Extracting Synonyms from Dictionary Definitions

Tong Wang and Graeme Hirst
Department of Computer Science
University of Toronto
Toronto, ON, M5S 3G4, Canada
tong,gh@cs.toronto.edu

Abstract

We investigate the problem of extracting synonyms from dictionary definitions. Our premise for using definition texts in dictionaries is that, in contrast to free-texts, their composition usually exhibits more regularities in terms of syntax and style and thus, will provide a better controlled environment for synonym extraction. We propose three extraction methods: two rule-based ones and one using the maximum entropy model; each method is evaluated on three experiments — by solving TOEFL synonym questions, by comparing extraction results with existing thesauri, and by labeling synonyms in definition texts. Results show that simple rule-based extraction methods perform surprisingly well on solving TOEFL synonym questions; they actually out-perform the best reported lexicon-based method by a large margin, although they do not correlate as well with existing thesauri.

Keywords

Lexical semantics, synonym extraction, dictionary definition mining, maximum entropy classification

1 Introduction

1.1 Motivation

Synonymy is one of the classic lexical semantic relations based on which lexical semantic taxonomies such as WordNet (Fellbaum et al., 1998) are constructed. Despite their usefulness in various NLP studies, such taxonomies are usually considered expensive resources since they are often manually constructed. Consequently, much research effort has been devoted to automatically extracting words of certain semantic relations from various free-text corpora (e.g., Hearst 1992; Lin 1998, etc) or even building an entire taxonomy (e.g., Chodorow, Byrd, and Heidorn 1985).

Meanwhile, identifying characteristic features for synonymy is non-trivial. Many studies in this direction have started out with intuitively appealing ideas, but in practice, there are always surprises for intuition. Dependency relations used by Lin (1998) relate two nouns if, for example, they often serve as the object of the same verb. When it comes to adjectives, however, the relation established as such is no longer guaranteed to be strict synonymy, since two adjective antonyms may modify the same noun as well. As another example, when two words in one language are often translated into the same word in another language, it seems very natural to regard the two words as synonyms in the source language (Wu and Zhou, 2003), but the mapping on the lexical level between languages is far from bijective, which in turn leads to many exceptions to the hypothesis.

The difficulties in finding features for strict synonymy partly come from the syntactic and stylistic diversity in free-texts — this is the motivation behind using dictionary definitions as a resource for synonym identification. Unlike the extraction strategy used by Hearst (1992), where hyponyms would necessarily follow the phrase *such as*:

$$NP_0 \text{ such as } \{NP_1, NP_2, \ldots, (\text{and}|\text{or})\}\ NP_n$$
$$\forall i = 1, 2, \ldots, n, hyponym(NP_i, NP_0)$$

there seems to be much fewer patterns, if any, to follow for using synonyms in free-text writing.

In contrast, in the composition of dictionary definitions, there is usually much more regularity in terms of how synonyms of the word being defined should and would appear. Consequently, dictionary definition texts, as a special form of corpora, can provide a better "controlled" environment for synonym distribution and thus, it would presumably be easier to find characteristic features specific to synonymy within definition texts. Given this assumption, our goal is to find synonyms for a give word (*target word*) from the collection of all definition texts in a dictionary and subsequently evaluate the quality of the proposed synonyms.

1.2 Related Work

One of the first attempts at extracting synonyms (or semantically related words in general) from dictionary definitions is that of Reichert et al. (1969)[1], where an *inverted index* of a dictionary is built to relate a *definiendum* (the word being defined, pl. *definienda*) to its *definientia* (defining words). Despite the coarse definition of relatedness, the idea itself proved to be inspiring in formalizing the problem in graph-theoretical language, with words corresponding to nodes and edges pointing from definienda to definientia.

On the basis of this graph (usually referred to as a *dictionary graph*), many interesting variants have evolved from the original idea of inverted indexing. Taking the graph as the web, Blondel and Senellart (2002) employed an algorithm similar to PageRank (Brin et al., 1998), and similarities between words can be computed using their adjacency matrix. Alternatively, the problem can be viewed from an information theory perspective and formalized to propagate information content instead of endorsement (Ho

[1] Unfortunately, we have not been able to access the original work of Reichert et al. (1969); we resort to the description of their method by Amsler (1980) instead.

