Ontology-driven information integration of food industry related RSS news feeds

Pieter van den Brink

p.h.b.vandenbrink@student.utwente.nl

University of Twente

Faculty of EEMCS

MSc. Business Information Technology, University of Twente
November 22nd, 2007

Graduation committee:
Drs. C. Huijs, Faculty of EEMCS, University of Twente
Dr. A.B.J.M Wijnhoven, Faculty of MB, University of Twente
Dr. R. M. Müller, Faculty of MB, University of Twente
T. Spieard, Infortellence

Table of contents

Summary

In this master thesis, an information system is described that integrates news articles from various online sources, focusing on RSS news feeds, in the food industry domain. The research was initially triggered by the consultancy company Infortellence, which is active in this domain. The business goal was stated as:

To aggregate food-industry related news articles and provide a selection of this news tailored to a customer's interest, resulting in more interest in other Infortellence services.

The system makes use of an ontology which contains terms from the food domain, to automatically expand user queries with the aim to improve relevancy of the results to the user. Thus, the system has been labelled FORCA – Food Ontology-driven RSS Content Aggregator. In addition, the system enabled users to create profiles of their interest. These provide direct access to the latest news articles matching the profile.

To measure relevancy, several experiments were done which measured the information retrieval (IR) metrics of recall (percentage of all relevant articles that were retrieved) and precision (percentage of relevant articles among the ones that were retrieved). Relevancy was established through use of a gold standard: a domain expert evaluated all articles in the corpus (around 1600) for their relevancy to each of the 14 test queries.

The first round of experiments did not use any other IR techniques such as stemming. In the second round of experiments, stemming was implemented. This was found to grant a 6% increase in recall, with precision remaining stable. However, automatic expansion of queries with the ontology was found not to be beneficial overall. Narrower terms and synonyms were found to have little effect. Related terms and broader terms resulted in a noticeable increase in recall, but the loss in precision resulted in a lower overall performance.

Further research could focus on using a larger document corpus with a larger set of queries, or using a pre-classified corpus from a large commercial database. This would take away some of the subjectivity of relying on a single domain expert to do the gold standard classification. Another avenue is to focus on different, more user-focused metrics. The system was found to have value in saving time and effort to obtain information for its users. This could be further confirmed with research using representative domain users to trial the system, and evaluate it with a standardized questionnaire.

Finally, the FORCA system and methodology can be viewed as a business model that can be applied to other companies as well. The idea of generating interest through news aggregation ties in most closely with information service-related companies, such as consultancy companies. However, it can provide value to any company interested in attracting more visitors to their corporate website and learning more about their existing or potential customers.

1. Introduction

Since the advent of the internet, the amount of information content and its global accessibility have been rapidly increasing. Because of this, information overload has also become more and more of an issue. There is a wealth of information available, but determining which information is both relevant to the user and of good quality is no simple task.
A further complication is that information is not provided in one, standardized format. Providers are autonomous and hence information heterogeneity occurs on many levels. This is a problem when it comes to integrating this information. Given the distributed and fragmented nature of information on the internet, integration is virtually always necessary to get a complete picture.

As said, different levels of heterogeneity exist, which require different methods to overcome. There is technical heterogeneity, regarding differences in hardware and operating systems. Another type is syntactical heterogeneity, resulting from differences in machine-readable data formats. Finally, there is semantic heterogeneity, which consists of differences in modelling structures and meaning of the information. This is closely related to the used terminology and the context in which it appears (Sheth et al, 1999).

While many advances have been made to overcome technical and syntactical heterogeneities, semantic heterogeneities remain a problem area. This is logical, because standardizing the technical structure of information is relatively simpler than standardizing the underlying concepts. Also, the basic structure has to be taken care of before standardizing on a higher conceptual level (i.e. semantics) becomes possible. Various authors have proposed ideas on how to tackle semantic heterogeneity (e.g. Bergamaschi, 2001; Buccafurri, 2006; Naumann, 1999; Sheth et al, 1999). These authors also use the information brokerage concept: a third party gathers and processes information from various sources, and provides a single integrated view of this information to its client. An example representation is given in the figure below.

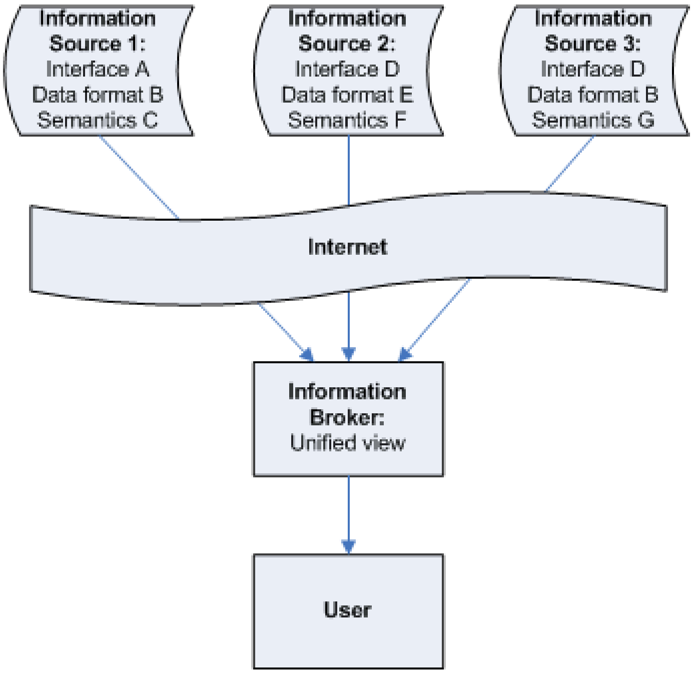

Information Source 1: Interface A Data format B Semantics C
Information Source 2: Interface D Data format E Semantics F
Information Source 3: Interface D Data format B Semantics G
Internet
Information Broker: Unified view
User

Figure 1 Information brokerage structure

Naumann (1999) further subdivides the information broker into a mediator-wrapper architecture. The mediator is responsible for hiding semantic differences from the user. Information is fed to the mediator by several wrappers, accessing different data sources. The wrappers thus hide technical and syntactic differences from the mediator and translate queries received from the mediator into a format understandable by the information source they access.

An important related field of research is that of domain ontologies. According to Fikes et al. (1995), domain-independent brokerage can only resolve syntactic heterogeneity. Effective brokerage can only be achieved by using domain-specific knowledge. A domain ontology forms a key part of that knowledge; it defines the terminology and the way terms are interrelated (this topic will be addressed in more detail in Chapter 3). This can form a basis that the mediator uses to create a unified view.

This thesis will contribute in this area by exploring how a domain ontology can be used as a mediating structure. By extending user queries with ontology terms and matching this against digital sources, the information that is most relevant to the user can be selected and presented in a single unified view. This prototype is created within the domain of a consultancy company operating in the Dutch food industry.

2. Case background

2.1. The Infortellence company

The context for this research project is the consultancy company Infortellence. It is a small start-up company based in the Netherlands, which targets small and medium enterprises in the food industry. An issue that applies specifically to Dutch SME’s is that they tend to lack attention to strategy and marketing (Syntens, 2004). The problem is not that there is no information to base strategic decisions on, but rather that there is an information overload. SME’s cannot divert the same amount of resources to obtaining and understanding this information as large enterprises. The result is that they tend to not concern themselves with business intelligence at all, because the value that can be gained is simply unclear to them. Thus opportunities are missed because of a lack of a clear strategic direction.

This is where Infortellence’s consultancy comes in. Infortellence aims to provide business intelligence (BI) to these SME’s, which means locating, obtaining and transforming relevant information to increase the value of the information to Infortellence’s customer. Thus, the information is presented in such a way that it can be understood more easily by the customer, so he can get insights from it to base strategic decisions on. The business intelligence reports created by Infortellence cover content such as new product analysis, competitor company profiles and country profiles. Tailoring information to the customer requires a good understanding of the needs of the customer. Therefore, Infortellence will first focus on serving SME’s in the Dutch food industry, since the company has considerable experience in this field. In time, Infortellence plans to expand their scope to German companies near the Dutch eastern border, as well as SME’s in other industries.

2.2. Business goal

One issue that remains is to make the customer aware of the added value this information has for them. Even if they are aware of a need for business intelligence, SME’s are typically hesitant to employ a consultant because of the high costs associated with it. The action problem for this case is thus as follows:

How to prove the added value of the consultancy information service to the customer?

An option that Infortellence is considering is to provide its services through a web portal. This can both lower the cost of the service, lowering the barrier for customers to get involved, as well as enable Infortellence to serve more customers, because it is not dependent on the consultant being physically present. The web portal would provide three distinct services:

- Providing paid content to customers. This includes both content generated by Infortellence itself, as well as relevant content selected from Infortellence's suppliers.
- Online consultancy: a structured communication channel to solve customer problems, for example by generating new reports on demand. If applicable, the newly created content can also be made available for purchase to other customers;
- Generating customer interest through additional free services.

The research will focus on the third aspect of generating interest, which is closely related to the action problem of proving the added value of the information. Tying this together with the subject of information integration and ontologies, the chosen method to realize this is to develop a prototype web application. This application will accumulate news articles from RSS feeds related to the food industry. A domain ontology will be used for expanding user queries, so that they may find and identify articles relevant to their interests with greater ease. Chapter 3 describes what exactly constitutes the concepts of RSS feeds and domain ontologies in more detail.

The method chosen to address the action problem is thus the creation of a news-aggregating application. Then, how can this demonstrate the added value of consultancy services? As stated in the previous section, SME's have little time to invest in gathering business intelligence, while this information certainly is valuable to them. An application that filters news articles from many sources, and provides the pertinent ones to the customer's interests, all in a single page, can help customers save time and effort. At the same time, they may read articles that pique their interest, and contact Infortellence for more in-depth information: an opportunity to deliver consultancy services.

Summarizing, the business goal of this project can be stated as follows:
To aggregate food-industry related news articles and provide a selection of this news tailored to a customer's interest, resulting in more interest in other Infortellence services.

3. Important concepts

This chapter addresses a number of core concepts which are essential for the understanding of this research. The first of these is that of news feeds – the online sources of information to be integrated. The other two concepts are the thesaurus and the ontology, which provide a means to integrate the information.

3.1. *News feeds*

News feeds, or web feeds, are data formats used to publicize regularly updated content on the internet. News feeds are designed to be machine-readable, making use of the XML language. Thus, they are interpreted by client software such as standard web browsers, or specialized

feed readers to conveniently provide news articles to a user. There are two main standards for news feeds: RSS (Really Simple Syndication) and Atom. These standards are similar, but are maintained by different parties and have different formats, for example relating to the names of elements. Although the proposed news-aggregating application will focus on dealing with RSS feeds, the formats are similar enough that Atom feeds will also be supported.
RSS news feeds follow a standardized format, which has at least the following structure for each individual news item:

```
<item>
      <title>          The heading of the news article </title>
      <link>           Hyperlink to the full text location of the article </link>
      <description>    A summary or the first few sentences of the article </description>
      <pubDate>        The date the article was published </pubDate>
</item>
```

In the rest of this document, the terms “news article” and “news item” refer to the information contained within such an <item> element in a news feed, which consists of a collection of several news items. When the complete text of an article is meant, rather than the summarizing information contained in the RSS feed, this will be explicitly referred to as “full text”.

The following is an example of how news information in a RSS feed looks like in practice:

```
<item>
      <title>Makers of Sodas Try a New Pitch: They’re Healthy</title>
      <link>http://www.nytimes.com/2007/03/07/business/07soda.html?ex=1330923600&en=133ff7c368883
      8ea&ei=5088&partner=rssnyt&emc=rss</link>
      <description>In coming months, Coca-Cola and PepsiCo will introduce carbonated drinks
      fortified with vitamins and minerals.</description>
      <author>ANDREW MARTIN</author>
      <guid isPermaLink="false">http://www.nytimes.com/2007/03/07/business/07soda.html</guid>
      <pubDate>Wed, 07 Mar 2007 08:05:24 EDT</pubDate>
</item>
```

As can be seen, often additional tags appear such as <author>, <category>, or <guid>. Also, even within the same fields the format may vary, most notably in the case of the date field. So, RSS news feeds are only semi-structured and some syntactic heterogeneity still exists which has to be resolved for the prototype.

3.2. *Thesaurus vs. ontology*

In this section a short overview of the terms ‘thesaurus’ and ‘ontology’ is given, since these are pivotal concepts within this research. A thesaurus is a model that describes a domain, through the use of standardized vocabulary terms and the interrelations between these terms. The term ‘ontology’ originated in philosophy, where Smith (2003) defines it as “the science of what is, of the kinds and structures of objects, properties events, processes, and relations in every area of reality.” However, in the context of this research we are more interested in the application of an ontology in the IS discipline – noted by Kishore et al. (2004) as a computational ontology. This differs from the philosophical ontology in the sense that it has a more limited scope (a domain), and has a pragmatic goal of contributing to IS applications. In the rest of this document, ‘ontology’ thus refers to a computational ontology. A succinct definition of a computational ontology is given by Gruber (1993): *an ontology is a formal explicit specification of a shared conceptualization*.

A thesaurus typically has only a limited set of relation types that describe relations between vocabulary terms. These core relations are shown in the table below.

Type	Description
SYN	Synonyms, the terms are essentially variations of the same concept.
NT	Narrower term, one term is a hyponym of the other term.
BT	Broader term, one term is a hypernym of the other term.
RT	Related term, a non-specific relation between the terms.

Table 1 Thesaurus relation types

So if we consider for example, a thesaurus that models the food industry, the following relation could exist: Apple-NT-Fruit ('Apple' is a Narrower Term of 'Fruit'.) Of course, the inverse of this relation also applies: Fruit-BT-Apple. The RT relationships is used as a generic link between terms that fall outside this hierarchical "is-a" classification, like for example Apple-RT-Apple juice.

A (computational) ontology is very similar to a thesaurus, but usually contains richer information through more specific relation types. For example, ontology relations could include "ingredient of" or "caused by." This provides more fine-grained information than the generic RT relationship found in a thesaurus. So, a distinction such as Apple-Ingredient of-Apple juice and Apple-Contains-Apple seed becomes possible, where both would be modelled simplistically as RT in a thesaurus.

For the purposes of this research, we consider thesaurus and ontology to the definition given by Gruber – *a formal explicit specification of a shared conceptualization*.

More specifically, a thesaurus is considered to be a subset of an ontology, containing fewer and less detailed relationship types in its specification.

4. Research method

This chapter describes various aspects of the research method used for this thesis. First, the overall purpose of the research will be addressed, and an overview of the information retrieval (IR) domain and prior related research will be presented. Then, the important IR concepts of recall and precision are introduced and put into the context of this research. Finally, based on all this, the research questions that will need to be answered are described.

4.1. Research purpose

The purpose of this research project is twofold. A design science approach is taken, which means one goal of the project is to construct a concrete artifact; in this case, an ontology based news information retrieval / aggregation system, which is of practical use to Infortellence. The other purpose is to add to existing literature by evaluating this prototype with an experiment.

The business goal of the project as previously stated is: *To aggregate food-industry related news articles and provide a selection of this news tailored to a customer's interest, resulting in more interest in other Infortellence services.*

Inspired by this statement we have labelled the prototype FORCA, which stands for Food Ontology-driven RSS Content Aggregator. When we use the terms "the system", "the prototype" or "the application", these all refer to FORCA unless noted otherwise. In addition to this business-oriented goal, there also is the research goal of contributing to the information retrieval and integration domain.

4.2. Domain and prior research

The main technical domain of the FORCA prototype is that of query expansion, which is a part of the broader information retrieval domain. An overview of research on query expansion is provided in Bhogal and Smith (2006). They define three approaches: relevance feedback, corpus dependent knowledge models and corpus independent knowledge models (thesaurus or ontology).

Relevance feedback involves users marking articles for relevance, once they have been retrieved as part of a search. Corpus dependent knowledge models attempt to find correlations within the content corpus, for example by determining co-occurrence between certain words to mark these as related. The relevance feedback and corpus dependent knowledge model approaches have the drawback that they depend on the available content: there must be a sufficiently large set of documents and each document must have a sizeable set of relevant terms, in order for query expansion to work adequately. Furthermore, corpus dependent models are less suitable for document collections that change often, which is the case for web content.

Therefore a corpus independent knowledge model approach (i.e., an ontology) is preferable for FORCA. Ontologies range from general to domain-specific. A well known example of a generic ontology is WordNet, which contains term definitions and relations for the entire English language. Although general ontologies cover a wide range of terms, this also means ambiguity is a problem. For narrow search tasks such as finding specific news articles, domain ontologies are preferred. The purpose of the FORCA prototype can be regarded as a narrow informational search, namely finding news articles related to the food industry. As such, using a domain ontology is the optimal strategy. Since the subject domain of FORCA is the food industry, a domain ontology related thereto should be used.

For this purpose, the AGROVOC database was chosen (AGROVOC, 2007). AGROVOC started out as a large thesaurus constructed by the Food and Agriculture Organization of the UN (FAO, 2007), and is currently being developed into a full ontology by extending it with more relations. A great advantage of this ontology is that it is large and provided by an authoritative source, thus it will keep being developed in the future. For the prototype we have obtained a copy of the AGROVOC database, which will reside on the same web server as the other prototype data. AGROVOC contains a large amount of thesaurus relations, whereas the more detailed ontology relations are still largely underdeveloped. Therefore FORCA, and this research, will make use only of the basic thesaurus relations: SYN, BT, RT and NT. Nevertheless, we still refer to AGROVOC as an ontology in recognition of their ongoing development project.

