

A Semantics-based Information Retrieval Model

Xiaohua Zhou

College of Information Science & Technology, Drexel University
Xiaohua.Zhou@drexel.edu

Abstract. In this paper, I propose a semantics-based IR model, which uses triplet (concept-relation-concept) instead of keywords to index documents. It uses the sense instead of the string of a term to index document, which not only makes the representation more accuracy, but also well solves the synonym problem. More importantly, this model avoids the frequently occurred situation in keyword-based IR model that two keywords co-occur in a document but they do not have any syntactic or semantic relation at all. Besides, the model well supports the integration with domain ontology. Thus, it is reasonable to expect higher performance on the proposed semantics-based IR model. I present the models and the methods for components of indexing, searching and matching in detail, which documents the technical feasibility. A case study is performed showing the performance improvement of the new model in comparison with keyword-based IR model.

1. Introduction

Most of present information retrieval (IR) systems including general search engines (e.g. Google, Yahoo and MSN) and scientific literature engines (e.g. PubMed and ACM Digital Library) use keywords to query, match and index documents. This traditional keyword-based IR model does not heavily rely on domain knowledge and thus the process of document indexing can be automated efficiently. However, the simplicity and efficiency of the model can't hide its weakness in expression of semantics. First of all, a keyword usually has several senses and its meaning is ambiguous without context. Second, one meaning can be expressed by many keywords, but it is difficult for users to exhaust all possible keywords with same meaning. Last, also the most problematic, the co-occurrence of keywords in a document does not mean that they are related semantically in most cases. To address above-mentioned three problems, I presented a semantics-based IR model in this paper, which is working in conjunction with domain ontology.

Inspired by AI work (Guha, McCool and Fikes 2004), I represented each document by weighted triplets instead of keywords. The triplet is in the form of concept-relation-concept, both concept and relation from controlled vocabulary. It is closer to natural language (which has the basic form of subject-verb-object) than does the keyword and therefore more effective to address user information needs. Also, triplets address the semantic relationship of two sense-disambiguated concepts and thus more expressive than keywords. Moreover, triplet-based IR model can be well integrated with domain knowledge since all concepts and relations are from controlled vocabulary. Thus, the overall performance of IR systems based on the proposed model is expected to be improved.

A large amount of work has been done on improving the performance of keyword-based IR systems. The selection of informative keywords for indexing, for example, is such a method that raises the discriminative power of document presentation. There is also full of work that attempts to address problems of keyword-based IR models mentioned in the opening paragraph. Sense-based IR model (Stokoe and Tait 2004) is designed to solve the ambiguity of keywords and therefore improve the precision of IR. Query expansion techniques (Akrivas et al. 2002) automate the inclusion of synonyms and other related terms of keywords in user query into the system-side query for the main purpose of recall improvement. These new models and methods do bring a little bit rise of performance to IR systems, but not significantly due to a couple of reasons. First of all, these new methods and models do not touch the third problem (keywords co-occurrence) of keyword-based model at all. Second, keyword-based query usually provides little contextual information for the understanding of user information needs. Third, sense disambiguation and query

expansion themselves are tough without domain knowledge. To this sense, the proposed IR model provides a good framework for the improvement of IR performance.

The rest of the paper is organized as follows: Section 2 describes models and methods for document representation (indexing), query formulation, and document matching, respectively. Section 3 discusses the strength and weakness of the proposed model in comparison with keyword-based model. Then a case study is followed in Section 4 to illustrate the model. A short conclusion finishes the paper.

2. Model and Method

A typical IR system is comprised of three core components: document representation (indexing), query representation (searching), and document matching. In this section, I describe the model and method for each of the three components. In order to make the model and method more understandable, a detailed case that uses UMLS (<http://www.nlm.nih.gov/research/umls/>) as domain ontology and PubMed (<http://www.ncbi.nlm.nih.gov/entrez/query.fcgi>) as document collection is illustrated.

