determine the extent to which that word's meaning is the same or different in the two corpora.

Requirements In the count-based distributional similarity model, we construct a co-occurrence matrix that stores the count of target words with the context words. The target word and the context words are treated as tokens in the corpus, we count the co-occurrence of the tokens with each other and store the counts in the cell of the co-occurrence matrix. We consider every word in corpora that contain up to roughly one billion tokens as a target word, hence the co-occurrence matrix formed is very large matrix. Suppose, there are V target words and also V context words, the matrix formed will be of dimensionality $V \times V$. For a corpus of one billion words, the matrix formed is very large. It requires a lot of memory to store such a large matrix. We used ACENET (Advanced Computational Excellence Network) resources that provide advanced computing resources to tackle difficulties

in computational research. ACENET provides computing infrastructure to solve problems that are either too large or too complex for traditional desktop computers. ACENET resources resolved our memory and disk issues to compute and store the $V \times V$ co-occurrence matrix.

4.2 Word2vec

Word2vec is an efficient technique for building a mapping from words in a vocabulary into vectors of real numbers based on a corpus. We discussed why word embeddings are useful and discussed efficient training techniques in section 2.2. The vector representation of words learned by Word2vec has been shown to carry semantic meanings of words [19]. Vector representations of words can then be used to find similarity between words. A Word2vec implementation is provided by the Gensim Python library [2]; it is an efficient implementation of the continuous bag-of-words and skip-gram architectures for computing vector representations of words. Given a text corpus, Word2vec learns a vector for every word in the vocabulary by using the Continuous Bag-of-Words or the Skip-Gram architectures. We must specify some parameters to train Word2vec on our corpus to learn embeddings. The parameters such as the desired vector dimensionality, the size of the context window, the model choice of Skip-Gram or Continuous Bag-of-Words, and the training algorithm: hierachical softmax (an efficient approximation to

[2] `https://radimrehurek.com/gensim/`

softmax) or negative sampling (which speeds up training using "negative", i.e., non-context words, as described in Section (2.2.2).

Parameters We specified the parameters to train Word2Vec. We choose the Continuous Bag-of-Words architecture and used negative sampling as the training algorithm. Both of these are default settings. We initialized the model with a list of iterable sentences. We fed the corpus to Word2Vec in a format where each sentence of the corpus ends with a period or full stop and is represented in a list. We choose a parameter $size = 200$, which is the dimensionality of the embedding vectors. We choose a $window = 5$, which depicts the maximum distance between the current and predicted word within a sentence. The frequency cut-off for our model is set to 50 which means all words with a total frequency lower than 50 are ignored during training. We choose a setting of $workers = 4$ which means this many worker threads are used to train the model (faster training with multicore machines). All the settings used to train Word2Vec are default settings.

Methods We implement two different methods of training Word2Vec on a pair of corpora discussed below:

Word2Vec1 In the first method, we train Word2Vec on one corpus (e.g., Canadian English corpus CA) to learn word embeddings for each word in the corpus, and then train Word2Vec on another corpus separately (e.g., non-Canadian English corpus NOT-CA). After that, we learned a linear transformation relationship between the vectors using the 1000 most frequent words from both corpora using the approach of Tan et

al. (2015) (discussed in section 2.3). Then, we compare word vectors from one vector space to another for the same word under this transformation.

Word2Vec2 We consider a second approach to training word2vec for a pair of corpora such that learning a transformation is not necessary. In this second approach, we train Word2Vec on one corpus representing not some specific variety of English e.g., NOT-CA, and then continue training on the other corpus, e.g., CA, to learn a single embedding space for all words in both corpora. We do this because in preliminary experiments we considered training on the corpora in the opposite order, but found this to perform relatively poorly. This approach is inspired by the method of Kim et al.(2014) for training word2vec on corpora from multiple time periods to detect lexical semantic change over time (discussed in section 2.4.2). For example, first we train Word2Vec on the non-Canadian English (NOT-CA) corpus and learn the word embedding space for all words in this corpus, then we start from the previously learned word embedding space and continue training on the Canadian English (CA) corpus to learn a set of embeddings for this corpus. This method is quite interesting because it is simple in that it does not require the step of learning a transformation between two vector spaces.

Requirements We used ACENET resources to train our word2vec models

because it requires a lot of memory to train word2vec on corpus of roughly one billion words.

Similarity Computing similarity between Word2Vec models is exactly the same as it is for the DSMs, except that the vectors are from Word2Vec instead of a DSM. After obtaining vectors of words, we calculate cosine similarity for a word based on its embeddings from the two corpora.

4.3 Keyword-Based Methods

We consider methods for identifying keywords of one corpus vs. another as baselines against which the other approaches are compared. In particular, if a word has a different meaning in one corpus compared to another, it might also have a very different frequency and could potentially be identified using a measure of keywordness. We used the Kilgarriff (2009) [13] method for identifying keywords based on the ratio of a word's frequency per million, plus a constant α, in two corpora. The advantage of the α parameter is that it allows us to specify whether we want to focus on higher-frequency or lower-frequency keywords, as discussed in section 2.5.3. We set the constant α to 1, 10, and 100. Hence, we have three different cases for each variety of English, i.e., Add 1, Add 10, and Add 100.