International Conference RANLP 2009 - Borovets, Bulgaria, pages 471–477

and Cédrick, 2004). Another example (Muller et al., 2006) is to simulate a random walk on the graph by building and iteratively updating a Markovian matrix, and the distance between words corresponds to the average number of steps the random walk takes from one node to the other.

Despite all these interesting variations, the original method of inverted indexing has been left unevaluated for decades, and one of the objectives of this paper is to bring the evaluation of this method into the modern paradigm of NLP. Together with this algorithm, two other extraction methods will be discussed in Section 2; evaluations of these methods are conducted in three experiments following in Section 3. Section 4 will conclude the study with prospects for future work.

2 Extraction Methods

In this section we propose three extraction methods, all of which extract synonyms from definition texts in the *Macquarie Dictionary* (Delbridge et al., 1981). The first two methods are rule-based and use original definition texts, while the third one (a maximum entropy model) is based on POS-tagged definitions[2].

2.1 Inverted Index Extraction

As mentioned above, inverted index extraction (IIE) is one of the earliest attempts at using dictionary definitions to extract synonyms or related words. Specifically, for a given target word t , the entire dictionary is scanned in search of words whose definition texts contain t, and such words are considered related to t. Since both synonymy and relatedness are symmetric, it is equivalent to say that every definiendum is related to all its definientia, or that the dictionary graph is an unweighted, undirected graph where every pair of neighbors is considered equally related.

Many problems arise with the simplicity of this notion for relatedness. For example, every word in a definition is treated equally, regardless of its POS, syntactic function, or position in the definition text. In practice, however, some definientia appear in insignificant positions (such as part of a subjunctive clause or a phrase, etc.) and thus are not as related as they are taken to be.

There are simple heuristics to deal with such false positives. Taking POS for example, one can specify the POS of a given target word and only extract words that are of the same POS. Constraints can also be applied on where a target word is allowed to appear in order to be considered a synonym of the corresponding definiendum. Apart from all these, a more pertinent issue, as it turned out in later experiments, is actually the low coverage for low-frequency target words, which do not appear often (or even not at all) in other words' definitions. In fact, this conforms with the claim made by several previous authors (e.g., Wu and Zhou 2003) that coverage is a key issue for dictionary-based methods.

2.2 Pattern-based Extraction

The intuition behind pattern-based extraction (PbE) is based on the regularity in dictionary definition text composition. In PbE, instead of relating a definiendum to every word in its definition (as in IIE), we focus more on those definientia that follow particular patterns synonyms tend to follow in definition texts. Consequently, one of the objectives of PbE is to discover such patterns.

2.2.1 Basic Algorithm

The synonym extraction and pattern finding process are related in a bootstrapping manner, as shown in Figure 1. We start with 1) a set of words containing only the target word $W_0 = \{t\}$, e.g., *split*, and 2) a small set of regular expressions P_0 capturing the most basic and intuitive patterns that synonyms usually follow in definition texts. For example, if there is only one word in the definition text, it must be a synonym of the definiendum; this corresponds to the first regex pattern in the left-most block in Figure 1, i.e., `^(\w+).$`, and the same idea applies the other two patterns as well.

Synonym extraction

Given W_0 and P_0, we now follow this procedure for synonym extraction:

1. If any word w matches any pattern $p \in P_i$, extract w as synonym of t and update the word list $W_i = W_i \cup \{w\}$.
2. If t matches any pattern $p' \in P$ in the definition text of some other word w', extract w' as synonym of t and update the word list $W_i = W_i \cup \{w'\}$.
3. Take each word $w \in W_i$ as target word and repeat 1 and 2; add all resulting synonyms to W_i and denote the new set W_{i+1}.

Pattern bootstrapping

For the moment, we assume that words in W_{i+1} appear in each other's definition in patterns other than the ones we started with in P_i.[3] We update[4] the regex set P_i by adding these new patterns, and repeat synonym extraction with W_{i+1} and P_{i+1}.

The above process will converge if our hypothesis on the regularity of definition text composition is valid, i.e., when composing definition texts, lexicographers tend not to use random patterns to include synonyms in the definition texts. In practice, it converges in all the test cases used.

