Extracting Synonyms from Dictionary Definitions

by

Tong Wang

A reserach paper submitted in conformity with
the requirements for the degree of Master of Science
Department of Computer Science
University of Toronto

Extracting Synonyms from Dictionary Definitions

Tong Wang

Department of Computer Science, University of Toronto

Toronto, ON, M5S 3G4, Canada

Abstract

Automatic extraction of synonyms and/or semantically related words has various applications in Natural Language Processing (NLP). There are currently two mainstream extraction paradigms, namely, lexicon-based and distributional approaches. The former usually suffers from low coverage, while the latter is only able to capture general relatedness rather than strict synonymy.

In this paper, two rule-based extraction methods are applied to definitions from a machine-readable dictionary. Extracted synonyms are evaluated in two experiments by solving TOEFL synonym questions and being compared against existing thesauri. The proposed approaches have achieved satisfactory results in both evaluations, comparable to published studies or even the state of the art.

1 Introduction

1.1 Synonymy as a Lexical Semantic Relation

Lexical semantic relations (LSRs) are the relations between meanings of words, e.g. synonymy, antonymy, hyperonymy, meronymy, etc. Understanding these relations is not only important for word-level semantics, but also has found applications in improving language models (Dagan et al., 1999), event matching (Bikel and Castelli, 2008), query expansion, and many other NLP-related tasks.

Synonymy is the LSR of particular interest to this paper. By definition, a synonym is "one of two or more words or expressions of the same language that have the same or nearly the same meaning in some or all senses" (Merriam-Webster, 2003). One of the major differences between synonymy and other LSRs lies in its emphasis on the more strict sense of similarity in contrast to the more loosely-defined relatedness; being synonymous generally implies semantic relatedness, while the opposite is not necessarily true. This fact, unfortunately, has been overlooked by several synonymy-oriented studies; although their assumption that "synonymous words tend to have similar contexts" (Wu and Zhou, 2003) is valid, to take any words with similar contexts as synonyms is quite problematic. In fact, words with similar contexts can represent many LSRs other than synonymy, even including antonymy (Mohammad et al., 2008).

Despite the seemingly intuitive nature of synonymy, it is one of the most

difficult LSRs to identify from free texts, since synonymous relations are established more often by semantics than by syntax. Hearst (1992) extracted hyponyms based on the syntactic pattern "A, such as B". From the phrase "The bow lute, such as the Bambara ndang, is plucked and ...", there is clear indication that "Bambara ndang" is a type of "bow lute". Given this successful example, it is quite tempting to formulate a synonym extraction strategy by a similar pattern, i.e., "A, such as B and C", and to take B as a synonym to C. Unfortunately, without semantic knowledge, such a theory is quite fragile, since the relationship between B and C greatly depends on the semantic specificity of A, i.e., the more specific A is in meaning, the likely B and C are synonyms. This point is better illustrated by the following excerpt from the British National Corpus, in which the above-proposed heuristic would establish a rather counter-intuitive synonymy relationship between *oil* and *fur*:

> ... an agreement allowing the republic to keep half of its foreign currency-earning production such as *oil* and *furs*.

Another challenge for automatic processing of synonymy is evaluation. Many evaluation schemes have been proposed, including human judgement and comparing against existing thesauri, among other task-driven approaches; each exhibits problems in one way or another. The details pertaining to evaluation are left to Section 3.

1.2 Automatic Extraction of LSRs

1.2.1 Synonym Extraction

There are currently two major paradigms in synonym extraction, namely, distributional and lexicon-based approaches. The former usually assesses the degree of synonymy between words according to their co-occurrence patterns within text corpora, under the assumption that similar words tend to appear in similar contexts. The definition of context can vary greatly, from simple word token co-occurrence within a fixed window to position-sensitive models such as n-gram models to even more complicated situations where the syntactic/thematic relations between co-occurring words are taken into account.

One successful example of the distributional approach is that of Lin (1998). The basic idea is that, two words sharing more syntactic relations with respect to other words are more similar in meaning. Syntactic relations between word pairs were captured by the notion of *dependency triples* (e.g., (w_1, r, w_2), where w_1 and w_2 are two words and r is their syntactic relation). Semantic similarity measures were established by first measuring the amount of information $I(w_1, r, w_2)$ contained in a given triple through *mutual information*; such measure could then used in different ways to construct similarity between words, e.g., by the following similarity measure:

$$sim(w_1, w_2) = \frac{\Sigma_{(r,w)\in T(w_1)T(w_2)} I(w_1, r, w) + I(w_2, r, w)}{\Sigma_{(r,w)\in T(w_1)} I(w_1, r, w) + \Sigma_{(r,w)\in T(w_2)} I(w_2, r, w)}$$

where $T(w)$ denotes the set of relation-word pairs (r, w') that ensures $I(w, r, w') > 0$. The resulting similarity was then compared to lexicon-based similarities built on two existing thesauri and were shown to be closer to WordNet than to *Roget's Thesaurus*. Note that the first step of measuring the relatedness of a given word w and its contexts (in this case, another word of a specific syntactic relation r to w) is known as *association ratio* (Mohammad and Hirst, 2005).

Several later variants followed the work of Lin (1998). Hagiwara (2008), for example, also used the concept of dependency triples and extended it to *syntactic paths* in order to account for less direct syntactic dependencies. When building similarity measures, the *pointwise total correlation* were used as the association ratio as opposed to the *pointwise mutual information* (or *PMI*) used by Lin (1998). Wu and Zhou (2003) used yet another measure of association ratio, i.e., *weighted mutual information* (or WMI) on the same distributional approach, claiming that WMI could correct the biased (lower) estimation of low-frequency word pairs in PMI. In addition, Wu and Zhou (2003) also used a bilingual corpus in synonym extraction, the intuition behind which is that "two words are synonymous if their translations are similar". This was modelled by the notion of *translation probability* in computing similarity scores.

Multilingual approaches can also be found in later studies, e.g., by Van der Plas and Tiedemann (2006), hypothesizing that "words that share translational contexts are semantically related"; the details of their approaches,

however, differ in several important ways, such as the resource for computing translation probabilities (corpus versus dictionary) and the number of languages involved in the multilingual settings (eleven versus two), etc. Resulting synonym sets are compared against an existing thesaurus (the *Euro WordNet*), an approach similar to that of Wu and Zhou (2003). Since both the corpora and the gold standards are different in these two studies, the results bear no comparable meanings other than the figures themselves.

Another example of distributional approaches is that of Freitag et al. (2005), where the notion of context is simply word tokens appearing within windows. Several probabilistic divergence scores were used to build similarity measures and the results were evaluated by solving simulated TOEFL synonym questions, the automatic generation of which is itself another contribution of the study.

In contrast to distributional measures, there are many studies that use lexica, especially dictionaries, for synonym extraction. Particularly in recent years, one popular paradigm is to build a graph on a dictionary according to the defining relationship between words: vertices corresponding to words, and edges pointing from the words being defined to words defining them.

Given such a *dictionary graph*, many results from graph theory can then be employed to explore synonym extraction. Blondel and Senellart (2002) applied an algorithm on a weighted graph (similar to PageRank Page, Brin, Motwani, and Winograd 1998); weights on the graph vertices would converge to numbers indicating the relatedness between two vertices (words), which

are subsequently used to define synonymy.

Muller et al. (2006) built a Markovian matrix on a dictionary graph to model random walks between vertices, which is capable of capturing the semantic relations between words that are not immediate neighbors in the graph. Ho and Cédrick (2004) employed concepts in information theory, computing similarity between words by their *quantity of information exchanged* (QIE) through the graph.

1.2.2 Mining Dictionary Definitions

Back in the early 1980s, extracting and processing information from machine-readable dictionary (MRD) definitions was a topic of considerable popularity, especially since the Longman's Dictionary of Contemporary English (or LDOCE, Procter et al., 1978) had become electronically available. Two special features have been particularly helpful in promoting the dictionary's importance in many lexicon-based NLP studies. Firstly, the dictionary uses a *controlled vocabulary* (CV) of only 2,178 words to define approximately 207,000 lexical entries. Although the lexicographers' original intention was to facilitate the use of the dictionary by learners of the language, this design was later proved to be a valuable computational feature. The *subject code* and *box code*, on the other hand, tag each lexical entry with additional semantic information, the former specifying a thesaurus-category-style classification of the domains of usage, and the latter representing selectional preferences/restrictions. Figure 1 gives an example of such codes for the

```
...
28290107<0100<T1;X0<NAZV< H XS
...
```

Figure 1: One line in the LDOCE record for the word *rivet* containing subject/box codes. In this example, `NAZV` is the subject code indicating "nautical" subject area, and the box code `_H_XS` indicates preferences (selectional restrictions) for human subject and solid object.

word *rivet* (Boguraev and Briscoe, 1989).

Note that it is debatable whether a learner's dictionary is indeed more suitable for the purpose of machine-based learning for NLP. A controlled vocabulary can also complicate the definition syntax, since there is usually a trade-off between the size of the defining vocabulary and the syntactic complexity of definitions (Barnbrook, 2002). Nonetheless, with all the computationally friendly features, LDOCE soon attracted significant research interests. Boguraev and Briscoe (1989) covered various topics in using this MRD, from rendering easier on-line access and browsing (which involved many engineering challenges under the then rather primitive computing environment) to semantic analysis and utilization of the definition texts. The latter is of great relevance to the topics discussed in this paper.

Alshawi (1987) (included in Boguraev and Briscoe 1989) conducted a phrasal analysis of LDOCE definitions by applying a set of successively more specific phrasal patterns on the definition texts. The goal was to mine semantic information from definitions, which is believed to be helpful in "learning" new words with the knowledge of the controlled vocabulary in LDOCE.

Guthrie et al. (1991) exploited both the controlled vocabulary and the *subject code* features. The controlled vocabulary were firstly grouped into "neighborhoods" according to their co-occurrence patterns; the subject codes were then imposed on the grouping, resulting in so-called *subject-dependent neighborhoods*. Such co-occurrence models were claimed to better resemble the polysemous nature of many English words, which in turn could help improve word sense disambiguation (WSD) performance. Unfortunately, no evaluation has ever been published to support this claim.

The work of Chodorow et al. (1985) is an example of building a semantic hierarchy by identifying "head words" (or *genus terms*; see Section 1.2.3) within definition texts. The basic idea is that these head words are usually hypernyms of the words they define. If two words share the same head word in their definitions, they are likely to be synonymous siblings under the same parent in the lexical taxonomy. Thus, by grouping together words that share the same hypernyms, not only are synonyms extracted from the definition texts, but they are also, at the same time, organized into a semantic hierarchy.

1.2.3 Properties of Dictionary Definitions

In lexicographical terminologies, the word being defined in a dictionary is called *definiendum*, and words that defining it are called *definientia*. This subsection discusses special features of these elements in monolingual English dictionarys that facilitate synonym extraction.

Within the definientia, there is usually a special component that is more

closely related to the definiendum than the rest of the definition: the *genus term*. Usually, genus terms are either synonyms or hypernyms of the definienda, as in the example of ***automobile***`: a motor` ***car*** and ***summer***`: the second and warmest` ***season*** `of the year`, where `car` is synonymous to `automobile` while `season` a hypernym of `summer`.

