UNIVERSITAT JAUME I
DPTO. DE LENGUAJES Y SISTEMAS INFORMÁTICOS

Contextualizing a Data Warehouse with Documents

Ph. D. Thesis
Presented by Juan Manuel Pérez Mártinez
Supervised by Dr. Rafael Berlanga Llavori
and Dra. María José Aramburu Cabo

Castellón, February 2007

Contextualizando un Almacén de Datos con Documentos

Juan Manuel Pérez Martínez

Trabajo realizado bajo la dirección los Doctores
Rafael Berlanga Llavori y María José Aramburu Cabo,
presentado en la Universitat Jaume I
para optar al grado de Doctor por la Universitat Jaume I

Castellón, Febrero de 2007

Este trabajo ha sido parcialmente realizado durante los proyectos TIC2002–04586–C04–03 y TIN2005–09098–C05–04, subvencionados por la CICYT y P1B2004–30, subvencionado por UJI-Fundació Bancaixa.

To my parents, sister and brother, and my beloved aunts.

To Maria.

Aportaciones, Conclusiones y Trabajo Futuro

Resumen

La tecnología actual de los almacenes de datos y las técnicas OLAP permite a las organizaciones analizar los datos estructurados que éstas recopilan en sus bases de datos. Las circunstancias que rodean a estos datos aparecen descritas en documentos, típicamente ricos en texto. Algunos de estos documentos provienen de fuentes internas a la organización, como por ejemplo, un informe sobre un nuevo proceso de producción. Muchos otros son de origen externo, por ejemplo estudios de mercado publicados en línea. Hoy en día, la Web se ha convertido en una de las mayores fuentes sobre el entorno de las organizaciones.

La información sobre el contexto de los datos registrados en el almacén es muy valiosa, ya que nos permite interpretar los resultados obtenidos en análisis históricos. Por ejemplo, una crisis financiera relatada en una revista digital sobre economía podría explicar una caída de las ventas en una determinada región geográfica. Sin embargo, no es posible explotar esta información contextual utilizando directamente las herramientas OLAP tradicionales. La principal causa es la naturaleza no-estructurada, rica en texto, de los documentos que recogen dicha información. Actualmente es posible encontrar muchos de estos documentos en formato XML.

Esta tesis propone integrar un almacén corporativo de datos estructurados, con un almacén de documentos XML ricos en texto. El resultado es un *almacén contextualizado*: un nuevo tipo de sistema de apoyo a la decisión que permite combinar las fuentes de información estructurada y de documentos de una organización, y analizar estos datos integrados bajo distintos contextos. Así, en un almacén con noticias sobre economía podremos recuperar los documentos que relatan una crisis y estudiar las ventas registradas en el almacén corporativo que se efectuaron en las regiones y periodos citados en estos documentos. Es

decir, seremos capaces de analizar la evolución de nuestras ventas en el contexto de una crisis y averiguar, por ejemplo, qué productos fueron los más afectados.

Dado que el almacén puede albergar documentos sobre temas muy distintos, aplicaremos técnicas de recuperación de la información para seleccionar los documentos que constituirán el contexto de análisis. De este modo, el usuario especificará un contexto de análisis proporcionando una secuencia de palabras clave (por ejemplo, “crisis financiera”). El análisis se realizará en un nuevo tipo de cubo multidimensional, denominado *R-cubo*, que se materializará recuperando los documentos y hechos relacionados con el contexto de análisis. Cada hecho de un *R-cubo* tendrá asociado el conjunto de documentos que describen su contexto. Adicionalmente, asignaremos a cada hecho un valor numérico que representará su relevancia con respecto al contexto de análisis (es decir, cuán importante es el hecho para una “crisis financiera”). De este modo, los *R-cubos* contienen dos dimensiones especiales: la dimensión de contexto y la de relevancia; de ahí el nombre *R-cubo* (cubo Relevancia).

Esta tesis realiza una revisión de los trabajos en los que se combina la tecnología de los almacenes de datos y la Web. La tesis presenta la arquitectura de un almacén contextualizado. En ésta se propone un nuevo modelo de recuperación para medir la relevancia de los hechos con respecto a un contexto de análisis. El modelo se basa en técnicas de modelado de relevancia. Asimismo, se definen formalmente los *R-cubos*, proporcionando su modelo de datos y álgebra. Finalmente, presentamos un prototipo junto a varios casos de uso que demuestran la utilidad de nuestra aproximación.

El trabajo de esta tesis puede continuarse siguiendo distintas direcciones. Estas líneas de investigación incluyen: completar el álgebra de los *R-cubos* con operadores binarios, evaluar la eficiencia del prototipo en otros escenarios con colecciones de datos mayores, la aplicación de técnicas de análisis multidimensional para explotar el almacén de documentos XML, o analizar directamente los hechos descritos en los documentos sin contextualizar un cubo corporativo.

Objetivos

Ya en los primeros trabajos sobre almacenes de datos se destaca la importancia de la información sobre el contexto de los datos, a la hora de interpretar el resultado de un análisis histórico. Sin embargo, prácticamente no se encuentra trabajos en la literatura que aborden el uso de este tipo de información en un almacén de datos. La principal razón es que mientras la tecnología tradicional de los almacenes de datos y sus técnicas de análisis, comúnmente conocidas como OLAP (del inglés *On-line Analytical Processing*), han sido desarrolladas para ser aplicadas sobre base de datos estructurados, la naturaleza de los documentos que contienen esta información contextual es no-estructurada, rica en texto.

Actualmente, la Web se ha convertido una de las mayores fuentes sobre el entorno de las organizaciones. Debido a la estandarización del uso de XML para el intercambio de información sobre Internet, cada vez es más fácil encontrar en la Web los documentos publicados en formato XML. Adicionalmente, hoy en día, la mayor parte de las aplicaciones utilizadas en las organizaciones (como por ejemplo, los procesadores de texto u hojas de cálculo) permiten exportar sus datos como documentos XML.

En esta tesis asumimos que la información contextual se encuentra recogida en un almacén de documentos XML ricos en texto. El objetivo de la tesis es proporcionar un marco formal para integrar un almacén de datos tradicional con dicho almacén de documentos XML y analizar los datos junto a su contexto. Llamamos al sistema resultante *almacén contextualizado*.

Metodología

El trabajo de esta tesis combina dos campos de investigación distintos: los almacenes de datos y los sistemas de recuperación de la información.

En la actualidad, la mayor parte de las aplicaciones que precisan gestionar grandes colecciones de documentos ricos en texto, aplican técnicas de recuperación de la información. La tesis propone un nuevo modelo de recuperación de la información para seleccionar los documentos que describen un contexto de análisis y estimar la relevancia de los hechos que se citan en estos documentos. Dicho modelo de recuperación se basa en técnicas de modelado de la relevancia [33], principalmente por dos razones. En primer lugar, las técnicas de modelado de la relevancia proporcionan un marco formal, basado en la teoría de probabilidad, que es adecuado para las operaciones de análisis multidimensional. En segundo lugar, a la hora de calcular la relevancia, estas técnicas consideran conjuntos de documentos, en lugar de un único documento, lo cual es apropiado para representar el contexto de los hechos en el almacén.

Denominamos *R-cubos* a los cubos de análisis de un almacén contextualizado. Los *R-cubos* tienen dos dimensiones especiales, la dimensión de relevancia y la dimensión de contexto. Para definir formalmente los *R-cubos*, la tesis extiende el modelo de datos multidimensional presentado en [62] con estas dos nuevas dimensiones. Adicionalmente, se redefinen los operadores unarios de selección, agregación y proyección del álgebra de dicho modelo para operar los *R-cubos*.

Aportaciones

La tesis realiza una revisión de los trabajos en los que se combina la tecnología de los almacenes de datos y la Web. Estudia las tecnologías XML utilizadas en la actualidad para integrar, recuperar y procesar información en la Web, y cómo estas tecnologías pueden aplicarse en el campo de los almacenes de datos. La tesis aborda la integración de almacenes de datos heterogéneos e introduce el problema de trabajar con datos semi-estructurados y no-estructurados en los almacenes de datos y las técnicas OLAP.

