Relevance-Ranked Domain-Specific Synonym Discovery

Andrew Yates, Nazli Goharian, and Ophir Frieder

Information Retrieval Lab, Georgetown University
{andrew,nazli,ophir}@ir.cs.georgetown.edu

Abstract. Interest in domain-specific search is growing rapidly, creating a need for domain-specific synonym discovery. The best-performing methods for this task rely on query logs and are thus difficult to use in many circumstances. We propose a method for domain-specific synonym discovery that requires only a domain-specific corpus. Our method substantially outperforms previously proposed methods in realistic evaluations. Due to the difficulty of identifying pairs of synonyms from among a large number of terms, methods have traditionally been evaluated by their ability to choose a target term's synonym from a small set of candidate terms. We generalize this evaluation by evaluating methods' performance when required to choose a target term's synonym from progressively larger sets of candidate terms. We approach synonym discovery as a ranking problem and evaluate the methods' ability to rank a target term's candidate synonyms. Our results illustrate that while our proposed method substantially outperforms existing methods, synonym discovery is still a difficult task to automate and is best coupled with a human moderator.

Keywords: Synonym discovery, thesaurus construction, domain-specific search

1 Introduction

Interest in domain-specific search has grown over the past few years. Researchers are increasingly investigating how to best search medical documents [7, 14, 16], legal documents [10, 11, 19], and patents [2, 21]. With the growing interest in domain-specific search, there is an unmet need for domain-specific synonym discovery. Domain-independent synonyms can be easily identified with resources such as thesauri, but domain-specific variants of such resources are often less common and less complete. Worse, synonyms can even be corpus-specific or specific to a subdomain within a given domain. For example, in the legal or e-discovery domain, an entity subject to e-discovery may use its own internal terms and acronyms that cannot be found in any thesaurus. In the medical domain, whether or not two terms are synonyms can depend entirely on the use case. For example, a system for detecting drug side effects might treat "left arm pain" as a synonym of "arm pain" because the arm pain is the relevant part. On the other hand, "left arm pain" would not be synonymous with "arm pain" in an electronic health record belonging to a patient who had injured her left arm.

Furthermore, domain-specific document collections (e.g., e-discovery or medical) are often significantly smaller than the collections that domain-independent synonym discovery is commonly performed on (e.g., the Web). We present a domain-specific synonym discovery method that can be used with domain-specific document collections. We evaluate our method on a focused collection consisting of 400,000 forum posts. Our results show that our method can be used to produce ranked lists that significantly reduce the effort of a human editor.

The best-performing synonym discovery methods require external information that is difficult to obtain, such as query logs [33] or documents translated into multiple languages [12, 25]. Other types of synonym discovery methods (e.g., [31, 32]) have commonly been evaluated using synonym questions from TOEFL (Test Of English as a Foreign Language), in which the participant is given a target word (e.g., "disagree") and asked to identify the word's synonym from among four choices (e.g., "coincide", "disparage", "dissent", and "deviate"). While this task presents an interesting problem to solve, this type of evaluation is not necessarily applicable to the more general task of discovering synonyms from among the many terms (n candidates) present in a large collection of documents. We address this concern by evaluating our method's and other methods' performance when used to answer domain-specific TOEFL-style questions with progressively larger numbers of incorrect choices (i.e., from 3 to 1,000 incorrect choices). While our proposed method performs substantially better than strong existing methods, neither our method nor our baselines are able to answer a majority of the questions correctly when presented with hundreds or thousands of incorrect choices. Given the difficulty of choosing a target term's synonym from among 1,000 candidates, we approach domain-specific synonym discovery as a ranking problem in which a human editor searches for potential synonyms of a term and manually evaluates the ranked list of results. To evaluate the usefulness of this approach, we use our method and several strong existing methods to rank lists of potential synonyms. Our method substantially outperforms existing methods and our results are promising, suggesting that, for the time being, domain-specific synonym discovery is best approached as a human-moderated relevance-ranking task.

