Mining Linguistic Cues for Query Expansion: Applications to Drug Interaction Search

Sheng Guo
Department of Computer Science
Virginia Tech, Blacksburg, VA 24061, USA
guos@cs.vt.edu

Naren Ramakrishnan
Department of Computer Science
Virginia Tech, Blacksburg, VA 24061, USA
naren@cs.vt.edu

ABSTRACT

Given a drug under development, what are other drugs or biochemical compounds that it might interact with? Early answers to this question, by mining the literature, are valuable for pharmaceutical companies, both monetarily and in avoiding public relations nightmares. Inferring drug-drug interactions is also important in designing combination therapies for complex diseases including cancers. We study this problem as one of mining linguistic cues for query expansion. By using (only) positive instances of drug interactions, we show how we can extract linguistic cues which can then be used to expand and reformulate queries to improve the effectiveness of drug interaction search. Our approach integrates many learning paradigms: partially supervised classification, association measures for collocation mining, and feature selection in supervised learning. We demonstrate compelling results on using positive examples from the DrugBank database to seed MEDLINE searches for drug interactions. In particular, we show that purely data-driven linguistic cues can be effectively mined and applied to realize a successful domain-specific query expansion framework.

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]: Linguistic processing; H.3.3 [**Information Search and Retrieval**]: Query expansion

General Terms

Algorithms

Keywords

Text mining, domain-specific query expansion, partially supervised classification, collocation, SVM, syntactic parsing

1. INTRODUCTION

Given a drug under development, what are other drugs or biochemical compounds that it might interact with? History abounds with instances where this question had been investigated incompletely, causing severe personal, monetary, and professional losses. Two recent examples serve to illustrate this aspect well. In Sep 2004, drug maker *Merck* voluntarily recalled their anti-inflammatory drug *Vioxx* because they discovered that patients taking Vioxx faced heightened risk of heart attacks. In May 2009, it was discovered that older men using the *Flomax* drug (to address their urinary tract problems) are twice as likely to experience 'floppy iris syndrome,' a condition that can cause inflammation around the eye, retinal detachment, and other serious side effects [2]. Thus there is great interest in mining for drug interactions before product development, clinical trials, and market releases.

Formally, a drug interaction can be defined as "the pharmacological or clinical response to the administration or co-exposure of a drug with another substance that modifies the patient's response to the drug" [38]. While the intent of inferring drug interactions is often to avoid them, sometimes it is actually desirable to encourage such interactions. For instance, in treating complex diseases such as cancers [35], multiple, or multi-component, drugs are usually administered in order to enhance combinatorial selectivity [12]. Here the different components work cooperatively to inhibit (or activate) a protein or protein network of interest in a diseased tissue, but are not particularly toxic for normal tissues.

Many computer-aided systems for exploring drug interactions have been established for clinical decision making. One such system [20] uses the CYP3A cytochrome (a key component of many drug metabolism pathways) as the focal pont to rank potential drug interactions. Another interesting system exploits the inherent network structure of drug interactions [28]. The key issue in these systems (and all others) is the *completeness* of their drug interaction information and ensuring that they stay current with published literature. On one hand, it is inconvenient for a clinician to search the scattered literature for information about a specific drug. On the other, database maintainers are concerned with issues of soundness, coverage, and trustworthiness.

The aim of our work is, hence, to design a framework for automatically identifying drug interaction sentences from large online corpora. Given a specific drug name such as *darbepoetin alfa*, a naive search for *darbepoetin alfa* in public corpora often gives a poor starting point, yielding hits such as the usage of *darbepoetin alfa* in the treatment of *anaemia*. By using state-of-the-art query expansion algorithms [30, 31, 44, 45], *darbepoetin alfa* gets expanded to *darbepoetin alfa anemia* or *darbepoetin alfa cancer*, which still leads to results of usage and is unsuitable for drug interaction identification. Even a manually formulated query, like *darbepoetin alfa interaction* or *darbepoetin alfa use together*, gives the result such as "Recombinant human epoetin beta in the treatment of renal anemia"[1] – they still do not cover drug interactions very well. In this paper, we turn to designing a domain-specific query

CIKM'09, November 2–6, 2009, Hong Kong, China.

[1] http://www.ncbi.nlm.nih.gov/pubmed/18488073

expansion capability (similar in spirit to [13, 24, 32]) by mining linguistic cues to combat this lack of specificity.

There are several research issues to be considered. Admittedly, one way to expand queries is to 'learn to expand,' i.e., to mine linguistic cues from a training dataset and use them to expand future queries. However to conduct such learning, negative examples are required and these are quite rare to come by. So the first research issue is to be able to work with only positive examples. Second, the precise nature of linguistic cues can involve just a single term (e.g., *darbepoetin alfa* can be expanded to *darbepoetin alfa coadministration* or a complete collocation (e.g., *darbepoetin alfa* might get expanded to *darbepoetin alfa concomitant use*). Mining cues and collocations is the second research issue and association measures have to take into account the complexity of syntactic constructs in the published literature. Finally, in order for the mined cues and collocations to be successfully used, they will have to be ranked so that they prominently accompany positive examples but not negative examples. Feature selection to help separate out the cues is hence important.

Our primary contributions are:

1. We present a domain-specific solution to drug interaction search that fuses multiple learning paradigms: partially supervised classification, association measures for collocation mining, and feature selection in supervised learning. These paradigms are used, respectively, to overcome lack of negative examples, to identify promising cues and associations between them, and to rank cues so that they lead to confirming instances of drug interactions. We highlight here that our approach works without requiring any external thesauri or ontology. Nevertheless, we refer to it as a domain-specific solution due to its reliance on corpora specific to the domain and because we haven't evaluated its utility in other domains.

2. We propose a new collocation mining approach which utilizes both positive and negative datasets. Hence the cues we mine are not just co-located but also possess good discriminative properties.

3. We apply our framework on the DrugBank [2] database (which features only positive instances of drug interactions) and show how it can be used to seed MEDLINE [3] searches for drug interactions. To the best of our knowledge, this is the first integrated framework for query expansion in drug interaction search.

