UIC at TREC-2004: Robust Track

Shuang Liu, Chaojing Sun, Clement Yu

Database and Information System Lab
Department of Computer Science, University of Illinois at Chicago
{sliu, csun, yu}@cs.uic.edu

Abstract

In TREC 2004, the Database and Information System Lab (DBIS) at University of Illinois at Chicago (UIC) participates in the robust track, which is a traditional ad hoc retrieval task. The emphasis is based on average effectiveness as well as individual topic effectiveness.

In our system, noun phrases in the query are identified and classified into 4 types: proper names, dictionary phrases, simple phrases and complex phrases. A document has a phrase if all content words in a phrase are within a window of a certain size. The window sizes for different types of phrases are different. We consider phrases to be more important than individual terms. As a consequence, documents in response to a query are ranked with matching phrases given a higher priority. WordNet is used to disambiguate word senses. Whenever the sense of a query term is determined, its synonyms, hyponyms, words from its definition and its compound concepts are considered for possible additions to the query. The newly added terms are used to form phrases during retrieval. Pseudo feedback and web-assisted feedback are used to help retrieval. We submit one title run this year.

1. Introduction

Our recent work [LL04] shown that phrase and word sense disambiguation can help improve retrieval effectiveness in text retrieval. Component content words in a phrase can be used to disambiguate other component words in the same phrase. This allows selective synonyms, hyponyms, words from the definition, and compound concepts to be added for improving retrieval results.

To recognize different type of phrases, the strengths of several existing software and dictionary tools, namely Minipar [Lin94, Lin98], Brill's tagger [Brill], Collins' parser [Coll97, Coll99], WordNet [Fell98] and web feedback are fully utilized. Furthermore, additional techniques are introduced to improve the accuracy of recognition.

The recognition of each type of phrases in documents is dependent on having windows of different sizes. Specifically, a proper noun would have the component words in adjacent locations; a dictionary phrase in a document may have its component content words separated by a distance of no more than w1 words; a simple phrase in a document may have its component content words separated by a distance of no more than w2 words, with $w2 > w1$; similarly, a window containing the component content words of a complex phrase should have a size no larger than w3, with $w3 > w2$.

We consider phrases to be more important than individual content words when retrieving documents [LY03, LL04]. Consequently, the similarity measure between a query and a document has two components (phrase-sim, term-sim), where phrase-sim is the similarity obtained by matching the phrases of the query against those in the document and

term-sim is the usual similarity between the query and the document based on term matches. The latter similarity can be computed by the standard Okapi similarity function [RW00]. Documents are ranked in descending order of (phrase-sim, term-sim). That is, documents with higher phrase-sim will be ranked higher. When documents have the same phrase-sim, they will be ranked according to term-sim.

WordNet is used for word sense disambiguation. For words in one phrase or adjacent query words, the following information from WordNet is utilized: synonym sets, hyponym sets, and their definitions. When the sense of a query word is determined, its synonyms, words or phrases from its definition, its hyponyms and its compound concepts are considered for possible addition to the query. If a synonym, hyponym, word/phrase in the definition of a synonym set, or compound concept is brought in by a query term and this term forms a phrase with some other query terms, new phrases can be generated.

Feedback terms are brought in by pseudo-feedback [LY03] and web-assisted feedback [GW03, YC03]. Additional weights are assigned to the top ranked feedback terms, if they can be related to the original query terms through different WordNet relations.

In the remaining part of this paper, how phrases in a query are recognized and how they are classified into different types are discussed in section 2. Section 3 presents how WordNet can be utilized to disambiguate word sense and bring in new terms. Section 4 describes pseudo feedback and web-assisted feedback, and how we assign weights to terms brought in by feedbacks. In section 5 we analyze the run we submitted this year. Section 6 concludes this paper.

2. Phrase Recognition

Noun phrases in a query are classified into proper names, dictionary phrases, simple phrases and complex phrases. Existing tools including Brill's tagger, Minipar, Collins' parser, WordNet and web retrieval feedback are used to recognized different type of phrases. They are utilized in such a way that their strengths are fully made use of.

