Combining Information Retrieval with Information Extraction for Efficient Retrieval of Calls for Papers

Fotis Lazarinis

Department of Computing Science, University of Glasgow
Glasgow, Scotland

Abstract

In many domains there are specific attributes in documents that carry more weight than the general words in the document. This paper proposes the use of information extraction techniques in order to identify these attributes for the domain of calls for papers. The utilisation of attributes into queries imposes new requirements on the retrieval method of conventional information retrieval systems. A new model for estimating the relevance of documents to user requests is also presented. The effectiveness of this model and the benefits of integrating information extraction with information retrieval are shown by comparing our system with a typical information retrieval system. The results show a precision increase of between 45% and 60% of all recall points.

1 Introduction

Information retrieval (IR) systems, also called text retrieval systems, facilitate users to retrieve information which is relevant or close to their information needs.

Even though specific words may be key attributes of a domain, conventional IR systems process them as ordinary terms using general statistical methods [18, 26]. Usually such terms appear several times in a document collection and so they lose their power to discriminate among documents. However, a term may appear several times in a document collection but with different significance each time. In calls for papers (CFPs), for example, there exist some past dates along with the conference's date. When users pose queries about conferences held in a specific month all the calls for papers where the specific month name appears are retrieved even though most of them are irrelevant. Another problem of the conventional approach is caused by the fact that in collections about specific subjects, synonyms and/or abbreviations are often encountered. Traditional IR systems treat the variations of a term as different terms. Stemming algorithms [10] attempt to partly solve the problem with variations but they cannot effectively cope with synonyms and abbreviations. This affects the retrieval and requires either the integration of a thesaurus [22] or users to specify all the alternative forms in their query if they wish to retrieve all the relevant documents. As we will see in section 3 this is not a problem in our system because we implicitly use a thesaurus for the important terms of our domain. The last problem of typical text retrieval systems is that they consider two terms to be equally important if they exist the same times in a document or in a document set. For example, imagine a document collection of medical case records. Certain disease names will be treated equally in the retrieval with other words that are not important simply because their frequencies of occurrence are equal.

These problems seriously affect the retrieval in collections about specific subjects such as medical cases or financial news where the important terms are encountered several times. The solution to the above problems proposed in this paper is to employ information extraction (IE) [6] techniques in order to identify the useful information that would lose its significance if it was processed by a standard text retrieval system. Since IE is a highly domain-dependent task we concentrate on calls for conference papers and our aim is to automatically identify conferences' date and location. In calls for papers there exist many other locations and dates in addition to the conference's date and location. With a typical IR system when a user wishes to retrieve meeting announcements held in a specific place or in a specific date all the CFPs that contain the user specified data will be retrieved since the IR system cannot distinguish among the appearing dates and locations. As soon as the attributes, i.e. location and date, have been identified the rest of a CFP's content is processed using standard IR techniques.

After this processing, documents are represented by a set of keywords (index terms) describing the subject of a document and a set of solid attributes, i.e. date and location in our system. The most important issue arising in this case concerns the computation of the similarity between documents and queries. In typical information retrieval systems the similarity between documents and queries is based either on the probabilistic model [26] or on the vector space model [19]. With the incorporation of attributes the direct use of these models is not suitable, at least for queries based partly on attributes. A general model for estimating the relevance between queries based on

attributes and documents and a model for mixing the results of queries based on both content (index terms) and attributes are presented and analysed later on the paper.

The rest of the paper explains the algorithms employed in CIERS (Combined Information Extraction and Retrieval System). CIERS is an information retrieval system that combines the strengths of IR and IE. Figure 1 shows a prototype user interface in Java for CIERS (http://www.dcs.gla.ac.uk/~lazarinf/project.html).

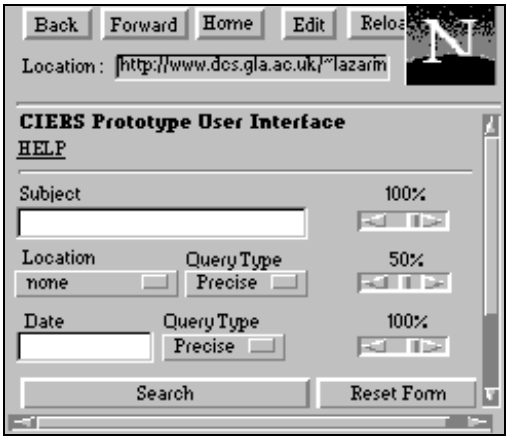


Figure 1: CIERS Prototype User Interface

The standard techniques employed in IR engines and thus in the standard IR module of CIERS are described briefly in the next section. Section 3 presents information extraction and the rules that achieve the automatic identification of location and date from meeting announcements. Section 4 describes the model on which the similarity computation between documents and queries is based. Finally, we present the results of experiments that show that information extraction techniques can benefit information retrieval.

2 Information Retrieval Engines

The basic operations of typical information retrieval systems can be grouped into two main categories: indexing and matching (or retrieval). The purpose of the indexing process is to identify the most descriptive words existing in a text. After the elimination of the stopwords and the identification of the unique stems of the remaining words, the term frequency (tf) and the inverse document frequency (idf) of each unique stem are calculated. Each document is then described by a set of keywords along with their tf and idf [9].

