Information Retrieval Techniques for Templated Queries

Giridhar Kumaran & James Allan
Department of Computer Science
University of Massachusetts Amherst
Computer Science Building
140 Governors Drive
Amherst MA 01003 USA
{giridhar, allan}@cs.umass.edu

Abstract

Queries in template form are gaining in popularity as a means of conveying specific information needs to search engines. We explore the utility of Information Retrieval (IR) techniques in the context of templated queries. Our investigations show that IR techniques known to be well-suited for *ad hoc* retrieval don't seamlessly extend to the case of templated queries. We show that what works is a combination of IR techniques and intuition-driven modifications to the templated queries, resulting in statistically significant improvements over the baseline.

Introduction

Queries in template form are a new and emerging paradigm in the way users can convey complex information needs to search engines. Templated queries find most use when the same type of information is repeatedly queried about. The following templated query from the TREC 2006 complex Interactive Question Answering (ciQA) (Dang et al.,2006) track is one such example. It consists of two parts – the template itself and a narrative.

Template: What evidence is there for transport of [*drugs*] from [*Bonaire*] to [*the United States*]?
[G] [L_1] [L_2]
Narrative: The analyst would like to know of efforts made to discourage narco traffickers from using Bonaire as a transit point for drugs to the United States. Specifically, the analyst would like to know of any efforts by local authorities as well as the international community.

A user interested in the general topic of transfer of certain goods from one location to another can use this template by simply instantiating the free slots with the goods and locations she is interested in. To further expatiate upon the information need, the user can also include free-form text in the narrative section. Once the user has instantiated a template, the next task is to convert this templated query into one understandable by a search engine. Using the entire query in its original form can result in poor effectiveness due to the presence of extraneous terms like *analyst*, *specifically*, and *international community*. Hence there is need for an intermediary system, or a query processor, to convert this complex information need into an effective query in the search engine's query language. In creating this effective query, additional information about the template itself can be leveraged. For example, for the template given above, we can create a query that specifies that the locations *Bonaire* and *United States* should co-occur in a document, and that the term *drugs* should be expanded with a list of drugs. The possibility of incorporating template-specific features sets templated querying apart from the domain of free-form searching.

Some earlier analogies to templated queries include the advanced search options available in most commercial search engines. These options allowed users to select date ranges, document types, URL restrictions and so on. Using these options the user could focus and clarify their queries. However, use of advanced search options is not very popular (Spinks et al., 1999).

Non-commercial search engines like INQUERY (Turtle & Croft, 1991) and Indri[1] provide a framework for posing complex information needs using a structured query language. Using the structured query language, a user can create queries indicating phrases, patterns of text, terms within certain proximity, synonyms, term absence, term presence and so on. However harnessing the full power of structured query languages requires thorough knowledge of not only the query language, but implementation details of individual features.

Our focus in this paper is on the templated queries defined as part of the Defense Advanced Research Projects (DARPA) Global Autonomous Language Exploitation (GALE) program[2]. The goal of this program is to create a system that will quickly return specific information relating to a user's information need, from broadcast and newswire sources. The sources could be in English, Chinese or Arabic languages. Creating such a system requires the amalgamation of technologies relating to Machine Translation, Automatic Speech Recognition, Information Retrieval, Information Extraction and Text Summarization. While the final system was required to output snippets of relevant text, our goal in this paper is centered on the Information Retrieval aspect - retrieval of relevant documents with high precision to serve as high-quality input for the downstream processes of text summarization and snippet extraction. It is worth noting that we could have used ciQA or the TREC Genomics Track as a source of templated queries. However, both tracks have far fewer and less diverse sets of queries.

Within GALE's first evaluation, the information needs of end-users of the final system were conveyed through one of ten possible templates. In this paper, we focus on seven of them, listed in Table 1. The lack of adequate number of training and test queries led us to drop the remaining three.

Template Number	Template
1	LIST FACTS ABOUT EVENTS DESCRIBED AS FOLLOWS: [event]
2	PRODUCE A BIOGRAPHY OF [person]
3	PROVIDE INFORMATION ON [organization]
4	FIND STATEMENTS MADE BY OR ATTRIBUTED TO [person] ON [topic(s)]
5	DESCRIBE THE PROSECUTION OF [person] FOR [crime]
6	HOW DID [country] REACT TO [event]?
7	IDENTIFY PERSONS ASSOCIATED WITH [organization] WHO HAVE BEEN INDICTED ALONG WITH HOW THEY'RE RELATED

Table 1. The seven GALE templates.

In this paper we attempt to blend the utility of templated queries with the expressive power of structured query languages. We apply a variety of IR techniques to create structured queries from the templated ones. We show that not all IR techniques are suited for satisfying the information needs conveyed through templated queries, and devise combinations of IR techniques and manual formulations to build effective structured queries.