We compute the chi-square statistic using Python SciPy to obtain chi-square values for the words in one corpus vs. the other corpus (e.g., AU vs. NOT-AU). We assign a *positive* and *negative* polarity to the signed root chi-square

values of words as discussed in section 2.5.1.

We similarly compute log-likelihood ratios for the words in each pair of corpora using the Python SciPy log-likelihood ratio (LLR) test. We again assign a positive and negative polarity to the signed root LLR values of words as discussed in section 2.5.2.

Chapter 5

Evaluation Resources and Methods

In this chapter, we discuss the resources required to evaluate our approaches to identifying lexical semantic variation based on known regionalisms and pseudo words, and the evaluation measure, the receiver operating characteristic curve, used to compare approaches.

5.1 Regionalisms Evaluation

To evaluate our methods for identifying lexical semantic variation, we require lists of words known to have meanings that are particular to a variety of English. We refer to these words as *Regionalisms*. We use lists of regionalisms of American English, Australian English, and Canadian English from

dictionaries of these varieties of English as described below:

American English We obtained the list of words documented in the online version of the Dictionary of American Regional English (DARE, Hall 2012) [5]. DARE contains words that have a usage that is particular to a region within the United States.

Australian English We obtained the word list from the online version of the Australian National Dictionary. It documents terms that are common and specific to Australian English (AND, Ramson 1988) [24].

Canadian English We used a list of all words marked as Canadian from The Canadian Oxford Dictionary (CanOx, Barber 2005) [1].

We extracted all single word entries from each above word list ignoring multiword expressions. We then generate a list of regionalisms from these dictionaries for evaluation purposes. The regionalisms are randomly selected from words present in the corpora and in the dictionary word lists. We compare regionalisms with a set of distractors (non-regionalisms) chosen randomly from corpora that are not in the dictionary word lists to measure the ability of our models to detect lexical semantic differences. We set criteria for selecting the regionalisms and distractors from the corpus of language varieties and distractors as follows:

Selection of regionalisms from Corpus 1 and Corpus 2

· Start from a dictionary-based wordlist.

· Remove multiword expressions, retain only single words entries.

· Retain words with raw frequency greater than 50 in Corpus 1 and Corpus 2.

· Retain words present in the NLTK (Bird et al. 2009) [3] English word list.

· Randomly select a sample up to 100 words from this list of words.

The DSM and Word2Vec models represent single words as vectors. We could use the same technique in the future to identify semantic differences among multiword expressions by concatenating the multiwords in the corpora before feeding them into the models. In this work, however, we only looked at single word entries and their vectors. For low frequency words, the vectors obtained by the DSM and Word2Vec provide poor representations of the meaning of words, and so cannot be used to identify lexical semantic variation. Therefore, the frequency cutoff for selecting regionalisms for evaluation is set to 50. The set of distractors are selected using a similar process as for regionalisms, described below, but these words are not regionalisms.

Selection of distractors from Corpus 1 and Corpus 2

· Start from NLTK [3] Wordlist.

· Remove multiword expressions, retain only single word entries.

· Retain words with raw frequency greater than 50 in Corpus 1 and Corpus 2.

· Remove words present in the regionalisms list.

· Randomly select a sample upto 100 words from this list of words.

5.2 Pseudo Words-Based Evaluation

We also generate a list of artificial words called *pseudo words* for evaluation purposes. These pseudo words are artificial examples of semantic change that simulate a word having entirely different meanings in two varieties of English. To create a pseudo word, we select a word from one language variety corpus and then select a different word from another language variety corpus. These two different words have different meanings. We compare the pseudo words with regular words chosen randomly from the corpora to detect (artificial) lexical semantic differences. If we consider a word in one corpus and then in another corpus, this word would be expected to have the same meaning in both corpora. These words are referred to *regular words*. For example, if we compare the pseudo word *waste* in one corpus with the pseudo word *swimming* in another corpus, these words have very different meanings. On the other hand, the regular word *enquiry* is expected to have roughly the same meaning in the two corpora. We set criteria for selecting the regular and pseudo words from the corpora of language varieties. We use a large lexical database of English known as Wordnet [20] where nouns, verbs, adjectives and adverbs are grouped into sets of synonyms. We obtain

nouns from Wordnet to form a list of words for evaluation. The following criteria show how we chose regular and pseudo words:

Selection of pseudo words from Corpus 1

· We start from all words occurring in Corpus 1.

· Choose words with the raw frequency greater than 50 and less than 1000 in Corpus 1.

· Choose words which are nouns from Wordnet (ignoring verbs, adjectives, and adverbs).

· Randomly select a sample up to 100 words from this list of words.

Selection of pseudo words from Corpus 2

· We start from all words occurring in Corpus 2.

· Choose words with the raw frequency greater than 50 and less than 1000 in Corpus 2.

· Choose words which are nouns from Wordnet (ignoring verbs, adjectives, and adverbs).

· Randomly select a sample up to 100 words from this list of words.