Note that when combining the three steps in the synonym extraction phase, the algorithm is actually building a dictionary graph in which a definiendum is related to only those definientia following specific patterns. This is different from the dictionary graphs in IIE and its variants, which relate a definiendum to all its definientia.

2.2.2 Transitivity of Synonymy and Transitive Closure

The notion of transitivity of synonymy is implicitly adopted, especially in Step 3 in the synonym extraction

[2] We used the Stanford POS tagger for this purpose (`http://nlp.stanford.edu/software/tagger.shtml`).

[3] In case they do not, we can either start off with a group of known synonyms instead of one target word t, or even with another word that does lead to more appropriate situations in terms of W_{i+1}, because at this stage, we aim at bootstrapping for patterns rather than finding synonyms for any particular word.

[4] The update is currently done manually, and could be replaced by automatic recognition of the most general regular expression patterns by, for example, dynamic programming.

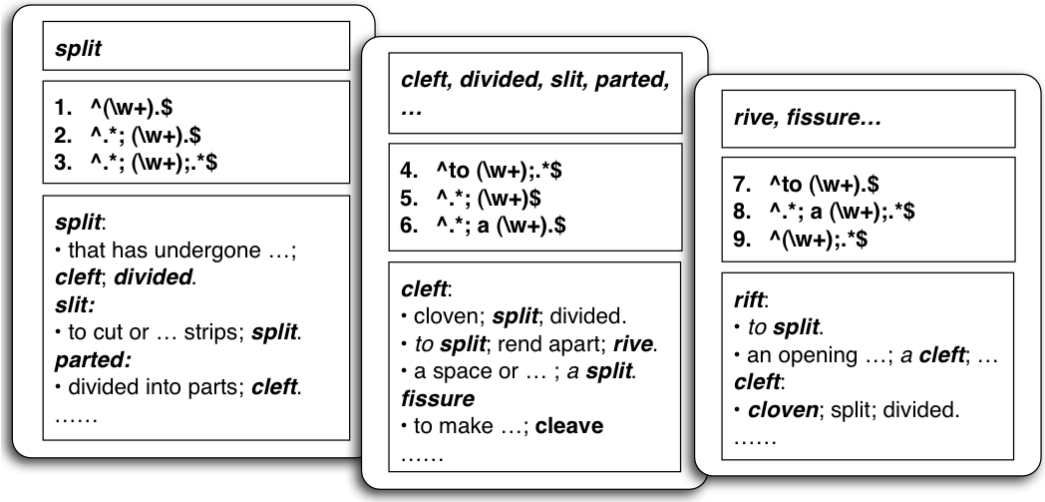

Fig. 1: *Bootstrapping between synonym extraction and pattern finding. The three rounded squares in the horizontal layout represent three iterations of bootstrapping; within each of these, the three vertically distributed squares list, from top to bottom, the extracted synonyms, newly-added regular expression patterns, and related definitions, respectively.*

phase above. In general, if a word A is synonymous to B and B to C, it is usually fair to deduce that A is a synonym of C. In the context of dictionary definitions, however, the validity of such deduction is severely compromised, because the input to the extraction process is word tokens under different senses and/or even of different POS.

Similar to IIE, there are easy remedies for confusions on POS, e.g., by specifying a POS with a target word (say, *adjective* for *split*) and looking only for definitions under the specified POS. In view of transitivity, most words following any of our patterns (e.g., *cleft*) can be safely assumed to have the same POS as that of their corresponding definienda, and starting from these words, we can follow definitions under the same POS (e.g., *cleft* as an adjective instead of a verb) for further extraction. For word senses, however, this assumption is no longer valid, since universal specification of word senses is unavailable in most dictionaries.

A more general solution to this problem is to find *transitive closure*, corresponding to circles in the dictionary graph. This idea is based on the hypothesis that definitions are circular in nature (Jurafsky and Martin, 2008). Starting from a given target word, transitivity is applied regardless of POS and word sense differences; once we encounter the target word again[5], we consider every word on this circle synonyms to the target word. As is shown in later experiments, where the extracted words are compared with existing thesauri, this approach almost triples precision at relatively low cost in recall.

