Paraphrase Pattern Acquisition by Diversifiable Bootstrapping

Hideki Shima

CMU-LTI-14-003

May 2015

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213

Thesis Committee:
Teruko Mitamura (chair), Carnegie Mellon University
Eric Nyberg, Carnegie Mellon University
Eduard Hovy, Carnegie Mellon University
Patrick Pantel, Microsoft Research

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in Language and Information Technologies.

Abstract

Texts that convey the same or close meaning can be written in many different ways. Because of this, computer programs are not good at recognizing meaning similarity between short texts. Toward solving this problem, researchers have been investigating methods for automatically acquiring paraphrase templates (*paraphrase extraction*) from a corpus.

State-of-the-art approaches in paraphrase extraction have limited ability to detect variation (e.g. "X died of Y", "X has died of Y", "X was dying of Y", "X died from Y", "X was killed in Y"). Considering practical usage, for instance in Information Extraction, a paraphrase resource should ideally have higher coverage so that it can recognize more ways to convey the same meaning in text (e.g. "X succumbed to Y", "X fell victim to Y", "X suffered a fatal Y", "X was terminally ill with Y", "X lost his long battle with Y", "$X(writer)$ wrote his final chapter Y"), without adding noisy patterns or instances that convey a different meaning than the original seed meaning (semantic drift).

The goal of this thesis work is to develop a paraphrase extraction algorithm that can acquire lexically-diverse binary-relation paraphrase templates, given a relatively small number of seed instances for a certain relation and an unstructured monolingual corpus. The proposed algorithm runs in an iterative fashion, where the seed instances are used to extract paraphrase patterns, and then these patterns are used to extract more seed instances to be used in the next iteration, and so on.

The proposed work is unique in a sense that lexical diversity of resulting paraphrase patterns can be controlled with a parameter, and that semantic drift is deferred by identifying erroneous instances using a distributional type model. We also propose a new metric DIMPLE which can measure quality of paraphrases, taking lexical diversity into consideration.

Our hypothesis is that such a model that explicitly controls diversity and includes a distributional type constraint will outperform the state-of-the-art as measured by precision, recall, and DIMPLE. We also present experimental results to support this hypothesis.

Acknowledgments

First and foremost, I would like to express the sincere gratitude to my academic advisor, Dr. Teruko Mitamura. She has been extremely supportive for the entire years of my graduate student life. My academic achievements including this dissertation work would not have been possible without the assistance of her. I cannot thank her enough for everything she has done for me.

I would like to extend my deep appreciation to Dr. Eric Nyberg. I learned a lot from him through research experiences in multiple projects he led, especially the Javelin QA project, and four courses I assisted as a TA. Through working closely together for years, his philosophy in science and engineering impacted me largely.

I am truly grateful for the support by my committee members, Dr. Eduard Hovy and Dr. Patrick Pantel. I am greatly honored to be able to receive eye-opening advice and research questions from the most recognized and enthusiastic experts I can imagine in the field.

My appreciation also goes to many people for their help in developing my academic career. Dr. Kemal Oflazer at CMU in Qatar gave me an opportunity to expand my knowledge in technology-driven education. Dr. Noriko Kando at NII in Japan mentored me in organizing a few NTCIR tasks. DeepQA/Watson team at IBM Research, especially Dr. James Fan, Dr. J. William Murdock, Dr. Adam Lally, Dr. Chris Welty, Dr. Hiroshi Kanayama, Dr. Takeda Koichi and Dr. Eric W. Brown, gave me an exciting opportunity to be a part of Jeopardy! grand-challenge as well as the Machine Reading project.

I thank faculty members, staff members and colleagues in CMU, especially Dr. Jaime Carbonell for leading and continuing to improve the world's top-level education and research organization, the Language Technologies Institute.

My thanks also go to Ms. Susan Holm for helping us to create a part of data used in this thesis.

Finally, I would like to thank my family and friends for their patience, love and support for many years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

What is Paraphrase?

Paraphrasing is one of disciplines in Natural Language Processing which concerns about paraphrases, or natural language expressions with meaning equivalence or similarity.

Table 1.1: Sub-areas of paraphrasing research

Classification of Paraphrase Research	Usage / Application
(1) Paraphrase Recognition <kill, murder> → {Y, N} (word/phrase-level) $<S_1, S_2>$ → {Y, N} (sentence-level) $<D_1, D_2>$ → {Y, N} (document-level)	• Question Answering • Text Summarization • Automatic Grading • Plagiarism Detection
(2) Paraphrase Generation • die → <decease, pass away, kick the bucket> • He had a lot of admiration for his job → He had plenty of admiration for his job	• Query Expansion • Reference Expansion in Automatic Evaluation
(3) Paraphrase Extraction ⛁ → {word, phrase, sentence} -level paraphrases • with/without variables • with/without structure <writer, author> <a lot of X, plenty of X> SUBJ FROM SUBJ TO <X buy Y, Y sell X> <X wrote Y, X is the writer of Y> $<S_1, S_2>$	• Resource for (1) and (2) ▪ Paraphrase dictionary ▪ Sentence-aligned paraphrase corpus

Paraphrasing research can be roughly classified into three subareas, namely paraphrase recognition, generation, and extraction (see Table 1.1). The table illustrates the variations of a paraphrase unit: words, phrases, templates (text fragment with variables), syntactically structured templates, sentences, and even longer texts. The three subareas of paraphrasing are as follows. Paraphrase recognition is a binary classification problem of deciding whether given two texts are paraphrases or not. Paraphrase generation is a problem about creating a paraphrase given text. Paraphrase extraction is a data-driven language resource acquisition problem where the goal is to find sets of paraphrase texts, given a text collection.

What exactly is paraphrase? According to literature, a paraphrase is defined as: "an alternative way to convey the same information" (Barzilay and McKeown, 2001); "an alternative surface form in the same language expressing the same semantic content as the original form" (Madnani and Dorr, 2010); "conveying the same or almost the same information" (Malakasiotis and Androutsopoulos, 2011). Although these definitions share a common keyword, *the same*, one extreme view could be that there is no such texts that have different surfaces but convey exactly *the same* semantics. For example, given a pair of synonyms ⟨ buy, purchase ⟩, it is possible to differentiate the two terms in a way that *buying* is general whereas *purchasing* is more formal. Consider a sentence pair ⟨ John bought oranges, Oranges were bought by John ⟩ as another example. Between the two sentences, the content word *bought* is common, but there is a syntactic difference: active v.s. passive voice. One could argue that even with this case, there is a difference in meaning, because the stress is put differently on the subject John and oranges, respectively.

According to Inkpen (2007), "there are very few absolute synonyms, if they exist at all" and they are "limited mostly to technical terms (distichous, two-ranked; groundhog, woodchuck) and groups of words that differ only in collocational properties, or the like".

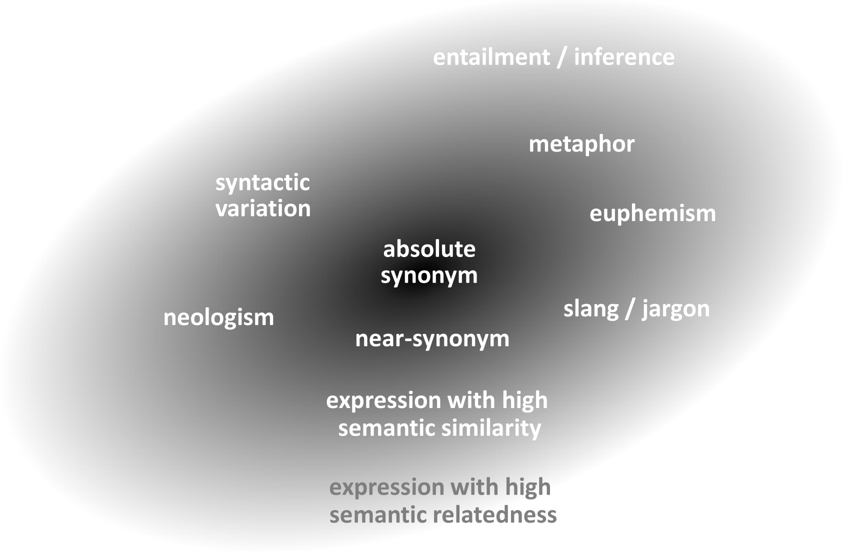


Figure 1.1: Granularity of meaning equivalence.

While absolute synonyms are truly *the same* intersubstitutable[1] phrases, there is a practical need to utilize a broad range of diverse expressions in a close meaning. Figure 1.1 visualizes similarity of meaning in a gradient where the level of darkness represents the strength of similarity.

[1] Words are intersubstitutable if they can be replaced each other in a context.

As the vague boundaries in the figure shows, we consider that meaning is continuum[2].

A very interesting question arises from Figure 1.1 is, where to draw a line to define *paraphrase* in the universe of meaning similarity. Should we limit paraphrase as absolute synonyms that are truly *the same* in meaning? Or should be instead consider a wider range?

Some previous works have already taken a broader definition of paraphrase with words such as "approximately", "essentially", "roughly" as follows: (paraphrases are) "approximately conceptually equivalent" (Barzilay and McKeown, 2001; de Beaugrande and Dressler, 1981); "essentially the same meaning" (Das and Smith, 2009); "roughly interchangeable given the genre" (Ibrahim et al., 2003); "The real world uses far more quasi-paraphrases than the logically equivalent ones" (Bhagat, 2009). The last paper argues that "a large number of paraphrases used in the real world are quasi-paraphrases". The authors take an example of ⟨killed, beheaded⟩ and states that the pair should be considered as paraphrases or quasi-paraphrases for a practical reason, even though they are not synonyms each other.

There seems to be no single definition of paraphrase that every researcher can agree on. The definition of paraphrase depends, as discussed by Dras (1999): "it is necessary to examine paraphrase under a particular context and application". In other words, it is impossible to come up with a universal definition of paraphrase, since the value of paraphrase is defined given a specific application. For example, in the paraphrase generation problem, preservation of grammaticality is a requirement; otherwise paraphrasing rules cannot be applied to generate a new sentence. In addition, meaning preservation might also be a requirement for automatic evaluation of Machine Translation where human-written gold standard references can be expanded with paraphrase generation methods. However, when it comes to Information Extraction related problems such as Question Answering (QA), paraphrase templates with high lexical and syntactic variety are desired. One extreme in this direction would be Patent Information Retrieval/Extraction problem, which is a recall-oriented task with a full of obfuscated expressions.

In this thesis, we will adopt the *quasi-paraphrases* definition; we define "paraphrase" in a broad range of semantically similar expressions, given a particular context and application.

Why Do We Want to Extract Paraphrases?

Recognizing meaning equivalence is a common key challenge in various NLP applications where analyzing meaning similarity or equivalence could be useful. Figure 1.2 gives an example scenario in QA problem[3] where binary-argument paraphrase template would help[4] to find an answer candidate from a corpus.

Suppose an input to a QA system is the following question: "What did John Lennon die of?". A system would first transform this interrogative sentence into an affirmative form "John Lennon died of what". With the sentence, the system would then try to find answer-bearing passages from a corpus such that can align. However, there is no guarantee that the corpus

[2] Although the figure is in 2 dimension, it may be more appropriate to define the meaning similarity to be in a continuum of N-dimensional space with N-linguistic properties.

[3] Modern QA system automatically finds answers from open-domain unstructured text collections, rather than a "structured" database, given a question posed in a natural language.

[4] Note that this is one of many possible approaches in QA.

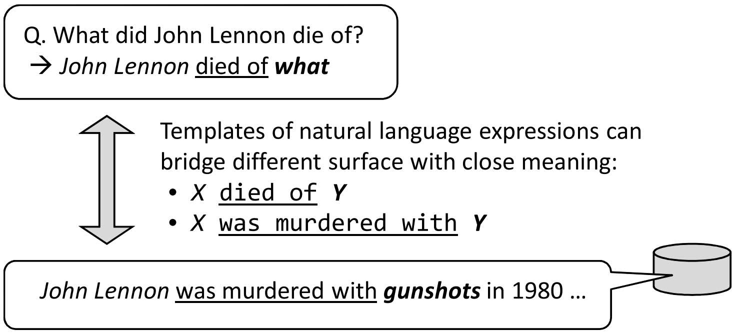

Q. What did John Lennon die of?
→ John Lennon died of what
Templates of natural language expressions can bridge different surface with close meaning:
• X died of Y
• X was murdered with Y
John Lennon was murdered with gunshots in 1980 …

Figure 1.2: Paraphrasing template used to help identify answer candidate in QA.

contains passages that contain the exact same verb phrase "die of". Instead, a passage may have a different expression "was murdered with" as in "John Lennon was murdered with gunshots …". In order to bridge the question and the answer-bearing passage in different surface texts, and then identify the answer candidate "gunshots", a paraphrase template such as $\langle X$ died of Y, X was murdered with $Y\rangle$ would be extremely useful.

Why Is Lexical Diversity Important in Paraphrase?

It is often the case that the same meaning is expressed in various different surface forms. In QA, the better the coverage of templates are, the more likely a system can capture correct answers, and boost the confidence with an accumulated evidence. For example, see a diversity of expressions in Figure 1.3.

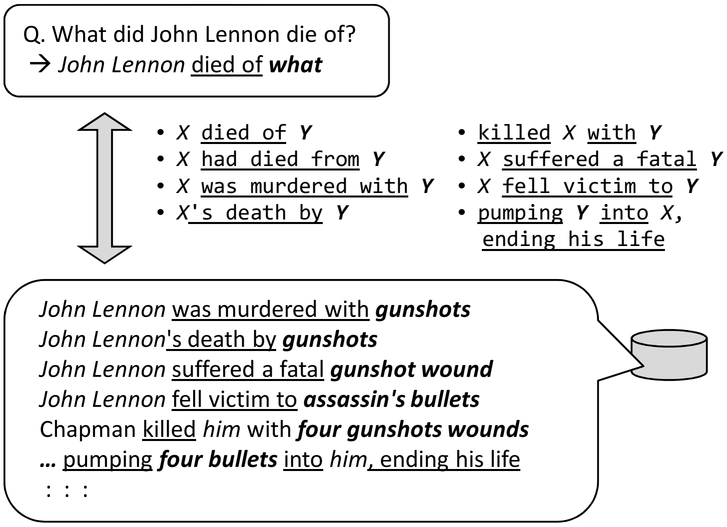

Q. What did John Lennon die of?
→ John Lennon died of what
• X died of Y
• X had died from Y
• X was murdered with Y
• X's death by Y
• killed X with Y
• X suffered a fatal Y
• X fell victim to Y
• pumping Y into X, ending his life
John Lennon was murdered with gunshots
John Lennon's death by gunshots
John Lennon suffered a fatal gunshot wound
John Lennon fell victim to assassin's bullets
Chapman killed him with four gunshots wounds
… pumping four bullets into him, ending his life
: : :

Figure 1.3: Diverse set of paraphrasing templates help accumulating evidence for answer candidates in QA

One might wonder if existing machine-readable thesauri are sufficient to be used to bridge mismatches of surface strings. For example, let us take a look at words that are associated with the first synset[5] of "die" (*pass from physical life and lose all bodily attributes and functions necessary to sustain life*) in WordNet (Miller, 1995):

> die, decease, perish, go, exit, pass away, expire, pass, kick the bucket, cash in one's chips, buy the farm, conk, give-up the ghost, drop dead, pop off, choke, croak, snuff it

Given these examples, one could observe that WordNet "synonyms" falls short for a few reasons. Firstly, these words lack useful contexts such as prepositions. Without the contexts, false positive may occur for a polysemy words such as *go* as a synonym of *die*.

Secondly, they lack coverage; traditional thesauri do not contain exhaustive list of expressions that can be paraphrases. See Table 1.2[6] for phrasal expressions that represent the meaning of dying, but do not contain a word synonymous to die. It is important to note that the kinds of paraphrases in Figure 1.3 and in Table 1.2 are not usually found in traditional thesauri.

Table 1.2: Diverse paraphrases for *die*.

Type	Example Paraphrases
Idioms	bite the dust, go west, give up the ghost go to a better place, pay the ultimate price
Religious euphemism	be carried away by angels, answer God's calling, go to heaven, reach nirvana
Euphemism by profession	*(author)* write *one's* final chapter, *(dancer)* dance *one's* last dance, *(gambler)* cashed in their chips
Slang in military	go Tango Uniform, go T.U., turn *one's* toes up, be KIA *(killed in action)*, be KIFA *(killed in flight accident)*, be DOW *(died of wounds)*
Slang in physician	be at room temperature, be bloodless, feel no pain, lose vital signs, wear a toe tag
Slang in gangsters	merc, merk, murk, snuff, smoke, bang, get a backdoor parole

Paraphrase recognition and generation are not types of problems a simple mathematical formula or supervised machine learning algorithm alone solves; in general, paraphrasing is a resource-intensive problem where the key is the knowledge source of meaning. Therefore, there is a strong need for solving lack-of-resource issue by automatically acquiring diverse paraphrase expressions.

By the way, note that we do not claim that lexical diversity is the single most important factor in a paraphrase resource. As we discussed earlier, it depends on application – lexical diversity is especially useful in recall-oriented applications such as exhaustive and comprehensive search of patent documents (Joho et al., 2010).

[5] A synset is a conceptual node in WordNet.

[6] More examples available at http://en.wikipedia.org/wiki/List_of_expressions_related_to_death and http://www.lvc.edu/rel314/euph.aspx

Paraphrase Research in Various Applications

High quality paraphrasing methods are effective in various NLP applications. The list below extends usages of paraphrasing seen in Table 1.1:

- **Question Answering**. Between question and answer-bearing sentence (Bogdan et al., 2008; Duboue and Chu-Carroll, 2006; Harabagiu and Hickl, 2006; Hermjakob et al., 2002).
- **Information Retrieval**. For query expansion and exhaustive high-recall retrieval (Clinchant et al., 2006; Parapar et al., 2005; Riezler et al., 2007; Zukerman and Raskutti, 2002).
- **Information Extraction**. For increasing a chance of matching a vocabulary in a pattern (Kouylekov, 2006; Romano et al., 2006).
- **Text Summarization**. For measuring the meaning redundancy among summary candidates (Barzilay et al., 1999; Lloret et al., 2008; Tatar et al., 2009).
- **Intelligent Tutoring**. For checking whether a student's answer can entail a reference answer (Nielsen et al., 2009).
- **Automatic evaluation**. For automatically evaluating textual system output against human references in Machine Translation (Kauchak and Barzilay, 2006; Padó et al., 2009; Zhou et al., 2006a), Text Summarization (Zhou et al., 2006b), and Question Answering (Dalmas, 2007; Ibrahim et al., 2003).
- **Plagiarism Detection**. For identifying plagiarisms across multiple documents (Burrows et al., 2012; Özlem Uzuner et al., 2005).
- **Collocation error correction**. (Dahlmeier and Ng, 2011).

High coverage paraphrase resource would be critical in recall-oriented tasks such as Patent Document Retrieval, which is an exhaustive retrieval task where "the objective is to use some text from each patent topic to automatically retrieve all cited patents found in the collection"(Magdy and Jones, 2010).

1.1 Fundamental Problems

So far we have discussed why it is worthwhile to automatically extract paraphrases from a large collection of text data, especially taking into account lexical diversity. In this section, we will discuss problems remaining to be solved in paraphrase extraction research community.

Corpus Restriction

Previous paraphrase extraction methods require special corpora, especially, parallel or bitext corpora. For instance, sentence-aligned bilingual parallel corpora (Bannard and Callison-Burch, 2005; Callison-Burch, 2008; Kok and Brockett, 2010), and sentence-aligned monolingual comparable corpora (Chen and Dolan, 2011; Dolan et al., 2004; Dolan and Brockett, 2005; Quirk et al., 2004; Tiedemann, 2007) are often used as sources to extract paraphrases.

The state-of-the-art works often use bilingual parallel corpora, which are originally used in Statistical Machine Translation (SMT). Such a corpus contains pairs of sentences where a pair is translation of each other. These works have an interesting perspective that a translation pair is a paraphrase in a different language, and utilize unsupervised phrase-alignment technique in SMT. Basically, an algorithm works in the following way. First, find an alignment of a phrase in language A with phrases in language B. Then, find the reverse direction of alignment from these phrases in language B back to phrases in language A. In this way, one can find a group of phrases in language A that are paraphrases. The is at least one downside in extracting paraphrases from a parallel corpus for SMT; they are aligned with literal translations where meaning is preserved as much as possible. Therefore, one cannot expect to obtain diverse ways of expressions.

One of the biggest problems in the paraphrase extraction methods using either a comparable or bilingual parallel corpus is that it requires a such special sentence-aligned corpus, which requires huge effort build. Therefore, the methods do not scale in terms of source corpus. Another issue is that domain specific paraphrase extraction becomes challenging due to potential bottleneck of lack of corpus.

Ideally, paraphrases are extracted from a monolingual unstructured corpus that does not require any additional post-processing on top of raw-text.

Lack of Lexical Diversity

The state-of-the-art pattern learning algorithms could be applied to paraphrase extraction problem. However, patterns extracted from these algorithms typically lack in lexical diversity. See Figure 1.4 for an illustrative example. Without explicitly addressing lexical diversity problem, extracted paraphrases would result in those in the left hand side, with a small variations of content words.

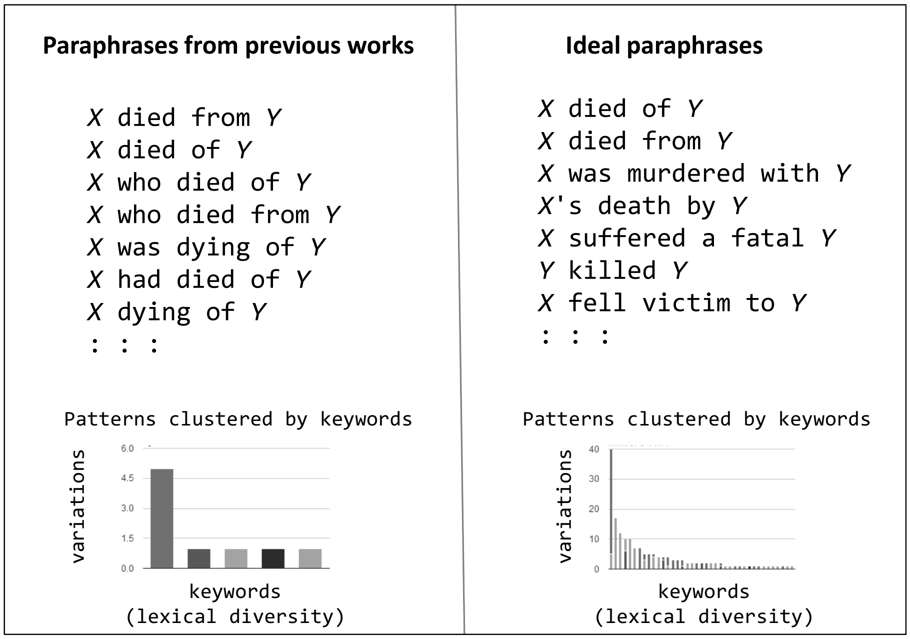


Figure 1.4: Contrasting differences in paraphrases (binary-argument templates for *died-of* relation) with syntactic and lexical diversity

Semantic Drift in Iterative Extraction Models

Bootstrap is a minimally-supervised method for acquiring lexical resources, which we will use to harvest paraphrases in this thesis.

During iterations in bootstrapping, a noisy item (those ambiguous or erroneous instances or patterns) can be extracted. Such noise can affect the bootstrapping process by contaminating more and more result in later iterations. This phenomena is called Semantic Drift (Curran et al., 2007), which is one of the key challenging problems to be solved in this type of lexical resource acquisition method.

Consider learning binary-argument templates for "died-of" using bootstrapping technique. The inputs are concrete entity mentions of X and Y (called seed instances) and a monolingual corpus. The expected output would be texts that represents a person X died due to the cause Y: $\langle X$ died of Y , X died from $Y, \ldots \rangle$.

Let us present a concrete example illustrating how Semantic Drift happens. Suppose $\langle X$ died in $Y \rangle$ pattern has been found at the n-th iteration. This pattern can capture correct pairs of a person and the cause of death as in a sentence: "John died in a car accident", as well as pairs of a person and the year of death as in "John died in 1979".

If there is no mechanism to detect erroneous instances, patterns acquired after (n+1)-th iteration would be also partly erroneous such that represents a year of a person's death, instead a cause.

By the way, Named Entity Recognition (NER) is a commonly used technique in Natural Language Processing where the goal is to identify spans of text in a sentence which are named entities. Figure 1.5 illustrates how one would check the type using NER. This approach may work for some relations where X and Y are types supported in NER. However, this approach is be limited for the cause-of-death relation, which instances could vary in multiple types such as disease and accidents whereas common NER systems identify a small number of coarse-grained types[7].

Evaluation

To the best of our knowledge, there is no evaluation metric on extracted paraphrases that takes into account lexical diversity, and also is backed up by empirical evidence thorough a large scale meta-evaluation. Existing metrics, for example precision, would give the perfect score for the paraphrases from the both approaches in Figure 1.4. When lexical diversity is also under concern[8], we ideally need a better metric that can distinguish these two sets and reward more for the one containing variety of relevant content words.

[7] For example, a widely used CoNLL shared-task NER dataset (Tjong Kim Sang and De Meulder, 2003) supports only PERSON, LOCATION, and ORGANIZATION.

[8] Again, we stress that we do not claim that lexical diversity is the single most important criteria in any circumstances.

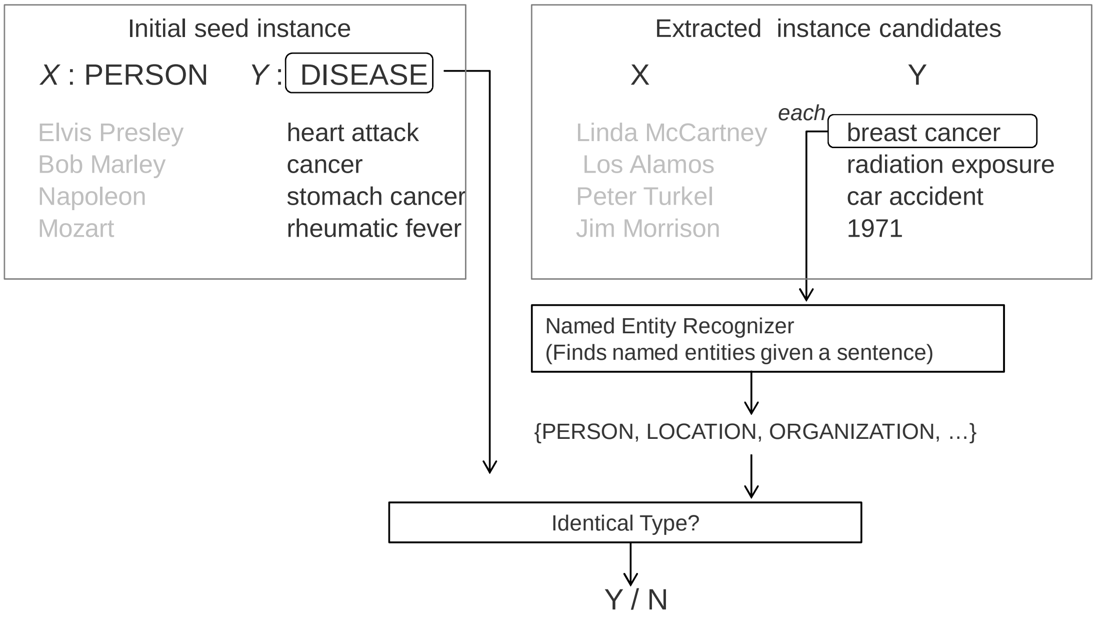


Figure 1.5: An invalid instance is detected using a NER system.

1.2 Approach

Being motivated by the need and importance of acquiring lexically diverse paraphrases, we will discuss a framework for automatically acquiring a set of templates, or text fragments with variables given just a monolingual unstructured corpus and a small number (5-20) of seed instances, or pairs of instances for X and Y.

1.2.1 Diversifiable Bootstrapping for Paraphrase Acquisition

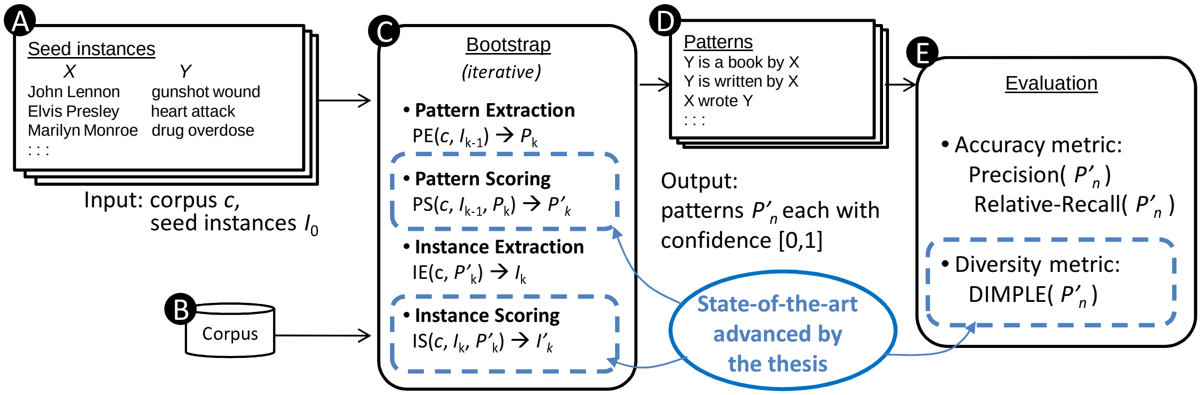


Figure 1.6: The bootstrap paraphrase acquisition framework.

We will present a *Diversifiable Bootstrapping* framework (see Figure 1.6) that takes a monolingual corpus c and a small number of (i.e. 5-20) seed instances I_0 as input, and generates scored binary-relation patterns P'_n as output after the n-th iteration. Each member of I_0 consists of a pair of entities in a certain relation (e.g. x-died-of-y, x-was-born-in-y, x-graduated-from-y). For example, for a *killing* relation, I_0 is a pairs of person names, or killers and victims: $I_0 = \langle\langle$"Mark David Chapman", "John Lennon" $\rangle, \langle$"Sirhan Sirhan", "Robert F. Kennedy"$\rangle, \ldots\rangle$. An entity is not necessarily a proper noun (e.g. location, person, organization), but could be any object or numeric expression. A pattern is a surface phrasal text that connects two variables (i.e. a killer and victim) in the relation e.g. X killed Y, X is a killer of Y. In the figure, (C) is the bootstrapping pattern acquisition part which can be divided into four steps: Pattern Extraction $PE(c,\ I_{k-1}) \rightarrow P_k$; Pattern Scoring $PS(c,\ I_{k-1},\ P_k) \rightarrow P'_k$; Instance Extraction $IE(c,\ P'_k) \rightarrow I_k$; and Instance Scoring $IS(c,\ I_k,\ P'_k) \rightarrow I'_k$. The next iteration starts after seed instances are expanded with harvested ones: $I_k = I_{k-1} \cup I_0\ (k > 1)$. Iteration continues until a certain convergence criterion is met. Under the Extended Distributional Hypothesis (Lin and Pantel, 2001), patterns that co-occur with similar instances can be treated as having similar meanings, and this is an underlying assumption we have about seeing relation patterns P'_n as paraphrase templates that can convey similar meaning. The bootstrapping algorithm is based on Pantel and Pennacchiotti (2006), with multiple clear differences. For example, in PS calculation step, we apply what we call *Lexical Diversification* where one can control how much lexical diversity to be realized in a set of patterns, by adjusting a parameter value. Unlike most bootstrapping framework aims to acquire instances and sees patterns as by-products, our work aim to acquire patterns.