Selection of regular words from Corpus 1 and Corpus 2

· We start from all words occurring in both Corpus 1 and Corpus 2.

· Words must have the raw frequency greater than 50 and less than 1000 in Corpus 1 and Corpus 2.

· Words must be present in the NLTK English word list.

· Choose words which are nouns from Wordnet (ignoring verbs, adjectives, and adverbs).

· Randomly select a sample upto 100 words from this list of words.

The reason for choosing a frequency cutoff greater than 50 and single word entries is the same as for the case of distractors or regionalisms. Here, however, the upper-bound on frequency provides greater control over the frequency of these artificial examples of semantic change.

5.3 Receiver Operating Characteristic (ROC) Curves

We run our experimental setup on eight models— one *distributional similarity model*, five *measures of keywordness* and two *Word2Vec* models to detect *regionalisms* in the language variety corpus against the *distractors*. We calculate the cosine similarity scores for DSM, Word2Vec1, Word2Vec2, and keywordness scores for Kilgarriff, Chi-square, and LLR for the selected

regionalisms and distractors. In these experiments, we rank all items, regionalisms and distractors, by the various approaches for each pair of corpora (CA with NOT-CA, AU with NOT-AU, and US with NOT-US corpus). For a pair of a corpora, we label the selected regionalisms and distractors as 1 and 0 respectively. We labeled all regionalisms and distractors. For keyword-based approaches, we sort the regionalisms and distractors from higher to lower keywordness values. For the distributional similarity and Word2Vec models, we sort cosine values from lower to higher. The idea is that regionalisms should be marked at the top of the list, and distractors at the bottom of the list, for an ideal result. We draw receiver operating characteristic (ROC) curves to show how accurately a model marks the regionalisms and distractors for a given pair of a corpora. The ROC curves are used to study the output quality of a classifier and show true positive rate on the Y axis, and false positive rate on the X axis.

System	Gold Standard		
		Regionalisms	Distractors
	Regionalisms	a	b
	Distractors	c	d

Table 5.1: Contingency matrix for computing true positive rate and false positive rate

The true positive rate (TPR) and false positive rate (FPR) are computed as:

$$TPR = \frac{a}{a+c} \quad (14)$$

$$FPR = \frac{b}{b+d} \quad (15)$$

At each point in the ranking list, the system computes the true positive and false positive rate for distractors and regionalisms. Consider the example in shown in Table 5.2 which has 3 regionalisms and 3 distractors ranked as follows::

0	
1	r
2	r
3	d
4	r
5	d
6	d

Table 5.2: Example ranking for generating an ROC curve

The computed true positive rate and false positive at each point in ranking are shown below:

Point	tpr	fpr
0	0	0
1	1/3	0
2	2/3	0
3	2/3	1/3
4	1	1/3
5	1	2/3
6	1	1

Table 5.3: The true positive rate and false positive rate at every point in the ranking

The ROC curve for the above example is shown in Figure 5.2. The area under curve of receiver operating characteristic (ROC) curves indicates how well a

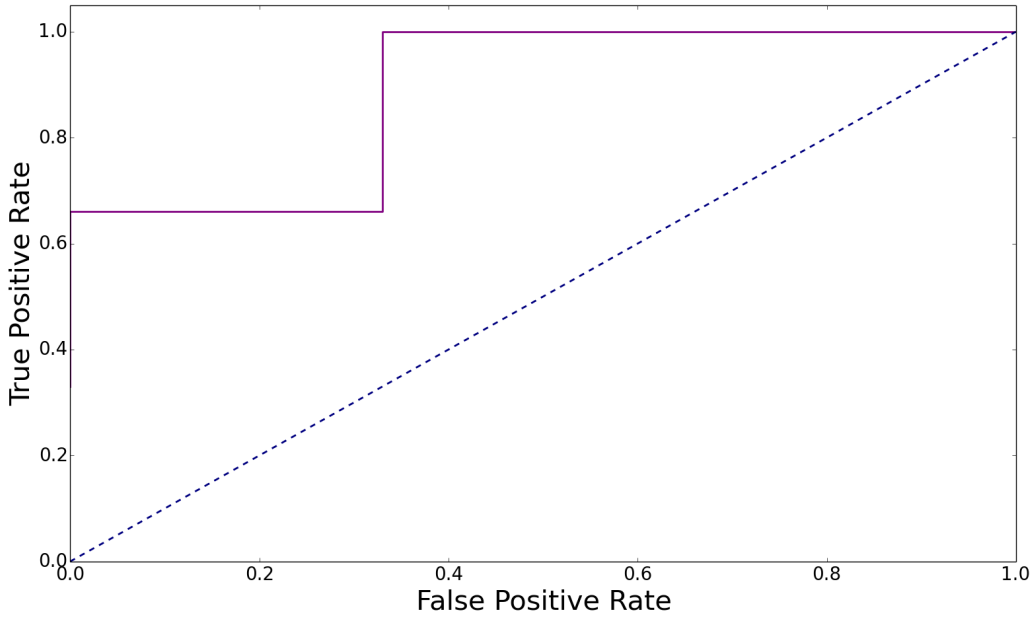

Figure 5.1: Example ROC curve based on the ranking in Table 5.2

model performs against other models to detect regionalisms in each language variety. The higher the area under curve in a ROC curve, the better are the results.

