ABSTRACT

The PHASAR (Phrase-based Accurate Search And Retrieval) system is an Information Retrieval and Text Mining system for professional applications. Following the implementation of a prototype in the biomedical domain, we are currently implementing PHASAR for professional search in the intellectual property (IP) domain.

General Terms

Professional search, Interactive search, Intellectual Property

1. INTRODUCTION

Professional search may be distinguished from what could be termed incidental search by the following characteristics: (1) The search is performed by professionals, in their own area of expertise; (2) The search is worth investing some (expensive) time and effort; (3) The search is over a very large collection of documents, many of which may be relevant; (4) The information need is clear but complex, the user can recognize relevant answers; (5) The information need may have to be answered by gathering (passages from) many documents; and (6) Repetitions of the search process with small modifications in the query are routine [3].

The prototype of the PHASAR search system [3] has been developed for professional search on the Medline data collection comprising 18,837,276 scientific abstracts from the biomedical domain. The PHASAR system expects a query to consist of phrases rather than keywords. In an interactive process, the searcher indicates which phrases should occur in the documents for them to be relevant to his/her information need (a form of query-by-example).

Recently, the project Text Mining for Intellectual Property (TM4IP) has started at the University of Nijmegen. In this project, the PHASAR system will be implemented for intellectual property search, i.e., search in a database of 9.5 million full-text patent documents. It has been observed that patent searchers prefer Boolean search over ranked search because they desire full control over precision and recall. They are willing to invest work in order to ensure that they retrieve all relevant information pertaining to a query. These user characteristics match well with PHASAR’s interactive formulation of phrase queries.

In this poster, we present the PHASAR search system in its current form and we discuss the challenges that we meet in implementing PHASAR for the intellectual property domain.

2. THE PHASAR SEARCH SYSTEM

In this section, we present the basic principles of the PHASAR system following one specific example: the question “What genes are induced by LPS in diabetic mice?”, taken from the set of queries used in the TREC 2007 genomics track2.

PHASAR performs sentence retrieval and presents the results in the form of short passages with a link to the complete document.

PHASAR uses phrases as terms. In the classical approach, a phrase is a sequence of (consecutive) words (e.g., using the sequence diabetic mice as a query instead of the separate words diabetic and mice). Instead of this type of word sequences, PHASAR uses Dependency Triplets (DTs) as terms. A dependency triplet is a pair of (lemmatized) words with their relation, e.g., [mouse, ATTR, diabetic]. PHASAR’s DT framework is based on the principle of aboutness. DTs have been used successfully in Question Answering [1] for the precise matching of input questions to their answers. In PHASAR, the DTs are obtained from both the indexed documents and the queries in the following steps: dependency parsing is followed by a transduction to DTs, in which (syntactic) variations are normalized onto a common representation. E.g., PHASAR maps the sentences “TNF-alpha is induced by LPS” and “LPS induces TNF-alpha” to a single representation in the index.

PHASAR expects phrase queries that are matched to

1See http://www.phasar.cs.ru.nl/TM4IP.html
2http://ir.ohsu.edu/genomics/
the index of DTs. In the current PHASAR search interface, the searcher fills (at least two of the three) slots for subject, verb and object. Taking ‘query-by-example’ literally, we can use the query \[\text{LPS induce TNF-alpha}\] in order to find passages confirming that “TNF-alpha is induced by LPS”. Replacing the contents of one of the slots by a question mark, PHASAR shows (besides the sentences from the corpus that match the query) a list of the terms that occur in the \(\bar{?}\)-position of the phrase, ordered by document frequency.

The user generalizes and specializes the query interactively. Query generalization can be achieved by either joining multiple terms using the or operator, or by setting a context from which the searcher fills (at least two of the three) slots for subject, the index of DTs. In the current PHASAR search interface, the searcher fills (at least two of the three) slots for subject, verb and object. Taking ‘query-by-example’ literally, we can use the query \[\text{LPS induce TNF-alpha}\] in order to find passages confirming that “TNF-alpha is induced by LPS”. Replacing the contents of one of the slots by a question mark, PHASAR shows (besides the sentences from the corpus that match the query) a list of the terms that occur in the \(\bar{?}\)-position of the phrase, ordered by document frequency.

