Diversifiable Bootstrapping for Acquiring High-Coverage Paraphrase Resource

Hideki Shima, Teruko Mitamura

Language Technologies Institute, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213, USA
{hideki, teruko}@cs.cmu.edu

Abstract

Recognizing similar or close meaning on different surface form is a common challenge in various Natural Language Processing and Information Access applications. However, we identified multiple limitations in existing resources that can be used for solving the vocabulary mismatch problem. To this end, we will propose the *Diversifiable Bootstrapping* algorithm that can learn paraphrase patterns with a high lexical coverage. The algorithm works in a lightly-supervised iterative fashion, where instance and pattern acquisition are interleaved, each using information provided by the other. By tweaking a parameter in the algorithm, resulting patterns can be diversifiable with a specific degree one can control.

Keywords: Paraphrase, Diversification, Bootstrapping

1. Introduction

There are often many different ways of expressing the same or similar meaning in text. For example, depending on a context, one may use a different verb (e.g. "X killed Y" and "X assassinated Y"), a different phrase (e.g. "X is the killer of Y" and "Y was killed by X"), or a different syntactic structure (e.g. "X killed Y" and "Y was killed by X")[1]. This semantic variability phenomenon (Romano et al., 2006) is a common key challenge in various semantic NLP applications, as seen in paraphrase-supported works such as: automatic evaluation for Machine Translation (Zhou et al., 2006a; Kauchak and Barzilay, 2006; Pad et al., 2009), automatic evaluation for Text Summarization (Zhou et al., 2006b), Information Retrieval (Riezler et al., 2007), Information Extraction (Romano et al., 2006), collocation error correction (Dahlmeier and Ng, 2011).

Section 2 will motivate that it is especially important to study the diversity aspect of a paraphrase resource, so that one can deal with various semantic variability phenomena. In Section 3, we will define the *Diverse Paraphrase Acquisition Problem* which goal is to acquire phrase-level paraphrases that have lexical diversity. Section 4 will discuss an iterative data-driven paraphrase acquisition framework for solving the problem. In Section 5, we will propose a novel framework called *Diversifiable Bootstrapping* with which one can build a lexically diverse paraphrase resource. In the framework, the diversification level can be controlled by a parameter which effect is compared in actual paraphrases harvested in a minimal-supervision. In Section 6, we will discuss the pros and cons of the proposed approach. Finally, we will present concluding remarks in Section 7.

2. Background

2.1. Why Diversity Matters

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 1, 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.

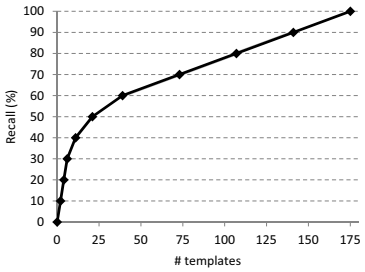


Figure 1: The number of most frequent templates necessary to reach different recall levels. We plotted this chart from the data at Table 5 in (Romano et al., 2006).

On the other hand, syntactic diversity, as exemplified in Table 1, also needs to be handled when processing semantics in text.

Note that we do not claim that diversity is the single most important factor in a paraphrase resource. Also, note that the importance of diversity may depend on applications. Diversified paraphrase resource would be especially useful in recall-oriented applications such as exhaustive and comprehensive search of patent documents (Joho et al., 2010).

[1] More examples are available in (Fujita, 2005; Androutsopoulos and Malakasiotis, 2009; Madnani and Dorr, 2010).

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

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

2.2. Limitation of Existing Paraphrase Resources

There are many existing resources that potentially address the vocabulary mismatch problem. To list some for the English language, there are WordNet (Miller, 1995), FrameNet (Baker et al., 1998), Nomlex (Macleod et al., 1998), VerbOcean (Chklovski and Pantel, 2004), VerbNet (Kipper et al., 2006); and automatically-learned entailment pattern collections such as DIRT (Lin and Pantel, 2001) and TEASE (Szpektor et al., 2004). Depending on a usage, these resources lack some important aspects described as follows.

- **Near-synonyms**. Near-synonyms are "words that have the same meaning but differ in lexical nuances" (Inkpen, 2007). For example, "terminate with extreme prejudice" and "kill" convey the same meaning with or without euphemism. Hirst (1995) points out there are other differences such as denotation, emphasis, implicature, formality, and attitude of speaker. Moreover, they claim that true synonyms are quite rare, "fully inter-substitutable", and "limited mostly to technical terms (distichous, two-ranked; groundhog, woodchuck) and groups of words that differ only in collocational properties, or the like". Given this, an ideal high-coverage paraphrase resource would support not only true synonyms but also near-synonyms.

