Automatic Thesaurus Construction for Cross Generation Corpus

HADAS ZOHAR, CHAYA LIEBESKIND, JONATHAN SCHLER, and IDO DAGAN, Bar-Ilan University, Israel

This article describes methods for semiautomatic thesaurus construction, for a cross generation, cross genre, and cross cultural corpus. Semiautomatic thesaurus construction is a complex task, and applying it on a cross generation corpus brings its own challenges. We used a Jewish juristic corpus containing documents and genres that were written across 2000 years, and contain a mix of different languages, dialects, geographies, and writing styles. We evaluated different first and second order methods, and introduced a special annotation scheme for this problem, which showed that first order methods performed surprisingly well. We found that in our case, improving the coverage is the more difficult task, for this we introduce a new algorithm to increase recall (coverage)—which is applicable to many other problems as well, and demonstrates significant improvement in our corpus.

Categories and Subject Descriptors: H 3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Thesauruses*

General Terms: Algorithms, Languages

Additional Key Words and Phrases: Language model, Hebrew, cultural heritage

ACM Reference Format:

Zohar, H., Liebeskind, C., Schler, J., and Dagan, I. 2013. Automatic thesaurus construction for cross generation corpus. ACM J. Comput. Cult. Herit. 6, 1, Article 4 (March 2013), 19 pages.
DOI: http://dx.doi.org/10.1145/2442080.2442084

1. INTRODUCTION

Automatic Thesaurus construction has been researched for a few decades. In this research we analyze the different methods for semiautomatic construction of a thesaurus for a cross generation and cross genre corpus. For evaluation, we used a Jewish juristic corpus with documents written in different periods and various genres across 2000 years. This is a challenging corpus in many respects: polarity of writing styles, mixture of various languages (Hebrew, Aramaic and Yiddish), massive use of acronyms and more. We describe the different methods and special handling we needed for this special resource type. For the automatic thesaurus creation, we evaluated first and second order methods and found that although second order methods traditionally received more attention in automatic thesaurus construction, first order methods performed surprisingly well in this setting. We also introduce a new algorithm to improve the coverage (recall) that can be useful in other cases as well. We conclude with

Author's address: J. Schler; email: schler@gmail.com.

DOI: http://dx.doi.org/10.1145/2442080.2442084

error analysis and highlight the differences of the various approaches and their advantages for this type of task.

2. BACKGROUND

The literature contains many different definitions of thesaurus. Schutze and Pederson [1997] defined a thesaurus simply as "a mapping from words to other closely related words." A more elaborate definition of a thesaurus was given by Miller [1997], which described a thesaurus as "a lexico-semantical model of a conceptual reality or its consistent, which is expressed in the form of a system of terms and their relations, and offers access via multiple aspects." In this article we will use the notion of target term. A target term is a phrase (or a term) that we use to look for its synonyms and other related terms in the corpus. A set of relations is predefined, and the related terms fulfill one of these relations. Such a set of relations can be general and contains a narrower term (NT), broader term (BT), or related term (RT), or it can be more specific and contain synonym, hyponym, hypernym, cohyponym, or opposite (or any predefined subset).

2.1 Thesaurus Construction Evolution

Thesauri have been incorporated in IR systems for nearly four decades [Lesk 1969; Salton 1971]. During these years researchers proposed various methods to create and use thesauri efficiently. One approach was to use handcrafted thesauri. However, handcrafted general-purpose thesauri have some drawbacks; they are very expensive and time consuming to build and maintain, as well as tending to suffer from problems of inconsistency and limited coverage.

Most of these drawbacks were reduced since the development of WordNet [Fellbaum 1998]. WordNet is a large public domain electronic thesaurus containing nouns, verbs, adjectives, and adverbs grouped into sets of cognitive synonyms (synsets). However, it soon became clear that a large general purpose thesaurus is not specific enough to offer synonyms for words as used in a domain-specific collection. For example, a synonym for *bug* in a general-purpose thesaurus, will probably not match the meaning of *bug* in a set of documents in the domain of computers.

A common approach for automatic thesaurus construction is based on word co-occurrence information. It assumes that a target term and its related terms frequently co-occur [Schuze and Pedersen 1994]. According to this approach, given a target term the co-occurrence of its surrounding terms is weighted using a co-occurrence measure, and the top-k terms are considered as related terms.

Despite the potential of this approach, it has also drawbacks. First, it is difficult to determine the appropriate word window size to consider a co-occurrence. Second, two words are considered as related only if they appear in the same document a certain number of times. For example, *astronaut* and *cosmonaut* are certainly synonyms but there is a negligible chance for them to appear together in same documents.

Another approach suggests taking advantage of linguistic information to extract related terms of a target term. It assumes that similar words are syntactically dependent or share similar head modifiers that indicate syntactic or grammatical relationships [Grefenstette 1994; Lin 1998b; Curran 2003; Weeds et al. 2004]. Applying such an approach requires a shallow or complete parsing of the corpus in order to extract the syntactical information. These various methods succeed in extracting related terms for a thesaurus and report good results when using it in IR applications.

Although the thesauri derived according to this approach don't suffer from the co-occurrence-based thesauri problems, they too had a distinct disadvantage. These methods tend to extract cohyponyms as related terms, as they regularly share the same modifying heads. However, words with similar modifiers are not always good candidates for query expansion, which is the major usage of these thesauri.

For example, *cat* and *dog* may share many syntactical modifiers, yet when you are interested in cats, you don't want to receive information about dogs.

Another approach to extracting related terms using linguistic information is the pattern-based approach [Hearst 1992; Iwanska et al. 2000]. This approach is usually used to extract certain relations, for example, Charniak and Berland [1999] tried to discover the *part-of* relation, and Girfu and Moldovan [2002] *causation* relations. They predefined a set of patterns according to the desired relation. Then, the instantiations of the patterns in the corpus are collected, and using statistical methods the desired related terms are constructed.

The pattern-based approach is characterized by high precision in the sense that the quality of the learned relations is very high. However, it suffers from very low recall due to the fact that patterns are very rare in a real corpus.

Eventually, it was shown that using thesauri of both types interactively performs IR tasks better than using a single type of thesaurus [Mandala et al. 2000 and Perez-Aguera and Araujo 2007]. The underlying idea is that each type of thesaurus has different characteristics. Therefore, their combination can provide a valuable and fruitful lexical resource.

2.2 Main Approaches for Automatic Thesaurus Construction

As we defined in the preceding, the main target in Automatic Thesaurus Construction is to extract for each predefined target term (also called *key term*), a set of synonyms and other related terms. The underlying assumption of most approaches is that words with similar meanings appear in similar contexts. However, the various approaches primarily differ in their definition of *context* and the way they calculate similarity of words. Another aspect that differs from one approach to another is the space they are being applied in. While the *bag-of-words* space (BOW) uses information from a raw corpus, the syntactic space makes use of the syntactic information derived from the corpus using syntactic analysis.

2.2.1 *First-Order Similarities.* The underlying assumption of these methods is that terms that co-occur more than we expect are probably related to each other. Therefore, the context of a target term is a collection of all the words that surround it in all its locations in the corpus.