2. FRAMEWORK OVERVIEW

Fig. 1 gives an overview of our framework which is based on text mining and data-driven collocation techniques. As stated earlier, we use linguistic cues for query expansion and aim to identify new drug interaction sentences which can then seed DrugBank evolution. We begin by using existing drug interaction description sentences in DrugBank as a positive dataset, and search MEDLINE with only drug names as queries. We then aim to construct a negative dataset as a subset of the search results. Since a manually labeled negative dataset is expensive, we construct the negative dataset by using the partially *supervised classification* method, specifically the 'learning from positive and unlabeled examples' paradigm (a.k.a. the LPU method)[4] [22]. LPU is a classification system based on positive and unlabeled datasets, and it learns

[2]http://www.drugbank.ca

[3]http://www.nlm.nih.gov/

[4]http://www.cs.uic.edu/∼liub/LPU/LPU-download.html

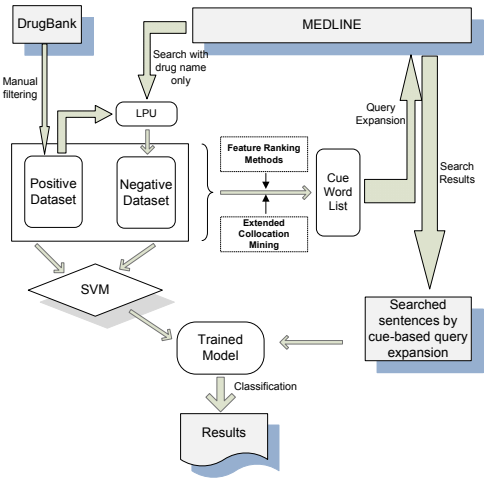


Figure 1: Flowchart for our drug interaction identification system.

the negative examples automatically. These positive and negative datasets serve as starting points for two types of analysis, as shown in Fig. 1. First, based on these training datasets, we design a novel algorithm to extract cue words that serve as confirming evidences for drug interaction relations. We use text feature selection approaches to extract single-term cues. We also extract multi-word cues by using and assessing 13 different association measures for collocation mining. All these cue words are used to query-expand, to help search for new drug interactions in MEDLINE. Meanwhile, the new search results from the expanded queries are segmented into sentence level units and classified by a state-of-art SVM, which is trained on the initial data sets (this is their second use, referred to earlier). Experimental results show that our system works in an effective and efficient manner. We perform multi-faceted evaluations. For instance, we show that the negative training dataset automatically generated is comparable to a manually labeled one and any differences do not have any influence on the accuracy of the overall mining process. We also compare our findings with information about existing drug interactions in DrugBank.

3. RELATED RESEARCH

Query expansion has a rich history of background research. Traditional query expansion uses term relationships between the original term and the expanding term. Global analysis [1, 17, 36] employs global statistical information gathered based on co-occurrence information from the entire collection, whereas local analysis [30, 31, 44, 45] is conducted using approaches modeled after pseudo-relevance feedback, i.e., using the top ranked relevant documents. Some other approaches [4, 7, 14, 42] mine logs of past query usage to construct term relationships. Significant work has also gone into incorporating external knowledge [3, 15, 23, 25]. A review of ontology-based query expansion is given in [3]. Works such as [19, 25, 41] use collocation techniques from NLP which are also based on term-term co-occurrence information. In many of these works, frequency of co-occurrence is often used as a surrogate for relevance and this leads to both false positives and false negatives [33]. This is especially true in drug interaction search. A detailed comparison with our approach and other query expansion methods is shown in Table 1. As is clear we aim for an automated, agnostic method that exploits local linguistic cues. The training imparted

Table 1: Comparison of our approach with other query expansion methods.

	Use local analysis	No user interaction	No external knowledge	Use linguistic property	Domain-specific
[30, 31, 44, 45]	✓				
[4, 7, 14, 42]	✓		✓		
[19, 41]		✓	✓	✓	
[24]	✓	✓			✓
[13, 32]	✓	✓	✓		✓
Our method	✓	✓	✓	✓	✓

to our system from drug interaction sentences endows it with the domain-specific query expansion facility. Other domain-specific approaches such as [13, 24, 32] are either focused on different application needs and/or they do not exploit collocations as heavily as done here.

4. MINING LINGUISTIC CUES

The most important part of our system involves the extraction of linguistic cues which is covered in detail here.

4.1 Single-term cue word extraction

We employ three criteria functions to rank all the tokens/terms appearing in the whole dataset. We classify sentences into two categories: positive and negative, according to whether they contain any drug interaction information. We consider the top 50 tokens ranked by each method as our single-term cues.

Mutual information (MI): The mutual information between term t_i and category c is given by:

$$MI\left(t_i, c\right) = \sum_{t_i \in \{0,1\}} \sum_{c \in \{+,-\}} P\left(t_i, c\right) \log \frac{P\left(t_i, c\right)}{P\left(t_i\right) P\left(c\right)}$$

Fisher kernel: We use the variant of the Fisher kernel as defined in [5], where the *F-score* for the *i*-th term is defined as:

$$F\left(i\right) = \frac{\left(\overline{x}_i^{(+)} - \overline{x}_i\right)^2 + \left(\overline{x}_i^{(-)} - \overline{x}_i\right)^2}{\frac{1}{n_+ - 1} \sum_{j=1}^{n_+} \left(x_{j,i}^{(+)} - \overline{x}_i^{(+)}\right)^2 + \frac{1}{n_- - 1} \sum_{j=1}^{n_-} \left(x_{j,i}^{(-)} - \overline{x}_i^{(-)}\right)^2}$$

Here, n_+ and n_- are the number of positive and negative sentences in the training dataset, $x_{j,i}^{(+)}$ and $x_{j,i}^{(-)}$ represent the *i*-th feature value in the *j*-th sentence in the positive and negative datasets, respectively. $\overline{x}_i, \overline{x}_i^{(+)}, \overline{x}_i^{(-)}$ are the average values of the *i*-th term's weight computed across the whole dataset, the positive dataset, and the negative dataset, respectively. The term weighting scheme thus influences the Fisher kernel; in our experiments, we use a simple binary weighting scheme so that the weight for a term is 1 if it appears in the sentence, 0 otherwise.

Relative frequency: The relative frequency *rf* for the *i*-th term was proposed in [21] as a ranking function:

$$\mathrm{rf}_i = \log(2 + \frac{a_i}{c_i})$$

where a_i is the number of positive sentences in which this term appears, and c_i is the number of negative sentences it appears in.

4.2 Bigram cue word generation

Multi-word cues, such as bigrams, require more sophisticated means than presented above. We extend *collocation* mining approaches from NLP to extract bigram cue words, ensuring properties of both **coexistence** and **discriminativeness**. Coexistence requires that the two terms in a bigram cue co-occur with each other a lot, while discriminativeness means these two terms in the bigram can be used to identify the target sentences.

4.2.1 Collocation

Collocation, as a linguistic concept, was introduced by J. R. Firth [11]. There are multiple definitions for what a collocation is, drawing upon different perspectives and the needs of specific applications. Smadja views collocations as lexical clusters that are domain-specific, context-recurrent, and cohesive [40]. Manning and Schutze [26] mention three attributes of typical collocations: *non-compositionality*, *non-substitutability*, and *non-modifiability*. Wermter and Hahn [43] provide a useful grouping of collocations into three classes:

- *Idiomatic Phrases*. In this class, the meaning of the collocation cannot be determined by the literal definition of the phrase components themselves, but refers to a metaphorical or figurative one. For example, the figurative meaning of "Wall Street" is the American financial market, and is unconnected to either 'Wall' or 'Street.'

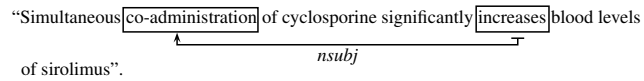
- *Support Verb Constructions/Narrow Collocations*. In this class, at least one component contributes to the overall meaning of the collocation expression. For example, "Sunday driver" means one who drives slowly.