Our contributions are (1) a new synonym discovery method that outperforms strong existing approaches (our baselines); (2) an evaluation of how well our method and others' methods perform on the TOEFL-style evaluations when faced with an increasing number of synonym candidates; (3) an evaluation of how well our methods and others' methods perform when used to rank a target term's synonyms; our method places 50% of a target term's synonym in the top 5% of results, whereas other approaches place 50% of a target term's synonyms in the top 40%.

2 Related Work

A variety of methods have been applied to the domain-independent synonym identification problem. Despite the limited comparisons of these methodologies, the best-performing methods are reported to use query logs or parallel corpora. We describe the existing methodologies and differentiate our approach.

Distributional Similarity. Much related work discovers synonyms by computing the similarity of the contexts that terms appear in; this is known as distributional similarity [26]. The intuition is that synonyms are used in similar ways and thus are surrounded by similar words. In [31], Terra and Clarke compare the abilities of various statistical similarity measures to detect synonyms when used along with term co-occurrence information. Terra and Clarke define a term's context as either the term windows in which the term appears or the documents in which the term appears. They use questions from TOEFL (Test Of English as a Foreign Language) to evaluate the measures' abilities to choose a target word's synonym from among four candidates. We use Terra and Clarke's method as one of our baselines (baseline 1: Terra & Clark). In [8], Chen et al. identify synonyms by considering both the conditional probability of one term's context given the other term's context and co-occurrences of the terms, but perform limited evaluation. In [27], Rybinski et al. find frequent term sets and use the term sets' support to find terms which occur in similar contexts. This approach has a similar outcome to other approaches that use distributional similarity, but the problem is formulated in terms of terms sets and support.

Distributional similarity has also been used to detect other types of relationships among words, such as hyponymy and hypernymy, as they also tend to occur in similar contexts. In [28], Sahlgren and Karlgren find terms related to a target concept (e.g., "criticize" and "suggest" for the concept "recommend") with random indexing [18], a method which represents terms as low-dimensional context vectors. We incorporate random indexing as one of our model's features and evaluate the feature's performance in our feature analysis. Brody and Lapata use distributional similarity to perform word sense disambiguation [5] using a classifier with features such as *n*-grams, part of speech tags, dependency relations, and Lin's similarity measure [20], which computes the similarity between two words based on the dependency relations they appear in. We incorporate Lin's similarity measure as a feature and derive features based on n-grams and part-of-speech n-grams. Strzalkowski proposes a term similarity measure based on shared contexts [30]. Carrell and Baldwin [6] use the contexts a target term appears in to identify variant spellings of a target term in medical text. Pantel et al. use distributional similarity to find terms belonging to the same set (i.e., terms which share a common hypernym) [24] by representing each term as a vector of surrounding noun phrases and computing the cosine distance between term vectors.

Lexico-syntactic Patterns. In [22], McCrae and Collier represent terms by vectors of the patterns [15] they occur in and use a classifier to judge whether term pairs are synonyms. Similarly, Hagiwara [13] uses features derived from patterns and distributional similarity to find synonyms. Hagiwara extracts dependency relations from documents (e.g., X is a direct object of Y) and use them as a term's context. Hagiwara finds that the features derived from distributional similarity are sufficient, because there is no significant change in precision or recall when adding features derived from patterns. Their analysis is logical given that lexico-syntactic patterns and distributional similarity are both concerned with the terms surrounding a target term. We use Hagiwara's method as another one of our baselines (baseline 2: Hagiwara).

Tags. Clements et al. [9] observe that in social tagging systems different user groups sometimes apply different, yet synonymous tags. They identify synonymous tags based on overlap among users/items. Other tag similarity work includes [29], which identifies similar tags that represent a "base tag". Tag-based approaches rely on the properties of tags, thus they are not applicable to domains in which tags are not used. For this reason we do not compare our method with tag-based approaches.

Web Search. Turney [32] identifies synonyms by considering the co-occurrence frequency of a term and its candidate synonym in Web search results. This method is evaluated on the same TOEFL dataset used by Terra and Clarke [31]; Terra and Clarke's method performs better. Similarly, other approaches [1, 3] rely on obtaining co-occurrence frequencies for terms from a Web search engine. We do not compare with Web search-based methods as they rely on a general corpus (the Web), whereas our task is to discover domain-specific synonyms in a domain-specific corpus.