2.3 Extraction using Maximum Entropy

Although PbE exhibits excellent precision in extracting synonyms (as all three experiments suggest in Section 3 below), relying solely on the limited number of patterns will again bring up the issue of coverage. This motivates more-general learning methods that would treat definition texts in a less-specific way.

As an initial attempt at machine learning approaches for processing dictionary definitions, we formulate the synonym extraction task as a classification problem: each word in a piece of definition text is a decision point, and a maximum entropy (MaxEnt) classifier is employed to decide whether or not it is a synonym of the corresponding definiendum.

The training data is based on 186,954 definition items (pairs of definiendum with corresponding definientia) in the *Macquarie Dictionary*. After POS-tagging, any word in a given definition text is labeled as a synonym of the definiendum if the word is (1) of the same POS as the definiendum, and (2) in the same WordNet synset as the definiendum. This labeled data would be partitioned and serve as both the training data for MaxEnt and the gold standard for the synonym labeling task in Section 3.3.

We choose the *opennlp.maxent* implementation of the classifier with generalized iterative scaling (GIS) capacity[6]. We use lexical features (e.g., previous, current, and next word), unigram POS features (e.g., previous, current, and next POS), and bigram POS features (e.g., previous and next POS bigrams). In addition, another group of features describes the position of each decision point by an integer counter starting from 1 to the length of a definition text. In order to capture the critical separators discussed in PbE (e.g., semicolons), a second position counter is also included, which resets to 1 whenever encountering any separators. In the definition of ***abbreviation****: reduction in length; abridgment.*, for example, the first counter assigns integers 1 to 6 to all definientia (including puncuations ";" and "."), whereas the second counter assigns 1 to 4 to definientia up to the semicolon but 1 and 2 to *abridgment* and the period.

Note that, in order to make fair comparisons with IIE and PbE, it is necessary to incorporate the dictionary graph into

[5] Or any of the words from the first round of PbE, which are usually highly synonymous to the target word

[6] Available at `http://sourceforge.net/projects/maxent/`.

MaxEnt method. Specifically, given a target word t, after extracting synonyms from its own definientia, we again go through other words' definitions in the dictionary; if t appears in the definition of another word w and is classified as a synonym, then w is taken as a synonym of t.

2.4 Interpretation of the Methods in Terms of the Dictionary Graph

So far in our discussion, the dictionary graph has been assumed to be *undirected*. For IIE, if we are to take the graph as *directed* (with edges pointing from definienda to definientia), then the in-neighbors of a target word are those related by an inverted index, and the out-neighbors are simply all its definientia. We will see how these two types of relatedness perform differently in Section 3.

In contrast, PbE and MaxEnt make fine distinctions about which part of the definitions to relate a target word to. For out-neighbors (words in the definientia of the target word), PbE and MaxEnt pick out words following specific patterns (either regular expression patterns or, implicitly, patterns learned by a classifier), as opposed to IIE which takes all definientia indiscriminately; for in-neighbors, IIE relates them all regardless of how or where the target word appears in other words' definitions, while PbE and MaxEnt, again, follow their respective patterns.

3 Evaluation

3.1 Solving TOEFL Synonym Questions

Originally introduced by Landauer and Dumais (1997), TOEFL synonym questions have gained much popularity in NLP studies as a task-driven evaluation for synonymy or semantic relatedness. The commonly used data set contains 80 questions, on which Jarmasz and Szpakowicz (2004) evaluated nine semantic similarity methods, not including a later work by Turney et al. (2003) with an accuracy of 97.5% — the highest in all reported results so far.

The popularity of this experiment partly resides in its straightforward setup and easy interpretation of results. Each question consists of one question word and four choices, one of which is a synonym to the questions word and thus, the correct answer. For a given question, we use the three extraction methods to extract synonyms for each of the five words (question word and the four choices), followed by computing the cosine similarity between the synonym set of the question word with those of the choices. The choice with the highest-scoring synonym set is proposed as the correct answer.

Ideally, due to the transitivity of synonymy, if two words are synonymous themselves, they would have a number of synonyms in common, which in turn would give a better score in the TOEFL synonym questions. Two practical issues, however, proved to be adversarial to this assumption. Firstly, finding the right answer does not necessarily depend on synonymy; relatedness, for example, is also transitive by nature. In fact, if overlapping is the only concern, one can even use sets of antonyms for finding the synonymous choice, since synonymous words share antonyms as well as synonyms. Fortunately in TOEFL synonym questions, the choices are either synonymous or unrelated to a question word, and thus, such considerations will not harm the performance on solving the questions.