- *Fixed Phrases*. This class denotes expressions whose components are all involved in the contribution of the overall meaning (e.g., the collocation "Christmas Eve").

A collocation useful for drug interaction search is "*concomitant administration*", where the occurrence of the collocation with a drug name is likely evidence of a drug interaction sentence.

The typical way to study collocations within a text dataset is to measure the co-occurrence frequency using statistical association measures [9] and use this information to identify three categories: *surface co-occurrence*, *textual co-occurrence*, and *syntactic co-occurrence*. Surface co-occurrence involves two words that appear within a certain distance, textual co-occurrence requires that the two words appear within the same textual unit, and syntactic co-occurrence requires the existence of syntactic relations between the terms. In our study, we use the restrictive notion of syntactic co-occurrence since it avoids setting a arbitrary distance for surface co-occurrence, and also avoids indirect and accidental co-occurrences. Further, compared to superficial co-occurrences, terms that are physically close to each other could be linguistically far apart, and hence should not be considered together as co-occurrence, whereas terms that are linguistically close could be physically remote. In the sentence 'Simultaneous co-administration of cyclosporine significantly increases blood levels of sirolimus', the word"co-administration" is not close to the word "increases",

but serves as a nominal subject, and this combination is quite frequent in drug interaction sentences (see below).



4.2.2 Constructing co-occurrence databases from dependency parses

To obtain syntactically bound co-occurrence information, we utilize dependency parsing, specifically using the Stanford typed dependency parser [5] [27]. We use collapsed typed dependencies (a.k.a. grammatical relations) to represent specific co-occurrence patterns, such as *adjective + noun*, *adverb + verb*, etc. However, we do not retain all possible collapsed typed dependencies, but only those with grammatical relations like *obj*, *dobj*, *iobj*, *pobj*, *subj*, *nsubj*, *nsubjpass*, *csubj*, *csubjpass*, *amod*, *advmod*, and *nn* [6]. A typical dependency parse of a sample sentence from our positive dataset is shown in Fig. 2.

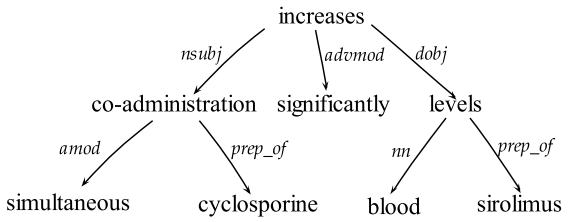


Figure 2: Dependency parse from the positive sentence "Simultaneous co-administration of cyclosporine significantly increases blood levels of sirolimus".

We treat grammatical relations as 2-tuple co-occurrence pairs, and store them into a co-occurrence database as candidates for bigram cues. We build 2-tuple co-occurrence records for both positive and negative datasets. As an example, 2-tuples from a sample sentence are shown in Table 2. Moreover, all the words are stored in lemmatized format, since words in the sentences could be in inflected forms. Lemmatization is used (e.g., *synchronized* is lemmatized to *synchronize*) to conflate these forms to a single bigram.

Table 2: The 2-tuple co-occurrence database with a positive sample sentence. "+" means the sentence comes from the positive dataset.

Sent ID	Term 1	Term 2	Dataset
1	co-administration	Simultaneous	+
1	increase	co-administration	+
1	co-administration	cyclosporine	+
1	increase	significantly	+
1	level	blood	+
1	increase	level	+
1	level	sirolimus	+
⋮	⋮	⋮	⋮

[5] http://nlp.stanford.edu/software, V1.6

[6] http://nlp.stanford.edu/software/dependencies_manual.pdf

4.2.3 Extended collocation model

By parsing all sentences from the training datasets, we create a syntactic co-occurrence database consisting of 2-tuples as candidates. We propose an extended collocation model to extract bigram cues from the co-occurrence databases. As we mentioned earlier, words in a bigram cue must have both co-existence and discriminativeness properties. Since traditional collocation mining approaches only guarantee the two terms co-exist with each other, we extend these approaches to make cues useful for identifying specific drug interaction relations.

We look beyond simple co-occurrence frequency and evaluate various association measures that scale it w.r.t. marginal frequencies. In overall, we investigate 13 association measures: the baseline co-occurrence frequency from the positive dataset and the 12 association measures from Table 5.

The measures from Table 5 can be understood in the context of contingency tables [9]. Table 3 depicts the traditional contingency table capturing overlaps in occurrences of two terms across sentences. To account for both positive and negative instances of drug interaction sentences, we create two virtual words: "**pos**" and "**neg**", and we make the assumption that any 2-tuple in the positive co-occurrence database automatically collocates with "pos", and that any 2-tuple in the negative database virtually collocates with the term "neg". Thus we 'lift' the traditional bigram contingency table into a trigram table as in Table 4. The entries of this table—i.e., observed and expected frequencies—directly feed into the definitions of the association measures (see Table 5).

Table 3: Traditional contingency table for measuring collocation between w_1 and w_2.

	w_2	$\neg w_2$	$TOTAL$
w_1	O_{11}	O_{12}	R_1
$\neg w_1$	O_{21}	O_{22}	R_2
$TOTAL$	C_1	C_2	N

Table 4: Extended contingency table with virtual words "pos" and "neg". For O_{ijk}, *i, j, k* means w_1, w_2 and virtual words, respectively.

	"pos"	*"neg"*	$TOTAL$
$w_1\ w_2$	O_{111}	O_{112}	R_{11}
$w_1 \neg w_2$	O_{121}	O_{122}	R_{12}
$\neg w_1\ w_2$	O_{211}	O_{212}	R_{21}
$\neg w_1 \neg w_2$	O_{221}	O_{222}	R_{22}
$TOTAL$	C_+	C_-	N

4.3 N-gram cues (n≥ 3)

A trigram or other higher order n-gram consists of n words. Although these words are not required to be sequential or next to each other, we require that any two of them are connected by a valid grammatical relation. The process of finding trigrams and other high order *n*-grams is similar to the one described earlier for bigrams. We first construct *n*-tuple co-occurrence databases, recording the *n*-gram candidates and their frequencies in both positive and negative datasets. We then import the same concept of virtual words "pos" and "neg" and change the *n*-gram cue mining problem to an *(n+1)*-gram collocation mining problem. We use association

Table 5: Modified association measures for mining bigram cues from the extended trigram contingency table (by adding the virtual words as the third term)