One work that is particularly relevant, and similar to FORCA, is the CIRI study (Concept-based Information Retrieval Interface) by Suomela and Kekäläinen (2006). Their system is based on a food domain ontology and a digital archive of a Finnish newspaper. It enables users to search for articles by selecting concepts from an ontology tree, as well as a basic search interface which does not use the ontology. They found that users found the ontology helpful to identify more search keys, but the basic search was slightly more effective overall. This is because when users could not find the concept they wanted in the ontology tree, they tended to choose other less relevant concepts. Another cause was unfamiliarity with the CIRI interface compared to a basic search interface.

There also are a few important differences between CIRI and FORCA. First of all, CIRI's document collection consisted of general articles from a single newspaper (thus, food-related articles were mixed in with other news). FORCA on the other hand has articles from multiple sources, but they are all related to food. For both systems a food ontology was used. However, it is likely that a food ontology is better in resolving semantic differences caused by different news providers within the same domain (FORCA) than it is in resolving ambiguity caused by the presence of articles and terms outside the subject domain, such as with CIRI. Furthermore, FORCA will not have an ontology tree interface like CIRI where users can select concepts. Instead, it will have a basic search interface; however, search terms will be mapped automatically to terms from the ontology as suggested by Suomela and Kekäläinen (2006). This mixed approach is also proposed by Bhogal (2006) as conducive to the navigability of the ontology, which is one of the success factors for query expansion.

4.3. Recall and precision

An essential aspect of the business goal of tailored news, is that this news should be *relevant* to the user. Determining relevancy is a long-standing issue in the information retrieval field. It is problematic, because relevancy is subjective in many aspects. Two users may judge the same article differently for the same query. Even a single user may consider the same article relevant or not relevant for the same query at different times, depending on exactly what information need he has in mind at that time, or what knowledge he already has obtained. We shall leave these issues for now – they will be revisited in Chapter 6, Experiment setup. First we consider the standard metrics that are used to measure relevancy: *recall* and *precision*.

Recall is defined as *the amount of relevant articles retrieved by the system, divided by the total amount of relevant articles in the collection*. Precision is *the amount of relevant articles retrieved, divided by the total amount of articles retrieved.* Apart from the core metrics of recall and precision, several derivative measures have been introduced, such as the *f-measure*, which is the harmonic mean of recall and precision. Such a metric makes it possible to summarize relevancy in a single value. For our purposes, we will keep to the basic recall and precision measures, so we can investigate the individual influence on both these aspects of relevancy.

A key issue in the information retrieval field is the tradeoff between recall and precision. By expanding queries (be it manual or automatic) recall increases, as more matches can be found with the additional terms. However, precision typically decreases at the same time, because there will be more results that do not match the concept the user had in mind. A nature-loving user might search for "trees", expecting to find information about those long cylindrical objects forests are composed of, but instead be confronted with page after page of tree graph models. The more terms are added to a query, the more ambiguity is introduced, especially if these terms are connected by a logical "or", as this can only result in more matches, not less.

Domain ontologies are able to help with keyword disambiguation by determining which meaning of the term should be used, as described in Bhogal (2006). However, this will not be the focus area for the FORCA research. The reason for this is the content of the FORCA corpus: all articles are related to the food industry. Thus, ambiguity is not so much an issue because alternative word meanings which would fall outside the domain do not result in additional matches.

This does not mean that the recall / precision trade-off is non-existent for this case. It is still possible for precision to drop when expanding a query, especially in the case of related terms (RT) where the relation between two terms is vague. Thus, we will look at which type of terms are best to enhance queries, and what the best method of enhancement is. A related work to this particular aspect is the research of Greenberg (2001). She noted that different types of relationships are better suited for different methods of enhancement. For example, Greenberg found that synonyms and narrower terms lend themselves well for automatic expansion. Using these terms resulted in greater recall without a statistically significant drop in precision. On the other hand, broader terms and related terms were more suited to interactive expansion, where users manually selected which terms they wanted to use to expand the query.

4.4. Research questions

We have to keep in mind the business goal of the prototype, which is generating interest and providing a tailored news overview. Our users, SME employees, do not want to be bothered with a laborious search process. This applies to users in general, but is even more pressing in this case because of the goal of generating interest. If the user finds the search process tiresome, he simply will not return. Users prefer a simple search process that can retrieve some relevant items over an involved one that produces the best results. Mann (1993) already identified this effect, calling it the *principle of least effort*. Thus, the query expansion process should be automated as much as possible.

With this in mind, we should focus initially on synonyms and narrower terms, as these are the most suited to automatic expansion according to Greenberg (2001). For the narrower terms, we will also investigate the effect of indirect narrower terms. Since an ontology is a tree-like structure, narrower terms often have narrower terms on their own again, which might also be useful for query expansion. Of course, broader terms and related terms will also be investigated, if only to confirm the results from Greenberg (2001).
Another aspect that will be investigated is the interaction between Boolean operators and query expansion. Boolean logic is often used in search engines, especially by more experienced users, and this could have a significant effect on the results of the query expansion.

Thus, we formulated the following main research question:

Which type of term relationships should be used to expand queries to ensure the greatest increase in recall, at the smallest cost to precision?

This question can be broken down into several sub-questions.

1. *What is the effect on precision and recall if the query is automatically expanded with every relationship type (BT, NT, SYN and RT), compared to a non-expanded query?*
2. *What is the effect on precision and recall if the query is automatically expanded with only synonyms and narrower terms, compared to a non-expanded query?*
3. *What is the effect on precision and recall if the query is automatically expanded with only broader terms and related terms, compared to a non-expanded query?*
4. *What is the effect on precision and recall of using a greater depth of narrower terms compared to expansion with only direct narrower terms?*
5. *What is the effect on precision and recall of the use of Boolean operators AND, OR, and phrases in conjunction with query expansion?*

The experiment that has been designed to answer these research questions is detailed in Chapter 6. First, the FORCA system itself will be described in Chapter 5. As a final note; the first implementation of FORCA expands queries automatically with NT, SYN and RT terms, so that a first idea of its functionality could be given.

5. The FORCA system

This chapter describes several design aspects of the software prototype under development
In the first two sections, the actors and use cases related to the system are described. Thus, the different types of user roles and ways of interaction with the system are considered. Section 4.3 provides an architectural overview of the system using the ArchiMate model notation (ArchiMate, 2007), and also discusses the technology used. In the last section, the logical and physical structure of FORCA is presented, illustrated with a class diagram.

5.1. Actor roles

There are three different types of actors that will access the FORCA prototype, namely the Basic User, the Registered User and the System Administrator. Rather than physical persons, these should be seen as actor roles. This is because one type of user can change into another, such as when a Basic User creates an account and becomes a Registered User. Another example is that the person who normally is a System Administrator may decide to just search for some articles, not using his administrator account, in which case he is actually accessing the system as a Basic User. Before moving on to the different interactions that these users can have with the system, first some more information about the user types:

Basic User: This is anyone who simply accesses the system to view news items, without having any account. Since the system will be freely accessible once it is taken into production, there are no restrictions on who can become a basic user.

Registered User: Once a basic user decides he wishes to use more features of the system, he will have to register an account, providing some information about himself. (In a later stage, the user’s account could also be linked to paid services of Infortellence. Then an actor role Paid User could be introduced. However, this is outside the scope of this prototype.)

System Administrator: The system administrator is responsible for maintaining the system, which includes setting parameters such as the RSS feeds to be used.

5.2. Use cases

In the following section the ways in which users can interact with the system are described, in the form of use cases. Use cases are a structured method to textually describe these interactions. The template used is given in the first table below. Furthermore, it should be noted that the use cases here are presented as compact as possible to keep clutter to a minimum. For example, there are no separate use cases for creation, updating and deleting of RSS sources, instead these are presented together as a single use case. This section starts with a use case diagram, which gives a global overview of all the uses cases, followed by the template used for the use case descriptions, and finally the use cases themselves.

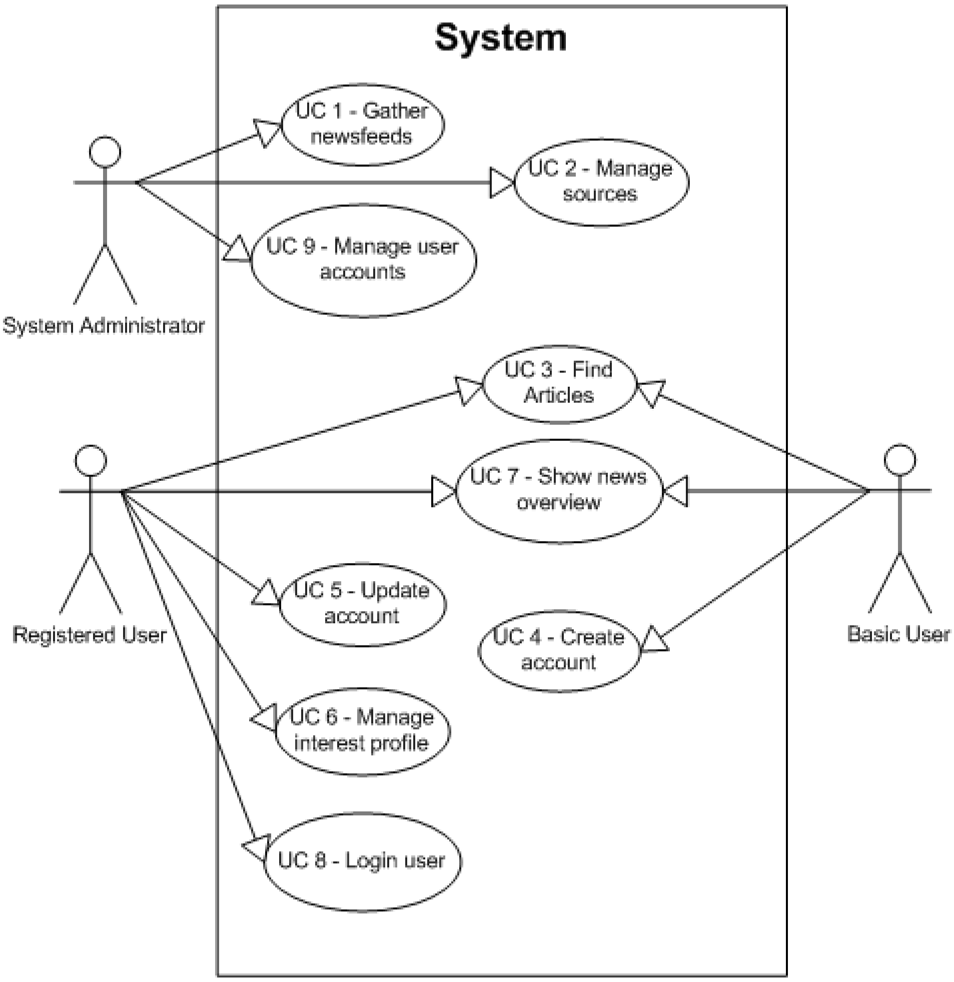


Figure 2 FORCA Use Case Diagram

The links between use cases and actor roles in the use case diagram are those that are the most relevant. Access rights are incremental from Basic User to Registered User to System Administrator. Thus, of course an administrator can also manage one of his interest profiles, but this use case is more relevant to the Registered User, as this is the role that is concerned with the use case.

Use Case #	Use Case Name
Summary	A short description of the use case.
Actors	Actors that are involved in this interaction.
Preconditions	Conditions that have to be true before the use case is executed, otherwise it cannot be guaranteed to produce the desired result.
Triggers	The action which causes the use case to be started.
Interactions	A stepwise description of the user actions and system responses, this is the main part of the use case.
Variations	Alternative actions / paths that may occur during the interactions.
Notes	Any other relevant information to this use case that does not fit in one of the other fields.

Use Case 1	**Gather newsfeeds**
Summary	Search various RSS news feeds and store new news articles in the local database.
Actors	System Administrator
Preconditions	• At least one RSS source has been entered into the system • System Administrator is logged in (if manual update)
Triggers	Time, or System Administrator requests a manual update of news feeds
Interactions	1. The system downloads a RSS feed. 2. The systems parses the feed to see if any new news articles have appeared. 3. The system stores the news articles in the local database. 4. The system adds source name and retrieval date as metadata to the articles. 5. The system updates the search index file with this article. 6. Repeat steps 1-5 for all other RSS feeds.
Variations	• If a certain RSS feed cannot be downloaded at step 1 for any reason, generate a warning to the system administrator and continue with the next RSS feed. • Optionally after step 6, if the System Administrator requested an update manually, provide an overview of the newly added articles.
Notes	-

Use Case 2	**Manage sources**
Summary	The System Administrator creates, edits or delete an RSS source that is to be aggregated by the system.
Actors	System Administrator
Preconditions	• System Administrator is logged in • If a new source is added: the source must consist only of food-industry related news items.
Triggers	The System Administrator activates the update sources function.
Interactions	1. The system presents an overview of current sources. 2. The System Administrator enters a new source, or modifies or deletes an existing one. 3. The results of the update are confirmed and the new overview is presented.
Variations	-
Notes	The precondition that the source must be food-industry related cannot be automatically verified; this is the responsibility of the system administrator.

Use Case 3	**Find articles**
Summary	A user finds news articles by entering a search query, which is automatically expanded by the system.
Actors	Basic User or Registered User
Preconditions	News articles are available in the database (i.e. Use Case 1 has been run successfully at least once)
Triggers	User initiates search for food-related news articles.
Interactions	1. The user enters a query via a search box. 2. The system expands the query with synonyms, broader terms, related terms and narrower terms. 3. The systems uses the expanded query to search through the news article database. 4. The matching news articles are presented to the user, as well as the expanded query that was used in the search, and an overview of the related ontology terms.
Variations	• Step 1: the query can also come from an interest profile, clicked by the user. • At step 3, the system could use the same query to search through other content (provided this content is structured the same way). The results are then presented separately at step 4. • For Basic Users, the extra results from the ontology expansion are not displayed. Instead they are told there are more results if they register.
Notes	Registered users can opt to use either basic search or ontology-enhanced search, and may choose not to display the overview of ontology terms.

Use Case 4	**Create account**
Summary	A Basic User creates an account by supplying some information about himself. He then becomes a registered user who can use all features of the system.
Actors	Basic User
Preconditions	-
Triggers	Basic User requests to be registered.
Interactions	1. The user supplies a desired account name and password, his real name, his company, and his e-mail address. 2. The user confirms. If all fields are properly filled in, the system saves the account and notifies the user. The keywords / categories supplied by the user are compared and mapped to the domain ontology categories. 3. Using the domain of the supplied e-mail address, the system performs a search on Google and retrieves the meta-tags from the first resulting page. These are added to the user's account as keywords. 4. The system prompts the user to create his first interest profile (see UC 6).
Variations	• If at step 2 not all fields are filled out correctly, the system gives a message and requests that the user fills these out correctly.
Notes	Step 3 will not be performed if the user supplied a free hosting provider such as Hotmail or Gmail, because this will not give relevant results. A list of such e-mail providers will be compiled, which the system can check.

Use Case 5	**Update account**
Summary	A Registered User changes his account information such as his password or e-mail address.
Actors	Registered User
Preconditions	The Registered User is logged in.
Triggers	Registered User initiates the account update process.
Interactions	1. The system displays all the account information which can be edited. 2. The user makes the desired changes to his information and confirms. 3. The system shows the updated information. If the user changed his e-mail address, the system also runs a search again to add new keywords to his account (the old keywords will always be saved.)
Variations	• If at step 2 not all fields are filled out correctly, the system gives a message and requests that the user fills these out correctly.
Notes	Users cannot delete their accounts.

Use Case 6	**Manage interest profile**
Summary	A Registered User creates, updates or deletes an interest profile. An interest profile consists of X English keywords that the user is interested in, in his own words. This will be used for showing him news overviews (UC 7.)
Actors	Registered User
Preconditions	The Registered User is logged in.
Triggers	Registered User initiates the account update process.
Interactions	1. The system displays the list of interest profiles. 2. The users selects one to edit or delete, or creates a new one. 3. If the user edits or creates an interest profile, the system prompts him to enter a number of keywords. 4. The user confirms. 5. The system converts the keywords to terms from the ontology and saves the profile. 6. The system presents the new overview of profiles.
Variations	• If the user opts to delete a profile at step 2, the system deletes the profile after confirmation and skips steps 3-5.
Notes	Users must have at least one interest profile, so they cannot delete a profile if it is the only one remaining.

Use Case 7	**Show news overview**
Summary	An overview of the X latest news articles is presented.
Actors	Basic User or Registered User
Preconditions	News articles are available in the database (i.e. Use Case 1 has been run successfully at least once)
Triggers	The user visits the home page or requests an overview for one of his profiles.
Interactions	1. The system retrieves the X latest news articles from the database. 2. The system displays the articles. (link, summary, and source)
Variations	• In the case of a Registered User that is logged in, before step 1, the keywords stored in his account and main interest profile are used to run a search. The X latest matching articles are then retrieved, instead of all X latest articles.
Notes	Articles are retained for 3 months and archived up to a year before deletion.

Use Case 8	**Login user**
Summary	A Registered User can log in with his username and password to access all features of the system.
Actors	Registered User, Administrator
Preconditions	The user has created an account before (UC4)
Triggers	The user initiates a login.
Interactions	1. The user fills in his username and password and confirms. 2. The system checks the username and password. 3. If correct, the user is taken to his account homepage.
Variations	• If at step 2, the username or password are incorrect, the system gives a message and requests that the user enters these correctly. • If the account belongs to an Administrator, he is taken to the Administrator home page instead at step 3.
Notes	A cookie could be used to automatically recognize and login the user.