The aim of the query matching process is to derive a list of documents ranked in decreasing order of relevance to a given query. When a query based on content (expressed in natural language) is submitted to the system, it undergoes a process similar to the indexing process. Now both documents and queries can be represented as weighted vectors where each term's weight is usually a combination of tf and idf. In this case the similarity between documents and queries is based on the vector space model [19]. In this model documents and queries are viewed as n-dimensional vectors, where n corresponds to the number of unique index terms. The similarity between query q and document d is computed by measuring the cosine of the angle between their vectors (figure 2).

$$\text{Similarity}(q,d) = \frac{\sum_{i=1}^{n} w_{iq} * w_{id}}{\sqrt{\sum_{i=1}^{n} w_{iq}^2 * \sum_{i=1}^{n} w_{id}^2}}$$

where

w_{iq}, the weight of query term i

w_{id}, the weight of document term i

n, the total number of terms

Figure 2: Cosine similarity measure

3 Information Extraction

A relatively new and increasingly important area in text processing is information extraction [6]. Information extraction aims at identifying special kind of data from domain-specific document collections. IE systems process documents trying to identify pre-defined entities and the relationships between them, filling a structured template with the extracted information. Hence, an IE system can be considered as converting some elements of unstructured text documents into structured database entries.

3.1 Related Work

Information extraction systems have been employed in the summarisation of medical case records by extracting diagnoses, test results, and treatments [17]. Postma et al. [14], and Chowdhury and Lynch [2] have used chemistry papers to extract data such as names, and scientific terms. In general technical reports are perfect candidates for information extraction because they contain data that have standard forms, e.g. references. Business IE systems extract details about companies, products, and services and other details of interest to businesspersons, e.g. in the message understanding domain (MUC) [23].

These examples are standalone IE systems and cover only special cases of text processing. Although information retrieval and information extraction are complementary there has been little work aimed at integrating the two areas. The most notable work is that of Gaizauskas and Robertson [11]. In their work they used the output of Excite [7] as input to an IE system, called VIE (Vanilla IE System). Their domain was management succession events and their scenario was designed to track changes in company management. The results of Excite searches were passed to VIE which produced a template filled with the company's name, the old manager's name, the new manager's name, etc. A natural language summary was also produced for the retrieved documents by populating the empty fields of a fixed-structure summary.

Since the purpose of Gaizauskas and Robertson was to create a system that would construct a structured data resource from free text they evaluated only the success of the information extraction procedure. Whereas we also evaluate the extraction procedure in order to measure its effectiveness we are more interested in the performance of the combined system. Our goal is to improve the efficiency of conventional IR systems, at least in some special cases. Therefore, we evaluate CIERS using the standard IR method and we compare it with a typical text retrieval system. Before we report the results of the evaluation of the combined system we need to explain the location and date extraction procedure and the model on which CIERS is based.

3.2 Extraction of Attributes

As many other applications of natural language processing information extraction systems rely on domain-specific dictionaries to extract specific kind of information from free text [3]. Such dictionaries contain lexical items that enable the recognition of the desired entities. For instance, if we are interested in English full names the dictionary must consist of all the first English names.

These domain-specific dictionaries contain only the minimal necessary information for the extraction procedure. When an item in the text is matched with an item in the dictionary a rule is activated which enables the extraction of the desired information. To continue the last example, when a word is found to be a proper first name then a simple rule like "the next word is a potential surname" may be activated.

Unfortunately, simple rules rarely have high success rates and complex rules (or heuristics) are often needed. This need arises from the fact that the same kind of data exhibit considerable variation in both the information they carry and in the way they are presented. For instance, the name of a person may appear in several different forms, e.g. "Peter Smith", or "Peter M. Smith", or "Smith Peter". In addition, the information may be scattered across several sentences. Finally, several instances of the same type of information may appear in the same text, e.g. many names may exist in a text in addition to the one of interest.

In order to construct rules that will enable the successful extraction of the desired facts, one has to examine thoroughly a representative sample of documents of her/his domain. This will also allow the accurate construction of the dictionary. As already mentioned, our aim was to automatically extract conferences' location and date from meeting announcements. Therefore, we examined 250 CFPs, a small part of our document collection consisting of 1927 meeting announcements[1]. This analysis allowed us to realise the different patterns of date and location and

[1] The CFPs collection can be found at http://www.dcs.gla.ac.uk/~lazarinf/CFPcoll.html

construct the extraction rules. The following two sections analyse briefly the location and date extraction procedure[2] respectively and section 3.2.3 presents the results of the evaluation of the IE module.

3.2.1 Location Extraction Procedure

In almost every of the 250 CFPs we examined, the conference's city and country was named while the continent was cited only in few announcements. Time limitations prevented the identification of the city because the required set of rules would be rather complicated and hence it was decided to detect only the country of each conference. Nevertheless, when a country of a conference is extracted it can be easily connected to its continent as we will see below. The second conclusion reached was that more than 50% of the conferences were held in USA. Almost all of these CFPs mentioned the state of the conference. Therefore, in order to offer users a wider choice of queries it was decided to extract the state name as well for conferences held in USA. Hence, CIERS identifies US states and countries for the rest of the world. In other words US states are treated as ordinary countries and USA as a continent.

In order to recognise country and state names the dictionary should contain all the formal country names, e.g. "Greece", and state names such as "California". Additionally, all the variations of a country's name, e.g. "Hellas" for "Greece", and all the state codes, e.g. "CA" for "California", should be incorporated into the dictionary because they are used very frequently to indicate a conference's location.