1.2.2 Measuring open-domain type similarity by distributional type model

We will present a novel approach for measuring type similarity between an entity and a set of entities. See Figure 1.7 for the general idea of the proposed approach (cf. traditional approach in Figure 1.5).

In this approach, the constraint of instances is not explicitly defined as a traditional ontological type; instead, the constraint is represented as a weighted vector of super-types. This way, there is no need to use a predefined ontology for instances, which is not always suitable as some instances, such as cause-of-death, has arguments which type can be a set of independent nodes in traditional ontology (e.g. disease, accident).

Using the proposed method, fine-grained constraints can also be applied on instances. To be more specific, in the cause-of-death example, the first argument can be softly restricted in a finer-grained types than a person, such as male, criminal, musician, novelist. The type constraint is distributional, so both coarse and fine grained types can be joined to form one type vector.

1.2.3 Comparison with the state-of-the-art

Difference and uniqueness of our work on paraphrase extraction part is summarized and contrasted with other state-of-the-art works in Table 1.3.

Previous works generally require a special corpus (e.g. bilingual parallel corpus, monolingual comparable corpus), or huge monolingual corpus as input. On the other hand, our work requires relatively smaller monolingual corpus. This is a big advantage when one cannot prepare a special corpus, but has an ordinary corpus. It is especially advantageous when someone wants to apply

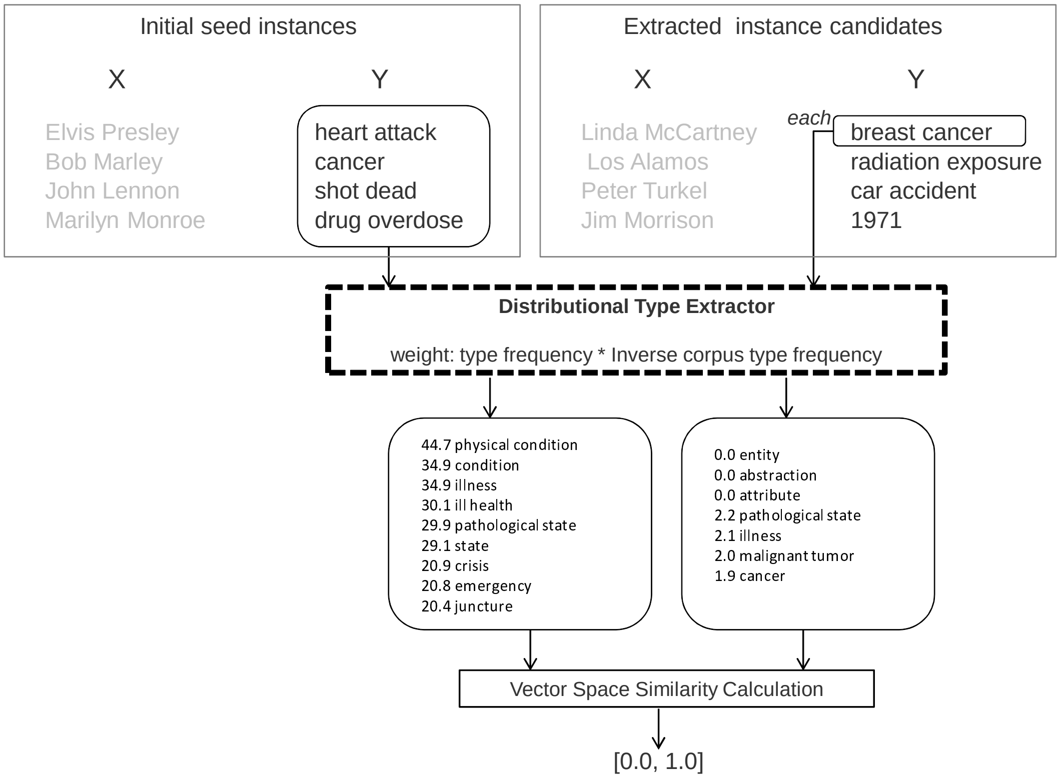


Figure 1.7: In the proposed method, an invalid instance can be detected through vector space similarity calculation.

our work to a new closed-domain (e.g. bio, patent, terrorism news, sports news, noisy texts from social media or speech transcript) or to a different language. On another note, dependency on NLP technologies is minimum in our basic framework, which is good for languages with scarce linguistic resources or tools.

Table 1.3: Uniqueness of Proposed Work in contrast with other data-driven paraphrase acquisition methods.

	Alignment-based (Callison-Burch, 2008; Kok and Brockett, 2010)	Distributional (Bhagat and Ravichandran, 2008)	Distributional (Metzler and Hovy, 2011)	**Distributional Bootstrapping (Proposed)**
Source corpus	Bilingual parallel corpus	Monolingual corpus (150GB)	Monolingual corpus (up to 4.5TB)	Smaller monolingual corpus (1∼10GB)
NLP required	Syntactic Parser	POS Tagger	Syntactic Parser	POS Tagger
Paraphrase target	Various phrases	Binary-relation	Verb phrases	Binary-relation (including noun phrase expressions)
Can control lexical diversity?	No	No	No	Yes

1.3 The Goal and Contribution of this Dissertation

The goal of this thesis work is to develop a paraphrase acquisition framework in *Bootstrapping* which can acquire lexically-diverse paraphrase rules represented as surface-text binary-relation template, given seed instances on a certain relation and a monolingual corpus. The contributions and the outline of the thesis are summarized in the table 1.4.

1.4 Thesis Outline

The rest of this thesis is organized as follows.

- Chapter 2 will give a literature review on existing state-of-the-art works on paraphrase extraction, bootstrap learning, and paraphrase evaluation. For each of these areas, we will identify limitations that are going to be addressed in the thesis work.
- Chapter 3 will start with a formalization of paraphrase diversity and semantic drift. We will also discuss the basic framework for paraphrase extraction, which will be the base which we will substantially extend in this work.
- Chapter 4 will discuss the new approach for calculating similarity of entities in a distributed type space. Distributional similarity between instance candidate types and original seed types are calculated. This way, we will be able to detect erroneous instances which would cause semantic drift.

Table 1.4: Summary of contributions.

Limitations in State-of-the-art (Ch. 2)	Ch. describing the contribution	Hypothesis	Evaluation Metric
Corpus Restriction: previous works have special corpus requirement e.g. parallel corpus, web as a corpus.	Ch. 3 Bootstrap Paraphrase Acquisition Framework	It is possible to extract paraphrase templates from an unstructured monolingual corpus given seed instances.	Precision, Recall, DIMPLE, and number of distinct keywords found.
Semantic Drift: bootstrap pattern-instance learning can easily mess up with erroneous or ambiguous item.	Ch. 4 Preventing Semantic Drift	Semantic drift risk from diversification be mitigated by distributional type restriction.	Precision by iteration (decreases when semantic drift happens)
Lack of Lexical Diversity: preventing semantic drift too much results in extracting patterns with poor lexical diversity.	Ch. 5 Diversifying Lexicons in Paraphrase Bootstrapping	Lexical diversity of acquired paraphrase can be controlled with a model of relevance-dissimilarity interpolation.	DIMPLE
Lack of Evaluation Metric: precision or recall does not reward lexical diversity.	Ch. 6 Diversity-aware Evaluation Metric for Paraphrase Patterns	Cumulative-gain style evaluation metric which gives reward to lexically diverse paraphrases is effective for paraphrase evaluation.	Correlation (Pearson's *r*) with paraphrase recognition task score.

- Chapter 5 will propose *Diversifiable Bootstrapping*, a novel algorithm that can explicitly control a degree of lexical diversity of paraphrases. This algorithm is a unique approach to solving the lack of lexical diversity problem in paraphrase extraction.
- Chapter 6 will propose a diversity-aware paraphrase evaluation metric called DIMPLE. The metric adds another perspective to paraphrase evaluation which were limited in traditional metrics, such as precision or recall.
- Chapter 7 will present experimental result and analysis of the results. The experiments cover algorithms appeared from Chapter 3 to Chapter 6.
- Finally, Chapter 8 will give concluding remarks on the proposed thesis works. We will also discuss some future works.
- Additionally, Appendix A gives examples for the extracted patterns, and Appendix B describes the guideline for annotating gold standard labels for each paraphrase.

Chapter 2

Related Works of Paraphrase Extraction

In this chapter, we will review state-of-the-art studies on paraphrase extraction and identify unsolved problems. The section structure is designed in the following way due to different aspects of paraphrase resource construction each with different limitations to be solved.

First, we will review the distributional approaches (Section 2.1) that are based on distributional similarity model. Then, we will go though alignment-based approaches (Section 2.2) which exploits the special corpus structure; paraphrases are extracted from parallel corpus where sentences are already aligned so that they have the same meaning (but in different language) or comparable in meaning.

Section 2.3 reviews bootstrap frameworks which is promising to automatically extract paraphrase expressions from a monolingual corpus with minimum supervision.

Data-driven approaches above constructs paraphrase data which is not necessarily perfect due to its algorithmic nature. One may ask if it is better to just use dictionary/thesaurus that contains a lot of synonyms and convert them into paraphrase templates, or ask human to write up paraphrases. To answer this question, we will also review resources built by lexicographer, and "crowd" or a massive number of non-experts in linguistics (Section 2.4), and why we need an automated paraphrase acquisition model.

Section 2.5 reviews evaluation methods of extracted paraphrases. Finally, Section 2.6 summarizes common limitations identified in this chapter, which are understudied and worthwhile to be solved in this thesis.

2.1 Paraphrase Extraction by Distributional Approaches

Early work on data-driven pattern acquisition has mainly been for Information Extraction (Agichtein and Gravano, 2000; Brin, 1998; Califf and Mooney, 2003; Riloff, 1996; Sudo et al., 2001) and its applied area Question Answering (Ravichandran and Hovy, 2002). Extraction patterns acquired in these works are not intended to be paraphrases, however, the works influenced later researches in paraphrase acquisition. The acquisition methods we review in this section are using Distributional Similarity. More specifically, the approaches use a variant of Distributional Hypothesis (See Section 3.2.2) either implicitly or explicitly.

The strength of this paraphrase extraction approach, in contrast with alignment-based approaches that will be covered in Section 2.2, is that only a monolingual corpus is required.

DIRT (Lin and Pantel, 2001)

Lin and Pantel (2001) pioneered the distributional approach for pattern learning, in their work DIRT. First, DIRT extracts dependency paths from a parsed monolingual corpus[1]. Then it scores patterns p using an measure based on Mutual Information in order to measure the strength of associations with slot fillers $Slot, w$.

$$mi(p, Slot, w) = \log\left(\frac{|p, Slot, w| \times |*, Slot, *|}{|p, Slot, *| \times |*, Slot, w|}\right) \tag{2.1}$$

where the notation $|p, SlotX, w|$ denotes the frequency of the triple $(p, SlotX, w)$ observed in a corpus and: $|*,*,*| = \sum_{p,s,w} |p,s,w|$. Subsequently, $sim(slot_1, slot_2)$ is computed to measure the similarity between a pair of slots:

$$sim(slot_1, slot_2) = \frac{\sum_{w \in T(p_1,s) \cap T(p_2,s)} mi(p_1,s,w) + mi(p_2,s,w)}{\sum_{w \in T(p_1,s)} mi(p_1,s,w) + \sum_{w \in T(p_2,s)} mi(p_2,s,w)} \tag{2.2}$$

where p_1 and p_2 are paths, s is a slot and $T(p_i, s)$ is the set of words that fill in s of p_i. The similarity between a pair of paths $S(p_1, p_2)$ is defined as the following geometric mean:

$$S(p_1, p_2) = \sqrt{sim(SlotX_1, SlotX_2) \times sim(SlotY_1, SlotY_2)} \tag{2.3}$$

where $SlotX_i$ and $SlotY_i$ are path i's slots $SlotX$ and $SlotY$.

Paşca and Dienes (2005)

Paşca and Dienes (2005) used the variable-length left and right n-gram contexts of the source term, with scoring based on overlap. In short, the algorithm works as quoted below (see also Algorithm 1 for more details):

> The proposed acquisition method collects large sets of word and phrase-level paraphrases via exhaustive pairwise alignment of small needles, i.e., sentence fragments, across a haystack of Web document sentences. The acquisition of paraphrases is a side-effect of the alignment.

Mavuno (Metzler and Hovy, 2011)

Metzler and Hovy (2011) used similar approach to Paşca and Dienes (2005) but the scoring function over the variable length n-gram contexts is based on cosine similarity rather than overlap scoring. The two algorithms proposed by them works as quoted below:

[1] 1 GB newswire for their experiment

Algorithm 1 Algorithm for paraphrase acquisition from Web document sentences (Paşca and Dienes, 2005)

```
Input: $\{S\}$ set of sentences
  $L_C$ length of constant extremities
  $Min_P, Max_P$ paraphrase length bounds
Output: $\{R\}$
  Set $\{N\} \leftarrow \emptyset$ {set of ngrams with attached info}
  Set $\{P\} \leftarrow \emptyset$ {set of pairs (anchor, candidate)}
  Set $\{R\} \leftarrow \emptyset$ {set of paraphrase pairs with freq info}
  for each sentence $S_i$ in $\{S\}$ do
    Generate ngrams $N_{ij}$ between length $2 \times L_C + Min_P$ and $2 \times L_C + Max_P$
    for each $N_{ij}$, attach additional info $Att_{ij}$ do
      Insert $N_{ij}$ with $Att_{ij}$ into $\{N\}$
    end for
  end for
  for each ngram $N_i$ in $\{N\}$ do
    $L_{N_i}$ = length of $N_i$
    $C_{st_L}$ = subseq [0, $L_C - 1$] of $N_i$
    $C_{st_R}$ = subseq [$L_{N_i}$, $L_C - 1$] of $N_i$
    $Var_i$ = subseq [0, $L_C - 1$] of $N_i$
    $Anchor_i$ = concat of $C_{st_L}$ and $C_{st_R}$
    $Anchor_i$ = concat of $Att_i$ and $Anchor_i$
    Insert pair $(Anchor_i, Var_i)$ into $\{P\}$
  end for
  Sort pairs in $\{P\}$ based on their anchor
  for each $\{P_i\} \subset \{P\}$ with same anchor do
    for all item pairs $P_{i1}$ and $P_{i2}$ in $\{P_i\}$ do
      $Var_{i1}$ = variable part of pair $P_{i1}$
      $Var_{i2}$ = variable part of pair $P_{i2}$
      Increment count of $(Var_{i1}, Var_{i2})$ in $\{R\}$
      Increment count of $(Var_{i2}, Var_{i1})$ in $\{R\}$
    end for
  end for
  Return $\{R\}$
```

Mav-N: Our proposed paraphrase acquisition approach using variable length n-gram contexts and cosine similarity for scoring. This is identical to the PD approach, except cosine similarity scoring is used in place of overlap scoring.

Mav-C: Same as Mav-N, except typed chunk contexts are used instead. The context of a phrase is defined as the concatenation of the chunk (and its type) immediately before and after the phrase. This is similar to the Bhagat and Ravichandran approach, except we do not limit ourselves to noun phrases as contexts and do not use locality sensitive hashing. It is also similar to the BCB-S approach, in that it requires paraphrases to have the same syntactic type (i.e. same chunk type) as the input phrase.

Bhagat and Ravichandran (2008)

Bhagat and Ravichandran (2008) used noun phrase chunks as contexts. They used locality sensitive hashing to reduce the dimensionality of the contextual vectors. Scoring is achieved using the cosine similarity between PMI weighted contextual vectors.

They evaluate the similarity between two phrases p_i and p_j as the similarity between two corresponding vectors:

$$sim(p_1, p_2) = \frac{V_i \cdot V_j}{|V_i| * |V_j|} \tag{2.4}$$

where the vector V is defined to be the set of words that occur with $p \in P$.

2.1.1 Limitations

Lack of Laxical Diversity

In Table 2.1, we compare paraphrases of “killed” acquired by different state-of-the-art distributional paraphrase acquisition algorithms. Depending on a method, small lexical diversity is observed (see the third column). In addition, most paraphrases are verbs. Note that it is very common that relations are represented in nouns phrases in text. For example, “X assassinated Y” can be written as “assassination of Y by X” or “Y’s assassin is X”.

Using the Web as a corpus

Some works above rely on using the web as a corpus, because the larger the source data is, the higher the chance of observing co-occurrence of events. However, there can be following issues in using the web (and web search engine):

- **Domain specific paraphrase learnability**. If a huge corpus is a must, it may mean extracting paraphrases from small non-web corpus is challenging. There is need for learning domain-specific paraphrases that are used in a small community only (e.g. slang and jargon used in legal documents, medical publications, terrorist documents).

Table 2.1: Comparison of paraphrases acquired for “killed”, from the dataset by Metzler et al. (2011).

Method	Paraphrases Acquired	Unique Correct Keywords Acquired
Bannard and Callison-Burch (2005)	murdered; died; beaten; been killed; are; lost; were killed; kill; have died	murder, die, kill (3)
Bhagat and Ravichandran (2008)	killed in; killed ,; that killed; killed NN people; killed NN; killed by; were wounded in; and wounding; dead , including; , hundreds	kill, dead (2)
Paşca and Dienes (2005)	used; made; involved; found; born; done; injured; seen; taken; released	N/A (0)
Metzler and Hovy (2011)	wounded; injured; arrested; left; that killed; were killed; involved; killing; claimed; shot dead	kill, dead (2)

- **Reproducibility**. Static source of information is a key to reproducible scientific outcomes. If we use the web as a corpus, result may change suddenly at one day because of its dynamic nature.
- **Punctuations indexability**. We cannot precisely count a pattern occurrences in the web, if it contains a punctuation or symbols[2] since they are often not usually indexed, and we do not have a control over the indexing strategies in existing commercial search engines.
- **Affect of query expansion**. Nowadays, a search engine such as Google performs a “semantic search” by automatically matching alternative expressions between a query and documents (e.g. “Steelers” and “Stealers”). Thus, researchers who use search engines to calculate corpus statistics are negatively impacted. On the other hand, a local (non-web) corpus with an open source search engine is free from these problems, as it allows one to control which tokens to index and how.
- **Noise**. Blindly using the web as a corpus is not always appropriate, due to a huge volume of noisy irrelevant documents. Studies report that local corpus may outperform the web in Question Answering (Clarke et al., 2002; Katz et al., 2004).

2.2 Paraphrase Extraction by Alignment

There are sentence-aligned parallel or comparable corpus where each sentence convey same or similar meaning (but may be in different languages). The paraphrase extraction methods reviewed here will take advantage of the parallel structure in bi-text (a.k.a. parallel corpus), and find paraphrases as a result of alignment.

[2] They are important sometimes, for instance, a comma is used for apposition, parentheses are used to give an attribute to an entity.

2.2.1 From Monolingual Parallel/Comparable Corpora

Corpora that has comparable views of the same or similar concepts are use to extract paraphrases. There are many variants along this line of research.

- **Multiple Translations**. Multiple translations can be seen as conveying the same meaning, where paraphrases can be extracted from the differences among them. Pang et al. (2003) used Multiple-Translation Chinese Corpus with 105 news documents (993 sentences), translated independently by 11 translation agencies. Other than newswire, Barzilay and McKeown (2001) and Ibrahim et al. (2003) used books of foreign novels such as *Twenty Thousand Leagues Under the Sea*, translated by different authors in different time periods and different countries.
- **News Contents**. Shinyama et al. (2002) assumes the same event on the same day is reported differently in multiple news articles, and they are paraphrases. They use preservation of Named Entities to find identical events.
- **Definitions**. Hashimoto et al. (2011) extracted paraphrases from definitional sentences found from the web, under an assumption that they convey mostly the same information.
- **Query logs**. Query logs are also used in combination with click information to acquire paraphrases. Zhao et al. (2007) take question logs form Microsoft Encarta and analyze inter-query paraphrases. Zhao et al. (2010) assumes a search engine query and its relevant document title convey the same meaning, and extracts paraphrases.

Researchers are actively developing automatic methods for creating comparable corpus. This line of works includes but not limited to the following.

- **News Headlines**. Monolingual comparable corpus of news headlines is built from different sources covering the same story.
- **News Content**. Monolingual clusters of news articles reporting the same event are collected as paraphrase corpus (Dolan et al., 2004; Dolan and Brockett, 2005; Quirk et al., 2004).
- **Subtitles**. Tiedemann (2007) built a comparable corpus of subtitles.
- **Video Descriptions**. Chen and Dolan (2011) collected multiple independent descriptions of short, unambiguous videos.

2.2.2 From Bilingual Parallel Corpora

We will go through techniques that make use of bilingual or multilingual corpora. The idea is that multiple translations of the same sentence in foreign language are paraphrases each other. One advantage of this series of research is that research outcomes from Statistical Machine Translation community, from theories to tools, are applied to the paraphrase extraction problem.

Bannard and Callison-Burch (2005) proposed an approach to acquire paraphrases from a bilingual corpus, by exploiting word alignment methods used in Machine Translation. The work is further extended by Callison-Burch (2008), so that syntactic constraints are introduced to limit paraphrase candidates to have a similar syntactic form as the original phrase.

Based on the alignment-based approaches (Bannard and Callison-Burch, 2005; Callison-Burch, 2008), Kok and Brockett (2010) modeled bilingual parallel sentences as a graph where a node corresponds to a phrase, and an edge represents whether their corresponding phrases are aligned. They used random walk sampling to compute the average number of steps it takes to reach a ranking of paraphrases with better ones being "closer" to a phrase of interest. According to an evaluation by the authors, it outperformed the work by Callison-Burch (2008).

More elaborated paraphrase scoring model makes use of a log-linear model over pairs of patterns and features extracted from them (Zhao et al., 2008a,b, 2009a,b).

2.2.3 Limitations

Corpus Restriction

The common key issue among the works reviewed in the section is that a special corpus or a data structure must be exploited in order to automatically find paraphrases.

Alignment-based approaches require either a monolingual comparable corpus or bilingual/-multilingual parallel corpora that are aligned in sentence level. We cannot always obtain such a resource in large-scale, for a specific domain or specific language. Metzler and Hovy (2011) points out the downside of approaches using these corpora as follows:

> On the other end of the spectrum are more complex approaches that require access to bilingual parallel corpora and may also rely on part-of-speech taggers, chunkers, parsers, and statistical machine translation tools. Constructing large comparable and bilingual corpora is expensive and, in some cases, impossible. Additionally, reliance on language-dependent NLP tools typically hinders scalability, limits inputs to well-formed English, and increases data sparsity.

Moreover, Heilman (2011) mentions these approaches as "relatively new and error-prone", and have quality concerns: "in an intrinsic evaluation of paraphrase quality, Callison-Burch (2007, p. 75) found that only 57% of automatically generated paraphrases were grammatical and retained the meaning of the original input".

Lack of Lexical Diversity in Paraphrase Patterns

Parallel corpus often used for training a Machine Translation system typically lack in variation of expression within a sentence-pair. It is because of a nature of parallel-corpus where meaning is preserved as much as possible each other, especially in texts translated in word-for-word style. Paraphrases extracted from such corpora as a result of word or phrase alignment process would lack in lexical diversity.

2.3 Review of Bootstrap Learning

2.3.1 Bootstrap Approaches

Acquiring a language resource in an automatic data-driven approach is essential for overcoming the lack of knowledge-base.

When it comes to language acquisition by human children, Linguistics community use the following *bootstrapping hypotheses* to explain how they learn syntax and lexicons: lexical and syntactic acquisition are “interleaved, each using partial information provided by the other”(Siskind, 1996).

In Computational Linguistics, there is also a similar but a different notion of *bootstrapping*, that is, a method for acquiring language resources through iterative semi-supervised processes of obtaining lexicons (often called *instances*) and their contexts (often called extraction *patterns*) (Carlson et al., 2010a; Komachi and Suzuki, 2008; McIntosh et al., 2011; Pantel and Pennacchiotti, 2006; Riloff and Jones, 1999; Szpektor et al., 2004; Thelen and Riloff, 2002; Yu and Agichtein, 2003).

It is important to note that many bootstrapping works aim to extract instances rather than patterns. Our work is critically different from them in a sense that we see the value in patterns, which used to be seen as kind of by-products in some previous works.

2.3.2 Semantic Drift Prevention

In bootstrapping, there is a well-known but unsolved problem called Semantic Drift, which occurs “when a lexicon’s intended meaning shifts into another category during bootstrapping”(Curran et al., 2007). There are various attempts made to suppress semantic drift.

- **Convergence Detection**. At the end of each iteration, one can detect if the bootstrapping is *converged*, or is the right time to stop the iteration. Iteration continues “until it extracts τ_1 patterns or the average pattern score decreases by more than τ_2 from the previous iteration” as done in Pantel and Pennacchiotti (2006)[3]. Unless there is convergence detection mechanism, an erroneous pattern might be selected as a valid pattern after going through a series of iterations. On the other hand, one should note that Komachi and Suzuki (2008) points out that “Espresso still shows semantic drift unless iterations are terminated appropriately”.
- **Top-*k* Reliable Selection**. Rather than adding all of the extracted instances or patterns, one can select top-*k* of them in order to lower the risk of including less-reliable or less-precise ones. Riloff and Jones (1999) proposed a multi-level bootstrapping which scores and ranks terms by reliability and selects only the top five in each iteration. Pantel and Pennacchiotti (2006) proposed to newly add the best pattern, and use top 200 instances in each iteration.
- **Generic Pattern Filtering**. According to Komachi et al. (2008), “a straightforward approach to avoid semantic drift is to terminate iterations before hitting generic patterns”. Generic patterns are “high recall / low precision patterns (e.g, the pattern *X* `of` *Y* can ambiguously refer to a part-of, is-a and possession relations)”(Pantel and Pennacchiotti, 2006). Since low precision patterns can cause a Semantic Drift, one may not want to use them for extracting instances[4]. Espresso’s generic pattern detection criteria is as follows:

[3] Espresso sets $\tau_1 = 5$ and $\tau_2 = 50\%$

[4] Unlike some previous works that completely discarded generic patterns, Pantel and Pennacchiotti (2006) makes use of them for improving instance reliability estimation, but not for extracting instances. We currently completely temporarily filter out generic patterns for saving computational cost.

"a pattern as generic when it generates more than 10 times the instances of previously accepted reliable patterns" Pantel and Pennacchiotti (2006).

- **Negative Category Based Approaches**. Negative class instances can be given as seed (Curran et al., 2007; Lin et al., 2003; McIntosh and Curran, 2008) or found in an unsupervised way (McIntosh, 2010) in order to detect instances that would cause a semantic drift. The downside of this approach is that it "requires expert knowledge" (Kiso et al., 2011).
- **Mutual Exclusion Bootstrapping**. A Mutual Exclusion method are applied as a hard binary constraint (Curran et al., 2007) or softly weighted constraint (McIntosh and Curran, 2008) in the state-of-the-art of bootstrapping algorithms. The latter one is an extension of the former, and it makes bootstrapping "significantly less susceptible to semantic drift" (McIntosh and Curran, 2008).
- **Seed Selection Based Approaches**. We may incorporate a prediction model that estimates goodness of seeds and selects top-*k* good ones, following approaches used by (Kiso et al., 2011; Kozareva and Hovy, 2010). In this way, ambiguous or erroneous seed instances that may cause Semantic Drift may be automatically eliminated. This approach would be appropriate especially when large existing seed candidates are available, such as Wikipedia Infobox instances. In Relation Extraction research community, there are also studies that make use of the Infobox data for positive examples (Fei and Weld, 2007, 2008, 2010; Weld et al., 2009; Welty et al., 2010).

2.3.3 Limitations

Semantic Drift

As we have reviewed, semantic drift is the key problem in bootstrapping research, and there exist variety of approaches to address the problem. Nevertheless, the problem is challenging that there is no critical solution yet. Therefore, we need a solution that can impose better constraints on instance or pattern candidates found during an iterative process.

Lack of Lexical Diversity

Preventing semantic drift too much is also harmful from the viewpoint of lexical diversity.

For instance, Figure 2.1 shows the result of Espresso. The ranked list of extracted patterns do not show lexical diversity as seen by their content words: *assassin*, *assassination*, and *assassinated*. The extracted patterns have syntactic and morphological variations, but not lexical. One possible explanation behind this result, the lack of lexical diversity, is that by relying on highly precise top-*n* patterns at the *n*-th iteration, a preference to select a new pattern became too conservative. In the end of the first iteration, only one pattern with the highest score was selected for the next iteration (e.g. "*X*, the assassin of *Y*"). As a result, instances harvested at the second iteration did not represent the expected relation (e.g. *killed*), but did represent more specific relation (e.g. *assassinated*[5]). Given these instances, the same thing would have applied to the pattern extraction in the second iteration. In this way, as iterations went on, patterns might have

[5] An assassination is a special kind of a deliberate killing act that could happen to a prominent person.

X, the assassin of Y

assassination of Y by X

X assassinated Y

the assassination of Y by X

of X, the assassin of Y

X assassinated Y in

X, the man who assassinated Y

Y's assassin, X

of Y's assassin X

of the assassination of Y by X

⋮

Figure 2.1: Patterns extracted by a *vanilla* Espresso given Ephyra seed data Schlaefer et al. (2006) and a Wikipedia corpus.

got skewed toward a set of very similar expressions, with no room for a heterogeneous pattern to be in.

2.4 Paraphrase Resources by Human Lexicographer or the Crowd

Thesaurus

There are many existing resources that are potentially useful for the paraphrase recognition problem. To name a few, there are WordNet (Miller, 1995), FrameNet (Baker et al., 1998), Nomlex (Macleod et al., 1998), VerbNet (Kipper et al., 2006), and Grady Ward's MOBY Thesaurus (Ward, 1996).

Crowdsourcing Approaches

There are attempts to collect sentence-level paraphrases by asking "the crowds" on the web, or non-expert workers who are not paid, or paid with extremely low reward. For example, Chklovski (2005) collected paraphrases through a game. Max and Wisniewski (2010) took the Wikipedia, a crowd-authored encyclopedia, and analyzed the edit history to regard what is edited as paraphrase expressions. Negri et al. (2012) collected Chinese Whisper game where participants are requested to explicitly change one part of sentence preserving the original meaning.