Proper nouns include names of people, organizations, places etc. Dictionary phrases are noun phrases which can be found in a dictionary such as WordNet. A simple phrase contains exactly two content words. A complex phrase contains three or four content words. Phrases involving more than four content words are unlikely to be useful in document retrieval, as very few, if any, documents contain all these content words. The four types of phrases are ordered from low to high, for example proper noun is the lowest type. If a phrase belongs to multiple types, it will be classified to the lowest type of the four types indicated above. For example, if a simple phrase is a dictionary phrase, then it is classified as a dictionary phrase.

In the phrase recognition system, Brill's tagger is used to assign a Part-of-Speech (POS) to each query word. The name entity finder Minipar is used for the purpose of recognition of proper nouns. The Collins' parser is used to obtain a parse tree for a query and the base noun phrases discovered by Collins' parser are used in further detailed analysis. We use WordNet to recognize dictionary phrases and some proper nouns; additionally, web retrieval feedback provides more contexts for short queries.

To recognize phrases in a query, the query is first fed into Minipar for proper name recognition. Next, each word in a query is assigned a POS by using Brill's tagger, After the query is parsed by Collins' parser [Coll97, Coll99], certain noun phrases are recognized as base noun phrases that is they cannot be decomposed by Collins' parser. A base noun phrase may be decomposed into smaller phrases using Minipar or by a statistical analysis process. Each noun phrase which is recognized by Collins' parse or by the statistical analysis process is passed to WordNet for further proper noun or dictionary phrase recognition. A noun phrase which is not a proper noun or a dictionary phrase is classified to be a simple phrase or a complex phrase based on the number of content words it contains. Phrases which are coordinate noun phrases (involving "and" or "or") or have embedded coordinate noun phrases are processed to yield implicit phrases. For example, in the phrase "physical or mental impairment", "physical impairment" is an implicit phrase. These newly generated phrases also undergo a similar procedure to determine its type.

3. Word Sense Disambiguation

Word sense disambiguation makes use of content words in the same phrase or adjacent words in the query. When a given query q is parsed, the POS of each word as well as the phrases in q are recognized. Suppose two adjacent terms t_1 and t_2 in q form a phrase p. From WordNet, the following information can be obtained. Each of t_1 and t_2 has (I) one or more synsets; (II) a definition for each synset in (I); (III) one or more hyponym synsets of each synset in (I) (containing IS-A relationships; for example, the synset {*male child, boy*} *is a hypo*nym synset of {*male, male person*}); (IV) definitions of the synsets of the hyponyms in (III). Suppose S_i is a synset of t_i; SD_i is the definition of S_i; H_i is a hyponyms synset of S_i; HD_i is the definition of H_i.

These four items (I), (II), (III) and (IV) can be used in 16 disambiguation rules. During disambiguation, conflicts may arise if different rules determine different senses for a query word. Experimental data provide some relative degrees of accuracy or weights of the individual rules. If multiple rules determine different senses for a word, then the sense of the word is given by the set of rules with the highest sum of weights, which determines the same sense for the word. Terms are added to the query after word sense disambiguation.

3.1. Disambiguation Rules

Rule 1. If S_1 and S_2 --- synsets of t_1 and t_2 --- have synonyms in common and S_1 and S_2 have the same POS, then S_1 is determined to be the sense of t_1 and S_2 is determined to be the sense of t_2.

Rule 2. If t, a synonym of t_1 (other than t_1) in synset S_1, is found in SD_2, and t has the same POS in both S_1 and SD_2, S_1 is determined to be the sense of t_1.