2.1 Document Representation

2.1.1 Logic Representation

A triplet is represented in the form $\{first\ concept, relation, second\ concept\}$, where both *first concept* and *second concept* are from concept ontology and *relation* from relation ontology. In some cases, the *relation* deals with only one concept, the *second concept* could be ignored. The *relation* also could be ignored in the case that the relation in context is not for sure or not defined in ontology.

The concept ontology provides information of each concept including its unique identity, definition, semantic types, and indicative strings. Usually, a concept could be associated with several terms. A term in different context could belong to several concepts (senses or meanings). A term has one main string and several morphological variants. Based on the indicative strings, terms can be identified in text. According to the context, the meaning of the term (i.e. the concept) could be further disambiguated.

The relation ontology defines all semantic types and relationships between any two semantic types. This information is very useful. Given two concepts (i.e. terms are already sense disambiguated), for example, the possible relations could be narrowed down by looking up the relation ontology. In return, provided that two terms have certain relation, their senses also could be narrowed down because combinations of some of their senses do not have a relation defined in the relation ontology.

A document is then represented by multiple weighted triplets. The following factors (but not limited to) could be taken into account of the weight of each triplet in a document. First of all, the position the triplet appears in a document is influential to the determination of the weight. For instance, a triplet occurring in title, abstract should be very important; a triplet appearing in the first and last sentence of a paragraph might well be important. Second, the coverage of the triplet is also very important. Most triplets are drawn on a couple of consecutive sentences and the two concepts of the triplet are often syntactic related. They are defined as local triplets. However, some global triplets couldn't be extracted unless the whole document is observed and analyzed. Obviously, global triplets are more important than local triplets. Third, occurrence frequency is very useful for weighting in practice. A triplet is frequently occurred in a document should be very important. A triplet containing frequently occurred concepts might well be important.

2.1.2 Data Structure

The data structure of document representation shown in Figure 1 is a little bit different from its logic representation out of considerations on technical limitations and flexibility in searching and matching.

Doc ID → {*triplet 1, weight 1; triplet 2, weight 2; ...; triplet n, weight n*}
Triplet → {*concept 1, semantic type 1, string 1, relation, concept 2, semantic type 2, string 2*}

Figure 1. The data structure of the representations of documents and triplets

The data structure of triplets provides much redundant information. In phase of indexing, if the term can't be sense disambiguated, only the string of the term is recorded. Another situation to record string instead of concept ID is that a term is guessed but it does not exist in ontology. It is often the case we can not find certain term at a particular moment in domain ontology because ontology is kept growing. But using some shallow methods such as parallel English structure, domain-specific starting words and ending words, we can predict terms with high confidence. In phase of searching, if the term can't be sense disambiguated, it is no choice but submit the string of term to the IR system for searching. To this sense, strings of terms are also required during indexing. The redundancy also offers the flexibility during document matching. With this data structure, for example, it is possible to perform three levels query matching, concept level, string level, and semantic type level, respectively.

2.1.3 Triplet Extraction

Triplet extraction is the most difficult portion of the triplet-based IR model. Though extraction could be manually done, we are more interested in the extraction of triplet from text. In this section, a framework for automated extraction of local triplets will be presented. The framework consists of four components, term identification, term-pair identification, term sense disambiguation, and relation finalization. The techniques used for these components really depend on the domain. In order to show the technical feasibility, I will review or suggest some techniques for each of them.

Term Identification

The task is namely to identify all terms of interest in text. The term identified should be pre-defined in concept ontology. It is generally related to two sub-areas in natural language processing. One is named entity recognition (Maynard 2001). The other is pronominal reference (Dimitrov et al. 2002) that identifies the real term the pronoun refers to in context.

Part of speech (POS) pattern is a general method to search term candidates because most multi-word terms are comprised of nouns and adjectives and end with noun (Zhou et al. 2005). The following four POS patterns (JJ and NN denote adjective and noun respectively), for example, could be used to find term candidates.