No.	Association Measures
1)	$\chi^2_{Yates}=\sum_{ijk}\frac{(\lvert O_{ijk}-E_{ijk}\rvert-0.5)^2}{E_{ijk}}$
2)	*t-score*$=\frac{O_{111}-E_{111}}{\sqrt{O_{111}}}$
3)	*log-likelihood*$=2\sum_{ijk}O_{ijk}\cdot\log\frac{O_{ijk}}{E_{ijk}}$
4)	*poisson-stirling*$=O_{111}\cdot(\log\frac{O_{111}}{E_{111}}-1)$
5)	*average-MI*$=\sum_{ijk}O_{ijk}\cdot\log_2\frac{O_{ijk}}{E_{ijk}}$
6)	*pointwise-MI*$=\log_2\frac{O_{111}}{E_{111}}$
7)	*local-MI*$=O_{111}\cdot\log_2\frac{O_{111}}{E_{111}}$
8)	$MI^2=\log_2\frac{O^2_{111}}{E_{111}}$
9)	$MI^3=\log_2\frac{O^3_{111}}{E_{111}}$
10)	*dice*$=\frac{3O_{111}}{(R_{11}+R_{12})+(R_{11}+R_{21})+C_+}$
11)	*jaccard*$=\frac{O_{111}}{N-O_{222}}$
12)	*odds-ratio*$=\log\frac{(O_{111}+0.5)(O_{221}+0.5)(O_{122}+0.5)(O_{212}+0.5)}{(O_{121}+0.5)(O_{211}+0.5)(O_{112}+0.5)(O_{222}+0.5)}$

measures to rank all these candidates, and high scoring collocations will be selected as valid cues. While the construction of bigram candidates was relatively straightforward, the construction of *n*-gram cues is more involved.

N-tuple concatenation: We construct *n*-gram candidates based on the original grammatical relations, which consist of only two words in each relation. When two grammatical relations share one word, they can be linked using the shared word as a pivot. This word linkage method is also used in [39] to mine traditional collocations from a single dataset. Without additional restrictions, the number of *n*-tuples using the word linkage method could be prohibitively large. Toward this end, we require that all concatenations be performed in the *same* sentence, i.e., only grammatical relations from the same sentence can be linked. This restriction immensely prunes the number of candidate *n*-tuples. We give an example of the concatenated 3-tuples in co-occurrence database in Table 6. The eight 3-tuples are constructed based on the seven 2-tuples in Table 2. As we can see, the number of the 3-tuples does not increase too much due to the single sentence restriction.

For the *n*-tuple concatenation case, we generalize in the obvious way the 3-tuple example above: *n*-tuples are derived from the lower order *(n-1)*-tuple lists again maintaining the sentence restriction, see Alg. 1. For example, *simultaneous co-administration* in Fig. 2 can be extended to the candidates of *simultaneous co-administration cyclosporine* or *simultaneous co-administration increase*. At each level, by introducing the virtual words of "pos" and "neg", the *n*-gram (n≥3) cue mining problem is lifted into a *(n+1)*-gram collocation extraction problem. The contingency table is extended similarly and the association measures from Table 5 are similarly generalized. For example, all the association measures in Table 5 for the bigram cue mining can be easily extended for the trigram: For the equations 1, 3, 5, we still process all the cells in the contingency table in the same way; For 2,4,6,7,8 and 9, we only need to replace O_{111} and E_{111} with the observed frequency and expected

Table 6: The 3-tuple co-occurrence database with a positive sample sentence. "+" means that this sentence comes from the positive dataset, and SID is short for "sentence ID".

SID	Term 1	Term 2	Term 3	Dataset
1	Simultaneous	co-administration	cyclosporine	+
1	Simultaneous	co-administration	increase	+
1	co-administration	cyclosporine	increase	+
1	co-administration	increase	significantly	+
1	increase	significantly	level	+
1	increase	blood	level	+
1	increase	level	sirolimus	+
1	blood	level	sirolimus	+
⋮	⋮	⋮	⋮	⋮

input : A group of *2-tuples* $T_2 =< t_{21}, t_{22}, \ldots, t_{2a} >$
output: $T_3, \ldots, T_n$; $n \geq 3$

for $i \leftarrow 3$ **to** n **do**
  $T = \varnothing$;
  $c = 0$;
  foreach $t_{(i-1)b}$ *in* T_{i-1} **do**
    result = `Expansion`$(t_{(i-1)b})$;
    while $\exists\, t \in$ result *and* $t \notin T$ **do**
      $t_{ic} = t$;
      `Add` t_{ic} **to** T;
      c++;
    end
  end
  $T_i = T$;
end

Algorithm 1: Level-wise n-tuple concatenation algorithm.

frequency for the same cells in the trigram contingency table where all the words co-occur. For Dice and Jaccard in 10 and 11, we apply the same idea from the set theory; For 12, a higher order odds-ratio should be used.

5. TEXT CATEGORIZATION USING SVM

Finally, as described in Fig. 1, an SVM classifier is used to classify search results from the expanded queries, thus speeding up drug interaction sentence identification. Joachims [18] highlights the many reasons why SVMs are promising algorithms for text classification: 1) the high dimensionality of representation spaces can result in overfitting for some other classifiers; 2) in text categorization, few features are irrelevant; 3) the feature vectors contain too many zero entries; and 4) most text categorization problems are linearly separable. In our experiments, we use SVMlight[7] with a linear kernel (which have been shown to outperform other kernels [46]).

Most text categorization implementations work with pre-processed input, such as mapping to a bag of words (BOW) representation and weighting the document (sentence) vectors suitably. We opted to not use stemming, since terms could play very different roles in identifying drug interactions even though they might have the same stem [8]. For example, in the sentence

[7]http://svmlight.joachims.org/, version 6.02

[8]Stemming is much more aggressive than lemmatization, e.g., *receiving* is changed to *receive* after lemmatization, but to *receiv* after being stemmed. Stemming is usually used when *receiving* and *received* need to be treated as a single form.

Table 7: SVM performance with different combinations of weighting schemes and kernels (F:F-measure, P:Precision, R:Recall)

Binary	Linear			Polynomial		
	F	P	R	F	P	R
1	89.99	95.07	85.44	76.62	95.29	64.08
2	93.33	94.78	91.93	84.14	95.57	75.16
3	92.38	94.68	90.19	83.56	94.6	74.84
4	92.02	94.8	89.4	83.24	96.65	73.1
5	90.42	93.46	87.58	81.88	96.58	71.07
Avg	91.63	94.55	88.90	81.89	95.73	71.65
LOOCV	94.09	95.54	92.7	85.68	96.63	76.96

TF	Linear			Polynomial		
	F	P	R	F	P	R
1	90.13	95.57	85.28	58.72	93.13	42.88
2	93.06	94.9	91.3	64.37	94.21	48.89
3	92.42	95.14	89.87	66.86	95.04	51.58
4	91.51	94.44	88.77	58.39	95.34	42.09
5	90.11	94.64	86.01	57.49	95.96	41.04
Avg	91.45	94.93	88.24	61.17	94.73	45.29
LOOCV	93.55	95.43	91.75	66.53	95.18	51.14

TF-IDF	Linear			Polynomial (-j 1.55)		
	F	P	R	F	P	R
1	90.07	93.53	86.87	85.28	85.76	84.81
2	93.86	95.72	92.09	70.44	54.56	99.37
3	93.36	95.53	91.3	73.07	58.06	98.58
4	91.82	94.03	89.72	78.37	94.22	67.09
5	91.08	93.98	88.36	78.42	90.53	69.18
Avg	92.04	94.55	89.66	77.12	76.62	83.80
LOOCV	95.08	96.36	93.84	90.21	95.74	85.3