2.4.1 Limitations

Coverage of dictionary

Dictionaries do not typically have phrasal expressions such as "fell victim to", "be terminally ill with", "lose *one's* long battle with" that are paraphrases of "die of". In addition, figurative expression and slang tend to be missing too: “write *one’s* final chapter”, “go T.U.”, “be at room temperature” (from Table 1.2).

Coverage of manually written patterns

Romano et al. (2006) investigated a relationship between recall in a relation extraction task and a number of manually provided extraction templates. They obtained 175 templates after normalizing the “syntactic variability phenomena” (i.e. passive form, apposition, conjunction, set, relative clause, coordination, transparent head, co-reference), which resulted in templates with lexical, but not syntactic, diversity. As seen in Figure 2.2, the curve is steep in the recall range between 0 to 50%; however after 50%, the curve is relatively gentle. It takes only 25 templates to achieve the 50% recall, but takes as many as 175 to achieve the 100%. This suggests that a large number of lexically diverse templates is one of keys to achieving high recall in a relation extraction task. Since extraction templates can be viewed as “a set of non-symmetric paraphrases” (Romano et al., 2006), it is implied that a large-scale lexically diverse paraphrase resource would play a very important role in dealing with the variability phenomena in text.

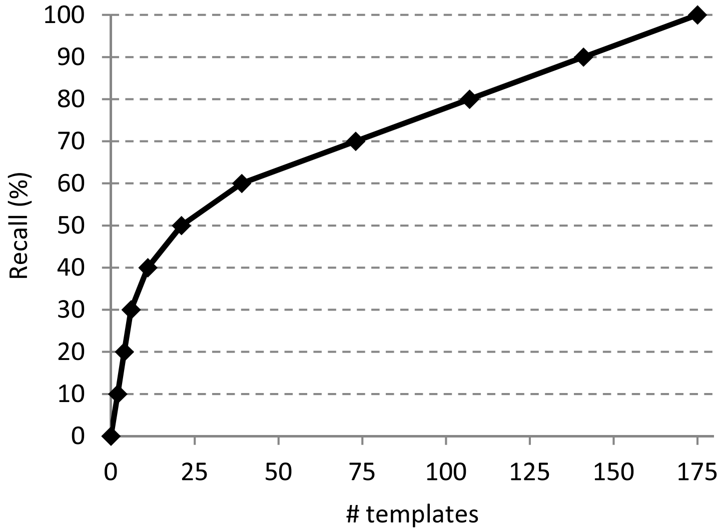


Figure 2.2: The number of most frequent templates necessary to reach different recall levels. We plotted this chart from the data at Table 5 in (Romano et al., 2006).

On the other hand, syntactic diversity, as exemplified in Table 2.2, also needs to be handled when processing semantics in text.

False Positives an Ambiguity

Table 2.3 shows a WordNet entry for *lead*, where different usages are grouped into conceptual units, or *synsets*. There are at least two issues in using this kind of synonymy source off-the-shelf. First, some words are highly ambiguous and expected to cause false-positives. For example, *lead* has the following “synonyms”: *take*, *result*, *head*, *go*, and *run*. Word-sense disambiguation

Table 2.2: Syntactic variability examples for a protein-protein interaction template "*X* activate *Y*", from Table 1 in (Romano et al., 2006)

Phenomenon	Example
Passive form	*Y* is activated by *X*
Apposition	*X* activates its companion, *Y*
Conjunction	*X* activates prot3 and *Y*
Set	*X* activates two proteins, *Y* and *Z*
Relative clause	*X*, which activates *Y*
Coordination	*X* binds and activates *Y*
Transparent head	*X* activates a fragment of *Y*
Co-reference	*X* is a kinase, though it activates *Y*

(WSD) has been studied for decades, but due to the difficulty of the WSD problem, some are concerned about whether WSD with high accuracy is an attainable goal (Brown, 2008). Secondly, words in such lexical resources are detached from contexts, which sometimes help in reducing ambiguity. For example, a certain linguistic phenomena may occur with a word in certain synset, such as passivization and use of a certain preposition. In addition, verbs and its derivationally related forms are not linked with contexts. WordNet synset comes with just one example sentence, and it is not enough considering the number of words associated with the synset. In other words, in addition to simply knowing *lead* and *leader* are derivational morphologies, it would be useful to know that "*X* led *Y*" convey the same meaning as "*X* was a leader of *Y*" (where the context part is underlined).

Table 2.3: Synonyms of "lead (v)" in WordNet.

Synset	Words	Definition
1	*lead, take, direct, conduct, guide*	take somebody somewhere
2	*leave, result, lead*	produce as a result or residue
3	*lead*	tend to or result in
4	*lead, head*	travel in front of; go in advance of others
5	*lead*	cause to undertake a certain action
6	*run, go, pass, lead, extend*	stretch out over a distance, space, time, or scope; run or extend between two points or beyond a certain point
7	*head, lead*	be in charge of
8	*lead, top*	be ahead of others; be the first
9	*contribute, lead, conduce*	be conducive to
10	*conduct, lead, direct*	lead, as in the performance of a composition
11	*go, lead*	lead, extend, or afford access
12	*precede, lead*	move ahead of others in time or space
13	*run, lead*	cause something to pass or lead somewhere
14	*moderate, chair, lead*	preside over

2.5 Paraphrase Evaluation

Evaluating paraphrase is challenging for various reasons. Many existing approaches use a paraphrase evaluation methodology where human assessors judge each paraphrase pair as to whether they have the same meaning. For example, Expected Precision (EP) is calculated by taking the mean of precision, or the ratio of positive labels annotated by assessors over a set of paraphrases (Bannard and Callison-Burch, 2005; Callison-Burch, 2008; Kok and Brockett, 2010; Metzler et al., 2011).

Different criteria can be considered in evaluating paraphrase resources. For example, such criteria can be syntactic consistency, grammatical correctness, or dissimilarity/diversity among paraphrases. Metzler et al. (2011) showed different trends can be seen when evaluating different paraphrase datasets. This suggests an importance of making use of multiple unique evaluation metrics when evaluating a paraphrase resource, or a method that generates such a resource.

With respect to evaluating diversity, there are studies that operate on *sentence-level* paraphrases. PEM (Paraphrase Evaluation Metric) (Liu et al., 2010) automatically evaluates paraphrases using three criteria: adequacy, fluency, and lexical dissimilarity. PINC (Paraphrase In N-gram Changes) (Chen and Dolan, 2011) also incorporates diversity in evaluating paraphrase sentences.

2.5.1 Limitations

The weakness of the EP-based approaches is an intrinsic measure that does not necessarily predict how well a paraphrase-embedded system will perform in practice. For example, a set of paraphrase pairs ⟨“killed”, “shot and killed”⟩, ⟨“killed”, “reported killed”⟩ ... ⟨“killed”, “killed in”⟩ may receive a perfect score of 1.0 in EP; however, these patterns do not provide lexical diversity (e.g. ⟨“killed”, “assassinated”⟩) and therefore may not perform well in an application where lexical diversity is important.

Actually, there is no single standard evaluation metric in paraphrasing research. This is partly because the value of paraphrase (e.g. precision, coverage as lexical and syntactic diversity, meaning preservation, grammaticality preservation, out-of-dictionary expression) can vary and tradeoff. There is would be a practical merit if a single metric, such as F-score (a harmonic mean of precision and recall) used in other domains, can be used as a standard metric. However, it is impossible to calculate recall for paraphrase because “the number of all correct pairs that could have been extracted from a large corpus (by an ideal method) is unknown” (Androutsopoulos and Malakasiotis, 2009). In addition, single-dimension binary judgement may not be appropriate due to the variety of values of paraphrases stated above.

Another problem in paraphrase evaluation is that steadily making gold standard annotation is challenging, since it is about semantics where subjectivity comes in. For the same reason, achieving high inter-annotator agreement is also challenging.

That said, we will have a better understanding of paraphrase extraction systems if we use multiple metrics and analyze extracted paraphrase from multiple perspectives. Since evaluation is on semantics, robust evaluation is challenging. A clearly documented guideline with affluent examples would be needed, which is not publicly available in the paraphrasing community yet.

2.6 Summary of Problems in Previous Works

We have reviewed various paraphrase extraction techniques in this chapter and identified important limitations. These limitations are summarized into the following four aspects, which raise key research questions.

1. State-of-the-art paraphrase extraction algorithms have a **corpus restriction** that requires special corpora to be used. Can we extract paraphrases from a casual corpus (i.e. non-web, non-parallel, monolingual)?
2. Paraphrase extraction by an iterative method suffers from **semantic drift**. Is there a better way to introduce constraints and mitigate the risk of semantic drift?
3. **Lack of coverage of paraphrases** especially due to low lexical diversity is a common issue in paraphrase data either extracted by state-of-the-art algorithms or constructed by human. Can we build a pattern scoring model that encourages more lexical diversity?
4. **Evaluation of lexical diversity** is understudied in paraphrasing extraction community. Can we design a metric that evaluates paraphrases considering the diversity?

This thesis presents solutions to the four limitations above, which are needed to be solved in order to advance the state-of-the-art of paraphrase extraction research.

Chapter 3

Bootstrap Paraphrase Acquisition Framework

As discussed in Section 2.2.3 (Chapter 2), majority of previous works of paraphrase acquisition require a special form of language resource (e.g. a parallel corpus, comparable news corpus, and the web as a corpus). In this chapter, we will first present formalization of two important notions along the line of bootstrap paraphrase extraction: semantic drift and paraphrase diversity in Section 3.1 (Note that unique contributions to each of them are presented later in Chapter 4 and 5, respectively.) Then in Sections 3.2 and 3.4, we will discuss an iterative paraphrase learning model, or bootstrapping framework.

3.1 Formalization of Paraphrase Diversity and Semantic Drift

3.1.1 Diversity

Consider a paraphrase acquisition task which goal is to extract a set of paraphrases P whose semantics are centered around concept c. Depending on a task, the representation of c might be a word with a description defining the meaning[1] in WordNet (Miller, 1995), a pointer to a word meaning[2], a word embeddings or vector space representation[3], or as in this work, pairs of entities that are in a specific relationship[4]. A paraphrase extractor is a system s such that finds $P = s(c)$.

Then, let us consider evaluating s with a performance measure m. We could infer s to be performing better than a baseline s' if the following is true: $m(s(c)) > m(s'(c))$. In order to conduct a reliable evaluation without chance effect, statistical significance should be tested using a large enough set of different $c_1, \ldots, c_n$.

Assume we use precision as a performance measure m, which is a fraction of correct paraphrases to all paraphrases extracted. One of difficulties in paraphrase evaluation is due to this

[1] For example, *kill* (cause to die; put to death, usually intentionally or knowingly)

[2] For example, WordNet synset kill#v#1

[3] For example, ((“destroy”, 0.596), (“exterminate”, 0.590), (“decapitate”, 0.567),. . .)

[4] For example, $\langle John\_Lennon, gunshot\_wound \rangle$, $\langle Bob\_Marley, cancer \rangle \ldots$

correctness; evaluation involves with ambiguous meaning where objective decision can not always be made. Ambiguity in meaning can result in inconsistent decisions on determining a correctness among evaluators, or even for the same person in different time.

We model the correctness judging process as $correct = rel_c(p) > \theta$ using a threshold θ, where meaning relevance *rel* is a function such that quantifies the relevance of p to the target concept c.

Although it may not be the exact process humans process meaning, this has a couple of advantages. First, continuity of meaning can be modeled in *rel*. Second, θ can be adjusted depending on a need where paraphrases are applied. Using Iverson bracket notation, precision is written as

$$prec_c(P) = \sum_{p \in P} \frac{[rel_c(p) > \theta]}{|P|}.$$

A problem of relying on precision as a single performance measure is that a very similar text fragments with minor differences, for example in an extreme case simply with or without a comma or a functional word, can receive a full credit (see Figure 2.1 in Chapter 2). Ideally, it would be convenient if there in a performance measure that takes into account both correctness and diversity.

Therefore, consider the following formalization of diversity. Let *sig* be a function that maps a paraphrase $p \in P$ into a signature of p. Diversity can be represented as the size of the set of paraphrase signatures where paraphrases must be relevant to concept:

$$diversity_c(P) = |\{s \,|\, s = sig(p), p \in P, rel_c(p) < \theta\}|.$$

The above formalization is simple enough that it makes us aware of the following important research questions:

1. What would be an ideal or practical function *sig*? Can it take into account different levels of diversity, such as morphological variations, content word variations, embedding distance, or pragmatic differences?
2. Should all s treated equally, or should it be given with a weight corresponding with relevance or any other criteria?
3. Is there a relationship between precision and diversity metrics, as illustrated in Figure 3.1?

Later in this thesis, Chapter will 6 addresses the first two questions, with more sophisticated diversity metrics presented. Also, Section 7.4 in Chapter 7 will presents experimental results that addresses the precision-diversity relationship.

3.1.2 Semantic Drift

In bootstrap lexical knowledge acquisition, there is a common key problem called Semantic Drift. This is a phenomenon where “a lexicon’s intended meaning shifts into another category during bootstrapping”(Curran et al., 2007). Although prior studies have been attempting to solve this in multiple different ways (see Section 2.3.2 in Chapter 2), it is still an unsolved problem. This is a hard problem because detecting the shift of meaning is about computing dissimilarity of meaning. And the other side of the coin is to compute similarity of meaning.

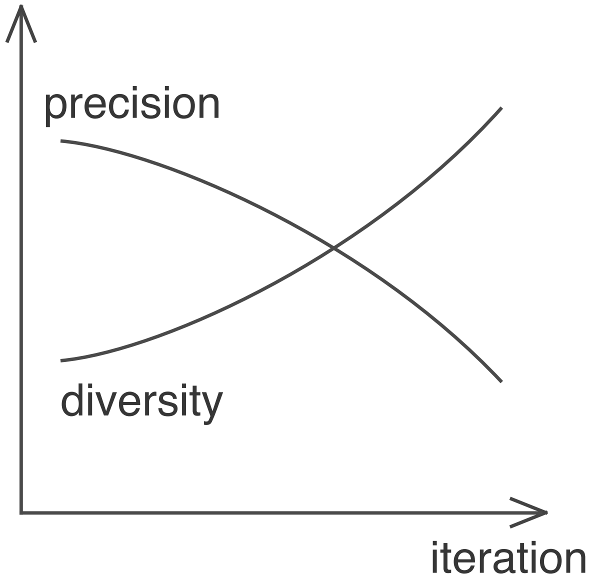


Figure 3.1: An illustration of Precision-Diversity trade-off hypothesis

Formally, using the notations described earlier in this chapter, the meaning shift *ms* of p from a concept c can be denoted as $1 - rel_c(p)$ assuming *rel* returns a normalized value in [0, 1]. Averaging over P, we obtain:

$$ms_c(P) = \sum_{p \in P} \frac{1 - rel_c(p)}{|P|}.$$

Note that *ms* can be modeled very similar to *prec* described in the previous chapter.

$$prec_c(P) = \sum_{p \in P} \frac{[rel_c(p) > \theta]}{|P|}.$$

Formally, under the assumption that above discussion is true, the following stands in the n-th iterations where $n = 0, 1, .., k$:

$$ms_c(p_1) < ms_c(p_2) < ... < ms_c(p_k).$$

$$prec_c(p_1) > prec_c(p_2) > ... > prec_c(p_k).$$

Therefore, it would make sense to assume as bootstrap iterates, semantic drift happens; and when that happens, the degree of semantic drift negatively correlates with precision. Later in Chapter 7, we will analyze precision curves to verify this.

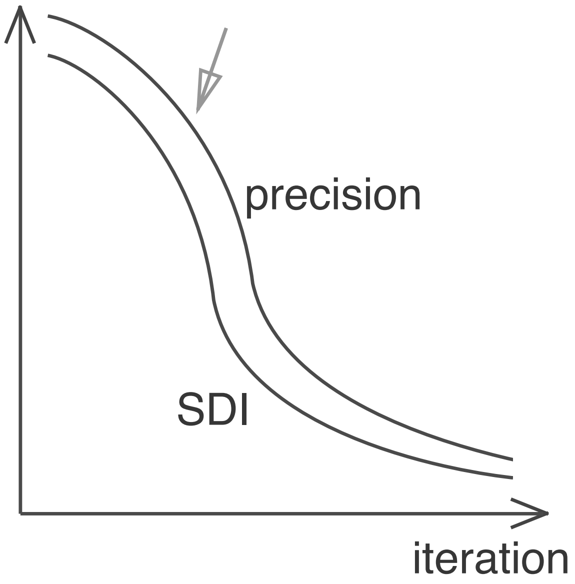


Figure 3.2: A conceptual illustration that shows that Semantic Drift Indicator (SDI) calculated in run-time needs to be able to predict the trend of precision that can be obtained in evaluation time, in order to detect a semantic drift (indicated by an arrow) that suddenly dominates *P*.

The above discussion applies to quantities available at run-time only. In order to detect and prevent semantic drift, a Semantic Drift Indicator must be needed such that correlates well with a precision (see Figure 3.2 that illustrates this).

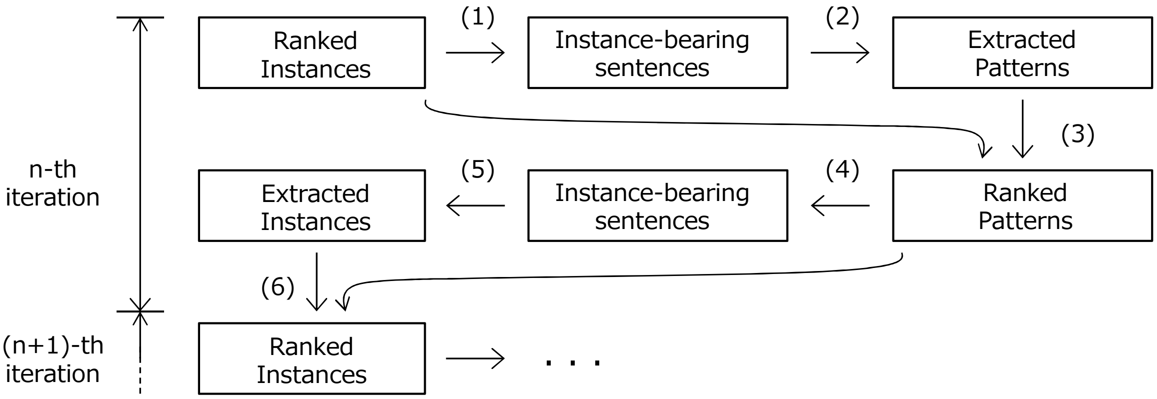
So far, our formalization attempt may be based on oversimplified assumption, however, this level of generality allows us to implement various kinds of *rel* simulation model as well as representations of c and p in order to detect and prevent semantic drift.

By the way, if our assumption that precision and meaning shift have negative correlation is true, we can expect that diversification has a negative impact on precision. Chapter 7 will presents experimental results on the trade-off between semantic drift and diversification in Section 7.4.2 and 7.4.3.

3.2 Paraphrase Acquisition by Bootstrapping

3.2.1 Base Framework: Espresso

Out of multiple different bootstrap learning methods (also see Section 2.3), we chose Espresso framework to base our algorithm on, because of its underlying theory, generality and adaptability. More specifically, the algorithm is modeled under the Distributional Hypothesis which will be discussed later in Section 3.2.2. Patterns and Instances are extracted by a simple "reliability" scoring model which is easily extensible. Also, the model is well-parameterized that one can optimize for adapting to different relations or domains.



(1) Retrieve sentences that contain $I \ni i = \langle X = x, Y = y \rangle$.
(2) Extract and generalize patterns, i.e. contexts of i.
(3) Score and rank patterns based on associations between P and I.
(4) Retrieve sentences that contain P.
(5) Extract instances from the retrieved sentences.
(6) Score and rank instances based on associations between P and I.

Figure 3.3: Overview of the Espresso algorithm. Many bootstrapping learning algorithms work more or less in the same way as described here.

An overview of the framework is illustrated in Figure 3.3. Espresso is a lightly-supervised general-purpose algorithm for acquiring instances and patterns in an iterative fashion. The input to the algorithm is a small number (e.g. between 5 to 20) of seed instances and a corpus. First, the instances are used to retrieve instance-bearing sentences from the corpus. Then the sentences are generalized into a set of longest common substrings, which are seen as patterns. Each pattern is assigned with a reliability score, based on an association with the instances in the corpus. In the *n*-th iteration, top *n* precise patterns with the highest reliability score are selected, and used to retrieve pattern-bearing sentences. These sentences are applied with the patterns to extract even more instances. The reliability score for each instance is calculated in a similar way as the pattern reliability calculation. A few hundred instances with the highest reliability score, together with the original seed instances, are used as the input for the next iteration. Iterations continue until one of convergence criteria is met. This way, we can obtain patterns from instances, and instances from patterns through iterations.

Instance Extraction

Instance is extracted with a constraint of part-of-speech (POS). Specifically, an instance candidate is such a sequence of string that satisfies the following POS requirement (Justesona and Katz, 1995), which is the one traditionally used in the related works:

```
( (Adj|Noun)+ | ((Adj|Noun)* ((Noun)(Prep))?  )  (Adj|Noun)* ) Noun.
```

The original Espresso has a capability of what is called Web expansion and syntactic expansions that addresses lack of redundancy problem in small corpora. However, we disable it for keeping the framework simple so that system behavior can be analyzed with less complication.

Pattern Extraction

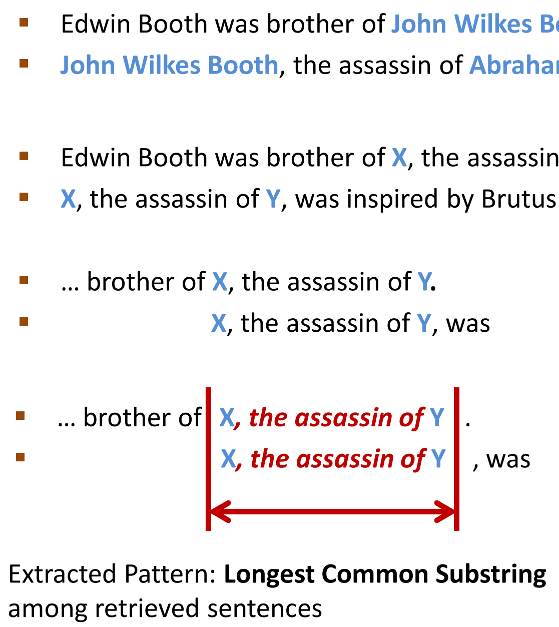
Figure 3.4 shows a concrete example in which instance-bearing sentences are extracted, instances are replaced with slots, and patterns are extracted using the Longest-Common-Substring algorithm.

Reliability: Score for Patterns and Instances

Espresso is unique in a sense that instances and patterns are scored in a principled, symmetric way. Instance reliability $r_\iota(i)$ and pattern reliability $r_\pi(p)$ are calculated as follows.

$$r_\iota(i) = \frac{\sum_{p \in P} \frac{pmi(i,p)}{\max_{pmi}} * r_\pi(p)}{|P|} \tag{3.1}$$

$$r_\pi(p) = \frac{\sum_{i \in I} \frac{pmi(i,p)}{\max_{pmi}} * r_l(i)}{|I|} \tag{3.2}$$



- Edwin Booth was brother of John Wilkes Booth, the assassin of Abraham Lincoln.
- John Wilkes Booth, the assassin of Abraham Lincoln, was inspired by Brutus.

- Edwin Booth was brother of X, the assassin of Y.
- X, the assassin of Y, was inspired by Brutus.

- … brother of X, the assassin of Y.
- X, the assassin of Y, was

- … brother of X, *the assassin of* Y .
- X, *the assassin of* Y , was

Extracted Pattern: **Longest Common Substring** among retrieved sentences

Figure 3.4: Pattern extraction.

Point-wise Mutual Information (PMI), originally proposed by Church and Hanks (1990), is a statistical measure of association between two random variables. PMI in Espresso is calculated as follows:

$$pmi(i,p) \quad = \quad \log \frac{|x,p,y|}{|x,*,y||*,p,*|} \tag{3.3}$$

where the notation $|x,p,y|$ represents the frequency of p with its slots filled with $i = \langle x,y \rangle$, and the notation '*' represents a wild card. $\max_{pmi}$ is the maximum PMI between all combinations of P and I.

Convergence Criteria

The algorithm's iteration stops when a certain criterion is met.

- τ_1: stops when number of patterns extracted is τ_1
- τ_2: stops at k-th iteration if the average pattern score (reliability) drops suddenly:

$$\frac{\frac{1}{|P_{k+1}|}\sum_{p \in P_{k+1}} r_\pi(p)}{\frac{1}{|P_k|}\sum_{p \in P_k} r_\pi(p)} < \tau_2. \tag{3.4}$$

3.2.2 Theory of Distributional Similarity

Meaning similarity in Espresso is modeled as Distributed Similarity, which is based on an important theory called Distributional Hypothesis:

Distributional Hypothesis: Words that occur in similar contexts tend to have similar meanings" (Harris, 1954; Miller and Charles, 1991).

Let us denote the semantic similarity of two objects o_1 and o_2 as $\text{sim}(o_1, o_2)$. In Distributional Similarity, an object o can be represented as a vector $\vec{v} = (e_1, e_2, \cdots, e_n)$, and these vectors are used to calculate the similarity: $\text{sim}(o_1, o_2) \approx \text{sim}(v_1, v_2)$. More specifically, weighted distribution of elements e are contexts associated with o. In a simplest example, elements can be words (or n-grams) that commonly appear with o in a corpus (within a certain distance e.g. N-word window), weights can be given using TFIDF, and similarity can be calculated using a cosine similarity.

Such models are called Distributed Semantic Models or DSM, which have been successfully used in various NLP applications where modeling semantic similarity is a key problem, such as: Query Expansion (Grefenstette, 1994); Part-of-Speech Tagging (Schütze, 1995); Synonymy Detection (Landauer and Dumais, 1997; Littman et al., 2003); Word Sense Disambiguation (Schütze, 1998); Thesaurus Generation and Expansion (Lin, 1998; Pantel et al., 2009; Rapp, 2004); PP-Attachment Disambiguation (Pantel and Lin, 2000); Probabilistic Language Modeling (Bengio et al., 2003).

While traditional works were focusing on building term-oriented co-occurrence matrix (e.g. term-term, term-Ngram, term-document), one interesting recent direction might be in building the matrix of untraditional unit of expressions such as phrases, proper noun instances, plain and structured patterns, etc. Among recent studies, pattern learning works (Bhagat and Ravichandran, 2008; Fujita and Sato, 2008; Lin and Pantel, 2001; Littman et al., 2003; Pantel and Lin, 2000; Pantel and Pennacchiotti, 2006; Veeramachaneni and Kondadadi, 2009) either implicitly or explicitly utilize a variant of the Distributional Hypothesis called Latent Relation Hypothesis (Littman et al., 2003), which states that "similar patterns tend to have similar semantic relations". From the instance-pattern learning point of view, the following two assumptions extended from the Distributional Hypothesis may better fit our settings.

Extended Distributional Hypothesis: Patterns that co-occur with similar pairs tend to have similar meanings (Lin and Pantel, 2001; Turney and Pantel, 2010). For example, according to this theory, if the patterns X *died of* Y and Y *killed* X would both co-occur with similar pairs of concrete entities X and Y in a large corpus, these patterns have similar meanings.

Latent Relation Hypothesis:. According to Littman et al. (2003), "Pairs of words that co-occur in similar patterns tend to have similar semantic relations". For example, according to this theory, if two pairs of words $\langle John\_Lennon, gunshot\_wound\rangle$ and $\langle Bob\_Marley, cancer\rangle$ tend to co-occur in similar patterns X *died of* Y and Y *killed* X etc, then these pairs would have similar semantic relation that Y is the cause of death of X.

3.3 Extending Espresso

In addition to pattern scoring with lexical-diversity (Chapter 5) and distributional type-based instance scoring (Chapter 4), we have made multiple extensions to ESPRESSO in various aspects, as explained in this section.

Instance Extraction via Sliding Window + Dictionary

Our approach of instance extraction extends the method used in Espresso. Again, the task here is to find instances (a.k.a. pairs of anchors, slot values, entity mentions) (e.g. $\langle\langle$John Lennon, bullet wound$\rangle, \ldots\rangle$) given a set of patterns (e.g. $\langle X$ died of $Y \ldots\rangle$") and a corpus (e.g. Wikipedia). Ideally, precision should be favored much more than recall because erroneous instance can lead to Semantic Drift.

In order to extract instances with high precision, we combine POS restriction with dictionary look-up. First, we apply varying-sized windows and validate if the POS constraint is satisfied. Then, we will check if the candidate instance can be found as an entry of YAGO2 Hoffart et al. (2012) database, which contains huge lexicons (i.e. 9.8 million entities from Wikipedia, GeoNames, and WordNet).

The following strategy is used to back-off in case the candidate is not found in YAGO2:

- longest noun phrase found in YAGO2
- longest noun phrase (proper)
- longest noun phrase (general)
- any noun

In Named Entity Recognition (NER) community, sliding window approach (Freitag, 1998) is an earlier work than sophisticated supervised sequential learning approaches that are based on Hidden-Markov Models (HMMs) (Borkar et al., 2001), Conditional Random Fields (CRFs) (McCallum and Li, 2003), and Semi-Markov CRFs (Sarawagi and Cohen, 2004). However, considering the paradigm shift that huge lexical resource such as YAGO2 is available and the precision-oriented nature of the task, we chose to implement the sliding window + dictionary approach (with a back-off to POS-based approach) for extracting instances.

Sentence-based Co-occurrence Statistics

One reason ambiguous or erroneous terms are wrongly ranked higher than others in semantic drift might attribute to a "unreliable" reliability estimation. We introduced the following approach for more precise co-occurrence calculation.

We preprocess a corpus by creating a sentence-based corpus. Each document is decomposed into multiple sentences using a sentence segmenter, and each sentence becomes a "document" to be indexed by an Indri search engine. In this way, we can estimate co-occurrence statistics more precisely by preventing false-positives that are counted by chance.

In order to count occurrences of expressions in a corpus, we use the `dumpindex` tool which is a part of Indri Search Engine (Strohman et al., 2005). By default, Indri does not index symbols, but we need to index them in order to count the occurrence of patterns with symbols, such as "X, who died of *Y*" where a comma follows *X*. For indexing symbols we preprocessed the corpus by replacing all symbols to a fake word, as exemplified in Table 3.1.