- (1) JJ NN NN
- (2) NN NN
- (3) JJ NN
- (4) NN

Each candidate then would be searched through the concept ontology. If it does exist in domain ontology, it is marked as a term for further processing.

A term sometime appear as a pronoun such as *it* in text. It is necessary to find the real term in context the pronoun refers to. Though pronominal reference is very tough area in NLP, some shallow methods achieve acceptable precision and recall (Dimitrov et al. 2002).

Term-pair Identification

This task is to find all term-pairs which are syntactic related within a sentence. A sentence may contain more than two terms. Therefore, it is difficult to determine which term pair is syntactic related. See the following example. Though protein A and C is semantically possible to have an interaction, this particular sentence addresses only the interaction of A and B as well as C and D.

...Unlike the interaction of protein A and B, the interaction of protein C and D will cause...

Linguistic pattern is a very simple method to identify term-pair. But it heavily depends on domains. Also, it is very difficult to exhaust possible patterns. The full use of syntactic parsing result seems an effective and efficient approach to term-pair identification. For instance, if one term serves as subject and the other as object, they are very possibly syntactic related. Among all syntactic parsers, Link Grammar (Sleator and Temperley 1993) is characterized by its rich linkage information. A linkage diagram consists of links between two words. Here I suggest a link grammar based method for term-pair identification (Zhou et al. 2005).

Suppose a node represents a word, and an edge represents a link. Then the linkage diagram of a valid sentence can be looked at as a connected graph. Furthermore, each edge can be weighted against the type of link according to the domain knowledge. Thus, the distance between any word pair can be calculated from the graph. I hypothesize that, for any term in the sentence, it has great chance to be syntactically related with the term that achieves minimum distance with the targeting term.

Term Sense Disambiguation

This task is to determine the concept (sense) the term refers to in the particular context. Word sense disambiguation (WSD) is one of the toughest sub-areas in NLP. In general, there are two approaches, unsupervised and supervised, to WSD (Nancy and Véronis 1998; Zhou and Han 2005). Supervised approaches use sense-tagged corpus to training WSD models by various machine learning methods such as probabilistic models, memory-based learning, neural network, and symbolic rules (Zhou and Han 2005). Unsupervised approaches simply use dictionary knowledge to disambiguate word sense. Usually supervised approaches achieve higher performance than unsupervised because the former are fed more knowledge sources. But it is not very practical to use supervised approaches to disambiguate term sense because it is extremely expensive to build a corpus for all terms in a large growing ontology. Unsupervised approach use only dictionary knowledge. Moreover, it is much less computing intensive than the supervised. Thus, it might well be a good choice for term sense disambiguation.

The iterative approach for WSD (Mihalcea and Moldovan, 2000) reaches 92% precision and 55% recall. Kwong (2001) reports a system with maximum performance, 68.79% precision and 68.80% recall. In general, the performance of unsupervised approach for WSD is acceptable. It is believed that term sense disambiguation in a particular domain will achieve better performance than general WSD because of the following two reasons. One is the fact that the number of sense per tem in a particular domain is, on the average, less than those of general words. The other is that the use of domain knowledge may improve performance. If one term is sense disambiguated, for example, the possible senses of the other in the term-pair may be dramatically narrowed down because they have a semantic relation in the context.

Relation Finalization

This task is to finalize the semantic relation of a term-pair. No matter two terms are sense disambiguated or not, we can get all possible relations of the term-pair by looking up the relation ontology. Then choose one relation according to the context. Relation finalization is similar to term sense disambiguation. I still recommend unsupervised approaches, i.e. use dictionary knowledge such as indicative words, and syntactic and semantic frames to pick the most appropriate relation.

For the purpose of relation finalization, relation ontology should provide some basic knowledge such as indicative words to tell each relation from others. Then some lexicon ontology such as WordNet (Miller et al. 1990) can be used to extend the knowledge of relation.