The user generalizes and specializes the query interactively. Query generalization can be achieved by either joining multiple terms using the or operator, or by setting a context from which the searcher fills (at least two of the three) slots for subject, the index of DTs. In the current PHASAR search interface, the searcher fills (at least two of the three) slots for subject, verb and object. Taking ‘query-by-example’ literally, we can use the query \[\text{LPS induce TNF-alpha}\] in order to find passages confirming that “TNF-alpha is induced by LPS”. Replacing the contents of one of the slots by a question mark, PHASAR shows (besides the sentences from the corpus that match the query) a list of the terms that occur in the \(\bar{?}\)-position of the phrase, ordered by document frequency.

The user generalizes and specializes the query interactively. Query generalization can be achieved by either joining multiple terms using the or operator, or by setting a context from which the searcher fills (at least two of the three) slots for subject, the index of DTs. In the current PHASAR search interface, the searcher fills (at least two of the three) slots for subject, verb and object. Taking ‘query-by-example’ literally, we can use the query \[\text{LPS induce TNF-alpha}\] in order to find passages confirming that “TNF-alpha is induced by LPS”. Replacing the contents of one of the slots by a question mark, PHASAR shows (besides the sentences from the corpus that match the query) a list of the terms that occur in the \(\bar{?}\)-position of the phrase, ordered by document frequency.

The user generalizes and specializes the query interactively. Query generalization can be achieved by either joining multiple terms using the or operator, or by setting a context from which the searcher fills (at least two of the three) slots for subject, the index of DTs. In the current PHASAR search interface, the searcher fills (at least two of the three) slots for subject, verb and object. Taking ‘query-by-example’ literally, we can use the query \[\text{LPS induce TNF-alpha}\] in order to find passages confirming that “TNF-alpha is induced by LPS”. Replacing the contents of one of the slots by a question mark, PHASAR shows (besides the sentences from the corpus that match the query) a list of the terms that occur in the \(\bar{?}\)-position of the phrase, ordered by document frequency.

The user generalizes and specializes the query interactively. Query generalization can be achieved by either joining multiple terms using the or operator, or by setting a context from which the searcher fills (at least two of the three) slots for subject, the index of DTs. In the current PHASAR search interface, the searcher fills (at least two of the three) slots for subject, verb and object. Taking ‘query-by-example’ literally, we can use the query \[\text{LPS induce TNF-alpha}\] in order to find passages confirming that “TNF-alpha is induced by LPS”. Replacing the contents of one of the slots by a question mark, PHASAR shows (besides the sentences from the corpus that match the query) a list of the terms that occur in the \(\bar{?}\)-position of the phrase, ordered by document frequency.

The user generalizes and specializes the query interactively. Query generalization can be achieved by either joining multiple terms using the or operator, or by setting a context from which the searcher fills (at least two of the three) slots for subject, the index of DTs. In the current PHASAR search interface, the searcher fills (at least two of the three) slots for subject, verb and object. Taking ‘query-by-example’ literally, we can use the query \[\text{LPS induce TNF-alpha}\] in order to find passages confirming that “TNF-alpha is induced by LPS”. Replacing the contents of one of the slots by a question mark, PHASAR shows (besides the sentences from the corpus that match the query) a list of the terms that occur in the \(\bar{?}\)-position of the phrase, ordered by document frequency.

The user generalizes and specializes the query interactively. Query generalization can be achieved by either joining multiple terms using the or operator, or by setting a context from which the searcher fills (at least two of the three) slots for subject, the index of DTs. In the current PHASAR search interface, the searcher fills (at least two of the three) slots for subject, verb and object. Taking ‘query-by-example’ literally, we can use the query \[\text{LPS induce TNF-alpha}\] in order to find passages confirming that “TNF-alpha is induced by LPS”. Replacing the contents of one of the slots by a question mark, PHASAR shows (besides the sentences from the corpus that match the query) a list of the terms that occur in the \(\bar{?}\)-position of the phrase, ordered by document frequency.