Otra de las aportaciones de la tesis es la defininción del almacén contextualizado. Los principales componentes de su arquitectura son: un almacén de datos corporativo tradicional, un almacén de documentos XML y la herramienta de extracción de hechos. El cometido de la herramienta de extracción de hechos es relacionar cada hecho del almacén corporativo con su contexto. Para ello, esta herramienta busca ocurrencias de los valores de dimensión de los hechos en el contenido de los documentos. A la hora de materializar un *R-cubo*, el usuario indica una condición de recuperación de la información (una secuencia de palabras clave) que establecerá el contexto de análisis; una condición de estructura, para especificar las secciones de los documentos a estudiar; y un conjunto de condiciones MDX, para seleccionar los hechos corporativos a analizar. Los documentos y hechos que satisfacen estas condiciones son recuperados del almacén correspondiente. Cada hecho seleccionado se registra en el *R-cubo* resultado, junto a los documentos que describen su contexto y su índice de relevancia con respecto a la condición de recuperación de la información.

Las dos aportaciones principales de esta tesis son: el nuevo modelo para la recuperación de documentos XML y el cálculo de la relevancia de los hechos, y el modelo de datos y álgebra de los *R-cubos*.

En el modelo de recuperación elegimos la representación tradicional, en forma de árbol, de los documentos XML. Dadas las condiciones de recuperación de la información y estructura, mostramos cómo recuperar los subárboles de los documentos que satisfacen estas condiciones. El modelo asigna un valor de relevancia a cada uno de los subárboles recuperados y los hechos que éstos describen. El proceso de evaluación de consultas propuesto asegura que únicamente se incluirá en el resultado el elemento más relevante de cada subárbol seleccionado. Calculamos la relevancia de un subárbol por la probabilidad de encontrar las palabras clave de la condición de recuperación de la información en las secciones de texto contenidas en dicho subárbol. La relevancia de un hecho se calcula como la probabilidad conjunta de encontrar el hecho y las palabras clave en el conjunto de subárboles relevantes para el contexto de análisis. Una propiedad interesante de esta aproximación es que la suma de los valores de relevancia asignada a los hechos es siempre igual a uno. Como no todas las colecciones de documentos son apropiadas para realizar todos los tipos de análisis, proponemos medir la calidad del resultado sumando los valores de relevancia asignados a los documentos.

Las dimensiones de relevancia y contexto de los *R-cubos* proporcionan al usuario información adicional sobre los hechos del cubo que puede ser de gran utilidad en las tareas de análisis. La dimensión de relevancia permite identificar las partes más importantes o interesantes de un cubo para un determinado contexto de análisis. Por ejemplo, esta dimensión podría utilizarse para detectar el periodo de una crisis política, o las regiones que se encuentran en desarrollo económico. Por otro lado, utilizando la dimensión de contexto en las operaciones de selección podemos restringir el análisis a aquellos hechos descritos en un subconjunto determinado de documentos (por ejemplo, aquellos más relevantes). En esta dimensión se representan los documentos que relatan las circunstancias de cada hecho. En todo momento, el usuario puede seleccionar un hecho del *R-cubo* y explorar estos documentos. En ellos se podría encontrar, por ejemplo, la explicación de una caída en las ventas. El modelo de datos de los *R-cubos* incluye la definición formal de las dimensiones de relevancia y contexto. El índice de relevancia de los hechos tras una operación de selección se incrementa en dos factores. El primero de estos factores representa la perdida de calidad que experimenta el *R-cubo* resultado. El segundo factor mide el incremento relativo de importancia de los hechos seleccionados en los documentos, cuando los hechos descartados por la operación de selección ya no se tienen en cuenta. En el resultado del operador de agregación, cada nuevo hecho representa un conjunto de hechos del *R-cubo* original. Los nuevos hechos se relacionan con aquellos documentos que se encontraban asociados a alguno de los hechos originales del correspondiente conjunto. La relevancia de cada conjunto de hechos se calcula sumando los valores de relevancia de los hechos originales incluidos en este conjunto.

Conclusiones

La revisión realizada en la tesis incluye trabajos dedicados a la construcción de repositorios de datos tomados de la Web. Estos trabajos abordan principalmente el almacenamiento eficiente, procesamiento de consultas, adquisición, control de cambio e integración del esquema de datos Web. También se ha estudiado trabajos orientados al diseño de bases de datos multidimensionales para fuentes de datos XML, y la extensión de técnicas OLAP para analizar datos XML externos al almacén. Estas aproximaciones están orientadas a datos XML altamente estructurados, y no son adecuadas para explotar la información descrita en documentos ricos en texto. Finalmente, encontramos en la literatura algunos trabajos sobre datos no estructurados y OLAP. En éstos se propone basar la construcción de sistemas de recuperación de la información en bases de datos multidimensionales. Sin embargo, estos trabajos no consideran el análisis de la información descrita en el contenido de los documentos. Hasta donde sabemos, no existe una revisión bibliográfica similar sobre almacenes de datos, XML y la Web como la realizada en esta tesis.

La arquitectura del almacén contextualizado propuesta en la tesis ofrece un marco en el que se enriquece el análisis de datos estructurados con la información que los documentos proporcionan sobre el contexto de estos datos. La tesis presenta un prototipo donde se muestra el potencial de este nuevo tipo de sistema de apoyo a la decisión en un escenario real. Concretamente, utilizamos una colección de artículos sobre economía para contextualizar un cubo corporativo que registra una recopilación histórica de los principales índices de mercado del mundo. En este escenario, empleamos los *R-cubos* para analizar en qué medida los sucesos relatados en los artículos afectaron a los distintos mercados. El prototipo valida en la práctica el modelo de datos y el álgebra de los *R-cubos*. Los experimentos realizados demuestran que es posible utilizar tanto las técnicas de extracción de información propuestas en [37, 68], como los metadatos ya incluidos en los propios documentos para identificar valores de dimensión en los documentos. Evaluamos el modelo de recuperación con dos colecciones de documentos distintas: un repositorio de artículos de *El País* y la subcolección TREC WSJ-1990 [27]. Los resultados obtenidos en ambos casos son satisfactorios. En las primeras posiciones del ranking encontramos los hechos más relevantes para los contextos de análisis especificados. Estos experimentos validan la fórmula del cálculo de la relevancia de los hechos. El valor de relevancia asignado permite diferenciar claramente aquellos hechos directamente relacionados con el contexto de análisis, de aquellos menos relevantes.

Del trabajo aquí presentado se derivan varias publicaciones, enumeradas en el último capítulo de la tesis.

Trabajo Futuro

La primera línea de investigación que proponemos es completar el álgebra de los *R-cubos* con operadores binarios. En este caso, dado que los *R-cubos* de entrada pueden representar contextos de análisis distintos, el problema principal estriba en calcular la relevancia de los hechos en el *R-cubo* resultado. Para este propósito puede aplicarse técnicas de fusión de datos [17] que combinen la relevancia de los hechos involucrados. Los operadores unarios propuestos en el álgebra del modelo multidimensional base [62] también permiten operar *R-cubos*. Sin embargo, en este caso, el resultado de estas operaciones ya no es un *R-cubo*, puesto que estos operadores no actualizarán convenientemente las dimensiones de contexto y relevancia. Aun así, dichos operadores pueden utilizarse para realizar interesantes análisis. Por ejemplo, podríamos agrupar por la dimensión de contexto y agregar los hechos descritos en cada documento. Otra cuestión interesante es utilizar los operadores binarios del modelo multidimensional base [62] para combinar un cubo tradicional con un *R-cubo*.

El prototipo presentado en esta tesis demuestra la utilidad potencial de un almacén contextualizado. Sin embargo, todavía es necesario evaluar la eficiencia del sistema con colecciones de datos mayores, así como estudiar estrategias de pre-agregación para los *R-cubos*. En el futuro planeamos desarrollar un prototipo distinto sobre un sistema de base de datos multidimensional comercial. Otro tema de investigación interesante es la integración de estos prototipos con los buscadores web existentes, mejorando de este modo la escalabilidad del sistema y ofreciendo la posibilidad de contextualizar los cubos con documentos en línea de la WWW.

Algunos trabajos [42, 46] proponen basar la implementación de sistemas recuperación de la información en bases de datos multidimensionales. Estos trabajos defienden que las dimensiones de los cubos OLAP proporcionan un mecanismo intuitivo de análisis que puede aplicarse también en el estudio de colecciones de documentos. Adicionalmente, las funciones de agregación optimizadas en las herramientas OLAP pueden utilizarse para evaluar de manera eficiente las fórmulas del cálculo de relevancia de los sistemas de recuperación de la información. Siguiendo una línea similar, en [65] esbozamos una arquitectura alternativa para los almacenes contextualizados. En esta arquitectura el usuario establece el contexto de análisis consultando un cubo OLAP que representa la colección de documentos. Las condiciones de recuperación de la información se especifican navegando en una dimensión de conceptos que clasifica los documentos por temáticas. En paralelo, los hechos descritos en los documentos seleccionados son analizados en un *R-cubo*. Actualmente estamos construyendo un prototipo de este sistema.