Word Alignment. Plas [25] and Grigonytė et al. [12] observe that English synonyms may be translated to similar words in another language; they use word alignment between English and non-English versions of a document to identify synonyms within a corpus. Wei et al. [33] use word alignment between queries to identify synonyms. Similarly, word alignment can be coupled with machine translation to identify synonyms by translating text into a second language and then back into the original language (e.g., [23]). While word alignment methods have been shown to perform well, their applicability is limited due to requiring either query logs or parallel corpora. Due to this limitation, we do not use any word alignment method as a baseline; we are interested in synonym discovery methods that do not require difficult-to-obtain external data.

3 Methodology

We compare our approach against three baselines: Terra and Clarke's method [31], Hagiwara's SVM method [13], and a variant of Hagiwara's method.

3.1 Baseline 1: Terra and Clarke

In [31], Terra and Clarke evaluate how well many statistical similarity measures identify synonyms. We use the similarity measure that they found to perform best, point-wise mutual information (PMI), as one of our baselines. The maximum likelihood estimates used by PMI depend on how term co-occurrences are defined. Terra and Clarke propose two approaches: a window approach, in which two terms co-occur when they are present in the same n-term sliding window, and a document approach, in which two terms co-occur when they are present in the same document. We empirically determined that a 16-term sliding window performed best on our dataset.

With this approach the synonym of a term w_i is the term w_j that maximizes $PMI(w_i, w_j)$. Similarly, a ranked list of the synonym candidates for a term w_i can be obtained using this approach by using $PMI(w_i, w_j)$ as the ranking function.

3.2 Baseline 2: Hagiwara (SVM)

Hagiwara [13] proposes a synonym identification method based on point-wise total correlation (PTC) between two terms (or phrases treated as single terms) w_i and w_j

and a context c_k in which they both appear. Hagiwara uses syntax to define context. The RASP parser [4] is used to extract term dependency relations from documents in the corpus. A term's contexts are the (*modifier term*, *relation type)* tuples from the relations in which the term appears as a head word.

Hagiwara takes a supervised approach. Each pair of terms (w_i, w_j) is represented by a feature vector containing the terms' point-wise total correlations for each context as features. Features for contexts not shared by w_i and w_j have a value of 0. That is, $vector_{w_i,w_j} = \langle PTC(w_i, w_j, c_1), \dots, PTC(w_i, w_j, c_n)\rangle$. We prune features using the same criteria as Hagiwara and identify synonyms by classifying each word pair as synonymous or not synonymous using SVM. We modified this approach to rank synonym candidates by ranking the results based on SVM's decision function's value.

3.3 Baseline 3: Hagiwara (Improved)

We modified Hagiwara's SVM approach to create an unsupervised approach based on similar ideas. The contexts and maximum likelihood estimates are the same as in Hagiwara's approach (described in section 3.2). Instead of creating a vector for each pair of terms (w_i, w_j), we created a vector for each term w_i and computed the similarity between these vectors. The vector for a term w_i is composed of the PMI measures between the term w_i and each context c_k. That is, $vector_{w_i} = \langle PMI(w_i, c_1), PMI(w_i, c_2), \dots, PMI(w_i, c_n)\rangle$. The similarity between w_i and w_j is computed as the cosine similarity between their two vectors. Similarly, we rank synonym candidates for a term w_i by ranking vectors based on their similarity to $vector_{w_i}$.

3.4 Regression

Our approach is a logistic regression on a small set of features. We hypothesize that a supervised approach will outperform statistical synonym identification approaches since it does not rely on any single statistical measure and can instead weight different types of features. While Hagiwara's original method used supervised learning, it only used one type of contextual feature (i.e., point-wise total correlation between two terms and a context). Like Hagiwara, we construct one feature vector for each word pair. In the training set, we give each pair of synonyms a value of (+1) and each pair of words that are not synonyms a value of (-1). To obtain a ranked list of synonym candidates, the probabilities of candidates being synonyms are used as relevance scores. That is, the highest ranked candidates are those that the model gives the highest probability of being a 1.