Table 3.1: Corpus preprocessing: symbol escape

Symbol	Replaced Strings
Period (.)	`lPERIODl`
Comma (,)	`lCOMMAl`
Apostrophe (′)	`lAPOSl`
Acute accent (`)	`lACUTEl`
Double quote (¨)	`lQUOTl`
Left parenthesis (()	`lLPARENl`
Right parenthesis ())	`lRPARENl`
Hyphen (-)	`lDASHl`
Percent (%)	`lPERCENTl`

The query sent to Indri is formulated using phrase operators. An ordered window #*N* (matches terms that occur with $N-1$ skips allowed between terms; order sensitive) and an unordered window operator #uw*N* (matches terms that occur within a window size of *N*, in any order). Below shows two example queries to calculate the co-occurrence between a pattern and an instance.

```
P: X (d. Y
I: [Liu Bei, 223]
count(P)   =   50347
    #1( lLPARENl d lPERIODl )
count(I)   =   36
    #uw20( #1( Liu Bei ) #1( 223 ) )
count(P&I) =   20
    #1( #1( Liu Bei ) #1( lLPARENl d lPERIODl ) #1( 223 ) )
PMI (discounted) = 8.4428

P: murder of X in Y
I: [John Lennon, 1980]
count(P)   =    1024
    #5( #1( murder of ) in )
count(I)   =    116
    #uw20( #1( John Lennon ) #1( 1980 ) )
count(P&I) =    2
    #1( #1( murder of ) #1( John Lennon ) #1( in ) #1( 1980 ) )
PMI (discounted) = 6.4251
```

Pattern Filtering

We filter out patterns that are too specific. Specifically, if a pattern includes numbers and proper nouns, it will be removed from a candidate. We detect proper nouns by a simple heuristic of whether a pattern contains uppercase letter or not. Although this filtering wrongly removes "legitimate" patterns with an uppercase such as "*X* is the CEO of *Y*", we think it is important to remove too specific patterns. The motivation behind this filtering is the lack in generality; these patterns can capture only a very limited instances. Without this filtering these, the algorithm would be trapped in a suboptimal situation where iteration converges with a lot of very similar patterns with small lexical diversity. Example of too-specific patterns are shown in Table 3.2.

Table 3.2: Example of too specific patterns

X, died of Y in Honolulu
X who died of Y, for a total of 25
X who died of Y, for a total of 269
X who died of Y, for a total of 29
X who died of Y, for a total of 246

On the other hand, we also made sure that too general patterns are not accepted either. In our definition, too general patterns are the ones consisting of stop-words and symbols only (no content word). Table 3.3 shows example of these patterns actually filtered.

Table 3.3: Example of too general patterns

Y, with his X	X had Y	X with Y	X of Y
X - Y	X and Y	Y of X	X has Y
Y and X	Y and her X	Y, (X	X, (Y
Y than X	Y in his X	X; of Y	Y, and her X
Y among his X	Y, and X	X, from Y	X by Y
X for Y	Y among X	X's (Y	X, Y
Y of his X	X" of Y	Y, and his X	Y, X
Y, was X	Y and his X	X from Y	Y in X
Y, as did his X	X about Y	Y, his X	X to Y
130 X by Y	X's Y		

Instance Filtering

We filter general instances such as pronouns (e.g. "he", "him") unless it is part of the original seed. This is because these words are too ambiguous, increasing a risk of semantic drift. For example, an instance pair ⟨ "he", "bullet" ⟩ is bad for the CAUSE_OF_DEATH relation because there exists multiple possible relations between these two instances (e.g. *died-of*, *fired*, *purchased*, etc). This pair of instances do not restricting relations well, as compared to concrete nouns. Therefore various irrelevant patterns may be introduced if these instances are used (semantic drift).

We also filter out instances based on type constraint which is softly defined as a vector from the original seed instances. An instance is filtered if the average type similarity of the two anchors are below a certain threshold. We experimentally set the threshold value to be 0.3. The details of this algorithm is described in Chapter 4.

3.4 Characteristics

This section discusses characteristics of the paraphrase extraction algorithm or the paraphrases extracted using the algorithm, including both pros and cons.

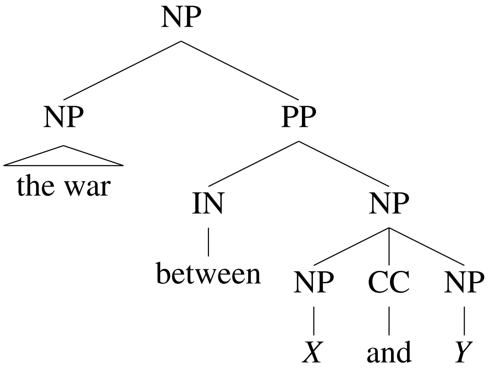
3.4.1 Pattern Representation

Paraphrase or patterns have been represented in various forms in previous studies.

Unstructured v.s. structured

A paraphrase pattern can be represented either as simple unstructured patterns made of surface texts, or complicated structured patterns represented with syntactic constraints. There is pros and cons in each representations.

- **Key context outside variables.** In the binary argument pattern extraction problem, structured-pattern approach extracts a pattern which is the shortest path between X and Y. For example, given a variable-replaced sentence "the causes of the war between X and Y.", structured-pattern based approach extracts only the structure "(NP (NP X) (CC and) (NP Y))" which misses the important keyword *war*.

  On the other hand, unstructured approach with Longest-Common-Substring extraction approach that we adapt will be able to find patterns such as $\langle$ a war between X and Y, X is fighting against $Y, \ldots \rangle$.

- **Parse Errors.** Since parsing cannot be done in 100% accuracy with the state-of-the-art techniques, approaches using structured patterns can be affected by parsing errors in both learning time and application time (Wang et al., 2009). When the target domain and corpus is different from a typical corpus that parsers are trained on, such as social media corpus with a frequent spelling and grammatical errors, or medical corpus with many out-of-dictionary technical terms, or legal corpus with longer sentences, there would be more chance of parse errors. In these domains, approaches that do not use deep linguistic analysis might be more appropriate.

- **Complexity and Usability.** Unstructured patterns are easy to compute, verify, and apply to text. For instance, suppose we will calculate an occurrence of a pattern in a corpus, a search engine such as Indri (Strohman et al., 2005) can straightforwardly do it with a built in module *dumpindex xcount*. On the other hand, patterns can be represented in more complicated structured way e.g. `X.N:subj:V<kill>V:obj:N>people>N:nn:Y.N`, where syntax and super type restriction are added. As more restrictions are added to a pattern, it becomes more sensitive to syntactic / semantic distinctions. One of disadvantages of complicated patterns is that they are more likely to suffer from data sparseness. In other words, they are observed less in corpus and thus harder to observe.

- **Dependency.** Structured patterns may be able to capture longer dependencies between words. It is often the case that adjectives, adverbs and appositive noun phrases are used to modify other words.
- **Clarity.** Paraphrase rules represented in surface text pattern are easy for humans to verify. If their representation is complicated (e.g. tree structure, feature-structure, bag-of-words), it is harder to tell their validity. This objective is important in using paraphrase for automatic evaluation of text outputs against gold standard snippets. For example, in Machine Translation (MT) evaluation paraphrase-supported metrics have been proposed to fill in gaps between different surface texts representing same meaning (Padó et al., 2009; Zhou et al., 2006a). According to the NIST MetricsMATR, an evaluation forum of automated metrics developed for the purpose of evaluating MT technology, *Intuitive Interpretation* is one of objectives "missing from current automated MT metrics" (Przybocki et al., 2009). If complicated paraphrase rules are used, the following may not be satisfied: "a complaint levied against most current automatic MT metrics is that a particular score value reported does not give insights into quality, nor is it easy to understand the practical significance of a difference in scores" (Przybocki et al., 2009).

Number of Arguments

Number of arguments in a pattern can vary from unary argument, binary argument to n-ary arguments. Consider intransitive verbs which take only one complement. Unary argument pattern is the right form for these verbs, as seen in works such as Komachi and Suzuki (2008). However, unary argument means a pattern has only slot; given sentences with at least one slot filling instance, it is hard to differentiate meanings in sentences, as there is a small constraint. To this end, one may want to restrict a pattern occurrence by modeling the context of the pattern. For instance, Kanayama and Nasukawa (2006) modeled intra- and inter-sentential contexts for a unary-argument pattern. N-ary argument patterns where $N \geq 3$ may be useful for events where multiple arguments such as person, location, time are involved. However, they are harder to acquire since it is rarer that all N instances are observed in a sentence. Binary-argument patterns are in a good balance in terms of ambiguity and chance of observation in a corpus, and therefore often studied by previous works (Lin and Pantel, 2001; Pantel and Pennacchiotti, 2006; Szpektor et al., 2004). Given these reasons, this thesis work focuses on binary-argument patterns. In other words, we do not target intransitive verbs which take only one argument (subject; e.g. *X* passed away), and ditransitive verbs which take three arguments (subject and two objects; e.g. *X* gives *Y* to *Z*).

3.4.2 Saving Human Effort

Return-on-investment is an important measure in practice; ideally, more output should be obtained from a system with less human effort. Humans are costly, becomes tired, need to be provided with a tool with a good usability, and may need to have a domain expert knowledge. Thus, in this work, we aim to come up with a data-driven computational approach with minimum human intervention at batch time in advance on seed development. And that is why we describe our work as weak-supervision, minimal-supervision or light-supervision.

Chapter 4

Preventing Semantic Drift

Section 2.3 in Chapter 2 presented semantic drift is a fundamental problem in bootstrapping yet to be solved. In this chapter, we will discuss the problem of measuring similarity of open domain entities, which include both general and proper nouns.

In traditional closed-domain Information Extraction problem, only small number of types (e.g. PERSON, LOCATION, ORGANIZATION) can be extracted. Given an instance candidate (preferably with a context), an NER system detects one of types, or no type. Therefore, NER in closed domain limits its applications. Although, an ideal system should be able to deal with wide range of open types, it is challenging to build such an NER system, as observed from the state-of-the-art NER research.

We propose a novel method that calculates semantic similarity between a set of instances and an instance (which can be either of general or proper noun). Given a set of seed instances $I = \{i_1, \ldots, i_n\}$ and an unseen instance candidate i_u, our method quantifies how likely i_u can belong to I.

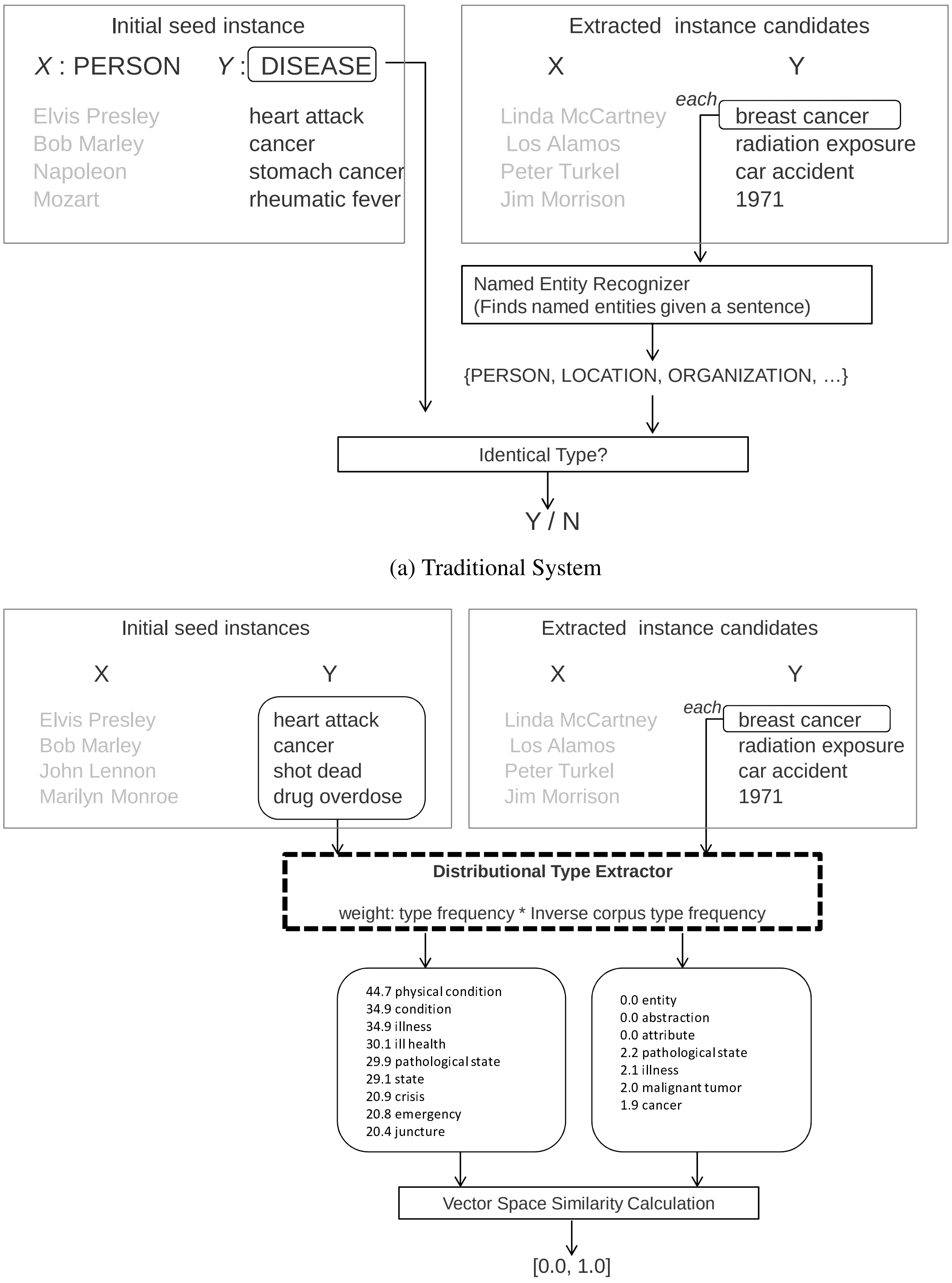
Our approach constructs vector space representations v_s and v_u from I and i_u respectively, where each element represent a conceptual node in WordNet. Then $similarity(I, i_u)$ is calculated as the similarity of the vectors $similarity(v_s, v_u)$.

The role of type similarity calculator in paraphrase extraction problem is illustrated in Table 4.1 where traditional and proposed systems are contrasts.

4.1 Introduction

Suppose there is an incomplete set of *seed* entities {“heart attack”, “pneumonia”, “drowning”} that have one thing in common – they can be a *cause of death*. Given a new entity (e.g. “cancer”, “gunshot wound”) that is not in the original seed set, we aim to algorithmically quantify how appropriately this entity can be added to the *seed* entities. In other words, our goal is to measure semantic similarity between the new entity (hereafter called the “unseen” entity) and the seed entities.

This is a challenging problem even with a large-scale lexical database WordNet Miller (1995). For example, see Figure 4.2 for the hypernym trees of some *cause of death* words. Types for these words are located in different places over the tree rather than being clustered closely.



(a) Traditional System

(b) Proposed System

Figure 4.1: Detecting an invalid instance using type information: traditional and proposed models.

For WordNet Similarity metrics that rely on hypernym trees, this is a problematic example that would result in a low score. Another issue is that proper nouns (e.g. “Motor Neurone Disease”, “Dettol”) or some noun phrase (e.g. “congestive heart failure”, “fractured skull”) are often not found in WordNet. Moreover, some general nouns (e.g. “overdose” and “drawning”) are only found as verb in WordNet, that has completely different hypernym tree structure as noun’s. WordNet-based Semantic Similarity algorithms (Banerjee and Pedersen, 2002; Hirst and St-Onge, 1998; Jiang and Conrath, 1997; Leacock and Chodorow, 1998; Lin, 1998; Resnik, 1995; Wu and Palmer, 1994) more or less have these issues in common.

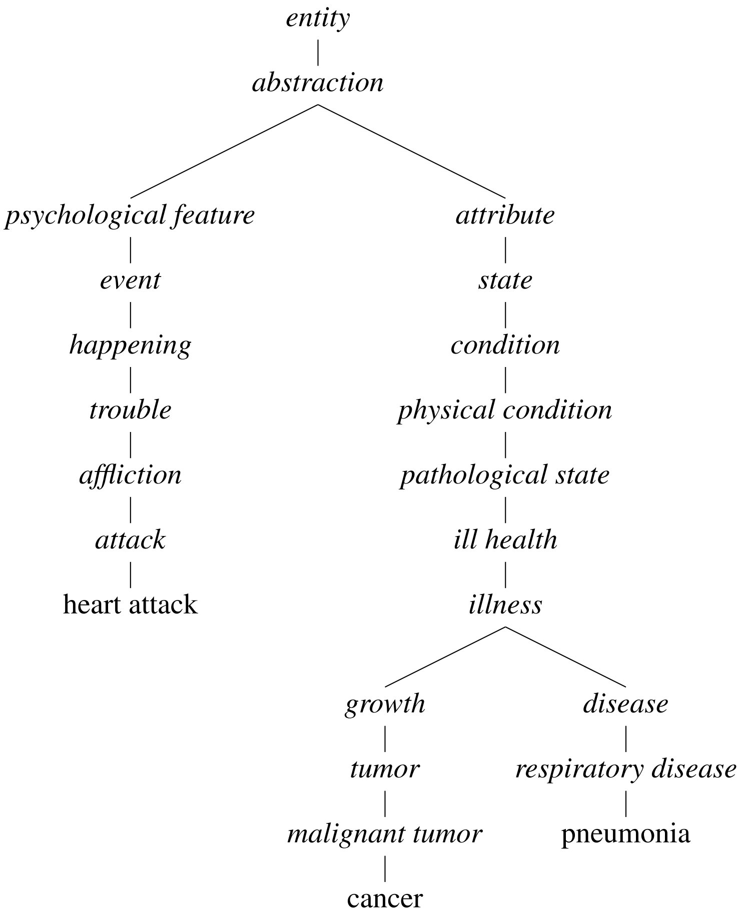


Figure 4.2: Hypernym tree from WordNet where higher-level types are abstract and shared among multiple entities, while lower-level ones are more concrete and distinctive. Notice that “heart attack” and “cancer” are located far away in the taxonomy, even though they are similar in terms of possibility to become a *cause of death*.

To address these issues in previous works, we will propose a similarity algorithm that models entities as vector space of types weighted in pseudo-TFIDF. Our approach can be applied to an arbitrary group of entities (e.g. *cause of death*) without being limited by a taxonomy tree design.

The experimental outcome shows that proposed approach with pseudo-TFIDF weights results in 97.0% in Precision@200, which is statistically significant over 84.0% from the Baseline model using co-occurrence statistics (n=200, p-value<0.01).

4.2 Proposed Method

Given a set of incomplete “seed” entities $E_s = \{e_{s1}, \ldots, e_{sn}\}$ and an unseen entity e_u, we are concerned about whether e_u can be another element of E_s to make it more complete set. Our goal is to come up with a function $sim(e_u, E_s)$ that can quantify how appropriately e_u can belong to E_s.

The reason why a numeric valued similarity score is ideal is because we can rank and prioritize unseen entities using them. This way, we can have a huge number of unseen candidate entities, select up to a certain number of entities. We can also classify an entity into positive/negative classes if we learn a threshold from a held out dataset.

The proposed method takes an approach in the vector space model of types. In other words, the method computes the appropriateness that an unseen entity being a valid candidate, using the similarity between type vectors.

In summary, similarity function $sim(e_u, E_s)$ is computed as follows. First, let us create vector $\mathbf{v_u}$ and $\mathbf{v_s}$ from e_u and E_s respectively. Then, calculate $sim(e_u, E_s)$ from the two vectors. We will discuss details in the following subsections.

4.2.1 Creating a Vector of Types

Consider a function *type* that returns a bag of types associated with an entity. Using the function, a vector $\mathbf{t}$ with binary score 0, 1 is obtained: $\mathbf{t} = type(e)$.

A vector $\mathbf{v_u}$ is simply created from e_u as: $\mathbf{v_u} = \mathbf{t_u} = type(e_u)$.

On the other hand, a set $E_s = \{e_{s1}, \ldots, e_{sn}\}$ is converted to vector space of types in a similar fashion: $T_s = \{\mathbf{t_{s1}}, \ldots, \mathbf{t_{sn}}\}$. Finally, a vector space for the seed entities is created as $\mathbf{v_s} = \sum_{i=1}^{n} \mathbf{t_{si}}$.

Pseudo-TFIDF weight

It is reasonable to give more weight to types that are concrete (e.g. *disorder*, *emergency*) than abstract (e.g. *abstraction*, *entity*)[1]. In order to realize this, we further extend the weighting model by introducing an idea similar to TFIDF used in Information Retrieval: $tf.idf = tf * \log(\frac{D}{1+df})$. TF in our case is type frequency; DF is frequency of a type in the lexical database; and D is the total number of types.

Example Weighted Vectors

Table 4.1 shows actual type distribution $\mathbf{v_s}$, sorted in descending order by its weight (frequency and pseudo-TFIDF and respectively), where the seed entities are Ys in Table 4.3. In Table 4.1 (a), types are weighed by the frequency. For example, *event* has a frequency of 10, which means this type has been observed 10 times out of 12 seed entities. Note that highly-weighted types are

[1] See Figure 4.2 where higher level types are too abstract to distinguish entities.

abstract in Table 4.1 (a). On the other hand, top types in (b) are observed to be representative types of the seeds.

Weight	Type
12	*abstraction*
12	*entity*
11	*psychological_feature*
10	*event*
10	*yagoPermLocEntity*
9	*cognition*
8	*physical_condition*
8	*condition*
8	*ability*
8	*state*
8	*attribute*
8	*medium*
8	*instrumentality*
8	*artifact*
8	*whole*

(a) type frequency

Weight	Type
67.4	*emergency*
67.4	*crisis*
67.3	*juncture*
59.6	*physical_condition*
56.4	*condition*
49.3	*ability*
48.7	*state*
47.8	*illness*
46.4	*ill_health*
46.4	*pathological_state*
44.8	*attribute*
43.5	*cognition*
42.3	*happening*
39.7	*know-how*
39.7	*method*

(b) pseudo-TFIDF

Table 4.1: Seed entity type vector where elements are sorted by two different weighting algorithms: (a) type frequency (b) pseudo-TFIDF. Only top 15 elements are shown.

4.2.2 Vector Space Model of Types

Using $\mathbf{v_u}$ and $\mathbf{v_s}$ from the above process, $sim(e_u, E_s)$ is calculated using the cosine similarity: $sim(e_u, E_s) = cos\theta(\mathbf{v_u}, \mathbf{v_s})$. Cosine similarity is a well-known method in Information Retrieval for calculating similarity between two weighted vectors $\mathbf{x}$ and $\mathbf{y}$: $cos\theta(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|} = \frac{\sum_{i=1}^{n} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{n} (x_i)^2} \cdot \sqrt{\sum_{i=1}^{n} (y_i)^2}}$, where θ is the angle between the two vectors in a n-dimensional vector space. Since $cos\theta$ takes a real value between 0 and 1 for positive-valued vector, $0 \leq sim(e_u, E_s) \leq 1$.

4.2.3 Type Knowledge Base: YAGO2

An ideal type look-up resource would be large enough to cover millions of Named Entities not just general words, and support various kinds of types. WordNet version 3.0 contains 155K (nouns: 118K) words and 118K (nouns: 82K) synsets, which lacks coverage of proper nouns in our problem.

In this regard, the type knowledge base we used is YAGO2 Hoffart et al. (2012). YAGO2 is a huge database that contains 9.8 million entities from Wikipedia, GeoNames, and WordNet.

Types are mostly defined over WordNet synset (concept) which varies in hundreds of thousands. Each entity is associated with a synset node in WordNet which enables one to look up types and create a type vector.

For instance, given an entity "heart attack", we can obtain all the types from the YAGO2 database as shown in Table 4.2.

abstraction	*instrumentality*
affliction	*juncture*
album	*medium*
artifact	*musical_organization*
attribute	*object*
cardiovascular_disease	*organization*
condition	*physical_condition*
crisis	*physical_entity*
disorder	*psychological_feature*
emergency	*social_group*
entity	*state*
event	*trouble*
failure	*whole*
group	*yagoLegalActor*
happening	*yagoLegalActorGeo*
heart_disease	*yagoPermLocEntity*

Table 4.2: Exhaustive set of types associated with "heart attack" in YAGO2.

4.3 Experiment

This section presents experimental design and results. First, we built an evaluation dataset by extracting entities from an unstructured corpus in a way that does not consider type. Then, using the proposed similarity model, we ranked entities and evaluated them. The following explains the details.

4.3.1 Building Evaluation Dataset

This subsection explains the way we built labeled entities to be evaluated. Entities are from a binary-relation instance acquisition algorithm (Pantel and Pennacchiotti, 2006). As input, it takes an unstructured Wikipedia corpus and a set of seed entities, or pairs of X and Y where X's cause of death is Y (see Table 4.3)[2].

Table 4.3 shows the seed entities used to acquire more entities. Notice that not all of them belong to one specific type such as *disease*. We focus on evaluating *Y* as it is more challenging.

[2] The reason we focus on this particular relation is because it can contains various types across lexical network. In the camera ready version, we will present results on a few more relations.

X	Y
Elvis Presley	heart attack
Bob Marley	cancer
Richard Feynman	cancer
Napoleon	cancer
Janis Joplin	overdose
Ronald Reagan	pneumonia
Mozart	rheumatic fever
Marilyn Monroe	overdose
Michael Price	Carbon Monoxide Poisoning
Malcolm Hale	Carbon Monoxide Poisoning
Akulina	hunger
Spr Jerkins	drowning

Table 4.3: Exhaustive list of seed entities used in the experiment. Each pair of X and Y are in *cause-of-death* relation where the following holds: X died from a cause Y.

ESPRESSO algorithm (Pantel and Pennacchiotti, 2006) is used to extract entities in the following way. First, all the sentences are retrieved from the Wikipedia corpus such that contains both X and Y within a sentence. Then longest common substrings among those extracted sentences are distilled. At this point, patterns such as “X suffered minor Y”, “X’s early death from Y”, ”X , who passed away from Y” etc are obtained. These candidates are ranked by co-occurrence statistics called Pointwise Mutual Information (PMI) (Church and Hanks, 1990; Cilibrasi and Vitanyi, 2007), often used in distributional semantics, against seed entities $\langle X, Y \rangle$. As a result, a pattern “X died from Y” was selected as the most reliable entity extraction pattern. This pattern is used to search in the corpus, finding pairs of noun phrases that fill in the slot X and Y. Note that this process does not take the type information into account. They are extracted as long as they satisfy the following POS sequential requirement (Justesona and Katz, 1995) that are traditionally used in the past works: `( (Adj|Noun)+ | ((Adj|Noun)* (NounPrep)? ) (Adj|Noun)* ) Noun.`

PMI between entity pair $i = \langle x, y \rangle$ and a pattern p is calculated as follows, where $|e|$ represents number of sentences that includes the expression e in the corpus.

$$pmi(i,p) = \log \frac{|x,p,y|}{|x,*,y||*,p,*|}$$

Labeling

Entities are binary-labeled by human as to whether it can be a cause of death. Below shows a sample of positive and negative labeled items.

Examples of typical positive-labeled entities are as follows: **disease or health problem** (Motor Neurone Disease; alcohol overdose; starvation), **accident** (traffic accident; lawn mower; fight; fire), **indirect cause of death** (overwork; curse; shame; Winter Quarters), **idiom** (own hand), and

phrase not typically found in lexical resources (week-long series of air raid; well-aimed rifle shots). On the other hand, negative-labeled ones are: **wrong type** (1975; Chugoku-Shikoku Public Works Office), **unobvious cause of death** (enemy; procedure; police)[3], **entity extraction error** (bone; abdominal; combination)[4], **corpus preprocessing errors** (hernias.Title; unknown disease .Fear), and **typo** (pnumonia; Hogkins Disease)[5].

Data Statistics

As the result of dataset development described in Section 4.3.1, there are 1,949 entities found where 720 of them are unique. After label annotation, 618 received positive and 102 received negative label.

When the type knowledge source does not find an entity, a lemma from Morpha (Minnen et al., 2001) is given as a back off approach. Out of 720 unique entities, 392 received a type vector from the YAGO2 database (cf. WordNet found a word entry and its hypernym for only 233 entities [6]).

On average, each entity are assigned with 10.7 types (19.6 types for those found in YAGO2).

4.3.2 Evaluation Results

There are three similarity methods used in the experiment: *Baseline* (ESPRESSO's reliability score based on PMI), *Proposed A* and *Proposed B* each with type frequency and pseudo-TFIDF weight, respectively. As discussed in Section 4.2.1, the latter proposed approach was expected to result in better. We applied the three methods over the 720 *unseen* entities described in Section 4.3.1. The scores are used to rank the entities which are evaluated with respect to Precision@N. Since N=200 is also the number used to extract entities in ESPRESSO (Pantel and Pennacchiotti, 2006), we mainly evaluate the methods with N=200.

Precision@N is a number of correctly classified entities out of top-N system prediction. For example, Precision@200 = 80% means top 200 out of 720 are ranked then selected, and 80% (=160) of them are judged as correct.

Table 4.4 shows the experimental result. Null hypothesis that the Baseline and the Proposed B has same distribution of Precision@200 is rejected with a statistically significant difference (confidence threshold $p<0.01$). As a result, we could conclude that Proposed B method is capable of scoring entity similarity better than the Baseline does.

Table 4.5 contains sample entities ranked by the Baseline and the proposed B algorithm, respectively.

[3] Imagining how *overwork* can be a cause of death would be considered directly connected, however, there could be various ways that *enemy* leads to someone's death.

[4] For example, an original text might have been *bone cancer* but only *bone* is obtained due to an error in entity extraction.

[5] pneumonia and Hodgkin are the correct spellings.

[6] Example entities found in YAGO2 but not in WordNet are following proper nouns: "Motor Neurone Disease" and "Dettol"; and noun phrases: "congestive heart failure" and "fractured skull"

Prec@N	Baseline	Proposed A	Proposed B
N=10	90.0%	100.0%	100.0%
N=20	90.0%	100.0%	100.0%
N=50	90.0%	96.0%	98.0%
N=100	84.0%	88.0%	***99.0%**
N=200	84.0%	88.0%	***97.0%**
(N=720)	85.8%	85.8%	85.8%

Table 4.4: Experimental results measured in Precision@N. *Statistical significance has been observed for the Proposed method against the Baseline method with a confidence threshold $p<0.01$.

Score	Entity
0.141	contracting SARS
0.139	violent murder
0.124	atrocities
0.122	thyroid cancer
0.108	friendly fire
0.105	multiple sclerosis
0.104	multiple myeloma
0.104	anxiety
0.098	trauma
0.097	apparent cerebral hemorrhage at age

(a) Baseline

Score	Entity
0.808	drowning
0.767	cancer
0.739	cardiac arrest
0.739	ventricular fibrillation
0.739	sudden cardiac arrest
0.701	bullet wound
0.701	gunshot wounds
0.701	gunshot wound
0.675	blunt trauma
0.675	internal bleeding

(b) Proposed B

Table 4.5: Top-10 ranked entities by the Baseline and the Proposed B algorithm. Note that similarity scores are not directly comparable between (a) and (b).

4.4 Discussion

Related Works. In traditional closed domain Named Entity Recognition problem, only small number of types are supported. For instance, CoNLL 2003 English NER dataset deals with only four entity types: *person*, *location*, *organization* and *miscellaneous*.

Super-sense tagging is a similar but much more challenging problem that aims to annotate entities in a text with labels defined as WordNet classes. Both require sufficient amount of context as input.