[1] http://www.lemurproject.org/indri
[2] http://www.darpa.mil/ipto/programs/gale/index.htm

Template Structure

In addition to the query statement with placeholders for specific instantiations, templated queries in the GALE program also contain some or all of a number of types of additional restrictions and supportive information. These include date, location and source constraints, additional (related) terms the user though were useful, and terms/phrases to be treated as equivalent. With the GALE XML query in Figure 1 as an example, we will now elaborate on these elements.

```
<?xml version="1.0" encoding="UTF-8"?>
 <!DOCTYPE queries SYSTEM "query.dtd">
<queries>
        <query template-number="5" query-id="T028">
                <query-text>FIND STATEMENTS MADE BY OR ATTRIBUTED TO [Alexander
                Downer] ON [splits within the EU over Iraq] </query-text>
                <query-arg arg-num="1" arg-type="Person">
                        <arg-value>Alexander Downer</arg-value>
                </query-arg>
                <query-arg arg-num="2" arg-type="Topic">
                        <arg-value>splits within the EU over Iraq</arg-value>
                </query-arg>
                <query-date date-type="Source">                          (1)
                        <start-date>04 April 2003</start-date>           (1)
                        <end-date>01 May 2003</end-date>                 (1)
                </query-date>                                            (1)
                <context>
                        <location>                                       (2)
                                <loc>Canberra</loc>                      (2)
                        </location>                                      (2)
                        <source-type>                                    (3)
                                <source>ALH</source>                     (3)
                                <source>VOA</source>                     (3)
                                <source>CNN</source>                     (3)
                        </source-type>                                   (3)
                        <equivalent-terms>                               (4)
                                <term> Australian Foreign Minister</term>(4)
                                <term> Alexander Downer</term>           (4)
                        </equivalent-terms>                              (4)
                        <equivalent-terms>                               (4)
                                <term> EU</term>                         (4)
                                <term> European Union</term>             (4)
                        </equivalent-terms>                              (4)
                        <related-words>                                  (4)
                                <term> divisions</term>                  (4)
                                <term> challenges</term>s                (4)
                                <term> security</term>                   (4)
                        </related-words>                                 (4)
                </context>
</query>
</queries>
```

Figure 1 : An example GALE XML templated query

1. Query Date

The query date can be one of two types – *Source* or *Activity*. By specifying the date to be of type 'Source', the user can indicate that she is interested in documents that were published or

broadcast in a particular time span. An 'Activity' date restriction however implies that any documents that refer to events that occurred in a particular time span, irrespective of the actual publish or broadcast date, are candidates for retrieval. For example, specifying an activity date 9/11 means that any document that refers to that date is to considered.

2. Location

As the name suggests, the user can indicate that she is interested in events that happened at a particular location. For example, a user might be interested in the statements (Template 4) made by a person on a particular topic *at* a particular location.

3. Source Type

This field is used to indicate the news sources that the user wants to restrict the search to. The news sources are specified by a shorthand notation three characters long. For example, AFA refers to the Arabic edition of Agence France Presse.

4. Related Words and Equivalent Terms

The user can indicate that certain groups of terms and phrases can be used interchangeably. In the example shown in Figure 1 *Australian Foreign Minister* and *Alexander Downer* are specified by the user as being equivalent. We interpreted the equivalent terms as synonyms. The user can also provide some terms they believe are related to the original query and help in retrieval. More often than not, these terms ended up degrading retrieval performance rather than actually helping it.

Using all the fields mentioned above, a user can create a very rich representation of not only her information need, but also her current knowledge. Our task was to create a query processor that would accept the GALE XML queries as input, and convert them into Indri (see following section) queries taking into consideration not only the various specifications provided by the user but also the nature of the template and relevant documents.

Experimental Setup

The goal of the GALE program was to develop a system that while reporting in English derived its responses from newswire and broadcast news in English, Chinese and Arabic. To this end, the data came from a variety of news sources.

Data

The articles from broadcast sources were provided as Automatic Speech Recognition (ASR) output, and the articles in Chinese and Arabic newswire and ASR were translated into English automatically using machine translation techniques. Table 2 provides a list of the news sources.

	Sources
Arabic	Agence France Presse, Al Hurra, Al Jazeera, An-Nahar, LBC, Ummah , Xinhua
Chinese	Agence France Presse, China News Agency, China Central TV, NTD TV, Xinhua, Zaobao
English	Agence France Presse, Associated Press, China News Agency, CNN, LA Times/Washington Post, MSNBC, NBC, New York Times, Ummah, Xinhua

Table 2 : Sources for Arabic, Chinese and English newswire and broadcast news.

All the data was annotated with *entities*, *values*, *temporal expressions*, *relations*, and *events* as outlined in the Automatic Content Extraction[3] (ACE) program. These annotations played a useful role in template-specific processing of the queries.

