Mining the Web for Collocations: IR Models of Term Associations

Rakesh Verma, Vasanthi Vuppuluri, An Nguyen, Arjun Mukherjee, Ghita Mammar, Shahryar Baki and Reed Armstrong

Computer Science Dept.
University of Houston
Houston, TX 77204, USA
rmverma@cs.uh.edu, vvuppuluri@uh.edu, anqnguyen@outlook.com,
arjun@cs.uh.edu, ghita.mammar@gmail.com, sh.baki@gmail.com,
rmarmstr@gmail.com

Abstract. Automatic collocation recognition has attracted considerable attention of researchers from diverse fields since it is one of the fundamental tasks in NLP, which feeds into several other tasks (e.g., parsing, idioms, summarization, etc.). Despite this attention the problem has remained a "daunting challenge." As others have observed before, existing approaches based on frequencies and statistical information have limitations. An even bigger problem is that they are restricted to bigrams and as yet there is no consensus on how to extend them to trigrams and higher-order n-grams. This paper presents encouraging results based on novel angles of *general* collocation extraction leveraging statistics and the Web. In contrast to existing work, our algorithms are applicable to n-grams of arbitrary order, and directional. Experiments across several datasets, including a gold-standard benchmark dataset that we created, demonstrate the effectiveness of proposed methods.

1 Introduction

Automatic recognition of semantic associations is a serious challenge and collocations are no different in this regard. Although there is no widely-accepted definition of the word collocation, Mel'cuk has proposed a characterization and definition in [28]. We take a relatively broader view of collocations than his proposal, which separates out idioms and quasi-idioms. In this paper, collocations are arbitrarily restricted lexeme combinations such as *look into* and *fully aware*.[1] The origin of the word lies in British traditional linguistics. In this paper, we adopt the notion of collocation in its broadest sense, following Hoey and Colson: "Collocation has long been the name given to the relationship of a lexical item with items that appear with greater than random probability in its (textual) context," [21, 11].

As many have observed before, these special lexemes are recognized by native speakers as belonging together. In [11], the author states: since Hoey's definition is

[1] Our definition includes the semantic phrasemes of [28]

based on a statistical criterion, collocations are likely to correspond to a broad range of more or less fixed expressions such as compound proper nouns, compound nouns, compound terms, noun-adjective combinations, idioms, routine formulae, proverbs and sayings, quotations and even well-known song or film titles. We adjust this list as follows: we do not allow well-known song, book or film titles, and we add verb particle constructions (also called phrasal verbs or phrasal-prepositional verbs), compound verbs and light verb constructions. Another important consideration is whether subunits of collocations are considered collocations are not. Although at first sight it seems that (ordered) subunits of collocations should be considered collocations based on the statistical criterion, there could be some difference of opinion on units such as idioms and constructions such as *ad hoc*. Therefore, we report results for both options: unmodified, meaning subunits are not considered collocations and subcollocation, meaning subunits are considered collocations.

Computer recognition of collocations is an important task with many implications. For example, methods that can identify collocations can be appropriately extended to identify multi-word expressions and idioms. Recognition of collocations can significantly improve many important tasks, e.g., summarization [3, 42, 43], question-answering [4], language translation, topic segmentation [15], authorial style [22], and others.

However, automatic recognition of collocations is a challenging task for many reasons: their rarity even in large or very large corpora, they are often not modelizable by string patterns and the evolution of natural languages with some constructs falling out of favor and new constructs being added with societal changes and advances. Thus, the simple approach of building a large database of collocations and looking up each phrase will over time become obsolete. Statistics, machine learning and data mining techniques can be applied on large corpora for identifying collocations, e.g., [40, 7, 23, 39, 17, 46, 37]. The problems with this approach is finding a good threshold [46] and/or availability of labeled data. We highlight here a few of these and defer the rest for the Related Work Section. Xtract is based on statistical methods for retrieving and identifying collocations from large textual corpora [40] with an estimated precision of 80%. In [16], a semiautomatic method for extracting nested collocations is presented. Parsing and co-occurrences are used in [46, 37], but the authors admit that "it is difficult to determine a critical value above which a co-occurrence is a collocation and below which it is not" [46]. Moreover, no results are presented since a collocation reference subset ("gold standard") is not yet constructed.

Another approach would be to search the Web for every phrase in a given text. The Web is huge and contains all types of data, curated and uncurated. There are several hurdles that must be overcome for this approach to be successful: noise, rate limits imposed on queries, sensitivity of search results to small variations in the syntax of query, and results returned are number of page hits rather than number of occurrences of the search query [9]. A full treatment of the many issues involved is beyond the scope of this paper, but see [25], who argue that the advantages often outweigh the disadvantages.

We address automatic recognition of collocations and make the following contributions:

1. We present new collocation extraction algorithms that combine the advantages of the web along multiple dimensions with those of dictionary look-up and minimize their respective disadvantages (Section 2). Our algorithms are general, i.e., they work for arbitrary order n-grams and are directional in the sense of Gries [19]. Gries observes that a serious deficiency of many association-based collocation extraction methods is that they use measures that are symmetrical. In other words, the value of the measure is the same regardless of whether the phrase is *look into* or *into look*.
2. We demonstrate the performance of our algorithms on several datasets and compare the performance with that of several baselines including MWEToolkit, NSP and Gries's algorithm [19] (Section 3).
3. We create a gold-standard dataset derived from the Wiki50 dataset [44] that we will share with the NLP community. We explain the creation of this dataset in detail (Section 3.2). The creation of this dataset sheds light on the difficulty of manually annotating corpora for collocations (Section 3.4).
4. We present the performance of eight volunteers at the task of collocation extraction. These volunteers were computer science students with some being native speakers of English and some non-native speakers. None of them were experts in linguistics. The volunteers were asked to use dictionaries to look up the phrases as a matter of course. Even when equipped with the Oxford Dictionary of Collocations and Oxford Dictionary of Idioms, performance (F1-score) of the volunteers ranged from 39% to 70%.