2.2 Query Representation

As we claimed in the opening of the paper, triplet is more expressive and closer to natural language than does keyword. It is more efficient and effective to transform natural language to triplets than keywords. Thus, I propose the use of natural language for searching interface. Also, with the built-in support of

domain knowledge in form of ontology, it is possible to develop some supplemental tools (e.g. information visualization) to facilitate the formulation of user information needs.

The query submitted to the IR system for document matching falls into two classes. In the case the sense of the term can not be disambiguated, the following query $\{string\ 1, relation, string\ 2\}$ will be sent to the IR system. Query expansion may be performed before user query is submitted to the IR system. A simple strategy is to add all synonyms of the term (no matter which concept it to) to the query. In the ideal case, the term is transformed into concept and then the following query $\{concept\ 1, relation, concept\ 2\}$ is submitted. In this case, query expansion is not necessary any more because all synonyms share one concept ID.

Two practical search interfaces are natural language and Boolean logic, respectively. Natural language is powerful to express their information needs especially for novice because they are not familiar with domain ontology. It also provides rich contextual information for the IR system to disambiguate term sense. But during the transformation of natural language into triplets, it is very possible to introduce information loss, noise, and even error. The conversion of natural language into triplets in phase of searching is same as the procedure done during indexing. It consists of four steps in order: term identification, term-pair identification, term sense disambiguation, and relation finalization.

Boolean logic search provides users especially experts with full control to manipulate the query. Like keyword-based IR systems, triplet-based IR systems allow users to specify any combination of multiple triplets using OR operator or AND operator. Its weakness lies in its requirement of users remembering concepts, and relations in domain ontology. An alternative is to develop some visual supplemental tools to help users find concepts and relations they want.

2.3 Document Matching

Document matching is a procedure to retrieve documents matching the query submitted to the IR system from all indexed documents in the collection. In this framework, I am not going to discuss the algorithms for matching, but the functionalities this component could have.

Single triplet matching does not have to be exact matching. One variant ignores the relation, i.e. it only cares the existence of semantic relation between two terms no matter what the relation is. The second variant ignores the second concept. The last variant uses first concept (term) or second concept (term) only. It is equivalent to sense-based (keyword-based) searching. Document matching can be performed at different level. The narrowest matching is obviously at the concept level. The broadest matching is at the level of semantic type. String level matching is in the middle.

Sorting is one of the most important functionalities of the component of document matching. Obviously, the weight of each triplet matched is a crucial factor. Besides, the degree of matching also affects the relevance of a document to the user query.

3. Discussion

The triplet-based IR model at least theoretically addresses the three problems of keyword-based IR model in one solution. Especially, it avoids the situation that two keywords co-occur in a document but they do not have any syntactic or semantic relation at all. Besides, domain knowledge can be fully integrated with triplet-based IR model, which is quite promising to further improve the performance of IR systems.

In each triplet, term is sense disambiguate, which supposes to raise the precision of IR. Even if the term is not sense disambiguated, its sense will be dramatically narrowed down in association with another term. Also, as I discussed in the preceding sections, triplets, similar to the basic form of natural language, are

more expressive than keywords. Therefore, it is reasonable to expect higher precision of returned documents with triplet matched than those of keyword matched.

The triplet matching is based on concept ID rather than term string. It well solves the synonym problem (the second problem of the keyword-based IR model) because all synonyms share the same concept ID. Even if under some situation the term is not sense disambiguated, we can still easily use query expansion because domain ontology is fed. Thus, the recall of IR is expected to be improved in this sense.

The triplet-based IR model not only has better performance at system side, but also presents friendlier searching interface at user side. It supports natural language based query. It can easily integrate visual tools to help users formulate their information needs. It has a good mechanism for sorting all retrieved documents in order of relevance.

In general, triplet-based IR model is theoretically super than keyword-based IR model in the phase of searching. In the worst case, the former can easily be transformed into the latter by relaxing the criteria of matching. However, indexing of the former is much more difficult than the latter.