TF-RF	Linear			Polynomial		
	F	P	R	F	P	R
1	89.2	96.33	83.07	69.76	96.64	54.59
2	92.88	96.11	89.87	75.70	97.51	61.87
3	91.5	95.52	87.82	71.46	96.76	56.65
4	91.08	95.17	87.34	70.75	96.97	55.7
5	89.88	95.09	85.22	65.55	96.92	49.53
Avg	90.91	95.64	86.66	70.65	96.96	55.66
LOOCV	93.48	96.34	90.8	74.31	97.19	60.15

> "the use of zolmitriptan in patients receiving MAO-A inhibitors is contraindicated,"

the combination of *use* and *receiving* (as a present participle) strongly hints at being treated with two different drugs simultaneously, but the stem *receiv* could be aquired by stemming a normal verb *receive*, which is rarely used when expressing two drug treatments at the same time. From the training datasets, only 31 sentences containing *receive* in the positive dataset are classified positive (0.7%), whereas 131 sentences have *receiving* as a present participle describing *patient(s)* (3.3%). Conversely, in the negative dataset, 224 instances contain *receive* (4.4%), whereas 21 sentences have *receiving* describing *patient(s)* as a present participle (0.4%). This demonstrates that authors prefer to use the present participle form *receiving* to express drug interaction. For term weighting, we investigate four different methods: binary, term-frequency (tf), (tf-idf), and (tf-rf). All these weighting schemes are similar to the concepts from traditional information retrieval domain. In the last tf-rf scheme, rf has the same meaning of relative frequency as in Section 4.1.

6. EXPERIMENTAL RESULTS

6.1 System Design and Setup

6.1.1 Data preparation

The DrugBank database contains 4772 drug entries (a.k.a. drug cards) corresponding to more than 12,000 different trade names and synonyms. Among all these drugs, more than 1350 are FDA approved small molecule and biotech drugs, and 3243 are experimental drugs. However, only 1036 of these 4772 entries contain drug interaction descriptions (short paragraphs) and we focus on these drugs.

Table 10: Some of bi-gram cues from extended collocation mining

bi-gram collocations		loglh	avgmi	dice	t-test
concomitant (*amod*)	administration	2911.79	2100.41	.0088	14.27
	treatment	1421.70	1025.54	.0006	3.44
	use	1354.52	977.08	.0042	9.83
	medication	930.61	671.30	.0006	3.69
	therapy	861.60	621.51	.0011	5.06
increase	coadministration (*subj-x*)	2121.22	1530.14	.0009	4.20
recommend		1172.57	845.83	.0004	3.21
decrease		1042.94	752.32	.0007	3.92
result		1002.32	723.02	.0020	6.70
interfere		955.16	689.00	.70	1.30
alter		818.49	590.41	.0002	2.37
administer	when (*advmod*)	2603.77	1878.22	.0074	13.06
coadministered		1822.00	1314.30	.0042	9.88
co-administered		1257.53	907.11	.0018	6.50
take		1102.67	795.41	.0007	3.89
give		1081.87	780.40	.0030	8.33
use		990.09	714.20	.0025	7.22
initiate		746.48	538.48	.0008	4.19
add		687.92	496.23	.0003	2.58

Table 11: Some of tri-gram cues from extended collocation mining. Tri-grams which are just simple extension of bi-gram cues have been removed.

tri-gram collocations			loglh	avgmi	dice	t-test
increase	plasma	concentration	5512.31	3976.29	.0045	9.32
when	administer	increase	5295.64	3819.99	.0004	2.80
decrease	plasma	concentration	4553.77	3284.85	.0014	5.28
when	administer	inhibitor	4496.14	3243.28	.0003	2.43
drug	increase	concentration	4319.32	3115.73	.0003	2.59

The 1036 drug entries in DrugBank include valid webpage links in the column *interaction_insert*. We retrieved all of these webpages about detailed drug interactions, extracted plain text using the HTML parser tool [9], and segmented them into sentences using the LingPipe[10] toolkit. The total number of the sentences extracted in this manner were 9407. To produce a high-quality training dataset, we created a simple drug name dictionary using DrugBank's *Generic Name*, *Brand Name*, and *Synonyms* fields, and removed all sentences that do not have any entry from this dictionary. This reduced the number of valid positive sentences to 3900. For instance, from the description paragraph for the drug *Nicotine*:

> "Physiological changes resulting from smoking cessation, with or without nicotine replacement, may alter the pharmacokinetics of certain concomitant medications, such as tricyclic antidepressants and theophylline. Doses of these and perhaps other medications may need to be adjusted in patients who successfully quit smoking."

we see that the first sentence is a valid drug interaction sentence but the second one merely gives a warning about required dose adjustment due to the interaction.

6.1.2 Generating negative examples

For the purpose of evaluation we generate negative examples of drug interaction sentences using both the LPU method and manual annotation. We first obtained 5141 candidate negative sentences