En esta tesis usamos los valores de dimensión encontrados en un documento para relacionar dicho documento con los hechos corporativos caracterizados por los mismos valores de dimensión. Tratar de analizar directamente los hechos extraídos de los documentos, sin considerar los hechos corporativos correspondientes, es un objetivo aún más ambicioso. En este caso, en los análisis aparecerán hechos incompletos (cuando los documentos no mencionan todas las dimensiones) o imprecisos (si los valores de dimensión citados en los documentos no pertenecen al nivel base de la jerarquía). El modelo multidimensional en el que se basan los *R-cubos* soporta hechos incompletos e imprecisos [62]. En un futuro planeamos explotar estas características para analizar los hechos descritos en los documentos que no se encuentran disponibles en el almacén corporativo.

Abstract

Current data warehouse and OLAP technologies are applied to analyze the structured data that companies store in databases. The context that helps to understand this data over time (e.g., the explanation of a sales fall) is usually described separately in text-rich documents. Some of these documents are self-produced company internal documents (e.g., technology reports), whereas some of them are available on the Web (e.g., a market-research article in a business journal). Although these documents include highly valuable information that should also be exploited by companies, they cannot be analyzed by current OLAP technologies because they are unstructured and mainly contain text. The current trend is to find these documents available in XML-like formats.

This thesis presents a framework for the integration of a corporate warehouse of structured data with a warehouse of text-rich XML documents, resulting in what we call a *contextualized warehouse*. A contextualized warehouse is a new kind of decision support system that allows users to obtain strategic information by combining all their sources of structured data and documents, and by analyzing the corporate integrated data under different contexts. For example, if we have a document warehouse with business news articles, we can analyze the evolution of the sales measures stored in our corporate warehouse in the context of a period of crisis as described by the relevant news. Thus, we could detect which products have been more affected.

Since the XML document warehouse may contain documents about many different topics, we apply Information Retrieval techniques to select the context of analysis from the document warehouse. First, the user specifies an analysis context by supplying a sequence of keywords (e.g., an IR condition like “financial crisis”). Then, the analysis is performed on a so-called *R-cube*, which is materialized by retrieving the documents and facts related to the selected context. Each fact in the *R-cube* will be linked to the set of documents that describe its context, and will have assigned a numerical value representing its relevance with respect to the specified context (e.g., how important the fact is for a “financial crisis”). Then, *R-cubes* will have two special dimensions: the context and the relevance dimension, thereby the name *R-cube* (Relevance cube).

This thesis surveys the most relevant research on combining data warehouses and Web data. The thesis presents an architecture for the contextualized warehouse. A novel retrieval model to measure the relevance of the facts for a selected analysis context is also proposed. This retrieval model is based on relevance modeling techniques. Afterwards, *R-cubes* are provided with a data model and an algebra by extending an existing multidimensional model to regard the relevance and context of the facts. Finally, we also present a prototype *R-cube* system, and some usage cases that show the usefulness of the approach.

Keywords: OLAP, text-rich XML documents, Information Retrieval

Acknowledgments

I had never imagined myself as a scientist. A sum of beautiful circumstances have brought me to the research world, and afterwards to write a thesis.

The thrill started during my last year of Computer Science studies at Universitat Jaume I of Castelló. I had to work on my final degree project and I found Rafael Berlanga and María José Aramburu as my advisors. They introduced me in the research world by involving me in the projects they had at the Temporal Knowledge Bases Group (TKBG). Then, I finished my degree and it was time to find a job. I knew that the University career was a long hard one, but I liked too much what I had done in my final degree project. I wanted to do research. Bancaixa and the Conselleria de Cultura i Educació of the Generalitat Valenciana financed my first three years of doctorate studies. During this time I participated in the projects CICYT TEL 97–1119 and UJI-Fundació Bancaixa PI.1B2000–14 of the TKBG under the supervision of Rafa and María José, who at this time had become my thesis advisors and very good friends. From Rafa and María José I have learned how to conduct quality research. The material presented here came out after fruitful discussions with them. Thanks a lot!

After the third year, I became a member of the Languages and Systems Department of Universitat Jaume I. Here, I should not forget to thank my department colleagues for all their patience and support. I would also like to thank Universitat Jaume I for several travel grants that allowed me to present the progress of my research in some European conferences. During this period I also received financial support from the projects CICYT TIC2002–04586–C04–0, UJI-Fundació Bancaixa P1B2004–30 and CICYT TIN2005–09098–C05–04 of the TKBG.

The work of this thesis could not have been completed without the collaboration of Torben Bach Pedersen. I did two stays during my doctoral studies at the Database and Programming Technologies group of Aalborg University under the supervision of Torben. The result of these stays was a series of publications that Rafa, María José and me co-authored with Torben. These papers are directly related to the work of this PhD dissertation. Torben, thanks a lot for your always interesting comments and collaboration!

I would also like to thank Michael Gould, David Losada, Torben Bach Pedersen, José Samos and Juan Carlos Trujillo for reviewing this PhD thesis and being part of the jury.

The friendship my colleagues at the TKBG offer me has been indispensable for the every day work. My thanks to Lola, Isma, Jordi, Ernesto, Aurora, Henri, Victoria, and of course to my office-mate Roxana. From 2002 the TKBG has been actively collaborating with an industrial partner, Hélide Technologies. I have also found a lot of support from people at Hélide Technologies.

My friends at Puerto de Sagunto and Castellón have been very important too. The list would be too long if I included all your names here. You are great!

Finally, I should not forget to thank my family, specially my parents Ani and Manolo, my sister Ana, my brother Javier, my beloved aunts Marta and Juani, and Maria.

Juanma

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This first chapter introduces general concepts like "Data Warehousing" (DW), "On-Line Analytical Processing" (OLAP) and "Information Retrieval" (IR) which constitute the research framework of this thesis. Afterwards, the motivation and objectives of the work are presented. The chapter concludes with the organization of the rest of the thesis.

1.1 General Concepts

In the last years, there has been a great deal of interest in both the industry and research communities regarding the Data Warehouse and OLAP technologies. Three of the pioneers in the field were W. H. Inmon, R. Kimball and E. F. Codd.

The classic definition of "Data Warehouse" (DW) by William Inmon states that a DW is a subject-oriented, integrated, non-volatile, and time variant collection of data in support of management's decisions [29]. Another, widely-accepted definition of DW is the one by Ralph Kimball who defined a DW as a copy of transaction data specifically structured for query and analysis [32]. Thus, Data Warehousing involves the construction of a huge repository where an integrated view of data is given, and which is optimized for analysis purposes. The main problems addressed by the DW technology, which make a DW different from traditional transactional database systems are surveyed in [86].

The information stored in a Data Warehouse is usually exploited by "Online Analytical Processing" (OLAP) tools. The term OLAP was first coined by E. F. Codd. In [16], E.F. Codd presented twelve rules to evaluate OLAP systems and emphasized the main characteristic of OLAP: the multidimensional analysis. Thus, OLAP tools rely on a multidimensional view of data, where data is divided into facts, the central entities/events for the desired analysis, e.g., a sale, and hierarchical dimensions, which provide contextual information for the facts, e.g., the products sold and the grouping of products into categories. Typically, the facts have associated numerical measures (e.g., profit) and queries aggregate fact measure values up to a certain level, e.g., total profit by product category and month, followed by either roll-up (further aggregation, e.g., to year), or drill-down (getting more detail, e.g., looking at profit per day) operations. An overview of multidimensional databases and their importance in Data Warehousing and OLAP is given in [61]. A survey on multidimensional modeling can be found in [11]. The work presented in [62] and [3] also classifies and compares the properties of different multidimensional models.