First of all, we do not see any promising work to deal with the extraction of global triplets because it involves inference and implicit knowledge (common sense). Failure to extraction of global triplet may decrease the recall of the IR system. Second, the extraction of local triplet is also tough though a four-step framework is provided. Especially, the last two of the four steps, i.e. term sense disambiguation and relation finalization, are quite tough.

The impact of the use of domain ontology is two-fold. On one hand, its integration with IR system will definitely improve the performance of IR. On the other hand, it limits the extension of IR system. As we know, the acquisition of domain ontology is extremely expensive. Usually, domain ontology is kept growing, which gives rise to the problem of synchronization between ontology and indexing. One possible side effect is brought by the modification of concept ID across different version of domain ontology. The other is related with the insertion and deletion of concepts. For example, when a new concept is added to the domain ontology, should we re-index the collection of documents or simply keep it untouched? If yes, it is extremely costly. Otherwise, the performance will be compromised.

4. A Case Study

In the preceding sections, I theoretically describe and analyze a semantics-based IR model. It may improve the performance (precision and recall) of IR systems in comparison with keyword-based IR model. But it needs to be documented by lots of real-world experiments. Since I did not implement the model yet, I manually performed information retrieval task following the method described in Section 2. Then I compared the result with those of keyword-based IR systems.

As we know, hypertension and obesity are often co-occurred. Suppose a research wants to find articles that address the interaction of these two diseases. I use PubMed as the keyword-based IR system, which allows users to use Boolean logic for searching. The keywords used are *hypertension* and *obesity*. I conduct two experiments in total, which analyze the precision and recall of the proposed model, respectively. Because the calculation of precision and recall involves counting of relevant documents, I simply assume even distribution of relevant documents in the returned document set. Because PubMed adopt exact Boolean logic match, this assumption is reasonable.

The first experiment aims to analyze the precision improvement of the proposed model. I search documents through PubMed with co-occurrence of keywords *hypertension* and *obesity* in abstract of the article. Total 6792 documents (March 8, 2005) are returned. Then I manually evaluate the first 100 documents, examining how much percentage of irrelevant documents could be removed by applying the new model and how much percentage of relevant documents the new model fails to retrieve. The domain ontology used is UMLS. The results are shown in Table 1. All 66 irrelevant documents could be removed if new

model is applied. However, the new model may fail to retrieve 10 relevant documents because it fails to extract global triplets.

Categories	Number of Doc	Can Be Removed	Fail to Retrieve
Direct Interaction	34	N/A	10
Indirect Interaction	18	18	N/A
Parallel Elements	28	28	N/A
No relation	20	20	N/A

Table 1. It shows the result of precision improvement analysis for the proposed model. The sample sentences for each category are illustrated in appendix.

The second experiment is designed to analyze the recall improvement of the new model. In this experiment, all synonyms of hypertension (concept ID: C0235220) and obesity (concept ID: C0028754) are added to the query. The original query and expanded query are listed in Figure 2. The results using query expansion should be equivalent to the result of new model. Because the expanded query returns 8062 documents, significantly larger than the number of the original query, 6792, and relevant documents are evenly distributed in the resulting set, I conclude that the recall of IR based on the new model is improved.

Original Query

obesity [Title/Abstract] AND hypertension [Title/Abstract]: 6792

Expanded Query

(obesity [Title/Abstract] OR adiposity [Title/Abstract] OR overweight [Title/Abstract]) AND (hypertension [Title/Abstract] OR high blood pressure [Title/Abstract] OR hyperpiesia [Title/Abstract]): 8062

Figure 2. Original query and expanded query

5. Conclusion

In this paper, I propose a semantics-based IR model, which uses triplet (concept-relation-concept) instead of keywords to index documents. The proposed model theoretically well addresses the three problems of keyword-based IR model. It uses the sense instead of the string of a term to index document, which not only makes the representation more accuracy, but also well solves the synonym problem. More importantly, this model avoids the frequently occurred situation in keyword-based IR model that two keywords co-occur in a document but they do not have any syntactic or semantic relation at all. Besides, the model well supports the integration with domain ontology. Thus, it is reasonable to expect higher performance on the proposed semantics-based IR model.