Example 3.2: *A query is "pheromone scents work". A synset of the verb "work" is {influence, act upon, work}. "influence" is in the definition of "pheromone", which is "a chemical substance secreted externally by some animals (especially insects) that influences the physiology or behavior of other animals of the same species". The word "influence" is a verb in both the synset {influence, act upon, work} and the definition of "pheromone".*

Rule 3. If t, a synonym of t_1 (other than t_1) in synset S_1, is found in H_2 or S_1 and H_2 are the same synset and S_1 and H_2 have the same POS, then S_1 is determined to be the sense of t_1.

Example 3.3: *A query is "Greek philosophy stoicism", a synset of "stoicism" is a hyponym synset of "philosophy"; "philosophy" and "stoicism" have the same POS. The sense of "stoicism" is determined.*

Rule 4. If t, a synonym of t_1 (other than t_1) in synset S_1, is found in HD_2 and t has the same POS in both S_1 and HD_2, S_1 is determined to be the sense of t_1.

Example 3.4: *A query is "Toronto Film Awards". A synonym "motion picture" in a synset of "film" appears in the definition of "Oscar" --- a hyponym of "award". The POSs of "motion picture" in the definition and in the synset of "film" are the same. Thus, the sense of film is determined.*

Rule 5. If SD_1 contains t_2 or synonyms of t_2 and t_2 or each synonym of t_2 in SD_1 has the same POS with t_2. Then S_1 is determined to be the sense of t_1.

Example 3.5: *Suppose a query contains the phrase "incandescent light". In WordNet, the definition of a synset of "incandescent" contains the word "light". Thus, this synset of "incandescent" is used.*

Rule 6. If SD_1 and SD_2 have the maximum positive number of content words in common, and for each common word, it has the same POS in SD_1 and SD_2, then the senses of t_1 and t_2 are S_1 and S_2 respectively.

Example 3.6: *Suppose a query is "induction and deduction". Each of the two terms has a number of senses. The definitions of the two terms which have the maximum overlap of two content words, namely "general" and "reasoning" are their determined senses. For "induction" and "deduction", the definitions of the determined synsets are "**reasoning** from detailed facts to **general** principles" and "**reasoning** from the **general** to the particular (or from cause to effect)", respectively.*

Rule 7. If SD_1 contains the maximum positive number of content words being hyponyms of S_2, and for each of these contained words in SD_1, it has the same POS in SD_1 and hyponyms synsets of S_2, then S_1 is determined to be the sense of t_1.

Example 3.7: *This is a continuation of Example 3.3. The definition of synset 2 of "stoicism" is "the philosophical system of the Stoics following the teachings of the ancient Greek philosopher ZenoContain" which contains "teaching" --- a hyponym of "philosophy", so the sense of "stoicism" can also be determined by this rule.*

Rule 8. If SD_1 has the maximum positive number of content words with the hyponyms synsets' definitions of S_2, and for each common word between SD_1 and some hyponyms synset definition of t_2, it has the same POS in both definitions, then S_1 is determined to be the sense of t_1.

Example 3.8: *A query is "Iraq Turkey water". The definition of a synset of Turkey has the common word "Asia" with the definitions of "Black Sea" and "Euxine Sea", where "Black Sea" and "Euxine Sea" are hyponyms of "water", so the sense of "Turkey" can be determined.*

Rule 9. If S_1 has the maximum positive number of hyponyms synsets which have common synonyms with S_2 or H_1 and S_2 are the same synset and S_1 and S_2 have the same POS, the sense of t_1 is determined to be S_1.

Example 3.9: *A query is "Tobacco cigarette lawsuit", where synset {butt, cigarette, cigaret, coffin mail, fag} is a hyponym synset of a sense of "tobacco", and they are all nouns, so the sense of "Tobacco" is determined.*

Rule 10. If S_1 has the maximum positive number of hyponyms synsets which have synonyms appearing in SD_2, and for each H_1 having synonyms appearing in SD_2, the POSs of the matching content words are the same, then the sense of t_1 is determined to be S_1.