There are exceptions, however, that forbid the extraction of synonyms or hypernyms through simple heuristics such as taking the first word of the same POS as the definiendum, which sometimes is either a quantifier or a word of very general meaning. This problem is known as *an empty head*. One example is the definition ***arum***`: a tall, white` ***type*** `of lily` (Guthrie et al., 1990). Identifying the genus terms or even the empty heads is itself a useful application in processing dictionary definitions.

Meanwhile, the composition of definientia usually exhibits great regularity in terms of syntax, style, and sometimes, vocabulary. Amsler (1980) showed that definitions of nouns and verbs in most dictionaries follow rigid stylistic patterns. In fact, this stylistic regularity in definitions goes beyond that of nouns and verbs; as is shown in Section 3.2, definitions of adjectives exhibit comparable or even greater regularities than those of nouns.

In conclusion, these characteristics of dictionary definitions altogether form the theoretical basis for this paper from the lexicographical point of view: the semantic relatedness between definienda and definientia validates the hypothesis that synonymy does exist in dictionary definitions, while the regularities in the definientia make it possible to develop algorithms for syn-

onym extraction.

2 Synonym Extraction from Dictionary Definitions

2.1 Data Preparation

The MRD used in this project is *The Macquarie Dictionary* (Delbridge et al., 1981). It comes as a single SGML-tagged file of 63.0MB with 78 types of tags. Information about each lexical entry, including pronunciation, part(s) of speech, definitions, related phrases, etc., is represented by a tree structure of tags, rooted at a tag named `RECORD`. There are altogether 106,964 such entries in the MRD. Figure 2 shows an example of the tree of SGML tags for the entry word *dictionary*.

To facilitate the processing of the definition texts, an API to the SGML file was written in Java using the SAX structure for XML parsing. For better efficiency, the 60M file of the dictionary was chunked into 26 files, each containing words beginning with one of the 26 letters in the English alphabet. The chunking also provides a basis for the multi-threaded architecture in the final implementation.

In order for the SAX XML parser to work, the raw data from the original file have to be corrected from a large number of errors and inconsistencies. For example, some of the values for in-tag properties were enclosed by quo-

```
<RECORD id="000020291">
  <HEAD>[dictionary]
  <SORTKEY>[DICTIONARY0990010000]
  <FLAGS>[BIGM N]
  <PRON>
    <PRN>['d1k47nri]
    <PRN TYPE="SAY">['dikshuhnree]
    <PRN>['d1k47n7ri]
    <PRN TYPE="SAY">['dikshuhnuhree]
  <BODY>
    <CHUNK>
      <POS>[noun]
      <INFLECTION>
        <INF NUMBER="PL">[dictionaries]
      <DEF id="322">
        <DTEXT>[a book containing a selection of the words of a
        language, usually arranged alphabetically, with explanations
        of their meanings, pronunciations, etymologies, and other
        information concerning them, expressed either in the same or
        in another language; lexicon; glossary.]
        <THES>[599.04.10]
      <DEF id="157">
        <DTEXT>[a book giving information on particular subjects or a
        particular class of words, names or facts, usually under
        alphabetically arranged headings]
        <IP>[a biographical dictionary.]
  <ETY>[, lit., a word-book, fromword. See]
    <LANG>[Medieval Latin]
    <I>[dicti[omacr ]n[amacr ]rium]
    <LANG>[Late Latin]
    <I>[dictio]
    <LINK>[diction]
  <STERM POS="N" LEMMA="HWD" TYPE="IINF" NUMBER="PL">[dictionaries]
```

Figure 2: The tree of SGML tags of the word *dictionary* in *The Macquarie Dictionary*

tation marks, i.e., in the form of `<INF tense="PAST">`, while others were not. By default, the SAX XML parser requires the quotation marks, and there are 127,927 instances that require adding quotation marks.

There are also errors that do not follow any easy-to-capture pattern and thus have been corrected manually. These corrections include tag pairs such as <subDEF> ... < /SUBDEF> (inconsistent capitalization), etc. Altogether 142,771 instances of corrections have been made.

2.2 Inverted Index Extraction (IIE)

As stated earlier, of the words in definientia, a considerable proportion are semantically related to its definiendum, though the degree of relatedness may vary. In terms of synonym extraction, an ideal case would be to first identify all important terms from the definientia, and then extract synonyms within the reduced search space. Which terms are more important than others, however, is itself quite a difficult question to answer.

In this section, a simpler extraction strategy is adopted as a first attempt to explore the relationship between definienda and their definientia. The basic idea is to build an *index of concordance* (a.k.a. *inverted index* in Information Retrieval and hence the name *Inverted Index Extraction*) on the dictionary. Each line l in the inverted index consists of a *target word* t (i.e., the synonyms of which are the targets of extraction), followed by a set S of

words that contain t in their definition texts, i.e.,

$$l = (t, S)$$

where

$$S = \{w : t \in dfn(w)\}$$

Here, $dfn(w)$ refers to the collection of defininetia of the word w in a dictionary. In IIE, such words are considered semantically related to t, regardless of the importance of t within their definitions. This idea was first developed by Reichert et al. (1969) and has ever since been extensively exploited by later studies as a basis for building dictionary graphs (see Section 1.2.1).

A complete list of words and phrases extracted by IIE for *look* is listed in Table 1. Many near synonyms of the target word *look* are successfully extracted, including words such as *see* and *watch*, which are synonymous to *look* in the most general sense, as well as *ogle*, *gaze*, *inspect*, *glance*, etc., that denote the action of *looking* with connotations of doing it in particular ways. Note that existing thesauri (e.g., *Roget's Thesaurus*, Sidney and Ronald 1990) do not list the word *gawp* or *rubberneck* as a synonym of *look*; these two words, among many others extracted by IIE, are in fact as synonymous to *look* as *scrutinize* and *glance* are. This means the results of IIE could indeed help improve the coverage of existing thesauri.

More interestingly, colloquial expressions denoting the action of *looking* also appear in the list, such as *dekko* and *Captain Cook*, which is not surpris-

(the) devil take the hindmost	evil	green	peep
air	expect	have	peer
an optic at	explore	have a perv	perv
appear	expression	have a screw	phenotype
appearance	eye	have a sticky	pout
aspect	eyeball	have eyes only for	pry
at	eyehole	health visitor	quiz
await	eyesore	hold	regard
babysit	eyewink	hook and eye	retrospect
bad hair day	face	hope	review
beam	face as long as a fiddle	horror	rubberneck
behold	faraway	hunt up	scowl
bend one's gaze on (or upon)	fascinate	independent	scrutiny
blink	fleer	inspect	search
blink at	flight control	introspect	see
butchers	front	keep house	shoofty
candid camera	frown	la femme	show
Captain Cook	gander	lamp	skew
care for	gawp	leer	smile
check	gaze	lemma	snapdragon
cherchez	geek	letter bomb	sneer
command	get	light-pen	snorkel
contemplate	get a	lo	speck
cook	get an eyeful of	load of	squint
cop	get on to	look	squiz
countenance	gink	look daggers at	stare
crane	give	lour	sticky
dekko	glance	mind	stony
despise	glare	nurse	tend
disdain	gleam in one's eye	nut	treat
district nurse	glimpse at	ogle	twig
double take	gloom	optimism	view
easy on the eye	glower	Orpheus	watch
envisage	goggle	overlook	withering
	Goth	oversee	
		peek	

Table 1: IIE result for the target word *look*.

ing given the Australian provenience of *The Macquarie Dictionary*. These are very good evidence for the claim made in previous studies that synonyms extracted from dictionaries would reflect some of the domain- or language-specific features of the dictionary of choice.

Semantic relatedness in IIE is built upon the occurrence of one word in the definition of another. This gives unique inter-connected structures among the proposed synonyms in the resulting sets, which is the dictionary graph mentioned earlier. Figure 3 is a visualization of the sub-graph for the word *look*. Since all the extracted words are related to the target word, the word itself and all its relatedness edges are omitted with no loss of information. As a result, words that are only connected to the target word would appear as isolated nodes, and thus are also omitted for simplicity of the diagram.

It is also worth noting that, the definitions of the vertices (words) in Figure 3 contain not only the target word *look*, but also some other vertices in the same sub-graph. A closer look at these sub-graphs can usually reveal surprisingly similar connotations of the words within, as in the case of {*glance*, *peep*, *peek*, *pry*, *peer*}, or {*treat*, *disdain*, *despise*}, etc.

2.2.1 False Positives and Target Word Frequency

Since all words within the definition texts are treated indiscriminatively, the relatedness defined by IIE is more loose than that of Chodorow et al. (1985). This is especially problematic for common English words, which are more likely to appear in the definition texts of many other words and thus, would

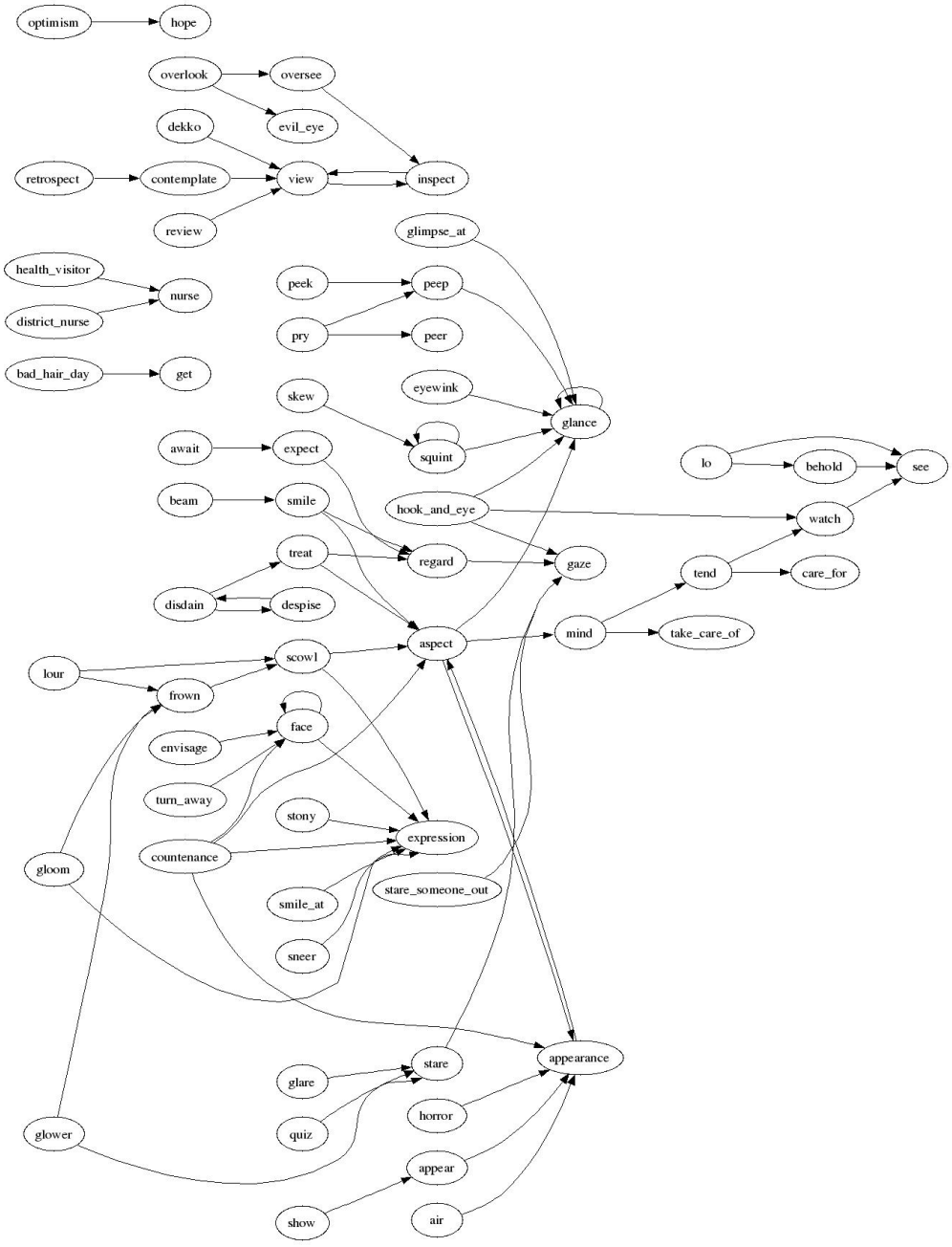


Figure 3: A visualization of the dictionary graph for the word *look*. The edge direction refers to the *being defined by* relationsship, i.e., N_1 points to N_2 if N_2 appears in the definientia of N_1.