In the bag-of-words space there are some options to consider "surrounding" words. It may be all the words in the documents in which the target term appears, or paragraph, or sentence, or a fixed word-window-size. In the syntactic space, the context of a term may be the collection of all the terms that are syntactically related to it, or terms that take part in specific types of syntactic relations in the corpus.

The next step after the context is defined is to calculate the similarity of each term in this context to the target term. Different similarity functions were used, such as Pairwise Mutual Information (PMI) [Church and Hanks 1990] Dice [Smadja 1993], Kullback-Liebler Divergence (KLD) [Cover and Thomas 1991]. There is no clear performance benchmark between these functions, and a big variation exists in terms of reported performance numbers. For example, Church and Hanks [1990] report the number of new occurrences (no use of recall and precision metrics), while Han et al. [2011] report the recall score for the first new synonym. The variation is dependent not only on the algorithms used, but also heavily dependent on the corpus it is applied to, and the need for the generated thesaurus.

2.2.2 *Second-Order Similarities.* The methods that use second order similarity interpret the notion of *context* mentioned in the distributional hypothesis in a more elaborated manner. These methods represent terms in a high dimension Vector Space Model (VSM). VSM is an algebraic model for representing terms (and documents) as vectors of a fixed set of features. The set of features is determined in advance, and each vector is represented by the weights of these features.

In the bag-of-words (BOW) space, the features are usually all the terms in the corpus, and the weight of each feature measures the co-occurrence of the feature-term and the term that the vector is representing. The weight function can simply count the number of documents in which the term and the feature-term co-occur, or any measure of their co-occurrence (e.g. PMI, Dice, etc.)

In the syntactic space, the features are a set of syntactic relations that were predefined or that were collected from the whole corpus. Then, all the instantiations of the syntactic relations are collected from the corpus. At last, each term is represented as a tuple of weights of these features, such that the weights are the counts (or probabilities) of the instantiations referring to this term. For example: assume that f1 = <adjective, colorful>, f2 = <adjective, effective> are two features. In the vector of *medicine* the weight of f2 would be probably higher than the weight of f1, (and vice versa for *painting*), as we expect to see *effective medicine* more than *colorful medicine* in the corpus (and more *colorful painting* than *effective painting*).

The commonly used second order similarity measures are: Cosine [Salton 1971a], Lin [Lin 1998a], MinMax [Grefenstette 1994; Curran and Moens 2002; Prior and Geffet 2003], balAPinc [Kotlerman et al. 2009]. For evaluation Curran and Moens [2002] used a gold standard manually composed from 3 electronic thesauri, and evaluated on a 70-word union of synonyms—they took the first 10 synonyms and compared them with respect to their precision and recall scores (recall between 15% and 45%). Kotlerman et al. [2009] used AP and recall scores to evaluate a collection of 1886 word pairs for valid and invalid lexical entailment. As can be seen, similar to first order similarities for thesaurus construction, there is no clear performance benchmark or baseline for these methods as well.

2.2.3 *Additional Aspects.* As mentioned in the preceding different types of context features are likely to capture different kind of semantic information. However, so far little is known about the influence of the context definition on the semantic information it presents. While most researchers choose one specific vector space model and apply it to their task, there have been few comparisons between the models in the literature [Pado and Lapata 2007]. Yet, without any knowledge of the linguistic characteristics of the models, it is impossible to know which approach is best suited for a particular task and why.

Peirsman et al. [2007, 2008] compared and evaluated the performance of three types of word-based models: (1) 1st order similarity in the BOW space, (2) 2nd order similarity in the BOW space, and (3) 2nd order similarity in a syntactic space (later called the dependency-based model). They tested four semantic relations that the three models retrieved (synonym, hyponym, hypernym, and co-hyponym) and investigated the implication of the results by comparing target terms (log) frequency, the distribution of the semantic relations, and the depth of returned related terms in the EuroWordNet hierarchy.

They showed that all three models returned significantly more semantically related terms for high-frequency nouns and especially more synonyms and hyponyms. Moreover, they indicated that the dependency-based model generally outperformed the other models, both in terms of overall performance and in terms of the relative frequency of specific semantic relations retrieved. However, the 1st order BOW model performed a bit less well, but very close to the first model, while the 2nd order BOW gave poor results in comparison to the two previous models.

However, 2nd order BOW models are quite popular in the literature, not only for reasons of data sparseness, but also because they are said to deal better with synonyms (which receive similar vector terms). Therefore they suggest that when the corpus provides enough data, 1st order models are the better alternative by far.

Another aspect that should be considered is corpus size. Curran and Moens [2002] show that the quality or accuracy of a system increases log-linearly to corpus size. On the other hand, the complexity of representing such a corpus in a VSM and applying 2nd order similarities significantly increases.

Given n target terms and m features (vector space size) the asymptotic time complexity is $O(n^2m)$, for 2nd order methods that involve pair-wise vector comparison of every target term with every extracted term. Some ways were suggested to overcome this problem by applying cutoffs on vectors size [Perez-Aguera and Araujo 2007], or transform to lower dimension spaces applying Latent Semantic Analysis (LSA) [Yang and Powers 2008; Giuliano et al. 2010]. (Latent Semantic Analysis (LSA) is a theory and method for extracting and representing the contextual-usage meaning of words by statistical computations applied to a large corpus of text [Landauer and Dumais 1997]). Curran and Moens [2002] investigated the speed/performance tradeoff using frequency cutoffs, and proposed a new approximate comparison algorithm based on canonical attributes. This approximation algorithm is much faster than simply pairwise comparison, with only a small performance penalty. Bayardo et al. [2007] presented a sparse matrix optimization strategy capable of efficiently computing the similarity between terms. Rychly and Kilgarriff [2007], Elsayed et al. [2008], and Agirre et al. [2009] used reverse indexing and the MapReduce framework to distribute the similarity computation across several machines. Recently, Pantel et al. [2009] combined these two strategies and efficiently computed the exact similarity between all pairs in a large corpus.

2.3 Hebrew Processing

2.3.1 *Linguistic Aspects.* Hebrew processing is substantially different from English along a variety of linguistic aspects.

First, and most noticeable, is the visual aspect. The Hebrew alphabet and the writing direction (right-to-left) are different than in English and most European languages.

In addition, Hebrew lacks vocalization, so that most words are ambiguous. For example: the word ספר can be read as *Sapar* and the meaning is *a barber*, or *Sefer* which means *a book*, or *Safar* which means *counted* [Wintner 2004].

Hebrew morphology is very rich, particularly in comparison to English morphology. Many English function words are encoded in Hebrew by prefixes, and a given normal form (root) has many derivative forms, many of which alter the normal form rather than merely augmenting it (for example: לשמש = ל+שמש (*La-Shemesh*, which means *to the sun* (NP)) or לשמש (*Leshamesh*, which means *to serve* (adv))) Wintner [2004]. Concatenating a prefix in one word increases the ambiguity and complexity, for example: ומשאלה = ו+משאלה *(and a wish)* or ומשאלה = ו+מ+שאלה (*and from a question*).

In Hebrew texts, and especially Rabbinic Hebrew texts, acronyms and abbreviations are widely used, even for common phrases (and not only for named entities as in English), thus creating another kind of ambiguity.

Removing stopwords is not a trivial task—many stopwords are ambiguous (for example: אם means *if* but also a *mother*, אבל means *but* and also *mourning*) [Choueka 1997].