Example 3.10: *A query is "heroic acts". "Action" is in a hyponyms synset of the second sense of "act". "Action" also appears in the definition of a sense of "heroic", which is "showing extreme courage; especially of actions courageously undertaken in desperation as a last resort". So the sense of "act"and the sense of "heroic"are determined.*

Rule 11. If S_1 has the maximum positive number of hyponyms synsets which have common synonyms with S_2's hyponym synsets, and hyponym synsets of S_1 and hyponym synsets of S_2 have the same POS, then the sense of t_1 is determined to be S_1.

Example 3.11: *This is a continuation of Example 3.9, where "Tobacco" and "cigarette" have common hyponyms, so the sense of both words can be determined.*

Rule 12. If S_1 has the maximum positive number of hyponym synsets which have synonyms appearing in the definition of S_2's hyponym synsets and for each H_1 having synonyms appearing in HD_2, the POSs of the synonyms' content words are the same as they are in H_1, then the sense of t_1 is determined to be S_1.

Example 3.12: *A query is "alcohol consumption", a hyponym "drinking" of a sense of "consumption" appears in several definitions of hyponym synsets of "alcohol", such as "brew", "brewage", so the sense of "consumption" is determined.*

Rule 13. If S_1 has the maximum positive number of hyponym synsets whose definitions contain t_2 or its synonym in some synset, say S_2, and for each HD_1 containing synonyms of t_2, every contained synonym has the same POS in both HD_1 and S_2, then the sense of t_1 is determined to be S_1.

Example 3.13: *Suppose the query is "tropical storm". A hyponym of the synset {storm, violent storm} is "hurricane" whose definition contains the word "tropical". As a result, the sense of "storm" is determined.*

Rule 14. If S_1 has the maximum positive number of hyponym synsets whose definitions have content words in common with SD_2, and for each HD_1 and SD_2 having words in common, a common word has the same POS in both HD_1 and SD_2, then the sense of t_1 is determined to be S_1.

Example 3.14: *A query is "transportation tunnel disaster". "Tunnel" is used to disambiguate the senses of "transportation". The first sense of "transportation" has a hyponym synset {mass rapic transit, rapid transit} whose definition has the common word "underground" with the first definition of "tunnel". As a result, the sense of "transportation" is determined.*

Rule 15. If S_1 has the maximum positive number of hyponym synsets whose definitions contain S_2's hyponyms, and for each HD_1 containing synonyms in H_2, a contained word has the same POS in HD_1 and H_2, the sense of t_1 is determined to be S_1.

Example 3.15: *A query is "foreclose on property"; "salvage" is a hyponym of sense 2 of "property", whose definition contains verb "save" which is a troponym of "foreclose". (troponym is the hyponym of the verb), so the sense of "property" is determined.*

Rule 16. If S_1 has the maximum positive number of hyponyms synsets whose definitions have content words in common with hyponyms synsets's definitions of S_2, and for each HD_1 and HD_2 having common words, a common word has the same POS in both HD_1 and HD_2, then the sense of t_1 is determined to be S_1.

Example 3.16: *A query is "cancer cell reproduction"; "lymphoma" is a hyponym of "cancer", "osteoclast" is a hyponym of cell,and their definitions have the common word "tissue", so the sense of cancer and cell can be determined.*

3.2. Choosing Word Sense

An ambiguous word may have multiple senses when different rules are applied. An algorithm is developed to select the best sense from the multiple choices. In this algorithm we first assign different degree of accuracies or weights to different rules based on historical data. The weight of each rule w_{Rj} is given in Table 1:

Table 1. Weights of Disambiguation Rules

Rule #	1, 3, 9, 11	2, 4, 5, 7, 10, 12, 13, 15	6	8, 14	16
Weight	0.1875	0.125	0.075	0.0625	0.005

If term t is disambiguated by different rules and S_i is a synset of t, S_i is given a total weight that is the sum of the weights of the rules which determine t to have sense S_i. It is $disam_wt(t_{Si}) = \sum_{rule\ j\ disambiguate\ t\ to\ Si} w_{Rj}$.