For training we were provided the Topic Detection and Tracking (TDT) data, specifically TDT-4 and TDT-5, as well as additional GALE-specific evaluation corpora. The TDT4 corpus contained 38,991 documents, TDT5 407,503 documents and Evaluation corpus 215,174 documents.

Queries

Of a set of 94 templated queries available with relevance judgments, we used 60% for training and 40% for testing. The break-up of queries by template in given in Table 3 (Table 1 provides a guide to the templates).

Template	1	2	3	4	5	6	7
Training Queries	8	9	7	14	6	10	6
Test Queries	4	5	4	8	4	6	4

Table 3 : The number of training and test queries per template

Indexing and Search

We used an offline-annotation supporting version of the open-source Indri search engine for our experiments. The Indri search engine was developed as part of the Lemur[4] project – a collaborative effort between the University of Massachusetts Amherst and Carnegie Mellon University. Indri provides a competitive search engine environment using a retrieval model that merges the best of language modeling techniques and inference networks (Metzler and Croft, 2004*b*). While the inference network-based retrieval framework of Indri permits the use of structured queries, the use of language modeling techniques provides better estimates of probabilities for query evaluation.

The Indri indexer provided the capability to index and retrieve not only the text of the documents, but also specific annotations of interest. This was particularly useful in situations like applying date restrictions to the retrieved content. The Indri structured query language[5] provided a robust and flexible framework for creating complex queries from the templates. Among others, the query language allowed complex phrase matching, synonyms, weighted expressions, boolean filtering, numeric (and dated) fields, and the extensive use of document structure (annotations).

Evaluation

For all systems, we report mean average precision (MAP) and precision at 10 (p@10). MAP is the most widely used measure in Information Retrieval. While precision is the fraction of the retrieved documents that are relevant, average precision is a single value obtained by averaging the precision values at each new relevant document observed. MAP is the arithmetic mean of the average precisions of a set of queries. Since the documents retrieved by our system are intended for use in summarization and snippet extraction, it was particularly important that we provided high-precision results at the top of the ranked list. Hence we chose

[3] http://www.nist.gov/speech/tests/ace/
[4] http://www.lemurproject.org
[5] http://www.lemurproject.org/lemur/IndriQueryLanguage.php

to also report the p@10 values, which reflect the percentage of documents in the top 10 of the ranked list that are relevant to the query.

Information Retrieval Techniques

To convert the GALE templated queries into structured queries that can be understood by the Indri search engine, we utilized a number of template-independent and template-dependent procedures. An important thing to note is that the techniques were applied to templates only when appropriate or when the required data was available. We now list the techniques utilized, along with accompanying examples, intuitions and justifications as appropriate.

Date Handling

1. Source Dates
   The handling of source (or publishing) date restrictions involved converting the specified dates in the templates into YYYYMMDD format and indexing the publication or broadcast dates in the documents as numeric fields. Using the filtering (#filreq) and numeric operators (#between, #less and #greater) in the Indri query language we were able to include the source date specifications as part of an Indri query.
2. Activity Dates
   The handling of activity dates required identifying mentions of dates in the documents. The data was annotated with such information, including normalized mentions of dates like *last Monday*, *the next day* and so on, we were unable to effectively utilize them. We investigated if treating Activity dates as Source dates was a good proposition. We picked a set of 30 queries with Activity date restrictions, and looked at the Source dates of relevant documents. We found that with the exception of three queries (Figure 2), all queries with Activity date restrictions had relevant documents with Source date greater than the start Activity date. Hence we chose to handle Activity dates by using a filter on the corresponding Source date such that only documents with a Source date greater than the start Activity date were considered.