2 Collocation Detection Algorithms

In most scenarios collocations tend to have a defined structure. Hence, we design two variants of the methods for extracting collocations, which help us observe the significance of parts-of-speech (POS).

Collocations without POS restrictions. This method ignores the POS of the components in the n-gram to determine whether it is a collocation.

Collocations with POS restrictions. A necessary condition for an n-gram to be a collocation is that the POS of at least one of its components belongs to {Noun, Adjective, Verb, Adverb}.

Although we provide two methods, the steps involved are essentially the same. The first component is splitting the text document into sentences and n-grams. Care must be taken in n-gram extraction to account for punctuation, and, of course, splitting into sentences is itself nontrivial because of abbreviations containing periods. After experiments with off-the-shelf NLP software, we use our own n-gram extractor.

2.1 Dictionary search

Our first method for collocation recognition is straightforward, viz., lookup. In this work, we used WordNet from NLTK corpus, even though it is small and limited in polylexical expressions, mainly because it is readily available.

Input for algorithms. *n*-grams extracted from text or given.

Algorithm 1 Collocation Extraction using Lookup

```
1: for each n-gram N do
2:     if N ∈ WordNet dictionary then
3:         N is a collocation
```

2.2 Web search for Title and URL

After searching WordNet, we then explore the largest source of data in determining if an n-gram N is a collocation - the Web. For this we do a phrase query of N using Bing search API[2] and retrieve the top 10 hits of the search. From each result retrieved, the title and URL are extracted. Now, the method checks if any word (substring) that is synonymous to the word 'dictionary', or any dictionary, is present in the title (URL). If the answer is yes, the method then checks if the exact match of N is present in the URL or if the stemmed components of N are present in the stemmed title. This is to avoid missing any component because of different inflectional forms. Snowball stemmer is used to stem the components. If a match is found, N is declared a collocation.

The two steps involved in this method ensure that the n-gram is not a random co-occurrence of lexemes. implying that the n-gram is a collocation. We note that the Bing search API used is not consistent in providing hit counts. Access to a stable web search API will improve this method.

Algorithm 2 Collocation Extraction using Web

```
1: for each n-gram N do
2:     Check top 10 search results (Titles/URLs) for words synonymous to 'dictionary'
3:     Titles = search titles that meet the requirement in line 2
4:     URLs = search URLs that meet the requirement in line 2
5:     if (N ∈ Titles) or (N ∈ URLs) then
6:         N is a collocation
```

We tested this method on six documents selected at random from the Wiki50 dataset using the Wiki50 annotations as gold standard. The F1-scores are in Table 1. We observe that this method alone achieves decent F1-scores, frequently better than 20%. Hence we decided to put this method second in the pipeline when the methods are sequenced.

We noticed that Wikipedia blocks requests after a certain number, and Wikipedia also appears the most in the dictionary websites. The problem is alleviated, however, because most Wikipedia URLs already contain the title.

2.3 Web search and substitution

Although the Web search method is often efficient, in some situations the top 10 results may not be sufficient to cover the diversity of myriad collocations. Hence, we use the following technique as a backup to determine if an n-gram N is a collocation. This

[2] https://datamarket.azure.com/dataset/bing/search

Table 1: F1 scores for Web Search on Title and URL

Document	**F1-score unmodified**	**F1-score subcollocation**
Bacteriological Water Analysis	0.2712	0.3552
Bearing an Hourglass	0.1176	0.2026
Budy Caldwell	0.2462	0.3754
Butch Hartman (racer)	0.2394	0.4040
Castlevania Chronicles	0.2006	0.2831
Myllarguten	0.1356	0.1875

method uses Bing Search API to obtain hit counts when a phrase query is formed from N. Then each word w in N is replaced by 5 random words that are of the same POS as w. This is done only for words whose POS is from {Noun, Adjective, Verb, Adverb} if we take POS into consideration. After each replacement, the n-gram with one of its words replaced is searched for in the web using Bing search API and the total number of search results returned is obtained. Once all the replacements are done and all search results, $\{S_{11}, S_{12}, S_{13}, S_{14}, S_{15}, S_{21}, ...S_{n5}\}$ are obtained, an average is computed as, S_{avg} of the non-zero S_{ij} values. The final step is to compare S_N against a suitably weighted S_{avg}, where the weight factor is a multiplicative constant c_{sub}. Note that since this method is based on hit counts for a phrase query, it is naturally directional in the sense of Gries.