For these reasons, the Hebrew language poses special challenges for researchers, while adopting existing NLP tools or developing new linguistic tools for Hebrew processing. The lack of comprehensive corpora and quality knowledge resources for Hebrew (in comparison to English), needed for developing and evaluating such tools, aggravates the situation.

2.3.2 *Thesaurus Usage.* Thesauri are used for a variety of tasks in information retrieval [Xu et al. 1996; Hersh et al. 2000; Tudhope et al. 2006; Bhogal et al. 2007; Rahman et al. 2010]. One commonality to all of the tasks is the need for a synonym for a given word (target term). Cross genre thesaurus (or cross period) is a unique task. Due to the changes in language there is a need for a thesaurus that provides the synonyms for any given word in the various genres or other generation vocabulary.

In this article we will focus on the automated thesaurus construction for a cross genre or cross generation corpus. In the next sections we will present our approach, describe the corpus we used for our experimental results, and present the results followed by a discussion.

3. OBJECTIVE

The objective of this research is to evaluate the various approaches for automatic thesaurus construction for a cross genre or cross period corpora. The main users of this thesaurus would be nonscholar users who are familiar with the modern language but less familiar with ancient terms or previous generation terminology.

For the purpose of this research we used the Responsa Corpus. This is a corpus of questions and detailed rabbinic answers (each question and answer in a separate article) on most walks of life: law, health, commerce, marriage, education, Jewish customs, and lifestyle. The Responsa Corpus contains 76,440 articles, divided into 686,009 paragraphs, and includes 101,191,619 word tokens. The articles are gathered in 1046 volumes and 202 books. The vocabulary contains 750,151 unique word forms (counting each inflected surface-form as a separate one). Average document length is 1,323 words.

Due to its cross generation nature this corpus has several important and unique characteristics.

(1) First, since the Jewish nation was dispersed all over the world for 2000 years, spoken Hebrew was affected by local languages according to settlement areas; hence the documents are of various genres.
(2) In addition, the Hebrew language evolved during a period of 1000 years, so the differences between the vocabulary and style of modern documents and ancient documents are significant.
(3) Another source of the variation in language is the special style of the response books. A response document in general includes all the Jewish juristic negotiation that led to the final decision. It presents all the arguments for and against, by citing earlier sources, like the Talmud and its commentators, the legal codes and earlier responses, enriching the writing style in an even more archaic manner [Koppel 2008].
(4) Pursuant to (1) and (3), the corpus actually involves three different languages: Hebrew, Yiddish and Aramaic.
(5) In Rabbinic Hebrew texts such as those in our corpus, acronyms and abbreviations are widely used, even for common phrases. In many of the documents in the Responsa Corpus, abbreviations make up about 20% of the token-words, and over one third of them allow for more than one expansion [Hacohen-Kerner et al. 2004].
(6) Last, this corpus differs from other modern corpora in that it was intended as a source of information and not merely as a source of exemplars of language use [Koppel 2008].

As for thesaurus entries, we used frequent dictionary entries in larger resources. We used article titles from Hebrew Wikipedia entries and the Index of Hebrew Periodicals (IHP) from the University of Haifa Library. We filtered this list according to our needs; dates, names of persons, and places were removed. In addition, rare terms were removed. From this list, we chose 70 of the most relevant yet frequent enough terms, of which we used 20 for a train set and 50 for a test set. The complete lists of train set and test sets can be found in Appendix A.

4. METHOD

We plan to evaluate the first and second order methods previously described. For this research we use the bag-of-words (BOW) space. We decided not to work in the syntactic space since it necessitates

syntactic and grammatical analysis of the corpus, which would not work well in our case for the following three reasons.

(1) Linguistic tools for Hebrew do not perform well enough, in comparison to similar tools for English.
(2) The linguistic tools that do work for Hebrew are based on statistical analysis of modern text, and as such are more fitting for analyzing Modern Hebrew.
(3) Applying these tools on the Responsa Corpus, which involves archaic Hebrew, Aramaic, and an extremely high percentage of abbreviations and acronyms, would adversely affect performance.

For this research we consider the context of a term to be the whole document, and work at the level of documents. Thus, from now on we will refer to a term's occurrence in the corpus as the number of documents in which it appears. We implemented all algorithms in house (using Java) in a manner similar to the description given in the literature. The entire flow was implemented as an add-on to the Responsa System, and potentially planned to be added to the system in one of its future versions.

4.1 First Order Methods

First order methods assume that related terms occur together frequently, and that the higher their co-occurrence, the more related they are to each other. For that purpose we used two common similarity measures: Dice coefficient [Van Rijsbergen 1979] and Pointwise Mutual Information (PMI) [Bouma 2009].

4.2 Second Order Methods

Second-order methods are based on the assumption that related terms occur in similar contexts but not necessarily together. The context of each target term is represented as a vector of features, which in the BOW model are tuples of context terms and their weights. A context term's weight is also referred to as an *association score*.

During the course of applying second-order methods on the train set, we encountered run-time difficulties while creating the term vectors. Therefore, we used an approximation for the representation of a term as a vector in all second-order algorithms. For the target term's vector we chose only the top-25-scored terms (based on the algorithm score used) for the feature set. In preliminary experiments, 25 has been proven to be the optimal number of terms (among various options of terms lists) which optimizes both recall as well as algorithm complexity and run-time. This choice is consistent with the distributional hypothesis principle, since these feature terms have high co-occurrence with the target term, and therefore can define its context. For the candidate term's vector, we used those same 25 features. In practice, each feature in the vector is the weight of the candidate term's co-occurrences with the feature term, while the weight of features that don't co-occur with the candidate term is zero.

To approximate the similarity measures of each target term with every term in the corpus, we defined for each target term a limited set of candidate terms, and calculated the similarity for those in this set alone. The candidate terms in the set were chosen to be the terms in all the related-terms vectors of the 25 feature-terms of the target vector. Recall that the vector100 terms are chosen according to highest first-order similarity. This produces 2500 candidate terms, and after eliminating duplicates there remained about 2000 candidate terms on average per target term.

Given a target term and its set of candidate terms, we use their vectors to give each candidate term a score based on one of the second-order measures. The 25 candidate terms achieving the highest scores are to be considered as related terms for the thesaurus.

4.3 Combined Method

The combined method is based on the assumption that different algorithms are characterized by different types of related terms. If there are two algorithms that contribute different types of related terms, their unification can improve coverage significantly, as we take the different words and approaches and combine them into one. Thus, we will improve recall significantly with low precision impact.

Using this method, we examined the combined output of pairs of algorithms consisting of the output of the top two performing algorithms: Dice and PMI—with each one of the other algorithms, resulting in 12 different combinations. The output of the combined algorithms is the union of the outputs of both algorithms. Please note that since the combined output consists of 50 terms (25 from each algorithm), we compared those results to the 50 top related terms from each original algorithm (instead of the top 25 as we did in the other experiments so far). By taking this approach we eliminated the possibility of seeing an increase in recall due to the fact of increased term list. This approach (which will also be validated statistically) will improve overall recall with little precision impact.

5. EVALUATION

In this section, we will present the results of all 19 algorithms introduced in the previous section: 3 algorithms based on first order similarities, 4 algorithms based on second order similarities, and 12 algorithms based on the combined approach.