We also experimented with SVM^{Rank} [17] and SVM, but found that a logistic regression performed similarly or better while taking significantly less time to train.

The features we used are:

1. The number of distinct contexts both w_i and w_j appear in, normalized by the minimum number of contexts either one appears in,

$$shared\_contexts = \frac{C(w_i,\ w_j)}{\min\left(c(w_i),\ c(w_j)\right)}$$

where $c(w_i)$ is the number of distinct contexts w_i appears in and $c(w_i, w_j)$ is the number of distinct contexts both w_i and w_j appear in. According to the distributional hypothesis [26], similar words should appear in the same context more often than dissimilar words do. We use Hagiwara's method as described in section 3.2 for finding contexts.

2. The number of sentences both w_i and w_j appear in, normalized by the minimum number of sentences either one appears in,

$$shared\_sentences = \frac{s(w_i,\ w_j)}{\min\left(s(w_i),\ s(w_j)\right)}$$

where $s(w_i)$ is the number of windows w_i appears in and $s(w_i, w_j)$ is the number of windows both w_i and w_j appear in.

3. The cosine similarity between w_i and w_j as calculated by the Hagiwara (Improved) method, as described in section 3.3. This method weights contexts by their PMI, whereas $shared\_contexts$ weights all contexts equally.
4. The Levenshtein distance between terms w_i and w_j. Our synonym list contains phrases; that is, terms may contain multiple words (e.g., "sore_throat"). We hypothesize that this feature will be useful because synonymous phrases may share common terms (e.g., "aching_throat" and "sore_throat").
5. The probability of the target term w_i appearing in an n-gram given that the candidate term w_j appears in the n-gram. We use all n-grams of size 3 that appear in our dataset (e.g., "have|a|headache") and replace the candidate and target terms with X (e.g., "have|a|X").

$$ngram\_pr = \Pr(w_i\ appears\ as\ X \mid w_j\ appears\ as\ X) = \frac{Pr\left(w_i\ and\ w_j\ appear\ as\ X\right)}{Pr\left(w_j\ appears\ as\ X\right)}$$

6. The probability of the target term w_i appearing in a part-of-speech n-gram given that the candidate term w_j appears in the part-of-speech (POS) n-gram. As with *ngram_pr*, we use n-grams of size 3. To construct POS n-grams, we replace the candidate and target terms with X as before and replace each term in the n-gram with its POS (e.g., "have|a|X" becomes "VBP|DT|X").

$$posng\_pr = \Pr(w_i\ appears\ as\ X \mid w_j\ appears\ as\ X) = \frac{Pr\left(w_i\ and\ w_j\ appear\ as\ X\right)}{Pr\left(w_j\ appears\ as\ X\right)}$$

7. The similarity between terms w_i and w_j as computed by Lin's information-theoretic term similarity measure (lin_sim) as described in [20]; this measure is computed using the dependency relations that terms w_i and w_j appear in.
8. The cosine distance between the vector for term w_i and the vector for term w_j as obtained using random indexing. We used the SemanticVectors (https://code.google.com/p/semanticvectors/) implementation of random indexing with the default parameters.

Features 5-7 (*ngram_pr*, *posng_pr*, and *lin_sim)* were inspired by features used in Brody and Lapata's effort on word sense disambiguation [5]; *random_indexing* was shown by Sahlgren and Karlgren to perform well at identifying related terms in [28]. We explore the utility of each feature in section 4.4.

4 Experiments

We describe our ground truth and corpus in section 4.1. In section 4.2 we evaluate the quality of our approach and various baseline methods using a more realistic variant of the TOEFL evaluation methodology commonly used in previous efforts. We approach synonym discovery problem as a ranking problem in section 4.3 and evaluate how well our approach and the baseline methods rank a target term's synonyms. Finally, we examine the impact of each feature used by our method in section 4.4.