Algorithm 3 Collocation Extraction using Web and Substitution

1: **for** each n-gram N **do**
2: S_N = Total hit counts for N (phrase query)
3: N' = new phrase obtained by replacing each word w in N with 5 randomly chosen words of same POS as w
4: SR = list(Total hit counts returned for each N')
5: S_{avg} = Average of non-zero values in SR
6: **if** $S_N > c_{sub} S_{avg}$ **then**
7: N is a collocation

For this method, we need to find the optimal value of c_{sub}, so we evaluated it on the document "Bearing an Hourglass," from Wiki50 without POS. First, exploring the range $[0,1]$ with 0.001 increment yielded F1-score lower than 3% for both versions: unmodified and subcollocation. Next we explored the range $[1,10001]$ with increment of 10. This gave the best F1-score between 6-7% for the subcollocation version and between 5 - 6% for the unmodified version. Based on the graph of F1-scores, we narrowed the search for c_{sub} to the interval $[1,2001]$. Using increment of 1 this interval was explored and the results then narrowed the search to the interval $[1,101]$. When this interval was explored in increments of 0.001 the best F1-scores of 5.65% and 6.69% were obtained at $c_{sub} = 43.7$ for unmodified version and $c_{sub} = 69.2$ for the subcollocation version.

For validation of these c_{sub} values a different article "Myllarguten" was selected. The optimal values of c_{sub} found above were tried and F1-scores of 1.15% (unmodified)

and 4.04% (subcollocation) for c = 43.7, and 1.18% (unmodified) and 3.78% (subcollocation) for c = 69.2 were obtained. These results are significantly worse than the results achieved for "Bearing an Hourglass," so we probed further in the range $[1,100]$. Again the F1-scores of between 6-7% for subcollocation and between 4-5% for the unmodified versions were observed for c_{sub} in the interval $[1,20]$. This suggests that: (i) perhaps our sample for tuning c_{sub} values may not be large enough. In other words, combining more articles into the training will give us a c_{sub} value that works better generally. (ii) The "randomness" inherent in the method may be affecting our search for the best c_{sub} value. It could make the best c_{sub} values vastly different for each run, and each article. Therefore, the best c_{sub} value found for only one training run may not be the best for the evaluation. While checking the F1-scores in these experiments, we noticed that the Wiki50 dataset annotations were not consistent with the Oxford Dictionary of Collocations and the Oxford Dictionary of Idioms so we deferred further experiments on the c_{sub} value to post gold-standard dataset creation, which is described below.

2.4 Web search and independence

This is another directional approach that does not use as many search queries as the above technique of Section 2.3. The idea of this method is to check whether the probability of a phrase exceeds the probability that we would expect if the words are independent. Hit counts are used to estimate these probabilities. There are two variants that differ in Line 8. The steps are described in Algorithm 4 below.

Algorithm 4 Collocation Extraction using Web and Independence - Method I

1: **for** each n-gram N **do**
2: $T(N)$ = Total hit counts for N
3: U_a = Universe of web pages containing 'a'
4: $P(N) = T(N)/U_a$
5: **for** each word w_i in N **do**
6: T(w_i) = Total hit counts for w_i
7: P(w_i) = $T(w_i)/U_a$ /* Prob. of w_i */
8: **if** $P(N) > f(n)\Pi_{i=1}^{n} P(w_i)$ **then**
9: N is a collocation

Method-2: The drawback of the first method is that it ignores word repetitions within the phrase, which we fix by modifying Line 8 of Algorithm 4. When the words in the n-gram are repeated, an adjustment is made based on the number of distinct permutations possible from words in the n-gram as follows. When the words in the n-gram are not unique: the n-gram is a collocation if

$$P(N) > f(n)\Pi_{j=1}^{k} n_j!\ \Pi_{i=1}^{n} P(w_i)/n!,$$

where $k =$ number of unique words, n_i = number of occurrences of the i^{th} word in the n-gram.

Optimizing the Independence Methods without POS - function $f(n)$. At first the inequalities for $P(N)$ in the above two algorithms were modified to introduce a similar constant c_{ind}, i.e., $f(n) = c_{ind}$, on the right hand side (RHS) as in the substitution method. However, the results were unsatisfactory. The problem is that the number of hit results for a single word is approximately 10 millions to billions, while the number of hit results for 'a' is 18 billions. The calculated probability of a single word, therefore, can be as low as 10^{-4}. When the length of the N-gram increases one word, the RHS of the inequality decreases by 10^{-4}, while the LHS decreases slightly. Example: Phrase "Bat Durston and the BEMS" had three hits while "Bat Durston and the" had four hits. As the RHS decreases too quickly compared to the LHS, we introduce a balancing function $f(n)$ that grows with the length of the n-gram fast enough to counter this effect. We chose the formula c^{n^p-1} for two reasons: the two parameters c and p give flexibility for optimization and $f(1) = 1$ ensures that unigrams cannot be flagged as collocations. A two-dimensional heat map (Figure 1) of F1-scores was constructed for c and p ranging in $[1,3]$ for the unmodified version of Method 1 first. The best F1-score of 11.25% was achieved at $p = 2.93$ and $c = 1.15$. The heat map pattern also shows that close to the highest score is achieved for other combinations of parameters as well.

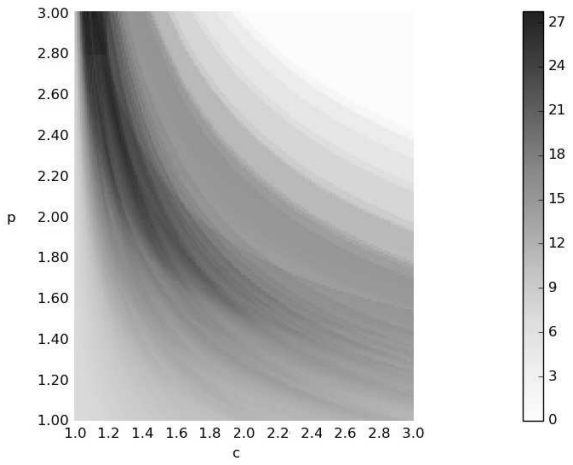

Fig. 1: Heat map for Unmodified Version

Next the heat map for the subcollocation version of Method 1 was constructed (Figure 2). The best F1-score of 12% was achieved at surprisingly the same combination of parameter values, $p = 2.93$ and $c = 1.15$. However, the pattern this time shows a narrower band of good parameter choices.