usually introduce more false positives into the results. The word *independent*, for example, is related to *look* by its definition "`sufficient to support someone so that they do not have to look for a living`", or *optimism* by "`tendency to look on the bright side of things`".

On the other hand, uncommon words have a much lower probability of appearing in other words' definition texts, and thus, the cluster size of such words is severely diminished or sometimes even empty. Further discussions of and solutions to these problems will follow in later sections.

2.2.2 Morphologies in Definition Texts

Another question to consider about IIE is whether it is necessary to match inflections, plural forms and other non-base forms of the target word in other words' definition texts. One hypothesis is that the semantically important definition components, e.g., genus terms, are more likely to appear in their base forms in definition texts. In fact, an examination of a set of 100 randomly selected definition texts in *The Macquarie Dictionary* reveals that 85% of the semantically significant words (e.g. genus terms, discriminators (Barnbrook, 2002), etc.) appear in their base forms. For the verb *look*, words extracted by IIE using its base form and non-base forms are listed in Table 2.

When the various non-base forms of the verb *look* are used as target words, the extracted words seem to be less similar in meaning. Evaluation results in Section 3.1 show that, when solving TOEFL synonym questions containing

(a) *looks*

looker	pessimist	sago snow	unhandsome
looker-on	pretty boy	scribbler	wallflower
outlook	prier	sight	zoetrope
pakapoo	retro-	spectator	
-ticket	rough as bags	stargazer	

(b) *looked*

Avernus	hunt the slipper	son	vortex
Eurydice	mother	unhoped-for	winding strip
-expectation	philistine	unlooked-for	

(c) *looking*

admiration	forbidding	monochromator	scopophilia
Apollo	foresight	on the make	taupata
bathing beauty	haggard	onlooking	voyeur
beat	here we (or you)	outlandish	voyeurism
birdwatcher	are	outlook	wallaby track
descry	homely	passant	washed-out
easy on the eyes	househusband	peep	watchout
feral	mandrill	reflective glass	well-favoured
-fine	mirror	regardant	wispy

Table 2: Word lists for various non-base forms of *look*

inflected target words, the coverage of IIE increases by a considerable margin (25.0 percentage points) when the target words are lemmatized.

2.3 IIE Filtering

As mentioned in Section 2.2.1, in terms of target word frequency in English, IIE exhibits problems on both ends: common target words introduce many false positives, while uncommon ones suffer from diminished size of the resulting synonym set. This section will focus on solving the first problem by

developing two mechanisms to eliminate false positives; the second problem is to be addressed in Section 2.4 by developing a different extraction strategy.

2.3.1 POS Constraining

In IIE, words that contain the target word w in their definitions are not necessarily of the same part of speech (POS) as that of w; meanwhile, if a particular POS is assumed for w, there is no easy way to tell whether an instance of w in another word's definition is of the same POS unless the definition texts are POS-tagged. A very undesired case would be, for example, when we want to investigate the synonyms of the verb *look* but come across with an adjective *faraway*, which has *look* as a noun in its definition (*faraway* : `abstracted or dreamy, as a` *look*). This is one of the causes for introducing false positives into the extracted results. One simple thing to do (without POS-tagging the definition texts) here is to specify a POS p for the target word w, and then limit the POS of the extracted words to p. The result is shown in Table 3.

The above result is generally appealing to intuition: words in the first table tend to be more synonymous to the verb *look* than those in the second one. However, for some of the verbs in the first table, e.g., the word *green*: "`(verb i): to transform the` *look* `of (a locale, especially an urban area) by planting a large number of trees`", the target word *look* actually appears as a noun instead of the assumed POS of verb. To properly deal with this problem, it is necessary to at least have the definition texts

(a) *Verbal forms*

appear (verb (copula))	disdain (v.t.)	glower (v.i.)	peep (v.i.)	show (v.i.)
await (v.t.)	envisage (v.t.)	green (v.t.)	peer (v.i.)	skew (v.i.)
babysit (v.t.)	expect (v.t.)	hope (v.t.)	perv (v.i.)	snorkel (v.i.)
beam (v.i.)	explore (v.t.)	inspect (v.t.)	pout (v.i.)	speck (v.t.)
behold (v.t.)	eyeball (v.t.)	introspect (v.t.)	pry (v.i.)	squint (v.i.)
blink (v.i.)	face (v.t.)	lamp (v.t.)	quiz (v.t.)	tend (v.t.)
check (v.t.)	fascinate (v.t.)	leer (v.i.)	regard (v.t.)	treat (v.t.)
command (v.i.)	frown (v.i.)	lour (v.i.)	retrospect (v.i.)	twig (v.t.)
contemplate (v.t.)	gawp (v.i.)	mind (v.t.)	review (v.t.)	view (v.t.)
cop (v.t.)	gaze (v.i.)	nurse (v.t.)	rubberneck (v.i.)	watch (v.i.)
crane (v.i.)	get (v.t.)	nut (v.i.)	scowl (v.i.)	window-shop (v.i.)
-despise (v.t.)	glance (v.i.)	ogle (v.t.)	search (v.t.)	
	glare (v.i.)	overlook (v.t.)	see (v.t.)	
	gloom (v.i.)	oversee (v.t.)		

(b) *Non-verbal forms*

(the) devil take the hindmost (phrase)	eyewink (noun)	have a perv (phrase)	regard (noun)
air (noun)	face (noun)	have a screw at (phrase)	scowl (noun)
appearance (noun)	faraway (adjective)	have a sticky (phrase)	scrub up well (phrase)
aspect (noun)	fleer (noun)	have an optic at (phrase)	scrutiny (noun)
bad hair day (noun)	flight control (noun)	have eyes only for (phrase)	shoofty (noun)
behold (interjection)	front (noun)	health visitor (noun)	smile at (phrase)
bend one's gaze on (or upon) (phrase)	frown (noun)	hold the fort (phrase)	smile on (or upon) (phrase)
blink at (phrase)	gander (noun)	hook and eye (noun)	smile (noun)
butchers (noun)	gaze (noun)	horror (noun)	snapdragon (noun)
candid camera (adjective)	geek (noun)	hunt up (phrase)	sneer (noun)
Captain Cook (noun)	get a load of (phrase)	independent (adjective)	squiz (noun)
care for (phrase)	get an eyeful of (phrase)	keep house (phrase)	stare someone out (phrase)
cherchez la femme (null)	get on to (or onto) (phrase)	lemma (noun)	stare (noun)
cook (noun)	gink (noun)	letter bomb (noun)	sticky (noun)
countenance (noun)	give someone the glad eye (phrase)	light-pen (noun)	stony (adjective)
dekko (noun)	glance (noun)	lo (interjection)	take care of (phrase)
district nurse (noun)	glare (noun)	mother's help (noun)	the glad eye (phrase)
double take (noun)	gleam in one's eye (phrase)	optimism (noun)	the greasy eyeball (phrase)
easy on the eye (phrase)	glimpse at (phrase)	Orpheus (noun)	turn away (phrase)
evil eye (noun)	gloom (noun)	peek (noun)	victim toy (noun)
expression (noun)	glower (noun)	peep (noun)	watch for (phrase)
eyehole (noun)	goggle (noun)	phenotype (noun)	watch out for (phrase)
-eyesore (noun)	Goth (noun)		withering (adjective)
	have a face as long as a fiddle (phrase)		

Table 3: POS-Constrained IIE for the target word *look*

POS-tagged, which is beyond the scope of discussion in this paper. Also, in the definition of *snorkel*: "`(verb i): to swim using such a device, in order to` *`look`* `at the seabed, fish, etc.`", the target word appears in a subordinate clause in the latter part of the definition, where words usually have less semantic contribution to the definition texts. This, again, requires further processing (e.g., parsing) of the definition texts.

Note that some nouns have verbal counterparts, such as *gaze* and *glance*, denoting the action of *looking*. Although they are filtered out for having different POS, their verbal counterparts have remained in the final result. Phrases remain problematic in IIE, since they do not have POS tags in *The Macquarie Dictionary*. In the above example, some phrasal verbs, such as *take care of* and *glimpse at*, become false negatives.

2.3.2 Filtering by Low Connectivity

Recall from Section 2.2.1 that high-frequency target words can introduce many false positives by appearing more frequently in other words' definitions. If the target word is relatively general in meaning, then many words in the IIE result could be hyponyms instead of synonyms of the target word. Table 4 shows the example of *fear* and various *-phobia*s proposed by IIE.

One of the most important aspects of synonym extraction, as specified in Section 1.1, is to single out strict synonymy as an LSR from other semantic relations between words. In this sense, the *-phobia*s in the above example are hyponyms instead of synonyms and thus should be eliminated from the result.

aerophobia	coprophobia	lyssophobia	shudder
affright	courage	necrophobia	shy
agoraphobia	Demogorgon	Negrophobia	squeal
alarm	doubt	nightmare	superstition
Anglophobia	dread	nosophobia	technophobia
angst	emotion	nyctophobia	terror
apprehension	ergophobia	ochlophobia	terrorism
aquaphobia	erythrophobia	pallor	thanatophobia
arachnophobia	foetal position	panic	thing
attrition	fright	parliamentary	toxiphobia
awe	funk	privilege	tremor
Bayard	gasp	passion	triskaidekaphobia
biopanic	gynophobia	perfect contrition	wheyface
blue funk	hair-raiser	persecution complex	xenophobia
bugaboo	hobgoblin	phobia	zoophobia
bugbear	homophobia	psychasthenia	
cancerophobia	horripilation	scare	
castration complex	horror	shock-horror	
	jealousy		

Table 4: IIE result for the target word *fear*

Since dictionary definitions usually use synonyms or hypernyms, instead of hyponyms, to define a word, from the perspective of the dictionary graph, these hyponyms usually do not connect to one another as much. Thus, given the original set of proposed synonyms, the basic idea here is to keep those with definition texts that contain other words in the same set, and filter out words that are "isolated" in terms of connectivity. Table 5 lists the result of applying this filter.