- **Polysemy**. Words such as "end"[2], "off"[3], "hit", and "fix" mean to kill someone depending on a context. A paraphrase resource could have these words as synonym or related words, but blindly using them would result in unexpected false positives.

- **Domain specific terms**. Words such as "slot" (used by British Army)[4], "gank" (used in online games), "187" (California crime code used by police and gangs)[5], and "72" (used by extremist Muslims)[6] mean to kill someone or a killing, in a specific community or a domain. Off-the-shelf language resources are usually for a general domain. Thus there is a lack of coverage issue in technical terms.

- **Neologism**. New words are not available in relatively old dictionaries. Some examples found from an urban dictionary includes: "kevork" (from Dr. Jack Kevorkian), "OJ" (from O.J. Simpson), and "merc" (short for mercenary).

The issues above are important for processing natural language texts, considering that there is an increasing need to process untraditional colloquial texts, such as speech transcript, social media, email, and chat log. In addition, for languages where linguistic resources (e.g. dictionaries, corpora, tools) are scarce, it is ideal to automatically acquire linguistic knowledge in an unsupervised or semi-supervised fashion.

3. The Diverse Paraphrase Acquisition Problem

Motivated by the need for lexically diverse paraphrase resources in the previous section, we define a resource acquisition problem in the following way.

The Diverse Paraphrase Acquisition Problem:
Given: a target concept c.
Find: a list of string S such that a string $s \in S$ conveys the same meaning as c, and that diversity of S is maximized.

The above problem definition is designed as general as possible in order to accommodate application-specific needs. A target concept c might be represented in a way such as a disambiguated conceptual node in a lexical network (e.g. WordNet synset). Depending on a target object to acquire, the term "string" straight-forwardly applies to word ("kill"), phrase ("do away with"), slotted surface template ("X killed Y"), slotted dependency paths template with a type restriction ("$X.N : subj : V$ <kill> $V : obj : N > people > N$:nn:$Y.N$"), etc.
A criterion for "the same meaning" may also vary. For example, when learning near-synonyms introduced in Section 2.2, one may want to require s and c to have the same meaning but with a different nuance. "Diversity" as well can vary, such as diversity in lexicon, morphology, or syntactic structure.

3.1. Extended Problem Settings

In this work's configuration, we extend the problem definition to be operational in the following way.

- c is a binary-argument relation, represented as a set of concrete entity mentions that can be the arguments of the relation. For example, when the relation is about killing, c is a set of killer-victim pairs: $c = \langle$"Mark David Chapman", "John Lennon"$\rangle$, $\langle$"Sirhan Sirhan",

[2] An example usage from the movie Flight Night: "I'm going to end him, or he's going to end me".

[3] An example usage from the movie Ordinary People: "I tried to off myself".

[4] An example usage in the movie Route Irish: "You think it was him that slotted Frankie?"

[5] An example usage in the movie Hollywood Homicide: "In pursuit of possible 187 suspects".

[6] An example usage from a jihadi Zeeshan Siddiqui's diary: "2 others 72ed".

"Robert F. Kennedy"$\rangle, \ldots \rangle$. A motivation behind this representation is that it is easy for humans to prepare them, and for algorithms to process.

- s is a slotted (binary-anchored) surface template, where slot values are specific mentions of a killer and a victim.
- The criterion for "the same meaning" is the one for near-synonyms described in the previous section.
- We mean maximizing diversity by aiming S to have as many different content-words as possible.

4. Bootstrapping for Paraphrase Acquisition

In this section, we will explore the bootstrapping approach to acquiring paraphrases, and discuss the lack of lexical diversity issue in paraphrases acquired by a simple bootstrapping implementation.

4.1. Language Acquisition by Bootstrapping

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*) (Riloff and Jones, 1999; Thelen and Riloff, 2002; Yu and Agichtein, 2003; Pantel and Pennacchiotti, 2006; Komachi and Suzuki, 2008; Carlson et al., 2010; McIntosh et al., 2011).

Given the similarity of the above two bootstrapping notions, we assumed that it is natural to model a solution to the Diverse Paraphrase Acquisition Problem using a bootstrapping technique[7].

Let us consider one of well-known bootstrapping frameworks, called Espresso (Pantel and Pennacchiotti, 2006). Espresso is a lightly-supervised general-purpose algorithm for acquiring instances and patterns in an iterative fashion, which overview is illustrated in Figure 3.

The input to the algorithm is a small number (e.g. from a few to 20 or so) 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.