The heat maps for both versions of Method 2 are similar to those for Method 1 so we omit them here. The highest F-score increases a little bit to 11.30% for the unmod-

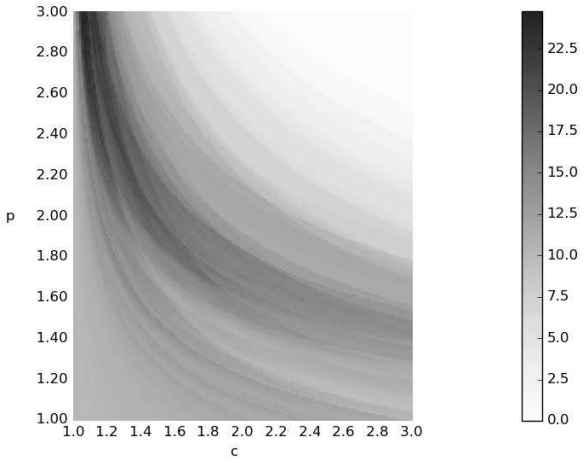

Fig. 2: Heat map for Subcollocation Version

ified version at $p = 3$ and $c = 1.21$. For the subcollocation version, the highest F-score increases a little bit to 12.15% at $p = 2.95$ and $c = 1.14$.

Validation on a different article. We ran both Independence methods without POS option using the three sets of p and c values obtained above on the article "Myllarguten." The F1-scores were even better (13.91% to 15.54% for $p = 2.93$ $c = 1.15$), (16.31% to 19.05% for $p = 3$ $c = 1.21$) and (13.79% to 15.31% for $p = 2.95$ $c = 1.14$) than those obtained for "Bearing an Hourglass." Further search for optimal values on the gold-standard dataset we created is described below.

3 Experimental Evaluation

This section details the experiments settings and results. We detail our metrics, the datasets used, comparison between Dataset 3 and Wiki50, the performance of our volunteers, and the performance results.

3.1 Baselines and Overall Methods

We use Mwetoolkit and NSP as our baselines. Our overall pipelines are: **Sub** - which executes Algorithms 1-3 in sequence, **T1** - Algorithms 1, 2 and 4 in sequence and **T2** - Algorithms 1, 2 and second variation of 4 that takes into account repetition of words.

3.2 Datasets

For the experiments, we used three different datasets extracted from several sources.

Dataset 1 A set of 400 collocations were extracted from listed websites based on POS structure. This dataset comprises 100 Adjective+Noun collocations from [14, 29], 100 Noun+Noun collocations from [45], 100 Verb+Noun collocations from [41, 5, 35, 12, 24] and 100 Verb+Preposition collocations from [13, 30, 18]. Each of these collocations is used as a test to verify the performance of the methods when a complete sentence is not given, and also to compare the performance of our methods with Mwetoolkit, which needs POS patterns of collocations to be extracted.

Dataset 2 This dataset is a collection of idioms obtained from the Oxford Dictionary of Idioms. The text file consisting of 1673 idioms is our input. Since all idioms are essentially MWEs and all MWEs are collocations, idioms would be the perfect choice for testing the software. Also, any non-ASCII characters are ignored while writing the idioms to the text file.

Dataset 3 This is a sentence dataset, which facilitates evaluation of the false positive rates of our methods and the baselines. Its creation was inspired by the discrepancies observed between the Oxford dictionaries and the Wiki50 annotations, while checking the F1-scores of our algorithms during the tuning of the parameters. A set of 100 sentences was selected at random from the Wiki50 dataset [44] and distributed to eight volunteers. Note that if two or more sentences were included in a single quotation, they were counted as a single sentence. Even though the Wiki50 dataset was annotated, we found that it was missing several collocations and a few idioms when we did a spot check with the Oxford Dictionary of Collocations and the Oxford Dictionary of Idioms. So the volunteers were asked to manually annotate all the collocations and the idioms in the 100 sentences using these two resources. Each volunteer was given 25 sentences and each sentence was given to exactly two volunteers.

The volunteers were given instructions on how to annotate since: (i) dictionaries contain abbreviations for generic pronouns such as *sth* for something or *sb* for somebody (ii) the verb forms can be different, e.g., dictionary may contain "make ones way" and sentence may contain "made his way" and (iii) there could be intervening words in the collocations.[3] After the volunteers completed the annotations, one of the co-authors resolved all the conflicts with dictionaries and also checked the phrases on which both volunteers agreed. Then, a different co-author went through all the sentences one more time looking carefully for false negatives and false positives. After this check, each volunteer was given feedback on the results and was asked to dispute the findings and find any remaining errors. This process ensured a final check of the results. After this final check, recall, precision and F1-score were calculated for each volunteer and the findings were compared with the Wiki50 dataset. We found that the Wiki50 annotators consistently annotated compound proper nouns and we do include these in our gold-standard collocations as well since we take the most general definition of collocation. Some examples of collocations identified by our process and not identified by the Wiki50 annotators include: "vowing to," "ardent supporter" and "be elected."