For each target term in the train and test sets, we gathered candidate terms from all the outputs of the tested algorithms. They were judged by an expert in the domain of Rabbinic Responses, who determined for each candidate term whether it should appear in this entry of the thesaurus or not. The annotator was instructed to consider synonyms, hypernyms, hyponyms, negations, and Halachic-related terms as positive, and other terms as negative. Terms that were judged to be distantly related were considered by us as negative. We considered the union set of all positive terms received from all algorithms as the set of relevant terms for a given target term. All of these positive related terms were used to construct a gold-standard thesaurus, which served for evaluating each thesaurus constructed by our algorithms.

5.1 Scoring Method

In the first set of experiments we compared the performance of the seven algorithms we presented in Section 4 by averaging their results on the test terms. For each algorithm, we measured the average of its precision score (P),[1] relative recall (R),[2] F1,[3] and average precision (AP).[4] However, we were more interested in relative recall than in precision and F1, since we prefer the final thesaurus to be complete and comprehensive, rather than it being accurate. We gave priority to the coverage, even at the expense of slightly noisier output, which in any case is planned to be manually filtered and edited by an expert.

While reviewing the automatically constructed set of related terms we noticed that since we didn't apply any morphological preprocessing, several terms occurred in multiple inflections. For example, for the target term צהבת (*tsahevet* – jaundice), we got the values:

[1] $P_{alg}(term) = \frac{|(relevant) \cap (retrieved)|}{|(retrieved)|}$.
[2] $R_{alg}(term) = \frac{|(relevant) \cap (retrieved)|}{|(relevant)|}$.
[3] $F1 = \frac{2 \cdot precision \cdot Recall}{Precision + Recall}$.
[4] $AP_{alg} = \frac{1}{R} \cdot \sum_{i=1}^{n} \frac{E(i) - correctUpToRank(i)}{i}$, $R - the\ number\ of\ relevant\ terms\ in\ the\ gold\ standard$
$E(i) = 1 - if\ the\ i^{th} candidate\ term\ is\ relevant, 0 - otherwise$
$CorrectUpToRank(i) = the\ number\ of\ relevant\ terms\ among\ the\ \ first\ i\ candidate\ terms.$

ירוק (*Yarok* – green);
ירוק ביותר (*Yarok BeYoter* – extremely green);
דבירוק (*U'Be'Yarok* – and in green);
הוא ירוק (*Hu Yarok* – he is green);

which are all inflections of *green*, which was used in the past to describe the look of jaundiced people. Another example is the target term אימוץ (*imuts* – adoption). For this term, we got the values:

שתוקי – *shtuki*;
או שתוקי – *o shtuki*;
ששתוקי – *she'shtuki*;
שתוקית – *shtukit*;
בשתוקי – *ke'shtuki*;
משתוקי – *me'shtuki*

which are inflections of the designation of a person who doesn't answer when he is asked who his parents are (*shtuki*). This is a side effect of the rich morphology of the Hebrew language. These inflections actually refer to one lexeme and therefore create unnecessary duplication in the output thesaurus. In addition, if one algorithm returns one or more inflections and another algorithm returns other inflections, we would like to consider them as overlapping and not as distinct values. Accordingly, we devised a second annotation scheme, which takes this duplication into consideration. The new annotation instructions require adding a positive group number to each positive term, and a negative group number to each negative term. The group number represents the lexeme of the term. We consider all the inflections of this lexeme as one group, and give them the same group number. We evaluate the output terms according to the group they belong to, as explained in the following.

In addition, since our index contains unigrams and bigrams only, the algorithm's output cannot contain phrases that consist of more than two words. However, in reality, many phrases are composed of more than two words. For these phrases, the output contains fractions of the phrase. For example, for the target term הפלה (*ha'pala* – abortion), we got משום איבוד (*mishum ibud* – due to loss), and איבוד נפשות (*ibud nefashot* – loss of life), which are fractions of the positive phrase משום איבוד נפשות (*mishum ibud nefashot* – due to loss of life). In the same way, the terms, אין דוחים, (*ein dohim* – don't trade) דוחים נפש (*dohim nefesh* – trade a soul) and נפש מפני (*mipnei nefesh* – from (another) soul), are fractions of the positive phrase אין דוחים נפש מפני נפש (*ein dohim nefesh mipnei nefesh*—don't trade one soul for another soul). Both are rabbinic phrases explaining the reasoning behind the prohibition of having an artificial abortion. The annotator was instructed to assign the same group number to each phrase and all its fractions.

According to our grouped-annotation approach, the annotation of each term contains its polarity (+ for positive terms, and − for negative terms), and the number of the group, both for positive and negative terms.

For the evaluation, which is now based on the these instructions, we took the original output (a list of related terms) and constructed a list of the relevant group numbers for each of the related terms. Then, we removed duplicates, and finally we evaluated the resulting list in terms of groups. Figure 1 demonstrates the representation of the related terms as a list of groups for evaluation. As can be seen in this example, for the target term אימוץ (*imuts)*, the annotator assigned the same group number (3) to the following related terms:

דין אסופי (*din asufi* – the law of an "asufi");
באסופי (*ke'asufi* – like an "asufi");
האסופי שנמצא (*ha'asufi she'nimtsa* – an "asufi" that was found);
אסופי או (*oh' asufi* - or an "asufi").

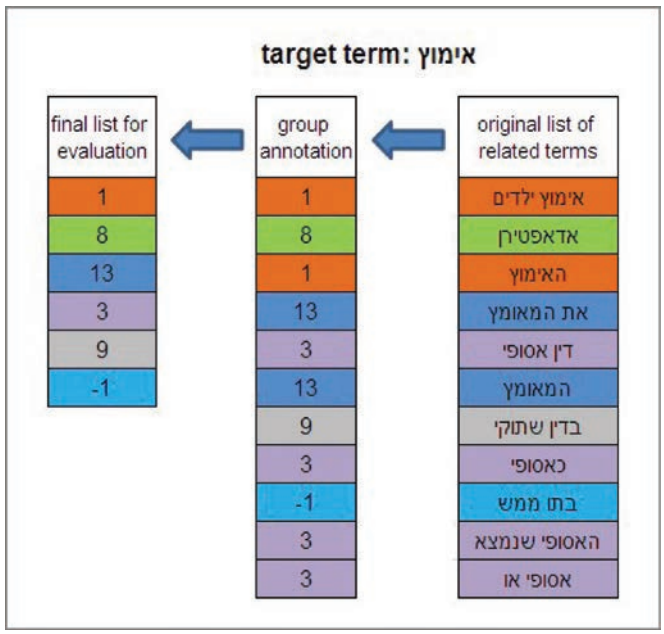

Fig. 1. Partial output of the Dice algorithm on the target term אימוץ (*imuts*—adoption): the retrieved related terms, the grouped annotation and the final group list with no duplicates, for evaluation.