As can be seen from the result, various *-phobia*s are indeed excluded from the result. More interestingly, the word *phobia* itself, being a synonym

(a) Densely-connected words

affright	bugbear	funk	shock-horror
alarm	doubt	hair-raiser	shudder
angst	Demogorgon	horror	terror
apprehension	dread	nightmare	thing
awe	emotion	passion	
-bugaboo	fright	phobia	

(b) Words with low connectivity

aerophobia	courage	nosophobia	squeal
agoraphobia	ergophobia	nyctophobia	superstition
Anglophobia	erythrophobia	ochlophobia	technophobia
aquaphobia	foetal position	pallor	terrorism
arachnophobia	gasp	panic	thanatophobia
attrition	gynophobia	parliamentary	toxiphobia
Bayard	hobgoblin	privilege	tremor
biopanic	homophobia	perfect contrition	triskaidekaphobia
blue funk	horripilation	persecution complex	wheyface
cancerophobia	jealousy	plex	xenophobia
castration complex	lyssophobia	psychasthenia	zoophobia
-coprophobia	necrophobia	scare	
	Negrophobia	shy	

Table 5: Discriminating words by connectivity

instead of a hyponym of *fear*, successfully remains in the first list, due to its connection to *dread* in one of its definitions.

This process could also be viewed as a second round of IIE with a search space reduced to the synonyms extracted from the first round. If IIE is based on the hypothesis that synonyms usually appear in definition texts of one another, then this filtering process is the contrapositive of the same claim, i.e., a word is less likely to be a member of a set of synonyms if it does not appear in other words' definition.

```
1. administered
   a.  managed
   b.  recognized
   c.  opposed
   d.  justified
```

Figure 4: Inflections in TOEFL synonym questions

2.3.3 A Further Note on Filtering

Despite the intuitively appealing performance of the two filtering mechanisms discussed in this subsection, applying filters on the sets of proposed synonyms is often a less desirable practice, since dictionary-based extractions suffer more often from poor coverage than accuracy. Sometimes the degree of synonymy between two words is determined by intersecting their synonym sets; due to diminished sizes, these sets often do not overlap regardless of the synonymous nature of the words. In these cases, shrinking the already small sets would only worsen the situation.

As for POS constraint filtering, the POS of the target words are not always available. Also, as in the evaluation task of solving TOEFL synonym questions, target words are sometimes inflected, making their POS even more ambiguous (see the example in Figure 4). There has not been a satisfactory solution to this problem at the time of this writing.

2.4 Pattern-based Extraction (PbE)

As stated earlier, the number of synonyms extracted by IIE largely depends on the frequency of the target word and is severely diminished if the target word is rare and thus, less likely to appear in other words' definitions. Consequently, a new extraction strategy is proposed in this section in an attempt to alleviate this problem.

Specifically, for a given word w, instead of following the reversed index of the dictionary to get other definienda as synonyms, the proposed approach extracts synonyms directly from the definition text of w according to certain patterns (hence the name Pattern-based Extraction or PbE). As a result, the frequency of w no longer matters, as long as its definition matches any of the extraction patterns. As is true with all existing lexicon-based methods, sparsity remains an issue for PbE, but evaluation on both tasks in Section 3 shows significant improvement on PbE's coverage over that of IIE.

2.4.1 Interpretive Parts and Synonymous Parts in Definitions

Before going into the details of PbE, it is necessary to examine the composition of definition texts more closely. From the several mono-lingual English dictionaries investigated in this study, definition texts can often be decomposed into two parts: the *interpretive part* and the *synonymous part*. The former is usually at the beginning of a definition as a relatively lengthy interpretation of the definiendum, using relatively simple vocabulary; many of these are then followed by one or more synonymous parts, each consisting

of a single word or phrase which is highly synonymous to the definiendum. These two parts, when appearing together within the same piece definition text, are usually separated by special typographical styles or delimiters, such as capitalization or semicolons.

Consider the example of the definition of *look* in Merriam-Webster (2003) "`to exercise the power of vision upon: EXAMINE`". Here, the beginning part of the definientia (before the colon) is the interpretive part, following which, capitalized and separated by the colon, is the synonymous part consisting of a single word *examine*, which is a synonym of the definiendum *look* under this sense. Another example is the definition of *looker-on* in Merriam-Webster (2003) (***`looker-on`*** `:` *one who looks on; a spectator*), where the synonymous part after the semicolon is, instead of one word, prefixed by the indefinite article for grammatical correctness. In terms of length and vocabulary, both examples conform with the previous observation on the differences between the interpretive and the synonymous parts.

For the interpretive part of a given definition, the synonym or synonyms, if any, could be identified only through syntactic and semantic knowledge of the definientia. Many attempts have been made to automate such deep analysis of the defining language (see Section 1.2.2). It is the simpler cases of the synonymous parts in definitions, however, that are mostly neglected by various studies on synonym extraction. Semantically, as is shown in the previous examples, such parts of definitions are indeed highly synonymous to the definienda, whereas terms extracted from the interpretive parts are usually

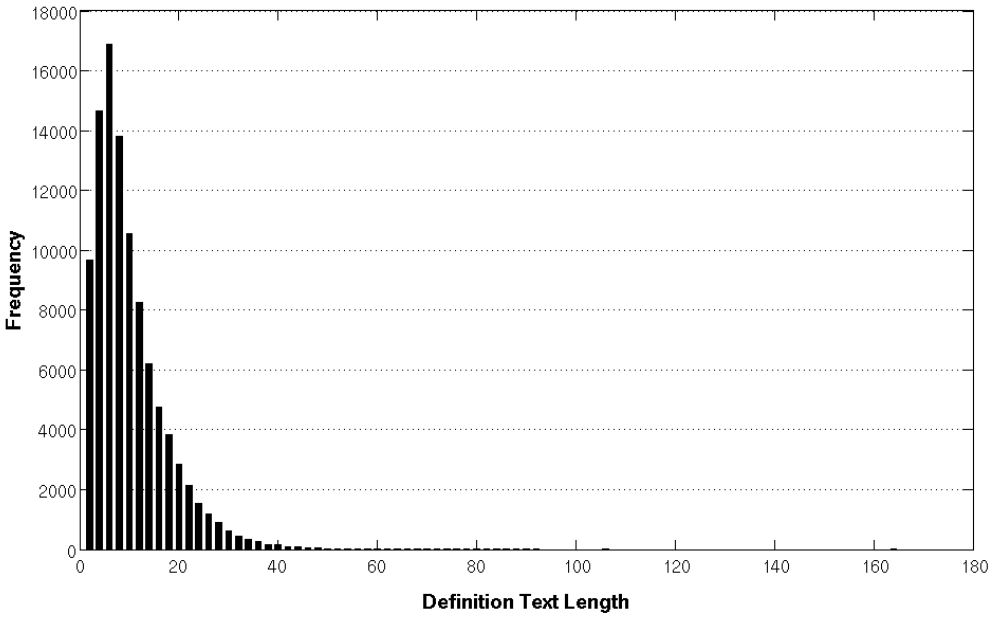


Figure 5: Distribution of definition text length in *The Macquarie Dictionary*

hypernyms. Syntactically, synonymous parts can be identified by very simple typographical patterns within the definientia. Although the patterns are dictionary-specific, their rigid and simple nature usually necessitates minimal human intervention.

The synonymous parts of a definition are of apparent significance in extracting synonyms from definition texts, and the rest of this section is devoted to capturing the features distinguishing such parts from the rest of the definition texts. To begin with, let's first look at the simpler case of very short definitions (consisting of one or two words), which are likely to have only the synonymous parts.

According to the distribution of definition text lengths of *The Macquarie*

Dictionary (Figure 5), there are a surprisingly large number of very short definitions: 5,359 consisting of only one word and 9,672 of two words. Below, these two special cases will be referred to as *single-word definitions* and *double-word definitions*, respectively.

Of all the single-word definitions, half of them are simply synonyms of the definiendum, with the other half being expansions of abbreviations being defined (which could be viewed as synonyms, too). The double-word definitions are more complicated. One scenario is a synonym following a function word, such as indefinite article in a noun definition (e.g., `jailhouse: a jail`) or infinitive *to* (e.g., `damask: to damascene`) in a verb definition. Sometimes both words are synonyms to the definiendum and are separated by a semicolon in between (e.g., `maculate: spotted; stained`). These two cases constitute a large proportion of the double-word definitions and are both useful for synonym extraction. However, there are also cases of a hypernym of the definiendum following a certain quantifier (e.g., `madrigal: any song`) or modifier (e.g., `ablaze: very angry`); these are actually examples of the empty head problem discussed earlier and are not suitable for synonym extraction.

Note that despite the seemingly large numbers of single- and double-word definitions, they are still far from dominant considering the size of most dictionaries; the majority of definitions are at best combinations of synonymous and interpretive parts; sometimes synonymous parts do not exist at all. It is therefore necessary to discover patterns that can deal with these

longer and compounded definitions.

2.4.2 The Pattern-based Extraction Algorithms

The basic idea behind Pattern-based Extraction is to discover occurrence patterns of synonyms in definition texts. In monolingual dictionaries, a word w can often have more than one definition; given a set of patterns $P = \{p_1, \ldots, p_n\}$, PbE looks at each definition of a target word and extracts words that follow any one of the patterns in P as synonyms.

In practice, a pattern p_i takes the form of a regular expression, e.g., "`^.*; (\w+).$`", and if a definition text matches this pattern, the word s corresponding to the group "`(\w+)`" will be proposed as a synonym. For example, if the target word is w ="*separate*", then one of its definitions "`separate: to disconnect; disunite`" matches a seed pattern p ="`^.*; (\w+)$`" and s ="*disunite*" is proposed as a synonym to w. This scenario is referred to as "**a definition matching a pattern** p **on a word** s", as in Line 7 in Algorithm 1.