The problem we aimed to solve is common, and thus has potential to be used in applications such as Set Expansion, Machine Reading (Carlson et al., 2010a; Wang and Cohen, 2007) and Bootstrap Instance/Paraphrase Acquisition (Shima and Mitamura, 2012).

There are other lexical resources such as FrameNet (Baker et al., 1998) and VerbNet (Kipper et al., 2006), but we used YAGO2 because it has much more coverage in lexicon.

Language Independence. Provided that a type knowledge source is available, the proposed approach essentially works language-independently. Since types in Yago2 are mostly obtained automatically from Wikipedia, we may be able to expect that a similar resource will be available for other languages.

Possible Application in Social Media. Since the our focus is open-domain, the proposed model is expected to satisfy industrial needs and work in practical problems involving big data. If the target domain is social media and tagged data is available via folksonomy, we could use tags instead of types to classify entities. This is where we can take advantage of open-type nature of the proposed work. For example, possible entities could be restaurant in restaurant review dataset; photo in online album; video in video-sharing website; or web page in social bookmark site.

Reproducibility. This work is highly replicable. Type knowledge resource we used is publicly available. Code and labeled evaluation data will be available to anyone once the paper is accepted.

Out-of-dictionary issue. Out-of-type-dictionary is a critical issue in a dictionary-based approach. In addition to lemmatization where we try to normalize inflected variations to increase a chance of finding a lexical item in a dictionary, it would be ideal if a system can support other back-off approaches, such as extraction of a head of a phrase (e.g. “bullet wound” to “wound”, “bombing injury” to “injury”) that hopefully increase a chance to find the item in the dictionary.

4.5 Summary

We proposed an approach to measuring semantic similarity among open domain entities. The approach addresses the weakness of WordNet based approach. As experimentally shown in Section 4.3.2, the proposed method is more precise at measuring similarity than a co-occurrence statistics algorithm. The results of using this method when incorporated into a bootstrap paraphrase extraction system will be presented in Chapter 7.

Chapter 5

Diversifying Lexicons in Paraphrase Bootstrapping

In Section 2.2.3 and 2.4.1 (Chapter 2), we discussed that previous research falls short in extracting paraphrase patterns with lexical diversity. To solve this problem, we will propose a novel technique called "Diversifiable Bootstrapping Model" which can control lexical diversity in a minimally-supervised iterative paraphrase acquisition process.

5.1 Diversifiable Bootstrapping

We propose *Diversifiable Bootstrapping*, a lexical diversification extension to general bootstrapping language acquisition methods which addresses the Diverse Paraphrase Acquisition Problem.

Let us use $r_\pi(p)$ to denote an original score of a pattern p that is used as a criterion for pattern ranking at each iteration. The proposed diversification model generates an updated score $r'_\pi(p)$ by taking into account a diversity score as a linear combination:

$$r'_\pi(p) = \lambda \cdot r_\pi(p) + (1 - \lambda) \cdot diversity(p) \tag{5.1}$$

The parameter λ, a real number ranging between [0,1], is used to interpolate the original score with the diversity score. In other words, by tweaking this parameter, patterns to acquire can be *diversifiable* with a specific degree one can control. When $\lambda = 1$, the score is unchanged from the original: $r'_\pi(p) = r_\pi(p)$. As a smaller λ is given, the more diversity score takes effect. Both $r_\pi(p)$ and $diversity(p)$ should range between [0,1], so that their linear interpolation $r'_\pi(p)$ also takes the same range.

5.1.1 Diversity function

We experimentally designed the diversity scoring function, the second term in Eq. (7.2), based on the D algorithm from Shima and Mitamura (2011) (see Algorithm 2[1]).

[1] The algorithm notation and grade range are slightly modified from the original one so that it fits to our problem.

The algorithm can measure the lexical diversity in a set of patterns. Input to the D function is a set of patterns that are sorted in the descending order with respect to the original score $r_\pi(p)$. Output from the function is a set of numeric grades which represent how much a pattern is lexically novel as compared to patterns ranked higher than that. The *extractContentWords* is a function that takes a string as input and outputs a set of content words[2]. For example, given "*X* shot *Y* to death" as input, *extractContentWords* is expected to return {"shot", "death"}. The *stemWords* is a function that takes a set of content word strings as input, and outputs a set of stemmed content words such that derivational morphological differences are normalized. For instance, given {"killing", "killed"}, *stemWords* is expected to return {"kill", "kill"}. We use the Porter algorithm (Porter, 1980) for obtaining a stemmed form of a word.

Algorithm 2 D score calculation

```
Input: patterns $p_0, \dots, p_n$
Output: $D$ array indexed by $1 \dots n$
  Set $history1 \leftarrow$ extractContentWords($p_0$)
  Set $history2 \leftarrow$ stemWords($history1$)
  $D[0] \leftarrow 2$
  for $i = 1 \rightarrow n$ do
    Set $W1 \leftarrow$ extractContentWords($p_i$)
    Set $W2 \leftarrow$ stemWords($W1$) // stemming
    if $W1 = \emptyset$ OR $W1 \cap history1 \neq \emptyset$ then
      $D[i] \leftarrow 0$ // word already seen
    else
      if $W2 \cap history2 \neq \emptyset$ then
        $D[i] \leftarrow 1$ // word's root already seen
      else
        $D[i] \leftarrow 2$ // unseen word
      end if
      $history1 \leftarrow W1 \cup history1$
      $history2 \leftarrow W2 \cup history2$
    end if
  end for
```

The diversity function is given as:

$$diversity(p_k) = \frac{D[k]}{2} \cdot r_\pi(p_0) \tag{5.2}$$

where the value from the D function is normalized into the range between [0,1]. It is also multiplied with the highest original score, in order to have a comparable magnitude of value as the first term. As a result, given p_0,

$$r'_\pi(p_0) = \lambda \cdot r_\pi(p_0) + (1 - \lambda) \cdot r_\pi(p_0) = r_\pi(p_0).$$

[2] A content word is a word that has a meaning (e.g. "eat" and "apple"), which can be contrasted with a function word that serves a grammatical role (e.g. "the" and "in").

Table 5.1: The exclusive list of seed instances for each relation.

(a) *killed*

X	*Y*
Nathuram Godse	Mahatma Gandhi
John Wilkes Booth	Abraham Lincoln
Yigal Amir	Yitzhak Rabin
John Bellingham	Spencer Perceval
Mohammed Bouyeri	Theo van Gogh
Mark David Chapman	John Lennon
Dan White	Mayor George Moscone
Sirhan Sirhan	Robert F. Kennedy
El Sayyid Nosair	Meir Kahane
Mijailo Mijailovic	Anna Lindh

(b) *died-of*

X	*Y*
Elvis Presley	heart attack
Bob Marley	cancer
Richard Feynman	cancer
Napoleon	stomach cancer
Janis Joplin	drug overdose
Ronald Reagan	pneumonia
Mozart	rheumatic fever
John Lennon	shot dead
Marilyn Monroe	drug overdose

(c) *was-led-by*

X	*Y*
India	Rajiv Gandhi
Australia	Paul Keating
Vichy France	Marshal Petain
United Kingdom	Elizabeth II
Cuba	Fidel Castro
Microsoft	Bill Gates
Uganda	Idi Amin

Table 5.2: Top 15 (out of hundreds or thousands) ranked list of paraphrases acquired by *Diversifiable Bootstrapping* are shown, after the 5th iteration. When a smaller λ was specified, the method preferred a pattern that gave more lexical diversity. When the lexical diversification was disabled ($\lambda = 1.0$), the patterns tended to have syntactic and morphological diversity.

(a) *killed*

$\lambda = 1.0$	$\lambda = 0.7$	$\lambda = 0.3$
X , the assassin of Y	X , the assassin of Y	X , the assassin of Y
assassination of Y by X	X assassinated Y	X , who killed Y
X assassinated Y	assassination of Y by X	Y was shot by X
the assassination of Y by X	Y was shot by X	X tells his version of Y
of X , the assassin of Y	X , who killed Y	X shoot Y
X assassinated Y in	the assassination of Y by X	X murdered Y
X , the man who assassinated Y	X assassinated Y in	Y 's killer, X
Y 's assassin, X	X tells his version of Y	Y , at the theatre after X
of Y 's assassin X	X shoot Y	Y , push X to his breaking point
of the assassination of Y by X	X murdered Y	X assassinated Y
X shot and killed Y	Y 's killer, X	assassination of Y by X
Y was assassinated by X	Y , at the theatre after X	X to assassinate Y
named X assassinated Y	Y , push X to his breaking point	X kills Y
Y was shot by X	X to assassinate Y	of X shooting Y
X to assassinate Y	of X , the assassin of Y	X assassinated Y in

(b) *died-of*

$\lambda = 1.0$	$\lambda = 0.7$	$\lambda = 0.3$
X died of Y	X died of Y in	X died of Y in
X died of Y in	X died of Y	X 's death from Y
X died of Y on	X 's death from Y	X passed away from Y
X died of lung Y	X passed away from Y	Y of X , news
X died of lung Y in	Y of X , news	Y of X , a former
X died of lung Y on	Y of X , a former	that X was suffering from Y
X died of Y in the	that X was suffering from Y	the suspected Y of X
X died of Y at	the suspected Y of X	X succumbed to lung Y
X died of stomach Y	X to breast Y in	X to breast Y in
X died of natural Y	X was diagnosed with ovarian Y	X was diagnosed with ovarian Y
X died of breast Y in	X dies of Y	X dies of Y
X died of a Y	X was dying of Y	X was dying of Y
X died of Y in his	X died of lung Y	X died of Y
X passed away from Y	X died of Y on	X 's death from Y in
X died of a Y in	X died of lung Y in	X died of lung Y

$\lambda = 1.0$	$\lambda = 0.7$	$\lambda = 0.3$
Y came to power in X in	Y came to power in X	Y came to power in X in
Y came to power in X	Y to power in X	regime of Y in X
Y to power in X	regime of Y in X	X 's dictator Y
Y came to power in X in the	Y came to power in X in	Y became chancellor of X
when Y came to power in X in	Y to power in X in	X 's president Y
when Y came to power in X	Y became chancellor of X	the rise of Y in X
Y took power in X	the rise of Y in X	X 's leader Y
Y rose to power in X	X 's dictator Y	Y , who ruled X
after Y came to power in X	X 's president Y	Y took control of X
Y became chancellor of X	Y took control of X	government of Y in X
Y came to power in X and	Y , who ruled X	X , led by Y
Y seized power in X	Y 's success and X 's saviour	quisling had visited Y in X
Y gained power in X	Y declared that X had	to flee X after Y
to power of Y in X	X 's leader Y	Y in X the year before
Y 's rise to power in X	government of Y in X	X , under the leadership of Y

(c) *was-led-by*

5.1.2 Espresso Diversification

We calculate instance reliability $r_\iota(i)$ and pattern reliability $r_\pi(p)$, following the Espresso algorithm. Espresso is unique in a sense that instances and patterns are scored in a principled, symmetric way.

$$r_\iota(i) = \frac{\sum_{p \in P} \frac{pmi(i,p)}{\max_{pmi}} * r_\pi(p)}{|P|} \tag{5.3}$$

$$r_\pi(p) = \frac{\sum_{i \in I} \frac{pmi(i,p)}{\max_{pmi}} * r_l(i)}{|I|} \tag{5.4}$$

Point-wise Mutual Information (PMI), originally proposed by Church and Hanks (1990), is a statistical measure of association between two random variables. PMI in Espresso is calculated as follows:

$$pmi(i,p) = \log \frac{|x,p,y|}{|x,*,y||*,p,*|} \tag{5.5}$$

where the notation $|x,p,y|$ represents the frequency of p with its slots filled with $i = \langle x,y \rangle$, and the notation '*' represents a wild card. $\max_{pmi}$ is the maximum PMI between all combinations of P and I.

By expanding Eq. (7.2) with Eq. (5.2, 5.4), we can obtain the updated pattern reliability score:

$$\begin{aligned} r'_{\pi}(p_k) &= \lambda \cdot r_{\pi}(p_k) + (1-\lambda) \cdot diversity(p_k) \\ &= \lambda \cdot \frac{\sum_{i \in I} \frac{pmi(i,p_k)}{\max_{pmi}} * r_l(i)}{|I|} \\ &\quad + (1-\lambda) \cdot \frac{D[k]}{2} \cdot r_{\pi}(p_0). \end{aligned} \tag{5.6}$$

5.1.3 Diversification with Different λ Values

We ran the *Diversifiable Bootstrapping* incorporated with the Espresso framework in order to harvest lexically diverse paraphrases.

As a corpus, we used Wikipedia that contains about 2.1 million articles. Since a pattern is found from within a sentence, but not across adjacent sentences, the corpus is preprocessed with a sentence segmenter, where 43 million sentences were annotated in total. The seed instances from Schlaefer et al. (2006) are shown in Table 5.1.

The acquired patterns are shown in Table 5.2. These results are sorted in the descending order with respect to the updated reliability score $r'_{\pi}(p)$. The values were chosen to represent different levels of diversification (where the original bootstrapping results without diversification are obtained when $\lambda = 1$). Notice that patterns became more diverse as a smaller λ value was given. We do not claim these are optimal values, or a smaller λ value is better.

5.2 Discussion

5.2.1 Comparison with MMR

In Query-Focused Text Summarization, given a query and a long text, one has to generate a short text that is relevant to the query and is diverse in topics. Carbonell and Goldstein (1998) proposed the Maximal Marginal Relevance (MMR) approach where relevance and redundancy is measured separately, and linearly combined.

In Information Retrieval, Search Results Diversification problem has been actively studied (Agrawal et al., 2009; Rafiei et al., 2010; Santos et al., 2011; van Leuken et al., 2009). An idea behind this problem is that, when an ambiguous query is given, diversifying topics would improve the chance of satisfying a user's information need. According to Santos et al. (2011), "most of the existing diversification approaches are somehow inspired" by MMR.

The proposed work in this chapter is similar to MMR in a sense that two components are separately measured and linearly interpolated. In the research problems above, the first component of MMR has been calculated with respect to the relevance to a query. However, in our problem, there is no notion of a query. Therefore, instead of using the relevance between query and summary candidate, or between query and search result, we used a reliability score of a pattern.

5.2.2 Diversification and Semantic Drift

Mcintosh (2009) implies that as less precise patterns are extracted in later iterations, lexicon's meaning start to drift. When a low λ parameter is given, our approach allows lexically-diverse but potentially less precise patterns to be selected. Therefore, diversification and semantic drift are in a trade-off relationship. We will investigate the relationship further in Chapter 7.

5.2.3 Limitations of Diversity Calculation

In this work, we used an off-the-shelf D calculation algorithm from Shima and Mitamura (2011), which leaves room for improvement. Since the algorithm is inspired by a graded relevance judgment in Information Retrieval evaluation, similar simple quantification (i.e. giving a score of 0, 1, or 2) is done in D. In other words, there should be a better way of representing diversity into a number. Another weakness might be that a useful paraphrase of "kill" such as "do away with" will be assigned with a score of 0, depending on an implementation of content word extraction algorithm. In addition, we did not discuss how to deal with a paraphrase that cannot be inter-substitutable due to a syntactic discrepancy e.g. "*X* killed *Y*" and "of *X* shooting *Y*". In an Information Extraction task, it would be ok to keep such a paraphrase; however, in a paraphrase generation task, only an inter-substitutable paraphrase would be appropriate to keep in the final list.

5.3 Summary

In this chapter, we proposed a lightly-supervised bootstrap approach called *Diversifiable Bootstrapping* that extends the paraphrase extraction framework presented in Chapter 3. By using the proposed approach, one can expect to extract binary-argument phrase-level paraphrases that are rich in lexical diversity, which is a missing piece in the state-of-the-art paraphrase extraction works. As seen in Table 5.2, some paraphrases extractable by the algorithm are phrasal expressions that are not found in common dictionary as synonyms. We will further present experimental results in the Chapter 7 that features experimental results and analysis.

Chapter 6

Diversity-aware Evaluation Metric for Paraphrase Patterns

In a literature review in Section 2.5.1 (Chapter 2), we identified that there is study in paraphrase evaluation metric that takes into account lexical diversity. In this chapter, we propose a diversity-aware paraphrase evaluation metric called DIMPLE[1], which boosts the scores of lexically diverse paraphrase pairs.

6.1 Introduction

Paraphrase pairs or patterns are useful in various NLP related research domains, since there is a common need to automatically identify meaning equivalence between two or more texts.

Consider a paraphrase pair resource that links "killed" to "assassinated" (in the rest of this thesis we denote such a rule as ⟨"killed"[2], "assassinated"[3]⟩). In automatic evaluation for Machine Translation (MT) (Kauchak and Barzilay, 2006; Padó et al., 2009; Zhou et al., 2006a), this rule may enable a metric to identify phrase-level semantic similarity between a system response containing "killed", and a reference translation containing "killed". Similarly in query expansion for information retrieval (IR) (Riezler et al., 2007), this rule may enable a system to expand the query term "killed" with the paraphrase "assassinated", in order to match a potentially relevant document containing the expanded term.

To evaluate paraphrase patterns during pattern discovery, ideally we should use an evaluation metric that strongly predicts performance on the extrinsic task (e.g. fluency and adequacy scores in MT, mean average precision in IR) where the paraphrase patterns are used.

Many existing approaches use a paraphrase evaluation methodology where human assessors judge each paraphrase pair as to whether they have the same meaning. Over a set of paraphrase rules for one source term, Expected Precision (EP) is calculated by taking the mean of precision, or the ratio of positive labels annotated by assessors (Bannard and Callison-Burch, 2005; Callison-Burch, 2008; Kok and Brockett, 2010; Metzler et al., 2011).

[1] DIversity-aware Metric for Pattern Learning Experiments

[2] Source term/phrase that contains "killed"

[3] Paraphrase that contains "assassinated"

The weakness of this approach is that EP is an intrinsic measure that does not necessarily predict how well a paraphrase-embedded system will perform in practice. For example, a set of paraphrase pairs ⟨"killed", "shot and killed"⟩, ⟨"killed", "reported killed"⟩ ... ⟨"killed", "killed in"⟩ may receive a perfect score of 1.0 in EP; however, these patterns do not provide lexical diversity (e.g. ⟨"killed", "assassinated"⟩) and therefore may not perform well in an application where lexical diversity is important.

The goal of this chapter is to provide empirical evidence to support the assumption that the proposed paraphrase evaluation metric DIMPLE correlates better with paraphrase recognition task metric scores than previous metrics do, by rewarding lexical diverse patterns.

6.2 DIMPLE Metric

Patterns or rules for capturing equivalence in meaning are used in various NLP applications. In a broad sense, the terms *paraphrase* will be used to denote pairs or a set of patterns that represent semantically equivalent or close texts with different surface forms.

Given paraphrase patterns P, or the ranked list of distinct paraphrase pairs sorted by confidence in descending order, DIMPLE$_k$ evaluates the top k patterns, and produces a real number between 0 and 1 (higher the better).

6.2.1 Cumulative Gain

DIMPLE is inspired by the Cumulative Gain (CG) metric (Järvelin and Kekäläinen, 2002; Kekäläinen, 2005) used in IR. CG for the top k retrieved documents is calculated as $\mathrm{CG}_k = \sum_{i=1}^{k} gain_i$ where the gain function is human-judged relevance grade of the i-th document with respect to information need (e.g. 0 through 3 for irrelevant, marginally relevant, fairly relevant and highly relevant respectively). We take an alternative well-known formula for CG calculation, which puts stronger emphasis at higher gain:

$$\mathrm{CG}_k = \sum_{i=1}^{k} \left(2^{gain_i} - 1\right).$$

6.2.2 DIMPLE Algorithm

DIMPLE is a normalized CG calculated on each paraphrase. The gain function of DIMPLE is represented as a product of pattern quality Q and lexical diversity D: DIMPLE at rank k is a normalized CG$_k$ which is defined as:

$$\mathrm{DIMPLE}_k = \frac{\mathrm{CG}_k}{Z} = \frac{\sum_{i=1}^{k} \left(2^{Q_i \cdot D_i} - 1\right)}{Z}$$

where Z is a normalization factor such that the perfect CG score is given. Since Q takes a real value between 0 and 1, and D takes an integer between 1 and 3, $Z = \sum^{k} \left(2^3 - 1\right)$. Being able to design Q and D independently is one of characteristics in DIMPLE. In theory, Q can be any quality measure on paraphrase patterns, such as the instance-based evaluation score (Szpektor et al., 2007), or alignment-based evaluation score (Callison-Burch, 2008). Similarly, D can be implemented depending on the domain task; for example, if we are interested in learning paraphrases that are out-of-vocabulary or domain-specific, D could consult a dictionary, and return a high score if the lexical entry could not be found. The DIMPLE framework is implemented in the following way. Let Q be the ratio of positive labels averaged over pairs by human assessors given pi as to whether a paraphrase has the same meaning as the source term or not. Let D be the degree of lexical diversity of a pattern calculated using Algorithm 3 below.

Algorithm 3 D score calculation

```
Input: paraphrases $w_1, \ldots, w_k$ for a source term $s$
Output: $D$ array indexed by $1, \ldots, k$
  Set $history1 \leftarrow$ extractContentWords($s$)
  Set $history2 \leftarrow$ stemWords($history1$)
  for $i = 1 \rightarrow k$ do
    Set $W1 \leftarrow$ extractContentWords($w_i$)
    Set $W2 \leftarrow$ stemWords($W1$)
    if $W1 = \emptyset$ OR $W1 \cup history1 \neq \emptyset$ then
      $D[i] \leftarrow 1$ // word already seen
    else
      if $W2 \cap history2 \neq \emptyset$ then
        $D[i] \leftarrow 2$ // word's root already seem
      else
        $D[i] \leftarrow 3$ // unseen word
      end if
      $history1 \leftarrow W1 \cup history1$
      $history2 \leftarrow W2 \cup history2$
    end if
  end for
```

6.3 Experiment

We use the Pearson product-moment correlation coefficient to measure correlation between two vectors consisting of intrinsic and extrinsic scores on paraphrase patterns, following previous meta-evaluation research (Callison-Burch et al., 2007, 2008; Przybocki et al., 2009; Tratz and Hovy, 2009). By intrinsic score, we mean a theory-based direct assessment result on the paraphrase patterns. By extrinsic score, we mean to measure how much the paraphrase recognition component helps the entire system to achieve a task. The correlation score is 1 if there is a perfect positive correlation, 0 if there is no correlation and -1 if there is a perfect negative correlation.

Using a task performance score to evaluate a paraphrase generation algorithm has been studied previously (Bhagat and Ravichandran, 2008; Szpektor and Dagan, 2007, 2008). A common issue in extrinsic evaluations is that it is hard to separate out errors, or contributions from other possibly complex modules. Our work presents an approach which can predict task performance in more simple experimental settings.

6.3.1 Annotated Paraphrase Resource

We used the paraphrase pattern dataset "paraphrase-eval" (Metzler and Hovy, 2011; Metzler et al., 2011) which contains paraphrase patterns acquired by multiple algorithms:

- **PD** Based on the left and right n-gram contexts of the source term with scoring based on overlap (Paşca and Dienes, 2005)
- **BR** Based on Noun Phrase chunks as contexts
- **BCB** Based on monolingual phrase alignment from a bilingual corpus using a pivot (Bannard and Callison-Burch, 2005)
- **BCB-S** Same as BCB except that syntactic type is constrained (Callison-Burch, 2008)

In the dataset, each paraphrase pair is assigned with an annotation as to whether a pair is a correct paraphrase or not by 2 or 3 human annotators.

The source terms are 100 verbs extracted from newswire about terrorism and American football. We selected 10 verbs according to their frequency in extrinsic task datasets (details follow in Section 6.3.3).

Following the methodology used in previous paraphrase evaluations (Bannard and Callison-Burch, 2005; Callison-Burch, 2008; Kok and Brockett, 2010), the labels were annotated on a pair of two sentences: an original sentence containing the source term, and the same sentence with the source term replaced with the paraphrase pattern, so that contextual information could help annotators to make consistent judgments. The judgment is based on whether the "same meaning" is present between the source term and its paraphrase. There is a lenient and a strict distinction on the "same meaning" judgments. The strict label is given when the replaced sentence is grammatically correct whereas the lenient label is given even when the sentence is grammatically incorrect.

In total, we have 10 (source terms listed in Table 6.1) 4 (paraphrase generation algorithms introduced above) = 40 sets of paraphrase patterns. In each set of paraphrase patterns, there are up to 10 unique ⟨source term, paraphrase⟩ pairs.

6.3.2 Intrinsic Paraphrase Metrics

We will discuss the common metric EP, and its variant EPR as baselines to be compared with DIMPLE. For each metric, we used a cutoff value of k=1, 5 and 10. EP: Our baseline is the Expected Precision at k, which is the expected number of correct paraphrases among the top k returned, and is computed as: where Q is the ratio of positive labels. For instance, if 2 out of 3 human annotators judged that pi = ⟨"killed", "fatally shot"⟩ has the same meaning, $Q_i = 2/3$. EPR (Metzler et al., 2011): extended EP with a Redundancy judgment, which we shall call EPR where lexically redundant paraphrases did not receive a credit. Unlike (Metzler et al., 2011) where humans judged redundancies, we do the judgment automatically with a Porter Stemmer (Porter, 1980) to extract and compare stemmed forms. In that way EPR's output become comparable to DIMPLE's, remaining redundancy scoring different (i.e. binary filtering in EPR and 3-level weighting in DIMPLE).

6.3.3 Extrinsic Evaluation Datasets

Ideally, paraphrase metric scores should correlate well with task performance metrics. To insulate the experiment from external, uncontrollable factors (e.g. errors from other task components), we created three datasets with slightly different characteristics, where the essential task of recognizing meaning equivalence between different surface texts can be conducted. The numbers of positive-labeled pairs that we extracted for the three corpus, MSRPC, RTE and CQAE are 3900, 2805 and 27397 respectively. Table 6.1 shows the number of text pairs selected in which at least one of each pair contains a frequently occurring verb.

Table 6.1: 10 most frequently occurring source verbs in three datasets. Numbers are positive-labeled pairs where the verb appears in at least one side of a pair.

Src verb	MSRPC	RTE	CQAE
found	89	62	319
called	59	61	379
told	125	34	189
killed	48	109	277
accused	30	44	143
to take	21	23	63
reached	22	18	107
returned	14	20	57
turned	22	10	94
broke	10	10	35

MSRPC The Microsoft Research Paraphrase Corpus (Dolan et al., 2004; Dolan and Brockett, 2005; Quirk et al., 2004) contains 5800 pairs of sentences along with human annotations where positive labels mean semantic equivalence of pairs.

RTE (Quasi-)paraphrase patterns are useful for the closely related task, Recognizing Textual Entailment. This dataset has been taken from the 2-way/3-way track at PASCAL/TAC RTE1-4. Positive examples are premise-hypothesis pairs where human annotators assigned the entailment label. The original dataset has been generated from actual applications such as Text Summarization, Information Extraction, IR, Question Answering.

CQAE Complex Question Answering Evaluation (CQAE) dataset has been built from 6 past TREC QA tracks, i.e. "Other" QA data from TREC 2005 through 2007, relation QA data from TREC 2005 and ciQA from TREC 2006 and 2007 (Dang et al., 2006, 2007; Voorhees and Dang, 2005). We created unique pairs consisting of a system response (often sentence-length) and an answer nugget as positive examples, where the system response is judged by human as containing or expressing the meaning of the nugget.

6.3.4 Extrinsic Performance Metric

Using the dataset described in Section 6.3.3, performance measures for each of the 40 paraphrase sets (10 verbs times 4 generators) are calculated as the ratio of pairs correctly identified as paraphrases.

In order to make the experimental settings close to an actual system with an embedded paraphrase engine, we first apply simple unigram matching with stemming enabled. At this stage, a text with the source verb "killed" and another text with the inflectional variant "killing" would match. As an alternative approach, we consult the paraphrase pattern set trying to find a match between the texts. This identification judgment is automated, where we assume a meaning equivalence is identified between texts when the source verb matches one text and one of up to 10 paraphrases in the set matches the other. Given these evaluation settings, a noisy paraphrase pair such as ⟨"killed", "to"⟩ can easily match many pairs and falsely boost the performance score. We filter such exceptional cases when the paraphrase text contains only functional words.

6.3.5 Results

We conducted experiments to provide evidence that the Pearson correlation coefficient of DIMPLE is higher than that of the other two baselines. Table 6.2 and 3 below present the result where each number is the correlation calculated on the 40 data points.

Table 6.2: Correlation between intrinsic paraphrase metrics and extrinsic paraphrase recognition task metrics where DIMPLE's Q score is based on *lenient* judgment. Bold figures indicate statistical significance of the correlation statistics (null-hypothesis tested: "there is no correlation", p-value<0.01).

	EP_k			EPR_k			DIMPLE_k		
	k=1	5	10	1	5	10	1	5	10
MSRPC	-0.02	-0.24	-0.11	0.33	0.27	-0.12	0.32	0.20	0.25
RTE	0.13	-0.05	0.11	0.33	0.12	0.09	**0.46**	0.25	0.37
CQAE	0.08	-0.09	0.00	-0.02	-0.08	-0.13	0.35	0.25	**0.40**

Table 6.3: Same as the Table 6.2, except that the Q score is based on *strict* judgment.

	EP_k			EPR_k			DIMPLE_k		
	k=1	5	10	1	5	10	1	5	10
MSRPC	0.12	0.13	0.19	0.26	0.36	0.37	0.26	0.35	**0.52**
RTE	0.34	0.34	0.29	**0.43**	**0.41**	**0.38**	**0.49**	**0.55**	**0.58**
CQAE	**0.44**	**0.51**	**0.47**	0.37	**0.60**	**0.55**	0.37	**0.70**	**0.70**

Table 6.2 shows that correlations are almost always close to 0, indicating that EP does not correlate with the extrinsic measures when the Q score is calculated in lenient judgment mode.

On the other hand, when the Q function is based on strict judgments, EP scores sometimes show a medium positive correlation with the extrinsic task performance, such as on the CQAE dataset.

In both tables, there is a general trend where the correlation scores fall in the same relative order (given the same cut-off value): EP $<$ EPR $<$ DIMPLE. This suggests that DIMPLE has a higher correlation than the other two baselines, given the task performance measure we experimented with. As we can see from Table 6.2, DIMPLE correlates well with paraphrase task performance, especially when the cutoff value k is 5 or 10. The higher values in Table 6.3 (compared to Table 6.2) show that the strict judgment used for intrinsic metric calculation is preferable over the lenient one.

6.4 Summary

We proposed a novel paraphrase evaluation metric called DIMPLE, which gives weight to lexical variety. We built large scale datasets from three sources and conducted extrinsic evaluations where paraphrase recognition is involved. Experimental results showed that Pearson correlation statistics for DIMPLE are approximately 0.5 to 0.7 (when k=10 and “strict” annotations are used to calculate the score), which is higher than scores for the commonly used EP and EPR metrics.

Chapter 7

Evaluation

This chapter will present experimental results for the paraphrase extraction methods described in detail from Chapter 3 through 5. As one of the evaluation metrics, we will use *DIMPLE* proposed in Chapter 6. Also see Appendix A for the actual sample of paraphrases extracted.