Use Case 9	**Manage user accounts**
Summary	An administrator can view and edit details of the accounts of all registered users, as well as their interest profiles.
Actors	Administrator
Preconditions	The administrator is logged in.
Triggers	The administrator accesses the manage user accounts section.
Interactions	1. The system displays an overview of all registered accounts. 2. The administrator selects an account and updates account or profile information. 3. The system displays the updated overview of accounts.
Variations	-
Notes	The administrator cannot view or change account passwords.

5.3. Architectural design

To model the architecture of the FORCA prototype, the ArchiMate diagram notation is used (ArchiMate, 2007). The full ArchiMate model consists of four layers: environment, business, application, and technology. However, we will only use the application and technology layers because we are specifically interested in the architecture of the prototype itself at this point. To realize the prototype, a web-based approach has been chosen. Thus, it will run on a web server, and all interaction happens by users navigating through their browser. This also has the advantage that user do not need to learn to use a completely new interface, which makes it easier for them to use and enables them to get better results faster (Suomela and Kekäläinen, 2006).

A further point of note is that the Model-View-Controller (MVC) pattern will be used for the development of the prototype. This is a design pattern that separates the presentation aspect (the view) from the business logic that determines what actions are taken (controller) and the underlying data (the model). Compared to traditional web application development, where logic from all layers is mixed, using the MVC pattern results in software that is better maintainable and extensible.

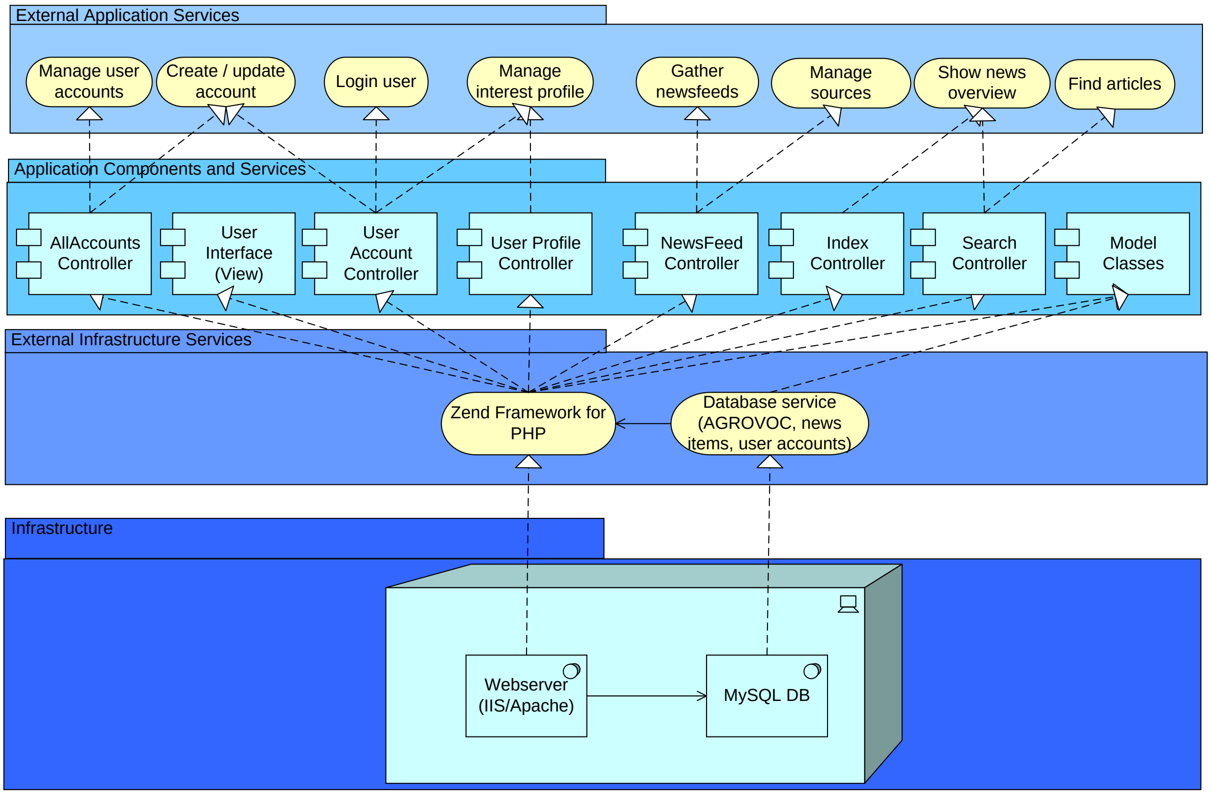


Figure 3 FORCA system architecture

The top layer in the above diagram consists of the External Application Services. These are the services that the prototype provides from the viewpoint of its users. Therefore they directly correspond to the use cases described in the previous section.
The second layer contains the Application Components and Services. These are the internal components of the prototype that are responsible for handling each of the use cases, consisting of Model, View and Controller classes. The Controller classes are responsible for connecting user requests to the underlying data via Model classes. Of the controllers, the search engine used in the Search Controller is the true centre of the application. It is based on Lucene, which is a well-documented, open source library of search engine functionality such as indexing and query parsing. Additional functionality has been developed to enable extension of queries with concepts from the domain ontology.
The View and Model classes are summarized in one component each to keep the diagram readable. The Model classes are abstractions of database tables, such as Newsfeed, Newsitem and Useraccount. These provide simple, high-level access to the underlying database service. The View classes (technically, they are not classes, but plain PHP / HTML files) are responsible for rendering output on the screen and gathering input via clicks and web forms.

The third layer is the External Infrastructure Services layer. The main service on this layer is the Zend Framework for PHP. This is a newly developed framework for creating PHP web applications. It has built in support for creating applications with the MVC pattern, and an implementation of the Lucene search engine. Furthermore it provides several other functions that are useful for FORCA, such as RSS parsing. Lucene is originally a Java application, but using the Zend implementation simplifies the prototype architecture, because now it can be developed entirely in PHP. Otherwise, a Java virtual machine and an interface between the Java application and the web layer would also have been necessary. This layer also contains the database service that provides data about customer accounts, news items and the ontology (the AGROVOC database).

The final layer consists of the infrastructure (devices and system software) and is rather simple. It contains a web server application and a database management system (MySQL) which both run on one physical server machine. Thus far, the prototype has been tested and run successfully on both Apache and Microsoft IIS web servers.

5.4. Application structure

As discussed in the previous section, FORCA follows the Model-View-Controller pattern. This chapter considers the logical and physical structure of the FORCA prototype. First, an overview of the classes is presented for both the controller and model classes. Then, the directory structure of the prototype is presented. Using this structure, the related controller, model, and view files are treated. Finally, there is a traceability matrix to couple the classes described in this section to the use cases described in the previous section.

5.4.1. Class diagram

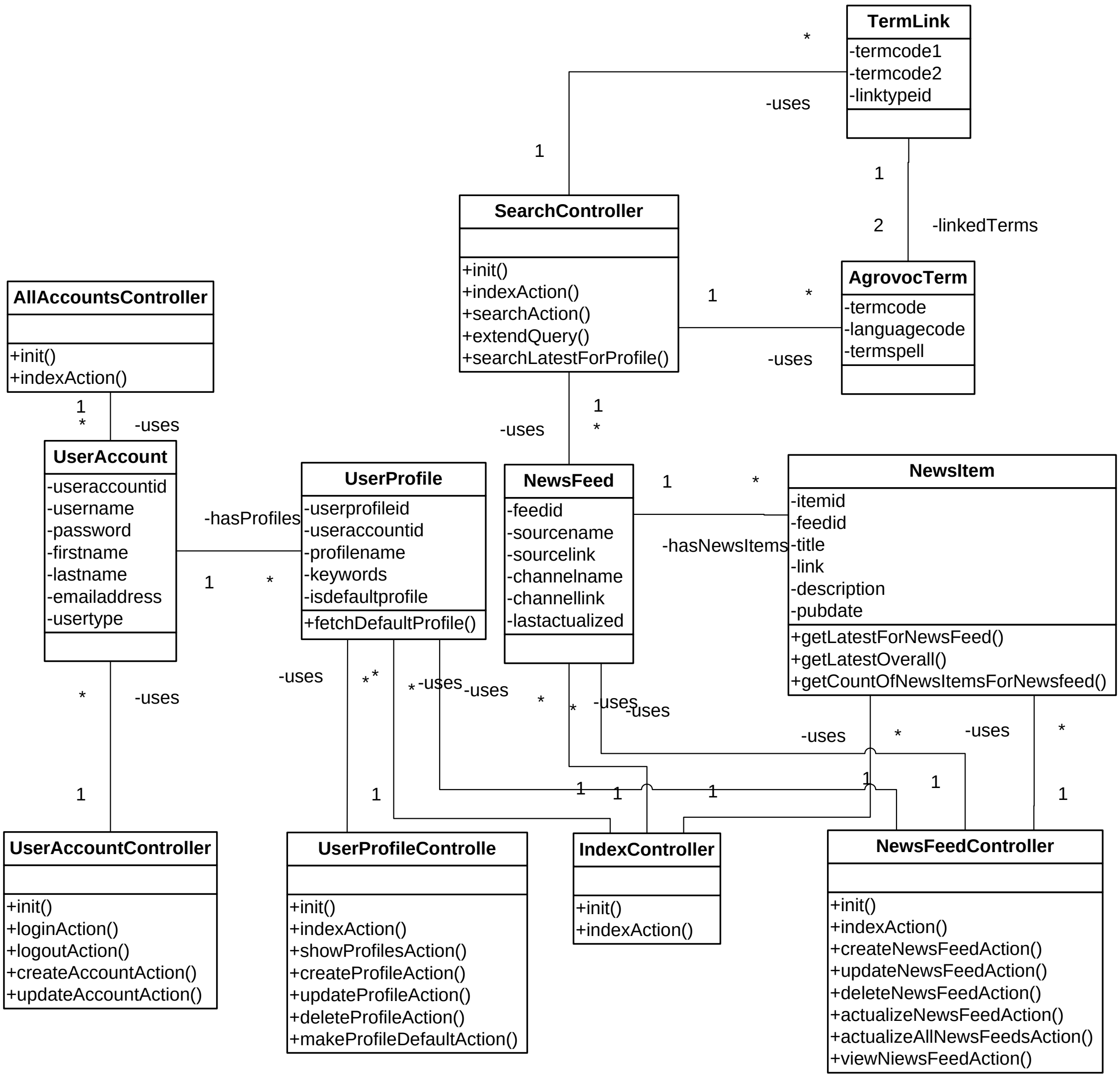


Figure 4 FORCA class diagram

The class diagram above shows all the Model and Controller files and their relationships. Since Views are not classes, they are not shown in the diagram. Note that the Model classes

are given attributes; these do not actually exist in the PHP class files, but rather in the database tables that correspond to each Model class.
A word about the database structure: for practical reasons, a single database is used. Thus, the FORCA-specific tables were added to the existing AGROVOC database. The AGROVOC tables themselves were not altered. Furthermore, the tables used from the Agrovoc database are AgrovocTerm and Termlink. AGROVOC contains more tables, but these are not used in FORCA and thus not modelled. The same goes for a number of fields within the TermLink and AgrovocTerm tables: only the fields used in FORCA are shown here.

5.4.2. Physical structure

The directory structure of FORCA is as follows:

```
FORCA\
        Application\
                Data\           Contains the Lucene index files of the ontology and news items.
                Models\         Contains classes responsible for accessing the database.
                Views\          Contains classes that are concerned with rendering a page as HTML.
                Controllers\    Contains the business logic classes that pass data to the views.
        Library\
                Zend\           Zend Framework classes.
        Public\
                Images\         Contains images, logos etc. used in the website.
                Styles\         Folder for CSS files, contains one master style sheet for the website.
```

The FORCA folder is placed in the web home directory of a web server. The application and library folders are secured with .htaccess files that deny direct remote access to the files themselves. Furthermore, the FORCA home directory contains an index.php file. This is a bootstrapper that catches all web requests, loads the necessary Zend Framework classes, and then dispatches the appropriate controller class, which takes further care of the request.

5.4.3. Controllers

The controller classes are responsible for most of the logic functions, such as updating the database with news articles or running the search algorithm. Controller classes do not store data themselves, but they access Model classes in order to obtain the necessary information. After processing the data the Controller classes pass the results of their action to a View, which will then be concerned with how it is presented as a web page.

The controllers folder contains the following files:

AllAccountsController.php	(AAC)
IndexController.php	(IC)
NewsFeedController.php	(NFC)
UserAccountController.php	(UAC)
UserProfileController.php	(UPC)
SearchController.php	(SC)

5.4.4. Models

The Models classes are abstractions of database tables. Each Model class corresponds to one database table, and provides access to this table in an object-oriented way, so that the Controller classes do not have to concern themselves with SQL statements. Because the data is stored in a database, and the Controller classes contain most of the function, some of the Model class files do not contain either functions or variables. However, the Model classes often provide convenience functions that give Controller classes easier access to information from the database.

The models folder contains the following files:

AgrovocTerm.php	(AT)
NewsFeed.php	(NF)
NewsItem.php	(NI)
TermLink.php	(TL)
UserAccount.php	(UA)
UserProfile.php	(UP)

5.4.5. Views

The Views are not actually classes, but rather plain files. They contain both HTML and PHP for formatting the data they are passed from a Controller and displaying it as a website. View files are also concerned with obtaining input from the user, for example through HTML forms. Once such input is obtained, it is passed to a Controller, which further handles it. It is possible to couple Views with a templating engine such as Smarty, but this has not been done in this case for the sake of simplicity.

The views folder contains the following folders and files:

Folder	Files
\	Footer.phtml Header.phtml
AllAccounts\	Index.phml
Index\	Index.phtml
NewsFeed\	Index.phtml _form.phtml Actualizeallnewsfeeds.phtml Actualizenewsfeed.phtml Createnewsfeed.phmtl Deletenewsfeed.phtml Index.phtml Updatenewsfeed.phtml Viewnewsfeed.phtml
Search\	_form.phtml Index.phtml Search.phtml
UserAccount\	Createaccount.phtml Index.phtml Login.phtml
UserProfile\	Createupdate.phtml Delete.phtml Index.phtml

Table 2 FORCA views

5.4.6. Use Case / MVC Traceability matrix

The traceability matrix shows which models and controllers are responsible for implementing each of the use cases. As can be seen from the matrix, all use cases have been implemented (at least to a certain extent) as they all have at least one corresponding model and controller. View classes are not presented in this matrix. This is mostly to maintain presentability, but also because the view classes are only concerned with rendering the information assigned to them by controllers, and are thus of secondary importance. Please refer to the previous Models and Controllers sections for a key of the used abbreviations.

Use Case	Responsible models and controllers										
	AT	NF	NI	TL	UA	UP	AAC	IC	NFC	UAC	UPC	SC
1 – Gather newsfeeds		X							X			
2 – Manage sources		X	X						X			
3 – Find articles	X	X	X	X								X
4 – Create account					X					X		
5 – Update account					X		X			X		
6 – Manage interest profile						X				X	X	
7 – Show news overview					X	X		X				X
8 – Login user					X					X		
9 – Manage user accounts					X		X					

Table 3 FORCA Use Case traceability matrix

5.4.7. Test classes

This section separately described the test classes which are used for the purpose of carrying out the experiment, so as not to mix them up with the rest of the prototype. The reasoning behind these classes can be found in Chapter 6, Experiment Setup. Here their structure and functionality is described. The class diagram on the next page shows an overview of the test structure.