Information Retrieval (IR) [7] is playing an important role in the Web, since it has enabled the development of useful discovery tools (e.g., web search engines) and digital library services. These applications deal with huge amounts of text-rich documents and have successfully applied IR techniques to query this type of repositories. In an IR system the users describe their information needs by supplying a sequence of keywords. The result is a set of documents ranked by relevance. The relevance is a numerical value which measures how well the document fits the user information needs. Traditional IR models (e.g., the vector model [75]) calculate this relevance value by considering the local and global frequency (tf-idf) of the query keywords in the document and the collection, respectively. Intuitively, a document will be relevant to the query if the specified keywords appear frequently in its textual contents and they are not frequent in the collection. Newer proposals in the field of IR include language modeling [69] and relevance modeling [33] techniques. The work on language modeling considers each document as a language model. Thus, documents are ranked according to the probability of obtaining the query keywords when randomly sampling from the respective language model. The relevance modeling approaches estimate the joint probability of the query's keywords and the document words over the set of documents deemed relevant for that query. The language and relevance modeling approaches still internally apply the keyword frequency to estimate probabilities, and they have been shown to outperform baseline tf-idf models in many cases [69, 33].

This thesis combines DW and OLAP technologies with IR techniques, and studies how DW and OLAP can benefit from IR, and vice-versa, how DW and OLAP can help IR.

1.2 Motivation and Objectives

Current DW and OLAP technologies can be efficiently applied to analyze the huge amounts of structured data that companies store in their databases. The context associated with this data (e.g., the explanation of a sales fall) can be found in other internal and external sources of documents.

Traditional DW and OLAP techniques have mainly focused on dealing with structured data. However, the organizations also produce many documents. Furthermore, nowadays the Web has become the company largest source of external information. Examples of internal and external sources of information include the following: purchase-trends and market-research reports, demographic and credit reports, popular business journals, industry newsletters, technology reports, etc. Although these documents cannot be analyzed by current OLAP technologies mainly because they are unstructured and contain a large amount of text, they include highly valuable information that should also be exploited by companies. The current trend is to find these documents available in XML-like formats [87]. Moreover, existing XML tagging techniques (e.g., [77]) can be applied to give some structure to plain documents by identifying the visual styles that characterize the different document sections. OLAP technologies have also difficulties to deal with binary data, e.g., the graphs and charts associated with these documents. XML documents can reference this binary files by means of links.

The proposal of this dissertation is to build XML document warehouses that can be used by companies to store unstructured information coming from their internal and external sources. This thesis presents an architecture for the integration of a corporate warehouse of structured data with a warehouse of text-rich XML documents. We call the resulting warehouse a *contextualized warehouse*. A contextualized warehouse is a new kind of decision support system that allows users to obtain strategic information by combining all their sources of structured data and unstructured documents, and by analyzing the integrated data under different contexts. For example, if we have a document warehouse with business news articles, we can analyze the evolution of the sales measures stored in our corporate warehouse in the context of a period of crisis as described by the relevant news. Thus, we could detect which products have been more affected. The same set of facts could be less revealing under a different context (e.g., regions in economical development).

The applications described above require both the availability of a document warehouse and its cooperation with the corporate data warehouse. The circumstances of the original facts are understood by analyzing their contexts, that is, the information available in the documents related to the facts. In this thesis, a context is defined as a *set of textual fragments that can provide analysts with strategic information important for decision-making tasks.* Contexts are thus unstructured, and cannot be managed by the well-structured corporate warehouse. Since the document warehouse may contain documents about many different topics, we apply well-known IR techniques to select the context of analysis from the document warehouse.

In order to build a contextualized OLAP cube, the analyst will specify the context under analysis by supplying a sequence of keywords. Each fact in the resulting cube will have a numerical value representing its relevance with respect to the specified context, thereby its name, *R-cube* (Relevance cube). A novel retrieval model to measure the relevance of the facts is proposed in this thesis. Moreover, each fact in the *R-cube* will be related to its context (i.e., the set of the relevant documents that describe the context of the fact).

This thesis extends an existing multi-dimensional data model [62] to represent these two new dimensions (relevance and context), and we study how the traditional OLAP operations affect them.

The relevance and context dimensions provide us further information about facts that can be very useful for analysis tasks. From the user point of view, the relevance dimension can be used to explore the most relevant portions of an *R-cube*. For example, it can be used to identify the period of a political crisis, or the regions under economical development. The usefulness of the context dimension is twofold. First, it can be used in the selection operations to restrict the analysis to the facts described in a given subset of the documents (e.g., the most relevant documents). Second, the user will be able to gain insight into the circumstances of a fact by retrieving its related documents. The graphs, charts, and other binary files possibly linked in the documents would also be presented to the user, easing the understanding of the analysis context.

1.3 Main Contributions

The main contributions of this dissertation are:

- In the last years, lots of work have been devoted to both DW and Web technologies. This thesis surveys the most relevant research on combining these two technologies.

- In [29] the importance of external contextual information to understand the results of historical analysis was emphasized. However, few approaches regarding contextual information in a DW can be found in the literature. One of the main reasons is that contextual information is typically available as text-rich documents (e.g., on-line new, company reports, etc.), which cannot be managed by the structured DW. This dissertation proposes to store the contextual information in an XML document warehouse and a new architecture for the integration of a traditional structured corporate warehouse with the XML document warehouse. The resulting system is called a *contextualized warehouse*.

- The IR model of the contextualized warehouse is based on relevance modeling techniques [33]. Relevance modeling has been traditionally applied to query plain text documents collections. The retrieval model presented in this thesis regards the structure of the XML documents and extends relevance modeling techniques to compute the relevance of the facts with respect to an IR query.

- The contextualized warehouse analysis cubes are called *R-cubes*, since their facts are ranked by their relevance to the selected context, in the same way that an IR presents the documents retrieved ordered by their relevance to the supplied keywords. Moreover, each fact in the *R-cube* is linked to the documents that describe the context of the fact. Thus, *R-cubes* are characterized by two new special dimensions, the relevance and the context dimensions. This thesis provides *R-cubes* with a formal data model and algebra. It extends an existing multidimensional data model [62] with the relevance and context dimensions and studies how the unary algebra operators affect these two new dimensions.

- Given the novelty of the system proposed in this thesis, it is difficult to find a proper testbed to prove its utility. A prototype system that operates real data along with a series of usage cases that show the usefulness of the approach are presented instead.

1.4 Organization

This section describes the organization of the remainder of the thesis. A brief summary of each chapter is shown below.

1.4.1 Second Chapter: The Web, XML and Data Warehousing

This thesis is about DW and Web data, more specifically about DW and text-rich XML documents. This chapter gives an overview of research that combines DW and Web technologies, and introduces the main limitations and opportunities that offer the combination of these two fields. Specifically, it starts by surveying the XML-based technologies that are currently applied to integrate, query and retrieve information in the Web. Afterwards, it describes how these technologies can be applied to DW systems. Thus, it addresses the XML-based integration of heterogeneous DWs, and introduces the problem of dealing with semi-structured data and un-structured data (e.g., documents) in DWs and OLAP.

1.4.2 Third Chapter: Warehouses Contextualized by XML Documents

This chapter proposes an architecture for the contextualized warehouse. The main components of the architecture are: a traditional corporate data warehouse; an XML document warehouse, which stores the documents that describe the circumstances of the corporate facts; and the fact extractor module. The objective of the fact extractor module is to relate each fact of the corporate warehouse to its context. For this purpose it identifies dimension values in the textual contents of the documents. The chapter also presents an example application scenario that will be followed along the rest of the thesis. The chapter concludes by studying how the analysis cubes are materialized in a contextualized warehouse. For this purpose, the analyst specifies an IR condition (a sequence of keywords that state a context of analysis), a path expression that specifies the document sections under study, and a set of MDX conditions to select the corporate facts to analyze. The documents and facts that satisfy the established conditions are retrieved from the respective warehouse. Each selected fact is placed in an *R-cube* along with the documents that describe its context, and its relevance value with respect to the IR condition.

1.4.3 Fourth Chapter: An IR Model for the Contextualized Warehouse

This chapter starts by reviewing the classical retrieval models (i.e., the boolean, vector and probabilistic models) and the more recent language modeling and relevance modeling approaches. Afterwards, it proposes a novel IR model for the contextualized warehouse.

The chapter shows how the XML documents are represented in the document warehouse. The usual tree-like representation of XML documents is chosen, so that the previous work concerning the evaluation of path expressions [14, 4] can be easily applied to the model. In this way, the document warehouse is a forest of XML document trees, where the elements of the original documents are mapped into nodes of the corresponding document trees.