[3] These are also the reasons that automatic creation of gold standard datasets is difficult even if text data is extracted from the Dictionaries mentioned.

3.3 Volunteer Demographics and Performance on Dataset 3

The eight volunteers include two females (25%) and six males (75%). Three are native speakers of English and five speak English as a second language. They range in years from 18 to 25. All are students: one high school senior, one undergraduate senior, two MS, and four PhD students. Their F1-scores range from 39.21% (high-school senior, native speaker) to 70.87% (PhD student, native speaker).

3.4 Comparison with Wiki50 annotations

A total of 263 collocations were identified in the 100 sentences of which six are idioms. The Wiki50 annotators had identified 159 of these 263 collocations (recall = 60.46%) and missed four out of six idioms we found. Both collocation numbers (ours and Wiki50) include compound proper nouns and compound nouns, except named entities.

3.5 Parameter Value Optimization

Now that we have reasonable confidence[4] in our gold-standard, we undertook an optimization procedure for the parameters on the 100 sentence Dataset 3. The dataset was divided into 60% training, 20% held-out and 20% testing sets. The top three sets of parameter values from the training set for each method and its versions (POS, No POS) and (Unmodified, Subcollocation) were tried on the validation set and the winner proceeded to the test set. No significant difference was observed for POS versus No POS versions. Slight difference was observed for the Unmodified versus Subcollocation versions of the Independence methods and significant difference was observed for the the Unmodified (optimal $c_{sub} = 13.1$) and Subcollocation (optimal $c_{sub} = 92.0$) versions of the Substitution method and these values were stable across POS and No POS versions. Here, we present the results on all datasets with $c_{sub} = 13.1$ for all versions of Substitution method and $c = 1.14$, $p = 2.95$ for all versions of Independence method.

3.6 Results

For datasets 1 and 2, only recall is relevant since the input is the gold standard itself (precision = 100%). Tables 2 to 5 give results for Datasets 1 and 2. Table 6 gives the precision, recall and F1-scores for Dataset 3. The Sub pipeline gives the best recall on all datasets but lower precision. The T1 pipeline gives the best F1-score on Dataset 3. Another interesting trend is that the recall of our methods is usually better for idioms than for collocations overall.

3.7 Comparison with baseline parameter values

In the following Tables 7, 8, we report for comparison the recall on Datasets 1 and 2 that we get with the parameter values chosen so that they do not make any difference

[4] Note that we do not claim perfection, but we expect mistakes to be rare.

Table 2: Recall on Datasets 1 and 2, $c_{sub} = 13.1$, $c = 1.14$, $p = 2.95$

Recall	Sub		T1		T2	
	No POS	POS	No POS	POS	No POS	POS
Dataset1	0.744	0.744	0.736	0.736	0.739	0.739
Dataset2	0.828	0.823	0.826	0.825	0.831	0.819

Table 3: F1-scores on Datasets 1 and 2

F Score	Sub		T1		T2	
	No POS	POS	No POS	POS	No POS	POS
Dataset1	0.853	0.853	0.848	0.848	0.85	0.85
Dataset2	0.902	0.899	0.9	0.9	0.903	0.897

Table 4: Recall on Datasets 1 and 2 for Subcollocation Versions

Recall	Sub		T1		T2	
	No POS	POS	No POS	POS	No POS	POS
Dataset1	0.652	0.652	0.641	0.641	0.658	0.658
Dataset2	0.826	0.823	0.824	0.823	0.844	0.82

Table 5: F1-scores on Datasets 1 and 2 for Subcollocation Versions

F Score	Sub		T1		T2	
	No POS	POS	No POS	POS	No POS	POS
Dataset1	0.789	0.789	0.781	0.781	0.793	0.793
Dataset2	0.897	0.895	0.896	0.895	0.905	0.894

Table 6: Percentage Precision, Recall and F1-Score for Dataset 3, $c_{sub} = 13.1$, $c = 1.14$ and $p = 2.95$

Corpus	Tech.	Evaluation method	POS			No POS		
			Precision	Recall	F-Score	Precision	Recall	F-Score
Bing	Sub	Unmodified	11.12	58.45	18.6899	10.64	62.46	18.1818
		Subcollocation	13.27	69.21	22.274	13.26	70.79	22.3422
	T1	Unmodified	22.37	51.86	31.2608	19.98	55.59	29.3939
		Subcollocation	34.86	48.03	40.3985	33.22	51.84	40.4933
	T2	Unmodified	19.45	55.01	28.7425	17.76	58.74	27.2788
		Subcollocation	28.19	51.97	36.5571	27.55	55.53	36.8237

$c_{sub} = 1$ and $c = 1$. In Table 9 we present the results for dataset 3 with baseline parameter values. As expected, since parameters were optimized on Dataset 3 where both recall and precision matter, whereas only recall matters for Datasets 1 and 2, the results improve with baseline parameter values for these two datasets. A clear degradation is observable for Dataset 3 with baseline parameter values.