The user generalizes and specializes the query interactively. Query generalization can be achieved by either joining multiple terms using the or operator, or by setting a context from which the searcher fills (at least two of the three) slots for subject, the index of DTs. In the current PHASAR search interface, the searcher fills (at least two of the three) slots for subject, verb and object. Taking ‘query-by-example’ literally, we can use the query \[\text{LPS induce TNF-alpha}\] in order to find passages confirming that “TNF-alpha is induced by LPS”. Replacing the contents of one of the slots by a question mark, PHASAR shows (besides the sentences from the corpus that match the query) a list of the terms that occur in the \(\bar{?}\)-position of the phrase, ordered by document frequency.

The user generalizes and specializes the query interactively. Query generalization can be achieved by either joining multiple terms using the or operator, or by setting a context from which the searcher fills (at least two of the three) slots for subject, the index of DTs. In the current PHASAR search interface, the searcher fills (at least two of the three) slots for subject, verb and object. Taking ‘query-by-example’ literally, we can use the query \[\text{LPS induce TNF-alpha}\] in order to find passages confirming that “TNF-alpha is induced by LPS”. Replacing the contents of one of the slots by a question mark, PHASAR shows (besides the sentences from the corpus that match the query) a list of the terms that occur in the \(\bar{?}\)-position of the phrase, ordered by document frequency.

The user generalizes and specializes the query interactively. Query generalization can be achieved by either joining multiple terms using the or operator, or by setting a context from which the searcher fills (at least two of the three) slots for subject, the index of DTs. In the current PHASAR search interface, the searcher fills (at least two of the three) slots for subject, verb and object. Taking ‘query-by-example’ literally, we can use the query \[\text{LPS induce TNF-alpha}\] in order to find passages confirming that “TNF-alpha is induced by LPS”. Replacing the contents of one of the slots by a question mark, PHASAR shows (besides the sentences from the corpus that match the query) a list of the terms that occur in the \(\bar{?}\)-position of the phrase, ordered by document frequency.

In the future, we also plan to extend the descriptive framework of the parser grammar with aspects of language that are not directly related to the aboutness of a sentence such as verb modalities and negation.

A second challenge lies in the further extension of the normalization process that takes place in the transduction from parse trees to triplets. In the current version, syntactic variations such as passive versus active voice are already covered (as exemplified in Section 2). A very important addition is the implementation of anaphora resolution using the statistics of the DTs. Using the current parser, we find among the most frequent dependency triplets many triplets with anaphora such as \([i t, S U B J, formed]\). An essential part of the normalization process is to match these anaphora to the correct antecedent, in order to have access to the information contained in the text. Another part of the normalization process is to match synonyms and to resolve abbreviations for technical terms, which are frequent in patents documents.

The last challenge is more of a meta-challenge: evaluating the PHASAR/TM4IP system during its development. We consider two areas of evaluation: (1) evaluation of the accuracy of the hybrid dependency parser and the normalizing transduction, for which we need suitable gold standards, and (2) evaluation of the PHASAR search system on intellectual property data. In 2009 we intend to participate in the first edition of the CLEF-IP track\(^4\). This will provide us with the opportunity to use common evaluation data. However, since PHASAR expects phrasal queries in an interactive setting, we will not be able to perform a fully automatic evaluation.

4 REFERENCES

---

Figure 1: Screen shot of PHASAR displaying the results from the Medline corpus for the structured query \[[\text{LPS induce} \bar{?}]\] in the context of “diabetic mice”.

---


---

A steering system is provided for holding the steering wheel generally parallel to rear wheels on the tractor and for turning the steering wheel through an angle and opposite to a steering angle of the tractor to bring the plow assembly behind the tractor during turns. (doc. XX000200)