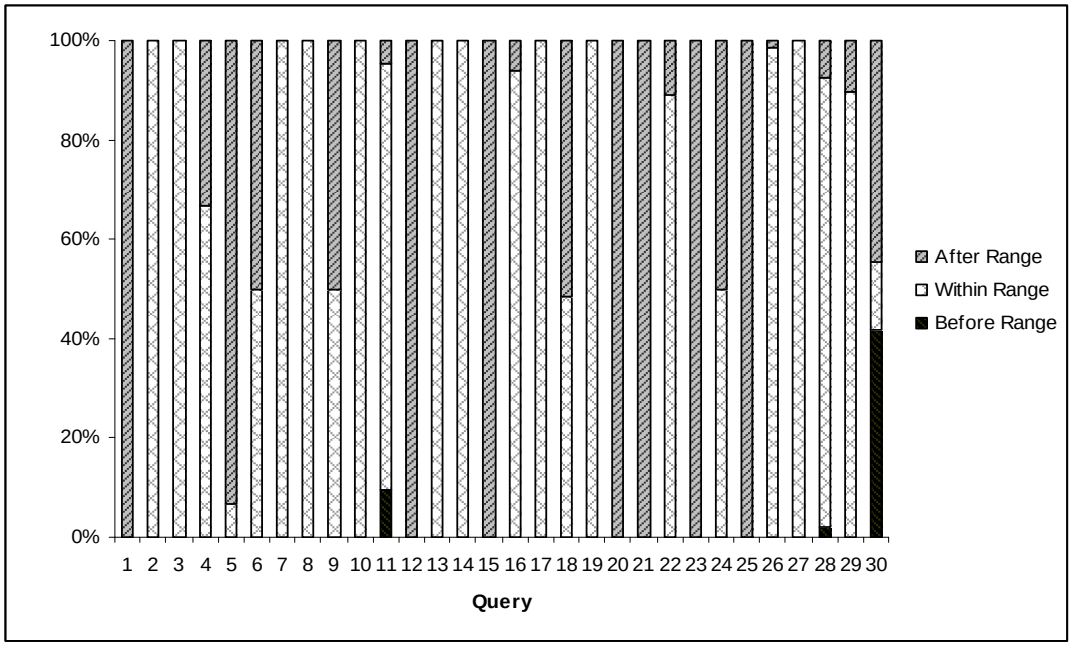

Figure 2 : Distribution of relevant documents by Source dates corresponding to specified Activity dates.

Simple Bag of Words (BOW[6])

By pulling together all the query arguments as well as related and equivalent terms we created a simple bag-of-words query – similar to free-form querying, the most popular paradigm in Information Retrieval. The equivalent terms were treated as synonyms. Figure 3 shows the query we generated for the example template in Figure 1. The #filreq operator is used to filter the collection by the source date range indicated by the #between operator. The #1 operator is used to constrain the enclosed terms to form a phrase, while the #syn operator is used to treat the enclosed terms/phrases as synonyms.

```
#filreq( #between(sourcedate 20030404 20030500)
        #combine (
                Alexander Downer        splits within the EU over Iraq      Canberra
                #syn (
                        #1 (Australian Foreign Minister) #1(Alexander Downer)
                    )
                #syn (
                        #1 (EU)          #1(European Union)
                    )
                divisions challenges security
                 )
        )
```

Figure 3 : The Indri structured query obtained using a bag-of-words approach

Related words weighting (weightREL)

The related words were included by the user based on the assumption that these terms will help the retrieval process. However, there is a possibility of vocabulary mismatch – relevant documents might contain synonyms of the related terms or the same concepts might be expressed in a way the user did not anticipate. We down-weighted the related terms in an effort to solve this problem. Parameter sweeps on the training queries led us to identify an optimal weight of 0.05 for the related terms and a weight of 0.95 for the rest. Figure 4 shows the modification to the query in Figure 4 due to this weighting scheme.

```
#filreq( #between(sourcedate 20030404 20030500)
        #weight (
        0.95 #combine (
                        Alexander Downer        splits within the EU over Iraq      Canberra
                        #syn    (
                                #1 (Australian Foreign Minister) #1(Alexander Downer)
                                )
                        #syn (
                                #1 (EU)          #1(European Union)
                                 )
                        )
        0.05 #combine (
                        divisions challenges security
                        )
                 )
        )
```

Figure 4 : Modified query due to selective weighting of related terms

[6] The term in parenthesis is used to identify runs that used that particular technique in addition to all the techniques in the list before it.

Phrases (Phrases)

The use of phrases (Croft et al., 1991; Fagan, 1987) in queries is known to enhance precision in a number of Information Retrieval tasks. Consider the case of the name *Alexander Downer*. When used as a simple query, there is possibility that a number of documents with references to Alexander, the more common of the two terms, will be retrieved even though the term Downer might not exist. If we instead searched with the phrase "Alexander Downer", we can ensure that only documents referring to that person are retrieved, omitting spurious matches and improving precision. Such a procedure however has a drawback. If either of the terms is misspelled unintentionally or due to cultural differences, there is a possibility of missing some relevant documents. To alleviate this problem, we included the individual terms as well as the phrase in the query.

Dependence Models (DM)

Dependence models (Metzler and Croft, 2004*a*; Gao et al., 2004) can be viewed as a query expansion mechanism that takes in to consideration the co-occurrence of query terms. While various statistical models in Information Retrieval consider query terms to occur independent of each other for easier treatment, the fact remains that co-occurrence of terms is an important though difficult to incorporate factor that can lead to improved retrieval performance. The dependence model builds upon this idea by considering combinations of query terms with proximity constraints. Since a complete treatment of dependence models is beyond the scope of this paper, we provide an example of how a query is translated by a dependence model.