All of these terms stem from the same term: אסופי (*asufi*—a foundling). Similarly, the related terms את המאומץ (*et ha'me'umats*—the adopted) and המאומץ (*ha'me'umats*—the adopted) were assigned the same group number (13), which means מאומץ (*me'u'mats*—adopted). The related term אדאפטירן (*adaptieren*—the Yiddish term for *adopted*) was assigned group number 8. The related term בדין שתוקי (*be'din shtuki*—the law of "shtuki") was assigned group number 9. Other algorithms returned the related terms: בשתוקי (*ki'shtuki*—like a "shtuki"), ששתוקי (*she'shtuki*—that a "shtuki") אבל שתוקי (*shtuki aval*—but a "shtuki"), and או שתוקי (*oh shtuki*—or a "shtuki"). All these related terms refer to the same meaning represented by the lexeme שתוקי (*shtuki*—someone who doesn't know or is embarrassed to answer who his parents are), and therefore they were also assigned group number 9.

We find that this annotation scheme better reflects the quality of the related terms, as it distinguishes between different meanings among related terms and avoids duplicates caused by different inflections of the same related term. For example, as shown in Figure 1, the related term בדין שתוקי (*be-din shtuki*—and the law of a "shtuki") was retrieved by the Dice algorithm, but the related (inflected) terms: בשתוקי (*ki'shtuki)*, אבל, או שתוקי שתוקי (*oh shtuki)* and ששתוקי (*she'shtuki)* were not retrieved by this algorithm. The common annotation approach refers to each related term as distinct. Therefore, recall is penalized since not all inflections were identified by the Dice algorithm. The grouped annotation approach requires only one inflection for each group, which is more reasonable.

5.2 Results

Since we are more interested in recall over precision, we'll sort the results based on their recall scores. We see that the best performing algorithms are the ones that use first-order methods, such as, Dice and PMI, followed by Rocchio algorithms. The second-order algorithms using PMI weights, and Dice weights came last. This is a quite interesting finding, compared to previous research, which found that

Table I. Results based on Grouped Annotation, Ordered by Recall

Algorithm	Precision	Recall	F1	Average Precision
Dice	31.30	65.22	37.61	49.85
PMI	28.46	64.82	35.27	49.04
Rocchio	27.77	62.93	35.11	46.19
Cos(PMI)	24.29	55.64	33.71	40.45
Lin(PMI)	23.63	54.73	32.75	37.28
Cos(Dice)	23.72	53.20	33.42	30.47
Lin(Dice)	25.22	52.04	33.90	29.48

Table IIa. Combined Algorithm Contribution to Recall (base Dice)

Combined algorithm	Dice Recall	alg.2 Recall	alg.2 addition	combined alg. Recall
Dice-Cos(Dice)	65.22	53.20	16.36	81.58
Dice-Rocchio	65.22	62.93	14.48	79.70
Dice-Lin(Dice)	65.22	52.04	11.83	77.05
Dice-Cos(PMI)	65.22	55.64	10.58	75.80
Dice-Lin(PMI)	65.22	54.73	9.51	74.73
Dice-PMI	65.22	64.82	6.28	71.50

Table IIb. Combined Algorithm Contribution to Recall (base PMI)

Combined algorithm	PMI Recall	alg.2 Recall	alg.2 addition	combined alg. Recall
PMI-Cos(Dice)	64.82	53.2	17.14	81.96
PMI-Rocchio	64.82	62.93	14.54	79.36
PMI-Lin(Dice)	64.82	52.04	13.47	78.29
PMI-Cos(PMI)	64.82	55.64	11.4	76.22
PMI-Lin(PMI)	64.82	54.73	9.54	74.36
PMI-Dice	64.82	65.22	6.68	71.5

for automatic thesaurus construction, second order algorithms work better than first order ones. Full results are shown in Table I.

As mentioned in 4.3, we assumed that due to difference in the nature of the approaches, the combination of the results from the different algorithms may introduce some improvements in the overall recall (which we are interested in) at the cost of some precision.

As the best performing first order algorithms, we used Dice and PMI as the base algorithms for the second order experiment and combined them with the other six algorithms, to which we refer here as the second algorithms. We measured the contribution of each of the second algorithms to the coverage of the Dice or PMI algorithm. In addition, we measured precision and F1 to give the total performance; in all cases we also tested statistical significance of the results. Results for base algorithms Dice and PMI are shown in Table II(a) and Table II(b).

As shown in Table II(a), the highest increase in recall was obtained by the combination of Dice and Cos(Dice) ($R = 81.58$). This result is surprising since the Cos(Dice) algorithm alone obtained poor results (as shown in Table I).

The second best result was achieved by the combination of Dice with Rocchio ($R = 79.70$), which is not usually used for thesaurus extraction, and is better known as a baseline algorithm for the query expansion task in IR. Both results were statistically significant with $p < 0.05$ over first order algorithms,

and with $p < 0.001$ over second order algorithms (which their initial performance was worse than the first order).

However, the advantage of the Rocchio and the Cos(Dice) algorithms in improving the coverage may be supported by their different approaches. The Dice algorithm is based on information about terms' co-occurrences, while the Cos(Dice) algorithm is based on measuring similarity of contexts, and the Rocchio algorithm is based on relevance feedback (information about documents). Each approach processes different information and as a result contributes different terms, in addition to common related terms. As mentioned before, in this comparison we compared the combined approach (composed from 50 terms—the combination of first and second order algorithms) to first or second approaches (separately) with 50 terms each, to eliminate bias of results due to increased term set.

For example, when looking at the target term אינקובטור (*incubator*—incubator), the first-order algorithms Dice and PMI returned the same five groups assigned to the related terms: שערו נגמר (*nigmar s'aro*—his hair is done), נגמר גידולו (*nigmar gi'dulu*—his growth is done), נגמרו צפרניו (*nigmeru tsi'pornav*—his nails are done), שהה ל' יום (*sha'ha sheloshim yom*—completed 30 days), and אינקובטור (*incubator*—incubator). These terms describe several prenatal development stages. The Rocchio algorithm covers more groups, representing the related terms: נולד לשמונה חודשים (*nolad li'shmona hodashim*—born after 8 months), נולד לשבעה חודשים (*nolad le'shiv'a hodashim*—born after 7 months), בן קימא (*ben ka'yama*—sustainable), כאבן, כלו לו חודשיו (*kalu lo hodashav*—completed his months). The Cos(Dice) algorithm adds נפל (*nefel*—abortus), גמרו סימניו (*gamru simanav*—his signs are completed), שערו וצפורניו גדלו (*se'aro ve'tsipornav gadlu*—his hair and nails grew), נולד לשבעה חודשים (*nolad le'shiv'a hodashim*—born after 7 months), נולד לשמונה חודשים (*nolad lishmona hodashim*—born after 8 months) (additional phrases for stages in prenatal development).

Table III summarizes the distribution of the different groups in the outputs of Dice, PMI, Cos(Dice), and the Rocchio algorithms, for the target terms: אבקת חלב (*avkat halav*—milk powder) and אינקובטור (*incubator*—incubator). The precision of the combined algorithms is presented in Table IV.

The observed decline in precision values is natural and can be attributed to the overlap between the positive related terms returned by the Dice algorithm and the second algorithm. Another reason for the decrease in precision is the originally lower precision of the second algorithm, which impacts overall precision. In addition, our grouped annotation approach has a negative impact on precision, as we explained in Section 5.1.

The combined algorithm that obtained the highest precision values is Dice-PMI ($P = 26.06$). This combination achieved the highest precision since it is based on the same approach as the Dice algorithm, and as previously explained and shown in Table IV, the related terms they returned were similar to Dice's related terms.