7.1 Evaluation Metrics

This section describes the four metrics used in the experiment. Note that patterns evaluated at each iteration are all the patterns extracted, not just the ones "accepted" for the next iteration.

Precision

Precision is the ratio of correct system response. Specifically, precision is defined as the number of correct paraphrases extracted divided by the number of all paraphrases extracted. For the paraphrases extracted from the i-th configuration, precision P_i is calculated as follows:

$$P_i = \frac{|Correct_i \cap Extracted_i|}{|Extracted_i|}.$$

Recall

Traditionally, recall is calculated as the number of correct system output divided by the total number of all the correct items. In paraphrase evaluation, it is impossible to calculate the denominator because the exhaustive list of correct paraphrases in the universe is unknown. Therefore, we put all the paraphrases from various different system configurations into a pool, and take the size of correct ones for the denominator of recall.

$$R_i = \frac{|Correct_i \cap Extracted_i|}{|\bigcup_j Correct_j|}.$$

This variation of recall is similar to the relative recall (Pantel et al., 2004), but the difference is that the calculation is relative to the pooled correct outputs, rather than a single baseline's output. The advantages of this metric over the relative recall against a single baseline is (1) the calculation does not have to rely on a single number which can easily fluctuate by slight change of parameter (e.g. convergence parameter); (2) the score can be calculated even when the baseline output is 0, which happens to our dataset; (3) the score takes a value between 0 and 1, which makes it easy to see the upper-bound.

DIMPLE

In Chapter 6, we have defined the DIMPLE (DIversity-aware Metric for Pattern Learning Experiments) metric. We describe a brief definition of dimple here again.

DIMPLE at rank k is a normalized CG_k which is defined as:

$$\mathrm{DIMPLE}_k = \frac{\mathrm{CG}_k}{Z} = \frac{\sum_{i=1}^{k}\left(2^{Q_i \cdot D_i} - 1\right)}{Z}$$

where Z is a normalization factor such that the perfect CG score is given. Since Q takes a real value between 0 and 1, and D takes an integer between 1 and 3, $Z = \sum^k \left(2^3 - 1\right)$. Being able to design Q and D independently is one of characteristics in DIMPLE.

Number of distinct keywords

In order to analyze the data further, we will also investigate the number of distinct keywords in a paraphrase set. It is similar to dimple in a sense that lexical diversity is measured, but is more intuitive to interpret.

For example, given $\langle X$ was assassinated by Y, Y is the assassin of X, Y killed $X\rangle$, an annotator extracts a keyword from the pattern with inflection and derivational morphology normalized: “assassinate” and “kill”. In this case, the number of distinct keywords is two.

7.2 Experiment Settings

7.2.1 Relations and Original Seed

We have completed annotation for 16 relations shown in Table 7.1. Each relation is a binary-relation, for example, `writer_was_born_in_city` is a relation between the city and a writer who was born in the city. The source column shows the origin of the seed which are publicly available. “N” indicates the NELL project Carlson et al. (2010a)[1], and “E” indicates the Ephyra QA dataset (Schlaefer et al., 2006)[2].

The criteria for choosing the relations are based on the balance of different properties. For example, arguments ideally range over multiple different types among the set of relations. In our relations, some types can be named entity: person, location, organization, date. On the other hand, some relation takes general nouns as an argument such as agricultural product

[1] http://rtw.ml.cmu.edu/rtw/

[2] http://www.ephyra.info/

Table 7.1: List of seed relations with argument types and seed source

Source	Relation ID	Arg1	Arg2
N	agricultural_product_came_from_country	product (agricultural)	org (country)
N	bank_bought_bank	org (bank)	org (bank)
N	book_writer	person (author)	book
N	competes_with	org (company)	org (company)
N	has_sister	person	person (female)
N	has_wife	person (male)	person (female)
N	person_birth_date	person	date
N	person_death_date	person	date
N	person_graduated_school	person	org (school)
N	person_leads_organization	person	org
N	person_moved_to_state_or_province	person	loc (state)
N	writer_was_born_in_city	person (author)	loc (city)
E	CAUSE_OF_DEATH	person	disease, accident
E	DATE_OF_START	event	date
E	KILLER	person	person
E	LEADER	org (company, country)	person

(agricultural_product_came_from_country), as well as a mixture of multiple general noun categories e.g. disease and accident (CAUSE_OF_DEATH). Some relations such as person_death_date, CAUSE_OF_DEATH, KILLER are expected to have diverse euphemistic paraphrases because of its tabooness about death.

7.2.2 Source Corpus

As the source corpora, we use English version of Wikipedia. Table 7.2 summarizes the corpus statistics.

Table 7.2: Candidate Source corpora for experiments.

Name	Type	Documents	Sentences	Total Terms
Wikipedia	Encyclopedia	2,114,541	50,118,286	1,002,377,340

Wikipedia[3] is a web-based encyclopedia which is collaboratively edited by millions of editors across the world.

7.2.3 Paraphrase Extractor Configurations

Below is the description of paraphrase extraction systems used in the experiment.

[3] http://dumps.wikimedia.org/enwiki

- **BPL:** This configuration, Bootstrap Paraphrase Learner (BPL), extends the vanilla version, which detail is described at Section 3.3 in Chapter 3. This configuration also implements the semantic-drift prevention mechanism by distributional type-scoring, described in Chapter 4.
- **D-BPL:** Diversifiable Bootstrap Paraphrase Learner (D-BPL) extends BPL where pattern scoring takes lexical diversity into account, which can be controlled by a parameter λ. We set the $\lambda = 0.75$ based on a parameter-sweep on a held-out dataset. The details of this algorithm is described in Chapter 5.
- **VANILLA:** This is basically a simple implementation of ESPRESSO (Pantel and Pennacchiotti, 2006) which is described at Section 3.2.1 in Chapter 3.
- **CPL:** For the relations which seeds are taken from the Never Ending Language Learning (NELL) dataset, we additionally compare the results with patterns learned by NELL's *Coupled Pattern Learner* (CPL) (Carlson et al., 2010a,b), which are binary relation patterns extracted from the same seed[4].

The iteration convergence criteria is $\tau_2 = 0.01$, which means the iteration stops when the average pattern score (reliability) becomes lower than 1% of the previous iteration's. In the experiment, we continue to run iterations even after the convergence criteria is met, because we would like to observe the long-term behavior of the bootstrapping.

In the i-th iteration, we accept top i patterns (ranked in reliability) to be used for instance extraction. The patterns at each iteration are selected for evaluation if the reliability of the pattern is more than 1% of the reliability of the highest ranked pattern.

7.2.4 Gold Standard Annotation

For each relation, a pool of paraphrases are created by running paraphrase extractors in different configuration. After duplications are removed, each paraphrase pattern is given with exactly one of the labels described in Table 7.3. In our evaluation, patterns with "M", "O" and "I" labels are treated as correct patterns (2-way labels). More detailed guideline is available in Appendix B.

Table 7.4 shows the resulting label distribution over patterns for each relation.

Inter-Annotator Agreement (Cohen's Kappa)

In order to analyze the consistency of gold standard labels, we calculated Cohen's Kappa coefficient (Viera and Garrett, 2005). Cohen's Kappa is a measure that represents agreement of two annotators, which is calculated as follows:

$$\kappa = \frac{\Pr(o) - \Pr(e)}{1 - \Pr(e)}$$

where $\Pr(o)$ is an observed agreement and $\Pr(e)$ is the expected agreement.

[4] CPL data has been obtained from the NELL's 860th iteration: `http://rtw.ml.cmu.edu/resources/results/08m/NELL.08m.860.extractionPatterns.csv.gz`

Table 7.3: Gold standard label description.

LABEL	DESCRIPTION
M	**M**atched: If the X and Y in the pattern is instantiated with concrete values (as seen in the Seed section in the judge UI), it is likely to match this criteria (**high certainty**): the template represents the intended meaning. (From a researcher's point of view, the patterns with this label are a set of "paraphrase templates").
O	Matched and **O**OD (Matched and Out-of-dictionary): A pattern is matched, AND its keyword is not a synonym according to WordNet. It could be a colloquial, metaphorical, idiomatic, or euphemistic expression. (From a language resource acquisition researcher's point of view, this kind of pattern is very valuable as it's worth finding automatically from a corpus.)
I	**I**nconclusive: It may or may not "match" depending on the context of the pattern in sentences. (**medium certainty**)
R	**R**elated: Even if instantiated with the correct slot values, the pattern does not represent the intended meaning (**no or very small certainty**). However, the pattern represents a related fact/event that may occur between X and Y.
A	**A**ntonym: It has the opposite meaning as "M"-label patterns.
W	**W**rong: None of the above.

Table 7.4: Distribution of gold-standard labels annotated on the extracted paraphrases.

Relation ID	M	O	I	R	A	W	Total
agricultural_product_came_from_country	328	4	374	107	18	5106	5937
bank_bought_bank	426	6	541	6	114	2279	3372
book_writer	1672	33	1193	101	9	3356	6364
competes_with	118	43	232	124	29	2670	3216
has_sister	120	1	220	58	1	3888	4288
has_wife	369	74	45	105	6	4735	5334
person_birth_date	223	0	18	14	34	4019	4308
person_death_date	374	4	64	1	123	3907	4473
person_graduated_school	219	17	289	199	4	1199	1927
person_leads_organization	752	161	325	13	0	957	2208
person_moved_to_state_or_province	71	4	144	26	6	3038	3289
writer_was_born_in_city	491	3	288	205	24	2495	3506
CAUSE_OF_DEATH	643	30	24	196	11	1761	2665
DATE_OF_START	212	126	218	35	172	3696	4459
KILLER	477	13	73	15	2	1093	1673
LEADER	690	139	153	0	1	3552	4535

The inter-annotator agreement for the five relations between two annotators are reported in Table 7.2.4. As mentioned earlier in this section, we treated "M", "O" and "I" labels as correct, and "W", "R", "A" as incorrect in the evaluation (2-way labeling). Notice that "moderate agreement"(Viera and Garrett, 2005), which ranges between 0.4 and 0.6, is observed in 2-way labeling except for `book_writer` (0.382). The table additionally contains Kappa coefficients for 6-way labeling.

Table 7.5: Inter-Annotator Agreement.

Relation ID	# of Patterns	2-way	(6-way)
CAUSE_OF_DEATH	2039	0.590	(0.456)
bank_bought_bank	1317	0.533	(0.390)
book_writer	4400	0.382	(0.275)
person_graduated_school	1170	0.566	(0.482)
writer_was_born_in_city	1624	0.489	(0.394)

7.3 Results: Effect of Diversification

We report experimental results that verify if diversifiable bootstrapping paraphrase learner (*D-BPL*) can successfully extract lexically diverse paraphrases than the baseline systems *BPL* and *VANILLA*.

In the experiment, a set of paraphrases are extracted for each relation by the paraphrase extraction algorithms. Figure 7.1 shows a comparison of paraphrases by each extraction algorithms as measured by different metric scores (y-axis; macro-averaged over relations) per iteration (x-axis). One can observe that (1) BPL and D-BPL are better than the baseline VANILLA (and also CPL); and that (2) precision and recall trades off between BPL and D-BPL.

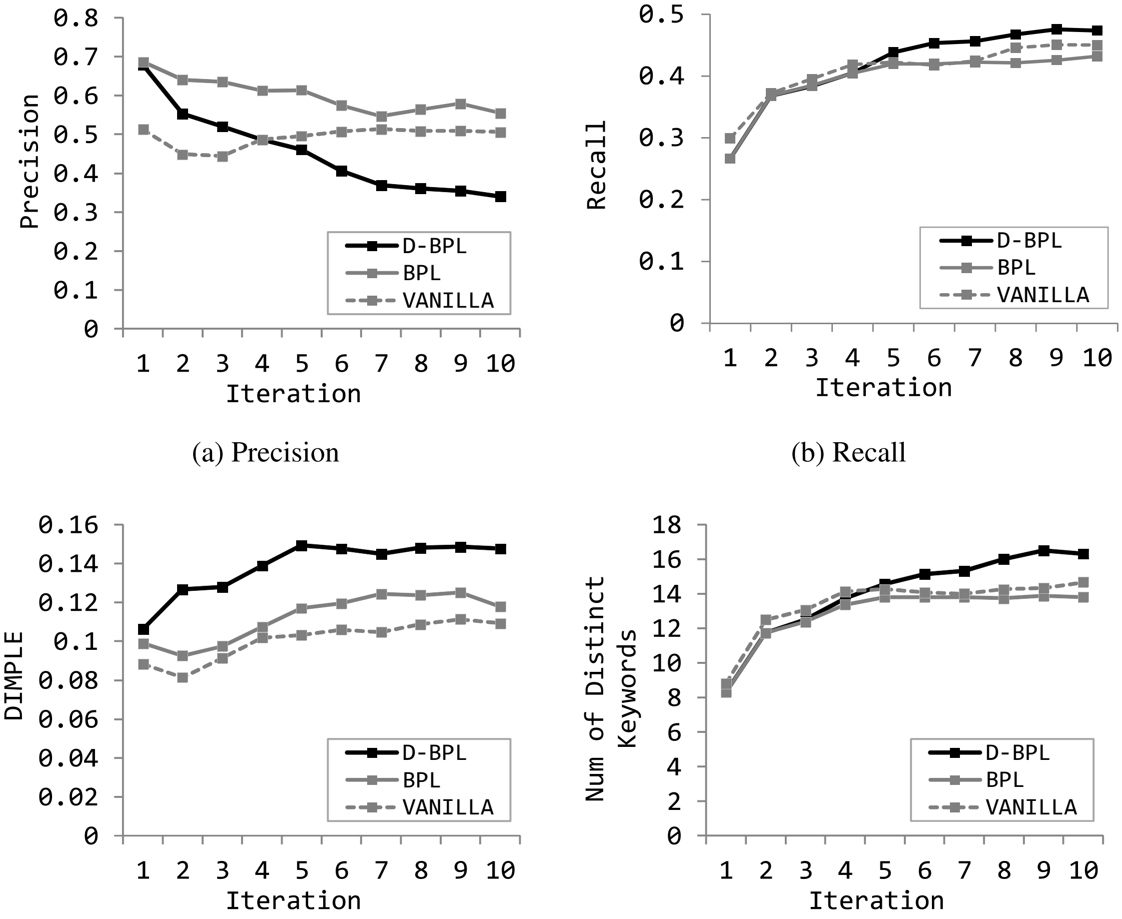
Table 7.6: Average iterations at convergence

	VANILLA	BPL	D-BPL
Avg Iteration	8.875	7.813	7.750

In the experiment in Figure 7.1, iterations were forced to continue up to 10-th, even though the convergence criteria are met. This is because a shared parameter ($\tau_2 = 1\%$)[5] for all the algorithm settings and relations might not be optimal. In Table 7.6, we show empirical average of iterations convergence happened.

On the other hand, Table 7.7 through 7.10 shows the metric scores at convergence. CPL scores are also reported for the relations which seeds are shared with the NELL project.

[5] Converge when the average pattern score decreases by τ_2 or more as compared to the previous iteration.



(a) Precision

(b) Recall

(c) DIMPLE

(d) Number of distinct keywords

Figure 7.1: Paraphrases extracted by diversifiable bootstrap paraphrase learner (D-BPL) is compared against the baselines, using different metric scores (y-axis; macro-averaged over relations) per iteration (x-axis).

Table 7.7: D-BPL result: Precision

Relation ID	CPL	VANILLA	BPL	D-BPL
agricultural_product_came_from_country	0.193	0.298	0.350	0.323
bank_bought_bank	0.426	0.814	0.840	0.689
book_writer	0.338	0.609	0.786	0.569
competes_with	0.031	0.097	0.729	0.332
has_sister	0.386	0.029	0.375	0.345
has_wife	0.296	1.000	0.444	0.444
person_birth_date	0.833	0.667	0.528	0.444
person_death_date	0.600	0.230	0.688	0.382
person_graduated_school	0.036	0.565	0.482	0.472
person_leads_organization	0.552	0.906	0.988	0.667
person_moved_to_state_or_province	0.071	0.000	0.000	0.108
writer_was_born_in_city	0.009	0.778	0.571	0.338
CAUSE_OF_DEATH	n/a	0.000	0.606	0.422
DATE_OF_START	n/a	0.332	0.330	0.179
KILLER	n/a	0.644	0.909	0.850
LEADER	n/a	0.726	0.855	0.470
Macro-average	0.314	0.481	0.593	0.440

Table 7.8: D-BPL result: Recall

Relation ID	CPL	VANILLA	BPL	D-BPL
agricultural_product_came_from_country	0.509	0.291	0.236	0.291
bank_bought_bank	0.409	0.636	0.455	0.545
book_writer	0.537	0.432	0.411	0.589
competes_with	0.457	0.314	0.314	0.429
has_sister	0.200	0.500	0.500	0.500
has_wife	0.615	0.077	0.077	0.077
person_birth_date	0.333	0.667	0.667	0.667
person_death_date	0.059	0.412	0.294	0.294
person_graduated_school	0.533	0.733	0.700	0.700
person_leads_organization	0.573	0.416	0.427	0.416
person_moved_to_state_or_province	0.360	0.280	0.320	0.280
writer_was_born_in_city	0.091	0.182	0.273	0.394
CAUSE_OF_DEATH	n/a	0.143	0.429	0.429
DATE_OF_START	n/a	0.532	0.532	0.574
KILLER	n/a	0.667	0.667	0.667
LEADER	n/a	0.357	0.321	0.304
Macro-average	0.390	0.415	0.414	0.447

Table 7.9 and 7.10 shows that D-BPL achieves in extracting paraphrases with better lexical diversity when compared in macro-averaged dimple and the number of distinct keywords. A statistically significant difference in DIMPLE was observed between D-BPL and VANILLA baseline ($p = 0.023 < 0.05$), and D-BPL and BPL ($p = 0.042 < 0.05$). This suggests that the proposed diversification method is effective in acquiring paraphrases with lexical diversity.

Table 7.9: D-BPL result: Dimple

Relation ID	CPL	VANILLA	BPL	D-BPL
agricultural_product_came_from_country	0.139	0.106	0.131	0.177
bank_bought_bank	0.127	0.217	0.110	0.164
book_writer	0.193	0.253	0.330	0.419
competes_with	0.010	0.069	0.110	0.137
has_sister	0.041	0.011	0.069	0.057
has_wife	0.081	0.010	0.023	0.014
person_birth_date	0.016	0.031	0.044	0.051
person_death_date	0.013	0.084	0.059	0.061
person_graduated_school	0.034	0.171	0.190	0.174
person_leads_organization	0.229	0.187	0.187	0.386
person_moved_to_state_or_province	0.059	0.000	0.000	0.079
writer_was_born_in_city	0.000	0.037	0.057	0.149
CAUSE_OF_DEATH	n/a	0.000	0.057	0.059
DATE_OF_START	n/a	0.156	0.166	0.174
KILLER	n/a	0.126	0.094	0.106
LEADER	n/a	0.150	0.164	0.187
Macro-average	0.078	0.101	0.112	0.150

Table 7.10: D-BPL result: number of distinct keyword

Relation ID	CPL	VANILLA	BPL	D-BPL
agricultural_product_came_from_country	28	16	13	16
bank_bought_bank	9	14	10	12
book_writer	51	41	39	56
competes_with	16	11	11	15
has_sister	2	5	5	5
has_wife	16	2	2	2
person_birth_date	1	2	2	2
person_death_date	1	7	5	5
person_graduated_school	16	22	21	21
person_leads_organization	51	37	38	37
person_moved_to_state_or_province	9	7	8	7
writer_was_born_in_city	3	6	9	13
CAUSE_OF_DEATH	n/a	1	3	3
DATE_OF_START	n/a	25	25	27
KILLER	n/a	4	4	4
LEADER	n/a	20	18	17
Macro-average	16.92	13.75	13.31	15.13

7.4 Additional Experiments

7.4.1 Effect of Type-based Instance Filtering

The diversifiable bootstrapping (introduced in Chapter 5) gives higher score to a pattern that contains an "unseen" content word. While it is beneficial in terms of lexical diversity, it increases a chance of promoting an erroneous pattern. In order to lower that risk, ambiguous instances (usually, general nouns) such that support erroneous patterns should be filtered.

Figure 7.2 presents experimental results of *D-BPL*, with (default; D-BPL(+)) or without (D-BPL(−)) type-based instance filtering (introduced in Chapter 4). The chart (a) shows that precision steeply drops after the first iteration in D-BPL(−), which indicates semantic drift. In contrast, the drop is not steep in D-BPL(+) which uses type-based filtering. The chart (d) shows number of distinct keywords found from extracted patterns is larger from D-BPL(+) than from D-BPL(−), which indicates type-based filtering helps to defer the semantic draft which results in finding new content words in later iterations.

7.4.2 Prediction of Semantic Drift

In Section 3.1.2, we discussed that semantic drift is a critical problem in bootstrapping, and an ideal indicator of semantic drift would follow a similar curve as precision that could be use to detect semantic drift. Then in Section 3.2.1, we hypothesized that one of the most important convergence criteria, the Reliability Ratio (RR) computed in run-time, can be a predictive measure of semantic drift. RR is calculated in the following way on which a threshold value τ_2 is applied in order to determine convergence:

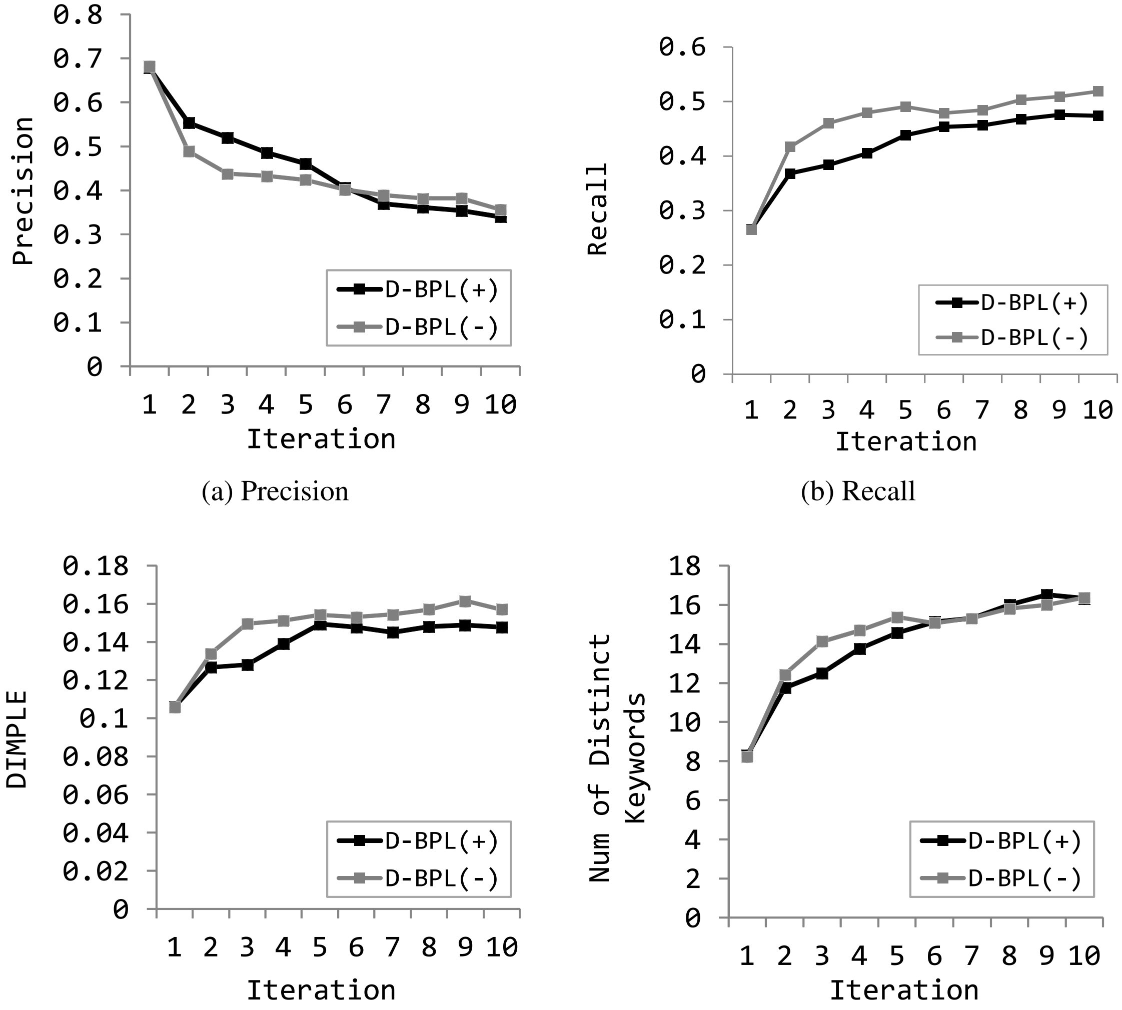
$$\frac{\frac{1}{|P_{k+1}|}\sum_{p\in P_{k+1}} r_\pi(p)}{\frac{1}{|P_k|}\sum_{p\in P_k} r_\pi(p)} < \tau_2. \tag{7.1}$$

We show comparison of Reliability Ratio (RR) and precision in Figure 7.3 where x-axis and y-axis represents iterations and compared values respectively. RR(P) and RR(I) are RR for prototypes and instances respectively. As can be seen from the figure, curves align generally well, considering RR is automatically calculated during run-time based on co-occurrence statistics between patterns and instances whereas Precision is calculated based on human-annotated labels.

By the way, an increase of RR(I) value between the first and second iteration may look counter-intuitive. This indicates that seed instances did not co-occur frequently with a pattern accepted in the first iteration. However instances actually have high co-occurrence once harvesting additional items in later iterations, which provides evidence that bootstrapping is actually working as expected.

7.4.3 Effect of varying level of diversification

As proposed in Section 5.1, the diversification can be realized by interpolating the underlying score with diversity score using the parameter λ where lower λ introduces more lexical diversity in harvested paraphrases $P \ni p$:



(a) Precision

(b) Recall

(c) DIMPLE

(d) Number of distinct keywords

Figure 7.2: Effect of diversification (D-BPL) is compared to BPL, using different metric scores (y-axis; macro-averaged over relations) per iteration (x-axis).

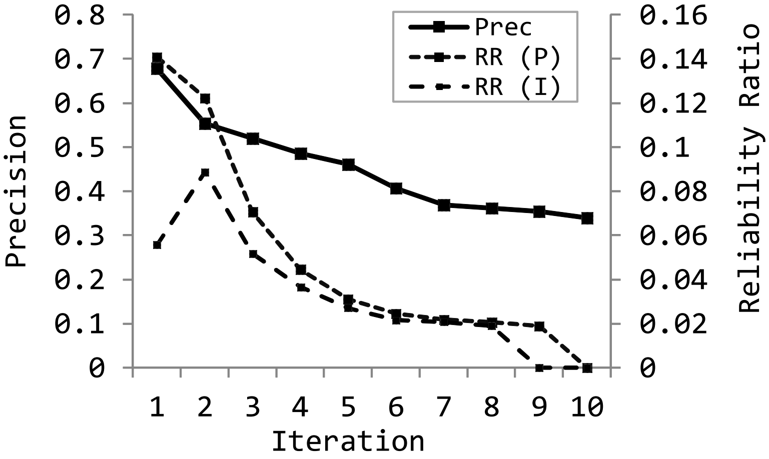


Figure 7.3: Comparison of Reliability Ratio and Precision.

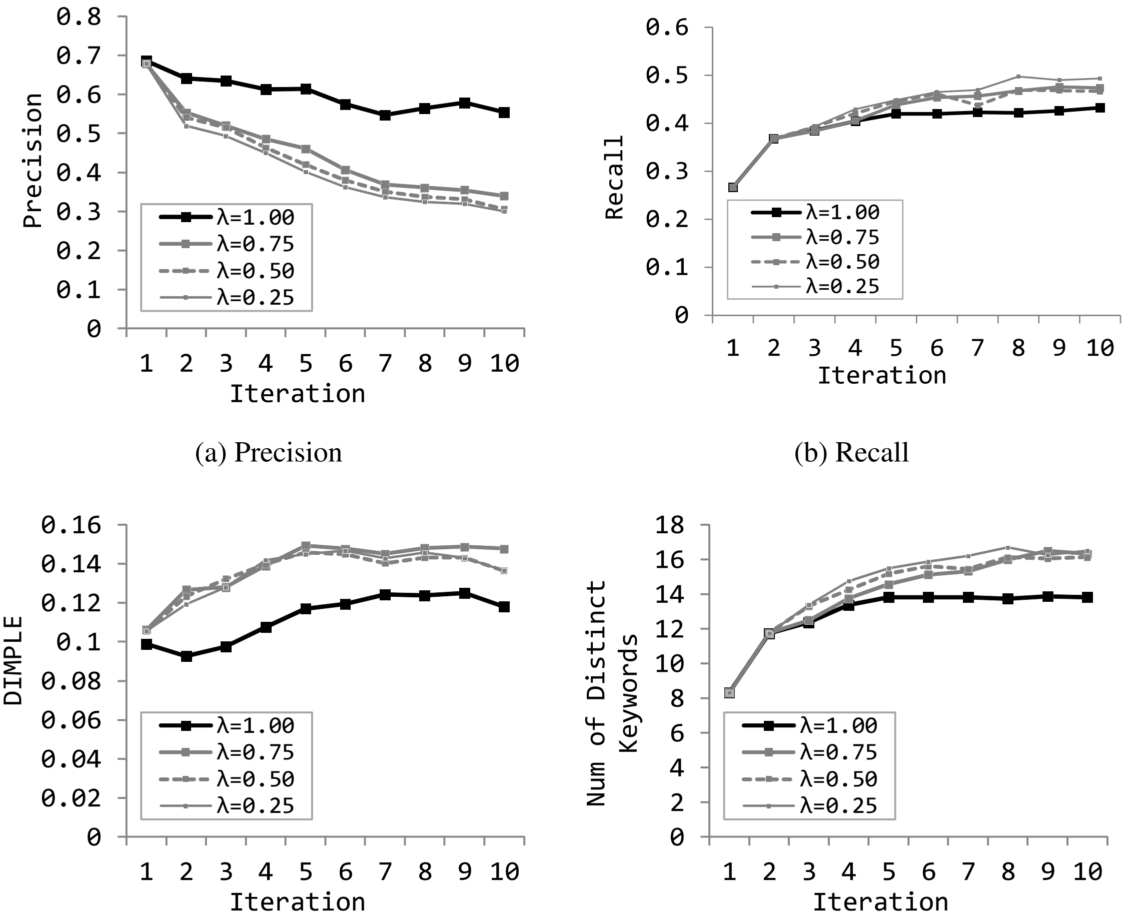
$$r'_{\pi}(p) = \lambda \cdot r_{\pi}(p) + (1 - \lambda) \cdot diversity(p) \tag{7.2}$$

Theoretically, by adjusting the parameter λ, patterns to acquire can be diversified. We show the empirical results in Figure 7.4 that presents how metric scores are affected with varying value of λ (0, 0.5, 0.75 and 1). As we can see the Precision graph in Figure 7.4 (a), the degree of diversification seems to negatively correlate with precision, and this is consistently true from the beginning to the end of iterations. This result suggests that precision, which is a true indicator of semantic drift as discussed in Section sec:sd-formalization, is in a trade-off relationship with lexical diversity of paraphrases. By the way, Figure 7.4 (c) and (d) shows lower λ results in acquiring more lexically diverse paraphrases over the course of iterations. This adds empirical evidence that the diversification algorithm proposed in this work is effective in harvesting diverse set of paraphrases.