Let us assume that the initial query is *Australian Foreign Minister*. Applying the dependence model to this query results in the Indri query shown in Figure 5.

```
#weight( 0.8 #combine(Australian Foreign Minister)
         0.1 #combine(#1(Australian Foreign)
                      #1(Foreign Minister)
                      #1(Australian Foreign Minister))
         0.1 #combine(#uw8( Foreign Minister)
                      #uw8(Australian Minister)
                      #uw8(Australian Foreign)
                      #uw12(Australian Foreign Minister)
                      )
       )
```

Figure 5 : Dependence model query

We can observe that the model, after weighting the original query high (0.8), considers combinations of terms occurring as phrases with a lower weight (0.1), and then lower weighted combinations of the terms again with looser proximity constraints. It is easy to imagine the combinatorial explosion that takes place as the number of query terms increases.

Annotations (Annotations)

Annotations are widely used in Information Retrieval-related fields like Question Answering. Once the type of the question is determined, QA systems usually look for specific types of answers using available annotations. For example, for a query like *"Where is the Taj Mahal?"* once the question is identified as requiring the location of a building, the query is re-written to focus on *location* entities. The availability of a large number of annotations (see Appendix 1) gave us a similar opportunity to specify general types of information we were interested in on a per-template basis. For example, for the query template on Biographies, one would expect relevant documents to include information about the person's date of birth, age, professional

and social acquaintances, organizations they were involved with and so on. Hence we included a search for any instances of annotations of type social, organization affiliations, general affiliations and "life" events (see Appendix). We also put to use the fact that in most templates the type of the arguments was known in advance. We included constraints in the query to convey that particular query terms were to be associated with annotations of particular types. For example, if we knew that an argument was the name of an organization, we restricted the search using the associated terms to only terms annotated as *organization* in the documents. Table 4 provides a guide to the annotation-type query aspects we included on a template-dependent basis, while Figure 6 provides an example of actual usage.

Template	Annotations
2. PRODUCE A BIOGRAPHY OF [person]	Constraining the search for [person] to *person* annotations; Including search for *life* events, *social* contacts, *organization affiliations* and *general affiliations* in the original query
3. PROVIDE INFORMATION ON [organization]	Constraining the search for [organization] to *organization* annotations
4. FIND STATEMENTS MADE BY OR ATTRIBUTED TO [person] ON [topic(s)]	Constraining the search for [person] to *person* annotations
5. DESCRIBE THE PROSECUTION OF [person] FOR [crime]	Constraining the search for [person] to *person* annotations; Including search for *justice* events
6. HOW DID [country] REACT TO [event]?	Constraining the search for [country] to *geo-political entity* annotations
7. IDENTIFY PERSONS ASSOCIATED WITH [organization] WHO HAVE BEEN INDICTED ALONG WITH HOW THEY'RE RELATED	Constraining the search for [organization] to *organization* annotations; Including search for *justice-arrest-jail* events in the original query

Table 4 : Summary of the annotations used in final queries for each template

```
#filreq( #between(sourcedate 20030404 20030500)
        #weight (
        0.95 #combine (
                #any:life    #any:per-soc     #any:org-aff   #any:gen-aff
                        #1(Alexander Downer).per
                        #syn    (
                        #1 (Australian Foreign Minister)     #1(Alexander Downer) .per
                                )
                        )
        0.05 #combine (
                        life political milestones controversies elections opponents
                        )
                )
        )
```

Figure 6 : Indri query that includes annotations for a hypothetical query 'PRODUCE A BIOGRAPHY OF [*Alexander Downer*]'

Template Specific techniques (tempSpecific)

The distinguishing feature of templated queries is that they are directed at particular types of information. In such a situation, simply relying on the query terms to return documents relevant to the information need is not enough. For the query 'FIND STATEMENTS MADE BY [*Alexander Downer*]', simply using the terms *Alexander* and *Downer* as query terms will retrieve a number of documents related to the person, but not necessarily containing statements made by him. Hence there is a need for including phrases or patterns of terms (Ravichandran & Hovy, 2001) in the query to direct it towards retrieving the actual type of information desired. These additions to the query were made mostly through manual effort driven by intuition and data exploration. Table 5 contains the template-specific additions we made to queries and a brief reason for our choices.

Template	Addition to query	Examples
4. FIND STATEMENTS MADE BY OR ATTRIBUTED TO [person] ON [topic(s)]	[person]'s name occurs within five terms of any of the terms *respond, clarify, quote, cite, said, reported, asserted, replied, insist, denied, speaking, called, ,reiterated, spokesman, spokeswoman, spokesperson, accused, told, declared, charged, address, message, call, appealed, acknowledge*	*Alexander Downer*, the Australian Foreign Minister *said…* The *Pope declared…* *Speaking* to reporters, *Hussein* said…
5. DESCRIBE THE RELATIONSHIP OF [person/org] TO [person/org]	$[person/org]_1$ and $[person/org]_2$ to occur within a term window of size 100	*Iran* today responded to….*IAEA* declared that nuclear…
6. HOW DID [country] REACT TO [event]?	Terms in [country] and [event] to be treated as synonyms, and the resulting sets to occur within a term window of size 100	*Libya* today spurned the world community's offer….*bombing of the airliner*….