The sense having the maximum total weight is chosen. It is $sense(t) = \arg\max_{t_{Si}}[\ disam_wt(t_{Si})]$

3.3. Query Expansion by Using WordNet

Whenever the sense of a given term is determined to be the synset S, its synonyms, words or phrases from its definition, its hyponyms and compound words (see case (4)) of the given term are considered for possible addition to the query as shown in the following four cases, respectively. Additionally, terms t_1 and t_2 in q are adjacent and they form a phrase p.

(1) Add Synonyms.

Whenever the sense of term t_1 is determined, we examine the possibility of adding the synonyms of t_1 in its synset S to the query.

For any term t' except t_1 in S, if t' is a single term or a phrase not containing t_1, t' is added to the query if either (a) S is a dominant synset of t' or (b) t' is highly globally correlated with t_2, and the correlation value between t' and t_2 is greater than the value between t_1 and t_2. The weight of t' is given by

$$W(t') = f(t', S)/F(t') \qquad (1)$$

where *f(t', S)* is the frequency value of *t'* in *S*, and *F(t')* is the sum of frequency values of *t'* in all synsets which contain *t'* and have the same POS as *t'*. We interpret the weight of *t'* to be the likelihood that *t'* has the same meaning as *t*.

Example 4.1: *In Example 4.1, the synset containing "incandescent" also contains "candent". It can be verified that the synset is dominant for "candent" and therefore "candent" is added to the query.*

(2) Add Definition Words.

We select words from the definition of S. If t_1 is a single sense word, the first shortest noun phrase from the definition can be added to the query if it is highly globally correlated with t_1.

Example 4.2: For *query term "euro" whose definition is "the basic monetary unit of ...", the noun phrase "monetary unit" from the definition can be added to the query if it is highly globally correlated with "euro".*

(3) Add Hyponyms.

Suppose *U* is a hyponym synset of t_1. A synonym in *U* is added to the query if one of the following conditions is satisfied:

(a) *U* is the unique hyponym synset of the determined synset of t_1. For each term *t'* in *U*, *t'* is added to the query, with a weight similar to that given by the Formula (3), if *U* is dominant in the synsets of *t'*.

(b) *U* is not a unique hyponym synset of the determined synset of t_1, but the definition of *U* contains term t_2 or its synonyms. For each term *t'* in *U*, if *U* is dominant in the synsets of *t'*; *t'* is added to the query with a weight given by Formula (3).

Example 4.3: *In Example 4.3, the definition of the hyponym synset of "hurricane" contains "tropical", and "hurricane" is the only element in this synset. Thus, "hurricane" is added to the query.*

(4) Add Compound Concepts.

Given a term *t*, we can retrieve its compound concepts using WordNet. A compound concept is either a word having term *t* as a sub-string or a dictionary phrase containing term *t*.

Suppose *c* is a compound concept of a query term t_1 and *c* has a dominant synset *V*. The compound concept *c* can be added to the query if it satisfies one of the following conditions:

(a) The definition of *V* contains t_1 as well as all terms that form a phrase with t_1 in the query.

Example 4.4: *A term is "nobel", and a query is "Nobel Prize winner". Both "nobelist" and "nobel laureate" are compound words of "nobel". Their definition (they are synonyms) is "winner of a Nobel Prize", which contains all query terms in the phrase "Nobel Prize Winner".*

(b) The definition of *V* contains term t_1, and *c* relates to t_1 through a "*member of*" relation.

4. Pseudo Feedback and Web-assisted Feedback

Pseudo feedback in our system has been described in [LL04, LY03]. In this section, we discuss web-assisted feedback and how pseudo feedback and web-assisted feedback are combined together to assign weights to the newly added terms.

Last year, two groups which had the best performance in robust task used the web or massive data to improve retrieval performance. This year, beside the pseudo feedback process, we adopt web-assisted feedback to help us find useful terms. Terms from pseudo feedback and web-assisted feedback are combined together. Term's weights are added together if it appears in both pseudo feedback and web-assisted feedback. Additional weights are assigned to feedback (pseudo and web) terms, if they are related to the original query terms through WordNet.