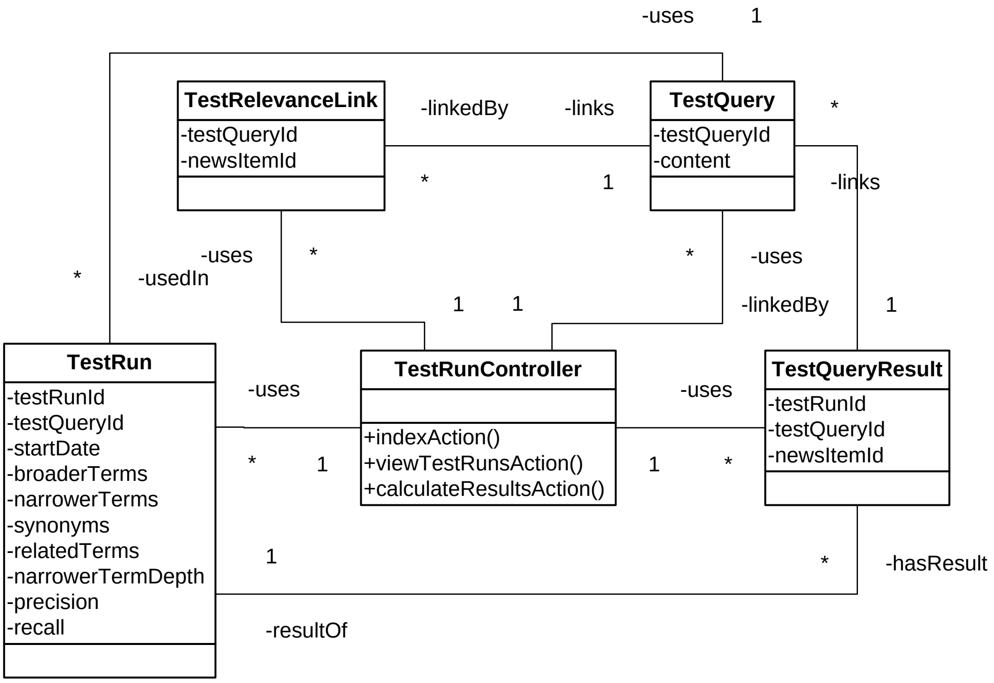


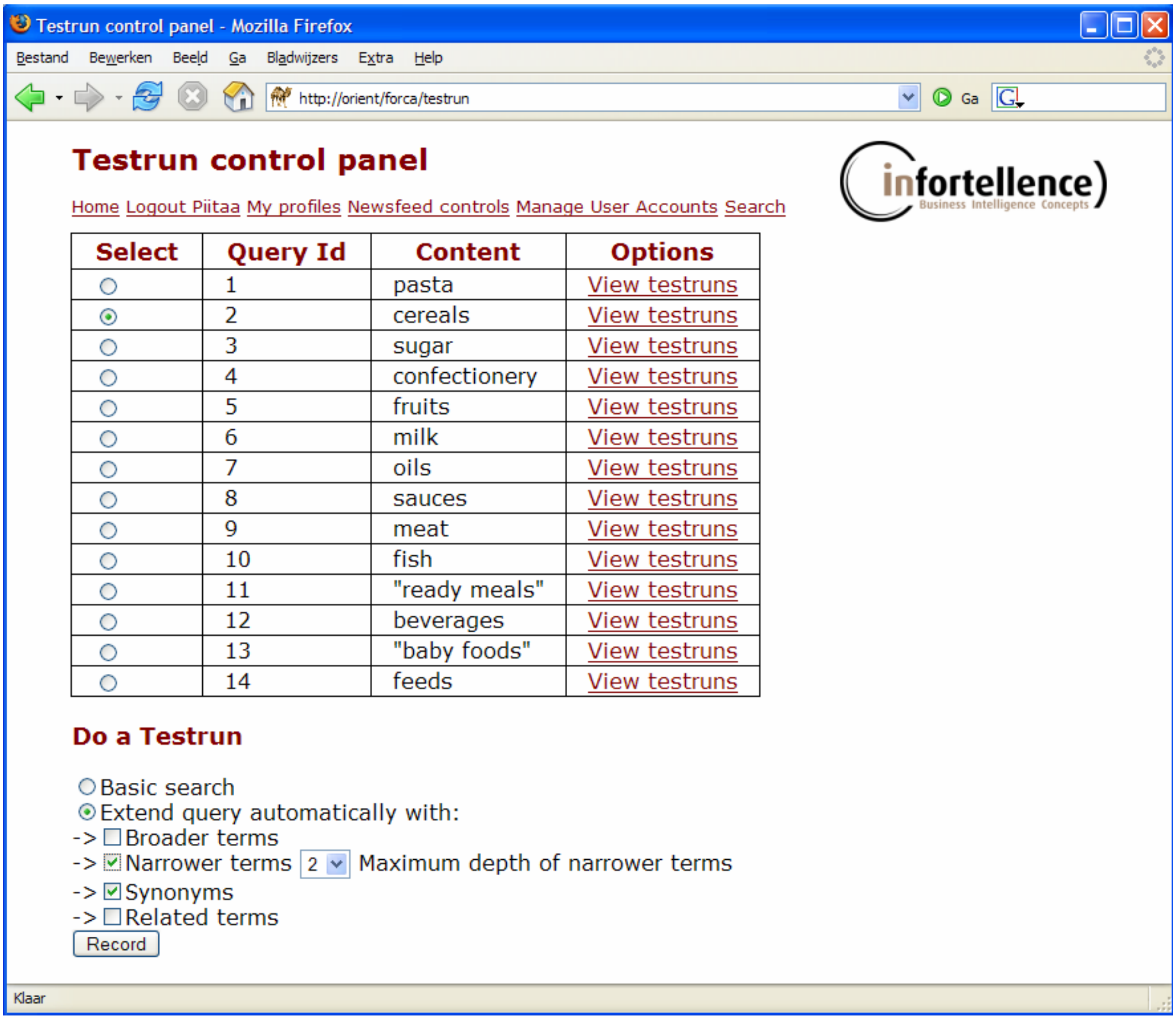
Figure 5 Class diagram for the test classes

The central class, TestRunController, is the central controller which coordinates all the actions related to testing. When visited, it shows a list of all the TestQueries and provides an interface to do new test runs with these queries. This interface can be seen in the FORCA screenshot on the next page.

The other four classes in the diagram are model classes that hold the necessary information. TestQuery contains the predefined query which will be tested. The TestRelevanceLink class is used to hold the expert classification of articles – each TestQuery is linked to the NewsItems that the expert considers relevant to this query.

The TestRun class hold some key information for each test run that is performed, such as the type of terms that were used for expansion, the depth of narrower terms used, and of course which TestQuery was used in this test run. The precision and recall fields are convenience fields, these will be filled automatically once the test run is complete and the values can be calculated. This provides direct access to the precision and recall values, as opposed to having to recalculate the results each time one wishes to access these values.

Finally, the TestQueryResult class corresponds to a NewsItem that was found with a TestQuery. A TestRun will thus likely have many TestQueryResults, which can be compared with the TestRelevanceLinks to calculate recall and precision.


Testrun control panel - Mozilla Firefox
Bestand Bewerken Beeld Ga Bladwijzers Extra Help
http://orient/forca/testrun
Ga
infortellence
Business Intelligence Concepts
Klaar

Testrun control panel

Home Logout Piitaa My profiles Newsfeed controls Manage User Accounts Search

Select	Query Id	Content	Options
○	1	pasta	View testruns
●	2	cereals	View testruns
○	3	sugar	View testruns
○	4	confectionery	View testruns
○	5	fruits	View testruns
○	6	milk	View testruns
○	7	oils	View testruns
○	8	sauces	View testruns
○	9	meat	View testruns
○	10	fish	View testruns
○	11	"ready meals"	View testruns
○	12	beverages	View testruns
○	13	"baby foods"	View testruns
○	14	feeds	View testruns

Do a Testrun

○ Basic search
● Extend query automatically with:
-> □ Broader terms
-> ☑ Narrower terms 2 Maximum depth of narrower terms
-> ☑ Synonyms
-> □ Related terms
Record

Figure 6 Testrun control interface

6. Experiment setup

In this chapter, the experiment that will be carried out using the FORCA prototype will be described. The research questions that this experiment will answer were defined in Chapter 4.4. They are provided here again for convenience.

Main research question:
Which type of term relationships should be used to expand queries to ensure the greatest increase in recall, at the smallest cost to precision?

Sub-questions:

1. *What is the effect on precision and recall if the query is automatically expanded with every relationship type (BT, NT, SYN and RT), compared to a non-expanded query?*
2. *What is the effect on precision and recall if the query is automatically expanded with only synonyms and narrower terms, compared to a non-expanded query?*
3. *What is the effect on precision and recall if the query is automatically expanded with only broader terms and related terms, compared to a non-expanded query?*
4. *What is the effect on precision and recall of using a greater depth of narrower terms compared to expansion with only direct narrower terms?*

5. *What is the effect on precision and recall of the use of Boolean operators AND, OR, and phrases in conjunction with query expansion?*

6.1. Preparation

In order to calculate the variables recall and precision, it becomes necessary to define exactly when an article is 'relevant'. No article is relevant without a context, so we will create a set of information requests as a point of reference. These requests and the queries that fulfill them should be neither too long nor too short. Bhogal (2007) found that query expansion is most effective if the original query is short. However, real queries are often longer than a single word. Furthermore, we are also interested in the effects of Boolean logic, also requiring more than a single word. Thus, we will use a test query size of two words per query.

With only two keywords per query to go on, it is hard to make a judgment of relevance. This is why broader information requests are used. In this context, information requests are a description of an information need using natural language. This practice is also used in the well-known TREC collection (Text REtrieval Conference, see for example TREC (2005)), where such detailed information requests are called 'topics'. In the TREC research, search engines had to derive their own queries from these topics, or automatically process the topics in another way to conduct the search. For the purpose of this research, we simply define a standard query that is related to each information request. This approach is also taken in other small to medium test collections such as the CACM and ISI collections.

Since the FORCA test collection is relatively small, very specific information requests and queries would result in queries that may return no matches, which would make evaluating results difficult. On the other hand, with information requests that are too generic the task of evaluating relevance becomes harder and more subjective. The test information requests that are used here attempt to strike a balance with some more generic requests, and some more specific ones, without going too far to either extremes.

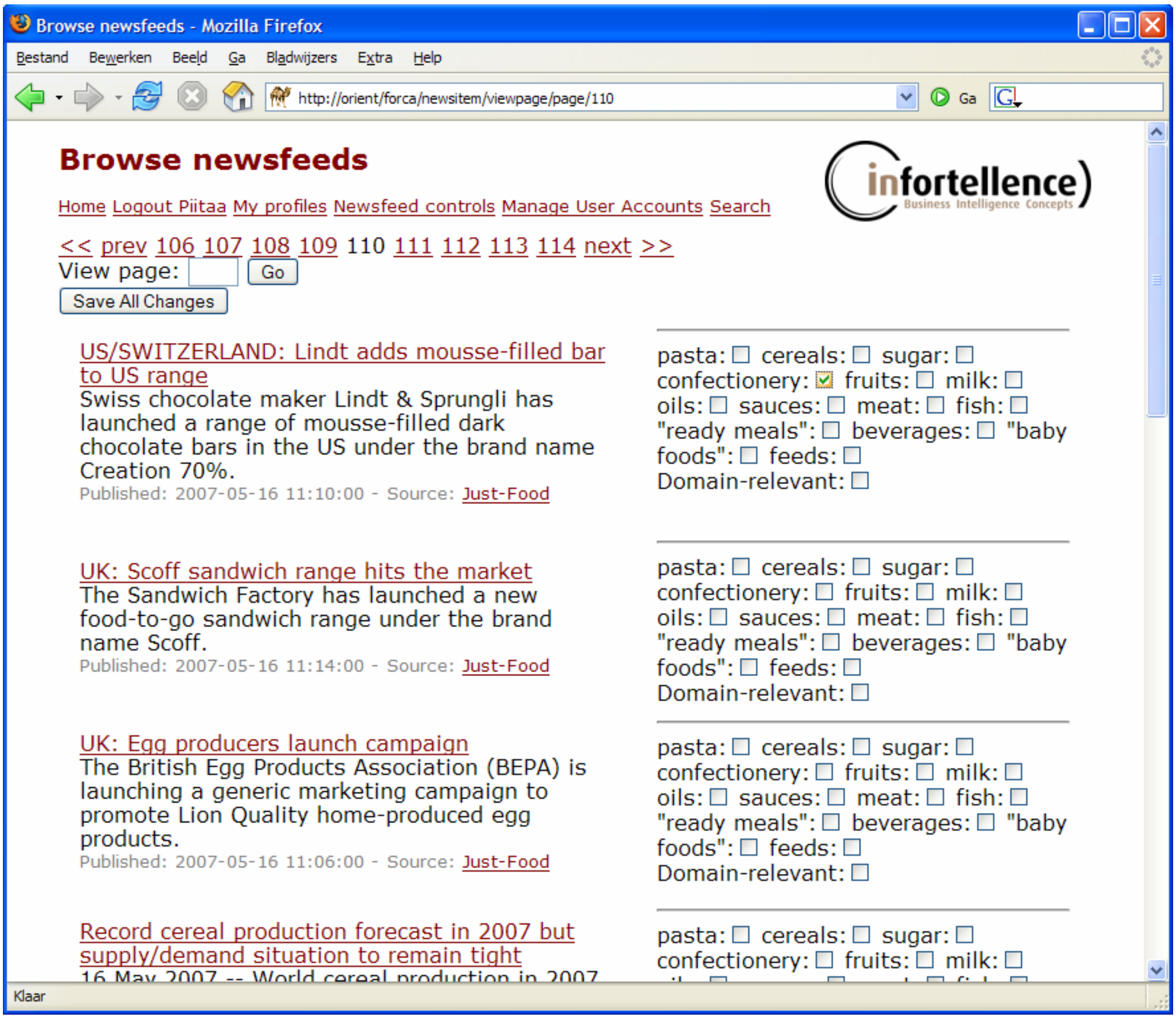
Nr.	Description of information request	Query keywords
1.	Which articles contain information about any sort of dairy products, such as milk, cheese or butter?	Dairy products
2.	Which articles describe products that are made from potatoes, or products that contain potato starch?	Potato starch
3.	Which articles contain information about ice cream products (the dessert type)?	Ice cream
4.	Which articles are about fruit juices (such as apple or lemon juice, but not including soft drinks like cola)?	Fruit juices
5.	Which articles are about any foods or drinks which are specifically intended for infants?	Baby foods
6.	Which articles are about tomato soup, describing for examples recipes or ingredients?	Tomato soup
7.	Which articles contain information about vinegar or vegetables that are often used with vinegar (for example in a salad)?	Vinegar vegetables
8.	Which articles deal with fatty acids, for example considering health effects, or discussing the fatty acid content of a specific product?	Fatty acids

9.	Which articles deal with products that are in powder form such as milk powder or coffee creamer?	Powder products
10.	Which articles consider eggs, products made from eggs, or other sources of protein?	Eggs protein
11.	Which articles deal with cheeses made in France or typically associated with France?	French cheese
12.	Which articles deal with products that are used as dairy substitutes?	Non dairy
13.	Which articles contain information about chocolate mousse, either as dessert or as part of other products?	Chocolate mousse
14.	Which articles are about ready meals, that is, convenience foods such as instant meals or junk food?	Ready meals

Table 4 Test information requests overview

Now we have a context for which we can say a result is relevant or not. However, the question of how to determine relevance still stands. We cannot use the ontology to automatically determine relevance, because the effect of the ontology itself on recall and precision is what we are trying to establish. For recall, there is also the problem of non-found, yet relevant articles. To overcome this issue, we use a gold standard approach where a food domain expert will manually classify which articles in the FORCA corpus are considered relevant for each of the queries above. This corpus consists of 1570 news articles gathered between February and June of 2007, from various RSS news feeds focussing on the food industry. The corpus is thus small enough to make a gold standard classification feasible. With this classification we can accurately calculate recall. On the other hand, the size is large enough that there will be matching articles for most, if not all, of the test queries. A screenshot of the FORCA classification interface is provided on the bottom of this page.

This classification of relevance can then be used to evaluate the performance of the FORCA ontology search results, using various methods of query expansion. The classification will be saved in the FORCA database, so that calculation of precision and recall can be done online automatically. In addition, the domain expert will also note for each article whether or not he considers it relevant to the food industry in general. (Some articles currently exist which are not b2b-oriented, or are mere advertisements.) This note will not be used for this experiment, but to improve the quality of the corpus later from a business point of view.


Browse newsfeeds - Mozilla Firefox
Bestand Bewerken Beeld Ga Bladwijzers Extra Help
http://orient/forca/newsitem/viewpage/page/110
Ga
infortellence
Business Intelligence Concepts
Klaar

Browse newsfeeds

Home Logout Piitaa My profiles Newsfeed controls Manage User Accounts Search

<< prev 106 107 108 109 110 111 112 113 114 next >>

View page: Go

Save All Changes

US/SWITZERLAND: Lindt adds mousse-filled bar to US range
Swiss chocolate maker Lindt & Sprungli has launched a range of mousse-filled dark chocolate bars in the US under the brand name Creation 70%.
Published: 2007-05-16 11:10:00 - Source: Just-Food

pasta: □ cereals: □ sugar: □ confectionery: ☑ fruits: □ milk: □ oils: □ sauces: □ meat: □ fish: □ "ready meals": □ beverages: □ "baby foods": □ feeds: □ Domain-relevant: □

UK: Scoff sandwich range hits the market
The Sandwich Factory has launched a new food-to-go sandwich range under the brand name Scoff.
Published: 2007-05-16 11:14:00 - Source: Just-Food

pasta: □ cereals: □ sugar: □ confectionery: □ fruits: □ milk: □ oils: □ sauces: □ meat: □ fish: □ "ready meals": □ beverages: □ "baby foods": □ feeds: □ Domain-relevant: □

UK: Egg producers launch campaign
The British Egg Products Association (BEPA) is launching a generic marketing campaign to promote Lion Quality home-produced egg products.
Published: 2007-05-16 11:06:00 - Source: Just-Food

pasta: □ cereals: □ sugar: □ confectionery: □ fruits: □ milk: □ oils: □ sauces: □ meat: □ fish: □ "ready meals": □ beverages: □ "baby foods": □ feeds: □ Domain-relevant: □

Record cereal production forecast in 2007 but supply/demand situation to remain tight
16 May 2007 -- World cereal production in 2007

pasta: □ cereals: □ sugar: □ confectionery: □ fruits: □ milk: □

Figure 7 Article classification interface

6.2. *Variables and scenarios*

In order to answer the 5 sub-questions, we will use a number of scenarios to evaluate each of the test queries. Before that, the relevant variables that will be measured must be defined. These can be derived from the research questions as follows.

- Term type (SYN, NT, BT and RT). Measuring this variable enables us to answer sub-questions 1, 2 and 3.
- Narrower term depth. Knowing the effect of this variable provides the answer to sub-question 4.
- Boolean operators (AND, OR and phrase logic). By measuring this variable sub-question 5 can be answered.

Each of the variables will have a separate set of scenarios, where the other variables are held constant and are considered independent. In the following sections the test scenarios for each variable will be described.

6.2.1. The baseline scenario

It is necessary to establish a baseline scenario with which the other scenarios can be compared, so that it is possible to evaluate whether there are improvements in recall and precision. For the baseline scenario, the queries will be used without using any term expansion or modification. The Boolean operator is by default OR. Thus, the query (maple syrup) would be interpreted as maple OR syrup. The baseline scenario will be executed once for each of the 15 queries, thus requiring 15 test runs.

6.2.2. Term type

For the different term types, first of all each of the terms will be considered separately. This gives four categories for synonyms, narrower terms, broader terms and related terms. The fully expanded scenario where all terms are used should also be considered to give an overview of the total effect.