Similarly, we computed cosine similarity scores for DSM, Word2Vec1, and Word2Vec2 for the selected pseudo words and regular words. We labeled all pseudo words and regular words as 1 and 0 respectively. Then, we rank pseudo and regular words from lower to higher values. We again plot receiver operating characteristic (ROC) curves for the ranking and computed area under curve scores to compare the three models. We do not consider measures of keywordness for the pseudo word evaluation as we controlled for frequency

(freq $>$ 50 and $<$ 1000) in the selection of pseudo and regular words.

Chapter 6

Experimental Results

In the following subsections, we consider results based on evaluations using regionalisms and pseudo words.

6.1 Regionalisms Evaluation

For regionalisms evaluation, we consider each pair of corpora: Canadian English (CA) and Non-Canadian English (NOT-CA), Australian English (AU) and Non-Australian English (NOT-AU), and American English and Non-American English (NOT-US). In each case, we perform an experiment for all regionalisms and distractors selected for each corpus pair. We experimented with 8 different methods: *Kilgarriff Add1*, *Kilgarriff Add10*, *Kilgarriff Add100*, *Chi-Square*, *LLR*, *Word2Vec1*, *Word2Vec2* and *DSM*. Among these methods, we identify the method that can best identify regionalisms

against distractors. We plot a receiver operating characteristic curve for each method on each corpus pair. The ROC curves show the performance of all the methods in identifying regionalisms in varieties of English. The area under curve computed for each method provides an insight into how well each method performs against the others for a particular variety of English. Table 6.1 shows the area under curve for Australian, Canadian and American English for each method.

Method	**Australian**	**Canadian**	**American**
Kilgarriff Add1	0.597	0.483	**0.541**
Kilgarriff Add10	0.606	0.498	**0.541**
Kilgarriff Add100	**0.616**	0.582	0.534
Chi-Square	0.606	0.405	0.485
LLR	0.605	0.408	0.486
DSM	0.490	0.518	0.494
Word2Vec1	0.524	0.391	0.509
Word2Vec2	0.606	**0.681**	0.457

Table 6.1: Area under curve in ROC curves for Australian, Canadian and American English based on dictionary-based words evaluation for each method. The best AUC for each variety of English is shown in bold face.

For a random classifier, we expect an area under curve (AUC) of 0.5. Amongst the Kilgarriff methods, Kilgarriff Add100 performs best in the case of Australian and Canadian English. It's clearly seen from the area under curve that Kilgarriff Add100 performs better than expected for a random classifier

for Australian and Canadian English. Indeed for Australian English, Kilgarriff Add100 is the best method overall (AUC=0.616). The performance of Kilgarriff Add1 and Kilgarriff Add10 is better than that of Kilgarriff Add100 in the case of American English and these methods perform best overall for this corpus pair.

For Australian English, the chi-square statistic (AUC=0.606) gives good performance, whereas for the other varieties it does not perform well. It performs poorly especially in the case of Canadian English (AUC=0.405). Similarly, the Log-likelihood ratio performs very well for Australian English (AUC=0.605), but again the performance of LLR on Canadian and American English is below the expected AUC of 0.50 of a random classifier.

The results of the distributional similarity model (DSM) are overall very poor, as it does not rank regionalisms higher than distractors. The area under curve is below or roughly similar to that expected by a random ranking for each pair of corpora. Nevertheless, in future work the count-based distributional similarity model could potentially be improved by experimenting with larger context window sizes, such as 5 or 8 (where the current approach uses a window size of 2).

Turning to the neural network inspired word embedding approaches, Word2Vec1 identifies regionalisms of Australian (AUC=0.524) and American English (AUC=0.509) only slightly better than random, and performs even worse for Canadian English (AUC=0.391). Word2Vec2 outperforms all other methods (the measures keywordness, count-based distributional similarity model,

and Word2Vec1) for Canadian English (AUC=0.681) by a very large margin, which means the model successfully ranks the regionalisms of Canada towards the top of the ranking. Moreover, in the case of Australian English (AUC=0.606), Word2Vec2 performs reasonably well compared to other methods. In particular the area under curve is close to that of the best method Kilgarriff Add100 (AUC=0.616). However in the case of American English, Word2Vec2 ranks the distractors higher than the regionalisms (AUC=0.457). Nevertheless, these findings indicate that lexical semantic variation in varieties of English is captured by the Word2Vec2 approach clearly in 2 out of 3 cases (Australian and Canadian English). Moreover, Word2Vec2 outperforms Word2Vec1 and DSM for identifying regionalisms in these cases.