As previously explained when a word of a document is matched with a dictionary entry a rule is activated. However, this cannot work with country names consisting of more than one words because it is impossible to automatically decide which words probably constitute a country and search them in the dictionary as one text element. As a result of this, apart from the proper country names and their variations the dictionary contains the rarer of the words making a country name (the rarer word is used to minimise the activation of rules), e.g. "Kingdom" for "United Kingdom".

In order to associate the location dictionary entries we add an attribute, named country type, to the dictionary. If an entry is a full country name then the country type's value is the country's continent. If it is a variation or a state code then the country type points to the proper country or state name. Finally, if an entry is a part of a country's name then the country type shows how many words before or after this part are needed in order to constitute a proper name.

The last conclusion reached from the examination of the 250 CFPs was that although several countries or US states may be mentioned in a CFP, in nearly all the cases the state or country that appears first is the conference's location.

So, if the matching term is a formal country name then the identification procedure ends successfully. If it is a variation or a state code is mapped to the formal country name and again the conference's country has been extracted. When the processed term is part of the name of a country the necessary previous or next words are taken and the new potential country name is searched in the dictionary. If it is found in the dictionary then the country name has been identified; otherwise the extraction procedure ends. Finally, if a proper country name has been detected it is connected to its continent via its country type value.

The above set of rules is only a subset of the actual rules employed in the implementation of CIERS. Space limitations prohibit us from explaining the rest of the rules that cover special cases such as collisions of state codes with stopwords, e.g. "IN" is both a stopword and the state code for "INDIANA". Even in that case the above description verifies that an IE system cannot be used in any document collection but only in some specific domain because the rules depend on the characteristics of the collection.

3.2.2 Date Extraction Procedure

Again the first step in the identification of a conference's date is the construction of a suitable dictionary containing the necessary terms that will activate the rules for the extraction procedure. The date dictionary is less populated than the location dictionary as it must contain only the 12 full month names and their 11 abbreviations ("May" is both the month's full name and the contraction).

The second observation made after the analysis of the 250 CFPs is that the latest date existing in a call for papers is usually the conference's date. Unfortunately in a very small percentage of calls for papers, some future dates announcing future meetings appear. But this problem does not significantly affect the identification procedure as it typically leads to a minor error (table 1).

[2] For a full description of the extraction procedure please consult [12].

Whenever a month is found in the text, CIERS first checks the succeeding words until the end of the sentence and then the preceding words until the beginning of the sentence. This search aims at identifying the day and the year of the conference. If a number from 1 to 31 or a word that starts with a number from 1 to 31 and ends in “st”, “nd”, “rd”, “th”, e.g. 1st, 2nd, 3rd, 4th, is found then this is the day of the conference. If a four-digit number is found which starts either with 19 or 20 then it is the desired year. As soon as a date is identified it is compared with the previous extracted one, if any, and the latest one is kept.

The description of the rules employed in the extraction of dates (again we omitted the description of the specialised rules that handle month names such as “may” which is both a month and an English modal verb) and locations leads us to some important conclusions. First, the context surrounding the activation terms is really important and is this that actually allows a rule to succeed. Second, the rules are complicated even if the desired information is simple and its alternative forms are limited. Moreover, the extraction of knowledge can only be based on heuristics because they depend entirely on the characteristics of the extracted entities and the context in which it appears. Finally, the dictionaries that contain the activation terms can be used as thesauri and can be utilised in queries. For example, by consulting the dictionaries user queries can be expanded to include all the variations of a term, thus retrieving more relevant documents.

3.2.3 Evaluation of the IE Module

Although regularly used in the evaluation of IR systems, the performance of an IE system can be measured using *Precision (P)* and *Recall (R)* [15, 16]. Precision measures the ratio of the correctly extracted information against all the extracted information. Recall measures the ratio of the correct information extracted from the texts against all the available information.

The effectiveness of the extraction procedure was tested with the entire collection consisting of 1927 calls for papers. These CFPs were gathered from the Internet from various archives and contain announcements for conferences mainly covering various fields of computer science. Also a significant portion of this collection is made up of psychology, engineering, and physics conference announcements.

Despite the diversity of the collection the system works extremely well and the employed rules achieve high rates of precision and recall. The results are summarised in the next table.

	Correctly extracted	**Erroneously extracted**	**Not extracted**	**Precision**	**Recall**
Country	1796	112	19	94.12%	93.20%
Date	1881	41	5	97.86%	97.61%

Table 1: Precision and Recall for the attribute extraction procedure

This high accuracy will eventually result in improved performance of the combined system over a typical IR system where queries based on attributes are expressed as ordinary queries based on content.

Before moving on to the next section two remarks should be made about the occasional failure of CIERS to detect date and location. Sometimes the system fails because the date and location do not appear in a call for papers. Whereas the system is not responsible for this failure, in the evaluation we accounted it to CIERS and thus we got slightly worse precision and recall values. Furthermore, a failure analysis should have followed the testing of the IE module. This analysis would have helped us to realise the possible sources of error and modify the set of the employed rules. However, time limitations prevented the analysis of the erroneous instances in both cases.

