Using Default Logic in Information Retrieval

Anthony Hunter

Department of Computing, Imperial College, London, UK

Abstract. Information retrieval involves uncertainty. In an information system, the user is not certain about the contents of the information system, and the system is not certain about the users needs. Information retrieval is about bridging this gap. In this paper, we show how this uncertainty problem can be addressed by using default logic.

1 Introduction

The aim of information retrieval is to provide a user with the "best possible" information from a database. The problem of information retrieval is determining what constitutes the best possible information for a given user. A common form of interaction for information retrieval is for the user to offer a set of keywords. These are then used by the information retrieval system to identify information that meets the users needs. For example, in a bibliographic database, a user might be interested in finding papers on some topic. The keywords would be an attempt to delineate that topic. This then raises key issues of precision (ensuring that a significant proportion of the items retrieved are relevant to the user) and recall (ensuring that a significant proportion of the relevant items are retrieved).

1.1 Statistical analysis versus semantic analysis

In order to determine how well matched an item in a database is for a query, most formal approaches to modelling the uncertainty in information retrieval use statistical information about keywords in the database. For example, a keyword in common between an item and a query, that occurs more in the item than in any other item, is a distinguishing feature of that item, and hence can increase the posterior probability that the item is of relevance to the user. A variety of such discriminating factors based on statistics have been proposed to quantify the similarity between items and queries (see for example Salton 1989, van Rijsbergen 1979).

Using probability theory has proven to be of significant value, and a variety of interesting proposals have been made including the probability ranking principle (Robertson 1977), and the binary independence model (van Rijsbergen 1989). Whilst there are a variety of problems with probabilisitc approaches (see for example Turtle 1995), we focus on the problem of the under-use of semantic information. In statistical analysis, the relationship between keyphrases is established by frequency ratios, whereas in semantic analysis, the relationship is established by "meaning". Not using semantic information is wasting valuable information that could be critical in matching a users needs to the information in the database. For example, suppose a search is undertaken using the keyword *car*, it would miss all items that only have the keywords *automobile* and *motor-car*. Similarly, suppose a search is undertaken using the request *computer-network or academic-communications*, an excessive number of items might be retrieved, whereas what might have been actually of interest to the user was the more specialized subject *internet*. Therefore, what is required is some formal representation of the semantic interrelationships between concepts, together with some ability to interpret users intended meanings when presenting requests.

1.2 The need for more than a thesaurus

Thesauri are widely-used tools in information retrieval. Essentially, a thesaurus is a database where for each keyword, there is a listing of synonyms, more specialized keywords, more general keywords, and related keywords. Usually they are not automated tools. Rather they are used directly by the user for consultation, with the onus being on the user to interpret and utilize the information in the course of composing a request.

In order to capture semantic information more fully in information retrieval, we need to automate the information in the thesaurus. In addition, we need more sophisticated information. In particular, we need context sensitivity. For example, suppose we have the keyword *car*. Then usually we would usually be interested in the synonym *automobile*. An exception would be if we also had the keyword *railway*. In which case we would usually be interested in the synonym *wagon*.

The lack of context dependency is one example of how there is no formal machinery for using thesauri. For automating the use of semantic information, we need to be able to specify when any particular specialization, generalization, synonym, or related term for a keyword can be used. Furthermore, we need to be able to extend this to resolving ambiguity such as arising from polysemes.

Previous approaches to semantic analysis in information retrieval do not provide a sufficiently expressive formal framework for exploiting semantic information. Yet, it is possible to use (non-monotonic) logics to handle semantic information about keywords, and so to identify logical relationships between items and queries. A logic-based approach can provide a richer alternative to probabilistic approaches.

In the following sections, we consider default logic as a formalism for semantic information in information retrieval. Default logic was proposed by Reiter (1980), and good reviews are available (Besnard 1989), and (Brewka 1991). Default logic, and variants, have well-understood properties including complexity and expressivity analyses. Whilst default logic is computationally problematical in the worst case, there are useful tractable subsystems. There is also promising work in approximations that is likely to provide, in the next few years, good quality reasoning with more appealing computational properties. In addition, viable theorem proving technology is now being developed, that should lead to robust inference engines in the next couple of years.

2 Using default rules for semantic information

In this section, we provide a framework, based on default logic, for capturing semantic information about keywords.

2.1 Keyphrase level

Let $\mathcal{K}$ be the usual set of formulae formed from a set of propositional letters and the connectives $\{\neg, \vee, \wedge\}$. We call $\mathcal{K}$ the set of keyphrases, and call any literal in $\mathcal{K}$ a keyword. The item keyphrase is a conjunction of literals. Intuitively this means the item contains information relating to each positive literal, and does not contain information relating to each negative literal. The request keyphrase is a specification by the user of what is of interest, and can be any formula in $\mathcal{K}$.