[9]http://htmlparser.sourceforge.net/

[10]http://alias-i.com/lingpipe/

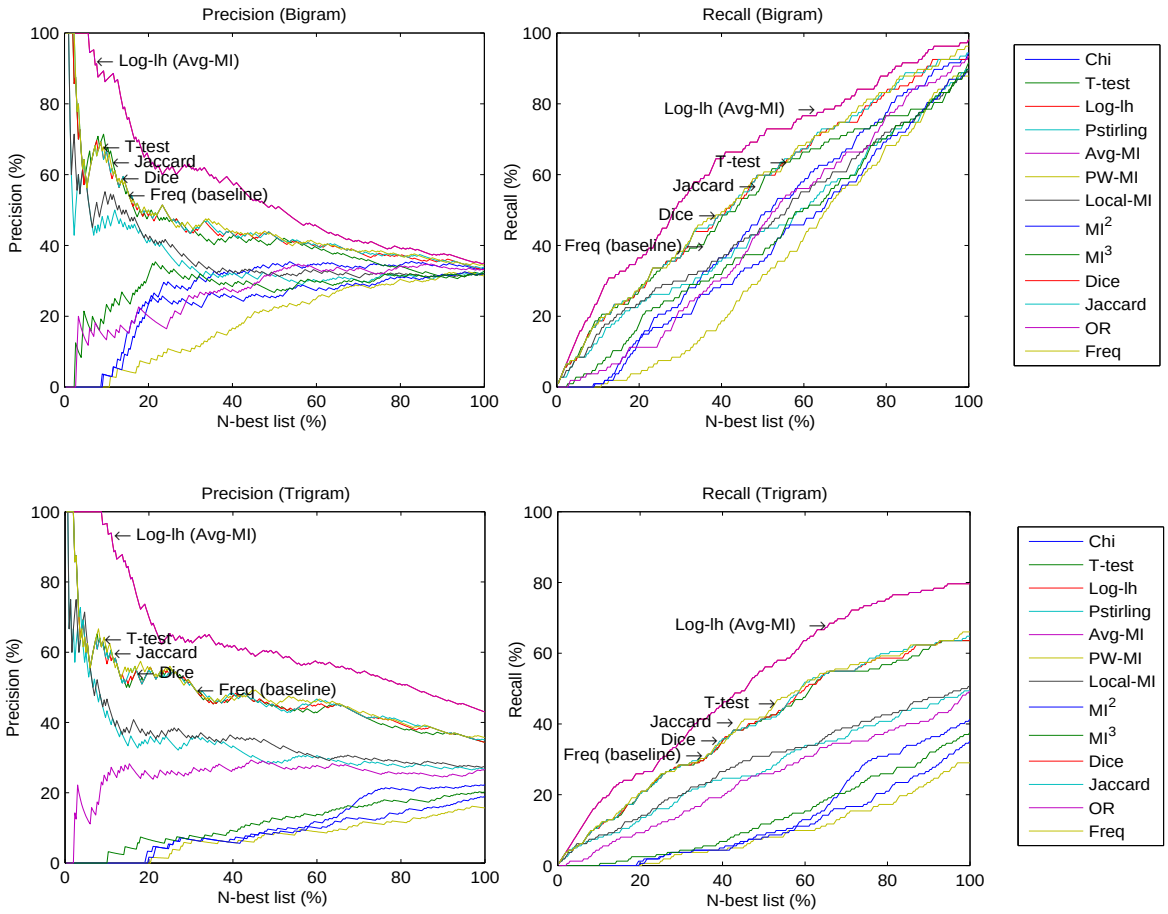


Figure 3: N-best evaluation (top 300): comparison of different methods on bigram and trigram by using precision and recall. Log-likelihood and average mutual information perform the best among all measures studied here.

Table 8: Top-20 single-term cues. "tf-pos" and "tf-neg" are the term frequency in the positive and negative datasets, respectively.

Rank	MI			F-score			RF		
	term	tf-pos	tf-neg	term	tf-pos	tf-neg	term	tf-pos	tf-neg
1	concomitant	398	16	concomitant	398	16	coadministration	236	4
2	when	512	131	when	512	131	closely	44	0
3	coadministration	236	4	other	442	138	coadministered	119	3
4	interaction	285	27	interaction	285	27	contraindicated	39	1
5	interactions	247	17	coadministration	236	4	drug-drug	38	1
6	administration	459	210	interactions	247	17	caution	150	4
7	concomitantly	178	6	administration	459	210	exercised	37	0
8	concentrations	329	113	administered	341	123	concurrently	90	3
9	administered	341	123	concentrations	329	113	concomitantly	178	6
10	increase	320	117	concomitantly	178	6	steady	28	0
11	caution	150	4	increase	320	117	co-administered	83	3
12	clearance	173	23	caution	150	4	concomitant	398	16
13	metabolized	136	6	clearance	173	23	adjustments	23	1
14	co-administration	142	9	agents	217	59	metabolized	136	6
15	agents	217	59	co-administration	142	9	careful	20	0
16	warfarin	128	7	metabolized	136	6	warfarin	128	7
17	coadministered	119	3	warfarin	128	7	steady-state	91	5
18	recommended	154	28	coadministered	119	3	depressants	18	0
19	metabolism	164	41	recommended	154	28	digitalis	18	0
20	concurrently	90	3	metabolism	164	41	monitored	100	6

Table 9: Some of our identification results (Agreement: drug interaction consistent with the one in DrugBank; Disagreement: results which disagree with the one in Drugbank; Supplement: new drug interaction for the same drug in DrugBank; New: new interaction while DrugBank has an empty entry; In vitro model: a special type of drug interaction.)

Drug pair	Category	Drug interactions in DrugBank	New drug interactions from MEDLINE
Abciximab, ticlopidine	Agreement	These medications have included heparin warfarin, beta-adrenergic receptor blockers, calcium channel antagonists, angiotensin converting enzyme inhibitors, intravenous and oral nitrates, ticlopidine, and aspirin. [DB00054]	concomitant abciximab plus ticlopidine treatment yields a platelet inhibition profile that is a composite of the effects of the 2 agents. [PubMed:10966553]
Alteplase, angiotensin-converting-enzyme inhibitor	Supplement	The interaction of Activase with other cardioactive or cerebroactive drugs has not been studied. In addition to bleeding associated with heparin and vitamin K antagonists, drugs that alter platelet function (such as acetylsalicylic acid, dipyridamole and Abciximab) may increase the risk of bleeding if administered prior to, during, or after Activase therapy. [DB00009]	they warn that patients who are taking an angiotensin-converting-enzyme inhibitor may be at increased risk for angioedema with concomitant alteplase therapy. [PubMed:10813008]
Goserelin, testosterone propionate	In-vitro model	No formal drug-drug interaction studies have been performed. No confirmed interactions have been reported between ZOLADEX and other drugs. [DB00014]	concomitant administration of goserelin or org 30276 and testosterone propionate to castrated rats resulted in a further decrease of the pituitary 5 alpha-reductase activity, compared to the castrated, gnrh-analogue treated rats. [PubMed: 2141376]
Rituximab, fludarabine	In-vitro model	There have been no formal drug interaction studies performed with RITUXAN. However, renal toxicity was seen with this drug in combination with cisplatin in clinical trials. [DB00073]	the study of the effect of fludarabine and rituximab in six freshly isolated b-cell chronic lymphocytic leukaemia (b-cll) samples showed that, in most cases, fludarabine has an additive cytotoxic activity with rituximab and complement. [PubMed: 11564066]
Etanerept, methotrexate	Disagreement	Specific drug interaction studies have not been conducted with ENBREL. However, it was observed that the pharmacokinetics of ENBREL was unaltered by concomitant methotrexate in rheumatoid arthritis patients. [DB00005]	inhibition of articular destruction was also proven by administration of the biologic agents etanercept and infliximab plus methotrexate. [PubMed: 12665969]
Lutropin alfa, follitropin alfa	New	Drug Interactions Not Available. [DB00044]	conclusion: subcutaneous co-administration of 75 iu lutropin alfa with follitropin alfa is safe and effective in inducing follicular development in women with profound gonadotrophin deficiency. [PubMed: 18485121]
Leuprolide, ethanol	New	Drug Interactions Not Available. [DB00007]	further, twice daily administration of leuprolide (50 microg/kg, s.c), concomitant with ethanol, prevented the gradual increase in marble-burying behavior after ethanol-withdrawal and this effect was comparable to fluoxetine (5 mg/kg, i.p.). [PubMed: 18448097]

from the search results by using only drug names. Later, using LPU, 45 sentences were treated as positive and removed automatically; in the manual labeling approach, about 132 positive sentences were removed (2.56% of 5141). Finally we harvested 5096 negative sentences through the LPU method, and 5009 manually labeled negative sentences in total.

6.1.3 Cue mining

As discussed earlier, for identifying single term cues, we utilized three measures as described in Section 4.1: mutual information, fisher kernel and relevance frequency. Table 8 shows that all these measures mined single term cues admirably. Since these ranking lists share many of the same terms, especially in the top portions of the lists, we use the union of the top 50 terms from each list as our single-term cues (consisting of 53 cue words after removing stop words and strong domain-related terms). 92.5% of all these single-term cue words are recognized as strong evidence words for identifying drug interaction during a manual check.