Table 5 : Template-specific techniques.

Co-reference resolution (corefRes)

Co-reference resolution is the process of identifying multiple representations of a given entity (usually a name). In newswire and broadcast news is it common to replace the entities referred earlier by either pronouns or shortened forms of the noun phrases. Co-reference resolution can be very useful in improving term statistics, and in matching patterns of text. For the query 'FIND STATEMENTS MADE BY [*Alexander Downer*]', answers could be of the form

- *He* said…
- *Alexander Downer* said…
- *Alexander* said…
- *Foreign Minister Downer* said…

Accurate co-reference analysis can help avoid missing potential answers even if the individual's name is referred to in many different ways. We used co-reference analysis only for person names.

Relevance Models (RM)

Relevance modeling (Lavrenko and Croft, 2001) builds a language model (probability distribution) of the vocabulary that is likely to occur in relevant documents. It does so by looking for the probability that words co-occur with query terms throughout the corpus.

Although the formal framework is elaborate and quite powerful, the implementation boils down to a variation on typical pseudo-relevance feedback. That is, the initial query is used to rank documents and the top several documents are assumed to be relevant. The vocabulary of those documents is analyzed to calculate a probability distribution of words that are related to the query—because the words occur in high-ranking documents. The resulting probability distribution is used as an additional component of the query with expanded vocabulary. Relevance models consistently improve retrieval performance over simpler language modeling approaches, and meet or beat other techniques based on automatic query expansion. We used the top 25 documents for feedback, and added 25 terms to the original query.

Let us assume again that the initial query is *Australian Foreign Minister*. Usage of the relevance model to expand this query resulted in the Indri query shown in Figure 7. The original query was assigned a weight of 0.8 and the rest a weight of 0.2. We can notice that almost all the terms added automatically are related to the original query terms.

```
#weight (
0.8     #combine (australian foreign minister)
 0.2     #weight( 0.0001 "australia"  0.0001 "minister"  0.0000768471477344 "foreign"  0.00003
"trade"  0.00003 "country"  0.00003 "two"  0.0000313826237483 "asean"  0.00003 "new"  0.00002
"state"  0.00002 "meet"  0.00002 "cooperate"  0.00002 "vietnam"  0.00002 "issue"  0.00002
"indonesia"  0.00001 "agree"  0.00001 "china"  0.00001 "singapore"  0.00001"relate"  0.00001
"free"  )
```

Figure 7 : Indri query resulting from use of a relevance model

Results

The results of applying each of the IR techniques presented in the previous section are provided in Table 6. One technique that worked across all templates was selective weighting of related terms. However, all the other techniques influenced different templates in a dissimilar manner. For example, relevance models (RM) are clearly better than the baseline approach for template 3, while they are unsuited for template 7.

MAP On Training Queries		Template						
		1	2	3	4	5	6	7
Technique	BOW	0.540	0.471	0.241	0.211	0.368	0.247	0.469
	weightREL	0.547	0.488	0.288	0.326	0.413	0.271	0.315
	Phrases	*0.547*	0.468	0.343	0.344	0.399	0.269	0.379
	DM	0.549	0.474	0.418	0.350	0.380	0.274	0.359
	Annotations	*0.549*	*0.474*	0.425	0.378	0.385	0.280	0.373
	corefRes	*0.549*	*0.474*	*0.425*	*0.378*	0.367	*0.280*	*0.373*
	tempSpecific	*0.549*	*0.474*	*0.425*	0.412	0.379	0.285	*0.373*
	RM	0.534	0.464	0.469	0.400	0.392	0.256	0.292

Table 6 : The mean average precision (MAP) for the seven templates as techniques are progressively applied to the query. The cells with italicized values indicate that the corresponding technique was not applied to that particular template. A higher value of MAP indicates better performance.

Table 7 reports the precision @10 (p@10) values for the various techniques on all templates. We can observe a general trend of improved p@10 scores as we progressively apply different techniques.

p@10 On Training Queries		Template						
		1	2	3	4	5	6	7
Technique	BOW	0.800	0.744	0.371	0.286	0.700	0.200	0.617
	weightREL	0.800	0.656	0.443	0.321	0.767	0.195	0.600
	Phrases	0.800	0.633	0.500	0.329	0.733	0.215	0.550
	DM	0.813	0.633	0.571	0.357	0.700	0.240	0.517
	Annotations	*0.813*	0.678	0.600	0.407	0.717	0.245	0.583
	corefRes	*0.813*	*0.678*	*0.600*	0.457	0.700	*0.245*	*0.583*
	tempSpecific	*0.813*	*0.678*	*0.600*	0.471	0.717	0.270	*0.583*
	RM	0.738	0.678	0.671	0.450	0.717	0.180	0.567

Table 7 : The p@10 values for the ten templates as techniques are progressively applied to the query. The cells with italicized values indicate that the corresponding technique was not applied to that particular template. A p@10 value of 0.8 means that 8 out of the top 10 documents in the ranked list were relevant

We now follow-up with a discussion on a per-template basis, and determine which techniques to include in the final version of the query processor. Our simple approach was to include only those techniques that improved performance over the previous technique in the order presented in Table 7.