The Dice-Rocchio combined algorithm and the Dice-Cos(Dice), achieved intermediate precision values ($P = 23.15$ and $P = 22.71$ respectively), strengthening our assumption that they retrieve different types of related terms, and have only little overlap with Dice's output.

Table V displays F1 results. The total performance presented by F1 is decreased because of the bad influence of the precision, and because F1 tends strongly towards the lower element between precision and recall. The Dice-Rocchio algorithm obtained $F1 = 32.80$, and the Dice-Cos(Dice) algorithm obtained $F1 = 32.35$, while the best F1 value was obtained by the Dice-PMI combination ($F1 = 34.37$), which didn't significantly extend the coverage achieved by Dice ($R = 71.50$).

We repeated the creation and analysis of combined algorithms for all algorithms combined with the PMI algorithm, and for all algorithms combined with the Rocchio algorithm. Surprisingly, in both sets we got similar results. For PMI, the best coverage was achieved by the combination with Cos(Dice) ($R = 81.96$, $P = 20.56$, $F1 = 30.56$), followed by the combination with Rocchio ($R = 79.36$, $P = 21.69$, $F1 = 31.23$). For algorithms combined with the Rocchio algorithm, the best recall was achieved with

Table III. The Distribution of the Different Groups in the Outputs of Dice, PMI, Cos(Dice), and the Rocchio Algorithms, for the Target Terms אבקת חלב and אינקובטור

Target term: אבקת חלב (*avkat halav* – milk powder)

Rocchio	Cos(Dice)	Pmi	Dice	related term	group
2	1	5	5	(milk powder) אבקת חלב	1
3		7	7	(powder) אבקה	2
1	1	2	4	(non jewish milk) חלב עכו"ם	3
1		1	4	(unpurified milk) חלב טמא	4
	1	1	1	(milk mix) תערובת חלב	6
	1			(holes in cheese)אטפי [גומות שבגבינה]	9
		1	1	(cow milk) חלב פרה	11
	1			(milk's law) גזירת חלב	13
	2			(fear of other milk) חשש חלב אחר	16
	1			(milk extraction) מצוי חלב	18
	1			(milk rhetorical) צחצוחי חלב	19
	1			(fear from jew)אימת הגוי מיהודי	21
3				(butter) חמאה	22
2	3	2		(cheese) גבינה	23

Target term: אינקובטור

Rocchio	Cos(Dice)	Pmi	Dice	related term	group
2	2	2	2	(incubator) אינקובטור	1
1	2	1	1	(completed his hair) שערו נגמר	2
2	3	1	2	(his growth complete) נגמר גידולו	3
	1			(hair and nails complete) שערו וצפרניו גדלו	4
1	1			(born after 8 months) נולד לשמונה חודשים	6
1	1			(born after 7 months) נולד לשבעה חודשים	7
2				(either 7 or 8 months) ספק בן ז או ח	9
	1			(completed his signs) גמרו סימניו	14
1	3			(aborted fetus) נפל	16
3				(like a stone) כאבן [מוקצה]	20
1				(sustainable) בן קימא	21
1				(completed his months) כלו לו חודשיו	22
2		2	2	(existed for 30 days) שהה ל' יום	23
	1	1	1	(nails completed) נגמרו צפרניו	27

Table IV. Combined Algorithms' Precision

Algorithm	Dice Precision	alg2. Precision	combined Precision
Dice-PMI	31.30	28.46	26.06
Dice-Lin(Dice)	31.30	25.22	24.52
Dice-Rocchio	31.30	27.77	23.15
Dice-Cos(PMI)	31.30	24.29	22.79
Dice-Cos(Dice)	31.30	23.72	22.20
Dice-Minmax(PMI)	31.30	23.27	21.50

Table V. Combined Algorithms' F1

Combined algorithm	Dice F1	alg.2 F1	combined alg. F1
Dice-PMI	37.61	35.27	34.37
Dice-Lin(Dice)	37.61	30.37	33.35
Dice-Rocchio	37.61	35.11	32.80
Dice-Cos(Dice)	37.61	30.64	32.35
Dice-Cos(PMI)	37.61	30.20	31.91
Dice-Minmax(PMI)	37.61	29.16	30.48

Table VI. Error Analysis Results For Dice, Rocchio, and Cos(Dice) Algorithms

	Percentage		
Type of error	Dice	Rocchio	Cos(Dice)
In common context	69	61	60
Target term ambiguity	10	1	3
Invalid related term	7	38	36
Rare bigram	15	-	-

Cos(Dice) (R = 81.20, P = 21.12, F1 = 30.98), followed by the combination with Dice (R = 79.70, P = 20.86, F1 = 32.80). These results strengthen our assumption as to the different nature of each of these algorithms, and the potential in using their combinations.

Although the improvement in recall was foreseen, we believe it indeed reflects the actual recall better, since what we are really interested in is different lexemes rather than different inflections.

5.3 Error Analysis

In the error analysis we will concentrate on the Dice-Rocchio and the Dice-Cos(Dice) algorithms, which proved to be the best-performing for the current task. The error analysis refers to each algorithm (Dice, Rocchio, and Cos(Dice)) separately, in order to better understand each algorithm's characteristics. In order to perform the error analysis, we randomly sampled 10% of false-positive related terms for each of the algorithms. The investigation of the sampled examples allowed us to identify several reasons for retrieving negative related terms, whose distribution is presented in Table VI.

The terms that fall into the first error type ("common context") can be referred to as related terms that are somewhat relevant, but were considered as nonrelated terms by strict annotation. However, the last two error types refer to returned terms that cannot be considered as related in any context.

We describe the reasons for each type of detected error, and give some examples in the following text.

(1) *In common context.* The algorithms we presented in this work are based on statistical methods. Therefore, they tend to retrieve a lot of terms common in the context of the target term that are

Table VII. "In Common Context" Errors

Target term	Related term	Possible context in the corpus
ביצית (= ovum, ovule)	שמעברת (to impregnate)	Discussions around pregnancy
שעות עבודה (=working hours)	פועל (a worker)	Working hours for a worker
גוסס (= moribund)	זקן (aged)	
משטרה (police)	ביוהכ"פ (on Yom Kippur)	In a discussion on the allowed activities of the police during Yom Kippur.

Table VIII. "Target Term Ambiguity" Errors

Target term	Related term	Target term's second meaning
המחאה (a cheque)	לא מיחה (didn't protest)	The (ה) protest (מחאה)
היתר מכירה (Halachic solution allowing to grow vegetables during Shnat Shmita)	הפאצט (business management license)	A selling permission
צורף (goldsmith)	בפרוטוקול (in the protocol)	Was attached.

Table IX. "Rare Bigram" Errors

Target term	Related term	The number of related term's occurrences	The number of target term and related term joint occurrences
חסכון (savings)	אומנים או (artists)	3	2
שמירת הלשון (keep words)	אחיו ומדוע (his brothers why)	3	2
יום הולדת (birthday)	בזה שהמקור (the source)	3	2

somewhat (although not directly) related according to the annotation. This may include for example: (rare) verbs associated with certain nouns (which are of course not synonyms of that noun), or different nouns discussed together but none of them is a synonym of the other and more. A few examples can be found in Table VII.