As mentioned earlier, some very short definitions consist only of synonymous parts, but there are many others that do not have a synonymous part at all. Besides, synonymous parts only provide one or two synonyms to a target word, which again brings up the problem of diminished sizes of synonym sets and, thus, low coverage of the extraction strategy. To address these two issues, after matching the definientia of w against the patterns, PbE scans the entire dictionary and looks at the definition

Algorithm 1 Simple Pattern-based Extraction

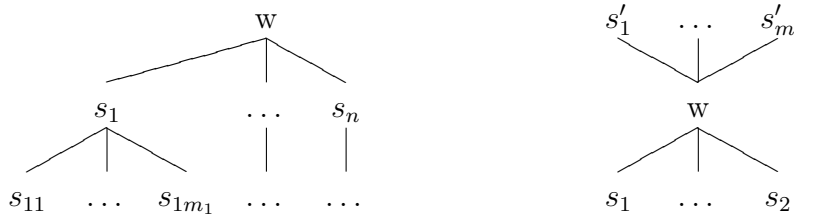
```
1: resultSet ← {}
2: newlyExtractedSet ← {targetWord}
3: repeat
4:    for all w in newlyExtractedSet do
5:       for all definition of w do
6:          for all p in PbEPatterns do
7:             if definition matches p on s then
8:                add s to newlyExtractedSet
9:             end if
10:         end for
11:      end for
12:      remove w from newlyExtractedSet
13:   end for
14:   add newlyExtractedSet to resultSet
15: until newlyExtractedSet is empty
16: return resultSet
```

texts of other words as well; if, in this phase, any word w' has a definition matching any pattern on w, then w' is extracted as a synonym to w. For example, for the target word w ="*separate*", PbE first proposes s ="*disunite*" as a result of Algorithm 1; in addition to this, as a result of Line 12 through 21 in Algorithm 2, PbE scans for definitions matching any patterns on the target word w ="*separate*". To do so, PbE first plugs w into the pattern p, resulting in p_s ="`^.*; separate$`", and then finds and proposes the word w'="*part*" whose definition "`part: to put or keep asunder...; disunite; separate`" matches the plugged-in pattern p_s on "*separate*" (Line 15, Algorithm 2).

The process of scanning other words' definitions resembles that of IIE in Section 2.2, with a difference that now, where and how w appears in the

Algorithm 2 Pattern-based Extraction with IIE-style Scanning

```
1: resultSet ← {}
2: newlyExtractedSet ← {targetWord}
3: repeat
4:    for all w in newlyExtractedSet do
5:       for all definition d of w do
6:          for all p in PbEPatterns do
7:             if d matches p on s then
8:                add s to newlyExtractedSet
9:             end if
10:         end for
11:      end for
12:      for all w′ in Dictionary do
13:         for all definition d′ of w′ do
14:            for all p in PbEPatterns do
15:               p_s ← plug_w_into_p(p, w)
16:               if d′ matches p_s on s′ then
17:                  s′ to newlyExtractedSet
18:               end if
19:            end for
20:         end for
21:      end for
22:      remove w from newlyExtractedSet
23:   end for
24:   add newlyExtractedSet to resultSet
25: until newlyExtractedSet is empty
26: return resultSet
```



(a) Multi-iteration of Algorithm 1

(b) IIE-resembling Scanning in Algorithm 2

Figure 6: Different growth patterns of result set size in Algorithm 1 and 2

definition of w' matters.

On top of the synonym set S extracted by Algorithm 1 for a target word w, Algorithm 2 can improve the coverage of PbE by the additional pass through the dictionary. Similar improvement can also be achieved by running Algorithm 1 for multiple iterations, but the quality of extracted synonyms would vary. If Algorithm 1 runs for more than one iteration, every element in $S = \{s_1, \ldots, s_n\}$ will serve as a target word, and the output of PbE grows like the tree in Figure 6(a); some of the elements in S (e.g., s_1), however, are related to the target word under rare senses, and the offsprings of these elements (e.g., $\{s_{11}, \ldots, s_{1m_1}\}$) will be significantly less synonymous to the original target word w (see Section 2.4.4 for details). In contrast, the IIE-like procedure in Algorithm 2 starts from the target word w instead of the elements in the extracted set S, and the resulting output of PbE grows "upwards" as in Figure 6(b). It is not difficult to see that the elements of $\{s'_1, \ldots, s'_m\}$ are more synonymous to w than those of $\{s_{i1}, \ldots, s_{im_i}\}, i = 1, \ldots, n$.

Here, a pass through the entire dictionary is required for extracting synonyms for every word, but various preprocessings can be used to improve the efficiency. For example, the dictionary size could be reduced to only those definitions which follow at least one of the synonymous part patterns, which is less than 16% in size of the original search space.

2.4.3 Pattern Bootstrapping

Note that the interpretive and synonymous parts are not discrete components rigourously adhered to by lexicographers when composing definitions and thus, they are not universal components of definition texts. Consequently, the number of definitions that follow synonymous-part patterns is quite limited. In addition, the hand-crafted definition texts usually exhibit many typographical variations: some definitions, for example, end with periods while others do not. It is therefore necessary to devise a pattern-finding mechanism that can both obtain new synonym patterns and accommodate variations with minimal hard-wiring or human intervention.

Bootstrapping can achieve both goals at the same time. Specifically, a bootstrapper is initialized with a word w as well as a seed regular expression (regex) pattern p, which could be the simplest pattern for synonymous parts within a given dictionary (e.g., "`.*; (\\w+)$`" for the Macquarie Dictionary). The output is a set of regex patterns that synonyms follow within definition texts. By applying p on the definientia of w, the bootstrapper gets a set S of synonyms of w. Given the fact that many dictionaries have circular

definitions of some sort (Chapter 19, Jurafsky and Martin 2008), it is not unreasonable to assume that at least some elements of S are to appear in the definientia of others'. If any of these occurrences follows some regex pattern p' other than p, p' is then added to the resulting pattern set.

Currently, p's are identified manually, i.e., the bootstrapper would output any of the circular definitions among the set of words S, which provides a human user with potential patterns followed by synonyms.

Figure 7 gives an example of how patterns are bootstrapped from a seed word (*split*) and two seed patterns. Starting from the seed word *split* and the seed patterns "`^(\w+).$`" and "`^.*; (\w+).$`", *cleft* and *divided* are firstly extracted from the two definitions of *split* shown here. *Parted* is also added to the synonym set since its inclusion of *cleft* follows one of the seed patterns. As the number of synonyms grows, it becomes more and more likely that some of the synonyms are to appear in the definitions of others' under patterns different from the seed patterns. In the second iteration, for example, *split* indeed appears in the middle of one definition of *cleft*, resulting in a new pattern "`^*;(\w+);.*$`". More synonyms could in turn be extracted using these new patterns.

The resulting synonym and pattern sets would usually converge, since the circular definition within a dictionary is usually limited to a small number of lexical entries. The resulting pattern set appeals well to intuition; it captures most of the patterns under which one synonym is used to define another. More interestingly, it can indeed accommodate some of the typographical in-

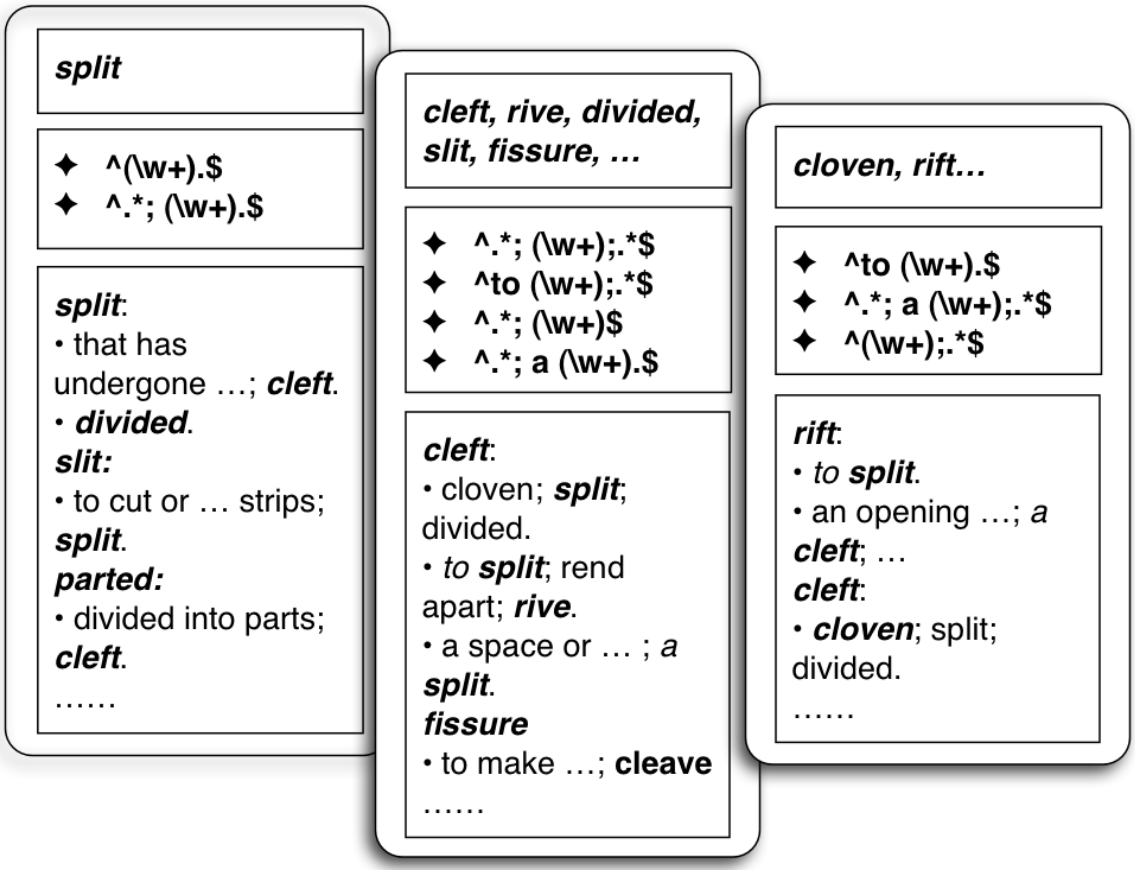


Figure 7: A example of bootstrapping patterns. The three rounded squares in the horizontal layout represent three iterations of bootstrapping; within each of these, the three vertically distributed squares list, from top to bottom, the extracted synonyms, newly-added regular expression patterns, and related definitions, respectively.

consistencies, such as the ending period in definitions (`^.*; (\w+).$` versus `^.*;(+)|`).

It is also worth noting that, since there is no constraint on part of speech in this process, patterns that are apparently for verbs, e.g., "`^to (\w+).$`" are mixed together with those for nouns, e.g., "`^.*; a (\w+).$`". This might have slightly negative effects on the algorithm's efficiency, but not the extraction performance, since POS constraints can always be easily imposed by checking the words' POS tags in the dictionary.

In practice, the bootstrapping algorithm is initialized with seed patterns "`^.*; (\w+\)$`", "`^.*; (\w+\).$`", and"`^.*; \w+\;.*$`", which, although seemingly very simple and specific, match as many as 15,228 (7.45%) definition texts in *The Macquarie Dictionary*. By bootstrapping, 16 more patterns have been discovered, which have doubled the coverage of the set of patterns (32,208 or 15.75% definition texts).

2.4.4 Transitivity of Synonymy

One important feature synonymy is transitivity, i.e., if word a is synonymous to word b and b to c, then it is usually plausible to infer that a is synonymous to c. Considering the common problem of dictionaries' low coverage on synonymy extraction, this property is especially useful since transitivity allows one to take c, and even synonyms of c, as synonyms of a.

Not surprisingly, upon transitive closure over successivly longer paths, the proposed synonyms usually get less synonymous to the original target word,

despite the fact that the synonymous parts of definitions (see Section 2.4.1) usually guarantee synonymy at each step. The different POS and senses of words from definientia are two reasons behind such decline in the degree of synonymy.

POS

Unlike some dictionaries used in previous studies (Muller et al., 2006), definition texts in *The Macquarie Dictionary* are not POS-tagged. Thus, if a word (token) with more than one commonly used POS appears in a definition, there is no easy way of telling which POS the occurrence corresponds to.