7.4.4 Effect of number per iteration to accept

Let us analyze the relationship between diversity and semantic drift from another perspective. Previous works has been attempting multiple different methods to prevent semantic drift as seen in Section 2.3.2. One of such methods is to be conservative about pattern selection trying to select only a set of high precision patterns, for example, by accepting only one more additional pattern per iteration. We loosened this semantic drift prevention criteria in order to see its effect on lexical diversity of resulting patterns. To be more specific, we have different BPL$_n$ configurations changed $n = \{1, 2, 4, 8, 16\}$ where n is the number of additional patterns accepted per iteration. In the i-th iteration, up to $n * i$ patterns are accepted and used to extract instances in the subsequent step.

We show the effect of varying n on number of distinct keywords in Figure 7.5. Note that BPL 1 (n=1) slowly and monotonically increases lexical diversity, while BPL 16 is much faster at first but looses diversity after the third iteration. It is because as n increases, there will be more risk



(a) Precision

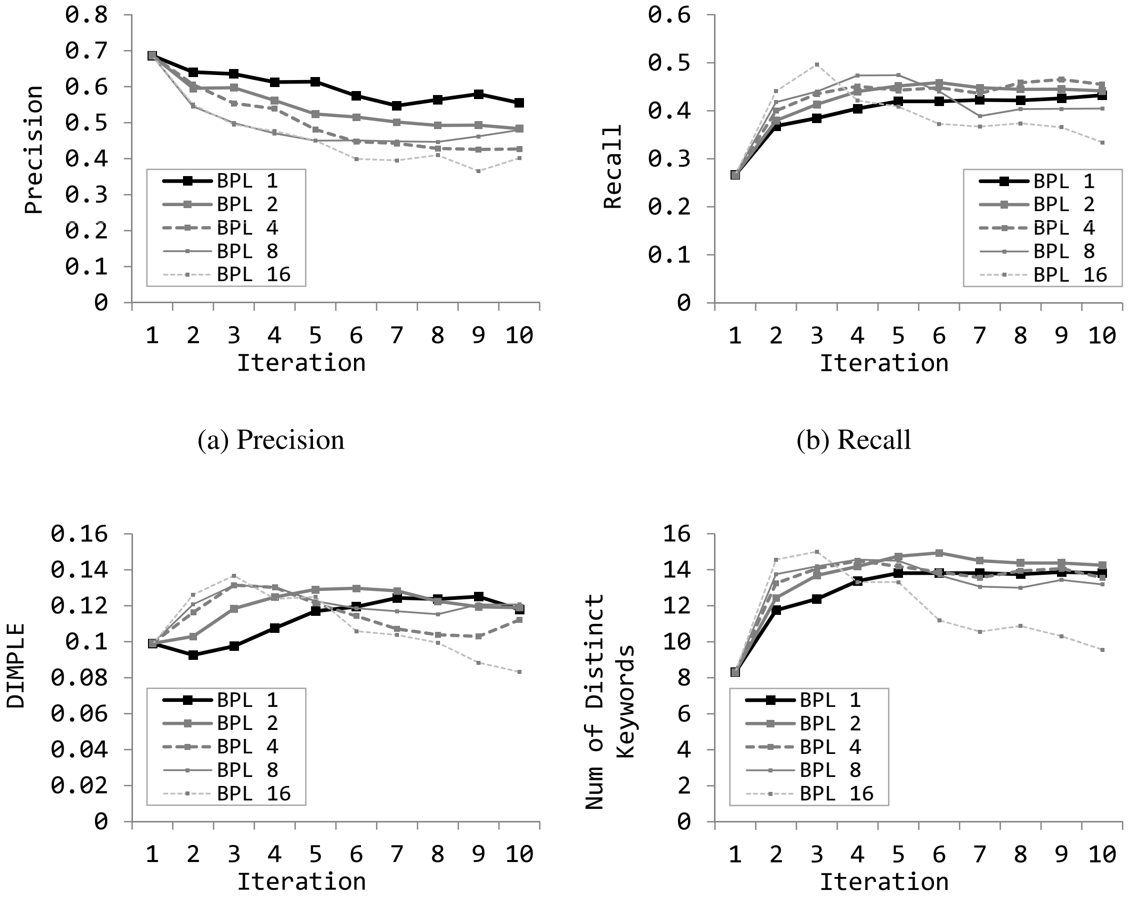
(b) Recall

(c) DIMPLE

(d) Number of distinct keywords

Figure 7.4: Effect of varying lambda on Precision, Recall, DIMPLE and number of distinct keywords.

in accepting noisy (e.g. ambiguous or erroneous) patterns that cause semantic drift.



(a) Precision

(b) Recall

(c) DIMPLE

(d) Number of distinct keywords

Figure 7.5: Effect of varying number of additional patterns to accept per iteration.

Chapter 8

Conclusions

In this thesis, we have first identified problems in the state-of-the-art of paraphrase extraction, and proposed solutions to each of them. See Table 8.1 for the summary of contributions.

Table 8.1: Summary of contributions.

Limitations in State-of-the-art (Ch. 2)	Ch. describing the contribution	Confirmed Hypothesis	Supporting Evidence
Corpus Restriction: previous works have special corpus requirement e.g. parallel corpus, web as a corpus.	Ch. 3 Bootstrap Paraphrase Acquisition Framework	It is possible to extract paraphrase templates from an unstructured monolingual corpus given seed instances.	BPL & D-BPL outperforms the baselines in precision, recall and number of distinct keywords (Section 7.3 in Chapter 7).
Semantic Drift: bootstrap pattern-instance learning can easily mess up with erroneous or ambiguous item.	Ch. 4 Preventing Semantic Drift	Semantic drift risk from diversification be mitigated by distributional type restriction.	When type-based instance filtering is enabled, precision is constantly above the baseline and does not steeply drop (Section 7.4.1 in Chapter 7).
Lack of Lexical Diversity: preventing semantic drift too much results in extracting patterns with poor lexical diversity.	Ch. 5 Diversifying Lexicons in Paraphrase Bootstrapping	Lexical diversity of acquired paraphrase can be controlled with a model of relevance-dissimilarity interpolation.	A statistically significant difference ($p < 0.05$) in DIMPLE was observed between the diversifiable bootstrapping and the baselines (Section 7.3 in Chapter 7).
Lack of Evaluation Metric: precision or recall does not reward lexical diversity	Ch. 6 Diversity-aware Evaluation Metric for Paraphrase Patterns	Cumulative-gain style evaluation metric which gives reward to lexically diverse paraphrases is effective for paraphrase evaluation.	The DIMPLE metric correlates with paraphrase recognition task performance, with a Pearson correlation of +0.5 $\sim$ +0.7 with a statistical significance in existence of correlation ($p < 0.01$) (Section 6.3 in Chapter 6).

One of the most critical limitations in the state-of-the-art paraphrase extraction algorithms is an ability to detect acquire lexically diverse paraphrases. The contribution of this thesis includes proposing an evaluation metric *DIMPLE* that can distinguish lexically less diverse patterns (e.g. "*X* died of *Y*", "*X* has died of *Y*", "*X* was dying of *Y*", "*X* died from *Y*", "*X* was killed in *Y*") against lexically diverse patterns (e.g. "*X* succumbed to *Y*", "*X* fell victim to *Y*", "*X* suffered a fatal *Y*", "*X* was terminally ill with *Y*", "*X* lost his long battle with *Y*", "$X(writer)$ wrote his final chapter *Y*"). In addition, we proposed a paraphrase extraction algorithm *Diversifiable Bootstrapping* which can explicitly control lexical diversity of paraphrases to be acquired.

In our experiment, we extracted paraphrases from an unstructured monolingual corpus given a small number of seed instances. As a result of the experiment, a statistically significant difference in DIMPLE was observed between the Diversifiable Bootstrapping (D-BPL) and the two baseline algorithms (D-BPL without diversification and vanilla Espresso). This evidence supports the hypothesis that the proposed diversification method is effective in acquiring paraphrases with lexical diversity.

8.1 Future Works

In this section, we will summarize the future direction of the works extending ideas presented in the thesis.

Co-reference Resolution

Counting co-occurrences of pattern and instance in a corpus is one of the most important processes when scoring patterns and instances. However, instances do not always appear in the same form in the article; an entity can be referred using pronouns and so on:
Reference by pronoun or general noun: Use of pronoun, as seen in the example (1a) below, is one of the most common issues that causes a pattern-instance co-occurrence to be miss-counted in text. Our analysis shows that, out of 1413 Wikipedia sentences that contain the phrase *was murdered by*, 256 sentences were identified to have "he" or "she" as the *X* slot-filler.

(1) a. Eventually he *was murdered by* Kusru Khan in 1321 AD which marked the beginning of the end of Khilji Dynasty paving the way for the Thuqlaq dynasty to establish control over sultanate of Delhi and much of northern India.

b. At Delft, the duke *was murdered by* revolutionaries (February 26, 1076).

First name or last name only: In the example (2a) below, we can see the underlined part is the first name, not the full name. It is uncommon that a full name of a person is repeated over and over within the same document.

(2) a. Ashley was married to Maxine Peacock in 1999, but he ended up a widower when Maxine *was murdered by* Richard Hillman.

In order to make the co-occurrence calculation more accurate and robust, we need a mechanism to count entity occurrence considering co-reference, rather than counting mentions of entities.

Corpus-specific paraphrase extraction

In our experiment, we extracted paraphrases from Wikipedia. It is an encyclopedia corpus, and we were able to extract some expressions specific to it, such as " X (d. Y " where d. is an abbreviation for decease (more examples from `person_death_date` relation are available in Appendix A.3). One important future direction of this work would be to extract corpus-specific paraphrases from various different sources, such as from domain-specific corpora (e.g. medicine, legal, sports), from corpora with different writing style (e.g. social media, speech transcript), or even from corpora in different languages.

Feature-based Trainable Scorer

High-precision-instance extractability is an important requirement of a pattern during bootstrapping. On the other hand, attribute of an ideal pattern changes after convergence, depending on application (e.g. grammaticality preservation, lexical diversity, etc). Since the value of paraphrase can vary, it would be great to have a trainable pattern scorer that can adapt to varying paraphrase valuation needs. There are many possible clues we may be able to utilize for using as features, not just co-occurrence statistics (PMI) and lexical diversity we used in the thesis, for example, selectional preference of instances, or contextual preference of patterns.

Appendix A

Example Paraphrases

This appendix shows the following: original seed, type vector generated from the seed, and extracted paraphrase data. The table labeled D-BPL is from the Diversifiable Bootstrap Paraphrase Learner proposed in this thesis. The overview of experimental settings are described in Section 7.2.3 in Chapter 7.

A.1 Relation: `LEADER`

Table A.1: Original seed and types extracted for LEADER.

(a) *The seed instances (exhaustive)*

X	*Y*
India	Rajiv Gandhi
Australia	Paul Keating
Vichy France	Marshal Petain
United Kingdom	Elizabeth II
Cuba	Fidel Castro
Microsoft	Bill Gates
Uganda	Idi Amin

(b) *Type vector elements by weight (selected top 10; used by BPL and D-BPL)*

X		*Y*	
39.17	system	24.74	head of state
37.65	economy	24.59	leader
35.39	state	23.88	representative
24.66	country	23.86	negotiator
17.43	group	23.15	worker
16.67	administrative district	21.97	communicator
16.32	district	21.40	president
13.74	democracy	19.35	skilled worker
12.99	abstraction	17.43	holder
11.48	region	17.41	owner

Table A.2: Ranked list of extracted paraphrases by bootstrapping.

(a) *VANILLA (top 30 @ itr 10)*	(b) *BPL (top 30 @ itr 10)*	(c) *D-BPL (top 30 @ itr 10)*
Y , President of X	Y , president of X	Y , president of X
Y , president of X	Y , former president of X	Y ’s regime in X
X - Y , President	president of X , Y	Y ’s government in X
Y , former president of X	X ’s president Y	X an leader Y
president of X , Y	Y , the president of X	X an dictator Y
President of X , Y	Y (president of X	Y to X to face trial
X n president Y	president Y of X	Y (captain general, X
X ’s President Y	X ’s president, Y	Y from power in X
X ’s president Y	Y , the current president of X	X , led by Y
Y , the president of X	Y is elected president of X	banned in X during Y
Y (president of X	X ian president Y	invaded and annexed by X (under Y
president Y of X	Y - president of X	war against Y ’s X
President Y of X	Y , the former president of X	Y to the presidency of X
Y - former president of X	Y becomes president of X	Y is made premier of X
X ’s president, Y	Y , current president of X	unification with Y ’s X
Y - president of X	Y , first president of X	supported Y ’s X
Y , the current president of X	Y as president of X	Y after the invasion of X
Y , the former president of X	Y was elected president of X	X , started after the removal of Y
Y is elected president of X	X an president Y	Y ’s rule in X
Y becomes president of X	Y , then president of X	X and met with Y
Y , current president of X	X , president Y	Y ’s revolution in X
Y , first president of X	president of X is Y	Y of X , represented
X ian president Y	Y takes office as president of X	rights in Y ’s X
presidents Y of X	X president Y	X ’s prime minister, Y
Presidents Y of X	Y , the then president of X	Y of X has declared
Y as president of X	election of Y as president of X	Y , X an politician
Y takes office as president of X	Y former president of X	Y dictatorship, in X
election of Y as president of X	Y became president of X	X since the fall of Y
Y was elected president of X	Y as the president of X	X in the era of Y
President Y of X and President	Y , who was president of X	Y of X awarded

Table A.3: “O” labeled paraphrases from the pool (selected).

Y wins the presidential elections in X	Y for thwarting a communist revolution in X
government of X during the reign of Y	Y of X crowned
X (during Y ’s rule	uncrowned Y of X
made a peer of X by Y	inaugurated as Y of X
Y came to power in X	X occupied Y
Y was firmly in power in X	X ’s relinquishing control of Y
Y , father of the nation, X	influential Y in X
Y declared X ’s independence	Y ’s accession to the throne of X
X under Y ’s regime	X on behalf of Y
kingdom of X at the time of Y	behest of Y of X
Y ’s reign of terror in X	Y acceded to X
deputy- Y of X	Y ’s partisans in X
Y of X , who ordered	Y of X was overthrown
order of Y of X	consul Y of X
takes office as Y of X	Y , the khedive of X
fought between Y , who later founded X	Y , the viceroy of X
X during Y ’s era	cardinal Y of X
X in the era of Y	Y dissolved X
Y received his charter for X	initiative of Y of X
Y leads a successful coup in X	Y conquers X

A.2 Relation: `person_graduated_school`

Table A.4: Original seed and types extracted for person_graduated_school.

(a) *The seed instances (exhaustive)*

X	*Y*
Akishino	Gakushuin University
Charles Anderwald	St . Gerard High School
Diane Price Baker	Barnard College
He	Auburn University
he	Brooklyn College
he	Brown University
He	Harvard
He	Johns Hopkins
he	University of Michigan
He	University of Toronto
James T . Tierney	Brown University
Joseph F . Unanue	Duke University
Joseph F . Unanue	University of North Carolina
Kim B . Clark	Harvard
Patricia Lapre	Bristol Community College
She	Barnard College
she	Bristol Community College
Sol Hoffman	Brooklyn College
Sol Hoffman	University of Michigan
Solomon D . Erulkar	Johns Hopkins
Solomon D . Erulkar	University of Toronto
Unanue	Duke University
Unanue	University of North Carolina
Vincent Jackson	Auburn University

(b) *Type vector elements by weight (selected top 10; used by BPL and D-BPL)*

	X		*Y*
112.7	personal pronoun	101.5	educational institution
107.8	pronoun	96.76	body
105.9	function word	85.28	institution
78.12	word	63.44	social group
56.43	language unit	61.02	group
48.89	part	59.87	organization
40.84	relation	46.14	college
19.49	abstraction	45.48	abstraction
9.39	ornithologist	41.05	yagoLegalActor
8.52	receiver	25.93	university

Table A.5: Patterns by NELL CPL (first 100)

X is a graduate of Y	X went to San Diego Y	X attended Regis Y
X paid her way through Y	X Arbor and Michigan Y	X is a graduate of Sheridan Y
X of Arms from Y	X attended Manhattan Y	X attended Georgia Y
Y students Nick X	X dropped out of Y	X mckewan Y
X graduated from Marist Y	X W Bush at Y	X attended Grinnell Y
X Dwight of Y	X attended Arizona Y	Y buddy Michael X
Y of William and X	X worked at Boston Y	X attended Franklin Y
X graduated from Pennsylvania Y	X graduated from Carroll Y	X is a graduate of Montana Y
X chrisitan Y	X graduated from the Ontario Y	X Perry at Y
X College and Massachusetts Y	X attended Dixie Y	X Gray of Colorado Y
X graduated from Harvard Y	X is a graduate of Lafayette Y	X graduated from Knox Y
X College and Hampshire Y	X is a graduate of Bethany Y	X waugaman Y
Y under the GI X	X attended Fullerton Y	X attended California Y
X McDougall at Y	Y activities director X	X State University and Cleveland Y
X is a graduate of Oregon Y	X is a graduate of Manhattan Y	X attended Knox Y
X graduated from Sam Houston Y	X graduated from Middlebury Y	Y transfer Anthony X
X attended Ball Y	X Biden will make Y	Y student in Santa X
X attended Hunter Y	X Witcombe of Sweet Briar Y	X peterson wittenburg Y
X is a graduate of Bethel Y	X graduated from Oregon Y	X University and Providence Y
X attended Texas Y	X was educated at Magdalen Y	X attended Fresno State Y
X is a graduate of Kent Y	X was at art Y	X enrolled at Ohio Y
X is a graduate of Pomona Y	X and Mary and Virginia Y	X is a graduate of Brooklyn Y
X attended Rhode Island Y	X attended Bard Y	X Community College and Michigan Y
X McCain graduated from Y	Y University and won X	X lyman Y
X attended Luther Y	X is a graduate of Mississippi Y	X kissel Y
Y at Dakota Wesleyan University in X	X Conley from Y	X attended Vassar Y
X graduated from Smith Y	X attended Washington Y	X attended Emerson Y
X Bellinger of Y	X graduated from Concordia Y	X attended Humboldt Y
X attended Community Y	X attended Jefferson Y	Y of Biological Sciences at UC X
X spent time after Y	X completed his undergraduate degree at Y	Y Law School with X
X Hall and provides Y	X attended Antioch Y	X peru state Y
Y pioneer drive X	X academy caldwell nj Trinity Y	X graduated from Seneca Y
X was Professor Emeritus at Y	X attended Barnard Y	X was educated at Malvern Y
		X s jesuit Y

Table A.6: Ranked list of extracted paraphrases by bootstrapping.

(a) *VANILLA (top 30 @ itr 10)*
High School, X attended Y
high school, X attended Y
School, X attended Y
school, X attended Y
X attended Y
X graduated from Y
graduating from high school, X attended Y
high school, X attended Y where he played
X attended Y where he played
Y , where X majored
attended Y , where X
X taught at Y
X has taught at Y
X is a graduate of Y
X received his undergraduate degree from Y
X studied at Y
English at Y , where X
Y , where X graduated
X later attended Y
X graduated at Y
attended Y where X
X graduated from Y in 1951
Y , where X earned
Y , where X
X graduated from Y in Providence
X was graduated from Y , Providence, Rhode Island
X joined the faculty at Y
X graduated from Y , Providence, Rhode Island
X attended Y and graduated
X graduated with honors from Y

(b) *BPL (all 29 @ itr 10)*
X graduated from Y
X is a graduate of Y
X has taught at Y
attended Y , where X
Y , where X majored
X attended Y
X taught at Y
X received his undergraduate degree from Y
X studied at Y
X graduated at Y
Y , where X graduated
high school, X attended Y
X joined the faculty at Y
X graduated with honors from Y
X was graduated from Y
X was a visiting professor at Y
X later attended Y
attended Y where X
Y , where X studied
Y , where X earned
X was also a visiting professor at Y
studied at Y where X
X was a professor at Y
Y , where X received
studied at Y where X received
attended Y , where X graduated
X graduated cum laude from Y
Y where X received
educated at Y where X graduated
graduated from Y , where X

(c) *D-BPL (top 30 @ itr 10)*
X graduated from Y
attended Y , where X
X has taught at Y
Y , where X majored
X received his undergraduate degree from Y
X joined the faculty at Y
X studied at Y
X was a visiting professor at Y
Y , where X earned
X accepted a position at Y
X then went to Y
Y , where X was a member
X played college football for Y
X is a graduate of Y
science at Y , where X
president of Y , where X served
transferred to Y where X
history at Y , where X
X was the head coach at Y
X matriculated at Y
Y , where X led
X became an overseer of Y
X enrolled at Y
X entered Y
Y , where X obtained
X attended Y
alumnus of Y , where X
X had met at Y
X was educated at Y
Y , where X came

A.3 Relation: `person_death_date`

Table A.7: Original seed and types extracted for person_death_date.

(a) *The seed instances (exhaustive)*

X	*Y*
Emperor Meiji	1912
FDR	1945
Franklin D . Roosevelt	1945
John Bonham	1980
Jonas Salk	1995
Kim Il Sung	1994
Mahatma Gandhi	1947
Pope John Paul II	April 2005

(b) *Type vector elements by weight (selected top 10; used by BPL and D-BPL)*

X		*Y*	
34.91	militant	78.63	assassin
33.69	reformer	66.17	murderer
33.57	disputant	65.32	killer
30.77	intellectual	57.14	principal
29.88	politician	56.03	wrongdoer
29.81	adult	55.98	bad person
28.69	leader	50.72	prisoner
27.67	alumnus	50.00	criminal
26.55	scholar	45.34	unfortunate
25.91	unfortunate	20.75	causal agent

Table A.8: Patterns by NELL CPL (exhaustive)

X died in July Y
Y after the death of drummer X
Y after drummer X
X died in office in Y
X died in April Y

Table A.9: Ranked list of extracted paraphrases by bootstrapping.

(a) *VANILLA (top 30 @ itr 8)*
Death of X in Y
death of X in Y
X 's death in Y
X 's death on Y
X 's death on Y 5
X 's death in Y 1953
death of X in Y 1953
X 's death in Y 1953, Beria
X died on Y
X died on Y 5
Joseph X in Y 1953
X in Y 1953
X died on Y 5, 1953
X 's death on Y 5, 1953
death of Joseph X in Y
death of Joseph X in Y 1953
X died in Y 1953
Joseph X 's death in Y 1953
X died in Y
death of X in Y 1976
Joseph X in Y
X 's death (Y
X 's death (Y
X 's death in Y 1994
death of X in Y 1991
X , on Y 6
Kurt X in Y 1994
Y hosted the grave of X
sometimes considered the death of X in Y
X 's death on Y 21

(b) *BPL (all 29 @ itr 5)*
death of X in Y
vacant since the death of X in Y
X (d. Y
followed by the death of X in Y
X 's death in Y
X died in Y
funeral of X in Y
Y following the death of X
murder of X in Y
X in late Y
X , who died in Y
Y murder of X
X (died Y
X 's Y album
X 's Y song
born Y), manga X
Y) (layout X
born Y), contemporary X
Y execution of X
Y , X was found guilty
X (died in Y
son, the second X , who succeeded in Y
X (b. Y
X) (b. Y
X died Y
film, released in Y , directed by X
Y novel by X
selected X , and again voted in Y conclave
Y), (contemporary X

(c) *D-BPL (top 30 @ itr 4)*
death of X in Y
X (d. Y
X died in Y
funeral of X in Y
murder of X in Y
born Y), manga X
Y) (layout X
X in late Y
son, the second X , who succeeded in Y
Y execution of X
Y , X was found guilty
X 's Y album
X 's Y song
X (b. Y
film, released in Y , directed by X
Y novel by X
selected X , and again voted in Y conclave
Y), (contemporary X
vacant since the death of X in Y
X 's death in Y
followed by the death of X in Y
X , who died in Y
X 's death, in Y
Y , following the death of X
Y , after the death of X
X (died Y
Y following the death of X
Y murder of X
born Y), contemporary X
Y after the death of X

Appendix B

Paraphrase Annotation Guideline

B.1 Task Description

This annotation task is about assigning labels. Specifically, given a set of **binary-argument templates** (hereafter patterns) that may or may not represent a specific **relation** between the two arguments, the task is to assign the appropriate label.

A **binary-argument pattern** is a segment of string with variables X and Y. For example, in the CAUSEOFDEATH relation, a pattern is “X died of Y”, “X passed away from Y”, “due to Y, X died” etc.

Concrete examples of X and Y are shown in the **Seed** section in the UI.

B.2 Label Definition

B.3 Examples by Label

B.3.1 Label M (Matched)

Relation = bankboughtbank

- (M) X acquired Y
- (M) X which acquired Y
- (M) X acquired Y and then
Note: additional string “and then” is appended, but it doesn’t change the meaning.
- (M) X acquired Y in DDDD
Note: DDDD indicates a date.
- (M) X eventually acquired Y
- (M) finally, X unexpectedly acquired Y
- (M) Y was acquired by X
- (M) Y (acquired by X

LABEL	DESCRIPTION
M	**M**atched: If the X and Y in the pattern is instantiated with concrete values (as seen in the Seed section in the judge UI), it is likely to match this criteria (**high certainty**): the template represents the intended meaning. (From a researcher's point of view, the patterns with this label are a set of "paraphrase templates").
O	Matched and **O**OD (Matched and Out-of-dictionary): A pattern is matched, AND its keyword is not a synonym according to WordNet. It could be a colloquial, metaphorical, idiomatic, or euphamistic expression. (From a language resource acquisition researcher's point of view, this kind of pattern is very valuable as it's worth finding automatically from a corpus.)
I	**I**nconclusive: It may or may not "match" depending on the context of the pattern in sentences. (**medium certainty**)
R	**R**elated: Even if instantiated with the correct slot values, the pattern does not represent the intended meaning (**no or very small certainty**). However, the pattern represents a related fact/event that may occur between X and Y.
A	**A**ntonym: It has the opposite meaning as "M"-label patterns.
W	**W**rong: None of the above.
D	**D**efer decision. The annotator looked at the pattern, but postponed a decision for now. The difference from "I" is that "I" has been the decision that the pattern is inconclusive, whereas "D" just means no decision is made yet.
-	The annotator hasn't looked at this pattern yet.

- (M) X's acquisition of Y

Relation = attack

- (M) X attacked Y
- (M) war between X and Y
- (M) X-Y conflict began
- (M) X was at war with Y
- (M) fighting broke out between X and Y
- (M) X conquered Y
- (M) X fought against Y

B.3.2 Label O (Matched and out-of-dictionary)

Relation = persongraduatedschool

- (M) X graduated from Y
- (O) X holds a BA from Y
- (O) X completed studies at Y

Relation = acquire

- (M) X acquired Y

- (O) Y came under the ownership of X

Relation = CAUSEOFDEATH

- (M) X died of Y
- (M) X passes to Y
- (M) X perished in Y
- (M) X succumbed to Y
- (O) X fell victim to Y
- (O) X was terminally ill with Y
- (O) X suffered a fatal Y

  Note: content words in above patterns with (O) are not synonyms of death (and die) in WordNet.

Relation = arrest

- (M) X was arrested by Y
- (M) X was detained by Y
- (M) X was captured by Y
- (M) X was recaptured by Y
- (I) X was released by Y
- (O) X was taken into custody by Y
- (O) X turned himself in to Y

B.3.3 Label R (Related)

Relation = bankboughtbank

- (M) X acquried Y
- (R) X-Y deal could

  Note: A deal between two banks can refer to a number of possible financial deals, not only an acquisition deal. Therefore there's ***very small certainty*** *that this deal is an acquisition. However, an acquisition is a type of deal between banks, so deal is "related" to acquisition and the "R" label is appropriate.*

  Relation = PLACEOFBIRTH

- (R) X grew up in Y
- (R) X lived in Y
- (R) X lives in Y

  Relation = persongraduatedschool

- (M) X graduated from Y
- (R) X enrolled in Y
- (R) X is currently a senior in Y
- (R) X took a semester off from Y

- (R) X attended Y
  Note: Not all people enrolled in/attending school eventually graduate.

Relation = ATTACK
- (M) X attacked Y
- (R) X confronts Y
- (R) X besieged Y
- (R) Y was encircled by X
- (R) Y was occupied by X
- (R) Y took control of X
- (R) X signed an armistice with Y
- (R) X and Y signed a peace treaty
  Note: Between two warring countries, a peace treaty can be made to end the fighting.

Relation = FOUND
- (M) X founded Y
- (R) X left Y
  Note: A founder may leave the company some years later.

B.3.4 Label I (Inconclusive)

Relation = acquired
- (M) X purchased Y
- (I) X-Y merger
  Note: The template is ambiguous. "X-Y merger" might have occurred as a result of the bank X buying bank Y. However, it's possible that "X-Y merger" occurred because Y bought X. Without context, both cases are possible. So label I (Inconclusive) is appropriate.
- (I) X merged with Y
- (I) Y merged with X
- (I) X, the parent company of Y
  Note: Being a parent company can be due to a business purchase, but other possibilities also exist (e.g. the parent company could have created a child company). So the label I (Inconclusive) is appropriate.

Relation = bankboughbank
- (M) X acquired Y
- (I) X) acquired Y
  Note: Here's a sample text that could make this not at all mean that XboughtY "We were told that Z (who had always hated X) acquired Y, X's main competitor."
- (I) X, acquired Y
  Note: Here's a sample text that could make this not at all mean that X bought Y "Z, which

is located near X, acquired Y.”

Relation = KILLER

- (I) the possibility of Y to assassinate X
- (I) Y attempt to assassinate X
- (I) Y plot to assassinate X
- (I) Y tried to kill X
- (I) failed assassination attempt on X, in Y

  Note: A possibility/attempt/plot to kill doesn’t guarantee a killing event actually happened.

B.3.5 Label A (Antonym)

Relation = bankboughtbank

- (A) Y did not eventually acquire X
- (A) Y’s failed acquisition attempt of X
- (A) block Y’s acquisition of X

Relation = persongraduatedfromschool

- (M) X graduated from Y
- (A) X left Y

B.3.6 Label W (Wrong)

Relation = CAUSEOFDEATH

- (M) X died from Y
- (W) X-in-law died from Y

  Note: “X” in “X-in-law” didn’t actually die.