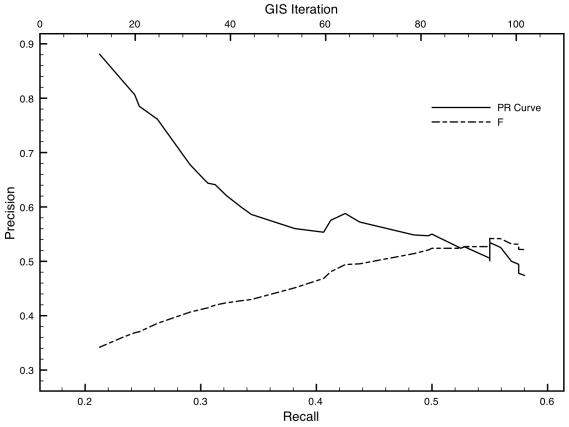

Fig. 2: *Precision-recall curve and F with respect to MaxEnt GIS training iterations (ranging from 10 to 100)*

Secondly, as mentioned earlier, if the target word is infrequent, it will have a low in-degree in the dictionary graph, and this is especially the case for TOEFL synonym questions, which tend to test the participants on a more-challenging vocabulary of relatively low-frequency words. Consequently, as it turned out in the experiment, there are often cases where the extracted set of words for the question word does not overlap with any of those of the choices. The notion of *recall* is thus borrowed to denote the percentage of questions that can be solved without such ties.

Another feature specific to the TOEFL synonym question task is lemmatization, since many of the words in these questions are inflected. As we will see shortly from the results, using these inflected words as target words for extraction gives drastically different performance from using their corresponding lemmata.

Table 1 shows the performance of various methods on the TOEFL synonym questions task. The result is reported in terms of precision, recall and F; accuracy is also included since all other published results are reported in terms of accuracy. For comparison, we separate the extraction result of IIE into in-neighbor-only, out-neighbor-only, and a combination of the two (see Section 2.4 for their differences).

The two variants of IIE with only in- or out-neighbors (IIE_{in} or IIE_{out}) perform more or less the same in terms of F, but are complementary in precision and recall. Out-neighbors are definition texts with many stop words, which are helpful in overlapping and hence the high recall (fewer ties), whereas the low frequency of the target words in TOEFL tests results in fewer in-neighbors, and thus fewer chances of overlapping and more ties.

When using the lemmatized words as target words, there is a 25% increase in recall; the best result for IIE is achieved when it is combined with lemmatization, resulting in an accuracy of 85% — better than any lexicon-based method reported in the *ACL wiki*[7]. When definition texts are combined with PbE results, precision increases by an additional 3.4 percentage points, giving the best accuracy of 88.3% on this task.

In contrast, although it is a more-sophisticated model, MaxEnt fails to perform as well; neither precision nor re-

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

Table 1: *Performance on solving TOEFL synonym questions.* IIE_{in} *and* IIE_{out} *denote variants of* IIE *with in-neighbors only and with out-neighbors only, respectively;* IIE *without subscripts corresponds to the original* IIE *method (with both in- and out-neighbors).*

	IIE_{in}	IIE_{out}	IIE_{in} + Lemma	IIE+Lemma	PbE+Lemma	PbE+DefText	MaxEnt+Lemma
Precision	**100.0**	51.3	93.8	87.2	93.6	90.6	55.0
Recall	50.0	**97.5**	75.0	**97.5**	77.5	**97.5**	54.6
F	66.7	67.2	83.4	92.1	84.8	**93.9**	54.8
Accuracy	50.0	50.0	70.4	85.0	72.5	**88.3**	30.0

call is outstanding, with an accuracy of 30% — slightly better than the simplest baseline of random guessing (25% in accuracy). In general, the number of training iterations i in the GIS algorithm has a positive correlation with the average number of words $\bar{n}$ extracted from each definition (Figure 2): $\bar{n}$ is less than one when $i = 10$, resulting in a recall as low as 15%; the best result is achieved at 80 iterations. Error analysis reveals that the words proposed by MaxEnt are far from synonymous to their corresponding definienda — partly due to the raw quality of training data as discussed in Section 3.3 below.