A factor in deciding whether an item is of interest to user, is whether the item keyphrase classically implies some or all of the request keyphrase. For example, take the item keyphrase α, and the request keyphrase $\alpha \vee \beta$, then the item would be of interest by this factor.

The choice of keyphrase can affect the recall and precision of the retrieval. A keyphrase might be based on concepts too general, or too specialized, or may fail to incorporate important synonyms. For each keyphrase, it is important to consider whether a more general, or specialized keyphrase should be used, or whether it should be used with some synonym, or even replaced by some synonym. We call this reasoning activity positioning.

2.2 Positioning for keyphrases

In order to formalize positioning for an item keyphrase, we assume semantic information is represented as a set of default rules. For this, let $\mathcal{L}$ be the set of predicates formed as follows: If α is a propositional letter in $\mathcal{K}$, then $in(\alpha)$ is in $\mathcal{L}$ and $out(\alpha)$ is in $\mathcal{L}$. Intuitively, $in(\alpha)$ is an argument for α being in the positioned keyphrase, and $out(\alpha)$ is an argument for α not being in the positioned keyphrase. We form the usual set of formulae from $\mathcal{L}$ and the connectives $\{\neg, \wedge, \vee\}$. We then form the default rules from $\mathcal{L}$ as usual.

For a default theory (D,W), D is some set of default rules and W is the smallest subset of $\mathcal{L}$ such that if α is a positive literal in the item keyphrase, then $in(\alpha)$ is in W.

For positioning, two important types of default rules are expansion and contraction. Expansion, of which the following is an example, intuitively states that if there is an argument for α being in the positioned keyphrase, then there is an argument for γ being in the positioned keyphrase.

$$\frac{in(\alpha) : in(\beta)}{in(\gamma)}$$

Contraction, of which the following is an example, intuitively states that if there is an argument for α being in the positioned keyphrase, then there is an argument for γ not being in the positioned keyphrase.

$$\frac{in(\alpha) : out(\beta)}{out(\gamma)}$$

The positioned keyphrase is generated as follows, where E is an extension generated as usual from (D,W).

$$keywords(E) = \{\alpha \mid in(\alpha) \in E \wedge out(\alpha) \notin E\}$$

If $keywords(E) = \{\alpha_1, .., \alpha_i\}$, then the positioned keyphrase is $\alpha_1 \wedge .. \wedge \alpha_i$. In this way the arguments 'for' and 'against' some α being in the positioned keyphrase are such that the arguments against α take precedence over arguments for α.

For example, suppose in a database of newspaper articles, we had an article with the item keyphrase, $mexico \wedge usa \wedge trade$. A reasonable generalization could be captured by the following expansion default rule:

$$\frac{in(trade) \wedge (in(mexico) \vee in(usa) \vee in(canada)) : in(nafta)}{in(nafta)}$$

Assume the default theory (D,W) where D is the above default, and W is $\{(in(mexico), in(usa), in(trade)\}$. Since $in(trade) \wedge (in(mexico) \vee in(usa) \vee in(canada))$ follows classically from W, and $in(nafta)$ is consistent with W, and the consequents of the defaults applied, then $in(nafta)$ holds. Hence the positioned keyphrase becomes $mexico \wedge usa \wedge trade \wedge nafta$.

In this example, we have positioned by using only one default rule. In practice, we would require many default rules.

2.3 Types of positioning

For a keyphrase β, we consider three types of positioning. These are defined, using the classical consequence relation $\vdash$, as follows, where β^* is the positioned keyphrase.

$$\text{(Strengthening) } \beta^* \vdash \beta \text{ and } \beta \not\vdash \beta^*$$

$$\text{(Weakening) } \beta^* \not\vdash \beta \text{ and } \beta \vdash \beta^*$$

$$\text{(Shifting) } \beta^* \not\vdash \beta \text{ and } \beta \not\vdash \beta^*$$

The intuitive nature of strengthening and weakening is clear. In shifting it is usually the case that there is some γ such that $\beta \vdash \gamma$ and $\beta^* \vdash \gamma$, and γ is only slightly weaker than both β and β^*. We show this by examples.