I present the model and the method for indexing, searching and matching in detail. I conduct a short review on the techniques for local triplet extraction, the most difficult portion of the model, with the conclusion that automated triplet extraction is technically feasible nowadays. Then I theoretically discuss the strength and weakness of the new model. The advantage lies in its good performance in phase of searching while its disadvantage lies in difficulty of triplet extraction and its heavy reliance on domain ontology. Finally, a case study is performed to show the performance improvement of the new model in comparison with keyword-based IR model.

Reference

Akrivas, G., Wallace, M., Andreou, G., Stamou, G., and Kollias S. 2002. Context – Sensitive Semantic Query Expansion, *Proceedings of the IEEE International Conference on Artificial Intelligence Systems (ICAIS)*, Divnomorskoe, Russia.

- Dimitrov, M., Bontcheva, K., Cunningham, H., and Maynard, D. 2002. A Light-weight Approach to Coreference Resolution for Named Entities in Text, *Proceedings of the Fourth Discourse Anaphora and Anaphor Resolution Colloquium (DAARC)*, Lisbon.
- Guha, R., McCool, R. and Fikes, R. 2004. Context for the Semantic Web, Third International Semantic Web Conference, Hiroshima, Japan.
- Kwong, O.Y. 2001. Word Sense Disambiguation with an Integrated Lexical Resources. *Proceedings of the NAACL WordNet and Other Lexical Resources Workshop*.
- Maynard, D., Tablan, V., Ursu, C., Cunningham, H. and Wilks, Y. 2001. Named Entity Recognition from Diverse Text Types, *Recent Advances in Natural Language Processing 2001 Conference*, Tzigrav Chark, Bulgaria.
- Mihalcea, R. & Moldovan, D. 2000. An Iterative Approach to Word Sense Disambiguation. *Proceedings of Flairs 2000*, 219-223. Orlando, USA.
- Miller, G. et al. 1990. WordNet: an On-line Lexical Database, *International Journal of Lexicography*, 235-245.
- Nancy, I. and Véronis, J. 1998. Word sense disambiguation: The state of the art, *Computational Linguistics*, 24(1): 1-40.
- Sleator, D. and Temperley D., 1993. Parsing English with a Link Grammar, *Third International Workshop on Parsing Technologies*.
- Stokoe, C. and Tait, J. I. 2004. Towards a Sense Based Document Representation for Information Retrieval, in *Proceedings of the Twelfth Text REtrieval Conference (TREC)*, Gaithersburg M.D.
- Zhou, X. and Han, H. 2005. Survey of Word Sense Disambiguation Approaches, *Proceedings of the 18th International Florida AI Research Society Conference*, Clearwater Beach, Florida, USA.
- Zhou, X., Han, H., Chankai, I., Prestrud, A., and Brooks, A. 2005. Converting Semi-structured Clinical Medical Records into Information and Knowledge, *the International Workshop on Biomedical Data Engineering*, Tokyo, Japan.

Appendix

Sample Text:

Direct Interaction

Furthermore, in contrast to previous data, *obesity* alone does not increase the risk of perimenopausal fracture, but in association with *hypertension* the risk seems to be markedly elevated.

Indirect Interaction

..., *hypertension*, diabetes, and *obesity* are associated with increased arterial stiffness...

Parallel Elements

In the study, 48% participants had *obesity*, 17% had *hypertension*....

No Relation

PURPOSE: To describe the anesthetic management of a patient with extreme *obesity* undergoing bariatric surgery whose intraoperative narcotic management was entirely substituted with dexmedetomidine.
CLINICAL FEATURES: We describe a 433-kg morbidly obese patient with obstructive sleep apnea and pulmonary *hypertension* who underwent Roux-en-Y gastric bypass.