Performance does not improve significantly when incorporating the dictionary graph into MaxEnt (only about 1% increase in F, not reported in Table 1).

3.2 Comparing with Existing Thesauri

In addition to the indirect, task-driven evaluation by solving TOEFL synonym questions, we set up a second experiment in which the extracted synonyms will be directly compared with existing thesauri. The goal is to evaluate the degree of synonymy among the extracted words.

In order to compare with published results, we try to set up this experiment as close to that of Wu and Zhou (2003) as possible. The thesaurus of choice is artificially constructed, combining WordNet and *Rogets II: The New Thesaurus*[8]; target words are chosen from the *Wall Street Journal* according to different POS (nouns, adjectives, and verbs) and frequency (high, medium, and low). For each target word and each extraction method, there will be two sets of synonyms: one extracted by a given extraction method, the other from the combined thesaurus. The goal is to see how the automatically extracted set correlates with the one from the thesaurus.

With all the different POS and frequencies, the scores reported in precision, recall, and F for all three extraction methods populated a table of over 100 cells[9]. Here, we only focus on comparisons between IIE and PbE , and how their performance varies according to target word frequency (Figures 3 and 4). We also compare all three methods with published results (Figure 5).

Since POS is not of primary interest here, we average the results across the three different POS. Figure 3 shows how IIE compares with PbE across different target word frequencies. On average, PbE has slightly better precision and drastically better recall, resulting in F scores approximately 3–5 times as high as those of IIE. The performance of IIE is apparently "spiked" at medium target word frequency, conforming to our previous hypothesis that IIE would underperform when the target word frequency is too low or too high. In contrast, PbE exhibits "smoother" performance especially in precision and F score[10].

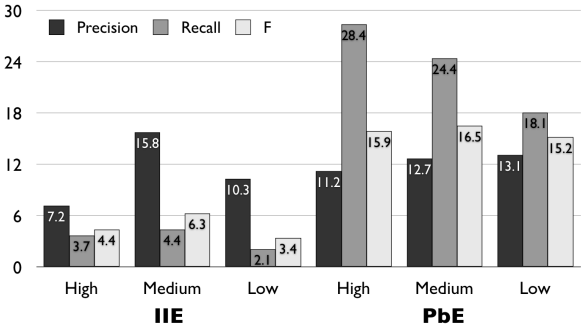

Fig. 3: *Inverted index extraction versus pattern-based extraction when compared with existing thesauri.* High*,* Medium*, and* Low *refer to different frequencies of target words in the* Wall Street Journal*.*

Precision of PbE increases as target word frequency decreases. We speculate that this is because the degree of polysemy of a word is approximately in proportion to its frequency; high-frequency words, being more polysemous, would have more chances of "digressing" to various branches of different senses; they also tend to appear in many different words' definitions under different senses. This is especially true when transitivity of synonymy is applied with no constraints. We will show shortly how transitive closure on the dictionary graph helps alleviate this problem.

The drop of recall in PbE with respect to frequency can be explained by different in- and out-degree of target words of different frequencies. Words of higher frequency would not only have a higher out-degree (due to their polysemy), but also a higher in-degree since they are more likely to appear in other words' definitions. In contrast, low-frequency words would have fewer senses and thus smaller numbers of definitions; if they are too infrequent to appear in other words' definitions, then these few definitions of their own would be the only source for synonyms, which would, not surprisingly, result in lower recall.

Figure 4 shows the improvement in PbE performance by finding transitive closure as mentioned in Section 2.2.2. Recall drops to about half of the original values after using transitive closure (denoted PbE_tc in the graph), but meanwhile precision is more than tripled in all frequencies. It is interesting to observe how precision responds differently to frequency change before and after using transitive closure:

[8] Available at `http://www.bartleby.com/62/`

[9] Available at `http://www.cs.toronto.edu/~tong/syn_ex/combo_thesaurus.pdf`

[10] Even recall, which seemingly drops drastically as frequency decreases, is still smoother than that of IIE if drawn at equal scale.