In PbE, however, it is possible to distinguish between POS as long as the POS of the target word is provided. First of all, in most monolingual English dictionaries, if a word has more than one POS (and thus, more than one definition), its definitions are usually grouped together by POS. Under this assumption, when the POS of the target word is specified in the input as a regular expression, PbE would only look at definitions under the matching POS tags. Figure 8 gives an example for the word *change*. When the POS is specified at the input as *verb.**, PbE only searches in the first definition (id#745) under POS tag "verb (i)" and extracts *alter* as a synonym; if, on the other hand, the POS is specified as a noun, then PbE goes through the other definition (id#354) and extracts *variation*, *alteration*, etc.

This method works well even though the definition texts are not POS-tagged. Words following the bootstrapped patterns are not only synonymous

```
<RECORD id="000097907">
  <HEAD>[change]
  <BODY>
    <CHUNK>
      <POS>[verb (i)]
        <DEF id="745">
          <DTEXT>[to become different; alter]
          ...
    <CHUNK>
      <POS>[noun]
        <DEF id="354">
          <DTEXT>[variation; alteration; modification;
                  deviation; transformation.]
          ...
```

Figure 8: *Look* and its definitions under different POS

to the definiendum, but also, more often than not, have the same POS. Similar to the POS constraints discussed in section 2.3.1, this filtering mechanism would increase precision of the extraction at the cost of recall.

However, a target word's POS may not always be available. In TOEFL synonym questions, for example, the POS of the question words and the choices are unknown. A procedure for determining the POS of the group of words has actually been developed, but the POS constraint does not help much in improving the performance, partly because, as a dictionary-based approach, low recall is a more prominent issue than precision.

Word Sense

For polysemous words, some senses are much more common than others;

when a rare sense of a word is synonymous to a rare sense of another, two irrelevant words would be associated as synonyms by PbE, and the error could easily be amplified by applying PbE in an unrestricted manner. An extreme example is *look* and *cultivate*: one of the definitions of *cultivate* is "`way of looking or appearing to the eye or mind; aspect`". *Look* is therefore taken synonymous to *aspect*, which is then related to *apparel* by an archaic meaning of the latter: "`aspect; guise`". *Apparel* is then related to *dress* through "`to dress or clothe; adorn; ornament`", which eventually leads to *cultivate* through an agricultural usage of *dress*: "`to cultivate (land, etc.)`".

Here, a special feature in *The Macquarie Dictionary* can be used to further refine the notion of synonymous transitivity. `LABEL` and `NPLABEL` are two tags implying semantic properties of a definition. `LABEL` is usually associated with special meanings or usages of a word, including `Colloquial`, `Obsolete`, `Archaic`, etc., while `NPLABEL` implies the domain specificity of a definition, taking values such as `Agricultural`, `Law`, `Surgery`, etc. Definitions (senses) with these labels are usually where the synonymous transitivity begins to "wander off", and thus, when encountering either of these two tags, the current implementation of PbE terminates transitivity and stops branching off from the word(s) with such tags.

fear(noun)
    solicitude(noun)
        fear(noun), care(noun)
    anxiety(noun)
        eagerness(noun), fear(noun), care(noun),
        jump(noun), disease(noun)
    shock-horror(noun)
        fear(noun), terror(noun)
    terror(noun)
        shock-horror(noun)
    apprehend(verb (i))
        understand(verb (t)), anticipate(verb (t))
        fear(verb (i)), conceit(verb (t))

Figure 9: The tree structure of PbE output for *fear*

2.4.5 Recurrence Filtering

Let us now view the output of PbE as a tree structure to facilitate the discussion that follows. The root of the tree is the target word w, the immediate children of which are the synonyms proposed by PbE in the first round ($S = \{s_1, \ldots, s_n\}$). When there is more than one iteration of PbE, each synonym $s_i \in S$ is taken to be the root of a sub-tree, from which sprout more proposed synonyms (Figure 9).

Ideally, synonymous transitivity ensures that any child node s is synonymous to w, no matter how deep s resides in the tree. In practice, however, the semantic distance between w and a child node s would increase with the depth of s. Due to the circular nature of dictionary definitions, there must be cases in which certain pathes in the tree would return to w after several

fear(noun)
solicitude(noun)
fear(noun), ~~care(noun)~~
anxiety(noun)
~~eagerness(noun)~~, **fear(noun)**
~~care(noun)~~, ~~jump(noun)~~, ~~disease(noun)~~
shock-horror(noun)
fear(noun), **terror(noun)**
terror(noun)
shock-horror(noun)
apprehend(verb (i))
~~understand(verb (t))~~, ~~anticipate(verb (t))~~
fear(verb (i)), ~~conceit(verb (t))~~

Figure 10: Recurrence Filtering (**Bold** for recurrence, ~~strikeout~~ for filter-out)

iterations. A nonempty path p between w and itself is called a *recurrence path*, and the intuition behind *recurrence filtering* is that words on recurrence paths should be more synonymous to the target word than those that "wander off" and never come back. Thus, by finding these closed paths in the dictionary graph, recurrence paths is an intuitively feasible way to deal with the negative effect of polysemy on synonym transitivity. [1]

Thus, to filter out false positives from PbE, only those words in recurrence paths starting from the the target word are proposed as synonyms. However, experiments show that recurrence paths of only the target word is quite sparse; considering the fact that words extracted in the first round of PbE are

[1] In addition to finding closed paths, Professor Gerald Penn also suggested to look at cliques or subgraphs with certain density threshold. The reason why closed paths are prefered here again comes from the sparsity of connectivity in the dictionary graph. Based on my observation, the size of a clique seldom exceeds three, and thus, coverage would again become a prominent issue if only cliques are considered synonym sets.

usually highly synonymous to the target word, the filtering strategy is relaxed to also include recurrence paths of these words. Such relaxation has proved to be successful in that it increases the coverage at little cost to precision. In the example of *fear*, the extracted synonyms are: *solicitude*, *anxiety*, *shock-horror*, *terror*, and *apprehend*, while words such as *care*, *eagerness*, *jump*, *disease*, *understand*, *anticipate*, *conceit*, and *doubt* are filtered out (Figure 10). Both groups generally appeal well to intuition. When compared with existing thesauri (Section 3.2), the precision of the filtered results increases by 18.3 percentage points on average, at a recall loss of about 7 percentage points.

3 Evaluation

One of the most straightforward ways of assessing the quality of a set of automatically extracted synonyms is to compare them against synonymous knowledge possessed by humans. Access of such knowledge can be obtained either by employing human judges to score the results (Muller et al., 2006), or by comparing the results against existing synonymy resources compiled by humans, e.g., thesauri, etc. (Wu and Zhou, 2003).

The idea of human evaluation is appealing since human judges — being native speakers of the language in question — presumably have the best knowledge about synonymy. However, it also exhibits several serious drawbacks. Firstly, this approach is more expensive than an automated process,

both financially and in terms of efficiency. Secondly, subjectivity due to individual differences inevitably has a negative impact on the final evaluation results unless a large sample size is available. Consensus is usually hard to reach, with the resulting evaluation probably biased in some subtle ways.

It is also possible to compare proposed synonym sets against existing, handcrafted resources. This approach, to some extent, lies between human judgement and automated evaluation, in that it is an automated procedure imposed on human knowledge. In most cases, the resource of choice would be thesauri, due to their natural resemblance to the output of synonym extraction algorithms. Nonetheless, apart from availability issues, Muller et al. (2006) argues that "comparing (extracted synonyms) to (an) already existing thesaurus is a debatable means, when automatic construction is supposed to complement an existing one". In the same study, it is shown that even thesauri themselves do not correlate well: when several French thesauri were compared against one another, none of them scored over 60% in F-measure.

Another way to look at synonym evaluation is to establish a mapping between synonymy and semantic similarity. This idea has actually been implicitly adopted by many previous studies. Given a similarity measure, the notion of synonymy can be implemented by listing words in non-increasing order in terms of their similarity scores with respect to a target word or concept. Conversely, once there is a way of extracting synonyms for a target word, a similarity measure can be built, for example, by computing the overlap between the synonym sets of any two words (see Section 3.1.1 for

details).

In this section, both the similarity-based and the thesaurus-based methods are used to evaluate the quality of synonyms extracted by the algorithms developed in Section 2. In Section 3.1, overlapping between extracted synonym sets is used as a similarity score, which, in turn, is used to solve TOEFL synonym questions. Section 3.2 compares the synonym sets to existing thesauri. Both evaluations achieved results comparable to existing studies, which is especially notable considering the lean resource used in this study.

3.1 Solving TOEFL Synonym Questions

3.1.1 Experiment Setup

TOEFL is a standardized test for assessing the English level of non-native speakers. Part of the test is on synonymy, where each question consists of a question word and four candidates, one of which is a synonym to the question word and therefore, the correct answer. Landauer and Dumais (1997) first compiled and used eighty of these questions, which have ever since been frequently used for task-based evaluations in many lexical semantics studies.

The result below comes from the set of the original 80 synonym questions used by Landauer and Dumais (1997). I also collected another 40 questions used as development set.

To evaluate the proposed synonyms from the previous section using these

questions, synonym sets are firstly converted into a semantic similarity measure using Jaccard similarity. Specifically, given two target words w_1 and w_2, their respective feature representations are the extracted synonym sets $S_i = \{s_{i1}, \ldots, s_{in_i}\}$, where n_i is the number of proposed synonyms for w_i, for $i = 1, 2$. The semantic similarity between w_1 and w_2 is then given by:

$$sim(w_1, w_2) = \frac{|S_1 \bigcap S_2|}{|S_1 \bigcup S_2|} \tag{1}$$

For a TOEFL synonym question, the similarity between the question word and each of the candidates can now be computed, and the candidate scoring highest is proposed as the correct answer.

For example, given the question word *fabricate* and four candidates *construct*, *alter*, *select*, and *demonstrate*, each word is firstly associated with the synonym set proposed by a synonym extraction algorithm, e.g., IIE; the question word gets the set {*fabricate, coin, trump up, prefabricate, mint, invent, forge, spin*}, the first candidate *construct* gets {*construct, fabricate, cantilever, improvise, laminate, ...*}, and so on. Note that a word is always considered a synonym to itself, and thus included in the synonym set. In the above synonym question, the first candidate *construct* is the only one with a set that overlaps with the set of the question word and consequently, it receives the highest score and is considered the correct answer.

Such an evaluation scheme can indeed reflect the degree of synonymy of the extracted synonyms to their target word, since the quality of synonym

extraction can positively determine the quality of Equation 1 as a similarity score, which in turn would result in better scores in solving the TOEFL synonym questions. In addition, such evaluation also has the advantage of allowing for easy interpretation of the results.

This evaluation scheme also exhibits several immediate problems, however. First of all, synonymy is sufficient but not necessary for achieving higher scores on the test. Suppose, as an extreme example, that the sets S_i are composed of antonyms instead of synonyms to their corresponding target words w_i $(i = 1, 2)$; if two words are synonymous, then they are equally likely to have similar antonyms instead of synonyms, and the two sets S_1 and S_2 can also correlate well and help to achieve better scores in the same synonym questions. Consequently, higher scores in such tests are only necessary but not sufficient to infer synonymous relations between the target words and their corresponding extracted sets.