To further compare approaches, we examine receiver operating characteristic (ROC) curves in Figures 6.1, 6.2 and 6.3. As seen in Figure 6.1, for Australian English, Kilgarriff Add1, Kilgarriff Add10, Kilgarriff Add100, Chi-square, LLR, and Word2Vec2 show a substantial improvement over a random ranking. A random ranking would have an equal true positive and false positive rate throughout the curve. Word2Vec1 on the other hand starts with a high true positive rate; but it then drops below a random ranking. DSM starts by misclassifying the regionalisms, however, it shows some improvement in middle of the curve. However, Word2Vec2 and measures of keywordness maintain a high true positive rate by continuing to identify regionalisms as opposed to distractors. These results clearly show that Word2Vec2 and the keywords-based methods give good performance for Australian English.

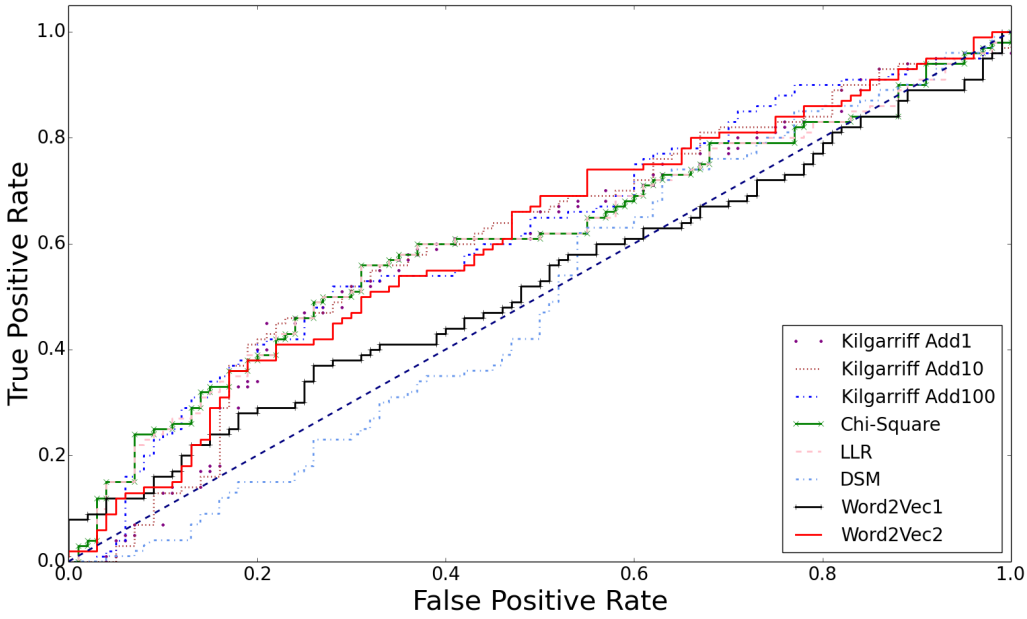

Figure 6.1: ROC curve for Australian English based on regionalisms evaluation.

For Canadian English in Figure 6.2, Word2Vec2 and Kilgarriff Add100 are above a random ranking throughout most of the ROC curve. Word2Vec2 keep a high true positive rate throughout the analysis by ranking the regionalisms towards the top of the list. For Canadian English, Word2Vec2 clearly gives the best performance.

For American English in Figure 6.3, the keywordness measures Kilgarriff Add1, Kilgarriff Add10 and Kilgarriff Add100 show some improvement above a random ranking, whereas, chi-square, LLR, DSM, Word2Vec1, and Word2Vec2 perform poorly overall. Therefore, we conclude that the Kilgarriff measures of keywordness perform best in identifying regionalisms of the United States,

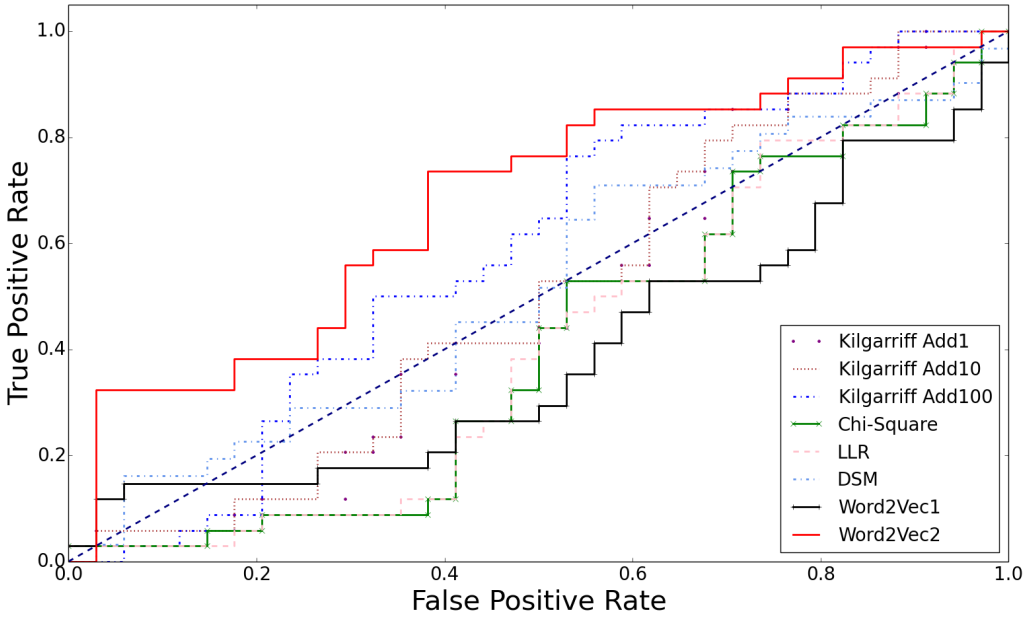

Figure 6.2: ROC curve for Canadian English based on regionalisms evaluation.