4.1 Web-assisted Feedback

We use Google to help us do the web retrieval, the procedure is as follows:

(1) A query with the recognized phrase is submitted to Google to get the top ranked 100 documents.

(2) Get the top Google retrieved pages. If a page and its cache page doesn't contain any query term, ignore it and skip to the next page; otherwise, it is analyzed by the next step..

(3) Assign a "degree of satisfaction" to each retrieved web page according to the following criteria: Suppose the number of content words of the query submitted to Google is k. A general query phrase p containing x content words has a weight x/k, an individual word in the phrase has a weight 1/(2k). An individual query word which is not a part of a query phrase has a weight 1/k. The "degree of satisfaction" of a web page with respect to a query is given by the sum of the weights of the terms and the phrases which appear in a web retrieved page.

(4) Assign a weight to each term which appear in a web retrieved page is as follows:

$$selection_weight(t) = \sum_{i=1}^{n} satisfication_\deg(d_i) \times ratio(di) \times tf_{di}(t) \qquad (2)$$

where *satisfaction_deg(d_i)* is the satisfaction degree of document d_i, *ratio(d_i)* is 1 if the document length *dl(d_i)* is less than or equals to the average document length *avgdl*, otherwise it is given by *avgdl/dl(di)*. $tf_{di}(t)$ is the term frequency of term t in document d_i. The sum is over all top ranked documents.

(5). Terms are ranked in descending order of selection weight, and the top ranked 20 terms are chosen as candidates.

(6). The selection weight of each candidate is normalized between 0 and 0.5 by following the following criterion: if the selection weight is greater or equals to 3, its weight is 0.5, otherwise its weight is given by *selection_weight/(3.0*2)*.

4.2 Combine Pseudo Feedback and Web-assisted Feedback Results

Each of the terms brought in by one of the two processes (pseudo-feedback and web-assisted feedback) will initially be given a weight which is dependent on its correlation with the query terms. When a term is brought in by both processes, their weights are added together.

A term *t* appears frequently in the top retrieved documents either in the initial retrieval of documents in the given collection of documents or in a web search. By using technique described above, it is not known which query term brings in *t*. However *t* may be related to some query term t_1 (i) by being a synonym of t_1; (ii) by being a hyponym of t_1; (iii) by being a coordinate term of t_1; (iv) by being a direct hypernym of t_1; (v) having a definition which contains t_1 or (vi) a definition of t_1 contains *t*. In the cases of (i) and (ii), *t* is a non-dominant of t_1; otherwise, it is brought in by Wordnet in previous phrase. In all these cases, the weight of *t* is given by how it relates to t_1, using the same computation as given by formula (1). In case (iv), the weight *is f(t, S)/F(t) * 1/h*, where S is the synset containing *t* and *h* is the number of direct hypernyms of t_1.

The weight of *t* that based on its correlation with the query in the top retrieved documents in the given collection of documents or web documents, and the weight that based on its relation with a query term t_1 are added together but the sum is bounded by 1.

5. Robust Track

In the robust track, we submit only 1 run to test our system. This run use title only. WordNet is used to disambiguate word senses and supply synonyms, hyponyms, definition words, and compound concepts. Pseudo-feedback and web-assisted feedback are applied. Table 2 gives the average precisions of the run over the entire 249 topics consisting of 200 old topics, the 50 hard topics, and the 49 new topics.

Table 2. Mean Average Precision for TREC 2004 Robust Track

Topics	200 Old Queries	50 Hard Queries	49 New Queries	249 Queries
MAP	0.3047	0.1942	0.3250	0.3087

The average precision gives the overall performance. The individual effectiveness is measured by the (a) number of topics with no relevant document retrieved in the top 10 positions and (b) the area under MAP(X)-vs-X measure where X is the number of topics (queries) having the worst mean precision and MAP(X) is the mean precision of the Xth worst topic [Robust]. These two measures reflect the robustness of any given retrieval strategy. Table 3 gives the number of topics with no relevant document in the top 10 positions for the old, the hard, the new and overall queries sets.