Next, the chapter addresses query processing. Given the IR condition and the path expression used for establishing a context of analysis, it shows how to retrieve the document subtrees that satisfy these conditions. Our approach ensures that only the most relevant document nodes of each selected subtree will be included in the result. The proposed retrieval model adapts relevance modeling techniques to rank the document nodes in the result and the facts of the corporate warehouse described within these nodes. An interesting property of the model is that the sum of the relevance values of all the facts is always equal to one. Since not all document collection are suitable for representing all kinds of contexts, we propose a measure of the overall quality of a query result.

Experimental evaluations show the usefulness of the proposed retrieval model.

1.4.4 Fifth Chapter: A Relevance-Extended Multidimensional Model

In a contextualized warehouse, the analysis is performed on an *R-cube*. *R-cubes* are characterized by two special dimensions, the relevance and context dimensions. In order to formalize the definition of *R-cubes*, this chapter extends a multi-dimensional data model [62] with these two new dimensions. The chapter also redefines the unary algebra operators of [62] (selection, aggregate formation and projection) to regard the relevance and context of facts.

The *relevance-extended selection* operator can be applied to dice and only study a region of the *R-cube*. The relevance values of the facts after the operation are increased by two factors. The first one represents the quality lost in the resulting *R-cube*. The second one measures the increment of importance of the selected facts in the documents, when the non-selected facts are no longer taken into account.

The *relevance-extended aggregate formation* groups the facts characterized by the same values in a given set of dimension categories, and afterwards, it evaluates an aggregation function over each group of facts. In the resulting *R-cube*, there is a new fact representing each group of facts of the original *R-cube*. Each new fact is related to the documents that were associated with any of the original facts of the corresponding group. The relevance of each group is calculated by adding up the relevance values of the original facts in the group.

The *relevance-extended projection* operator is a restriction of original one to avoid removing the relevance or the context dimensions from the *R-cube*.

1.4.5 Sixth Chapter: The Prototype

This chapter presents a prototype of a contextualized warehouse. It shows an example usage case, which consists of a structured warehouse of historical stock market data and a document repository of business journals. The goal is to analyze major world stock market values and determine the causes of the increases and decreases of their stock indexes.

1.4.6 Seventh Chapter: Conclusions

The last chapter addresses conclusions and future work. A list of papers published as the result of this thesis work is included.

CHAPTER 2

The Web, XML and Data Warehousing

The Web is nowadays the World's largest source of information. The Web has brought interoperability to a wide range of different applications (e.g., web services). This success has been possible thanks to XML-based technology, which constitutes a means of information interchange between applications, as well as a semi-structured data model for integrating information and knowledge.

IR is also playing an important role in the Web, since it has enabled the development of useful resource discovery tools (e.g., web search engines). Relevance criteria based on both textual contents and link structure has shown very useful for effectively retrieving text-rich documents. Recently, Information Extraction techniques are also being applied to detect and query the factual data contained in the documents (e.g., Question & Answering Systems).

Finally and more recently, the Web is being enriched with semantic annotations (e.g., RDF and OWL formats), allowing the retrieval and analysis of the Web contents in a more effective way in the near future.

This thesis is about DW and Web data. It proposes a framework to exploit the information contained in text-rich XML documents during OLAP-like analysis. This chapter presents the state of the art of the thesis. Its goal is to survey the most relevant research done on combining DW and Web technologies. Specifically, Section 2.1 summarizes the large range of XML technologies available today. Section 2.2 describes how these technologies can be applied to integrate distributed heterogeneous DW systems. Section 2.3 introduces the problem of dealing with semi-structured data in DW and OLAP systems. Section 2.4 addresses unstructured data, IR and DW technology. Finally, Section

2.5 provides some conclusions.

2.1 XML-Based Web Technology

"The Web is huge and keeps growing at a healthy peace. Most data is unstructured, consisting of text (essentially HTML) and images. Some is structured, mostly stored in relational databases. All this data constitutes the largest body of information accessible to any individual in the history of humanity" [87]. However, in order to exploit all this information in applications new flexible models are required.

In this context, semi-structured data models, and in particular the standardization of XML [39] for Web data exchange plays an important role and opens a wide new range of possibilities. Two main features of its semi-structured data model are the (potential) lack of a predefined schema, and its facilities for representing both the data contents and the data structure integrated into the same document. Specifically, the structure of a document is given by the use of matching tag pairs (termed elements) and the information between matching tags is referred to as a content element. Furthermore, an element is allowed to have additional attributes, where values are assigned to the attributes in the start tag of the element. Figure 2.1 shows an example XML document. XML documents can be associated with and validated against a schema, e.g., a Document Type Definition (DTD). The DTD of an XML document specifies the different elements that can be included in the document, how these elements can be nested and the attributes they may contain.

Other advantages of XML as a semi-structured data format are its simplicity and flexibility. Moreover, XML is free, extensible, modular, platform independent and well-supported.

A number of technologies are evolving around XML. These technologies include among others: XML Schemas [22], an alternative to DTDs that improves data typing and constraining capabilities; the XPath language [14], which is used to refer to parts of XML documents; XQuery [74], the standard query language for XML documents, which provides powerful constructs for navigating, searching and restructuring XML data; XPointer and XLink [21], which define linking mechanisms between XML documents; and XSL [20], which is a family of recommendations for defining XML document transformation and presentation rules.

Nowadays, the hot topic in Web research is the Semantic Web. The objective of this technology is to describe the semantics of Web resources in order to facilitate their automatic location, transformation and integration by domain-specific software applications [18]. A number of languages have been proposed to describe the semantics of resources, namely: Topic Maps (XTM) [63], Resource Description Framework (RDF, RDF/S) [44] and Ontology Web Language (OWL) [85].

The World Wide Web Consortium (W3C) leads the development of the XML standard and related technologies. We refer the reader to the W3C web site (http://www.w3.org) where further details can be found.

2.2 XML-Based DW Integration

The Internet has opened an attractive range of new possibilities for DW applications. First, companies can now publish some portions of their corporate warehouses on the Web. In this way, customers, suppliers and people in general will be able to access this “public” corporate data by using web client applications. The benefits of “plugging” the corporate warehouse into the company web site are discussed in [67]. [29] and [32] study the development of e-commerce applications and click-stream analysis techniques to analyze the behavior of the clients when surfing the company online shop site, and then to provide a user customized view of this web site according to his/her preferences. On the other hand, an even more challenging issue is to apply Internet technology to provide interoperability between distributed heterogeneous warehouses, and to build new (virtual) warehouses where the information available in these heterogeneous warehouses is exploited in a uniform, homogeneous, integrated way. In this context, XML plays an important role as a standard format of data interchange.

This section describes work focused on the definition on XML languages to represent the data and metadata of warehouses. Afterwards, it discusses some XML-based DW integration architectures proposed in the literature.

```
<business_newspaper date=‘‘Dec.1,1998’’>
<cubeFacts version=‘‘0.4’’
xmlns=‘‘http://www.xcube-open.org/V0_4/XCubeFact_base.xcds’’>
<cube id=‘‘sale’’>
   <cell>
      <dimension id=‘‘product’’ node=‘‘LA-123’’/>
      <dimension id=‘‘time’’ node=‘‘2005-08-03’’/>
      <fact id=‘‘sales’’ value=‘‘10’’/>
   </cell>
   <cell>
      <dimension id=‘‘product’’ node=‘‘RS-133’’/>
      <dimension id=‘‘time’’ node=‘‘2005-08-03’’/>
      <fact id=‘‘sales’’ value=‘‘5’’/>
   </cell>
...
</cube>
</cubeFacts>
```

Figure 2.1: Example *XCubeFact* document [28]

2.2.1 XML Languages for DW Interoperability

The first step on the road to interoperability and integration of heterogeneous warehouses is defining a common language for interchanging multidimensional data. With this objective, in [28] a set of XML document formats was proposed, including: *XCubeSchema*, which describes the structure of a data cube by providing its measures and dimension schemata (hierarchy of levels in each dimension); *XCubeDimension*, which defines the members for each dimension level; and *XCubeFact*, which represents the cells of the data cube (i.e., how the dimension and measure values are linked). Figure 2.1 shows an example *XCubeFact* document depicting two cells with sales made on August 3, 2005 for the products LA-123 and RS-133, respectively.

The work presented in [43] also includes its own XML language to interchange data and metadata. This paper describes a Web Service interface to evaluate MDX queries in a remote OLAP system. The main difference between the approaches [28] and [43] resides in their underlying multidimensional model, which in the second case is tightly related to MDX [81]. Apart from these, the authors of [83] propose a UML-based multidimensional model along with its representation in XML. In this case the XML language is only focused on metadata interchange.