but even these methods perform rather poorly. The reason for the poor performance in the case of American English might be because of our evaluation resource (DARE). DARE attempts to document words that are specific to particular regions of the United States. Although these terms are Americanisms, the senses of words from DARE we are trying to capture might have lower frequency then those from CanOx and AND. Therefore, it might be difficult for our models to perform well on American English.

DSM does not perform well in any of the three case studies. The possible reason For the poor performance of DSM, in contrast to Word2Vec, might be because DSMs give sparse representations for words, while Word2Vec

representations are dense.

Word2Vec2 performs better than Word2Vec1 for Australian and Canadian English. This could be due to parameter turning for learning the transformation for Word2Vec1. In particular, it might be possible that we learned a bad transformation. Word2Vec1 might perform better by tuning the number of top most frequent words used to learn the transformation.

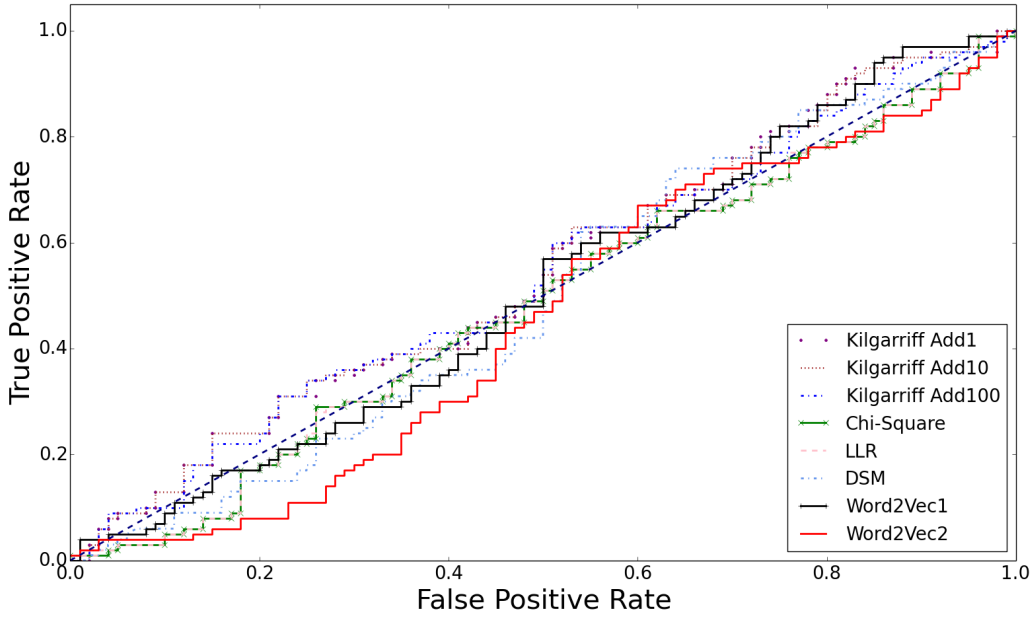

Figure 6.3: ROC curve for American English based on regionalisms evaluation.

6.2 Pseudo Words Evaluation

For the pseudo words experiments, we consider only one corpus pair: Canadian English corpus (CA) and Australian English corpus (AU). We perform an experiment for all pseudo words and regular words. We experimented with 3 different methods: *Word2Vec1*, *Word2Vec2* and *DSM*. We do not consider measures of keywordness for the pseudo word evaluation as we controlled for frequency (freq $>$ 50 and $<$ 1000) in the selection of pseudo and regular words. Therefore, frequency-based keywordness measures are not applicable. From all these methods, we identify the best method that can identify the pseudo words against the regular words. We plot receiver operating characteristic (ROC) curves for each method. The area under curve computed for each method shows how well one method performs against another. Table 6.2 shows the area under the curve for each method.

Method	**Area Under Curve (AUC)**
DSM	0.989
Word2Vec1	0.981
Word2Vec2	**1.0**

Table 6.2: Area under curve in ROC curves for pseudo words evaluation for each method.

For the count-based distributional similarity model, we observed a high area under curve (0.989). This clearly shows that pseudo words are ranked at the top, and regular words at the bottom, of the ranking list. Word2Vec1

also gives a high area under curve (0.981). However, comparing all three methods, Word2Vec2 performs extremely well. It gives a perfect ROC curve, i.e., area under curve = 1, which indicates that it correctly ranks all pseudo words above the regular words.