Template 1 (List facts about events described as follows: [])

This template comes closest to free-form querying. The parameter that is specified is very similar to what a user would input in a text box if searching in the usual way. Such a query, while not lending itself to techniques like Phrases, use of annotations or even template specific modification, would be expected to gain from dependence modeling (DM) and relevance modeling (RM). Going by the simple numerical improvement due to the user of dependence models, we decide to include that technique in our final query processor.

Template 2 (Produce a biography of [person])

With the exception of related term weighting, none of the other techniques improved over the baseline.

Template 3 (Provide information on [organization])

The only techniques that did not work were those that weren't applied in the first place. We could perform no template-specific processing, and did not have the annotations for co-reference resolution. Thus, every technique was included in the query processor for this template as all of them led to progressive improvements.

Template 4 (Find statements made by or attributed to [person] on [topics])

We intuited that template-specific processing would be most suited for information needs such as those conveyed by this template. This was confirmed by the improvement from 0.405 to 0.457 when template specific processing was included.

Template 5 (Describe the prosecution of [person] for [crime])

Dependence models and use of phrases did not work for this template. We believe the reason could be because use of these techniques biases the query towards either retrieving documents about the person in general, or about the crime only. The use of template-specific processing helped in this template too.

Template 6 (How did [country] react to [event])

The use of phrases provided the greatest improvements in performance in this template, while relevance modeling hurt the most.

Template 7 (Identify persons associated with [organization] who have been indicted…)

In a surprising result, the selective weighting of related terms had a negative impact on performance. Closer inspection revealed a single outlier in the training data - a query with two relevant documents. The bag-of-words baseline approach ranked these two documents at the very top. However, the selective weighting of related terms caused both documents to be ranked lower in the ranked list causing a drop in MAP from 1.000 to 0.079. Discounting this effect due to the outlier, we choose dependence modeling and use of annotations for inclusion in the final query processor.

The knowledge we gained about the utility of the various techniques for each template enabled us to follow a greedy approach in designing a query processor for the test queries. Table 8 gives an overview of the techniques selected for each template.

Training Queries		Template						
		1	2	3	4	5	6	7
Technique	BOW	X	X	X	X	X	X	X
	weightREL	X	X	X	X	X	X	X
	Phrases			X	X			X
	DM	X	X	X	X		X	
	Annotations			X	X	X	X	X
	corefRes							
	tempSpecific				X	X	X	
	RM			X		X		

Table 8 : Summary of techniques incorporated into the final system. A 'X' in a cell indicates that the technique listed in the corresponding row was used for the template listed in the corresponding column.

Table 9 shows the results of using the final query processor on the test queries. The baseline bag-of-words approach does very well on six of the seven templates. This implies that the original queries contained focused instantiations of the templates and good related terms, and targeted topics that were easy to distinguish from others in the corpora. Improving performance further for these templated queries was inherently difficult.

MAP of Test Queries	Template						
	1	2	3	4	5	6	7
BOW	0.479	0.902	0.779	0.587	0.844	0.705	0.245
Final	**0.574**	0.844 ↓	0.606 ↓	**0.738**	0.858	0.720	**0.552**

Table 9 : The results of using the final query processor on the test queries. Improvements that are statistically significant are shown in **bold**. Statistical significance was measured using the paired t-test, with $\alpha = 0.05$

The performance of the final query processor with respect to MAP was a mixed success. While significant improvements were obtained in three templates, and numerical improvements in two more, in two templates performance deteriorated (indicated by a ↓). Table 10 reports the corresponding p@10 values. The results are similar to the MAP values.

p@10 of Test Queries	Template						
	1	2	3	4	5	6	7
BOW	0.600	0.500	0.525	0.350	0.475	0.380	0.325
Final	0.725	0.420↓	0.425↓	0.400	0.475	0.460	0.650

Table 10 : p@10 values for the test queries

Conclusions

Simply extending techniques that are known to work for ad hoc retrieval has limited utility for templated queries. A different set of techniques worked for each template. The reasons for failure of some well-known IR techniques can be attributed to the fact that the information needs in the case of templated queries are more specific. Consider the case of the template for finding *statements*. While relevance modeling might prove to be a good technique for retrieving documents related to the topic prevalent in the top *n* documents, we believe that it is incapable of focusing on parts of documents (for example, statements) to identify expansion terms. We believe that modifications of relevance modeling that work at the passage level are a potential solution. Working at the passage level could possibly help focus on the *type* of information being queried for.