Suppose in our article database, we have an article with the item keyphrase, $in(olive) \wedge in(oil) \wedge in(cooking)$. A reasonable specialization could be captured by the following default rule.

$$\frac{in(oil) \wedge in(cooking) : in(\neg petroleum)}{in(\neg petroleum)}$$

Since $in(oil) \wedge in(cooking)$ follows classically from the item keyphrase, and $in(\neg petroleum)$ is consistent with the original item keyphrase, and consequents of the defaults applied, then $in(\neg petroleum)$ holds. So the positioned keyphrase becomes $olive \wedge oil \wedge cooking \wedge \neg petroleum$. This strengthening limits the ambiguity of the keyword *oil*, since the positioned keyphrase wouldn't be concerned with articles about *petroleum*.

Now suppose in our article database, we have an article with the keyphrase, $rail \wedge car$. Since *car* might not be regarded as an optimal keyword, the following could be useful.

$$\frac{in(rail) \wedge in(car) : in(wagon)}{in(wagon)} \quad \frac{in(rail) \wedge in(car) : out(car)}{out(car)}$$

From this, the positioned keyphrase becomes $rail \wedge wagon$. This is an example of shifting, and in this case it is intended to limit the ambiguity of using the keyword *car*.

Finally, we consider an example of weakening. Suppose we in our article database, we have an article with the item keyphrase *computer-networks* $\wedge$ *internet*. Since there are now many articles on computer-networks, it is perhaps better to focus on this article being about internet. This can be achieved by the following default.

$$\frac{in(computer\text{-}networks) \wedge in(internet) : out(computer\text{-}networks)}{out(computer\text{-}networks)}$$

From this, the positioned keyphrase becomes *internet*.

3 Obtaining default rules

In order to obtain default rules, we need a strategy for training (or generating) default rules, and for testing (or validating) them. We consider these processes in outline below.

3.1 The training process

Let $\mathcal{I}$ be some set of identification numbers for items, and $\mathcal{K}$ is the set of keyphrases. Let $\mathcal{R}$ be the set of pairs $(n, \beta_1 \wedge .. \wedge \beta_i)$ where $n \in \mathcal{I}$ and $\beta_1, .., \beta_i$ are keywords in $\mathcal{K}$. Let $\Gamma \subseteq \mathcal{R}$. Each keyword denotes a class in which the item n is a member, and so n is in the intersection of $\beta_1, .., \beta_i$. Now let Γ be a training set for deriving default rules. We regard the set of items (identification numbers) as a space that is divided by the classes generated by the keywords. Then we ask a user, or appropriate substitute, to consider this space of items and use their own keywords to classify the items. The default rules are derived from the mapping between how the item keyphrases classify the items, and how the user classifies the items.

For example, let Γ contain $(23, ford \wedge car)$, $(25, volvo \wedge car)$, and $(26, fiat \wedge automobile)$, and suppose the user classifies 23, 25, and 26 as $motorcar$. For this a default rule could be as follows.

$$\frac{in(car) \vee in(automobile) : in(motorcar)}{in(motorcar)}$$

We would then repeat this process for a number of users. This would allows us to capture a number of the synonyms, polysemes, and related terms that the users would expect when identifying items such as those found in the training set.

This process assumes that the training set is a reasonable approximation of the whole possible space of items. If not the default rules derived might cover an inadequate subset of the items, and furthermore, errors could be introduced. If there are significant examples missing, then exceptions to default rules might not be identified. This means default rules could be generated that could be applied in incorrect circumstances. The only guard against these problems is taking a sufficiently large training set and sufficiently large number of users. What constitutes "sufficiently large" can only be estimated by repeated training and testing cycles. In this sense, there is a commonality with knowledge engineering and inductive learning issues.

3.2 The testing process

To test, we assume a form of retrieval defined as follows, where $\Delta \in \wp(\mathcal{R})$, and $(n, \alpha) \in \mathcal{R}$. Let α be a request keyphrase.

$$\Delta \mid\!\sim_x (n, \alpha) \text{ iff } (n, \beta) \in \Delta \text{ and } match_x(\alpha, \beta)$$

where $match_x$, and hence $\mid\!\sim_x$, could be defined in a number of ways. We consider the following definitions in more detail here, called $match_1$ and $match_2$, where β^* is the positioned version of β.

$$match_1(\alpha, \beta) \text{ if } \beta \vdash \alpha$$

$$match_2(\alpha, \beta) \text{ if } \beta^* \vdash \alpha$$

The first definition is classical, or Boolean, retrieval. For both definitions, an item is retrieved only if its keyphrase is totally exhaustive with respect to the request keyphrase.