Table 3. Number of Topics with no Relevant Document in the Top 10 Positions

Topics	200 Old Queries	50 Hard Queries	49 New Queries	249 Queries
no-re-l0	11	2	3	16

Table 3 lists the area under MAP(X)-vs-X statistic information. For the entire set of 249 topics, X ranges from 1 to 62. For the set of 200 old topics, X ranges from 1 to 50. For two sets of 50 and 49 topics (50 hard and 49 new), X ranges from 1 to 12.

Table 3. Area under MAP(X)-vs-X evaluation

Topics	200 Old Queries	50 Hard Queries	49 New Queries	249 Queries
MAP(X)-vs-X	0.0265	0.0263	0.0466	0.0284

The Kendall correlation between predicted and actual difficulty of our estimation is 0.623.

6. Conclusion

Our TREC-2004 experiment shows that robust retrieval result can be achieved by: (1) effective use of phrases, (2) a new similarity function capturing the use of phrases, (3) word sense disambiguation and the utilization of synonyms, hyponyms, definition words and compound concepts which are properly chosen. (4) Web does help retrieval. We are experimenting with more complicated techniques of word senses disambiguation in the document retrieval, and the use of more phrases in feedback retrieval which hopefully will yield much better effectiveness in the future.

Reference

[BR99] R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval, ACM Press, Addison-Wesley, 1999.

[BS95] C. Buckley and G. Salton. Optimization of relevance feedback weights. ACM SIGIR95, p351-357

[Brill] Eric Brill. Penn Treebank Tagger, Copyright by M.I.T and the University of Pennsylvania

[Coll97] M. Collins, Three Generative, Lexicalized Models for Statistical Parsing, Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics, 1997.

[Coll99] M. Collins, Head-driven statistical models for natural language parsing. PhD thesis, University of Pennsylvania, 1999.

[Fell98] C. Fellbaum, *WordNet An Electronic Lexical Database*. The MIT Press, 1998.

[GF98] David Grossman and Ophir Frieder, Ad Hoc Information Retrieval: Algorithms and Heuristics, Kluwer Academic Publishers, 1998.

[GW03] L. Grunfeld, K.L. Kwok, N. Dinstl, P. Deng, TREC 2003 Robust, HARD and QA Track Experiments using PIRCS, Queens College, CUNY, p.510, TREC-12, 2003

[Lin94] D. Lin, PRINCIPAR---An Efficient, broad-coverage, principle-based parser. In Proceedings of COLING-94. pp.42--488, Kyoto, Japan, 1994.

[Lin98] Lin, D. Using collocation statistics in information extraction. In Proceedings of the Seventh Message Understanding Conference (MUC-7), 1998

[LL04] S. Liu, F. Liu, C. Yu and W. Meng, An effective approach to document retrieval via utilizing Wordnet and recognizing phrases, ACM SIGIR 2004, pp.266-272.

[LY03] S. Liu, and C. Yu, UIC at TREC-2003: Robust Task, TREC-2003, 2003

[Mill90] G. Miller, WordNet: An On-line Lexical Database, International Journal of Lexicography, Vol. 2, No. 4, 1990.

[Porter] Martin Porter, Porter Stemmer http://www.tartarus.org/~martin/PorterStemmer/index.html

[RW00] Robertson S E, Walker S. Okapi/Keenbow at TREC-8. TREC-8, 2000

[Robust] Robust Track Guidelines. http://trec.nist.gov/act_part/tracks/robust/04.guidelines.html

[YC03] D.L. Yeung, C.L.A. Clarke, G.V. Cormack, T.R. Lynam, E.L. Terra, Task-Specific Query Expansion (MultiText Experiments for TREC 2003), University of Waterloo, p810, TREC-12, 2003