4 Combined System

CIERS supports two types of queries: attribute and content queries. Users are able to ask either individual content or attribute queries or any combination of them. The first issue arising is how a single estimate of relevance is derived in any combination. A possible solution is to use data fusion techniques [8, 24, 27] and to merge the ranked lists for the different types of queries. In the next section we define a model for merging the results of the different types of queries. The subsequent issue is how a list of relevant documents in attribute queries is obtained. A model for computing the relevance or closeness of documents to attribute queries is proposed below.

4.1 Merging of Results

Data fusion is a technique used for combining the results of different retrieval strategies into one unified output. In traditional IR systems data fusion is used for improving performance by allowing each strategy's relevance estimate to contribute to the final result. The combined list is typically more accurate than any of the individual results.

Fusion of results takes two different forms. Data fusion aims at merging the results of different strategies for a given query on a single document set. In its second form, known as collection fusion [27], the goal is to combine the retrieval results from multiple, independent collections into a single result such that the performance of the combined system will exceed the performance of searching the entire collection as a single collection.

Our work is a combination of the two different forms. CIERS uses three different data sets, i.e. index terms, date, and location attributes. Also different strategies are used for each query type. Before we proceed in the explanation of how the combination of the output for the different kinds of queries is achieved, it is worth mentioning some work done in the area of data fusion.

A number of different methods for combining the results of different strategy implementations of the same query have been proposed. Fox et al. [8] determine documents' overall relevance by adopting the maximum score for each document of all the strategy outputs. Thompson [24] combines the results of the different methods based on the performance level of each method. That is, each method is evaluated independently and its performance level is measured. Then the methods are combined in proportion to their performance level. Both of these approaches are acceptable in a single query and a single set of data because the aim is to enable the best strategy to affect most the retrieval. Nevertheless, in our case this is not desirable as it will result in retrieval biased against one query type.

Similarly to the approaches described above we use a linear combination of the output lists. That is, the overall relevance estimate is the sum of scaled estimates of the individual queries (figure 3).

$$S(q,d) = \Theta_1 S_1(q_1,d) + \Theta_2 S_2(q_2,d) + \Theta_3 S_3(q_3,d)$$

Figure 3: Parameterised mixture of the individual relevance estimates

$S(q,d)$ is the combined estimate for the combined query q and document d. $S_i(q_i,d)$ [3] is the similarity between the subpart q_i of the original query and document d[4]. Θ_i are free parameters in the model set by users, granting them with full flexibility over the retrieval (as shown in figure 1). Θ_i is the scale of the query subpart q_i; Θ_1 is the scale of the query about subject, Θ_2 of the query about the location[5], and Θ_3 of the query about the date. So, if only the subject and the date are of interest then Θ_1 and Θ_3 will be 1 (100%) and Θ_2 will be 0 (0%). That way, searchers get the combined estimate of only these two subqueries. Furthermore, the last equation allows users to specify their rate of interest in each subquery. For instance, if the subject and the date of conferences are essential and location is less important then users can define Θ_1 and Θ_3 to be 1 (100%) and Θ_2 to be 0.5 (50%) or less.

At this point it is necessary to underline that the results of the individual queries must be consistent so before the merging of the individual estimates each list is divided by its maximum score. That way the partial scores lie between 0 and 1 and in the merging of the results each counts as much as its scale, i.e. Θ_i, defines.

4.2 Semantics of Attribute Query Matching

As explained sometimes CIERS fails to identify a conference's date or location so the attribute values will be missing. This fact originates partially from the occasional failure of the system to identify the attributes and partially from the fact that in some conference announcements (usually preliminary) the date and/or location are not cited.

The missing values of attributes could be denoted with the special value "null". Null values have been used extensively in databases to denote that a value is missing [4, 5, 25]. However, indicating a missing value with null is not adequate because no distinction is made between missing and non-applicable attribute values.

[3] $S_1(q,d)$ is the estimate for the content query and is based on the Vector Space model (section 2).

[4] The equation of figure 3 can be extended if more attributes are incorporated by adding the scaled estimates of the strategies based on the new attributes.

[5] The location of a conference may be its country or its continent.

A more suitable solution for differentiating the two cases is to use two special values, Not-Known (NK) and Not-Applicable (NA) [13]. Not-known represents attribute values that the system fails to identify and not-applicable denotes attribute values that are not defined in a CFP. In addition to the problem of missing values another issue stems from the fact that usually conferences last more than one day. This means that the date attribute cannot actually be represented by a single value but by a range of values denoting the conference's duration. Morrissey [13] uses a special value named p-range. A p-range is denoted with a lower and upper limit, indicating the limits of the range of values. An example of p-range would be [19/7/97-25/7/97] meaning that a conference starts on 19 July 1997 and ends on 25 July 1997. The current implementation supports only the NK special value. The other special values are included in the model for future expandability, e.g. calls for journals where location is not applicable could be processed by the system. Below the system, the queries, and the similarity computation are formally described.

4.2.1 System

D: the finite set of all stored documents. An individual document is denoted by d_i.

A: the finite set of all attributes. a_i represents a specific attribute.

V: the set of all possible different value sets. A specific attribute value set is denoted by V_{a_i} and the value for a specific attribute is denoted by u_i.

F: the set of all functions that map documents to attribute values. For reasons of convenience a function is denoted as the attribute $a_i(d) = u_i$, e.g. country(d) = UK.

4.2.2 Queries

Since information retrieval aims at identifying those objects that are relevant or close to a user's needs our model supports two different types of attribute queries, namely *Precise* and *Close*.