Besides, there are several possible ways of getting ties among choices. Two choices might have exactly the same Jaccard similarity with the question word, although this situation never happens in the 120 questions used in this experiment. Another case is when none of the choices overlap with the question word (i.e., none of them share any synonyms with the question word), or, when the synonym set for the question word is empty from a specific extraction strategy. For IIE, for example, uncommon words are less likely to appear in other words' definitions and thus would produce diminished or even empty synonym sets. This turns out not to be a singular case

in the data, since one of the characteristics of TOEFL is to test non-native speakers on a more advanced vocabulary including many uncommon words.

In contrast, similarity-based approaches usually assign non-zero scores to most word pairs. Consequently, results reported in such studies are only concerned with how many questions were correctly solved. Here, questions with ties are taken to be "unsolvable" and no choice would be assigned. Thus, the results later presented borrow the notions of precision and recall from information retrieval. Recall, in this case, denotes the proportion of questions that have non-zero scores and thus, are solvable by choosing the highest score among the four, while precision denotes of how many are solved correctly among these solvable questions.

One way to break ties is to combine IIE synonym sets with definientia, which are readily accessible and more likely to correlate. To better reflect synonymy, extracted synonyms and definientia are weighted differently when computing the norms in Equation 1. Specifically, for two target words w_1 and w_2, instead of using only their extracted synonym sets $S_i = \{s_{i1}, \ldots, s_{in_i}\}$ as their feature vectors, words in their definientia $D_i = \{d_{i1}, \ldots, d_{im_i}\}$ are also taken into account, i.e., $v_i = S_i \bigcup D_i, i = 1, 2$. Meanwhile, when two target words are compared, their degree of overlap not only depends on how many common elements they have in v_i's, but also on what kind of elements they have in common. The basic idea is to give more weight to synonyms than to ordinary definiens words: suppose w_1 and w_2 have two common elements v_{1i} and v_{2j} in the i-th and j-th positions of their feature vectors, respectively,

	Precision	Recall	F_1	Accuracy
Definientia (Baseline)	51.3%	**97.5**%	67.2%	50.0%
IIE	**100.0%**	50.0%	66.7%	50.0%
IIE+Lemmatization	93.8%	75.0%	83.4%	70.4%
Weighted IIE+Definientia	87.2%	**97.5**%	92.1%	85.0%
PbE	95.5%	55.0%	69.8%	52.5%
PbE+Lemmatization	93.6%	77.5%	84.8%	72.5%
Weighted PbE+Definientia	90.6%	**97.5**%	**93.9**%	**88.3**%

Table 6: Evaluation of extracted synonyms on TOEFL synonym questions

and the weight is α for synonyms and β for ordinary definientia. If $v_{1i} \in S_1$ and $v_{2j} \in S_2$, then this overlapping is weighted by α^2; if $v_{1i} \in S_1$ while $v_{2j} \in D_2$, or vice versa, then the overlapping weight is $\alpha \cdot \beta$; if both are parts of the ordinary definientia, then the weight becomes β^2. In the current implementation, $\alpha = 5$ and $\beta = 1$ (a maximum likelihood estimation based on the development data). The same weighting scheme has also been applied to PbE.

3.1.2 Evaluation Results and Discussions

Table 6 shows the evaluation results of solving TOEFL synonym questions by IIE, PbE, and their variants as discussed in Section 3.1.1. The baseline uses all the words from the definientia of each target word, which resembles the Lesk algorithm used in word sense disambiguation (Lesk, 1986). The comparison it makes here is interesting in that IIE and PbE try to distinguish synonyms from the rest of the definientia, while the baseline uses them all non-discriminatively; improvements over the baseline would thus reflect how

well the discrimination is made.

The results show that, due to the problem of coverage, half of the questions have their choices tied up when using IIE. Nonetheless, for the solvable ones, IIE scored 100% in precision. PbE has, on the other hand, exhibited higher recall, although by a very small margin. In contrast to the results of the other experiment in Section 3.2, PbE with recurrence filtering (Section 2.4.5) is not included in Table 6, because the bottleneck here is on recall, which will only decrease when any filtering is used.

Further examination of the tied questions reveals that they are either due to diminished synonym sets that do not overlap with one another, or empty synonym sets of the question word that assign zero scores to all four candidates. For the question <`functional: alternate; unknown; original; usable`>, although all of the five words have non-empty synonym sets, none of them have any word in common and thus, all four candidates receive a score of zero.

Meanwhile, for inflected words in the data, using base forms in IIE and PbE indeed improves the coverage (recall) by a large margin (25.0 and 22.5 percentage points, respectively), even through a very simple lemmatization process. This has very well supported the argument made in Section 2.2.2 about morphology issues.

The best F_1 score is achieved by the weighted PbE+Definientia approach, which, by including the definientia in the feature vectors, significantly improved the recall (20.0 percentage points) at a relatively smaller cost to pre-

cision (3.0 percentage points).

The state of the art results for solving TOEFL synonym questions is shown in Table 7[2]. The percentage of correctly solved questions is equivalent to "accuracy" in Table 6. By including definientia in feature vectors, PbE ranks first among the lexicon-based methods (underlined) by a margin of nearly 10 percentage points over the second best result; when using only the extracted synonyms as feature vectors, PbE is still comparable to existing results.

[2]http://aclweb.org/aclwiki/index.php?title=TOEFL_Synonym_Questions_%28State_of_the_art%29

Algorithm	**Reference for algorithm**	**Reference for experiment**	**Type**	**Correct**	**95% confidence**
RES	Resnik (1995)	Jarmasz and Szpakowicz (2003)	Hybrid	20.31%	12.89−31.83%
LC	Leacock and Chodrow (1998)	Jarmasz and Szpakowicz (2003)	Lexicon-based	21.88%	13.91−33.21%
LIN	Lin (1998)	Jarmasz and Szpakowicz (2003)	Hybrid	24.06%	15.99−35.94%
Random	Random guessing	1 / 4 = 25.00%	Random	25.00%	15.99−35.94%
JC	Jiang and Conrath (1997)	Jarmasz and Szpakowicz (2003)	Hybrid	25.00%	15.99−35.94%
LSA	Landauer and Dumais (1997)	Landauer and Dumais (1997)	Corpus-based	64.38%	52.90−74.80%
Human	Average non-English US college applicant	Landauer and Dumais (1997)	Human	64.50%	53.01−74.88%
DS	Pado and Lapata (2007)	Pado and Lapata (2007)	Corpus-based	73.00%	62.72−82.96%
PMI-IR	Turney (2001)	Turney (2001)	Corpus-based	73.75%	62.72−82.96%
HSO	Hirst and St.-Onge (1998)	Jarmasz and Szpakowicz (2003)	Lexicon-based	77.91%	68.17−87.11%
JS	Jarmasz and Szpakowicz (2003)	Jarmasz and Szpakowicz (2003)	Lexicon-based	78.75%	68.17−87.11%
PMI-IR	Terra and Clarke (2003)	Terra and Clarke (2003)	Corpus-based	81.25%	70.97−89.11%
CWO	Ruiz-Casado et al. (2005)	Ruiz-Casado et al. (2005)	Web-based	82.55%	72.38−90.09%
PPMIC	Bullinaria and Levy (2006)	Bullinaria and Levy (2006)	Corpus-based	85.00%	75.26−92.00%
GLSA	Matveeva et al. (2005)	Matveeva et al. (2005)	Corpus-based	86.25%	76.73−92.93%
PbE+Definientia	This Paper	This Paper	Lexicon-based	88.30%	79.72−94.73%
LSA	Rapp (2003)	Rapp (2003)	Corpus-based	92.50%	84.39−97.20%
PR	Turney et al. (2003)	Turney et al. (2003)	Hybrid	97.50%	91.26−99.70%

Table 7: Solving TOEFL synonym questions — state of the art

3.2 Comparison Against a Combined Thesaurus

3.2.1 Experiment Setup

The experiment discussed in this section resembles that of Wu and Zhou (2003). Target words are first selected from a corpus according to POS and frequency. A thesaurus is then constructed by combining WordNet synsets and an on-line version of Roget's Thesaurus (Phelps, 1995). After applying synonym extraction algorithms to the target words, the resulting synonym sets are compared against the combined thesaurus.

The corpus for choosing the target words is the 1987-1989 WSJ. POS of these target words include nouns, verbs, and adjectives. Word frequencies range from approximately 8,000 (high) to 1,000 (medium), to 50 (low) occurrences in the corpus. The combined thesaurus is constructed in exactly the same manner as by Wu and Zhou (2003), i.e., given a target word, its corresponding WordNet synsets and synonym sets from the Roget's Thesaurus are extracted and combined into a larger synonym set. The resulting thesaurus is then used as a gold standard against which the extracted synonyms are compared. The final results are reported in terms of precision, recall, and F_1.

3.2.2 Evaluation Results and Discussions

The results for PbE, RIE and their variants are listed in Table 8. The letters `H`, `M`, and `L` stand for the high, medium, and low frequency of target words in

		NN			JJ			VB	
	P	R	F	P	R	F	P	R	F
RIE	H .057	.032	.041	.045	.053	.049	.113	.026	.043
	M .168	.040	.051	.132	.051	.073	.175	.041	.066
	L .053	.023	.032	.087	.013	.023	.168	.026	.046
PbE	H .109	.189	.138	.109	.291	.158	.119	.372	.180
	M .132	.175	.150	.125	.288	.175	.125	.269	.171
	L .109	.164	.131	.117	.172	.139	.168	.208	.186
PbE_Recurrence	H .329	.113	.168	.334	.174	.229	.489	.181	.264
	M .347	.112	.169	.332	.157	.213	.415	.149	.219
	L .225	.090	.128	.335	.089	.140	.370	.116	.177
PbE_Recurrence + RIE_Filtered	H .267	.121	.167	.262	.185	.217	.160	.185	.263
	M .035	.114	.166	.308	.161	.212	.043	.151	.220
	L .215	.091	.127	.340	.095	.148	.361	.117	.177

Table 8: Evaluation of extracted synonyms on the combined thesaurus

the WSJ, and `P`, `R`, `F`, for precision, recall, and F_1, respectively. Since these experiments differ from those of Wu and Zhou (2003) in terms of corpus, frequency counts, and POS coverage, the results are not directly comparable and thus, their results are not listed.

As is shown in Table 8, the output of RIE does not correlate well with the combined thesaurus. Precision is, in general, slightly better than recall, but F_1 seldom scores over 5%. Also note that RIE usually exhibits better precision for low frequency words and better recall for high frequency ones; medium-frequency target words uniformly perform better than the other two groups, mainly because these words are neither too frequent to appear in many words' definitions, nor too rare to not appear at all. This accords with the argument about RIE and target word frequency (Section **??**).

In contrast, PbE exhibits significant improvements over RIE on both precision and recall (on average, twice as high in precision and five times so in recall). It also appears more robust to variation in frequency. After the recurrence filtering, the precision of PbE is almost tripled, which, at a relatively small cost of recall drop (6-7 percentage points), yields the best F_1 score among all proposed methods. A combination of the filtered versions of PbE and RIE has also been compared to the gold standard, but the result (on the fourth row) is rather disappointing.