For a test set Δ, we can ascertain the probabilities $p(retrieved_x)$, which is the proportion of items in Δ retrieved by $\mid\!\sim_x$, and $p(relevant)$, which is the proportion of items in Δ that are relevant. We assume that items are classified as $relevant$ or $\neg relevant$ by some oracle. We also assume that if the training process is successful, then we have the following inequalities,

$$[1]\ p(relevant \wedge retrieved_2) > p(relevant \wedge retrieved_1)$$

$$[2]\ p(relevant \mid retrieved_2) > p(relevant \mid retrieved_1)$$

In [1] and [2], we make explicit the assumption that after training positioning is better at retrieving relevant items than Boolean retrieval. Though of course, it is not necessarily the case that the following holds.

$$[3]\ p(retrieved_2) > p(retrieved_1)$$

Using these probabilistic terms, we can define recall and precision as follows.

$$precision_x = \frac{p(retrieved_x \wedge relevant)}{p(retrieved_x \wedge relevant) + p(retrieved_x \wedge \neg relevant)}$$

$$recall_x = \frac{p(retrieved_x \wedge relevant)}{p(retrieved_x \wedge relevant) + p(\neg retrieved_x \wedge relevant)}$$

From these definitions and assumptions, we can derive the following.

$$precision_2 > precision_1$$

$$recall_2 > recall_1$$

Note, if $p(retrieved_1 \wedge relevant)$ is already close to 1, then there seems to be little need to use positioning. The need for positioning increases as $p(retrieved_1 \wedge relevant)$ decreases.

4 Conclusions

The use of logic, and particularly default logic, offers a more lucid and more complete formalization of uncertainty between items and requests. In addition, statistical and syntactic information can also be presented as default rules. Clearly, qualitative abstractions of statistical information about relationships between keywords can be used. In addition, syntactical information is often of the form of heuristic rules, and hence can also be harnessed. To illustrate, consider that in English there are about 250 suffixes, and that heuristic rules can be identified for adding or removing these suffixes from words. Since a request keyword and item keyword might have the same stem, but different suffixes, they would not match without the hueristics to translate them into the same form.

Since the logic-based approach needs to be non-monotonic, and needs to formalize meta-level reasoning, terminological logics, such as MIRTL (Meghini 1993), that have been proposed for information retrieval, do not seem to be an adequate alternative to default logic. Elsewhere (van Rijsbergen 1986, Chiaramella 1992), a conditional logic with probabilistic semantics has been proposed to capture the uncertainty pertaining to pairs of items and requests. The work in this paper constitutes an improvement on that approach, since we are assuming some well-studies formalisms and analyses in non-monotonic logics. Another attempt to provide a logic-based framework for modelling information retrieval, dicusses in detail how strict co-ordinate retrieval and Boolean retrieval can be viewed logically (Bruza 1994). This complements the work presented here, since both pieces of work are contributions to an analysis of information retrieval at the level of the consequence relation, though the work here is more focussed on non-monotonic issues.

5 Acknowledgements

This work has been supported by the ESPRIT Defeasible Reasoning and Uncertainty Management Systems (DRUMS2) project. The author is grateful for discussions with Howard Turtle and Steve Robertson, and for feedback from Peter Bruza and Fabrizio Sebastiani.

6 References

Besnard Ph (1989) An Introduction to Default Logic, Springer

Brewka G (1991) Common-sense Reasoning, Cambridge University Press

Bruza P and Huibers T (1994) Investigating aboutness axioms using information fields, in Proceedings of the 18th ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'94), ACM Press

Chiamaramella Y and Chevallet J (1992) About retrieval models and logic, The Computer Journal, 35, 233-242

Etherington D (1988) Reasoning with Incomplete Information, Pitman Press

Meghini C, Sebastiani F, Straccia U, and Thanos C (1993) A model of information retrieval based on a terminological logic, in Proceedings of the 16th ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'93), 298-307, ACM Press

Reiter R (1980) A logic for default reasoning, Artificial Intelligence, 13, 81-132

van Rijsbergen C (1979) Information Retrieval, Cambridge University Press

van Rijsbergen C (1986) A non-classical logic for information retrieval, The Computer Journal, 29, 481-485

van Rijsbergen C (1989) Towards an information logic, in Proceedings of the 12th ACM SIGIR Conference on Research and Development in Information Retrieval, 77-86, ACM Press

Robertson S (1977) The probabiltiy ranking principle in information retrieval, Journal of Documentation, 33, 294-304

Salton G (1989) Automatic Text Processing, Addison-Wesley

Turtle H and Croft W (1995) Uncertainty in information retrieval, in Smets Ph and Motro A, Uncertainty Management in Information Systems: From Needs to Solutions, Kluwer