- Individual, SYN / NT / BT / RT separately for a total of 4 scenarios
- Complete, SYN, NT, BT and TR combined for 1 scenario

The total amount of scenarios required for term types is thus 5. Each of these scenarios will be compared with the baseline scenario. Here, narrower term depth will be held constant at 1, i.e. only directly narrower terms are taken into account for the query expansion where narrower terms are used. The Boolean logic is OR, for the whole expanded query.

6.2.3. Narrower term depth

These scenarios consider the effect of using further, indirectly connected narrower terms. If for example the ontology contains the relations wines-NT-alcoholic beverages-NT-beverages, then ‘wines’ will be added to the query when searching for ‘beverages.’ For evaluation, the scenarios will not be compared with the baseline scenario. Instead, they will be compared with the individual term type scenario with only narrower terms of depth 1, as is part of the previous section.
The maximum term depth of the ontology is unknown, however, in many cases the amount of steps required to go from the most generic term (e.g. “food”) to the most specific one (e.g. “white wines” is 4 or less. The most generic terms will not be used in queries in this experiment, for obvious reasons (too little selectivity). Consequently, there will be scenarios for narrower term depths of 2 and 3 levels deep. The total amount of scenarios for the narrower term depth variable is thus two. Again, all the terms are linked together with a logical OR.

6.2.4. Boolean operators

As we wish to research the effects on Boolean operators on an expanded query, we cannot use the baseline queries as a comparison. This presents the problem of which kind of expanded query to use. Obviously evaluating *all* term type scenarios again with AND and phrase logic would result in an explosion in the required test runs. Furthermore, it is not necessary to do this, because we are not interested in the correlation between Boolean operators and specific term types. Thus we choose to use the “complete” term type scenario as a standard to compare with. This requires two additional scenarios, one were terms are linked together with AND logic, and one where terms are linked as a phrase. This only applies to the terms in the basic queries, further terms that are added to the query by expansion will still be linked by OR.

7. Initial experiment results

In this chapter the initial results of the experiment are presented. The main text will contain the results in diagram format, and discuss them in the order they were presented in in the previous chapter: starting with the baseline results, then the different term types, and lastly results with Boolean AND, and phrase logic. Also available are the results in numerical format, for these please refer to Appendix A: Experiment result tables. This appendix also contains the absolute number of articles retrieved with each query. Because precision and recall are given in percentages, the actual number of articles can give additional information when percentage differences are small.

7.1. Baseline results

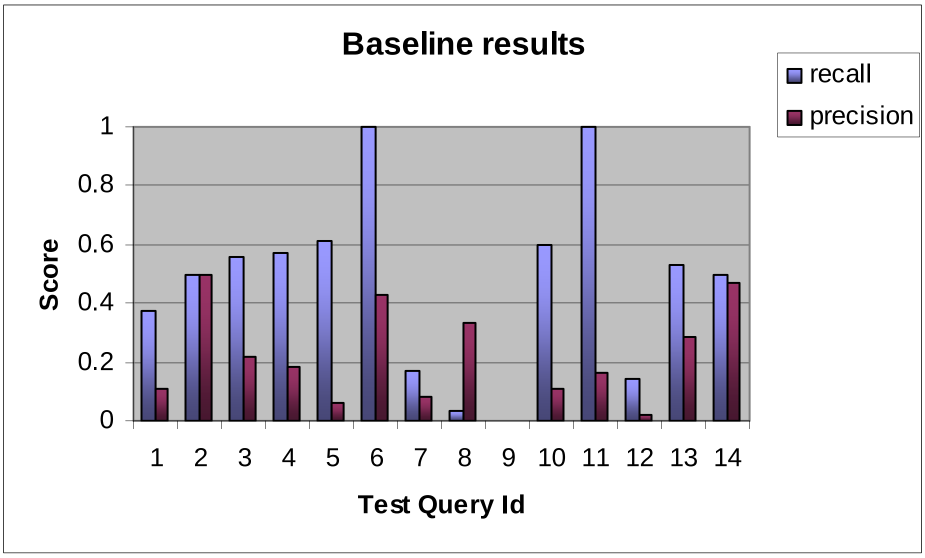


Figure 8 Baseline results

The above diagram shows the results of the baseline search, without any term expansion or modifications. A few observations can immediately be made. First of all, query number 9 (powder products) produced 0 recall and precision. There are a few articles which were judged relevant for this information need in the collection, but none of these were retrieved, thus resulting in the 0 score for both recall and precision. Furthermore, queries 6 (tomato soup) and 11 (French cheese) both had a perfect recall score of 1, even with just the baseline query. For the other queries, recall was mostly in the 0.3 – 0.6 area, with an average of 0.47. Precision was overall lower than recall, having an average measure of 0.21. The only exception to this is query 8 (fatty acids), where precision is significantly higher than recall. This is likely due to the small amount of matching results for this query (only 3). There are many more relevant articles on this topic, but these do not contain the words fatty or acids.

7.2. *Term type results*

Since the term type results overall did not vary much from the baseline scenario, only each term's individual scenario and the scenario with all terms together are shown here. Each section also contains some remarks about the important observations for each scenario.

7.2.1. Broader terms individually

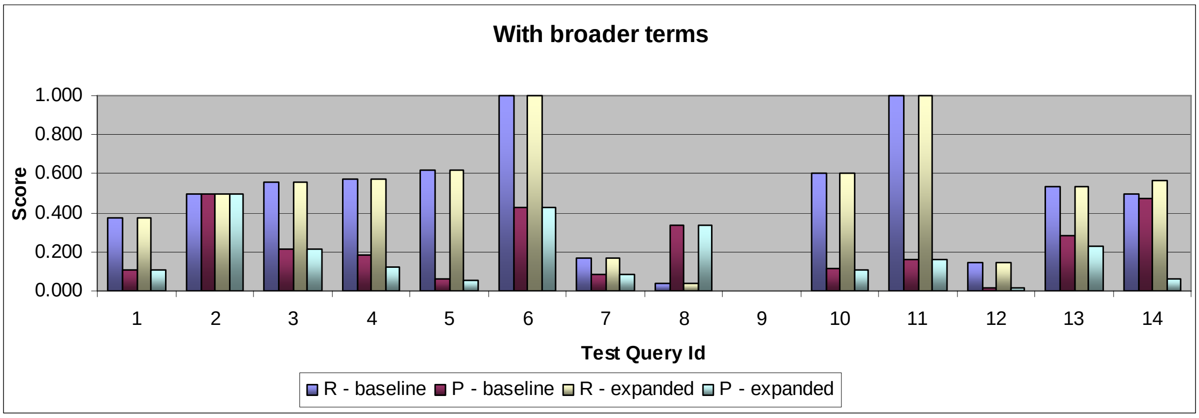


Figure 9 Results with broader terms

The figure above shows the results for the baseline query expanded with broader terms. Comparing this with the baseline charts, there is little difference in recall performance, with only query 14 (ready meals) showing a slight improvement. However, there is also a big drop in precision for this query. Some other queries also display drops in precision, but not as noticeable. This suggests that more articles have been found thanks to the expansion, but those additional articles were not among the ones judged relevant to the topic. For other queries, no broader terms could be found, or the broader terms added did not result in any new articles (whether relevant or not).

7.2.2. Narrower terms individually

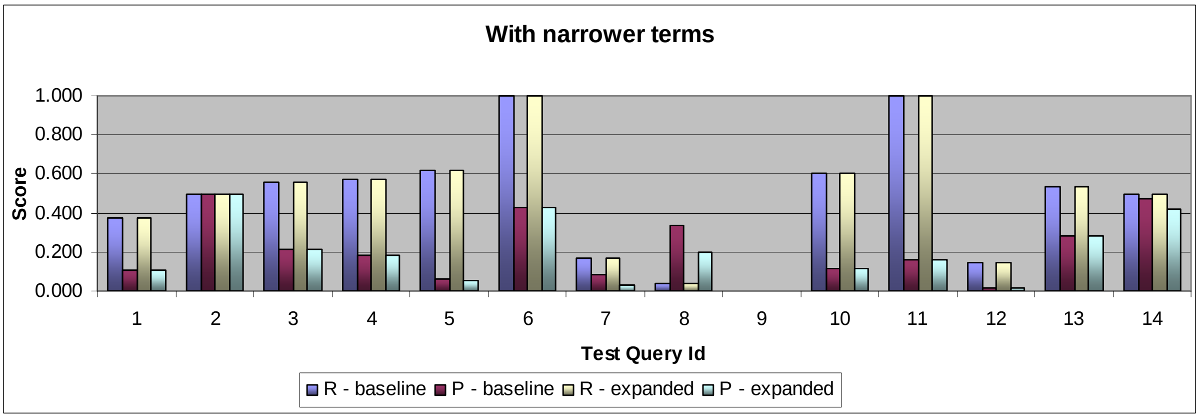


Figure 10 Results with narrower terms

The narrower term results were overall comparable with the baseline results. No improvements in recall were found in this set, but the decrease in precision was also very small, namely 0.19 compared to the baseline value of 0.21.

7.2.3. Synonyms individually

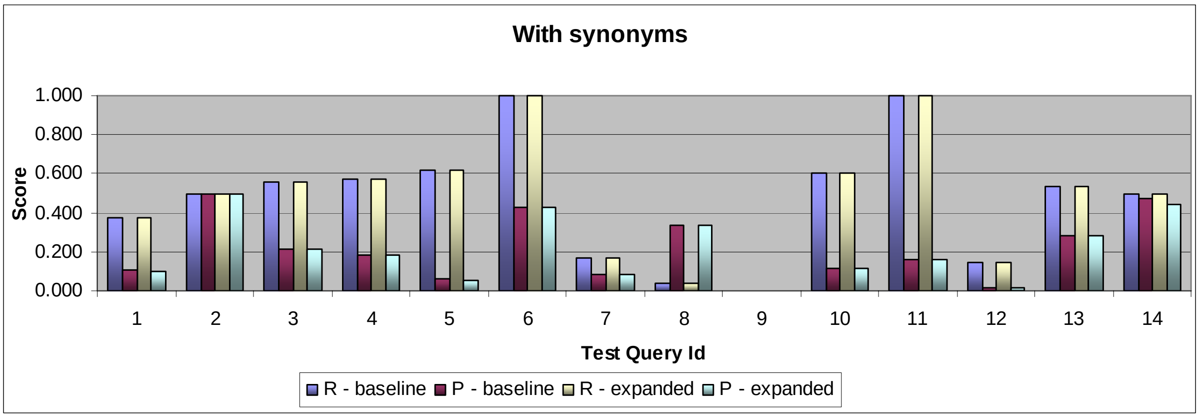


Figure 11 Results with synonyms

For the synonyms, as with narrower terms, the results were almost the same as with the baseline scenario. A few additional articles were found for some of the queries, but these unfortunately were not relevant. The precision value was virtually identical to that of the baseline scenario. Since recall did not increase, this means that expanding with synonyms resulted in fewer additional results than expanding with narrower terms.

7.2.4. Related terms individually

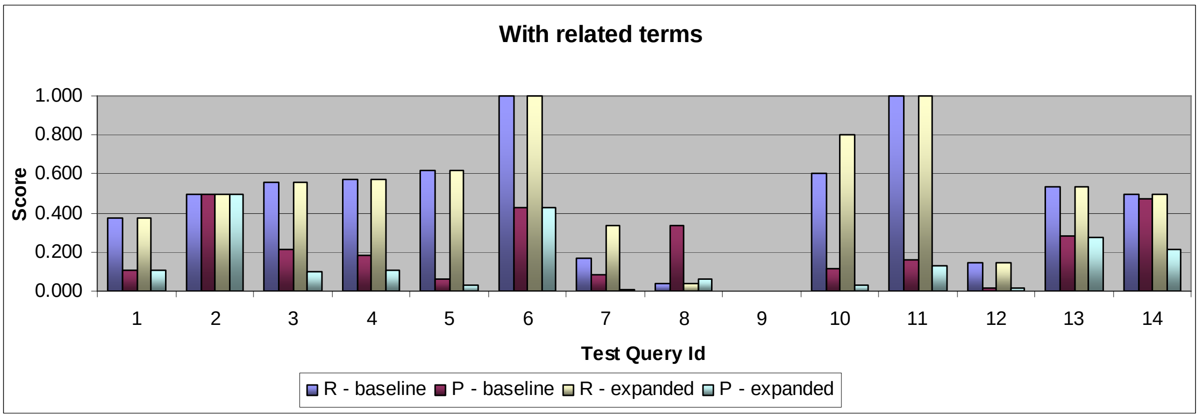


Figure 12 Results with related terms

Expansion with related terms showed an improvement in recall for queries 7 (vinegar vegetables) and 10 (eggs protein). However, the drop in precision was also the largest here, among the individual term expansions. The average precision measure when using related terms fell down to 0.14, compared to the 0.21 of the baseline scenario.

7.2.5. All terms

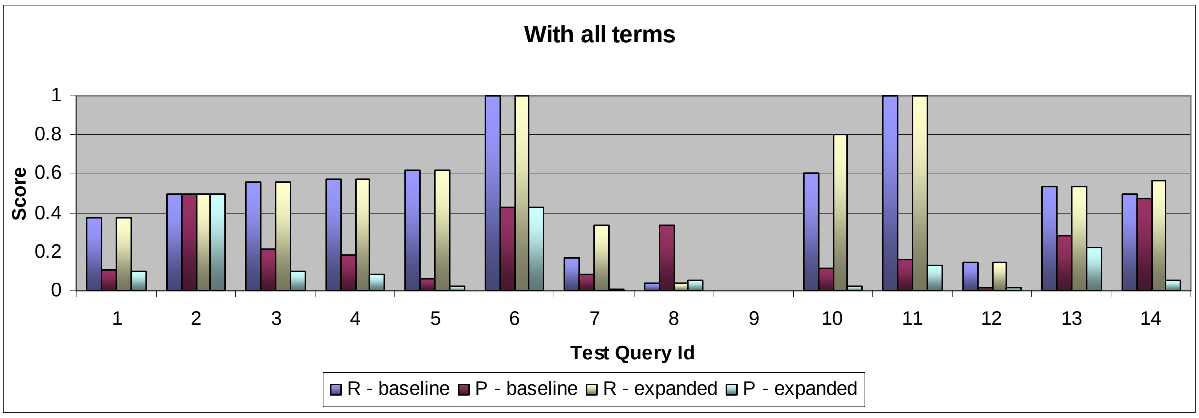


Figure 13 Comparing baseline and all terms results

The above diagram compares the effects of full query expansion (narrower terms up to one deep) with the baseline scenario. Note that R in the legend stands for recall, and P for precision. As discussed in the individual term sections, there was no large change in recall overall, with a few queries seeing some improvements. However, there was a considerable loss in precision in a number of queries (8 and 14 most noticeably) with small losses in most of the other queries. For queries 2 and 6 (potato starch and tomato soup, respectively), none of the term expansions had any effect as the retrieved amount of articles remained exactly the same. Also, no query expansions resulted in relevant articles being found for query 9. The average recall was 0.50 (baseline 0.47), and the average precision was 0.12 (baseline 0.21).

7.3. Narrower term results

7.3.1. Depth 2

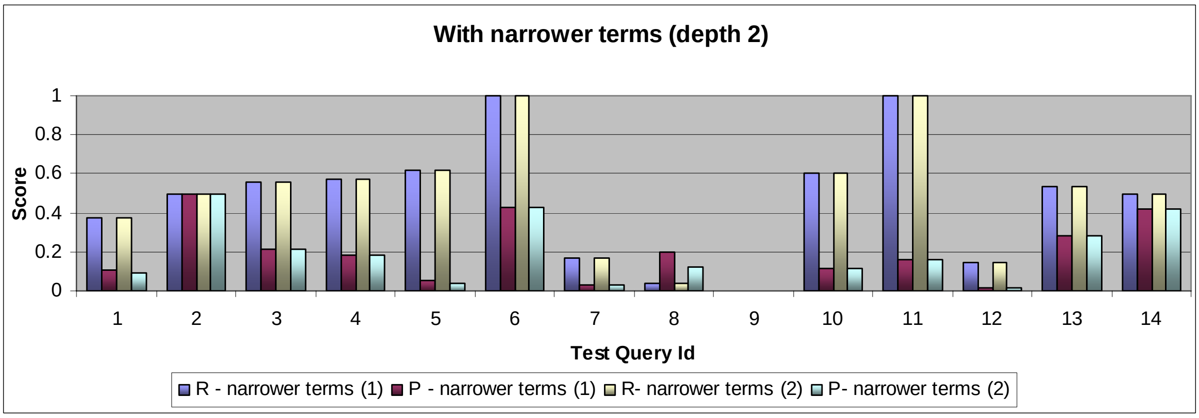


Figure 14 Results with narrower terms up to 2 deep

The figure above compares the results for term expansion with narrower terms 1 deep (also shown in a previous section) with narrower terms of up to 2 deep. The results are almost identical, with no recall improvements and only a minor drop in precision.