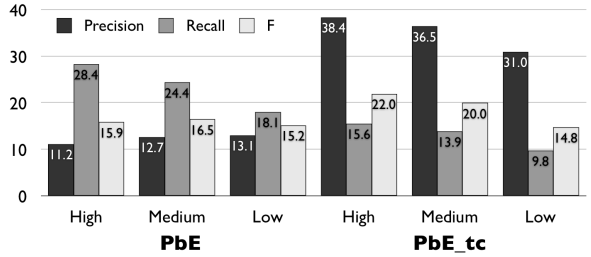

Fig. 4: *Performance before and after using transitive closure on pattern-based extraction (denoted* PbE *and* PbE_tc*, respectively).* High*,* Medium*, and* Low *refer to different frequencies of target words in the* Wall Street Journal.

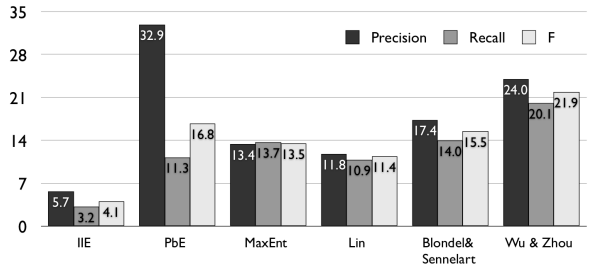

Fig. 5: *Comparing with published results on the combined thesaurus experiment*

without transitive closure, precision increases as frequency decreases, while after transitive closure is introduced, it varies in the opposite direction. This indicates that using transitive closure is most helpful for high-frequency target words. This is, again, due to their polysemy and better chances of "digression", and thus, transitive closure indeed helps to effectively eliminate false positives introduced by such digressions; low-frequency words would already have relatively better precision due to their having fewer number of senses, and transitive closure appears less helpful in this case.

Figure 5 shows how our methods compare with other published results. IIE is outperformed by all other methods by large margins. PbE has the best precision (32.9%) but falls behind that of Wu and Zhou (2003) in terms of F due to low recall. MaxEnt has better recall than both IIE and PbE, but F score is not as good as that of PbE. Results of Blondel and Senellart (2002) is included as an example of dictionary-based method for comparison, and Lin (1998) as an example of corpus-based approach[11]. Wu and Zhou (2003) combines the methods of Blondel and Senellart (2002) and Lin (1998), as well as a novel method using bilingual resources, achieving the best result among all methods being compared here.

3.3 Definition Text Labeling

Recall that the MaxEnt model labels synonyms in a piece of definition text for a given target word; in fact, PbE and IIE could also be viewed as processes of labeling synonyms in definition texts in more or less the same way. The basic

[11] Results of both Blondel and Senellart (2002) and Lin (1998) are reported by Wu and Zhou (2003).

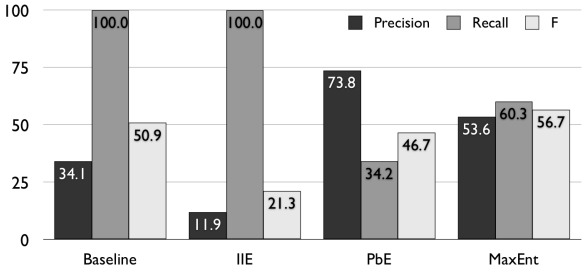

Fig. 6: *Performance on synonym labeling in definitions*

idea of this third experiment is to see how well each method performs in such a labeling task.

Note that in terms of synonym extraction, the former two experiments (Section 3.1 and 3.2) are the main approaches for evaluating the extraction quality of various methods; this section, in contrast, stresses more the nature of the training data for the MaxEnt model.

The data is prepared in the same way as described in Section 2.3; it does not necessitate any human labeling, though at the cost of the quality of synonym labels (to be discussed at the end of this section).

The labeling criteria for the three methods follow the discussion in Section 2.4: IIE takes all definientia as synonyms, while PbE takes only those following certain prespecified patterns. MaxEnt makes predictions for each defining word based on its training. We also introduce a baseline that chooses a defining word as a synonym if it shares the same POS as that of the definiendum.

The results are presented in Figure 6. The baseline and IIE both have 100% recall according to the experiment setup. IIE and PbE are both outperformed by the baseline. PbE has the highest precision and meanwhile, the lowest recall due to its dependence on specific patterns.