For mining multi-word cues, we explored both basic bigrams and higher order *n*-grams (n$\geq$3). We first parsed all sentences in both positive and negative datasets, and curated a total of 113577 grammatical dependencies. By filtering out 31 trivial relations (e.g., "det", "auxpass", "cop", etc.; most of them either are related with stop-words like "the", or create duplicate connections), and removing all the bigrams which appear in positive dataset for less than 10 times, we retained 83417 of them to construct 2-tuple co-occurrence information. Next, with the concatenation idea from Section 4.3, we retrieved 167302 valid records in the 3-tuple co-occurrence database. However, by requiring that valid 3-gram cues must appear in positive dataset for more than 5 times, the number of valid 3-tuples in the candidate co-occurrence database is dramatically reduced to 644. Finally, the 12 association measures from Table 5, together with the baseline measure of co-occurrence frequency in the positive dataset, were calculated for both the 2-tuple and 3-tuple co-occurrence databases. Some of the ranked bigram and trigram results are listed in Table 10 and 11, respectively. Further experiments revealed that mining higher order *n*-gram (n$\geq$4) cues are not necessary for drug interaction search, since all of them are just simple extension of tri-gram cues.

6.1.4 SVM classification

We compared the classification capability of eight SVM models with different combinations of two kernels (linear and polynomial) and four weighting schemes (binary, tf, tf-idf, tf-rf). We evaluated these models by using *F-measure*, given by $F = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$. We also conducted a *5-fold cross validation* and a *leave one out cross validation* (LOOCV). Table 7 summarizes the detailed model evaluation results. We observe that the linear kernels with binary weighting and tf-idf weighting provide the best performance, and use the former for convenience of implementation.

6.2 Evaluation

We perform three categories of evaluation. We evaluated the effect of our new collocation mining approach, and also we tested the

quality of the training data labeled by LPU method. We also listed some representative findings from MEDLINE.

6.2.1 N-best evaluation

To evaluate our multi-word collocation mining approaches, we employed the *n*-best evaluation method, which is widely used for evaluating collocation measures [9, 10, 29, 34, 37]. We first scanned all the *n*-tuples satisfying the threshold requirement (frequency in positive dataset greater than 5), and generated a human-labeled set of real cues as C_{real}. For each association measure, we considered the *n*-tuples from the ranking lists as *valid* only if they also appear in set C_{real}. Obviously, good association measures will always put the real valid cues near the top of their ranking lists. Therefore, after scanning the top-n results in the ranking lists, we draw *precision-recall* graphs for these measures, by calculating the proportion of valid cues in the *n*-best list (*precision*) and the fraction of the number of valid cues from the *n*-best list to the size of C_{real}(*recall*). Fig. 3 visualizes the precision-recall graphs using the top 300 lists from each measure. The *x*-axis represents the proportion of the ranking list, while the *y*-axis depicts the corresponding precision (recall) values. The *n*-best evaluation shows that log-likelihood (Log-lh) and average mutual information (Avg-MI) perform significantly better than other measures from the precision-recall graphs (they are the most top curves), a result consistent with prior research [8, 29]. Measures such as the Dice coefficient, Jaccard and t-test have performances comparable to the baseline approach of positive frequency (Freq), while the other measures perform poorly. For the rest of this paper, we choose the log-likelihood method as our association measure.

6.2.2 Evaluating the quality of LPU labeling

Our experiments also show that the negative dataset generated automatically by the LPU is comparable to the one labeled manually, and that the difference between these two methods for constructing negative datasets has little influence on the final results of cue word mining. We compared the *average precision* values and also the *normalized discounted cumulative gain* (nDCG) values [6, 16] of the multi-word cue mining results from these two different negative datasets. Average precision is calculated by averaging the precision values from the rank positions where a valid cue is retrieved, and the nDCG value for the top-p list is calculated as $nDCG_p = \frac{DCG_p}{IDCG}$. Here the DCG is defined as:

$$DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2 i}$$

where rel_i is 1 when the *i*-th *n*-gram in the list is judged as a valid cue, and 0 otherwise. IDCG means the possible maximum DCG value when all the valid cues are ranked at the top [16].

Table 12 shows that there is no significant decrease of average precision or nDCG values when we use the LPU method to generate the negative dataset. In fact, the final ranking lists (top 100) based on these two different negative datasets share many cue words. For the bigram approach, 96 of the top 100 cues are identical, and for the trigram approach, 97 of the top 100 are identical.

6.2.3 Representative results

Table 9 depicts some of the results of our analysis, which classifies all inferred drug interaction sentences into five categories. Most of the results are consistent with the original drug interaction descriptions in DrugBank (same drug, same interaction). Some describe additional interactions (i.e., same drug, but interactions with new drugs). We also found some interaction description sentences which disagree with the ones in DrugBank. Among the 4772 entries in DrugBank, most of them (3736) have empty drug interaction records, and we found new interaction information for some of these drugs. Finally, our approach also identifies drug interactions from in-vitro studies, a category often ignored in clinical studies.

Table 12: Collocation mining effects of log-likelihood method by using different negative datasets (top 100 ranking lists)

Method	Bigram		Trigram	
	AvgP (%)	nDCG	AvgP (%)	nDCG
Manually labeling	83.84	0.967	85.50	0.971
LPU	80.34	0.958	84.07	0.968

7. CONCLUSIONS AND FUTURE WORK

We have introduced a framework for mining data-driven linguistical cues and for using these cues to aid querying for drug interactions. Our famework integrates traditional collocation mining with discriminative association with positive/negative training instances/ Compared to other systems which require *a priori* domain knowledge, our system seeks to solve this problem with the help of linguistic features alone. By identifying many drug interactions not currently curated in DrugBank, our experimental results demonstrate that this is a promising approach.

Since our framework uses only linguistic cue words to augment queries, this approach can be re-targeted toward many other domain-specific needs, such as for modeling biochemical interactions and subjective opinions. This is one direction of future work. A second issue we are investigating is to develop more structured, graph-theoretic, representations of linguistic cues from dependency parses, i.e., beyond simple sets of words as used in collocations.

8. ACKNOWLEDGMENTS

This work is supported in part by a grant from the Institute for Critical Technology and Applied Science (ICTAS), Virginia Tech. We thank all the three reviewers for their valuable comments and suggestions.

9. REFERENCES

[1] J. Bai, D. Song, P. Bruza, J.-Y. Nie, and G. Cao. Query expansion using term relationships in language models for information retrieval. In *CIKM '05*, pages 688–695. ACM Press, 2005.

[2] C. M. Bell, W. V. Hatch, H. D. Fischer, G. Cernat, J. M. Paterson, A. Gruneir, S. S. Gill, S. E. Bronskill, G. M. Anderson, and P. A. Rochon. Association between tamsulosin and serious ophthalmic adverse events in older men following cataract surgery. *JAMA : the journal of the American Medical Association*, 301(19):1991–1996, 2009.