7.3.2. Depth 3

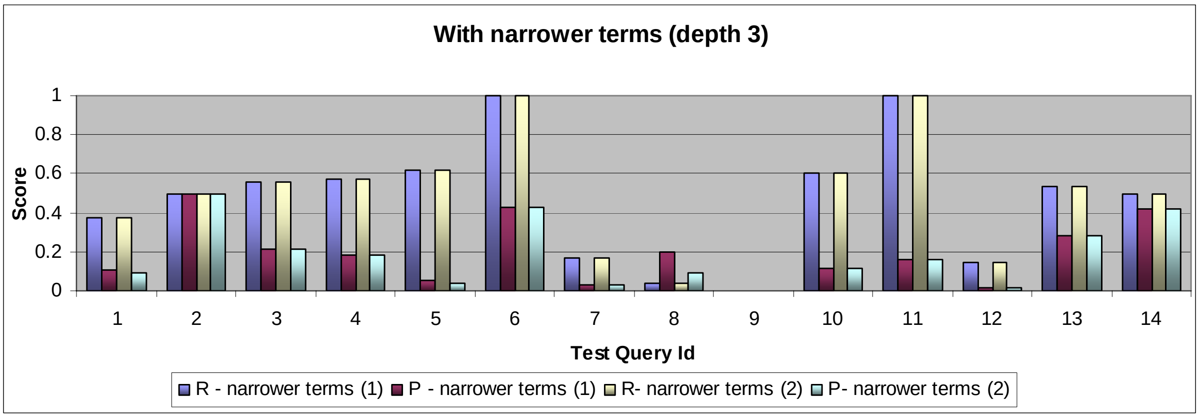


Figure 15 Results with narrower terms up to 3 deep

The additional effect of using three levels of narrower terms is even smaller than that which 2 levels has. Only queries 1, 5 and 8 had additional matches through this expansion, none of which were relevant. The effect this had on average precision was extremely marginal.

7.4. Boolean results

7.4.1. AND operator results

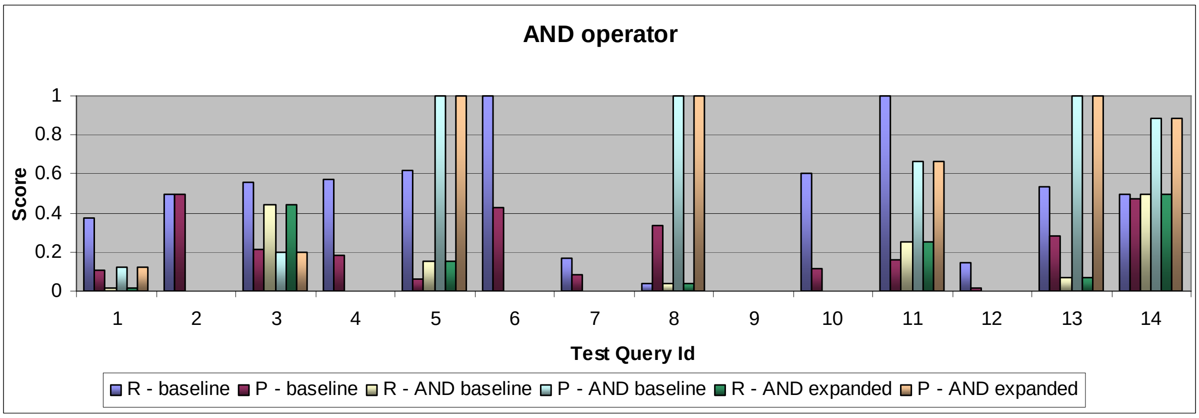


Figure 16 Results with the AND operator

The diagram above contains three recall-precision pairs for each query. These consist of:

1. The baseline scenario as defined in the beginning of this chapter.
2. A baseline version of the AND operator scenario, e.g. dairy AND products.
3. The expanded baseline scenario, where all term types where used to expand the query.

The second baseline scenario was added to be able to better evaluate the effect of query expansion in the context of the AND operator. As can be seen in the diagram, use of the AND operator improves precision, while drastically reducing recall. This is logical, given the restricting nature of the AND operator. However, this resulted in the fact that for some queries, no results were found at all, where they were found with the baseline scenario. When comparing the AND baseline and the AND expanded scenarios, they give 100% equal results. Since the terms are added to the query on an OR basis, the most restricting part is still the original two query words which must be present; hence term expansion effectively does nothing with the presence of an AND operator.

As an example consider this query: Apples AND Oranges OR malus OR "citrus fruits". This query requires that matches contain both the term apples and oranges, whereas the latter two terms are optional. Replacing OR with AND will quickly give no results since the query becomes too restrictive. Another alternative is to use brackets as such: (Apples AND Oranges) OR malus OR "citrus fruits". However, this goes away from the idea of the original AND query, since if the terms Apples and Oranges are not both present, a match would still be found on an article that only contains the term malus. Therefore, this alternative has not been considered further.

7.4.2. Phrase-based results

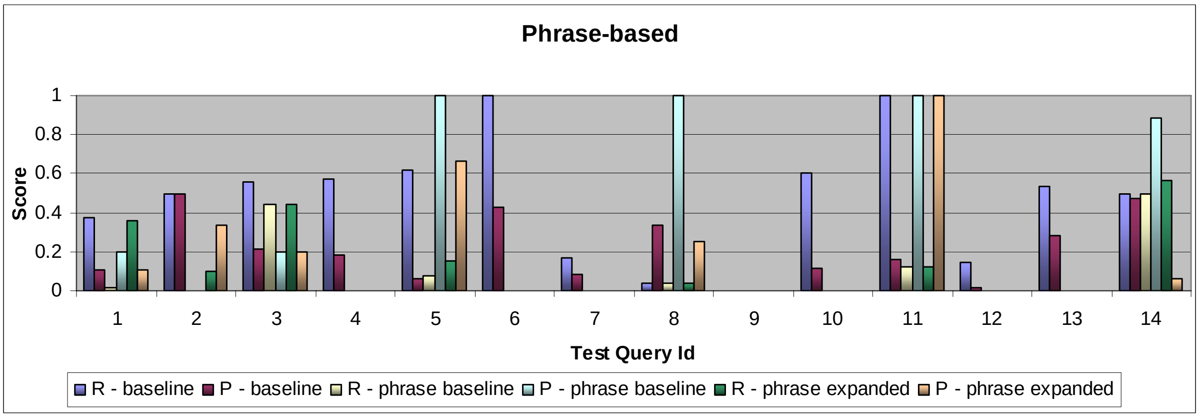


Figure 17 Results with phrase-based queries

For the phrase-based scenarios, all queries were interpreted as phrases, such as "dairy products". This is an even more restrictive setting than the AND operator. As such, precision tended to increase further with recall falling lower, and more queries not returning any results. Comparing the phrase baseline to the phrase expanded scenarios, it can be seen that the expansion actually does have effect here. Many queries showed at least some improvement in recall over phrase baseline, but with precision values often taking a big hit. Also, the phrase-expanded recall values did not exceed the baseline recall values, with the exception of query 14 (this query also suffered a large loss in precision). One remarkable result is query 2: here the phrase baseline did not return any results, yet the phrase-expanded was able to identify a number or relevant articles.

8. Discussion of initial findings

This section discusses the implications of the results obtained from the initial experiment. First the performance of the ontology query enhancement will be evaluated. After that, suggestions for improvement of the enhancement methods will be made.

8.1. Evaluation

The overall results of the initial experiment generally do not show a big improvement in recall by using term expansion. Recall values tended to increase slightly, with most of this improvement unexpectedly coming from related or broader terms. At the same time precision dropped significantly – thus suggesting that the improvement in recall was due to a few lucky hits from the noise that the great number of broader and related terms produced. These results show that the query expansion in its current form is not of much value to enhance recall and precision. By looking through the document collection and the enhanced queries, we've identified a number of possible problem areas.

1. The amount of articles in the database. for most of the queries there are not many relevant articles, limiting the 'room for improvement' which query expansion can have.
2. The presence of articles which are relevant but cannot be identified with term expansion, such as foreign language articles .
3. The implementation of the query expansion algorithm, specifically how terms from the query are mapped to ontology terms to extend. this is currently done on a per-keyword best match, which does not always give good results. E.g. the query dairy products results in ontology terms "dairy byproducts" and "products". Treating the query as a whole, e.g. "dairy products", helps in some cases but in others this makes it worse because as a whole, no matching terms can be found (for example for "eggs protein"). Thus, there does not seem to be a one size fits all solution.

Problems 1 and 2 cannot be influenced within the context of this experiment. However, there is room for improvement with regards to the third problem – the way query expansion is handled.

8.2. Further improvements

To improve the results of query expansion, some changes to the expansion algorithm and the search index file could be made. After implementing these changes, another experiment with the same scenarios could be carried out, to evaluate whether this makes query expansion more effective. The proposed changes are as follows:

1. Implement a stemming filter, so that words are reduced to a common stem. For example, this would make “development”, “developing” and “developer” all match since they reduce to the common stem “develop.” Thus, this will increase matches, and thus recall is expected to increase even without term expansion already.
2. Changing the expansion algorithm to a two-step process, where first a query is tried entirely as a phrase. If there are no direct matches, then the query is broken up in separate keywords to expand. This gives the best of both phrase-based and simple OR-based approaches.

With these proposed changes, the mapping of query keywords to ontology terms is expected to improve. Hence, the potential improvement of query expansion can better be assessed with the implementation of these changes.

9. Follow-up experiment

For the follow-up experiment, the two changes to the query enhancement method suggested in the previous chapter were implemented. No other changes were made; the document set and the gold standard classification remained the same for the follow-up experiment.
To enable stemming, the search indexes had to be stemmed as well. Both the news item index and the ontology index were rebuilt using stemmed keywords. On the input side, incoming user queries are stemmed as well. Both changes were necessary to enable comparing stems to stems.

The proposed algorithm change was also realized. Thus, queries such as: dairy products are treated as a phrase at first for the purpose of finding matches in the ontology. Only if no match could be found for such a phrase in the ontology, would the query terms be expanded individually. This process is only applicable to searching the ontology for matching terms to base the expansion on. The search process through the news items remains unchanged, so the query: dairy products would be treated as two separate terms, unless quotes are explicitly added by the user.

The rest of this chapter deals with the results of the scenario's in the same order as the original experiment, starting with the establishing of a new baseline for the stemmed search. All precision and recall score results for this experiment are also available in numerical form in the tables found in Appendix B.

9.1. Baseline vs. stemmed performance (unexpanded)

To be able to compare the effect query enhancement has with stemming in place, it is necessary to establish a new baseline. This baseline consists of the unexpanded query, but with the use of stemmed search indexes and stemmed query keywords. In the graph below, the performance of the stemmed baseline is contrasted against the normal baseline as established in the previous chapter.

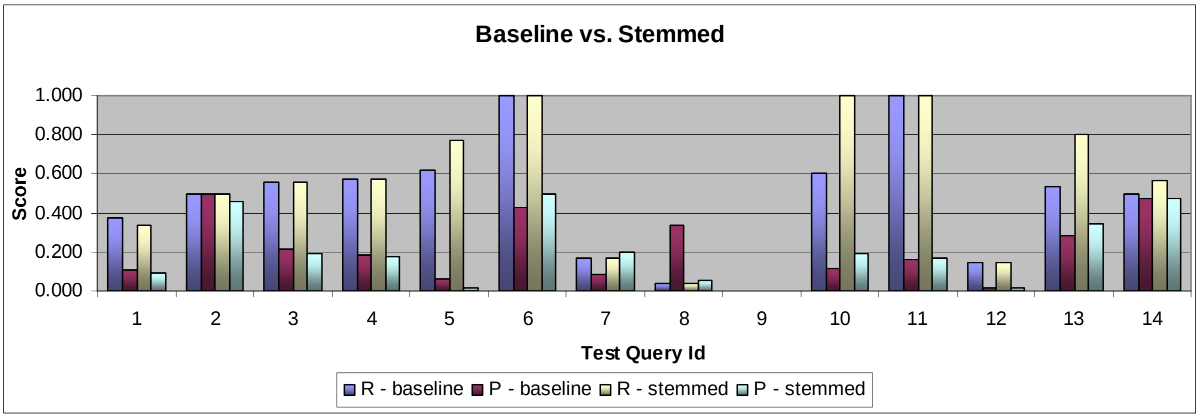


Figure 18 Baseline vs. stemmed performance

The figure shows a noticeable overall increase in recall for the stemmed search compared to baseline. Queries 5 (baby foods), 10 (eggs protein) and 13 (chocolate mousse) showed the largest increase while most others had a smaller increase in recall. The exception was query 1 (dairy products) with a small decrease. Precision values fluctuated somewhat; some queries suffered a small decrease, while others gained a small increase in precision through stemming. Query 8 (fatty acids) showed a large decrease in precision. This is due to a large number of articles about different types of acids from a scientific source, which contained the term acid (in singular), and thus were not found without stemming.
Overall though, stemming has a noticeable positive effect on recall, without a large cost to precision, in some cases even improving precision scores. The net effect was that average precision remained the same at 0.21, whereas recall increased to 0.53 from the baseline 0.47.

9.2. Term type results

As with the first experiment, each term's results will be shown individually, followed by an overview of the results of all terms together.

9.2.1. Broader terms individually

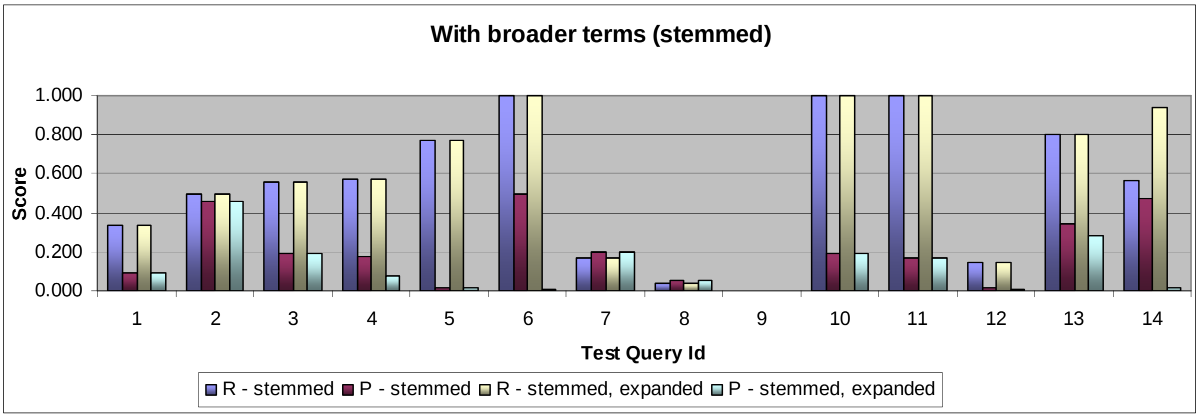


Figure 19 Results with broader terms (stemmed)

Recall values did not change much overall with broader terms. Again, query 14 benefited most from the expansion, this time reaching well into 90% recall. However, precision dropped to 1%, making the actual value of this increase doubtful. Query 6 (tomato soup) also suffered a large precision loss. For both queries, the cause was expansion with the term “foods”, which caused an explosion in the amount of hits retrieved. Most of the other queries had a small drop in precision, thus a few more articles were found that were not judged relevant.

9.2.2. Narrower terms individually

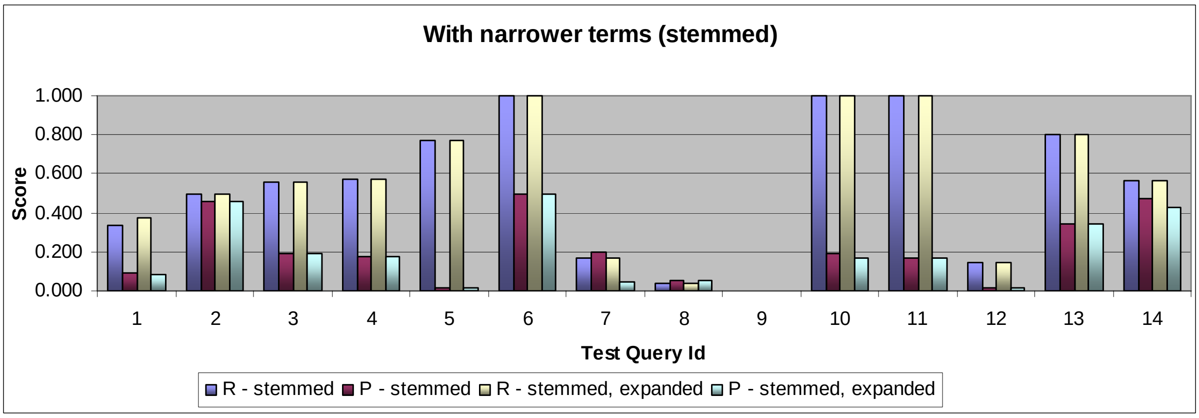


Figure 20 Results with narrower terms (stemmed)

The narrower term expansion showed little difference with the baseline stemmed results. As in the first experiment, a small decrease in precision was found. Unlike the first experiment, there was an increase in recall, but only for the first query, the others remained stable.

9.2.3. Synonyms individually

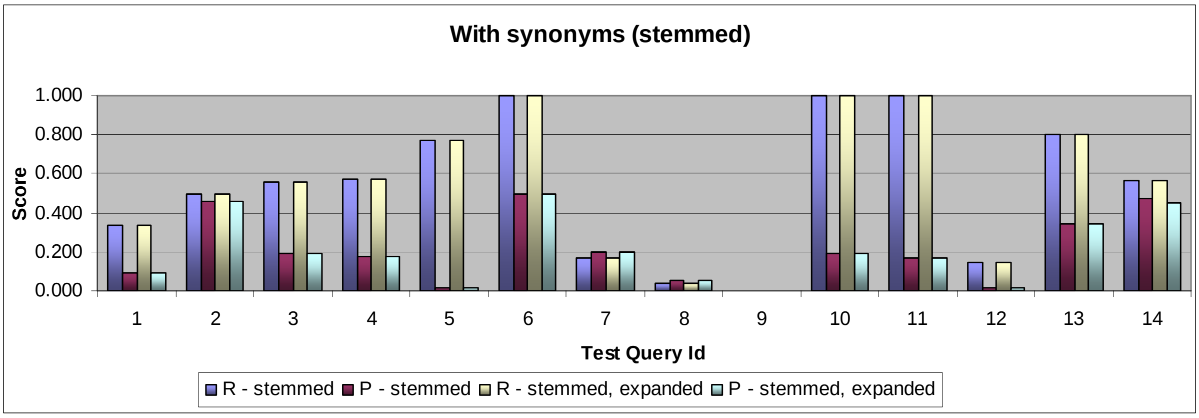


Figure 21 Results with synonyms (stemmed)

Results with synonyms were almost identical to the stemmed baseline. The same was the case in the first experiment, so stemming made little difference to the effect of synonyms. In almost all cases, synonyms did not lead to additional search results except for query 14 (ready meals), where one extra article was found. The article was non-relevant, however, it did consider a convenience food firm: this demonstrates the difficulty with evaluating relevance; as some might consider this relevant for the given information need. This point will be elaborated upon further in the conclusion chapter.