In precise attribute queries users are interested in documents that definitely satisfy their needs. In close queries searchers are additionally interested in conferences that are close to their information need, so in their requests they specify the desired value for an attribute and the tolerance for the values of that attribute. For example, one may be quite interested in conferences held on 15 July 1997 but she/he may be also interested in conferences held a few days before or after the specified date. This type of attribute queries would be also useful in queries about countries. If the location dictionary is extended to include the concept of neighbouring countries the system will be able to retrieve conferences held close to the original place. Users simply have to specify how far from their original goal they are willing to deviate. The current implementation supports only the concept of continents for grouping countries in a geographic area. Consequently, all the CFPs of conferences held in countries of the same continent will be retrieved but ranked lower than conferences held in the user specified country.

Formally an attribute query is denoted as $Q_{u_i}^{a_i}$, which means that stored documents must have value u_i for the attribute a_i in order to satisfy the query. There are three sets of documents that match attribute queries; those that exactly (are known to the system to) match a query denoted as $K_{u_i}^{a_i}$, those that possibly satisfy a query denoted as $P_{u_i}^{a_i}$, and those that are close to the initial request represented as $C_{u_i,t}^{a_i}$. Below each set of relevant documents is defined formally.

$$K_{u_i}^{a_i} = \bigcup \{d : a_i(d) = u_i\}$$

$$P_{u_i}^{a_i} = \bigcup \{d : a_i(d) = \text{NK or } a_i(d) = \text{NA}\}$$

$$C_{u_i,t}^{a_i} = \bigcup \{d : a_i(d) = u'_i, \text{where } \text{dist}(u_i, u'_i) \leq t\}$$

where

$d \in D$, $a_i \in A_i$, u_i and $u'_i \in V_{a_i}$

t, specifies the limit of the range of values
a user considers close to her/his query

Figure 4: Formal definition of the sets satisfying attribute queries

u'_i is a value for the attribute a_i for which the distance (dist) between it and the query specified value is less or equal than the specified tolerance (t).

Precise attribute queries are satisfied by $K_{u_i}^{a_i} \bigcup P_{u_i}^{a_i}$ which means that those documents that have value u_i for the attribute a_i and those for which the value of the specified attribute is missing, (possibly) satisfy the query. For example the $K_{\text{July 1997}}^{\text{date}}$ contains documents like date(d1) = 15/7/1997 or date(d2) = [12/17/1997 – 18/7/1997] and the $P_{\text{July 1997}}^{\text{date}}$ is consisted of documents such as date(d3) = NK or date(d4) = NA.

The set of documents that satisfy close attribute queries is the union of all the three discrete sets described above. For instance, a user may be particularly interested in conferences held on 15 July 1998 but she/he would be interested in meetings that happen 15 days before or after that date. Examples of documents belonging to $C_{\text{15 July 1998, 15}}^{\text{date}}$ are: date(d1) = 12/7/1998 or date(d2) = 30/7/98 or date(d3) = [18/7/1998-19/7/1998].

In any of the previous cases, documents that definitely satisfy the query should be ranked first, those that are close (in close queries) should be ranked second, and last should be ranked those that possibly satisfy a query. The ranking of documents close to the initial request should depend on how close a document is to the original query. In the last example, CFPs for conferences held 5 days after the specified date should be ranked before conference announcements held 10 days later. Additionally, the ranking of documents close to a request should depend on the number of different possible close values. For example, if a user poses a query about country then close CFPs should be ranked higher if the close countries are 5 than when the close countries are 10 because in the second case the deviation from the initial request is greater.

4.2.3 Query Matching

Similarity between attribute queries and documents is computed by the equation of figure 5. This equation is based on entropy [20, 21]. Entropy calculates the uncertainty associated with the decision to retrieve an object that satisfies a query. The uncertainty is inversely proportional to the information the system has for an object. Since we are interested in finding how close a document is to a given query, entropy has been modified and our score is maximum when a document matches the attribute query and minimum when a document possibly matches it.

$$\text{Similarity}(q,d) = 1 - \frac{-\sum_{k=1}^{m} P_k * \log_2 P_k}{\log_2 n}$$

where

n, the number of all different values of the query specified attribute

m, the number of attribute values that match the query

P_k, the probability of a document to match the query

Figure 5: Similarity score based on entropy

The last equation depends on the set of documents that match a request and the number of attribute values that match it. In $K_{u_i}^{a_i}$ m is 1, since only one value for the specified attribute satisfies the query. In $C_{u_i,t}^{a_i}$ m is equal to the number of values that match a query or in other words to the distance between a close attribute value and the query specified one. For example, in a query about date, m is 5 for conferences held five days later than the desired date and 10 for conferences held ten days later. In the last set of documents, $P_{u_i}^{a_i}$, since a_i may have any of the n different values m equals n.

For all the three document sets P_k is the probability of a document to correspond to a query, i.e. to have such an attribute value that matches the query. While P_k varies for different attribute values and should depend on the error in the extraction of attributes, for simplifying its calculation it is assumed that every document of a particular set has the same chance to match a query and it is defined to be 1/m. Let us now illustrate the use of the last equation with an example from our document collection[6].

[6] The number of different countries in our location dictionary is 254.

Query: "List all calls for papers of conferences held in California".