Table 7: Recall on Datasets 1 and 2 for Unmodified Versions, baseline parameters

Recall	Sub		T1		T2	
	No POS	POS	No POS	POS	No POS	POS
Dataset1	0.85	0.847	0.779	0.779	0.859	0.859
Dataset2	0.86	0.865	0.87	0.869	0.925	0.923

Table 8: Recall on Datasets 1 and 2 for Subcollocation Case, baseline parameters

Recall	Sub		T1		T2	
	No POS	POS	No POS	POS	No POS	POS
Dataset1	0.828	0.826	0.7	0.7	0.812	0.812
Dataset2	0.868	0.868	0.87	0.869	0.932	0.931

Table 9: Percentage Results on Dataset 3 with baseline parameters

Corpus	Tech.	Evaluation method	POS			No POS		
			Precision	Recall	F-Score	Precision	Recall	F-Score
Bing	Sub	Unmodified	5.4	82.24	10.13	5.4	82.24	10.13
		Subcollocation	8.06	89.61	14.8	8.06	89.61	14.8
	T1	Unmodified	4.76	59.6	8.81	4.84	63.32	9
		Subcollocation	5.7	87.89	10.7	5.7	88.16	10.7
	T2	Unmodified	2.73	66.48	5.24	2.8	70.2	5.38
		Subcollocation	4.82	96.18	9.17	4.82	96.32	9.18

Comparison: MWEToolkit, NSP and Gries's Delta P Although Mwetoolkit is described as MWE extraction software, the definition of an MWE in [33] aligns with our definition of collocation, hence, this is a valid comparison. MWEToolkit needs POS patterns of collocations to be able to extract them. For example, to extract a verb+noun phrase, it requires a pattern 'VN' to be declared prior to execution. So, we use Dataset-1 as input. The recall is 93%, the precision is 10.15% and the F-score is 18.31% even when MWEtoolkit is run on the four different types of collocations separately. The reason for the low precision is that, based on the patterns specified, MWEtoolkit takes the

POS tagged words as input and creates phrases. Often, new phrases not part of the input are created and then checked.

NSP provides many association measures for bigrams, a subset of four for trigrams and only log-likelihood (LL) for 4-grams. We used the four trigram measures for both bigrams and trigrams and the best possible thresholds on a development set consisting of 20 sentences of Dataset 3. The highest F-score was 18% for bigrams and 7% for trigrams with PMI, LL and PS. For 4-grams the highest F-score was 4%.

In [19], Gries proposed to use ΔP to differentiate bigram collocations from bigram non-collocations. Although he did not give any thresholds, we took a set of 25 bigram collocations at random from Dataset 3 and 15 bigram idioms at random from Dataset 2 and then found the threshold that gave the best F1-score (32.26%) on Dataset 3, which came out to 1. When this threshold was used for the same set of bigrams but with the British National Corpus corpus supplying the frequencies, the F1-score was 0, since Recall was 0. The threshold that gave the best F1-score of 1.0 for the British National Corpus was -0.4, which when used with the 100 sentences of Dataset 3 for supplying frequencies gave an F1-score of 17.8%.

4 Related Work

Collocation extraction has been well-studied. We include here the closest related work under the following threads:

Statistical measure based approaches: One of the classical methods of discovering collocations is to measure the association strengths of candidate n-grams. The key idea is to ascertain whether appearance of terms in an n-gram is more often than just random chance. In [6], significance of bigrams was computed by measuring the actual frequencies with expected frequencies using a normal approximation to the binomial distribution yielding the z-score. [10] used pointwise mutual information. In [38], an entropy method was proposed relying on the idea that collocations tend to be less noisy than non-collocating n-grams from an information-theoretic perspective. [40] proposed a threshold based approach that first discovers bigrams and then detects collocations based on a threshold and the context of nearby words (or their POS tags) appearing in the sentence. In [8], a technique based on log-odds ratio was proposed. In N-gram statistics package [2], a variety of association measures were implemented. While statistical measures have association evaluation strengths, they are quite dependent on the input corpus and no single method works well in discovering the whole gamut of collocations. In practice, a combination of measures renders better accuracy in collocation discovery [32]. Our proposed methods overcome the major problems with all these approaches, viz., sparsity and lack of directionality [19], by using the Web and devising new directional methods.

Parsing/Multi-lingual/MWEs/Idioms: Since we do not use any information obtained through parsing, these approaches [27, 31, 36, 47, 1, 46] are not directly comparable to ours. Our algorithms are quite general and suitable for extensions in the *multi-*

lingual context,[5] some of which was studied in [20, 17, 37]; and we note *approaches for MWEs/idioms:* [26, 34].

Parsing and dependency based approaches have also been explored. In [27], an information theoretic approach over parse dependency triples were proposed. [31] compares several techniques with his own in which he exploited synonym substitution via WordNet within parse dependency pairs. These could discover collocations such as "emotional baggage" from less frequently occurring phrases "emotional luggage" by substituting luggage by its synonym baggage. [36] explores syntax based approaches for collocation extraction. In [47] shallow syntactic analysis based on compositionality, substitutability, and modifiability statistics were leveraged to discover collocations. The method was evaluated on a specific cases of German PP-verb combinations. [1], proposed a lexical acquisition technique based on a dependency parser for extracting a verb-particle constructions (e.g., hand in, climb up, drop down, etc.) which are a special case of collocations. Although these methods have made progress, parsing based approaches tend to be sensitive on inherent threshold that need tuning which is often non-trivial and heuristic [46].

Another thread of research exploits aligning multi-lingual corpora for collocation extraction. In [20] lingual collocations were described from sentence-aligned parallel corpora. [17] focused on the special case of verb and its objective noun collocations in bilingual corpora. [37] proposed a framework based on deep syntactic parsing and rule-based machine translation for extracting lexical collocations form multi-lingual corpora. Methods based on syntactic tree-patterns need high quality and large coverage parsers.