Relation = persongraduatedschool

- (M) X is a graduate of Y
- (W) X is a graduate of Pennsylvania Y

  Note: The pattern is very restrictive and lacks generality. When seeds are inserted in Y, it doesn’t make sense.

B.4 Examples of borderline cases

B.4.1 Argument order difference

Depending on the relation, a variable order difference can result in a different label.

Relation = acquired

- (M) X acquired Y

- (W) Y acquired X
  Note: It's never likely that a company being purchased buys its parent company.

Relation = ATTACK

- (M) X attacked Y
- (R) Y attacked X
  Note: Between two fighting entities, a retaliatory attack is possible.

Relation = hassister

- (M) X's sister Y
- (M) Y's sister X
  Note: X and Y can be swapped for this relation.

B.4.2 Slight Differences

Relation = CAUSEOFDEATH

- (M) X was diagnosed with Y, and died
- (R) X was diagnosed with Y
  Note: Simply being diagnosed with Y doesn't mean X died of it.
- (R) X became ill with Y
- (R) X became very ill with Y
- (O) X became terminally ill with Y

B.4.3 Labels compared by certainty level

Sometimes, label decisions must be made with respect to a subjective certainty level.
Relation = bankboughtbank

- (M) X stepped in to buy Y
  Note: fairly certain that it occurred.
- (I) X to buy Y
  Note: whether or not it actually occurred is uncertain.
- (I) X buying Y
  Note: whether or not it actually occurred is uncertain.
- (I) X said it agreed to buy Y
  Note: occurrence likelihood is 50-50.
- (I) proposed acquisition of Y buy X
  Note: not at all certain that it occurred.
- (I) X announced it would acquire Y
  Note: not at all certain that it occurred.

B.4.4 Present Tense

Present tense may be “M” or “I” depending on the certainty (fairly certain or not).

Relation = bankboughtbank

- (I) Y is being bought by X
  Note: Especially for this relation (acquisition of bank/company), it’s possible that this event won’t occur, since hostile takeovers of companies often fail.

Relation = CAUSEOFDEATH

- (I) X who is dying of Y

Bibliography

Eugene Agichtein and Luis Gravano. Snowball: extracting relations from large plain-text collections. In *Proceedings of ACL 2000*, 2000. 2.1

Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. Diversifying search results. In *Proceedings of WSDM 2009*, pages 5–14, 2009. 5.2.1

Ion Androutsopoulos and Prodromos Malakasiotis. A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research*, 38:135–187, 2009. 2.5.1

Collin Baker, Charles Fillmore, and John Lowe. The berkeley framenet project. In *Proceedings of COLING-ACL 1998*, 1998. 2.4, 4.4

Satanjeev Banerjee and Ted Pedersen. An adapted lesk algorithm for word sense disambiguation using wordnet. In *Proceedings of CICLING 2002*, pages 136–145. Springer, 2002. 4.1

Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. Open information extraction for the web. In *IJCAI*, volume 7, pages 2670–2676, 2007.

Colin Bannard and Chris Callison-Burch. Paraphrasing with bilingual parallel corpora. In *Proceedings of ACL 2005*, 2005. 1.1, 2.1, 2.2.2, 2.5, 6.1, 6.3.1

Regina Barzilay and Kathleen McKeown. Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL-2001)*, 2001. 1, 1, 2.2.1

Regina Barzilay, Kathleen R. McKeown, and Michael Elhadad. Information fusion in the context of multi-document summarization. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 550–557, 1999. 1

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155, 2003. 3.2.2

Rahul Bhagat. *LEARNING PARAPHRASES FROM TEXT*. PhD thesis, FACULTY OF THE GRADUATE SCHOOL, UNIVERSITY OF SOUTHERN CALIFORNIA, CA, USA, 2009. 1

Rahul Bhagat and Deepak Ravichandran. Large scale acquisition of paraphrases for learning surface patterns. In *Proceedings of ACL-08: HLT*, pages 674–682, 2008. 1.3, 2.1, 2.1, 3.2.2, 6.3

Sacaleanu Bogdan, Constantin Orasan, Christian Spurk, Shiyan Ou, Óscar Ferrández, Milen Kouylekov, and Matteo Negri. Entailment-based question answering for structured data. In *Proceedings of COLING '08 22nd International Conference on on Computational Linguistics: Demonstration Papers. Manchester, UK*, 2008. 1

Vinayak Borkar, Kaustubh Deshmukh, and Sunita Sarawagi. Automatic segmentation of text into structured records. In *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data*, SIGMOD '01, pages 175–186, New York, NY, USA, 2001. ACM. ISBN 1-58113-332-4. doi: 10.1145/375663.375682. URL `http://doi.acm.org/10.1145/375663.375682`. 3.3

Sergey Brin. Extracting patterns and relations from the world wide web. In *WebDB Workshop at 6th International Conference on Extending Database Technology, EDBT 1998*, pages 172–183, 1998. 2.1

Susan Windisch Brown. Choosing sense distinctions for wsd: Psycholinguistic evidence. In *Proceedings of ACL-08: HLT, Short Papers*, 2008. 2.4.1

Steven Burrows, Martin Potthast, and Benno Stein. Paraphrase acquisition via crowdsourcing and machine learning. *(to appear) Transactions on Intelligent Systems and Technology (ACM TIST)*, 2012. 1

Mary Elaine Califf and Raymond J. Mooney. Bottom-up relational learning of pattern matching rules for information extraction. *Journal of Machine Learning Research*, pages 177–210, 2003. 2.1

Chris Callison-Burch. Syntactic constraints on paraphrases extracted from parallel corpora. In *Proceedings of EMNLP 2008*, 2008. 1.1, 1.3, 2.2.2, 2.5, 6.1, 6.2.2, 6.3.1

Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. (meta-) evaluation of machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation - StatMT 2007*, 2007. 6.3

Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. Further meta-evaluation of machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation - StatMT 2008*, 2008. 6.3

Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of SIGIR 1998*, pages 335–336, 1998. 5.2.1

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr., and Tom M. Mitchell. Toward an architecture for never-ending language learning. In *Proceedings of AAAI 2010*, 2010a. 2.3.1, 4.4, 7.2.1, 7.2.3

Andrew Carlson, Justin Betteridge, Richard C. Wang, Estevam R. Hruschka, Jr., and Tom M. Mitchell. Coupled semi-supervised learning for information extraction. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, pages 101–110, New York, NY, USA, 2010b. ACM. ISBN 978-1-60558-889-6. doi: 10.1145/1718487.1718501. URL `http://doi.acm.org/10.1145/1718487.1718501`. 7.2.3

David L. Chen and William B. Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 190–200, 2011. 1.1, 2.2.1, 2.5

Timothy Chklovski. Collecting paraphrase corpora from volunteer contributors. In *Proceedings* *of the 3rd international conference on Knowledge capture*, 2005. 2.4

Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29, 1990. 3.2.1, 4.3.1, 5.1.2

Rudi L Cilibrasi and Paul MB Vitanyi. The google similarity distance. *Knowledge and Data Engineering, IEEE Transactions on*, 19(3):370–383, 2007. 4.3.1

Charles L. A. Clarke, Gordon V. Cormack, M. Laszlo, Thomas R. Lynam, and Egidio L. Terra. The impact of corpus size on question answering performance. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2002. 2.1.1

Stéphane Clinchant, Cyril Goutte, and Eric Gaussier. Lexical entailment for information retrieval. In *Lecture Notes in Computer Science, 2006, Volume 3936/2006*, pages 217–228, 2006. 1

James R. Curran, Tara Murphy, and Bernhard Scholz. Minimising semantic drift with mutual exclusion bootstrapping. In *Proceedings of the Conference of the Pacific Association for Computational Linguistics (PACLING)*, pages 172–180, 2007. 1.1, 2.3.2, 3.1.2

Daniel Dahlmeier and Hwee Tou Ng. Correcting semantic collocation errors with l1-induced paraphrases. In *Proceedings of EMNLP 2011*, pages 107–117, 2011. 1

Tiphaine Dalmas. *Information Fusion for Automated Question Answering*. PhD thesis, University of Edinburgh, Edinburgh, Scotland, 2007. 1

Hoa Trang Dang, Jimmy Lin, and Diane Kelly. Overview of the trec 2006 question answering track. In *Proceedings of TREC 2006*, 2006. 6.3.3

Hoa Trang Dang, Diane Kelly, and Jimmy Lin. Overview of the trec 2007 question answering track. In *Proceedings of TREC 2007*, 2007. 6.3.3

Dipanjan Das and Noah A. Smith. Paraphrase identification as probabilistic quasi-synchronous recognition. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, pages 468–476, 2009. 1

Robert-Alain de Beaugrande and Wolfgang V. Dressler. *Introduction to text linguistics*. Longman, 1981. 1

Bill Dolan, Chris Quirk, and Chris Brockett. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th International Conference on Computational Linguistics*, 2004. 1.1, 2.2.1, 6.3.3

William B Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005. 1.1, 2.2.1, 6.3.3

Mark Dras. *Tree Adjoining Grammar and the Reluctant Paraphrasing of Text*. PhD thesis, Department of Computing, Macquarie University, Australia, 1999. 1

Pablo Ariel Duboue and Jennifer Chu-Carroll. Answering the question you wish they had asked: the impact of paraphrasing for question answering. In *Proceedings of the Human Language*

Technology Conference of the NAACL, Companion Volume: Short Papers, pages 33–36, 2006. 1

Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam Mausam. Open information extraction: The second generation. In *IJCAI*, volume 11, pages 3–10, 2011.

Wu Fei and Daniel S. Weld. Autonomously semantifying wikipedia. In *Proceedings of CIKM 2007*, 2007. 2.3.2

Wu Fei and Daniel S. Weld. Automatically refining the wikipedia infobox ontology. In *Proceedings of WWW 2008*, 2008. 2.3.2

Wu Fei and Daniel S. Weld. Open information extraction using wikipedia. In *Proceedings of ACL 2010*, 2010. 2.3.2

Dayne Freitag. Multistrategy learning for information extraction. In *Proceedings of the ICML 1998*, ICML '98, pages 161–169, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 1-55860-556-8. URL `http://dl.acm.org/citation.cfm?id=645527.657302`. 3.3

Atsushi Fujita and Satoshi Sato. Computing paraphrasability of syntactic variants using web snippets. In *Proceedings of IJCNLP*, pages 537–544, 2008. 3.2.2

Gregory Grefenstette. Explorations in automatic thesaurus discovery. *Kluwer International Series in Engineering and Computer Science*, 278, 1994. 3.2.2

Sanda Harabagiu and Andrew Hickl. Methods for using textual entailment in open-domain question answering. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, 2006. 1

Zellig Harris. Distributional structure. *Word*, 10(23):146–162, 1954. 3.2.2

Chikara Hashimoto, Kentaro Torisawa, Stijn De Saeger, Jun'ichi Kazama, and Sadao Kurohashi. Extracting paraphrases from definition sentences on the web. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, pages 1087–1097, 2011. 2.2.1

Michael Heilman. *Automatic Factual Question Generation for Reading Assessment*. PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, 2011. 2.2.3

Ulf Hermjakob, Abdessamad Echihabi, and Daniel Marcu. Natural language based reformulation resource and web exploitation for question answering. In *Proceedings of TREC-2002*, 2002. 1

Graeme Hirst and David St-Onge. Lexical chains as representations of context for the detection and correction of malapropisms. *WordNet: An electronic lexical database*, 305:305–332, 1998. 4.1

Johannes Hoffart, Fabian Suchanek, Klaus Berberich, and Gerhard Weikum. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. *Special issue of the Artificial Intelligence Journal*, 2012. 3.3, 4.2.3

Ali Ibrahim, Boris Katz, and Jimmy Lin. Extracting structural paraphrases from aligned monolingual corpora. In *Proceedings of the Second InternationalWorkshop on Paraphrasing (ACL*

2003), 2003. 1, 1, 2.2.1

Diana Inkpen. A statistical model for near-synonym choice. *ACM Transactions on Speech and Language Processing*, 4:1–17, 2007. 1

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. In *ACM Trans. Inf. Syst., Vol. 20, No. 4. (October 2002)*, pages 422–446, 2002. 6.2.1

Jay J Jiang and David W Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. 1997. 4.1

Hideo Joho, Leif Azzopardi, and Wim Vanderbauwhede. A survey of patent users: An analysis of tasks, behavior, search functionality and system requirements. In *Proceedings of the Third Information Interaction in Context Symposium (IIiX 2010)*, pages 13–24, 2010. 1

John S. Justesona and Slava M. Katz. Technical terminology: some linguistic properties and algorithms for identification in text. In *Proceedings of ICCL-95*, pages 539–545, 1995. 3.2.1, 4.3.1

Hiroshi Kanayama and Tetsuya Nasukawa. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *Proceedings of EMNLP 2006*, 2006. 3.4.1

Boris Katz, Jimmy Lin, Daniel Loreto, Wesley Hildebrandt, Matthew Bilotti, Sue Felshin, Aaron Fernandes, Gregory Marton, and Federico Mora. Integrating web-based and corpus-based techniques for question answering. In *Proceedings of the Twelfth Text REtrieval Conference, 2003*, 2004. 2.1.1

David Kauchak and Regina Barzilay. Paraphrasing for automatic evaluation. In *Proceedings of HLT-NAACL 2006*, 2006. 1, 6.1

Jaana Kekäläinen. Binary and graded relevance in ir evaluations - comparison of the effects on ranking of ir systems. In *Information Processing and Management, 41*, pages 1019–1033, 2005. 6.2.1

Karen Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. Extending verbnet with novel verb classes. In *Proceedings of LREC 2006*, 2006. 2.4, 4.4

Tetsuo Kiso, Masashi Shimbo, Mamoru Komachi, and Yuji Matsumoto. Hits-based seed selection and stop list construction for bootstrapping. In *Proceedings of ACL 2011*, 2011. 2.3.2

Stanley Kok and Chris Brockett. Hitting the right paraphrases in good time. In *Proceedings of HLT-NAACL 2010*, 2010. 1.1, 1.3, 2.2.2, 2.5, 6.1, 6.3.1

Mamoru Komachi and Hisami Suzuki. Minimally supervised learning of semantic knowledge from query logs. In *Proceedings of IJCNLP 2008*, 2008. 2.3.1, 2.3.2, 3.4.1

Mamoru Komachi, Taku Kudo, Masashi Shimbo, and Yuji Matsumoto. Graph-based analysis of semantic drift in espresso-like bootstrapping algorithms. In *Proceedings of EMNLP 2008*, pages 1011–1020, 2008. 2.3.2

Milen O. Kouylekov. *Recognizing Textual Entailment with Tree Edit Distance: Application to Question Answering and Information Extraction.* PhD thesis, University of Trento, Computer Science, Electronics and Telecomunication, JAPAN, 2006. 1

Zornitsa Kozareva and Eduard Hovy. Not all seeds are equal: Measuring the quality of textmining seeds. In *Proceedings of NAACL-HLT 2010*, 2010. 2.3.2

Thomas K. Landauer and Susan T. Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104(2):211–240, 1997. 3.2.2

Claudia Leacock and Martin Chodorow. Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database*, 49(2):265–283, 1998. 4.1

Dekang Lin. Automatic retrieval and clustering of similar words. In *Proceedings of COLING-ACL 1998*, pages 768–774, 1998. 3.2.2, 4.1

Dekang Lin and Patrick Pantel. Dirt - discovery of inference rules from text. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 2001*, pages 323–328, 2001. 1.2.1, 2.1, 3.2.2, 3.4.1

Winston Lin, Roman Yangarber, and Ralph Grishman. Bootstrapped learning of semantic classes from positive and negative examples. In *Proceedings of the ICML-2003 Workshop on The Continuum from Labeled to Unlabeled Data*, pages 103–111, 2003. 2.3.2

Michael L. Littman, Jeffrey Bigham, and Victor Shnayder. Combining independent modules to solve multiple-choice synonym and analogy problems. In *Proceedings of RANLP-03*, pages 482–489, 2003. 3.2.2

Chang Liu, Daniel Dahlmeier, and Hwee Tou Ng. PEM: A paraphrase evaluation metric exploiting parallel texts. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 923–932, 2010. 2.5

Elena Lloret, Óscar Ferrández, Rafael Mu noz, and Manuel Palomar. A text summarization approach under the influence of textual entailment. In *Proceedings of the 5th International Workshop on Natural Language Processing and Cognitive Science (NLPCS 2008)*, 2008. 1

Catherine Macleod, Ralph Grishman, Adam Meyers, Leslie Barrett, and Ruth Reeves. Nomlex: A lexicon of nominalizations. In *Proceedings of EURALEX'98*, 1998. 2.4

Nitin Madnani and Bonnie J. Dorr. Generating phrasal and sentential paraphrases: A survey of data-driven methods. *Computational Linguistics*, 36, 2010. 1

Walid Magdy and Gareth J.F. Jones. Pres: A score metric for evaluating recall-oriented information retrieval applications. In *Proceedings of SIGIR 2010*, pages 611–618. ACM, 2010. ISBN 978-1-4503-0153-4. 1

Prodromos Malakasiotis and Ion Androutsopoulos. A generate and rank approach to sentence paraphrasing. In *Proceedings of EMNLP*, pages 96–106, 2011. ISBN 978-1-937284-11-4. 1

Aurélien Max and Guillaume Wisniewski. Mining naturally-occurring corrections and paraphrases from wikipedia's revision history. In *Proceedings of LREC 2010*, 2010. 2.4

Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the HLT-NAACL 2003*, CONLL '03, pages 188–191, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1119176.1119206. URL http://dx.doi.org/10.3115/1119176.1119206. 3.3

Tara Mcintosh. *Reducing Semantic Drift in Biomedical Lexicon Bootstrapping*. PhD thesis, The University of Sydney, Australia, 2009. 5.2.2

Tara McIntosh. Unsupervised discovery of negative categories in lexicon bootstrapping. In *Proceedings of EMNLP 2010*, 2010. 2.3.2

Tara McIntosh and James R. Curran. Weighted mutual exclusion bootstrapping for domain independent lexicon and template acquisition. In *Proceedings of the Australasian Language Technology Workshop*, 2008. 2.3.2

Tara McIntosh, Lars Yencken, Timothy Baldwin, and James Curran. Relation guided bootstrapping of semantic lexicons. In *Proceedings of ACL 2011*, 2011. 2.3.1

Donald Metzler and Eduard Hovy. Mavuno: A scalable and effective hadoop-based paraphrase harvesting system. In *Proceedings of the KDD Workshop on Large-scale Data Mining: Theory and Applications (LDMTA 2011)*, 2011. 1.3, 2.1, 2.1, 2.2.3, 6.3.1

Donald Metzler, Eduard Hovy, and Chunliang Zhang. An empirical evaluation of data-driven paraphrase generation techniques. In *Proceedings of ACL-HLT 2011*, 2011. (document), 2.1, 2.5, 6.1, 6.3.1, 6.3.2

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751, 2013.

George Miller and Walter Charles. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6:1–28, 1991. 3.2.2

Geroge A Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38: 39–41, 1995. 1, 2.4, 3.1.1, 4.1

Guido Minnen, John Carroll, and Darren Pearce. Applied morphological processing of english. *Natural Language Engineering*, 7(3):207–223, 2001. 4.3.1

Matteo Negri, Yashar Mehdad, Alessandro Marchetti, Danilo Giampiccolo, and Luisa Bentivogli. Chinese whispers: Cooperative paraphrase acquisition. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, 2012. 2.4

Rodney D. Nielsen, Wayne Ward, and James H. Martin. Recognizing entailment in intelligent tutoring systems. In *Natural Language Engineering, Volume 15 Issue 4, October 2009*, 2009. 1

Özlem Uzuner, Boris Katz, and Thade Nahnsen. Using syntactic information to identify plagiarism. In *Proceedings of the second workshop on Building Educational Applications Using NLP*, pages 37–44, 2005. 1

Marius Paşca and Péter Dienes. Aligning needles in a haystack: Paraphrase acquisition across the web. In *Processings of IJCNLP 2005*, pages 119–130, 2005. 2.1, 2.1, 1, 2.1, 6.3.1

Sebastian Padó, Michel Galley, Dan Jurafsky, and Christopher D. Manning. Robust machine translation evaluation with entailment features. In *Proceedings of ACL-IJCNLP 2009*, 2009. 1, 3.4.1, 6.1

Bo Pang, Kevin Knight, and Daniel Marcu. Syntax-based alignment of multiple translations: Extracting paraphrases and generating new sentences. In *Proceedings of the Human Language*

Technology Conference of the North American chapter of the Association for Computational Linguistics (HLT/NAACL-2003), 2003. 2.2.1

Pantel Pantel, Deepack Ravichandran, and Eduard Hovy. Towards terascale knowledge acquisition. In *Proceedings of COLING-04*, pages 771–777, 2004. 7.1

Patrick Pantel and Dekang Lin. An unsupervised approach to prepositional phrase attachment using contextually similar words. In *Proceedings of ACL 2000*, 2000. 3.2.2

Patrick Pantel and Marco Pennacchiotti. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of COLING-ACL 2006*, 2006. 1.2.1, 2.3.1, 2.3.2, 4, 3.2.2, 3.4.1, 4.3.1, 4.3.1, 4.3.2, 7.2.3

Patrick Pantel, Rahul Bhagat, Bonaventura Coppola, Timothy Chklovski, and Eduard H Hovy. Isp: Learning inferential selectional preferences. In *HLT-NAACL*, pages 564–571, 2007.

Patrick Pantel, Eric Crestan, Arkady Borkovsky, Ana-Maria Popescu, and Vishnu Vyas. Web-scale distributional similarity and entity set expansion. In *Proceedings of EMNLP 2009*, pages 938–947, 2009. 3.2.2

David Parapar, Alvaro Barreiro, and David E. Losada. Query expansion using wordnet with a logical model of information retrieval. In *IADIS AC 2005*, pages 487–494, 2005. 1

Martin F Porter. An algorithm for suffix stripping program. In *14(3)*, pages 130–137, 1980. 5.1.1, 6.3.2

Mark Przybocki, Kay Peterson, Sébastien Bronsart, and Gregory Sanders. The nist 2008 metrics for machine translation challenge–overview, methodology, metrics, and results. In *Machine Translation, Volume 23 Issue 2-3*, 2009. 3.4.1, 6.3

Chris Quirk, Chris Brockett, and William Dolan. Monolingual machine translation for paraphrase generation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP-2004)*, 2004. 1.1, 2.2.1, 6.3.3

Davood Rafiei, Krishna Bharat, and Anand Shukla. Diversifying web search results. In *Proceedings of WWW 2010*, pages 781–790, 2010. 5.2.1

Reinhard Rapp. A freely available automatically generated thesaurus of related words. In *Proceedings of LREC 2004*, pages 395–398, 2004. 3.2.2

Deepak Ravichandran and Eduard Hovy. Learning surface text patterns for a question answering system. In *Proceedings of ACL 2002*, 2002. 2.1

Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. pages 448–453, 1995. 4.1

Stefan Riezler, Alexander Vasserman, Ioannis Tso-chantaridis, Vibhu Mittal, and Yi Liu. Statistical machine translation for query expansion in answer retrieval. In *Proceedings of ACL 2007*, 2007. 1, 6.1

Ellen Riloff. Automatically generating extraction patterns from untagged text. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, pages 1044–1049, 1996. 2.1

Ellen Riloff and Rosie Jones. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of AAAI 1999*, pages 474–479, 1999. 2.3.1, 2.3.2

Lorenza Romano, Milen Kouylekov, Idan Szpektor, Ido Dagan, and Alberto Lavelli. Investigating a generic paraphrase-based approach for relation extraction. In *Proceedings of EACL 2006*, 2006. (document), 1, 2.4.1, 2.2, 2.2

Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. Intent-aware search result diversification. In *Proceedings of SIGIR 2011*, 2011. 5.2.1

Sunita Sarawagi and William W. Cohen. Semi-markov conditional random fields for information extraction. In *In Advances in Neural Information Processing Systems 17*, pages 1185–1192, 2004. 3.3

Nico Schlaefer, Petra Gieselmann, Thomas Schaaf, and Alex Waibel. A pattern learning approach to question answering within the ephyra framework. In *Proceedings of the Ninth International Conference on TEXT, SPEECH and DIALOGUE (TSD), 2006*, 2006. (document), 2.1, 5.1.3, 7.2.1

Hinrich Schütze. Distributional part-of-speech tagging. In *Proceedings of EACL 1995*, pages 141–148, 1995. 3.2.2

Hinrich Schütze. Automatic word sense discrimination. *Computational Linguistics*, 24(1), 1998. 3.2.2

Hideki Shima and Teruko Mitamura. Diversity-aware evaluation for paraphrase patterns. In *Proceedings of TextInfer 2011: The EMNLP 2011 Workshop on Textual Entailment*, 2011. 5.1.1, 5.2.3

Hideki Shima and Teruko Mitamura. Diversifiable bootstrapping for acquiring high-coverage paraphrase resource. In *LREC*, pages 2666–2673, 2012. 4.4

Yusuke Shinyama, Satoshi Sekine, and Kiyoshi Sudo. Automatic paraphrase acquisition from news articles. In *Proceedings of HLT*, 2002. 2.2.1

Jeffrey Mark Siskind. A computational study of cross-situational techniques for learning word-to-meaning mappings. *Cognition*, 61(1–2):39–91, 1996. 2.3.1

Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211. Association for Computational Linguistics, 2012.

Trevor Strohman, Donald Metzler, Howard Turtle, and W. Bruce Croft. Indri: A language-model based search engine for complex queries (extended version). IR 407, University of Massachusetts, 2005. 3.3, 3.4.1

Kiyoshi Sudo, Satoshi Sekine, and Ralph Grishman. Automatic pattern acquisition for japanese information extraction. In *Proceedings of the first international conference on Human language technology research. Morristown, NJ, USA: Association for Computational Linguistics*, 2001. 2.1

Idan Szpektor and Ido Dagan. Learning canonical forms of entailment rules. In *Proceedings of RANLP 2007*, 2007. 6.3

Idan Szpektor and Ido Dagan. Learning entailment rules for unary templates. In *Proceedings of COLING 2008*, 2008. 6.3

Idan Szpektor, Hristo Tanev, Ido Dagan, and Bonaventura Coppola. Scaling web-based acquisition of entailment relations. In *Proceedings of EMNLP 2004*, 2004. 2.3.1, 3.4.1

Idan Szpektor, Eyal Shnarch, and Ido Dagan. Instance-based evaluation of entailment rule acquisition. In *Proceedings of ACL 2007*, 2007. 6.2.2

Doina Tatar, Andreea Diana Mihis, Dana Lupsa, and Emma Tamaianu-Morita. Entailment-based linear segmentation in summarization. In *International Journal of Software Engineering and Knowledge Engineering 19(8)*, pages 1023–1038, 2009. 1

Michael Thelen and Ellen Riloff. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *Proceedings of EMNLP 2002*, 2002. 2.3.1

Jörg Tiedemann. Building a multilingual parallel subtitle corpus. In *Proceedings of CLIN 17*, 2007. 1.1, 2.2.1

Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In Walter Daelemans and Miles Osborne, editors, *Proceedings of CoNLL-2003*, pages 142–147. Edmonton, Canada, 2003. 7

Stephen Tratz and Eduard Hovy. Bewte for tac 2009's aesop task. In *Proceedings of TAC-09, Gaithersburg, Maryland*, 2009. 6.3

Peter D. Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188, 2010. 3.2.2

Reinier H. van Leuken, Lluis Garcia, Ximena Olivares, and Roelof van Zwol. Visual diversification of image search results. In *Proceedings of WWW 2009*, pages 341–350, 2009. 5.2.1

Sriharsha Veeramachaneni and Ravi Kumar Kondadadi. Surrogate learning – from feature independence to semi-supervised classification. In *Proceedings of NAACL Workshop on Semi-Supervised Learning*, 2009. 3.2.2

Anthony J Viera and Joanne M Garrett. Understanding interobserver agreement: the kappa statistic. *Family Medicine*, 37(5):360–363, 2005. 7.2.4

Ellen M. Voorhees and Hoa Trang Dang. Overview of the trec 2005 question answering track. In *Proceedings of TREC 2005*, 2005. 6.3.3

Richard C Wang and William W Cohen. Language-independent set expansion of named entities using the web. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 342–350. IEEE, 2007. 4.4

Xiaoyin Wang, David Lo, Jing Jiang, Lu Zhang, and Hong Mei. Extracting paraphrases of technical terms from noisy parallel software corpora. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 197–200, 2009. 3.4.1

Grady Ward. Moby thesaurus. 1996. 2.4

Daniel S. Weld, Raphael Hoffmann, and Wu Fei. Automatically refining the wikipedia infobox ontology. *ACM SIGMOD Record*, 37:62–68, 2009. 2.3.2

Chris Welty, James Fan, David Gondek, and Andrew Schlaikjer. Large scale relation detection.

In *Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading*, 2010. 2.3.2

Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138, 1994. 4.1

Hong Yu and Eugene Agichtein. Extracting synonymous gene and protein terms from biological literature. In *Proceedings of the 11th Int. Conf. on Intelligent Systems for Molecular Biology (ISMB-2003)*, 2003. 2.3.1

Shiqi Zhao, Ming Zhou, and Ting Liu. Learning question paraphrases for qa from encarta logs. In *Proceedings of International Joint Conferences on Artificial Intelligence (IJCAI)*, pages 1795–1800, 2007. 2.2.1

Shiqi Zhao, Cheng Niu, Ming Zhou, Ting Liu, and Sheng Li. Combining multiple resources to improve smt-based paraphrasing model. In *Proceedings of ACL-08: HLT*, pages 1021–1029, 2008a. 2.2.2

Shiqi Zhao, Haifeng Wang, Ting Liu, and Sheng Li. Pivot approach for extracting paraphrase patterns from bilingual corpora. In *Proceedings of ACL-08: HLT*, pages 780–788, 2008b. 2.2.2

Shiqi Zhao, Xiang Lan, Ting Liu, and Sheng Li. Application-driven statistical paraphrase generation. In *Proceedings of Joint conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2009)*, pages 834–842, 2009a. 2.2.2

Shiqi Zhao, Haifeng Wang, Ting Liu, and Sheng Li. Extracting paraphrase patterns from bilingual parallel corpora. *Journal of Natural Language Engineering (JNLE)*, 15:503–526, 2009b. 2.2.2

Shiqi Zhao, Haifeng Wang, and Ting Liu. Paraphrasing with search engine query logs. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*, pages 1317–1325, 2010. 2.2.1

Liang Zhou, Chin-Yew Lin, and Eduard Hovy. Re-evaluating machine translation results with paraphrase support. In *Proceedings of EMNLP 2006*, 2006a. 1, 3.4.1, 6.1

Liang Zhou, Chin-Yew Lin, Dragos Stefan Munteanu, and Eduard Hovy. Paraeval: Using paraphrases to evaluate summaries automatically. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, 2006b. 1

Ingrid Zukerman and Bhavani Raskutti. Lexical query paraphrasing for document retrieval. In *Proceedings of the 19th international conference on Computational linguistics - Volume 1*, pages 1–7, 2002. 1