4.2. Lack of Diversification

Using our Espresso implementation given the killing seeds from (Schlaefer et al., 2006) and a Wikipedia corpus, we obtained a ranked list of patterns as shown in Figure 2 after the 5-th iteration.

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
$\vdots$

Figure 2: Patterns acquired by a bootstrapping method.

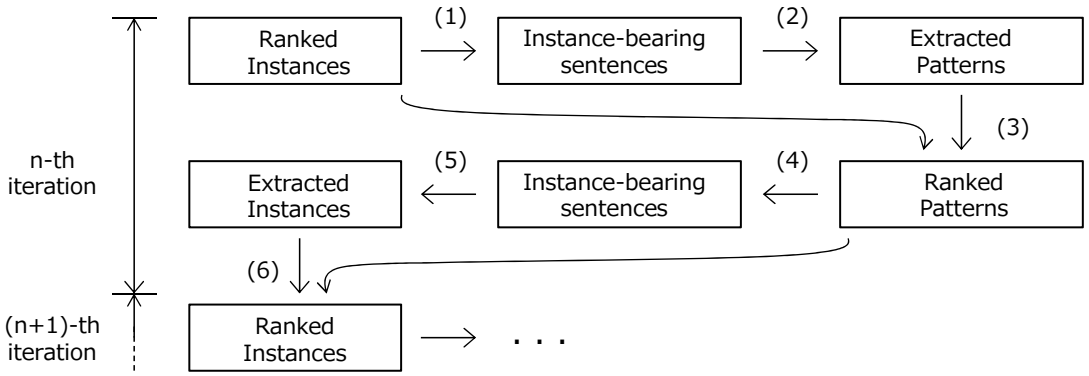
In Figure 2, we can clearly see the lack of lexical diversity. The algorithm, in this specific example, succeeded in finding syntactic and morphological variations of "assassinate". However, it is likely that there is a coverage issue when used in an application such as Relation Extraction. One possible explanation behind the lack of lexical diversity issue here 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*[8]). 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 got skewed toward a small number of narrow-sense relations. In the next section, we will discuss a solution that aims to explicitly solve this issue.

5. Diversifiable Bootstrapping

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

[7] Note that, although the bootstrapping method in Computational Linguistics may be inspired by the bootstrapping theory in Linguistics, the goal of this paper is not about mimicking the exact same way as children acquire lexicons. For instance, children start lexical acquisition "without any prior knowledge that is specific to the language being learned"(Siskind, 1996), whereas a small number of seed instances can be given in a bootstrapping algorithm.

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



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

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:

$$r'_\pi(p) = \lambda \cdot r_\pi(p) + (1 - \lambda) \cdot diversity(p) \quad (1)$$

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. Diversity function

We experimentally designed the diversity scoring function, the second term in Eq. (1), based on the D algorithm from Shima and Mitamura (2011) (see Algorithm 1[9]):
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 diversity function is given as:

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

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,

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

Algorithm 1 D score calculation

Input: patterns $p_0, \ldots, p_n$
Output: D array indexed by $1 \ldots n$
  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

5.2. 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.

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

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

(a) *killed*

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

(b) *died-of*

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

(c) *was-led-by*

Table 2: The exclusive list of seed instances for each relation.

$$r_\iota(i) = \frac{\sum_{p \in P} \frac{pmi(i,p)}{\max_{pmi}} * r_\pi(p)}{|P|} \quad (3)$$

$$r_\pi(p) = \frac{\sum_{i \in I} \frac{pmi(i,p)}{\max_{pmi}} * r_l(i)}{|I|} \quad (4)$$

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:

$$pmi(i,p) = \log \frac{|x,p,y|}{|x,*,y||*,p,*|} \quad (5)$$

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. (1) with Eq. (2, 4), we can obtain the updated pattern reliability score:

$$\begin{aligned} r'_\pi(p_k) &= \lambda \cdot r_\pi(p_k) + (1-\lambda) \cdot diversity(p_k) \\ &= \lambda \cdot \frac{\sum_{i \in I} \frac{pmi(i,p_k)}{\max_{pmi}} * r_l(i)}{|I|} \\ &\quad +(1-\lambda) \cdot \frac{D[k]}{2} \cdot r_\pi(p_0). \end{aligned} \quad (6)$$

5.3. Experiment

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 2.

The acquired patterns are shown in Table 3. 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.

6. Discussion

6.1. Reviewing the limitations of existing resources

Limitations of existing paraphrase resources presented in Section 2.2 would be addressed by the *Diversifiable Bootstrapping* method in the following reasons.