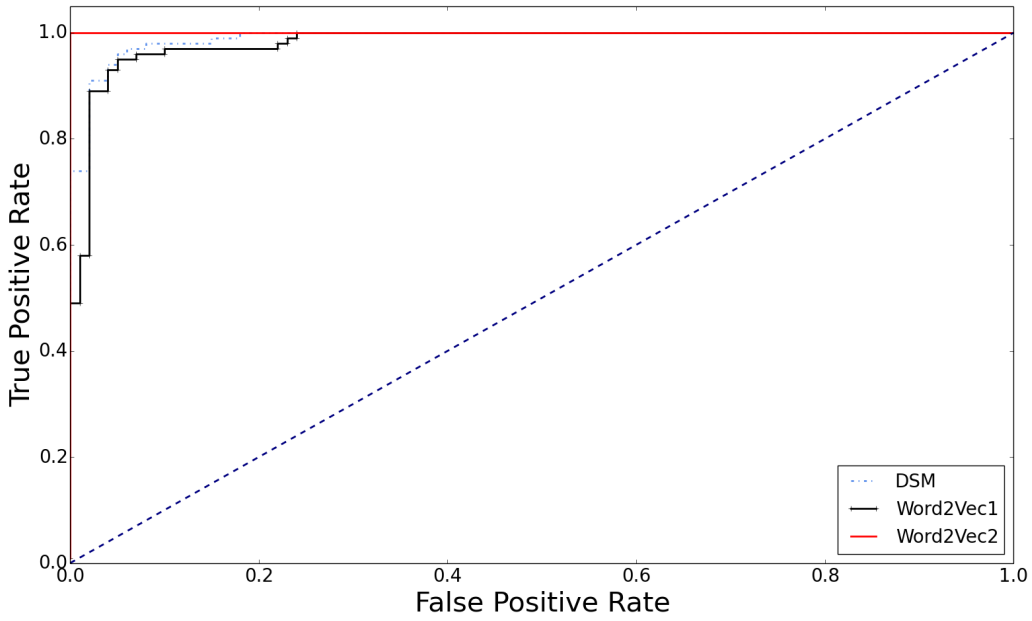

Figure 6.4: ROC curves for pseudo words evaluation for each method.

The ROC curves are shown in Figure 6.4. From all the three methods, Word2Vec2 gives the best performance— a straight line at true positive rate = 1. The pseudo words evaluation further suggests that Word2Vec2 outperforms the other methods for identifying lexical semantic change.

Chapter 7

Conclusions

This chapter summarizes the contributions this thesis has made and then describes a number of directions for future work.

7.1 Summary of contributions

The investigation of lexical semantic variation in language varieties of English is still in its early stage of research. One important problem for the field is the development of an automatic approach to lexical semantics that can discover differences in the meaning of a word in different language varieties. This thesis has described and evaluated an automatic approach to discover lexical semantic differences in varieties of English. In this thesis, we have evaluated methods based on distributional similarity, word embeddings, and methods of keywordness for this task. In particular, this thesis is the first work to

apply word embeddings to identifying lexical semantic differences between varieties of English. This is an interesting and important contribution in the area of lexical semantics.

In chapter 3, we describe how the sub-corpora for varieties of English are built from data available for four countries, i.e., Australia, the United States, the United Kingdom, and Canada. In chapter 4, we propose methods on using a count-based distributional similarity model, two Word2vec methods and five different methods of keywordness to identify regionalisms. In chapter 5, we describe the ways in which regionalisms for three varieties of English, and a set of pseudo words, are created for evaluation.

We evaluate our methods based on their ability to rank regionalisms of varieties of English higher than distractors. In chapter 6, we compare our eight methods by drawing receiver operating characteristic curves (ROC) and computing area under these curves. We showed that Word2vec2 was able to identify regionalisms for Australian English and Canadian English, but not for American English, and was the best method overall for Canadian English. Moreover, Word2vec2 outperformed Word2vec1 on Australian and Canadian English.

7.2 Future directions

In this section, we discuss a number of future directions related to the work carried out in this thesis.

The focus of this study is English specifically Australian, American, and Canadian English. The methods considered in this thesis could however be applied to any corpus pair, and potentially to identify lexical semantic differences between varieties of other languages, for example, European Spanish and American Spanish, Quebec French and Acadian French, and Belgian and Netherlandic Dutch. These methods could also be applied to other varieties of English, such as Indian English.

At the more computational side, we think it is important to explore some variations of the models that we have not been able to discuss here. For example, it would be interesting to use different parameter settings for Word2Vec and the distributional semantic models, for example, a window size of 2 or 8, using skip-gram, dimensionality of 500 or 800 for Word2vec, and also exploring window sizes of 5 or 8 for the distributional similarity model. We could also consider different α parameters for Kilgarriffs keywordness approach, such as 1000 or 10000. We could also apply models for automatic identification of new word-senses (Cook et al. 2014 [7]) to identify differences in meaning in varieties of English.

Bibliography

[1] Katherine Barber and Robert Pontisso, *The Canadian Oxford Dictionary*, Don Mills, Ont.: Oxford University Press, 2005.

[2] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin, *A neural probabilistic language model*, Journal of machine learning research **3** (2003), no. Feb, 1137–1155.

[3] Steven Bird, Ewan Klein, and Edward Loper, *Natural language processing with python: analyzing text with the natural language toolkit*, 2009.