2.2.2 XML-Based Integration Architectures

This section surveys relevant research on integrating distributed data warehouses. These proposals use XML languages to express the metadata describing data sources, or as a canonical language to transfer data between the different components of the system.

A framework that combines the federation and mediation architectures is presented in [40]. As Figure 2.2 shows, the proposed architecture is organized into four layers, namely: the local, mediation, federated and client layers. The lower local layer consists of a collection of independent heterogeneous DW systems distributed over the Internet. These systems execute queries coming from the mediation layer and return the results to the corresponding mediator. In order to participate in the federation, each DW should provide its local schema to the corresponding mediator. At the mediation layer, each mediator module receives subqueries from the federated layer, translates them into the local DW query language, restructures the results and returns them to the federated layer. Mediators also provide the federated layer with export schemata, which are the translation of local schemata into a common canonical data model. The federated schema imports the export schemata of the local DW systems and integrates them into a single DW schema. In the federated layer, the queries of the client applications are first divided into subqueries that are issued to the corresponding mediators, and afterwards, the results are merged and returned to the client application. The applications of the client layer will access the federated warehouse using a single homogeneous interface.

In this work, XML documents are used to represent the local, export and federated schemata. Since these documents represent DW schemata, they are similar to the *XCubeSchema* documents proposed in [28]. The mapping between the federated and the import schemata is also specified in an XML document, in which we can find, for example, the correspondence between federated and local warehouse dimension names.

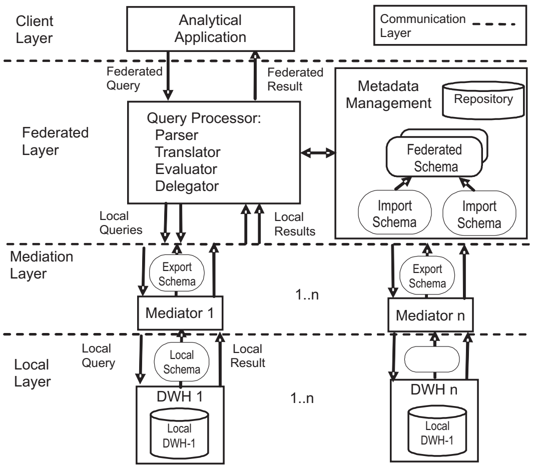


Figure 2.2: Federated DW architecture [40]

A similar architecture was proposed in [48] but with a different underlying canonical multidimensional model called *MetaCube* [49]. The authors of this work define a new type of XML document called *MetaCube-X*, which is the XML expression of a *MetaCube* schema representing the export and federated schemata. None of the approaches [40, 48] address query processing or the use of XML for representing the results of the local and federated queries. They only focus on schema integration issues. However, as stated by [40], in order to completely overcome semantic heterogeneity in DW integration (e.g., different hierarchies for the same dimension) a deeper study of the mapping strategies is required.

The work made in [84] classifies the main issues that arise from the semantic integration of heterogeneous warehouses, and studies how they should be addressed. This work also proposes a federated architecture in which mediator components are replaced by native XML databases (see Figure 2.3). Each native XML database stores an XML version of the cubes available in the corresponding local warehouse along with their export schemata. Each local database manager provides its *site metadata* which is a formal description of the dimensions and the semantics of the measures involved in the exported cubes.

Heterogeneity conflicts between export schemata are solved semi-automatically by studying the *site metadata*, and by designing and evaluating XQuery statements to update the exported XML data cubes and their schemata. Finally, the resulting cubes are integrated into a global cube that can be analyzed by users.

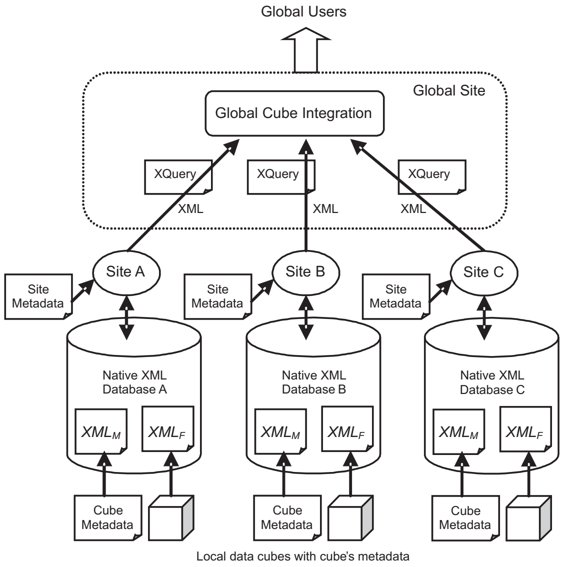


Figure 2.3: Federated DW architecture [84]

A different architecture, based on Grid technology [23], is proposed in [50, 51]. Figure 2.4 shows the system architecture. Analysis takes place as follows. (1) A virtual “universal” data warehouse schema representing all the data available in the warehouses is presented to the user. (2) The user establishes an analysis query. (3) The *Collection Server* analyses the query, and according to a distribution schema (i.e., how the data is distributed between the different warehouses) sends request to the relevant warehouses. (4) The involved warehouses compute the selection and aggregation calculations in parallel. A Grid-based distributed computing platform is used to perform this distributed data processing. (5) The *Collection Server* receives the data and performs a final aggregation, if needed. (6) The *Collection Server* sends the resulting cube data to the OLAP Server. (7) The user analyses the cube in the OLAP Server.

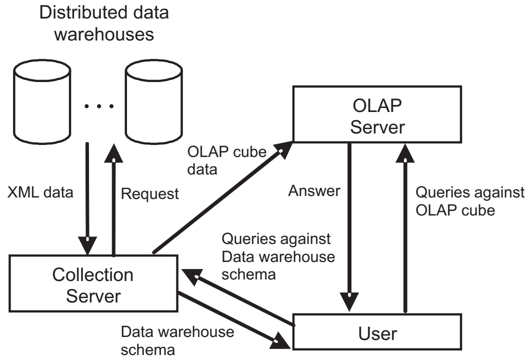


Figure 2.4: Distributed DW architecture [50, 51]

In this approach XML is used to represent the "universal" cube schema, the initial user query, the distribution schema, the data returned by each warehouse, and the final analysis cube data. The authors of [50, 51] propose to transform the XML data returned by the warehouses into the format suitable for the OLAP Server by applying standard XML tools like XSLT. The main contribution of [50, 51] is the use of Grid technology to distribute the computation needed in the cube construction process. However, they do not show how the heterogeneity conflicts are solved.

Although the application of XML technology has supposedly been a great advance for DW integration, so far this integration has been mostly syntactic, as it simply consists of translating DW schemata into DTD or XML files. Semantic heterogeneity discrepancies between DW schemata are still handled manually [40] or semi-automatically [84]. Trying to address these conflicts, some work has applied Semantic Web languages to describe the DW conceptual schemata. For instance, [12] follows a federation approach too, and applies Topic Maps to describe the local multidimensional schemas. Thus, the measures, dimensions and hierarchy dimension levels are represented by topics in the local topic maps. Association relations are used for modeling the facts structure (i.e., the dimensions and measures that constitute the fact) and the roll-up relationships between dimension levels. Afterwards, at the federated layer, a global topic map provides the unifying view of the local schemas. Thus, the global topic map deals with the semantic conflicts between the local schemas. For example, consider the *Time* dimension defined in two different local schemas. These dimensions include two equivalent levels *day* and *tag* (day in German). In the global topic map there will be only a topic *day* with two scopes, *English* and *German*. Then, each scope will be linked to the corresponding dimension.

2.3 DWs for Semi-Structured Data

With the emergence of XML as the lingua franca of the Web, semi-structured information is now widely available, and several solutions have been proposed to build warehouses for XML data. This section first introduces work oriented towards the construction of XML web data repositories, then presents the research done on the design of multidimensional databases for XML data, and finally focuses on the extension of OLAP techniques for analyzing XML data.

2.3.1 XML Web Data Repositories

The problem of gathering and querying web data is not trivial, mainly because data sources are dynamic and heterogeneous. In this context, some papers are focused on the construction of repositories for XML [87] or web documents [82]. The main issues of this research area include the efficient storage, indexing, query processing, data acquisition, change control and schema integration of data extracted from dynamic and heterogeneous web sources. This section summarizes the main results of two important projects: Xyleme [87] and Whoweda [82].