For documents of $K^{country}_{California}$: $Similarity(q,d) = 1 - \frac{-\sum_{k=1}^{1} \frac{1}{1} * \log_2 \frac{1}{1}}{\log_2 254} = 1$

For close documents (51 close states): $Similarity(q,d) = 1 - \frac{-\sum_{k=1}^{51} \frac{1}{51} * \log_2 \frac{1}{51}}{\log_2 254} = 0.28$

For $P^{country}_{California}$: $Similarity(q,d) = 1 - \frac{-\sum_{k=1}^{254} \frac{1}{254} * \log_2 \frac{1}{254}}{\log_2 254} = 0$

The last example shows that the equation of figure 5 meets the requirements set in the previous section where the two different attribute query types were defined. Nevertheless, the assumption for the probability leads to equal treatment of the different attribute values. For example, conferences for which the system has no definite information, i.e. NK or NA, have the same chance to take place in "Greece" and in "Italy" . Clearly, this assumption is incorrect but time restrictions prevented us from analysing our collection and calculating the probability in each case.

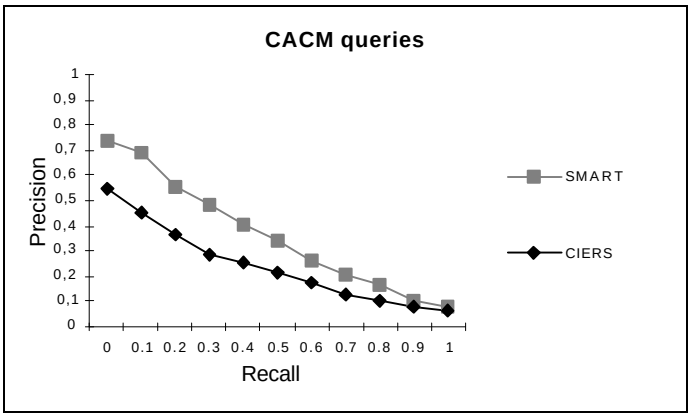
5 Evaluation

Our experiments were divided into three phases. First the system was tested with only content queries, then with only attribute queries, and finally with combined queries. That way it was possible to determine the performance of each query type and thus to realise the effect of employing information extraction techniques in IR systems. In all three cases, CIERS was tested against SMART [1], one of the most widely used experimental IR systems.

For our experiments we constructed our own document collection comprising of calls for papers because none of the existing experimental collections was suitable for CIERS as they did not fit the IE module. A set of realistic requests was provided by some members of the Computing Science Department of the University of Glasgow. Considering that it is a tedious and time consuming task we made the relevance judgements for these queries ourselves. However, this may lead to questionable estimates as it is difficult to decide if the subject of a conference matches a query. On the contrary, in queries about country, continent, and date the relevant documents can be easily determined as these data are easily accessible. Therefore, before the tests with our document collection, CIERS was tested with the CACM standard IR test collection in order to establish the baseline performance of the CIERS standard IR engine. In other words with this test we estimated the efficiency of CIERS against SMART in content queries.

5.1 CACM tests

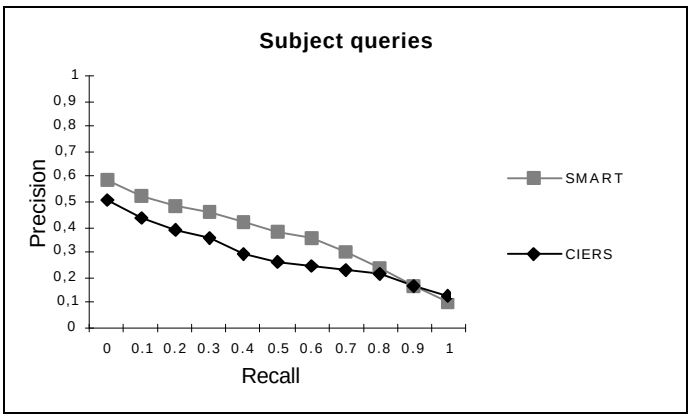
For the CACM tests 50 (content) queries were used and 5 different tests were run, each using a different combination of term weights for documents and queries. Due to space restrictions we present only the first test where the difference between the two systems was maximum.