9.2.4. Related terms individually

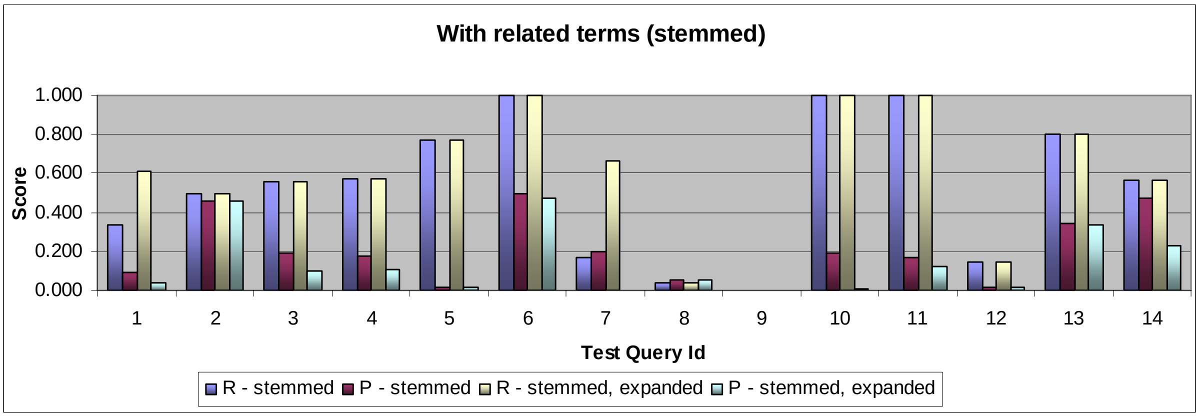


Figure 22 Results with related terms (stemmed)

As with the first experiment, the use of related terms shows the biggest change in results: drastic increases in recall accompanied by sharp decreases in recall; although for several queries the expansion had little effect (typically the queries for which few terms of any type were available in the ontology.)

9.2.5. All terms (stemmed)

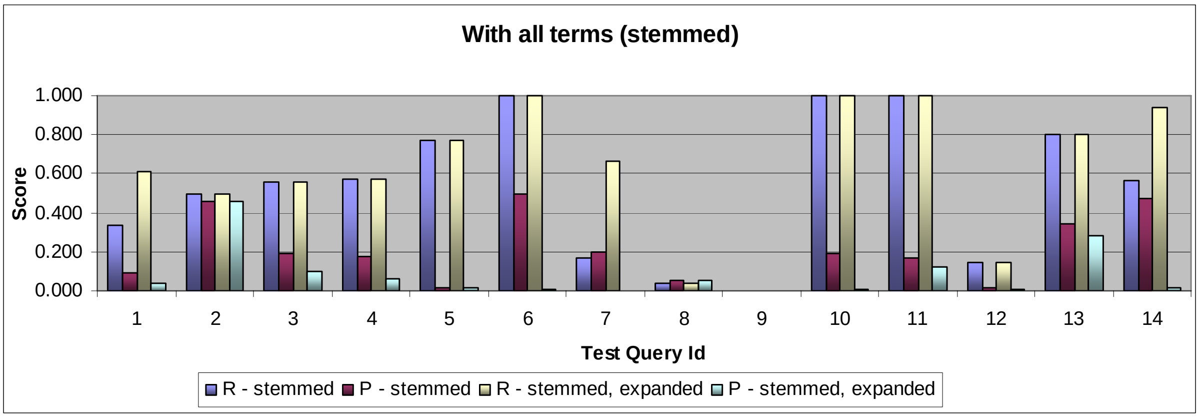


Figure 23 Results with all terms (stemmed)

Expanding with all available terms gives the above results. The trends are similar to what was observed during the first experiment. Recall increased from 0.53 to 0.61 – this is a larger increase than full expansion provided in the original experiment (0.47 to 0.50). Precision dropped by a large margin again, from 0.21 to 0.08. This loss was also greater than in the first experiment (0.21 to 0.12). Thus, while recall increased noticeably, it is still hard to justify a full automatic expansion, given the low average precision.

9.3. *Narrower term results (stemmed)*

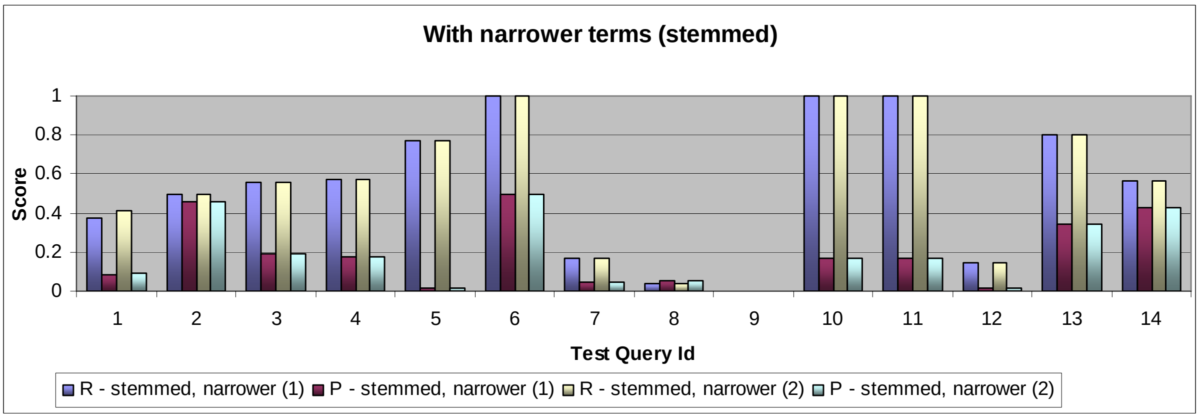


Figure 24 Results with narrower terms (stemmed)

The baseline in this picture is the results with stemmed narrower terms up to depth 1. The results for including narrower terms up to depth 2 are identical, except for query 1 (dairy products), which showed a small increase in recall and negligible (less than 1%) increase in precision. When including terms up to depth 3, no changes were found at all, unlike in the first experiment. The cause is that the additional articles were found already in this case through the use of stemming.

9.4. *Phrase-based results*

Note that the Boolean AND scenario was left out for the second experiment, because the first experiment showed that query expansion could never provide more results, because of the fundamental characteristics of the AND operator. The results for phrase-based queries, with the use of stemming, were as follows.

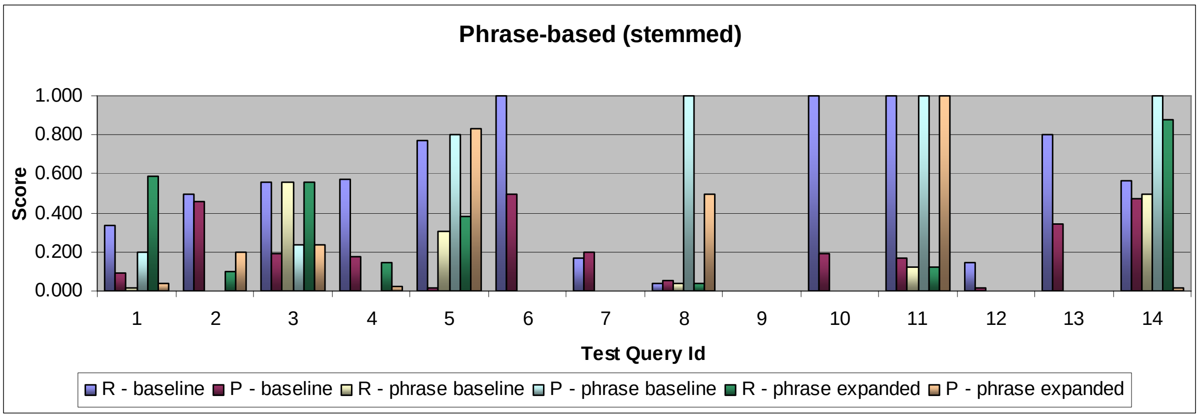


Figure 25 Phrase-based results (stemmed)

It is difficult to find trends from the phrase-based results. For some queries, precision increases and recall decreases with phrase-based searches. In other cases it is just the opposite. The expanded phrase is often less accurate than the baseline phrase, in some case even less accurate than the non-phrase baseline. The latter happens in the case of ‘result explosion’ when very generic terms get used in the expansion. As was observed in the first experiment, expanded phrases can get results in some cases when phrase baseline cannot retrieve anything.

One thing that becomes clear from these results is that there is no one size fits all approach to searching. Some information needs and queries lend themselves better to be formulated as a phrase, while others are better left as two separate terms (e.g. two terms that do not actually form a phrase, such as query 10, “eggs protein”.) Thus, there remains a need for expertise and judgement from the user as to how to obtain the best results for his particular informational need.

10. Conclusion and discussion

In this chapter, the findings from the research are summarized and discussed. First, the main research question is addressed, after which the results for each sub-question will be reviewed in more detail. Finally, the implications of these results are discussed from the business perspective.

10.1. Main research question

Which type of term relationships should be used to expand queries to ensure the greatest increase in recall, at the smallest cost to precision?

From the results obtained in this research, no clear ‘win-win’ combination of terms could be identified. The overall performance for expanded queries was comparable to baseline searches, although notably worse in a few cases where ‘term explosion’ resulted in precision dropping to near zero. Thus the F-measure (harmonic mean of precision and recall) was actually lower for expanded queries than for the baseline. These results are comparable to the CIRI project (Suomela and Kekäläinen, 2006), where the baseline queries were found more effective. A side note here is that while CIRI is nearly identical to this research in its domain (food industry news articles), the ontology approach was slightly different. In their ontology-enhanced search, ontology concepts were selected by users from a tree, rather than added to user-entered keywords as with FORCA. Nevertheless, the similarity in results is worth noting.

Returning to the topic of term types, broader terms and related terms were found to be not suitable for automatic expansion. Although they provided a noticeable increase in recall, the loss in precision was too big to make using them worthwhile. This result is in agreement with the results of Greenberg (2001). These terms might still be of use when a user wishes to conduct a broad, orientating search, but should not be part of an automatic query expansion by default.

Synonyms and narrower terms, which Greenberg (2001) reported to be better candidates for automatic expansion, were not found to yield much effect in this research. The absolute number of extra articles found with these term relations was small. In most cases there was no noticeable improvement in recall with these terms, while most of the test queries had a very small drop in precision. Therefore it does appear that narrower terms and synonyms are 'safer' terms to use for automatic expansion. Unlike broader and related terms, the use of these did not cause an 'explosion' of irrelevant results for some queries. Finally, in terms of absolute results, term expansion did result in most cases in at least some additional news items being retrieved.

10.2. Sub-questions in detail

1. *What is the effect on precision and recall if the query is automatically expanded with every relationship type (BT, NT, SYN and RT), compared to a non-expanded query?*

In the first experiment, where no term stemming was used, full expansion resulted in only a 3% increase in recall, while precision decreased with 9% compared to the baseline non-expanded query. The recall increase was quite small in this case, because query terms were not matched to ontology terms well. This resulted in few and inaccurate term expansions. Stemming solved this matching issue. Under the effects of stemming, a full term expansion resulted in a 8% increase in recall, and a 13% decrease in precision, thus both the positive and negative effects of expansion were increased. The overall effect can be given by the F-measure, which is the harmonic mean of recall and precision, thus calculated as follows:

F = 2 * (Precision * Recall) / (Precision + Recall)

For the second experiment with stemming, the baseline F-measure value is:
2 * (0.21 * 0.53) / (0.21 + 0.53) = 0.30. The F-measure for the fully expanded scenario is 2 * (0.08 * 0.61) / (0.08 + 0.61) = 0.14. Since this is a clearly lower score, the overall effect of full automatic expansion on retrieval of relevant documents is negative.

2. *What is the effect on precision and recall if the query is automatically expanded with only synonyms and narrower terms, compared to a non-expanded query?*

In both experiments, expansion with synonyms and narrower terms had very little effect on recall and precision. This observation differs from the research of Greenberg (2001), where synonyms and narrower terms were found to be the best suited for automatic expansion. A possible explanation lies in the scope of this research. Synonyms are relatively sparse compared to other term relations, with a single ontology term typically having two or three synonyms at most. Many terms that were used in the queries had no synonyms listed at all. Narrower terms were more common, but still did not have much effect.

3. *What is the effect on precision and recall if the query is automatically expanded with only broader terms and related terms, compared to a non-expanded query?*

Compared to other term types, broader and related terms had a larger effect on precision and recall. Of theses two types, broader terms had a smaller overall effect, only noticeably increasing the recall of one of the fourteen queries (no. 14). On the other hand, precision dropped sharply for several other queries as well with broader terms. As for related terms, these showed great increase in recall for queries 1 and 7, of upwards to 40%. However, precision values were down to 5% and below in these cases, as well as falling in about half of all the queries, even the ones were recall was not significantly increased. It appears that there are a few good expansion terms mixed in with broader and related terms. However, the presence of wrong or vague terms (such as foods as a related term for milk, where it should obviously be a broader term several levels higher) makes automatic expansion with these term types unbeneficial.

4. *What is the effect on precision and recall of using a greater depth of narrower terms compared to expansion with only direct narrower terms?*

A greater depth of narrower terms did not result in significantly different results for precision and recall. The results for query 1 demonstrate that further narrower terms can potentially improve both recall, and precision. However, narrower terms further than one level deep were only present for a minority of the test queries, so no definitive conclusions can be reached from this experiment. Further investigation of this specific aspect requires a different set of queries, with terms that are sufficiently broad that there are several levels of narrower terms available to them.

5. *What is the effect on precision and recall of the use of Boolean operators AND, OR, and phrases in conjunction with query expansion?*

As the OR logic was the default behaviour in this experiment, it applies to all results that were described above in this chapter. The AND operator did not work well in conjunction with query expansion; this is because it is intrinsically opposed to the goal of query expansion. The AND operator aims to limit the amount of results retrieved, while expansion aims to increase this amount. If the standard query terms are linked with AND, and the expanded terms are added tot that with OR, the expanded terms will have no effect at all, because the standard terms must be present in all cases anyway. If all terms, standard and expanded terms were to be linked with AND there will be no or very few results at all. Furthermore, it does not make much sense to require that e.g. all synonyms of a term should be present in a single document.

For phrases, the results varied with the type of query. For queries that were actually a phrase such as “dairy products”, results improved in both recall and precision compared to phrase baseline, as well as to the normal baseline. Exceptions to this occurred when there was an explosion in the number of results because of inclusion of a very generic term (full expansion with all term types was used in the phrase tests.) On the other hand, queries which were not actual phrases such as “eggs protein” showed a worse performance, or did not come up with any results at all. In practice, this will not be an issue since users will not use phrases for such queries. These results do show that automatically treating all queries as phrases (and only as phrases) for searching and expansion is not a good idea. Thus, although automated assistance helps, obtaining the best search results still requires that users create their queries carefully.

10.3. Business implications

In this section, the results of the experiment with the FORCA system are placed into the business context again. The purpose is to evaluate the value that the system has as a whole for Infortellence, as well as for business in general. The basis for all this comes from the business goal of the project, which shall be addressed first:

To aggregate food-industry related news articles and provide a selection of this news tailored to a customer's interest, resulting in more interest in other Infortellence services.

Precision and recall measure one aspect of the accuracy of searches, and thus how well tailored news items can be to a customer's interest. On this aspect, the ontology expansion did not provide the benefit that was expected. However, the system as a whole does provide the intended value. Through the use of profiles, users can quickly get an overview of different information needs that are relevant to them. Also, a news service that integrates articles from many different sources and provide a single interface to them did not yet exist for the food domain. Even though the ontology did not offer better recall / precision when measured with the gold standard, it is still there for users' benefit. Simply seeing some related or alternative terms for their queries can trigger them to search for other concepts, or simply raise interest, although this is hard to measure objectively.

Most importantly, the FORCA system as a whole offers a package that can save users time keeping up to date with developments in the food industry. Instead of visiting a handful of different websites, checking newsletters in the mail, and running some Google searches, they can get all this information on a single website. Because there is value present for these users, they come back, and interest can be generated in the company offering this service, especially when FORCA is further integrated into Infortellence's current corporate website.

Moreover, there is additional value for the business itself, as it can gain valuable insights in what its customers are interested in. It starts from examining the profiles that users are creating. With additional data mining functionality, this can be expanded to recording user searches, and analyzing these to find overall trends, or issues that concern specific users often. Such information is invaluable to better assist customers, when consultancy services, Infortellence's core business come into play. When providing such services, privacy concerns should be closely monitored as well. It is vital to create a detailed privacy policy to explain users what their data is going to be used for, and that it will not be disclosed to third parties. Taking the necessary steps to shield user's personal information from potential abuse is crucial to gain and maintain the users' trust.