Xyleme [87] was an ambitious project aimed at building a warehouse for all the XML data available on the Web. The Xyleme system runs on a network of distributed Linux PCs. In order to store such a huge amount of XML data, a hybrid approach is proposed to keep the tree structure of XML documents in a traditional DBMS until a certain depth, and then store the pieces of documents under the selected depth as byte streams. Thus, the upper part of the XML documents structure is always available, but the lower sections require parsing to obtain the structure. Query processing is based on an algebra operator that returns the set of documents which satisfy a given tree pattern. Xyleme partitions the XML documents into clusters corresponding to different domains of interest (e.g., tourism, finance, etc.) which allow indexing each cluster on a different machine. Since the documents in a cluster may follow different DTDs, an abstract DTD for the cluster along with the mappings to the original DTDs is inferred. In this way, the user queries the cluster by using the abstract DTD. In order to acquire the XML documents several crawlers run in parallel. The refreshment of a copy is performed depending on the importance of the document, its estimated rate change, or under the request of the owner of the document (i.e., in a notification/subscription basis).

The Whoweda (Warehouse of Web Data) project is also aimed at warehousing relevant data extracted from the Web [82]. Their efforts have been mainly focused on the definition of a formal data model and an algebra to represent and manage web documents [9], their physical storage [88] and change detection [10]. In their data model, called WHOM (Warehouse Object Model) [9], a web warehouse is conceived as a collection of web tables. The tuples of these tables are directed graphs where each node represents a document, and the edges depict hyperlinks between documents. In order to manage the data stored in the web tables, a set of algebraic operators is provided (i.e., global web coupling, web join, web select, etc.). For example, the global web coupling operator retrieves a set of inter-linked documents satisfying a query with conditions on the metadata, content, structure and hyperlinks of the documents. The result of the operation is a new web table where each new tuple matches a portion of the WWW satisfying the constraints of the query. In the web join operator, the tuples from two web tables containing identical nodes are "concatenated" into a single joined web tuple. Two nodes are considered identical if they represent the same document with the same URL and modification date.

XML data change is an important issue that has spawned a lot of research. Xyleme [87] allow users to subscribe to changes in an XML document [47]. When such a change occurs, subscribers receive only the changes made, called *deltas* [41, 15], and then incrementally update the old document. This approach is based on a versioning mechanism [41] and an algorithm to compute the difference between two consecutive versions of an XML document [15]. The Whoweda project addresses change detection over sets of inter-linked documents, instead of over isolated XML documents. The global coupling algebra operator may be used to state a set of relevant inter-linked documents to "watch". Given two versions of this set of inter-linked documents materialized in two different web tables, the differences between these two versions are calculated by applying the web join and the web outer join algebra operators. The authors of [89] considered a more general problem by studying how to update materialized views of graph-structured data when the sources change. In [5] an adaptative query processing technique for federated database environments was proposed. Finally, [55, 56] considers adaptivity in a federation of XML and OLAP data sources (see Section 2.3.3).

2.3.2 XML Multidimensional Database Design

This section surveys the most relevant research on multidimensional design for XML data. Specifically, the work by Golfarelli et al. [25], Pokorný [66], and Jensen et al. [30] is studied.

The authors of [25] argue that existing commercial tools support data extraction from XML sources to feed a warehouse, but both the warehouse schema and the logical mapping between the source and target schemas must be defined by the designer. They show how the design of a data mart can be carried out starting directly from an XML source, and propose a semi-automatic process to building the DW schema.

Since the main problem in building a DW schema is to identify many-to-one relationships between the involved entities, they first study how these relationships are depicted in the DTD or the XMLSchema of the XML documents. Such relationships are modeled by sub-elements nesting in DTDs and XMLSchemas, and key/keyRef in XMLSchemas. ID/IDREF(s) attributes of the DTDs are not considered, since IDREF(s) are not constrained to be of a particular element type. For example, if ID attributes are defined for the elements `car` and `manufacturer`, and an IDREF attribute is stated for an `owner` element, the IDREF attribute of the `owner` element may reference either a `car` or a `manufacturer` element in an instance XML document. Just focusing on DTDs, the authors provide an algorithm which represents the structure modeled by the DTD as a graph, and starting from a selected element (the analysis fact), semi-automatically builds the multidimensional schema by including the dimension and dimension levels depicted by the many-to-one relationships found between the elements and attributes of the graph. In order to understand why the designer participation is needed, consider the following example: In a DTD the definition `owner(car*)` states that an owner may have many cars. However, the cardinality of the inverse relationship is not stated in the DTD. That is, the same car may belong to several owners. They solve the problem by querying the document instances and asking the user.

In [25] it was assumed that the schema of the source XML data is provided by a single DTD or XMLSchema. In [66] a different approach is followed, by considering that when the source XML data is gathered from different sources, then each source will provide its particular DTDs. Thus, dimensions are modeled as sequences of logically related DTDs, and the XML-star schema is defined by considering the facts as XML elements (see Figure 2.5). In order to build the dimension hierarchies, this approach defines a subDTD as the portion of a source DTD that characterizes the structure of a dimension member. Then, XML view mechanisms are applied to select the members of each dimension. The concept of referential integrity for XML data is applied to establish hierarchical relationships between them.

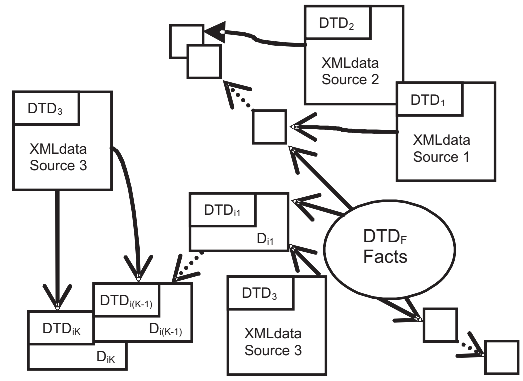


Figure 2.5: XML-star schema [66]

The work in [30] deals with the conceptual design of multidimensional databases in a distributed environment of XML and relational data sources. This approach use UML diagrams [52] to describe the structure of the XML documents as well as the relational schema. For relational databases, commercial reverse engineering tools can be applied to build the corresponding UML diagrams. For XML documents, they propose an algorithm [31] that builds the UML diagram from the DTDs of the XML sources. They also provide a methodology to integrate the source schemata into an UML snowflake diagram, and take special care in ensuring that XML data can be summarized. For example, they study how XML elements with multiple parents, ID-references between elements or recursive element nesting should be managed. The resulting UML schema can be applied for the integration of sources in a multidimensional database.

2.3.3 Extending OLAP Techniques to XML Data

This section mainly studies the work by Pedersen et al. on the extension of OLAP techniques to analyze XML data [60, 54]. Pedersen et al. argue that the dynamicity of today's business environments are not handled well by current OLAP systems, since physically integrating data from new sources is typically a long, time-consuming process, making logical integration the better choice in many situations. Thus, by considering the increasing use of XML for publishing web data, they aim their work at the logical federation of OLAP and XML data sources. Their approach allows the execution of OLAP operations that involve data contained in external XML data. In this way, XML web data can be used as dimensions [60] and/or measures [54] of the OLAP cubes.

In this work, OLAP-XML federations use links for relating dimension values in a cube to elements in an XML document (e.g., linking the values of a Store-City-Country dimension to a public XML document with information about cities, such as state and population). Thus, a federation consists of a cube, a collection of XML documents, and the links between the cube and the documents. The most fundamental operator in OLAP-XML federations is the *decoration operator* [57], which adds a new dimension to a cube based on the values of the linked XML elements. This work presents an extended multidimensional query language called SQL_{XM} that supports XPath expressions and allow linked XML data to be used for decorating, selecting and grouping fact data. For example, the query `SELECT SUM(Quantity), City/Population FROM Purchases GROUP BY City/Population` computes the total purchases grouped by the city population which is found only in the XML document.

Figure 2.6 shows the architecture of the system proposed in [60]. Along with the Federation Manager, it includes an OLAP component (i.e., a commercial OLAP server able to evaluate multidimensional queries), and an XML component (i.e., an XML database system with an XPath interface). The Federation Manager receives SQL_{XM} queries and coordinates their execution in the two

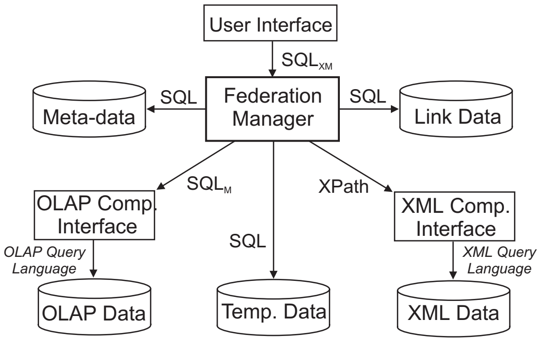


Figure 2.6: OLAP-XML federation architecture [60]

repositories. The metadata, link data and temporary data databases (e.g., traditional relational databases) are also managed by the Federation Manager component.