[3] J. Bhogal, A. Macfarlane, and P. Smith. A review of ontology based query expansion. *IP&M*, 43(4):866–886, 2007.

[4] B. Billerbeck, F. Scholer, H. E. Williams, and J. Zobel. Query expansion using associated queries. In *CIKM '03*, pages 2–9. ACM Press, 2003.

[5] Y.-W. Chang and C.-J. Lin. Feature ranking using linear svm. *JMLR*, pages 53–64, 2008.

[6] B. Croft, D. Metzler, and T. Strohman. *Search Engines: Information Retrieval in Practice*. Addison Wesley, 1 edition, 2009.

[7] H. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma. Query expansion by mining user logs. *IEEE TKDE*, 15:829–839, 2003.

[8] T. Dunning. Accurate methods for the statistics of surprise and coincidence. *Comput. Linguist.*, 19(1):61–74, 1993.

[9] S. Evert. *Corpora and collocations*, volume A, chapter 58. Mouton de Gruyter, Berlin, 2008.

[10] S. Evert and B. Krenn. Methods for the qualitative evaluation of lexical association measures. In *ACL '01*, pages 188–195, 2001.

[11] J. R. Firth. A synopsis of linguistic theory, 1930-1955. *Studies in Linguistic Analysis*, pages 1–32, 1957.

[12] J. Fitzgerald, B. Schoeberl, U. Nielsen, and P. Sorger. Systems Biology and Combination Therapy in the Quest for Clinical Efficacy. *Nat. Chem. Biol.*, Vol. 2(9):458–466, 2006.

[13] G. W. Flake, E. J. Glover, S. Lawrence, and L. C. Giles. Extracting query modifications from nonlinear svms. In *WWW '02*, pages 317–324. ACM Press, 2002.

[14] B. M. Fonseca, P. Golgher, B. Pôssas, B. Ribeiro-Neto, and N. Ziviani. Concept-based interactive query expansion. In *CIKM '05*, pages 696–703. ACM Press, 2005.

[15] W. R. Hersh, R. T. Bhupatiraju, and S. Price. Phrases, boosting, and query expansion using external knowledge resources for genomic information retrieval. In *TREC*, pages 503–509, 2003.

[16] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM TOIS*, 20(4):422–446, 2002.

[17] Y. Jing and W. B. Croft. An association thesaurus for information retrieval. In *RIAO '94*, pages 146–160, 1994.

[18] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In *ECML '98*, pages 137–142. Springer Verlag, Heidelberg, DE, 1998.

[19] M.-C. Kim and K.-S. Choi. A comparison of collocation-based similarity measures in query expansion. *IP&M*, 35(1):19–30, 1999.

[20] J. M. Kovarik, D. Beyer, and R. L. Schmouder. Everolimus drug interactions: application of a classification system for clinical decision making. *Biopharmaceutics & Drug Disposition*, 27(9):421–426, 2006.

[21] M. Lan, C. Tan, and H. Low. Proposing a new term weighting scheme for text categorization. In *AAAI '06*, pages 763–768, 2006.

[22] B. Liu, Y. Dai, X. Li, W. Lee, and P. Yu. Building text classifiers using positive and unlabeled examples. In *ICDM '03*, pages 179–186, 2003.

[23] S. Liu, F. Liu, C. Yu, and W. Meng. An effective approach to document retrieval via utilizing wordnet and recognizing phrases. In *SIGIR '04*, pages 266–272. ACM Press, 2004.

[24] Z. Liu and W. W. Chu. Knowledge-based query expansion to support scenario-specific retrieval of medical free text. In *SAC '05*, pages 1076–1083. ACM Press, 2005.

[25] R. Mandala, T. Tokunaga, and H. Tanaka. Combining multiple evidence from different types of thesaurus for query expansion. In *SIGIR '99*, pages 191–197, 1999.

[26] C. D. Manning and H. Schtze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999.

[27] M. Marneffe, B. Maccartney, and C. Manning. Generating typed dependency parses from phrase structure parses. In *LREC '06*, pages 449–454, 2006.

[28] Masataka.T. Network analysis of adverse drug interactions. In *IBSB '08*, 2008.

[29] B. T. Mcinnes. Extending the log likelihood measure to improve collocation identification. Master's thesis, University of Minnesota, 2004.

[30] M. Mitra, A. Singhal, and C. Buckley. Improving automatic query expansion. In *SIGIR '98*, pages 206–214. ACM Press, 1998.

[31] M. Okabe and S. Yamada. Semisupervised query expansion with minimal feedback. *IEEE TKDE*, 19(11):1585–1589, 2007.

[32] S. Oyama, T. Kokubo, and T. Ishida. Domain-specific web search with keyword spices. *IEEE TKDE*, 16(1):17–27, 2004.

[33] H. J. Peat and P. Willett. The limitations of term co-occurrence data for query expansion in document retrieval systems. *Journal of the American Society for Information Science*, 42(5):378–383, 1991.

[34] P. Pecina and P. Schlesinger. Combining association measures for collocation extraction. In *COLING/ACL '06*, pages 651–658. Association for Computational Linguistics, 2006.

[35] S. C. Piscitelli and K. D. Gallicano. Interactions among drugs for hiv and opportunistic infections. *The New England journal of medicine*, 344(13):984–996, 2001.

[36] Y. Qiu and H.-P. Frei. Concept based query expansion. In *SIGIR '93*, pages 160–169. ACM Press, 1993.

[37] D. P. School, D. Pearce, and B. Qh. A comparative evaluation of collocation extraction techniques. In *LREC '02*, 2002.

[38] C. D. Scripture and W. D. Figg. Drug interactions in cancer therapy. *Nature Reviews Cancer*, 6(7):546–558, 2006.

[39] V. Seretan, L. Nerima, and E. Wehrli. Multi-word collocation extraction by syntactic composition of collocation bigrams. In *RANLP '03*, Current Issues in Linguistic Theory, pages 91–100. John Benjamins, 2004.

[40] F. Smadja. Retrieving collocations from text: Xtract. *Comput. Linguist.*, 19:143–177, 1993.

[41] O. Vechtomova, S. Robertson, and S. Jones. Query expansion with long-span collocates. *Information Retrieval*, 6(2):251–273, 2003.

[42] X. Wang and C. Zhai. Mining term association patterns from search logs for effective query reformulation. In *CIKM '08*, pages 479–488. ACM Press, 2008.

[43] J. Wermter and U. Hahn. Collocation extraction based on modifiability statistics. In *COLING '04*. Association for Computational Linguistics, 2004.

[44] J. Xu and B. W. Croft. Query expansion using local and global document analysis. In *SIGIR '96*, pages 4–11, 1996.

[45] J. Xu and B. W. Croft. Improving the effectiveness of information retrieval with local context analysis. *ACM TOIS*, 18:79–112, 2000.

[46] Y. Yang and X. Liu. A re-examination of text categorization methods. In *SIGIR '99*, pages 42–49. ACM Press, 1999.