Multiword expressions (MWEs) are a special case of collocations. They range over linguistic constructions such as fixed phrases (per se, by and large), noun compounds (telephone booth, cable car), compound verbs (give a presentation), idioms (a frog in the throat, kill some time), etc. [26] provides a review of linguistic and distributional characteristics of MWEs. [33] developed a system called mwetoolkit and implemented 4 measures (MLE, Dice, t-score, and PMI) for extracting MWEs following certain patterns (e.g., POS sequence patterns). In [34], a distributional similarity of each component word and the overall expression was used in predicting the compositionality of MWEs.

It is difficult to find any free software or research prototype that computes collocations. Xtract is no longer maintained, Collocate `http://www.athel.com/colloc.html` charges money, and we could not find the system in [37]. We have compared our work with mwetoolkit [33], which is based on user-defined criteria and association measures with counts obtained from Internet search results, and NSP [2]. For MWEtoolkit, the user must first run the Treetagger software on a text file and then process the output with a script in MWEtoolkit to generate an XML file. Then a DTD must be created for the generated XML file and then the XML file can be processed by

[5] Languages in which long words can be constructed by glueing together two or more lexemes or languages that have writing systems without word separators are likely to prove much more challenging.

MWEtoolkit to extract multi-word expressions,[6] which are subsets of collocations. NSP requires preprocessing of text, before constructing n-grams since otherwise it constructs n-grams that span sentences and include punctuation as a separate unit. For instance, ", hard" can be a bigram.

In [19], Gries criticized much of the previous work on using association measures for collocation detection because the measures are symmetrical. He then proposed ΔP for differentiating bigram collocations from bigrams that are not collocations. He did not propose any thresholds for his methods, and it is not clear how to extend them from bigrams to higher-order n-grams.

In [11], a web-based search method is proposed that relies on computing the proportion of exact matches of the *n*-gram in a sample of results that are returned by the API for Yahoo (100 when the paper was written) on a single query. This method requires "subtle manipulation" of the API according to the author. It also requires details of filters for tackling spamdexing and noise (essentially repetitions of lines and paragraphs by the search engine), which are omitted by the author. With these clever techniques in place, the technique yields high recall and precision according to the author when the threshold is chosen by analyzing an unspecified number of collocations selected from a dictionary. The recall is calculated on a set of 3,807 collocations and the precision is calculated on a subset of 5-grams from Google's n-gram collection.

5 Conclusion

We have presented new approaches for detecting collocations that combine the advantages of look-up and Web and minimize their disadvantages. Two other advantages of our approach are that it can be extended to other languages based on availability of a WordNet or dictionary for that language and Web search for it, and, in contrast to approaches such as Mwetoolkit, our approach can be used to directly check phrases without requiring the context. Results of our approach are demonstrated on a variety of test sets including a gold-standard sentence dataset that has been created. We also report on the performance of human volunteers and shed some light on the difficulty of creating collocation datasets manually. Our independence algorithm is within the range of the human volunteers and shows promise for the future. More work is needed to make the Substitution approach robust.

Acknowledgments

We thank the anonymous reviewers for their constructive comments, which have improved the paper. Rakesh's and Vasanthi's research was supported in part by NSF grants DUE 1241772, CNS 1319212, and DGE 1433817. Arjun's research is supported in part by NSF grant CNS 1527364. Reed's research was supported by NSF grant IIS 1359199. Ghita's research was supported by the University of Houston. An Nguyen and Arjun Mukherjee contributed equally to this paper. The authors thank Luis Moraes, Daniel

[6] The term polylexical expressions is preferred by some researchers since it removes reliance on the ill-defined concept of a word.

Lee, Avisha Das, Arthur Dunbar, Nirmala Rai and Suvedh Srikanth for participating in the gold-standard creation.