In a broader perspective, the FORCA system and methodology can be seen as an information service business model that can be applied to other companies as well. The news aggregation information service can be built just as easily for other industries, provided with some knowledge of that domain. The same applies to the ontology, given that there is one available for the target domain. Because the actual implementation of ontologies is not very standardized yet, some technical changes are likely necessary to make a new ontology work with the existing FORCA system. Since the concept of what an ontology is remains the same, this can be easily addressed. The idea of generating interest through news aggregation ties in most closely with information service-related companies, such as consultancy companies. However, it can provide value to any company interested in attracting more visitors to their corporate website. And after all, learning more about existing or potential customers is fundamental to any company.

11. Future research

There is a number of outstanding issues that could be resolved in a further research project. In general terms, these relate to the gold standard, the document corpus, the ontology, the used queries, the type of metrics and the type of experiment. These issues will be addressed in this order, grouped together where this makes more sense.

11.1. Gold standard and document corpus

A tough issue throughout the research was the question of how to judge relevancy of results. The gold standard approach was chosen, however, this does not take away all subjectivity because building this standard itself is still a subjective affair, even when done by a domain expert. Having another independent domain expert classify the articles, then comparing and contrasting the results with the current classification could give more insight into the generic accuracy of the gold standard. However, this is a costly affair in terms of time required, and there is another issue related to the nature of the document corpus.

The corpus is relatively small and comprised of articles from different sources. As the ontology was not very extensive in many cases, this resulted in missing 'opportunities' for term expansion. This could be resolved either through increasing the corpus size or through using a more extensive domain ontology. At present, we are not aware of such an ontology for the food domain.

Another alternative is to use a different set of documents altogether, to evaluate the effectiveness of the gold standard. There are commercial databases which use teams of experts to classify articles, also specifically for the food domain. The content is both scientific articles and newspaper articles. However, this content does not match 1:1 to the current document corpus. Thus, this would mean using a new corpus with its own gold standard as a basis to run a new set of experiments.

11.2. Domain ontology

The domain ontology used in this research was found to be lacking in some aspects. Some terms did not have a lot of other terms linked to them, even when these would be clear to a person outside the domain. Presence of terms in the wrong places, such as "foods" as a related term for milk, rather than a broader term several levels up in the hierarchy also hampered performance.

An alternative ontology is not trivial to find, however. Constructing a new one specifically for this research is unfeasible, because this poses an immense task. Furthermore, a newly built ontology would not be able to approach the scope of existing ones, unless a large scale long term project was setup specifically for this purpose.

11.3. Queries

The set of 14 queries used in this research has been constructed by a domain expert, without prior knowledge of the structure of the ontology relating to the keywords in these queries. This was done to avoid this knowledge influencing the construction of queries, choosing the most optimal keywords that would give the best matches in the ontology. Such an approach could result in a skewed 'best-case' outcome for the experiment. Nevertheless, it may be interesting to construct queries with knowledge of the ontology structure, to see if this theoretical 'best-case' outcome differs significantly from the current results.

Another possibility is to simply increase the amount of queries. This would go together best with increasing the corpus size as well. In this way, not every query would be optimal with regards to the ontology. However, due to the increased overall number of queries, there will be greater number that have a good number of expansion terms from the ontology.

As noted before, the queries were framed in the context of 'information needs', that aimed to take away some of the ambiguity of short queries and make judgements more consistent. These information needs could be expanded to be more specific, but this would result in a reduction in the amount of matches. Because of the small size of the corpus, and the fact that queries were constructed without prior knowledge of the ontology, this approach was not taken in this research. There was a great chance that many queries would have no or very limited results when using very specific information needs. When using a much larger corpus size and a greater number of queries, this approach becomes more feasible.

11.4. Metrics and type of experiment

The metrics used in this research were the core information retrieval metrics of recall and precision. These are important metrics to measure search accuracy, and should thus always be taken into account. However, they are not perfect as they cannot measure every aspect of information retrieval. In an earlier chapter we already stated that users often prefer a simple system that provides some relevant results over a complex system that gives more complete results. This knowledge was taken into account when constructing the FORCA system, and the reason to focus on automatic query expansion rather than user-assisted semi-automatic expansion.

As can be concluded from the previous chapter, recall and precision do not present the complete picture of the value the FORCA system provides from a business point of view. To gain more insight into this, different experiments with different metrics could be setup, which focus more on the user experience. This would require a number of participants (preferably with some domain experience) to complete certain assignments (such as finding all articles on a topic) with, and without the help of the automatic expansion. Afterwards, a questionnaire could be filled in, in which they report their experience with the system on aspects such as response speed, interface, usefulness of the ontology and others. These experiments should offer further insight in the willingness-to-use and the interest generated by the system. Taking a user-centred approach will be appropriate when expanding the 'information service as a business model' outlined in the previous chapter. The feedback and insights gained from users will prove invaluable in further developing this model.

References

Agrovoc, available online at: http://www.fao.org/aims/aos.jsp.

Archimate, The Power of Enterprise Architecture, available online at: http://www.archimate.org/

Bergamaschi S., Castano, S., Maurizio, V., Benevantano, D., 2001, Semantic integration of heterogeneous information sources, *Data & Knowledge Engineering* 36, pp. 215-249.

Bhogal, J., Macfarlane, A., Smith, P., 2007, A review of ontology based query expansion, *Information Processing & Management* 43 - 4, pp. 866-886

Buccafurri, F., Lax, G., Rosaci, D., Ursino, D., 2006, Dealing with semantic heterogeneity for improving Web usage, *Data & Knowledge Engineering* 58, pp. 436–465.

Communications of the ACM, 2002, Special issue: Ontology applications and design. *Communications of the ACM*, 45(2), 39–65.

Fikes, R., Engelmore, R., Farquhar, A., Pratt, W., 1995, Network-Based Information Brokers, Paper presented at the AAAI Spring Workshop on Information Gathering from Distributed, Heterogeneous Environments.

Food and Agriculture Organization of the United Nations, available online at http://www.fao.org/.

Greenberg, J., 2001, Optimal Query Expansion (QE) Processing Methods with Semantically Encoded Structured Thesauri Terminology, *Journal of the American Society for Information Science and Technology,* 52(6), pp. 487–498.

Gruber, T., 1993, A translational approach to portable ontologies, *Knowledge Acquisition*, (5) 2, pp. 199-220.

Kishore, R., Sharman, R., Ramesh, R., 2004, COMPUTATIONAL ONTOLOGIES AND INFORMATION SYSTEMS: I. FOUNDATIONS, *Communications of the Association for Information Systems* (14), pp. 158-183

Mann, T., 1993, Library research models: A guide to classification, cataloging, and computers. New York: Oxford University Press.

Lucene, Apache Lucene, available online at http://lucene.apache.org/java/docs/index.html

Naumann, F., Leser, U., Freytag, J. C., 1999, Quality-driven integration of heterogeneous information systems, Proceedings of the 25th VLDB conference, Edinburgh, Scotland.

Sheth, A., Kashyap, V., & Lima, T. 1999, Semantic Information Brokering: How Can a Multi-agent Approach Help?, M. Klusch, O. M. Shehory, & G. Weiss (Eds.), LNAI, Vol. 1652, pp. 303-322.

Smith, B. (2003) Ontology and information systems. SUNY at Buffalo, Buffalo, NY.

Suomela, S., Kekäläinen, J., 2006, User evaluation of ontology as query construction tool, *Information Retrieval*, 9, pp. 455–475.

Syntens, 2004, Samenvatting Antennewijzer, Ministerie van Economische Zaken.

Voorhees, Ellen M., Harman, Donna K., 2005, TREC : Experiment and Evaluation in Information Retrieval

WordNet, A lexical database for the English language, Princeton University, available online at http://wordnet.princeton.edu/.

Appendix A: Initial experiment result tables

1. Baseline

testqueryid	precision	recall	no. of results
1	0.107	0.375	196
2	0.500	0.500	10
3	0.217	0.556	23
4	0.182	0.571	22
5	0.063	0.615	128
6	0.429	1.000	21
7	0.083	0.167	12
8	0.333	0.037	3
9	0.000	0.000	160
10	0.111	0.600	27
11	0.163	1.000	49
12	0.018	0.143	57
13	0.286	0.533	28
14	0.471	0.500	17

2. Broader terms

testqueryid	precision	recall	no. of results
1	0.107	0.375	196
2	0.500	0.500	10
3	0.217	0.556	23
4	0.125	0.571	32
5	0.054	0.615	148
6	0.429	1.000	21
7	0.083	0.167	12
8	0.333	0.037	3
9	0.000	0.000	160
10	0.107	0.600	28
11	0.160	1.000	50
12	0.017	0.143	58
13	0.229	0.533	35
14	0.063	0.563	142

3. Narrower terms

testqueryid	precision	recall	no. of results
1	0.106	0.375	198
2	0.500	0.500	10
3	0.217	0.556	23
4	0.182	0.571	22
5	0.050	0.615	160
6	0.429	1.000	21
7	0.031	0.167	32
8	0.200	0.037	5
9	0.000	0.000	162
10	0.111	0.600	27
11	0.163	1.000	49
12	0.018	0.143	57
13	0.286	0.533	28
14	0.421	0.500	19

4. Synonyms

testqueryid	precision	recall	no. of results
1	0.100	0.375	210
2	0.500	0.500	10
3	0.217	0.556	23
4	0.182	0.571	22
5	0.051	0.615	156
6	0.429	1.000	21
7	0.083	0.167	12
8	0.333	0.037	3
9	0.000	0.000	174
10	0.111	0.600	27
11	0.163	1.000	49
12	0.018	0.143	57
13	0.286	0.533	28
14	0.444	0.500	18

5. Related terms

testqueryid	precision	recall	no. of results
1	0.104	0.375	202
2	0.500	0.500	10
3	0.098	0.556	51
4	0.108	0.571	37
5	0.030	0.615	271
6	0.429	1.000	21
7	0.005	0.333	373
8	0.063	0.037	16
9	0.000	0.000	166
10	0.027	0.800	149
11	0.129	1.000	62
12	0.018	0.143	57
13	0.276	0.533	29
14	0.216	0.500	37

6. All terms

testqueryid	precision	recall	no. of results
1	0.096	0.375	218
2	0.500	0.500	10
3	0.098	0.556	51
4	0.087	0.571	46
5	0.025	0.615	316
6	0.429	1.000	21
7	0.005	0.333	392
8	0.056	0.037	18
9	0.000	0.000	182
10	0.027	0.800	150
11	0.127	1.000	63
12	0.017	0.143	58
13	0.222	0.533	36
14	0.056	0.563	160

7. Narrower terms, depth 2

testqueryid	precision	recall	no. of results
1	0.092	0.375	228
2	0.500	0.500	10
3	0.217	0.556	23
4	0.182	0.571	22
5	0.037	0.615	215
6	0.429	1.000	21
7	0.031	0.167	32
8	0.125	0.037	8
9	0.000	0.000	193
10	0.111	0.600	27
11	0.163	1.000	49
12	0.018	0.143	57
13	0.286	0.533	28
14	0.421	0.500	19

8. Narrower terms, depth 3

testqueryid	precision	recall	no. of results
1	0.088	0.375	239
2	0.500	0.500	10
3	0.217	0.556	23
4	0.182	0.571	22
5	0.037	0.615	218
6	0.429	1.000	21
7	0.031	0.167	32
8	0.091	0.037	11
9	0.000	0.000	204
10	0.111	0.600	27
11	0.163	1.000	49
12	0.018	0.143	57
13	0.286	0.533	28
14	0.421	0.500	19

9. Boolean AND

testqueryid	P - AND baseline	R - AND baseline	P - AND expanded	R - AND expanded	no. of results baseline	no. of results expanded
1	0.125	0.018	0.125	0.018	8	8
2	0.000	0.000	0.000	0.000	0	0
3	0.200	0.444	0.200	0.444	20	20
4	0.000	0.000	0.000	0.000	0	0
5	1.000	0.154	1.000	0.154	2	2
6	0.000	0.000	0.000	0.000	0	0
7	0.000	0.000	0.000	0.000	0	0
8	1.000	0.037	1.000	0.037	1	1
9	0.000	0.000	0.000	0.000	1	1
10	0.000	0.000	0.000	0.000	0	0
11	0.667	0.250	0.667	0.250	3	3
12	0.000	0.000	0.000	0.000	0	0
13	1.000	0.067	1.000	0.067	1	1
14	0.889	0.500	0.889	0.500	9	9

10. Phrase-based

testqueryid	P - phrase baseline	R - phrase baseline	P - phrase expanded	R - phrase expanded	no. of results baseline	no. of results expanded
1	0.200	0.018	0.108	0.357	5	185
2	0.000	0.000	0.333	0.100	0	3
3	0.200	0.444	0.200	0.444	20	20
4	0.000	0.000	0.000	0.000	0	23
5	1.000	0.077	0.667	0.154	1	3
6	0.000	0.000	0.000	0.000	0	0
7	0.000	0.000	0.000	0.000	0	0
8	1.000	0.037	0.250	0.037	1	4
9	0.000	0.000	0.000	0.000	1	1
10	0.000	0.000	0.000	0.000	0	0
11	1.000	0.125	1.000	0.125	1	1
12	0.000	0.000	0.000	0.000	0	0
13	0.000	0.000	0.000	0.000	0	0
14	0.889	0.500	0.060	0.563	9	149

Appendix B: Followup experiment result tables

1. Baseline (stemmed)

testqueryid	precision	recall	no. of results
1	0.090	0.339	212
2	0.455	0.500	11
3	0.192	0.556	26
4	0.174	0.571	23
5	0.012	0.769	810
6	0.500	1.000	18
7	0.200	0.167	5
8	0.053	0.037	19
9	0.000	0.000	179
10	0.192	1.000	26
11	0.170	1.000	47
12	0.019	0.143	53
13	0.343	0.800	35
14	0.474	0.563	19

2. Broader terms (stemmed)

testqueryid	precision	recall	no. of results
1	0.090	0.339	212
2	0.455	0.500	11
3	0.192	0.556	26
4	0.073	0.571	55
5	0.012	0.769	810
6	0.011	1.000	821
7	0.200	0.167	5
8	0.053	0.037	19
9	0.000	0.000	187
10	0.192	1.000	26
11	0.170	1.000	47
12	0.011	0.143	89
13	0.286	0.800	42
14	0.018	0.938	815

3. Narrower terms (stemmed)

testqueryid	precision	recall	no. of results
1	0.084	0.375	249
2	0.455	0.500	11
3	0.192	0.556	26
4	0.174	0.571	23
5	0.012	0.769	810
6	0.500	1.000	18
7	0.043	0.167	23
8	0.053	0.037	19
9	0.000	0.000	179
10	0.167	1.000	30
11	0.170	1.000	47
12	0.019	0.143	53
13	0.343	0.800	35
14	0.429	0.563	21

4. Synonyms (stemmed)

testqueryid	precision	recall	no. of results
1	0.090	0.339	212
2	0.455	0.500	11
3	0.192	0.556	26
4	0.174	0.571	23
5	0.012	0.769	810
6	0.500	1.000	18
7	0.200	0.167	5
8	0.053	0.037	19
9	0.000	0.000	179
10	0.192	1.000	26
11	0.170	1.000	47
12	0.019	0.143	53
13	0.343	0.800	35
14	0.450	0.563	20

5. Related terms (stemmed)

testqueryid	precision	recall	no. of results
1	0.039	0.607	882
2	0.455	0.500	11
3	0.098	0.556	51
4	0.108	0.571	37
5	0.012	0.769	810
6	0.474	1.000	19
7	0.004	0.667	1054
8	0.053	0.037	19
9	0.000	0.000	179
10	0.006	1.000	816
11	0.123	1.000	65
12	0.019	0.143	53
13	0.333	0.800	36
14	0.231	0.563	39

6. All terms (stemmed)

testqueryid	precision	recall	no. of results
1	0.038	0.607	884
2	0.455	0.500	11
3	0.098	0.556	51
4	0.059	0.571	68
5	0.012	0.769	810
6	0.011	1.000	821
7	0.004	0.667	1061
8	0.053	0.037	19
9	0.000	0.000	187
10	0.006	1.000	816
11	0.123	1.000	65
12	0.011	0.143	89
13	0.279	0.800	43
14	0.018	0.938	816

7. Narrower terms, depth 2 (stemmed)

testqueryid	precision	recall	no. of results
1	0.092	0.411	251
2	0.455	0.500	11
3	0.192	0.556	26
4	0.174	0.571	23
5	0.012	0.769	810
6	0.500	1.000	18
7	0.043	0.167	5
8	0.053	0.037	19
9	0.000	0.000	179
10	0.167	1.000	30
11	0.170	1.000	47
12	0.019	0.143	53
13	0.343	0.800	35
14	0.429	0.563	21

8. Phrase-based (stemmed)

testqueryid	P - phrase baseline	R - phrase baseline	P - phrase expanded	R - phrase expanded	no. of results baseline	no. of results expanded
1	0.200	0.018	0.040	0.589	5	818
2	0.000	0.000	0.200	0.100	0	5
3	0.238	0.556	0.238	0.556	21	21
4	0.000	0.000	0.021	0.143	0	48
5	0.800	0.308	0.833	0.385	5	6
6	0.000	0.000	0.000	0.000	0	0
7	0.000	0.000	0.000	0.000	0	0
8	1.000	0.037	0.500	0.037	1	2
9	0.000	0.000	0.000	0.000	0	0
10	0.000	0.000	0.000	0.000	0	0
11	1.000	0.125	1.000	0.125	1	1
12	0.000	0.000	0.000	0.000	0	0
13	0.000	0.000	0.000	0.000	0	0
14	1.000	0.500	0.017	0.875	8	810