[4] Anders Björkelund, Bernd Bohnet, Love Hafdell, and Pierre Nugues, *A high-performance syntactic and semantic dependency parser*, Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations, Association for Computational Linguistics, 2010, pp. 33–36.

[5] Frederic Gomes Cassidy, *Dictionary of American Regional English*, Belknap Press of Harvard University Press, 1985.

[6] Kenneth Ward Church and Patrick Hanks, *Word association norms, mutual information, and lexicography*, Computational linguistics **16** (1990), no. 1, 22–29.

[7] Paul Cook, Jey Han Lau, Diana McCarthy, and Timothy Baldwin, *Novel word-sense identification.*, COLING, 2014, pp. 1624–1635.

[8] Paul Cook and Suzanne Stevenson, *Automatically identifying changes in the semantic orientation of words.*, LREC, 2010.

[9] Andrew Giel and Ryan Diaz, *Document embeddings via recurrent language models*.

[10] Kristina Gulordava and Marco Baroni, *A distributional similarity approach to the detection of semantic change in the google books ngram corpus*, Proceedings of the GEMS 2011 Workshop on Geometrical Models of Natural Language Semantics, Association for Computational Linguistics, 2011, pp. 67–71.

[11] William L Hamilton, Jure Leskovec, and Dan Jurafsky, *Diachronic word embeddings reveal statistical laws of semantic change*, arXiv preprint arXiv:1605.09096 (2016).

[12] Dan Jurafsky, *Speech & language processing*, Pearson Education India, 2000.

[13] Adam Kilgarriff, *Simple maths for keywords*, Proceedings of the Corpus Linguistics Conference. Liverpool, UK, 2009.

[14] Yoon Kim, Yi-I Chiu, Kentaro Hanaki, Darshan Hegde, and Slav Petrov, *Temporal analysis of language through neural language models*, arXiv preprint arXiv:1405.3515 (2014).

[15] Vivek Kulkarni, Rami Al-Rfou, Bryan Perozzi, and Steven Skiena, *Statistically significant detection of linguistic change*, Proceedings of the 24th International Conference on World Wide Web, ACM, 2015, pp. 625–635.

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

[17] Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K Gray, Joseph P Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, et al., *Quantitative analysis of culture using millions of digitized books*, science **331** (2011), no. 6014, 176–182.

[18] Tomas Mikolov, Quoc V Le, and Ilya Sutskever, *Exploiting similarities among languages for machine translation*, arXiv preprint arXiv:1309.4168 (2013).

[19] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean, *Distributed representations of words and phrases and their compositionality*, Advances in neural information processing systems, 2013, pp. 3111–3119.

[20] George A Miller, *Wordnet: a lexical database for english*, Communications of the ACM **38** (1995), no. 11, 39–41.

[21] Joakim Nivre and Johan Hall, *A quick guide to maltparser optimization*, Dostupné z WWW http://maltparser. org/guides/opt/quick-opt. pdf,[cit. 2013-04-27] (2010).

[22] Yves Peirsman, Dirk Geeraerts, and Dirk Speelman, *The automatic identification of lexical variation between language varieties*, Natural Language Engineering **16** (2010), no. 04, 469–491.

[23] Jeffrey Pennington, Richard Socher, and Christopher D Manning, *Glove: Global vectors for word representation.*, EMNLP: Conference on Empirical Methods in Natural Language Processing, vol. 14, 2014, pp. 1532–1543.

[24] WS Ransom and William Stanley Ramson, *The Australian National Dictionary: a dictionary of Australianisms on historical principles*, Oxford University Press, 1988.

[25] Eyal Sagi, Stefan Kaufmann, and Brady Clark, *Tracing semantic change with latent semantic analysis*, Current methods in historical semantics (2011), 161–183.

[26] Roland Schäfer, *Commoncow: Massively huge web corpora from commoncrawl data and a method to distribute them freely under restrictive*

eu copyright laws, Proceedings of the tenth international conference on language resources and evaluation (LREC16), 2016, pp. 4500–4504.

[27] Helmut Schmid, *Treetagger— a language independent part-of-speech tagger*, Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart **43** (1995), 28.

[28] Roland Schfer, *Processing and querying large web corpora with the COW14 architecture*, Proceedings of Challenges in the Management of Large Corpora 3 (CMLC-3) (Lancaster) (Piotr Baski, Hanno Biber, Evelyn Breiteneder, Marc Kupietz, Harald Lngen, and Andreas Witt, eds.), UCREL, IDS, 2015.

[29] Luchen Tan, Haotian Zhang, Charles LA Clarke, and Mark D Smucker, *Lexical comparison between wikipedia and twitter corpora by using word embeddings.*, ACL (2), 2015, pp. 657–661.

[30] Yang Xu and Charles Kemp, *A computational evaluation of two laws of semantic change.*, CogSci, 2015.

Vita

Candidate's full name: Priyal Nagra
University attended (with dates and degrees obtained): University of New Brunswick, Sept 2015-May 2017, Master of Computer Science
Publications: In Progress
Conference Presentations: Research Exposition 2017 at University of New Brunswick