4.1 Dataset

We focus on the medical side-effect domain in our evaluation. To evaluate our methodology and compare with existing strong approaches (i.e., our baselines), we used a corpus of medical forum posts and the MedSyn synonym list [34] as our ground truth, which contains synonyms in the medical side-effect domain. A domain-specific thesaurus is required to train the synonym discovery methods for a given domain. We removed synonyms from the list that do not occur or occur only once in our corpus because it is impossible for any of the methods to detect them. We also removed terms from the list that had no synonyms in our corpus. This left us with 1,791 synonyms, which were split into a training set (291 pairs) which was used to tune our methods, and a testing set (1,500 pairs), which was used to perform our evaluations. On average, each term in the list had 2.8 synonyms ($\sigma = 1.4$). The maximum number of synonyms per term was 11 and the minimum number was 2. Of the 1,791 synonyms that we kept, 67% of the synonyms were phrases treated as a single term (e.g., "joint_pain") and the remaining 33% were single terms (e.g., "arthralgia").

We created questions (target terms) similar to those used in TOEFL from terms in the synonym list by choosing a target term as the question (e.g., "joint pain") and choosing a synonym (e.g., "arthralgia") and non-synonymous terms (e.g., "headache", "arthritis", and "arm pain") as choices (synonym candidates). The methods' task is to identify the correct synonym from among the choices (synonym candidates) given the target term. In the general TOEFL-style evaluation (section 4.2), each question has one correct choice and *n* incorrect choices. In the relevance ranking evaluation (section 4.3), each question *i* has m_i correct choices and *n* incorrect choices, where m_i is the number of synonyms that question *i* has in the synonym list.

Our corpus was built from a crawl of 400,000 forum posts made to the Breastcancer.org discussion boards[1] and the FORCE breast cancer message boards[2]. Both Websites divide their discussion boards into topics. In keeping with our goal of identifying domain-specific synonyms, we crawled only those topics related to general

[1] http://community.breastcancer.org/
[2] http://www.facingourrisk.org/messageboard/index.php

discussion or to side-effects. A complete list of the pages crawled is available at *http://ir.cs.georgetown.edu/data/medposts.txt*. While this dataset is focused on the medical side-effect domain, our methods do not take advantage of any medical domain knowledge and could be applied to find synonyms within any domain. We stemmed both the synonym list and corpus with the Porter stemmer. When tokenizing our corpus and synonym list, we transformed each multi-word term in the synonym list into a single term (e.g., "joint pain" became "joint_pain"). We define synonyms as equivalent terms, including spelling variations. Synonymous phrases may be the same except for one additional word (e.g., "arm_pain" and "left_arm_pain"). We do not include separate entries in our synonym list for every morphological variant of a term, however, because the synonym list is stemmed.

4.2 General TOEFL-Style Evaluation

In related research efforts, TOEFL (Test Of English as a Foreign Language) questions have been most commonly used to measure synonym identification accuracy. Succinctly, a TOEFL question consists of a target term and four synonym candidates. The task is to identify which one of the four candidates is a synonym of the target term. To create a more realistic TOEFL-style evaluation in which methods are faced with more than four choices (synonym candidates), we created TOEFL-style questions that consisted of one target word, one correct choice, and n incorrect choices (analogous to the TOEFL evaluation when n=3). We let n range from 3 to 138 in multiples of 15 (3, 18, 33, 48, …, 138) and from 150 to 1050 in multiples of 100 (150, 250, …, 1050) so that we could observe the methods' performance while the questions were gradually made harder (multiples of 15) and while the questions became harder more rapidly (multiples of 100). We used five-fold cross-validation with the supervised methods. As in previous work, we measured the performance in terms of the number of questions answered correctly as the number of incorrect candidates varies (correct@n).

The results for the *general TOEFL-Style evaluation* are shown in Figure 1. We also show the expected performance of a method that randomly selects synonyms (*Random*). *Terra and Clarke*'s method quickly overtakes *Hagiwara (Improved)* as n increases. Our method, Regression, performs substantially better than *Terra & Clarke* for all values of n (about 175% better @33, @150, and @450). The performance of all methods decreases as n increases. *Hagiwara (SVM)* performs the worst among the methods (91% worse than *Regression* @150) and quickly approaches the performance of *Random*. *Hagiwara (Improved)* performs better than *Hagiwara (SVM)*, but it performs much worse than *Regression* and *Terra and Clarke* (85% worse than *Regression* @150). At n=3, which is equivalent to the traditional TOEFL evaluations with one correct choice and three incorrect choices, *Regression* performs 49% better (67% vs. 45%) than the next best method, *Hagiwara (improved)*. If each question's number of correct choices increases to two or three (instead of one), the methods perform similarly and *Regression* continues to substantially outperform the other methods.