Due to the low quality of the training data, MaxEnt did not perform as well as expected. POS tags have many discrepancies, partly because the tagger is not trained on definition texts. On the other hand, using WordNet to create the gold standard in synonym labels also appears to be error-prone. For example, in the definition of ability (*power or capacity to do or act...*), *power* is labeled as a synonym of *ability* while *capacity* is not, since it is not in the same synset as that of *ability*. There are also cases where words in insignificant positions within the definition text happen to be in the same synset as that of the definiendum. All such cases will eventually confuse the learning process of MaxEnt.

4 Conclusions and Future Work

We proposed three methods for extracting synonyms from dictionary definition texts: by building an inverted index on the dictionary, by matching and bootstrapping regular expression patterns that synonyms tend to follow, and by developing and training a maximum entropy classifier. Their performance was evaluated in three experiments: by solving TOEFL synonym questions, by comparing against existing thesauri, and by labeling synonyms in definitions.

Our experiments show that simple extraction schemes perform surprisingly well on solving TOEFL synonym questions; IIE scores 85% in accuracy, and PbE performs

even better at 88.3% — almost 10% higher than that of the best reported lexicon-based method.

Nonetheless, when compared with existing thesauri, the quality of the extracted synonyms is not as satisfactory. In addition, the results from this comparison do not correlate well with those of the TOEFL synonym question task: simple extraction schemes, such as IIE, can perform well on the TOEFL task while failing badly in the comparison experiment, whereas on the other hand, the advantage of PbE and MaxEnt is not fully reflected in the TOEFL task. This leads to the question of whether the TOEFL task, though given the name of *synonym* questions, is indeed indicative of strict synonymy.

Freitag et al. (2005) generated over 23,000 questions through an automated process to compensate for the small number of questions available in the original TOEFL synonym questions data set; although the number of questions is important, it would also be interesting to devise a set of questions that include related but not synonymous words as decoys among the choices, in hope of better evaluating the degree of strict synonymy in the extracted word sets.

As claimed earlier, the maximum entropy model is developed as an initial step towards a machine learning treatment of definition text mining; it would be interesting to employ other classifiers in the future and compare their performance.

Finally, an interesting observation on the extracted words reveals that, due to the Australian provenience of the *Macquarie Dictionary*, all extraction methods have generated some synonyms unique to Australian English or culture (such as *toilet-dunny*). This phenomenon provided evidence for the adaptability of dictionary-based methods in different domains or cultures.

Acknowledgements

This study is supported by the Natural Sciences and Engineering Research Council of Canada. Discussions with Gerald Penn helped shaping an important part of this paper, and we are sincerely grateful to all anonymous reviewers for their insightful and generous comments.

References

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

V.D. Blondel and P.P. Senellart. Automatic extraction of synonyms in a dictionary. *Proc. of the SIAM Workshop on Text Mining*, 2002.

S. Brin, L. Page, R. Motwami, and T. Winograd. The PageRank citation ranking: bringing order to the web. In *Proceedings of ASIS98*, pages 161–172, 1998.

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.

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

C. Fellbaum et al. *WordNet: An electronic lexical database*. MIT press Cambridge, MA, 1998.

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.

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.

M. Jarmasz and S. Szpakowicz. Roget's Thesaurus and Semantic Similarity1. *Recent Advances in Natural Language Processing III: Selected Papers from RANLP 2003*, page 111, 2004.

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.

D. Lin. Automatic retrieval and clustering of similar words. *Proceedings of the 17th International Conference on Computational Linguistics*, pages 768–774, 1998.

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.

R. Reichert, J. Olney, and J. Paris. *Two Dictionary Transcripts and Programs for Processing Them – The Encoding Scheme, Parsent and Conix.*, volume 1. DTIC Research Report AD0691098, 1969.

P.D. Turney, M.L. Littman, J. Bigham, and V. Shnayder. Combining independent modules to solve multiple-choice synonym and analogy problems. *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP-03)*, pages 482–489, 2003.

H. Wu and M. Zhou. Optimizing synonym extraction using monolingual and bilingual resources. *Proceedings of the Second International Workshop on Paraphrasing*, pages 72–79, 2003.