(2) *Target term ambiguity.* These errors occur because the target term is ambiguous and the related term refers to one of the meanings. Examples for this error can be found in Table VIII.

(3) *Invalid related term.* These are nonrelated terms. For these words we didn't manage to discover any connection to the contexts that we would like to retrieve for the target term.

(4) *Rare bigram.* This error type refers to invalid related terms caused by a weakness of the Dice algorithm. The Dice measure tends to reward infrequent bigrams and give them a high score when most of their few occurrences appear together with the target term. Some examples can be found in Table IX.

In addition, this error analysis also indicates that the Dice algorithm is superior to the Rocchio and the Cos(Dice) algorithms in its performance (the quality of true-positive related terms), but also in the quality of its error (the percentage of invalid returned terms). The Dice algorithm, which achieved the best performance, returned 22% invalid terms. The Rocchio algorithm returned 38% invalid terms, and the Cos(Dice) algorithm returned 36% invalid terms, as shown in Table VI.

6. CONCLUSIONS

In this article, we described various algorithms based on different approaches for automatic thesaurus construction for a cross generation corpus. The methods we investigated were applied to the Responsa Corpus, a domain-specific Hebrew corpus. This corpus is challenging in many senses: the written language (Hebrew), which has not yet been investigated thoroughly, the plurality of writing styles (since the writing of this document collection extends over a long period by many writers from various countries), the use of terms in different languages, and the massive use of abbreviations and acronyms, which increases ambiguity. In this work, we intended to investigate to what extent the state-of-the-art methods can be effective in constructing a thesaurus from such a nontrivial corpus, whether there is a predominant method that outperforms the others, and whether the thesaurus can provide the added value of identifying various archaic terms related to modern target terms.

We found that first order methods work pretty well in these settings, but moreover were positively motivated by the combined algorithm, that has proved to provide better recall with a low penalty in precision. We have plans for further investigation and improvements in precision scores for these algorithms. Based on the error analysis, we found that some future research in the area of distinction between related terms and synonyms can be also very useful for this type of corpus.

APPENDIX: TRAIN AND TEST SETS

Train set

אימוץ (*imutz* – adoption)
אפילפסיה (*epilepsia* – epilepsion)
גפרור (*gafrur* - matches)
דכאון (*dikaon* – depression)
הפלה (*hapala* – abortion)
הצטננות (*hitstanenut* – cold)
התעמלות (*hitamlut* – gymnastics)
זיהום (*zihum* – infection)
טובת הילד (*tovat ha'yeled* – child's benefit)
כיסוי ראש (*kisui rosh* – hair cover)
מכונית (*mehonit* – car)
מניעת הריון (*meniat herayon* – contraception)
מסעדה (*misada* – restaurant)
סיגריה (*sigarya* – cigarette)
פמוט (*pamot* – candlestick)
צהבת (*tsahevet* – jaundice)
קוסם (*kosem* – magician)
שיתוק (*shituk* – paralysis)
תותבות (*totavot* – dentures)

Test set

אבקת חלב (*avkat halav* – milk powder)
אינקובטור (*inqubator* – incubator)
הוצאות נסיעה (*hotsaot nesia* – travel expense)
הזיות (*hasayot* – illusions)
היתר מכירה (*heter mehira* – sell allowance)
המחאה (*hamha'a* – bank check)
המרת דת (*hamarat dat* – conversion)
הפסקת הריון (*hafsakat herayon* – abortion)
הרדמה (*hardama* – anesthesia)
השכלה (*haskala* – education)
השקעה (*hashka'a* – investment)
חלב אם (*halav em* – milk)
חלב עכו"ם (*halav akum* – non jewish milk)
חסכון (*hisachon* – savings)
טהרת המשפחה (*taharat mishpaha* – family purity)
יום האם (*yom ha'em* – mother's day)
יום הולדת (*yom huledet* – birthday)
יחסי אישות (*yahasei ishut* – sexual relationship)
יפוי כח (*yipui koach* – power of attorney)
כאבי ראש (*ke'evey rosh* – headache)
כבוד המת (*kevod hamet* – respect for the dead)
כלי כתיבה (*klei' ktiva* – writing instruments)
כתובת קעקע (*ketovet ka'aka* – tattoo)
מגנט (*magnet* – magnet)
מלווה מלכה (*melave malka* – Saturday dinner)

איסור נגיעה (*isur negia* – jewish law about gender separation)
איסור קטניות (*isur kitnoiot* – legumes)
אריכות ימים (*arihut yamim* – longevity)
בדיקות דם (*bdikot dam* – blood test)
בול (*bul* – stamp)
ביצית (*beitsit* – ovule)
בית מלון (*bet malon* – hotel)
בית ספר (*bet sefer* – school)
בנק (*bank* – bank)
בשר חזיר (*basar hasir* – pork)
בת מצוה (*bat mitzvah* – bat mitzvah)
גוסס (*goses* – dying)
דמי מזונות (*dmei mezonot* – alimony)
דמי מפתח (*dmei mafteah* – special rent)
דמי שכירות (*dmei sehirut* – rent price)
דרכון (*darkon* – passport)
הגרלה (*hagrala* – lottery)
מצה שרויה (*matsa sheruya* – matsa bread)
משטרה (*mishtara* – police)
ניחום אבלים (*nihum avelim* – sympathy)
נימוסים (*nimusim* – manners)
נישואין אזרחיים (*nisuim ezrahi'im* – civil wedding)
נקמת דם (*nikmat dam* – vendetta)
סבון (*sabon* – soap)
עוור (*iver* – blind)
עישון (*ishun* – smoking)
פוקר (*poker* – poker)
פיצויים (*pitsuim* – severance)
פנס (*panas* – flashlight)
צורף (*tsoref* – goldsmith)
צער בעלי חיים (*tsa'ar ba'alei haim* – green peace)
רפואה טבעית (*refu'a tiv'it* – alternative medicine)
שלום בית (*shlom bayit* – family peace)
שמירת הלשון (*shmirat halashon* – not talking about others)
שעות עבודה (*shaot avoda* – working hours)

REFERENCES

AGIRRE, E., ALFONSECA, E., HALL, K., KRAVALOVA, J., PAS, M., AND SOROA, A. 2009. A study on similarity and relatedness using distributional and WordNet-based approaches. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*. 19–27.

BAYARDO, R. J., MA, Y., AND SRIKANT, R. 2007. Scaling up all-pairs similarity search. In *Proceedings of WWW*. 131–140.

BHOGAL, J., MACFARLANE, A., AND SMITH, P. 2007. A review of ontology based query expansion. *Inform. Proc. Manage. 43*, 866–886.

BOUMA, G. 2009. Normalized (pointwise) mutual information in collocation extraction. In *Proceedings of the Biennial GSCL Conference*. 31–40.

CHARNIAK, E. AND BERLAND, M. 1999. Finding parts in very large corpora. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*. 57–64.

CHOUEKA, Y. 1997. The complexity of development of Hebrew search engine. In *Proceedings of the Forum of Governmental Webmasters*, Givat Ram Campus (lecture notes in Hebrew).

CHURCH, K. W. AND HANKS, P. 1990. Word association norms, mutual information and lexicography. *ACL Comput. Ling. 16*, 4.

COVER, T. M. AND THOMAS, J. A. 1991. *Elements of Information Theory*. Wiley.