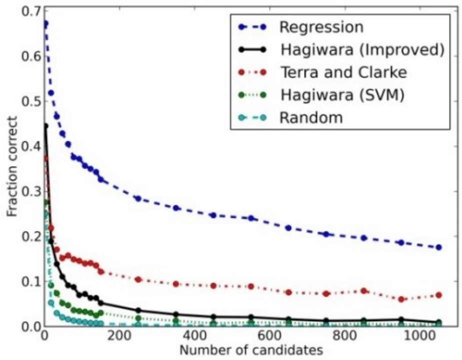

Figure 1: General TOEFL-Style Evaluation

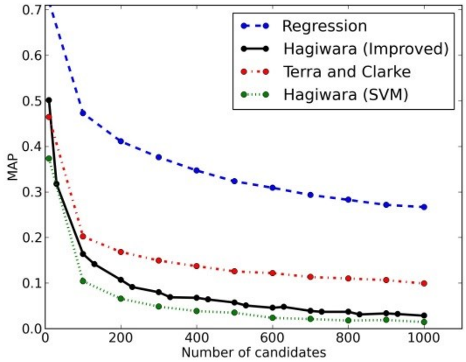

Figure 2: MAP

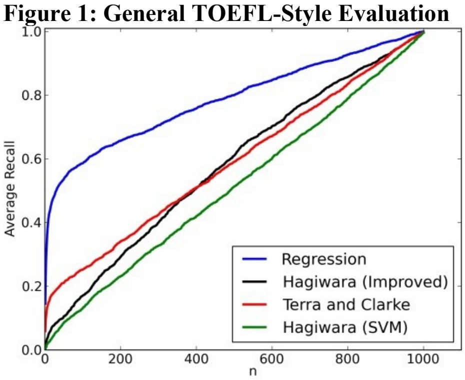

Figure 3: Recall@n

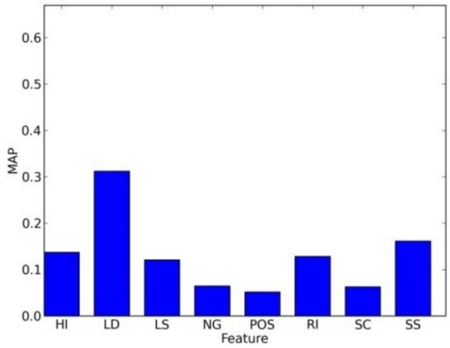

Figure 4: MAP@200 using single features

While *Regression* and *Terra and Clarke* perform much better than the two *Hagiwara* methods, they do not perform well on an absolute scale. When used to find a target term's synonym from among 451 choices (450 incorrect choice and 1 correct choice), *Regression* is only correct 25% of the time; when n=1000, *Regression* is correct only 18% of the time. This is not accurate enough for use as a domain-specific synonym discovery method. In the next section (section 4.3), we propose a solution.

4.3 Relevance Ranking Evaluation

It is clear from the general TOEFL evaluation (section 4.2) that currently-existing methods are incapable of discovering domain-specific synonyms with acceptable accuracy. Given this observation, we propose approaching the problem of domain-specific synonym discovery as a ranking problem, in which a human editor identifies a target term's synonyms by manually reviewing a ranked list of potential synonyms. While this process does require human effort, providing a high quality ranked list significantly reduces the amount of required effort. We evaluate the methods' abilities to produce ranked lists. To do so, each method is given a target term and required to rank the term's synonym candidates. To evaluate this approach, we generated TOEFL-style questions in which each question i has m_i correct choices and n incorrect choices, where m_i is the number of synonyms that question i has in the synonym list. That is, each m_i is fixed for each question i and n grows progressively larger. That is, there is no fixed number of correct choices as in the general TOEFL evaluation where there was only one correct choice. Instead, the number of correct choices for each question is the number of synonyms that actually exist; this is more realistic than fixing the number of correct choices in the evaluation. We started n at 10 and then allowed it to range from 100 to 1,000 in multiples of 100 (10, 100, 200, 300, …, 1000). The quality of the ranked lists produced by each method was measured with Mean Average Precision (MAP). We used five-fold cross-validation with the supervised methods, as we did in our previous evaluations. Each method was modified to produce a ranked list of results as described in the methodology sections