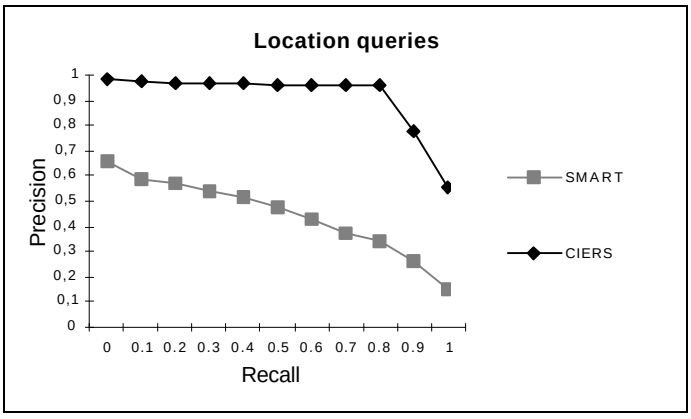


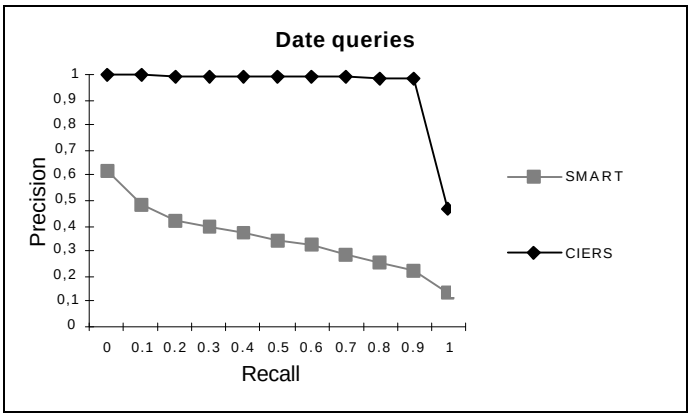
As expected in all the tests CIERS performed worse than SMART due to more advanced indexing, weighting, and matching schemes used in SMART. Their difference in performance lies between 8% and 12%.

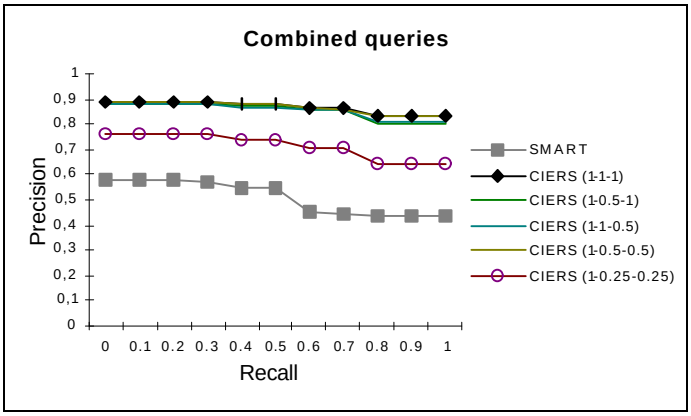
5.2 CFPs Collection tests

A set of 25 queries specifying the desired conference characteristics, i.e. subject, date, and location, were used for these experiments. The experiment with our document collection was divided into four stages. First we tested the system with queries concerning conferences' subject, then with queries about location and date, and finally with combined queries. In order to be fair to SMART in these experiments we used the weighting function of the first CACM test where SMART performed best.









Before explaining the significance of the last tests it is important to make some remarks. First, as it was seen in section 3 CIERS acts as a thesaurus. For example, users can retrieve CFPs of conferences held in "United Kingdom" even if their query is about "Great Britain". Again, to be fair to SMART all variations of a country's name were specified in queries about countries with more than one name. Moreover, location queries were concerned only with countries and not with continents since SMART would be unable to retrieve any of the matching documents. Additionally, only 12 distinct date queries were submitted each specifying one of the 12 months. This was necessary because SMART discards numbers in the indexing phase and it would not be able to retrieve the matching documents. Finally, all the queries were precise since CIER's notion of close queries is not supported by SMART.

Even though we tried to be fair to SMART, the last graphs show that the performance of CIERS is significantly improved over SMART in attribute queries. As anticipated, location queries performed slightly worse than queries about date since the error in the detection of location is greater than the error in the date extraction procedure. In both tests, the sudden drop in precision at the high recall points is because of the error in the extraction procedure.

The last test aimed at measuring the performance of CIERS against SMART with queries based on all the three different data sets, i.e. index terms, location, and date attributes. For the 25 combined queries posed to the system only those CFPs that met all the three conditions set in every query were considered relevant. While SMART was executing each request as one content query, CIERS was dividing them into three subparts. Each subpart was executed separately and the individual results were combined into one unified output. Five different combinations of the partial query subpart results were tested. In all these tests the objective was to examine the influence of the attribute queries in the combined retrieval. As it can be seen in the last graph the performance of the system is extremely high compared to the performance of SMART even when the attribute queries count only 25% each (CIERS 1-0.25-0.25). This means that the increase in precision is vital even when the retrieval is based primarily on the content and not on the attributes.

6 Conclusions and Further Work

The goal of our work was to improve the response of information retrieval systems by identifying and utilising in queries the key attributes of documents. This was achieved by employing information extraction techniques. Even when the extracted information is simple the extraction rules are complicated and depend on the context surrounding these entities. Nevertheless, the benefits gained are important. First the increase in precision is substantial and lies between 45% and 60% in attribute queries. At this point it must be underlined that the average precision of SMART would be lower if only one variation of a country's name and full dates were used. The increase in precision in combined queries is significant as well, even when the attribute subqueries count only 25% each. The significance of this increase is further realised if we take into account the performance level of the standard IR module of CIERS. The second advantage of CIERS is that it acts as a thesaurus. As we have seen the same information may be expressed with several alternative ways but users have to define only one variation in their requests. The last benefit of CIERS is that it supports a wider range of query expression. For example, it is possible to retrieve conference announcements for conferences held in a specific continent or in a specific date. But, as explained, for a fair comparison these features of CIERS where not used when comparing performance with SMART.

Although CIERS embodies a number of novel features there are several ways to improve its functionality and investigate its effectiveness. First, the standard IR module of CIERS should be improved by employing advanced IR techniques, such as the relevance feedback technique [18,26]. Furthermore, the erroneous instances in the extraction procedure should be analysed and the extraction procedure should be modified accordingly. Also our work should be applied in other domains to explore the difficulty of porting an IE system and its performance thereafter.