CURRAN, J. R. 2003. From Distributional to Semantic Similarity. Ph.D thesis, Institute for Communicating and Collaborative Systems, School of Informatics, University of Edinburgh.

CURRAN, J. R. AND MOENS, M. 2002. Scaling context space. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*. 231–238.

ELSAYED, T., LIN, J., AND OARD, D. 2008. Pairwise document similarity in large collections with MapReduce. In *Proceedings of the ACL: HLT* (Short Papers Companion Volume). 265–268.

FELLBAUM, C. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press, Cambridge, MA.

GIRJU, R. AND MOLDOVAN, D. 2002. Text mining for causal relations. In *Proceedings of the FLAIRS Conference*. AAAI Press.

GIULIANO, C., GLIOZZO, A. M., GANGEMI, A., AND TYMOSHENKO, K. 2010. Acquiring thesauri from Wikis by exploiting domain models and lexical substitution. In *Proceedings of the 7th Extended Semantic Web Conference (ESWC)*.

GREFENSTETTE, G. 1994. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers, Boston.

HAN, R., DONGHONG, J., AND JING, W. 2011. A Web knowledge based approach for complex question answering. In *Proceedings of the 7th Asian Information Retrieval Societies Conference*.

HACOHEN-KERNER, Y., KASS, A., AND PERETZ, A. 2004. Baseline methods for automatic disambiguation of abbreviations in Jewish law documents. *Proceedings of the 4th International Conference on Advances in Natural Language (LNAI)*.

HACOHEN-KERNER, Y., KASS, A., AND PERETZ, A. 2008. Combine One Sense Disambiguation of Abbreviations. In *Proceedings of ACL* (Companion Volume).

HEARST, M. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th International Conference on Computational Linguistics*.

HERSH, W., PRICE, S., AND DONOHOE, L. 2000. Assessing thesaurus-based query expansion using the UMLS Metathesaurus. In *Proceedings of the AMIA Symposium*. 344–348.

IWANSKA, L., MATA, N., AND KRUGER, K. 2000. Fully automatic acquisition of taxonomic knowledge from large corpora of texts. In *Proceedings of the Conference on Natural Language Processing and knowledge Processing*. 335–345, MIT/AAAI Press.

KOPPEL, M. 2008 The Responsa Project: Some promising future directions. Department of Computer Science, Bar-Ilan University, Ramat-Gan.

KOTLERMAN, L., DAGAN, I., SZPEKTOR, I., AND ZHITOMIRSKY-GEFFET, M. 2009. Directional distributional similarity for lexical expansion. In *Proceedings of the ACL-IJCNLP Conference*. 69–72.

LANDAUER, T. AND DUMAIS, S. 1997. A Solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psych. Rev. 104,* 2, 211–240.

LIN, D. 1998a. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference On Machine Learning.* 296–304.

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

LESK, M. E. 1969. Word-word associations in document retrieval systems. *Amer. Doc. 20*, 1, 27–38.

MANDALA, R., TOKUNAGA, T., AND TANAKA, H. 2000. Query expansion using heterogeneous thesauri. *Inform. Process. Manage. 33*, 3.

MANNING, C. D., RAGHAVAN, P., AND SCHÜTZE, H. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

MILLER, U. 1997. Thesaurus construction: Problems and their roots. *Inform. Process. Manage*. 481–493.

PADÓ, S. AND LAPATA, M. 2007. Dependency-based construction of semantic space models. *Comput. Ling. 33*, 2, 161–199.

PANTEL, P., CRESTAN, E., BORKOVSKY, A., POPESCU, A.-M., AND VYAS, V. 2009. Web-scale distributional similarity and entity set expansion. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 938–947.

PEIRSMAN, Y., HEYLEN, K., AND SPEELMAN, D. 2007. Finding semantically related words in Dutch. Cooccurrences versus syntactic contexts. In *Proceedings of the CoSMO Workshop* held in Conjunction with *CONTEXT-07*.

PEIRSMAN, Y., HEYLEN, K., AND SPEELMAN D. 2008. Putting things in order. First and second order contexts models for the calculation of semantic similarity. In *Proceedings of the Actes des 9i'emes Journ'ees internationales d'Analyse statistique des Donn'ees Textuelles (JADT)*. 907–916.

PEREZ-AGUERA, J. R. AND ARAUJO, L. 2007. Comparing and combining methods for automatic query expansion. In *Proceedings of the Advances in Natural Language Processing and Applications*.

PRIOR, A. AND GEFFET, M. 2003. Mutual information and semantic similarity as predictors of word association strength: Modulation by association type and semantic relation. In *Proceedings of the 1st European Cognitive Science Conference*.

RAHMAN, N. A., BAKAR Z. A., AND SEMBOK, T. M. T. 2010. Query expansion using thesaurus in improving Malay Hadith retrieval system. In *Proceedings of the International Symposium Information Technology (ITSim).* 1404–1409.

ROCCHIO, J. J. 1971. Relevance feedback in information retrieval. In Salton, G. (Ed.), *The SMART Retrieval System—Experiments in Automatic Document Processing*, Prentice-Hall, Englewood Cliffs, NJ, 313–323.

RYCHLY, P. AND KILGARRIFF, A. 2007. An efficient algorithm for building a distributional thesaurus (and other Sketch Engine developments). In *Proceedings of ACL-07*. Demo Session.

SALTON, G. 1971. Experiments in automatic thesaurus construction for information retrieval. *Inform. Process. 71*, 1, 115–123.

SALTON, G. 1971a. *The SMART Retrieval System—Experiments in Automatic Document Processing*. Prentice-Hall, Englewood Cliffs, NJ.

SCHÜTZE, H. AND PEDERSEN, J. O. 1997. A cooccurrence-based thesaurus and two applications to Information Retrieval. *Inform. Proc. Manage.* 307–318.

SCHÜTZE, H. AND PEDERSEN, J. O. 1994. A cooccurrence-based thesaurus and two applications to Information Retrieval. In *Proceedings of the RIAO Conference*. 266–274.

SMADJA, F. 1993. Retrieving collocations from text: Xtract. *Comput. Ling. 19*, 1.

TUDHOPE, D., BINDING, C., BLOCKS, D., CUNLIFFE, D. 2006. Query expansion via conceptual distance in thesaurus indexed collections. *J. Document. 62*, 4, 509–533.

VAN RIJSBERGEN, C. J. 1979. *Information Retrieval*. London: Butterworths.

WEEDS, J., WEIR, D., AND MCCARTHY, D. 2004. Characterizing Measures of Lexical Distributional Similarity. In *Proceedings of the COLING*.

WINTNER, S. 2004. Hebrew computational linguistics: Past and future. *Artif. Intell. Rev. 21*, 2.

XU, H. AND YU, B. 2010. Automatic thesaurus construction for spam filtering using revised back propagation neural network. *Expert Syst. Appl. 37*, 1, 18–23.

XU, J. AND CROFT, W. B. 1996. Query expansion using local and global document analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

YANG, D. AND POWERS, D. M. 2008. Automatic thesaurus construction. In *Proceedings of the 31st Australasian Conference on Computer Science (ACSC)*. 147–156.

Received May 2012; revised September 2012, December 2012; accepted January 2013