The results are shown in Figure 2. In this evaluation, *Regression* outperforms *Terra and Clarke* for all values of n (57% better @10, 135% better @200, and 170% better @1000). Similarly, *Hagiwara (Improved)* outperforms *Hagiwara (SVM)* for all values of n. As in the general TOEFL evaluation, *Regression* and *Terra and Clarke* perform much better than the *Hagiwara* methods. *Regression*'s MAP remains above 0.40 for $n \leq 200$ and has a MAP of 0.27 at $n = 1000$. This suggests that *Regression* produces a ranked list that a human editor would find useful.

We measured recall@n with 1,000 candidates to explore how useful a human editor would find these ranked lists. The results are shown in Figure 3. As with MAP, *Regression* outperforms the other methods. *Regression* achieves a recall of 0.54 @50, indicating that a human editor could use *Regression*'s ranked lists to find half of a target term's synonyms by looking at only the top 50 of 1,000 results (5%). This is a sharp contrast to the other three methods, which require at least the top 400 of 1,000 results (40%) to be viewed before achieving an equivalent recall; *Regression* performs 157% better than *Terra & Clarke* @50, suggesting that our method can significantly decrease the work performed by a human editor.

4.4 Feature Analysis

We examine *Regression*'s features to determine their contribution to *Regression*'s overall performance. To do so, we analyze the features in the context of the relevance ranking evaluation. We compare the MAP@200, achieved by single features, and feature pairs. We abbreviate the name of each feature as follows: Hagiwara_improved (HI), Levenshtein_distance (LD), Lin_sim (LS), ngram_pr (NG), posng_pr (POS), random_indexing (RI), shared_contexts (SC), and shared_sentences (SS).

The performance of each single feature is shown in Figure 4. LD performs the best, which is surprising given that our corpus was stemmed. We hypothesize that LD's utility both results from synonymous terms that stem to different roots and synonymous phrases that share some terms. SS, RI, LS, and HI follow LD, but achieve MAPs approximately 50% lower than LD's. The features that use dependency relations (HI and LS) perform similar to RI, which uses term co-occurrences. NG, POS, and SC perform the worst. When pairs of features are used, the pairs containing LD perform the best. All of these pairs perform similarly, but LD-SS performs best (25% better than LD alone); it is closely followed by LD-RI. Of the feature pairs that do not contain LD, three pairs that contain SS perform the best (LS-SS, RI-SS, and SS-HI), however, they perform approximately 50% worse than the pairs containing LD. These results mirror those obtained using single features. The performance achieved by the best performing feature combinations (LD and LD-SS) cannot be achieved simply by combining baselines (e.g., HI and RI).

5 Acknowledgements

This work was partially supported by the US National Science Foundation through grant CNS-1204347.

6 Conclusions

We proposed a new regression-based method for synonym discovery that substantially outperforms existing methods when used to rank a target term's synonym candidates. Additionally, our method performs better at a generalization of the TOEFL-style evaluation commonly used in prior work. When used to rank a target term's 100 synonym candidates, our method produces rankings with a MAP of 0.47, a 135% increase over the next-highest method. On average our method places 54% of a target term's synonyms in the top 50 of 1,000 results (5%), whereas other approaches place 50% of a target term's synonyms in the top 400 of 1,000 results (40%). While domain-specific synonym discovery is still a difficult task requiring a human editor, our method significantly decreases the number of terms an editor must review. Our method finds domain-specific synonyms, but the method itself is not domain specific. Future work could investigate the benefits of using a domain-specific method.