Results from the training queries did not carry over in a similar fashion to the test queries. This indicates that a simple greedy approach to determining successful techniques in not sufficient. In addition to the interactions between individual techniques being complex, the queries themselves vary widely in content and quality. For example, a query with good related terms or accurate co-reference resolution might score better than others in the same template lacking those properties.

Techniques like the use of annotations and dependence models worked for a majority of templates. This shows that text annotations and structured query languages are a powerful combination with wider applications in information retrieval.

Acknowledgements

This work was supported in part by the Center for Intelligent Information Retrieval and in part by the Defense Advanced Research Projects Agency (DARPA) under contract number HR0011-06-C-0023. Any opinions, findings and conclusions or recommendations expressed in this material are the authors and do not necessarily reflect those of the sponsor. We thank Ao Feng at University of Massachusetts Amherst for generating data on the distribution of relevant document source dates for queries specifying activity dates. We also thank the SRI team in the GALE program for providing Machine Translations, Automatic Content Extraction, cross-document co-referencing and Automatic Speech Recognition data.

References

Croft, W. B., Turtle, H. R., and Lewis, D. D. (1991). The use of phrases and structured queries in information retrieval. In *Proceedings of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, (pp. 32--45)

Fagan, J. L. (1987) Automatic phrase indexing for document retrieval: An examination of syntactic and non-syntactic methods. In *Proceedings of the 10th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, (pp. 91--101)

Hoa Trang Dang, Jimmy Lin and Diane Kelly (2006) Overview of the TREC 2006 Question Answering Track. In *Proceedings of the Fifteenth Text REtrieval Conference*, (pp. 62--68)

Jianfeng Gao and Jian-Yun Nie and Guangyuan Wu and Guihong Cao (2004) Dependence language model for information retrieval: *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, (pp. 170--177)

Lavrenko, V. and Croft W.B., (2001) Relevance Based Language Models, In *Proceedings of the 23rd Annual International. ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR 2001), (pp. 120--127)

Metzler, D. and Croft, W.B., (2005) A Markov Random Field Model for Term Dependencies, In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR 2005), (pp. 472--479)

Metzler, D. and Croft, W.B., (2004) Combining the language model and inference network approaches to retrieval. *Information Processing and Management*, 40(5) (pp.735--750)

Ravichandran D. and Hovy E. (2001) Learning surface text patterns for a Question Answering system, In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, (pp. 41--47)

Spink, A., Bateman, J., and Jansen, B. J. (1999). Searching the Web: Survey of EXCITE users. *Internet Research: Electronic Networking Applications and Policy*, 9(2), 117--128

Turtle H. and Croft, W.B., (1991) Evaluation of an inference network-based retrieval model. *ACM Transactions on Information Systems*, 9(3), 187--222

Appendix

Tables 11, 12 and 13 summarize the entity types and sub-types that were available as annotations in the data.

Type	Sub-type
FAC (facility)	Airport, Building-Grounds, Path, Plant, Subarea-Facility
GPE (Geo-Political Entity)	Continent, County-or-District, GPE-Cluster, Nation, Population-Center, Special, State-or-Province
LOC (Location)	Address, Boundary, Celestial, Land-Region-Natural, Region-General, Region-International, Water-Body
ORG (Organization)	Commercial, Educational, Entertainment, Government, Media, Medical-Science, Non-Governmental, Religious, Sports
PER (Person)	Group, Indeterminate, Individual
VEH (Vehicle)	Air, Land, Subarea-Vehicle, Underspecified, Water
WEA (Weapon)	Biological, Blunt, Chemical, Exploding, Nuclear, Projectile, Sharp, Shooting, Underspecified

Table 11 : Annotated Entity Types and Sub-types

Type	Sub-type
ART (artifact)	User-Owner-Inventor-Manufacturer
GEN-AFF (Gen-affiliation)	Citizen-Resident-Religion-Ethnicity, Org-Location
METONYMY	None
ORG-AFF (Org-affiliation)	Employment, Founder, Ownership, Student-Alum, Sports-Affiliation, Investor-Shareholder, Membership
PART-WHOLE (part-whole)	Artifact, Geographical, Subsidiary
PER-SOC (person-social)	Business, Family, Lasting-Personal
PHYS (physical)	Located, Near

Table 12 : Annotated Relation Types and Sub-types

Type	Sub-type
Life	Be-Born, Marry, Divorce, Injure, Die
Movement	Transport
Transaction	Transfer-Ownership, Transfer-Money
Business	Start-Org, Merge-Org, Declare-Bankruptcy, End-Org
Conflict	Attack, Demonstrate
Contact	Meet, Phone-Write
Personnel	Start-Position, End-Position, Nominate, Elect
Justice	Arrest-Jail, Release-Parole, Trial-Hearing, Charge-Indict, Sue, Convict, Sentence, Fine, Execute, Extradite, Acquit, Appeal, Pardon

Table 13 : Annotated Event Types and Subtypes