References

1. T. Baldwin. Deep lexical acquisition of verb-particle constructions. *Computer Speech & Language*, 19(4):398–414, 2005.
2. S. Banerjee and T. Pedersen. The design, implementation, and use of the ngram statistics package. In *CICLing*, pages 370–381, 2003.
3. A. Barrera and R. Verma. Combining syntax and semantics for automatic extractive single-document summarization. In *CICLING*, volume LNCS 7182, pages 366–377, 2012.
4. A. Barrera, R. Verma, and R. Vincent. Semquest: University of houston's semantics-based question answering system. In *Text Analysis Conference*, 2011.
5. `http://www.bbc.co.uk/worldservice/learningenglish/grammar/learnit/learnitv351.shtml`, 2015.
6. G. Berry-Rogghe. The computation of collocations and their relevance in lexical studies. *The computer and literary studies*, pages 103–112, 1973.
7. D. Biber. Co-occurrrence patterns among collocations: A tool for corpus-based lexical knowledge acquisition. *Computational Linguistics*, 19(3):531–538, 1994.
8. D. Blaheta and M. Johnson. Unsupervised learning of multi-word verbs. In *39th ACL and 10th EACL*, pages 54–60, 2001.
9. I. A. Bolshakov, E. I. Bolshakova, A. P. Kotlyarov, and A. F. Gelbukh. Various criteria of collocation cohesion in internet: Comparison of resolving power. In *Computational Linguistics and Intelligent Text Processing, 9th International Conference, CICLing 2008, Haifa, Israel, February 17-23, 2008, Proceedings*, pages 64–72, 2008.
10. K. W. Church and P. Hanks. Word association norms, mutual information and lexicography. In *27th Annual Meeting of the Association for Computational Linguistics, 26-29 June 1989, University of British Columbia, Vancouver, BC, Canada, Proceedings.*, pages 76–83, 1989.
11. J.-P. Colson. Automatic extraction of collocations: a new web-based method. *Proceedings of JADT*, pages 397–408, 2010.
12. `http://www2.elc.polyu.edu.hk/CILL/eap/2004/u5/verbs\&nounspart2.htm`, 2015.
13. `http://www.englishclub.com/vocabulary/ist.htm`, 2015.
14. `http://esl.about.com/od/grammarstructures/a/g_intadj_2.htm`, 2015.
15. O. Ferret. Using collocations for topic segmentation and link detection. In *COLING*, 2002.
16. K. T. Frantzi and S. Ananiadou. Extracting nested collocations. In *COLING*, pages 41–46, 1996.
17. F. Fukumoto, Y. Suzuki, and K. Yamashita. Retrieving bilingual verb-noun collocations by integrating cross-language category hierarchies. In *COLING*, pages 233–240, 2008.
18. `http://grammar.ccc.commnet.edu/grammar/phrasals.htm(07/23/2014)`, 2014.
19. S. T. Gries. 50-something years of work on collocations: what is or should be. *International Journal of Corpus Linguistics*, 18(1):137–166, 2013.
20. M. Haruno, S. Ikehara, and T. Yamazaki. Learning bilingual collocations by word-level sorting. In *16th International Conference on Computational Linguistics, Proceedings of the Conference, COLING 1996, Center for Sprogteknologi, Copenhagen, Denmark, August 5-9, 1996*, pages 525–530, 1996.
21. M. Hoey. *Patterns of lexis in text*. Oxford University Press, 1991.

22. D. L. Hoover. Frequent collocations and authorial style. *LLC*, 18(3):261–286, 2003.
23. S. Ikehara, S. Shirai, and H. Uchino. A statistical method for extracting uninterrupted and interrupted collocations from very large corpora. In *COLING*, pages 574–579, 1996.
24. http://www.johnsesl.com/templates/grammar/collocations.php, 2015.
25. A. Kilgarriff and G. Grefenstette. Introduction to the special issue on the web as corpus. *Computational linguistics*, 29(3):333–347, 2003.
26. V. Kordoni and M. Egg. Robust automated natural language processing with multiword expressions and collocations. In *ACL (Tutorial Abstracts)*, pages 7–8, 2013.
27. D. Lin. Extracting collocations from text corpora. In *First Workshop on Computational Terminology*, 1998.
28. I. Mel'ćuk. Collocations and lexical functions. *en Anthony Paul COWIE (ed.)(2001 [1998])*, pages 23–54, 1998.
29. http://www.myenglishteacher.eu/blog/, 2015.
30. http://www.oday.com/erbs/erbs_A.html, 2015.
31. D. Pearce. A comparative evaluation of collocation extraction techniques. In *Proceedings of the Third International Conference on Language Resources and Evaluation, LREC 2002, May 29-31, 2002, Las Palmas, Canary Islands, Spain*, 2002.
32. P. Pecina. An extensive empirical study of collocation extraction methods. In *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*, 2005.
33. C. Ramisch, A. Villavicencio, and C. Boitet. Multiword expressions in the wild? the mwetoolkit comes in handy. In *COLING (Demos)*, pages 57–60, 2010.
34. B. Salehi, P. Cook, and T. Baldwin. Using distributional similarity of multi-way translations to predict multiword expression compositionality. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014, April 26-30, 2014, Gothenburg, Sweden*, pages 472–481, 2014.
35. http://www.scielo.cl/pdf/signos/v45n78/a03.pdf, 2015.
36. V. Seretan. Syntax-based extraction. In *Syntax-Based Collocation Extraction*. Springer Netherlands, 2011.
37. V. Seretan. A multilingual integrated framework for processing lexical collocations. In *Computational Linguistics - Applications*, pages 87–108. Springer - Netherlands, 2013.
38. S. Shimohata, T. Sugio, and J. Nagata. Retrieving collocations by co-occurrences and word order constraints. In *ACL*, pages 476–481, 1997.
39. S. Shimohata, T. Sugio, and J. Nagata. Retrieving domain-specific collocations by co-occurrences and word order constraints. *Computational Intelligence*, 15:92–100, 1999.
40. F. A. Smadja. Retrieving collocations from text: Xtract. *Computational Linguistics*, 19(1):143–177, 1993.
41. http://www.stgeorges.co.uk/blog/ack/, 2015.
42. R. M. Verma, P. Chen, and W. Lu. A semantic free-text summarization system using ontology knowledge. In *Document Understanding Conference*, 2007.
43. R. M. Verma, D. Kent, and P. Chen. Ontology driven multi-document summarization. In *Text Analysis Conference*, 2008.
44. V. Vincze, I. Nagy, and G. Berend. Multiword expressions and named entities in the wiki50 corpus. In *RANLP*, pages 289–295, 2011.
45. http://www.vocabulary.com/lists/201117\#view=definitions\&word=cross\%20hair, 2015.
46. E. Wehrli. Parsing and collocations. In *Natural Language Processing*, pages 272–282, 2000.
47. J. Wermter and U. Hahn. Collocation extraction based on modifiability statistics. In *COLING 2004, 20th International Conference on Computational Linguistics, Proceedings of the Conference, 23-27 August 2004, Geneva, Switzerland*, 2004.