7 References

1. Alfonseca, E. et al.: Using context-window overlapping in synonym discovery and ontology extension. RANLP '05. (2005).
2. Azzopardi, L. et al.: Search system requirements of patent analysts. SIGIR '10. (2010).
3. Bollegala, D.: Measuring Semantic Similarity between Words Using Web Search Engines. WWW '07. (2007).
4. Briscoe, T. et al.: The second release of the RASP system. Proceedings of the COLING/ACL on Interactive presentation sessions -. (2006).
5. Brody, S., Lapata, M.: Good Neighbors Make Good Senses: Exploiting Distributional Similarity for Unsupervised WSD. COLING '08. (2008).
6. Carrell, D., Baldwin, D.: PS1-15: A Method for Discovering Variant Spellings of Terms of Interest in Clinical Text. Clin. Med. Res. 8, 3-4, (2010).
7. Cartright, M.-A. et al.: Intentions and attention in exploratory health search. SIGIR '11. p. 65 (2011).
8. Chen, L. et al.: Statistical relationship determination in automatic thesaurus construction. CIKM '05. (2005).
9. Clements, M. et al.: Detecting synonyms in social tagging systems to improve content retrieval. SIGIR '08. (2008).

10. Evans, D.A. et al.: E-discovery. CIKM '08. (2008).
11. Ghosh, K.: Improving e-discovery using information retrieval. SIGIR '12. (2012).
12. Grigonytė, G. et al.: Paraphrase alignment for synonym evidence discovery. COLING '10. (2010).
13. Hagiwara, M.: A Supervised Learning Approach to Automatic Synonym Identification based on Distributional Features. HLT-SRWS '08. (2008).
14. Hanbury, A.: Medical information retrieval. SIGIR '12. (2012).
15. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. COLING '92. p. 539 (1992).
16. Huang, J.X. et al.: Medical search and classification tools for recommendation. SIGIR '10. (2010).
17. Joachims, T.: Optimizing search engines using clickthrough data. KDD'02. (2002).
18. Kanerva, P. et al.: Random indexing of text samples for latent semantic analysis. CogSci '00. (2000).
19. Lewis, D.D.: Information retrieval for e-discovery. SIGIR '10. (2010).
20. Lin, D.: Automatic retrieval and clustering of similar words. ACL/COLING '98. (1998).
21. Lupu, M.: Patent information retrieval. SIGIR '12. (2012).
22. McCrae, J., Collier, N.: Synonym set extraction from the biomedical literature by lexical pattern discovery. BMC Bioinformatics. 9, (2008).
23. Nanba, H. et al.: Automatic Translation of Scholarly Terms into Patent Terms Using Synonym Extraction Techniques. LREC '12. .
24. Pantel, P. et al.: Web-Scale Distributional Similarity and Entity Set Expansion. EMNLP '09. (2009).
25. Plas, L. Van Der: Finding Synonyms Using Automatic Word Alignment and Measures of Distributional Similarity. COLING-ACL '06. (2006).
26. Rubenstein, H., Goodenough, J.B.: Contextual correlates of synonymy. Commun. ACM. 8, 10, 627–633 (1965).
27. Rybinski, H. et al.: Discovering Synonyms Based on Frequent Termsets. RSEISP '07. (2007).
28. Sahlgren, M., Karlgren, J.: Terminology mining in social media. CIKM '09. (2009).
29. Solskinnsbakk, G., Gulla, J.A.: Mining tag similarity in folksonomies. SMUC '11. (2011).
30. Strzalkowski, T.: Building a lexical domain map from text corpora. COLING '94. (1994).
31. Terra, E., Clarke, C.L.A.: Frequency estimates for statistical word similarity measures. HLT-NAACL '03. (2003).
32. Turney, P.D.: Mining the Web for Synonyms : PMI-IR versus LSA on TOEFL PMI-IR. EMCL '01. (2001).
33. Wei, X. et al.: Context sensitive synonym discovery for web search queries. CIKM '09. (2009).
34. Yates, A., Goharian, N.: ADRTrace: Detecting Expected and Unexpected Adverse Drug Reactions from User Reviews on Social Media Sites. ECIR'13. (2013).