As one advantage of using a dictionary as a source for synonym extraction, PbE and RIE both exhibit robustness across different POS.

As mentioned earlier, this experiment resembles those of Wu and Zhou (2003), whose best result is achieved by a linear interpolation of three methods (one lexicon-based and two distributional). Their experiments did not include adjectives; when comparing the results on nouns and verbs, however, the precision of PbE with recurrence filtering is, on average, 6 percentage points higher than their best result on nouns and 14.3 percentage points higher on verbs. Coverage, however, still remains a major issue, with recalls of PbE_Recurrence filtering on both nouns and verbs lower than that of Wu and Zhou (2003) (11.3 percent and 16.1 percentage points, respectively).

4 Future Work

4.1 Synonym Extraction

As mentioned in Section 2.4.4, one of the most prominent problems in PbE is that, due to the polysemous nature of many English words, transitivity of synonymy is not well preserved after several iterations of PbE. For example, the word *appearance* appears in definitions of both *forthcoming* (`a coming forth; appearance`) and *look* (`general aspect; appearance`); in both cases, *appearance* is indeed highly synonymous to the definiendum when the definitions are viewed separately. To know that *forthcoming* and *look* are not synonymous through their common synonym *appearance*, however, requires further knowledge about the definition texts, such as word sense disambiguation or sense-tagging, etc.

Another improvement can be made by automating the process of pattern bootstrapping. Identifying regular expressions from free texts is a challenging task; nonetheless, recall that in Section 2.4.3, the rules used in manually identifying patterns are well specified and thus, should be feasible for an automated process to follow.

Dictionary-based synonym extraction could also adopt paradigms that are completely different from PbE or IIE. Considering the sequential nature of definition texts, for example, one can train a Hidden Markov Model in which the states are semantic functions of defining words (e.g., synonyms, nuances, etc.) and the outputs are POS of the words or the word tokens

```
1. administered
   a.  managed
   b.  recognized
   c.  unregulated
   d.  justified
```

Figure 11: A revised TOEFL synonym question containing a related but not synonymous choice.

themselves. The details of this approach are discussed in Section 4.2.

There is also a promising outlook for developing new evaluation schemes of synonym extraction tasks. For example, the current version of TOEFL synonym questions used in Section 3.1 does not discriminate between strict synonymy and the more general notion of semantic relatedness because, for a given question, the three incorrect choices are almost always totally irrelevant to the question word. It would be interesting to make the decoys "trickier" by including semantically related but not synonymous words.

Figure 11 shows a similar example in contrast to Figure 4 in Section 2.3.3. Notice that the third choice has been changed from *opposed* to *unregulated*, which is related but not synonymous to the question word *administered*. It would be interesting to observe how *managed* and *unregulated* are to compete in synonym/related-word extraction systems. Such an investigation would undoubtedly shed light on the differences between strict synonymy and the more general notion of semantic relatedness.

It might be difficult to devise experiments that directly measure the qual-

ity of synonyms; nonetheless, there are many NLP applications that use synonyms as a component of their systems. Theoretically, all such applications can be used as task-oriented benchmarks for evaluating extracted synonyms, and a study of their individual characteristics and applicability would also be a valuable contribution to the field.

4.2 Nuances Differentiation and Lexical Choice

Aside from the *synonymous parts* of definition texts used in PbE, the *interpretive parts* (Section 2.4.1) also contain rich semantic information about definienda and genus terms. If a genus term is a hypernym of its corresponding definiendum (Chodorow et al., 1985), then in order to form a descriptive and precise dictionary definition, there must be additional information in the definition to distinguish the definiendum from other hyponyms of the genus term. Such information could be used as a representation of nuances between synonyms, which is a critical part of many applications (Edmonds and Hirst, 2002).

Let us again look at the example of `glance: to look briefly or quickly` from Section 2.2. Here, the definiendum *glance* is a hyponym of its genus term *look*; the adverbs *briefly* and *quickly* are used to distinguish the particular action of *glance* from other ways of *looking*. Together with a certain representation of context, such nuance representations (i.e., the two adverbs in this case) could then help us choose the most appropriate word from a group of synonyms: if a context somehow indicates hastiness instead of

scrutiny, then the system would propose *glance* rather than *examine*. This is the so-called *lexical choice* problem (Edmonds and Hirst, 2002).

To reveal such information on nuances from dictionary definitions, one can follow the idea of PbE to find their occurrence patterns. This, however, is not at all as straightforward as identifying synonymous parts. Intuitively, the POS of a definiendum and its nuance-representing definiens do exhibit certain patterns: verbs are usually distinguished by adverbs and nouns by adjectives. To find generic patterns for nuance information, however, is, if at all possible, quite beyond the capability of rule-based approaches like PbE, since interpretive parts of a definition usually come with greater syntactic variation and complexity. Accurate processing of such parts very likely requires POS-tagging or even parsing the definition text (as is done in Barnbrook 2002).

In view of the sequential nature of definition texts, a more plausible approach is to train a Hidden Markov Model (HMM) to capture various synonym-related semantic information. The hidden states can represent semantic functions, such as hypernyms (or genus terms, e.g., *look* in the previous example), nuances (e.g., *briefly*, *quickly*), and words that only serve syntactic functions (e.g., *to*, *and*). The output consists of either POS of the definiens words, the actual word tokens, or a combination of the two, the choice of which largely depends on their sparsity in the training data. Such an HMM, once trained, can propose the most probable state sequences given unseen definition texts, and various semantic functions can then be assigned

to the sequences of definiens words.

More interestingly, this approach can also work as a synonym extraction paradigm since synonyms, if any, can be regarded as a type of semantic function within definition texts and thus, be extracted along with the above-mentioned others.

Note that definition texts are but one of the possible resources for learning synonym nuance information. Inkpen and Hirst (2006) used synonym dictionaries to gather information on nuances. The results were placed into a hand-crafted hierarchy, and further filtered by several post-processing procedures. As for evaluation, both types of nuance representation are yet to be fit into the bigger image of the lexical choice problem, which involves the even more difficult task of representing the semantics of the contexts.

References

H. Alshawi. Processing Dictionary Definitions with Phrasal Pattern Hierarchies. *Computational Linguistics*, 13(3-4):195–202, 1987.

R.A. Amsler. *The Structure of the Merriam-Webster Pocket Dictionary*. PhD thesis, The University of Texas at Austin, 1980.

G. Barnbrook. *Defining Language: A Local Grammar of Definition Sentences*. John Benjamins Publishing Co, 2002.

Daniel M. Bikel and Vittorio Castelli. Event Matching Using the Transitive Closure of Dependency Relations. *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2008.

V.D. Blondel and P.P. Senellart. Automatic Extraction of Synonyms in a Dictionary. *Proc. of the SIAM Workshop on Text Mining*, 2002.

B. Boguraev and E. Briscoe. *Introduction to Computational Lexicography for Natural Language Processing*. Longman Publishing Group White Plains, NY, USA, 1989.

M.S. Chodorow, R.J. Byrd, and G.E. Heidorn. Extracting Semantic Hierarchies from a Large On-line Dictionary. *Proceedings of the 23rd Annual Meeting on Association for Computational Linguistics*, pages 299–304, 1985.

I. Dagan, L. Lee, and F.C.N. Pereira. Similarity-Based Models of Word Cooccurrence Probabilities. *Machine Learning*, 34(1):43–69, 1999.

A. Delbridge et al. *The Macquarie Dictionary*. Macquarie Library, McMahons Point, NSW, Australia, 1981.

P. Edmonds and G. Hirst. Near-Synonymy and Lexical Choice. *Computational Linguistics*, 28(2):105–144, 2002.

D. Freitag, M. Blume, J. Byrnes, E. Chow, S. Kapadia, R. Rohwer, and Z. Wang. New Experiments in Distributional Representations of Synonymy. *Proceedings of the 9th Conference on Computational Natural Language Learning*, 2005.

J.A. Guthrie, L. Guthrie, Y. Wilks, and H. Aidinejad. Subject-dependent Co-occurrence and Word Sense Disambiguation. *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pages 146–152, 1991.

L. Guthrie, B.M. Slator, Y. Wilks, and R. Bruce. Is There Content in Empty Heads. *Proceedings of the 13th Conference on Computational Linguistics*, pages 138–143, 1990.

M. Hagiwara. A Supervised Learning Approach to Automatic Synonym Identification based on Distributional Features. *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2008.

M.A. Hearst. Automatic Acquisition of Hyponyms from Large Text Corpora. In *Proceedings of the 14th Conference on Computational Linguistics-Volume 2*, pages 539–545. Association for Computational Linguistics Morristown, NJ, USA, 1992.

N.D. Ho and F. Cédrick. Lexical Similarity Based on Quantity of Information Exchanged - Synonym Extraction. *Proceedings of the Research Informatics Vietnam-Francophony, Hanoi, Vietnam*, pages 193–198, 2004.

D. Inkpen and G. Hirst. Building and Using a Lexical Knowledge Base of Near-Synonym Differences. *Computational Linguistics*, 32(2):223–262, 2006.

D. Jurafsky and J.H. Martin. *Speech and Language Processing*. Prentice Hall, 2008.

T.K. Landauer and S.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.

M. Lesk. Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation*, pages 24–26. ACM New York, NY, USA, 1986.

D. Lin. Automatic Retrieval and Clustering of Similar Words. *Proceedings of the 17th International Conference on Computational Linguistics*, pages 768–774, 1998.

I. Merriam-Webster. *Merriam-Webster's Collegiate Dictionary*. Merriam-Webster, 2003.

S. Mohammad and G. Hirst. Distributional Measures as Proxies for Semantic Relatedness. *Submitted for publication*, 2005.

S. Mohammad, B. Dorr, and G. Hirst. Computing Word-Pair Antonymy. *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 982–991, 2008.

P. Muller, N. Hathout, and B. Gaume. Synonym Extraction Using a Semantic Distance on a Dictionary. *Proceedings of TextGraphs: The Second Workshop on Graph Based Methods for Natural Language Processing*, pages 65–72, 2006.

L. Page, S. Brin, R. Motwani, and T. Winograd. The Pagerank Citation Ranking: Bringing Order to the Web, 1998.

M.F. Phelps. *Rogets II: The New Thesaurus*. Boston: Houghton Mifflin, 1995.

P. Procter et al. *Longman Dictionary of Contemporary English*. London, 1978.

R. Reichert, J. Olney, and J. Paris. Two Dictionary Transcripts and Programs for Processing Them. Volume I. The Encoding Scheme, Parsent and Conix. 1969.

I. Sidney and J. Ronald. *The Bantam Roget's Thesaurus*. Bantam Books, 1990.

L. Van der Plas and J. Tiedemann. Finding Synonyms Using Automatic Word Alignment and Measures of Distributional Similarity. *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 866–873, 2006.

H. Wu and M. Zhou. Optimizing Synonym Extraction Using Monolingual and Bilingual Resources. *Proceedings of the Second International Workshop on Paraphrasing-Volume 16*, pages 72–79, 2003.