In general, our work can be considered as initial experimentation towards the integration of IR and IE. There are still several open research issues in creating an integrated IR-IE system. For example, the usability of information extraction into text retrieval should be investigated with more complex pieces of data. That way it would be possible to realise the effects of IE in IR when the success in the extraction procedure would not be as high as in our work. Moreover, as mentioned in section 3, IE systems convert unstructured text elements to structured database entries. It would be rather interesting to investigate the impact of integrating a database system to a combined IR-IE system. The database system would provide more efficient storing mechanisms and enhanced modelling capabilities. The model for estimating the relevance of documents to attribute queries could be embedded in the database system. This would also allow the automatic computation of the prior probability P_k because the database system could compute it for each attribute value by simply analysing the stored objects.

To sum up we can say that information extraction can benefit information retrieval, especially when the success in the identification process is high. However, improvements are required and expected in both fields before their integration provides a powerful tool for text retrieval.

7 Acknowledgements

I would like to thank my supervisor Dr. Mark Dunlop for his valuable guidance and support throughout this project. Also I would like to acknowledge the help of the IR group of the Computing Science Department of the University of Glasgow.

8 References

1. Buckley C. Implementation of the SMART Information Retrieval System. Technical Report 85-686, Department of Computer Science, Cornell University, Ithaca, 1985

2. Chowdhury G G, Lynch M F. Automatic Interpretation of the Texts of Chemical Patent Abstracts. Journal of Chemical Information and Computer Science 1992; 32: 463-467

3. Church K. W., Rau L. Commercial Applications of Natural Language Processing. Communications of the ACM 1995; 38: 71-79

4. Codd E F. Extending the Database Relational Model to Capture more Meaning. ACM TODS 1979; 4: 397-434

5. Codd E F. Understanding Relations. ACM SIGMOD 1975; 7: 23-28

6. Cowie J, Lehnert W. Information Extraction. Communications of the ACM 1996; 39: 80-91

7. Excite. Keep It Simple, Searchers. WebWeek Magazine 1997, 7

8. Fox A E, Koushik M P, Shaw J, Modlin R, Rao D. Combining Evidence from Multiple Searches. TREC-1 Conference Proceedings, National Institute for Standards and Technology Special Publication 500-207, 1993, pp 319-328

9. Fox C. Lexical Analysis and Stoplists. In: Frakes W B, Baeza-Yates R (eds) Information Retrieval: Data Structures and Algorithms. Prentice-Hall, New Jersey, 1992, pp 102-130

10. Frakes W B. Stemming Algorithms. In: Frakes W B, Baeza-Yates R (eds) Information Retrieval: Data Structures and Algorithms. Prentice-Hall, New Jersey, 1992, pp 131-160

11. Gaizauskas R, Robertson A. Coupling Information Retrieval and Information Extraction: A New Text Technology for Gathering Information from the Web. RIAO'97 Conference Proceedings, Canada, 1997, pp 356-370

12. Lazarinis F. Combining Information Extraction with Information Retrieval. MSc Thesis, Computing Science Department, University of Glasgow, Glasgow, Scotland, 1997

13. Morrissey M J. A Treatment of Imprecise Data and Uncertainty in Information Systems. PhD Thesis, Department of Computer Science, University College Dublin, Dublin, Ireland, 1987

14. Postma G J, Van der Linden J R, Smits J R M, Kateman G. TICA: A System for the Extraction of Analytical Chemical Information from Texts. In: Karjalainen E J (ed) Scientific Computing and Automation. Elsevier, Amsterdam, 1990, pp 176-181

15. Robertson S E. The Parameter Description of Retrieval Systems: Overall Measures. Journal of Documentation 1969; 25: 93-107

16. Robertson S E. The Parameter Description of Retrieval Systems: The Basic Parameters. Journal of Documentation 1969; 25: 1-27

17. Sager N. Natural Language Information Processing: A Computer Grammar of English and its Applications. Addison-Wesley, Reading, Massachusetts, 1981

18. Salton G, McGill M. Introduction to Modern Information Retrieval. McGraw-Hill Book Company, Computer Science Series, New York, 1983

19. Salton G, Wong A, Yang C S. A Vector Space Model for Automatic Indexing. Communications of the ACM 1975; 8: 613-620

20. Salton G. Automatic Text Processing. Addison-Wesley, Reading, Massachusetts, 1989

21. Shannon C E. A Mathematical Theory of Communications. Bell System Technical Journal 1948; 27: 379-423 & 623-656

22. Srinivasan P. Thesaurus Construction. In: Frakes W B, Baeza-Yates R (eds) Information Retrieval: Data Structures and Algorithms. Prentice-Hall, New Jersey, 1992, pp 161-218

23. Sundheim B M. (ed). Proceedings of the Fifth Message Understanding Conference. Morgan Kaufmann, San Francisco, 1993

24. Thompson P. Description of the PRC CEO Algorithm for TREC. TREC-1 Conference Proceedings, National Institute for Standards and Technology Special Publication 500-207, 1993, pp 337-342

25. Ullman J. Universal Relation Interfaces for Database Systems. Information Processing 1983; 83

26. Van Rijsbergen C J. Information Retrieval. 2nd ed, Butterworths, London, 1979

27. Voorhees M E, Gupta K N, Johnson-Laird B. The Collection Fusion Problem. TREC-3 Conference Proceedings, National Institute for Standards and Technology Special Publication 500-236, 1995, pp 95-104