- **Near-synonyms**. The proposed method explicitly favors a paraphrase candidate that is lexically different from other selected paraphrases. Therefore, the method is cable of acquiring near-synonyms.
- **Polysemy**. Ambiguous patterns with multiple possible meanings would cause a semantic drift problem as iteration proceeds in a bootstrapping method. One can avoid mixing ambiguous and less ambiguous paraphrases if a proper semantic drift prevention method, e.g. Mutual Exclusion (McIntosh and Curran, 2008), is implemented.

$\lambda = 1.0$	$\lambda = 0.7$	$\lambda = 0.3$
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

(a) *killed*

$\lambda = 1.0$	$\lambda = 0.7$	$\lambda = 0.3$
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

(b) *died-of*

$\lambda = 1.0$	$\lambda = 0.7$	$\lambda = 0.3$
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*

Table 3: 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.

- **Domain specific terms**. Learning domain specific expressions can be realized by using a corpus from the target domain. The size of a closed domain corpus is typically smaller than an open domain one, which may cause a risk of data-sparseness. Statistical Corpus Expansion (Schlaefer, 2011) might be a potential solution that can alleviate this risk.

- **Neologism**. This too can be realized by using a specialized corpus that includes recently written documents such as the ones in social media.

6.2. 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; van Leuken et al., 2009; Rafiei et al., 2010; Santos et al., 2011). 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 paper 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.

6.3. Diversification and Semantic Drift

We have not explored the relationship between *Diversifiable Bootstrapping* and Semantic Drift; McIntosh and Curran (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 less precise patterns to be selected, therefore, there is a risk of semantic drift. It is our future work to investigate the relationship further, and to come up with a way to balance the trade-off.

6.4. Weakness

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.

7. Conclusion

In this paper, we defined the *Diverse Paraphrase Acquisition Problem* where the goal is to acquire a set of strings that have the same meaning, and at the same time, diversity among them is maximized. To this end, we proposed the lightly-supervised data-driven approach called *Diversifiable Bootstrapping*. The framework can be implemented together with a bootstrapping instance-pattern learning algorithm such as Espresso. Using the proposed approach, one can potentially learn phrase-level paraphrases that are rich in lexical diversity. The framework mitigates the risk of missing rarely occurring expressions, which shall enable one to build a paraphrase data that may include domain specific or neologism expressions.

8. Acknowledgements

This publication was made possible in part by a NPRP grant (No: 09-873-1-129) from the Qatar National Research Fund (a member of The Qatar Foundation). The statements made herein are solely the responsibility of the authors.

We also gratefully acknowledge the support of Defense Advanced Research Projects Agency (DARPA) Machine Reading Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-09-C-0172. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of the DARPA, AFRL, or the US government.

We also thank Eduard Hovy and anonymous reviewers for their helpful comments.

9. References

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

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

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

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

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

Timothy Chklovski and Patrick Pantel. 2004. Verbocean: Mining the web for fine-grained semantic verb relations. In *Proceedings of EMNLP 2004*.

K. W. Church and P. Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

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

Atsushi Fujita. 2005. *Automatic Generation of Syntactically Well-formed and Semantically Appropriate Paraphrases*. Ph.D. thesis, Nara Institute of Science and Technology, Japan.

Graeme Hirst. 1995. Near-synonymy and the structure of lexical knowledge. In *In AAAI Symposium on Representation and Acquisition of Lexical Knowledge: Polysemy, Ambiguity, and Generativity*, pages 51–56.

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

Hideo Joho, Leif Azzopardi, and Wim Vanderbauwhede. 2010. 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.

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

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

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

Dekang Lin and Patrick Pantel. 2001. Dirt - discovery of inference rules from text. In *Proceedings of KDD 2001*, pages 323–328.

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

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

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

Tara McIntosh and James R. Curran. 2009. Reducing semantic drift with bagging and distributional similarity. In *Proceedings of ACL-IJCNLP 2009*.

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

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

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

Patrick Pantel and Marco Pennacchiotti. 2006. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of COLING-ACL 2006*.

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

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

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

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

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

Nico Schlaefer, Petra Gieselmann, Thomas Schaaf, and Alex Waibel. 2006. 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*.

Nico Schlaefer. 2011. *Statistical Source Expansion for Question Answering*. Ph.D. thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA.

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

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

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

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

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

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

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

Liang Zhou, Chin-Yew Lin, Dragos Stefan Munteanu, and Eduard Hovy. 2006b. Paraeval: Using paraphrases to evaluate summaries automatically. In *Proceedings of HLT-NAACL 2006*.