The un-optimized approach to process an SQL_{XM} query is as follows. First, any XML data referenced in the query is fetched and stored in a temporary database as relational tables. Second, a pure OLAP query is constructed from the SQL_{XM} query, resulting in a new table in the temporary database. Finally, these temporary tables are joined, and the XML-specific part of the SQL_{XM} query is evaluated on the resulting table along with the final aggregation.

Pedersen et al. provide both rule-based and cost-based optimization strategies focused on reducing the amount of data moved from the OLAP and XML components to the temporary database. The rule-based optimization algorithm partitions an SQL_{XM} query tree, meaning that the algebra operators are grouped into an OLAP part, an XML part, and a relational part. Algebraic query rewriting rules are applied to push as much of the query evaluation towards the OLAP and XML components as possible. The cost-based optimization strategies are based on the cost model described in [58], and a set of the techniques that include in-lining literal XML data values into OLAP predicates, caching and pre-fetching [59].

In a more recent paper [54], Pedersen et al. show an implementation of their XML-OLAP federation for the commercial OLAP tool TARGIT Analysis, and extend their approach to allow the evaluation of federated OLAP queries with XML data as measures.

A different approach to analyzing XML data with OLAP technology was presented in [8]. This paper proposes an extension to XQuery with constructs for the grouping and numbering of results. The new constructs simplify the construction and evaluation of queries requiring grouping and ranking, and at the same time, they enable complex analytical queries.

Notice that these proposals [8, 60, 54] deal with highly structured XML data (e.g., on-line XML product pricing lists), from where the measures and dimensions can be directly selected using XPath expressions. However, these approaches are not suitable for analyzing text-rich XML documents, which require some kind of document processing to extract measures and dimension values from their textual contents [64]. The next section deals with the combination of DW and IR technologies to exploit text-rich XML documents.

2.4 DWs & IR for Unstructured Data

Many new web applications store unstructured data with large text portions requiring IR techniques to be indexed, queried, and retrieved. This section studies how the OLAP and IR approaches can be combined to build a warehouse of text-rich documents. Current research follows two main lines: the application of multidimensional databases to implement an IR system, and the extension of OLAP techniques to support the analysis of text-rich documents.

2.4.1 Cubes for Document Analysis and Retrieval

OLAP cube dimensions provide an intuitive general-to-specific (or vice-versa) method for the analysis of document contents. Moreover, the optimized evaluation of aggregation functions in multidimensional databases can be applied to efficiently compute the relevance formulas of IR systems. This section studies how multidimensional databases and OLAP can help IR.

The work presented in [42] implements an IR system based on a multidimensional database. As Figure 2.7 shows, the fact table measures the weights (i.e., the frequency) of each term at each document. Thus, the relevance of a document to a query is computed by grouping its terms weights, which are obtained by slicing the cube on the terms dimension. The final relevance value is calculated by applying the so called pivoted cosine formula [79] to the weights of the query terms. Furthermore, if the document collection is categorized by location and time, more complex queries can be formulated, like retrieving the documents with the terms ”financial crisis” published during the first quarter of 1998 in New York, and then drilling down to obtain those documents published in July 1998. By following this research line, in [35] the authors study different indexing strategies to improve the performance of their system, and in [36] propose a method for incorporating a hierarchical category dimension to classify the documents by theme.

The benefits of implementing an IR system on a multidimensional database are also discussed in [46] together with a novel user interface for exploring document collections. This approach defines a dimension for each subject of analysis relevant to the application domain (e.g., in a financial application, subjects such as economic indicators, industrial sectors and regions are relevant dimensions). Each dimension is modeled as a concept hierarchy. They choose

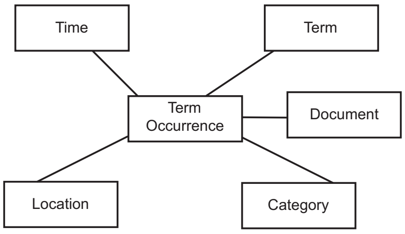


Figure 2.7: Multidimensional implementation of an IR system [42]

a star schema too, but instead of keeping term weights, the fact table links documents to categories of concepts.

Finally, a recent paper [53] provides mechanism to perform special text aggregations on the contents of XML documents, e.g., getting the most frequent words of a document section, their most frequent keywords, a summary, etc. Although these text-mining operations are very useful to explore a text-rich XML documents collection, they cannot be applied to evaluate OLAP operations over the facts described by document textual contents. This is the focus of the following section.

2.4.2 IR Techniques Applied to OLAP

Nowadays, most information is published on the Web as unstructured documents. These documents typically have large text sections and may contain highly valuable information about a company's business environment. The current trend is to find these documents available in XML-like formats [87]. This situation opens a novel and interesting range of possibilities for DW and OLAP technology: trying to include the information described by these text-rich XML documents in the OLAP analysis. We can thus imagine a DW system able to obtain strategic information by combining all the company sources of structured data and documents.

The approaches discussed in Section 2.4.1 to implement an IR system by using a multidimensional database are very useful to explore a text-rich XML documents collection. However, these techniques cannot be applied to evaluate OLAP operations over the facts described by document textual contents. The extension of OLAP techniques for XML data studied in Section 2.3.3 are not suitable for analyzing text-rich documents either. They only deal with highly structured XML data (e.g., on-line XML product pricing lists), from where the measures and dimensions can be directly selected using XPath expressions.

The objective of this thesis is to provide a framework to exploit the factual information found in the XML documents textual contents. For this purpose some kind of document processing to extract measures and dimension values from their textual contents is needed. In particular, we propose to contextualize the facts of a traditional corporate DW with the documents that describe their circumstances. The dimension values found in the documents will be used to relate documents and facts. Traditional DW technology does not regard contextual information, mainly because of the unstructured nature of these information sources [29]. Thus, a new architecture for a contextualized warehouse is needed. In this architecture the XML document sources play an important role, and a fact contextualization process is provided. In our approach, the user establishes an analysis context by supplying an IR query. The documents that satisfy the IR condition are retrieved, and then related to the corporate facts. The facts of the resulting OLAP cubes will be ranked by their relevance to the IR query. In this way, a new IR model for estimating the relevance of the facts to an IR query is required. Moreover, an IR extended multidimensional model and algebra to manage the relevance and context of the facts in the OLAP operations has also to be studied. All these requirements are addressed along this dissertation.

To our best knowledge, the unique work directly related to this thesis is [70]. This paper proposes to annotate external information sources (e.g., documents, images, etc.) by means of an ontology in RDFs format that comprises all the values of the data warehouse's dimensions. In this way, the results of OLAP queries can be associated with the external sources annotated with the same dimension values. However, unlike the work proposed in this thesis it does not provide a formal framework for calculating fact relevance with respect to user queries.

2.5 Conclusions

The advent of XML and related technologies is playing an important role in the future development of the Web. DW and OLAP tools take part in the Web revolution. This chapter has summarized the most relevant research on combining both DW and XML-Based Web technologies. As far as we know there does not exist any similar survey in the literature.

The chapter has studied the advantages of XML as an integration tool for heterogeneous and distributed DW systems. In this sense, it has first described research focused on the definition of XML languages to represent warehouses data and metadata, and then discussed different XML-based data warehouse integration architectures. It has also addressed the construction of warehouses for semi-structured XML web data. Specifically, it has introduced some work oriented towards the construction of XML web data repositories, the research done on the design of multidimensional databases for XML data, and the extension of OLAP techniques for analyzing external XML data. Nowadays, most information is published on the Web as unstructured (in the near future text-rich XML) documents. This chapter has shown how IR techniques and OLAP technologies can be combined to explore text-rich documents collections, i.e., the use of multidimensional databases for implementing IR systems.

As previously discussed, the analysis of the factual information described in the documents is a very hard issue. It is difficult to find work on the current literature that tries to address this problem. This thesis proposes a particular setting where this analysis is possible, called a contextualized warehouse. It proposes an architecture for such a system, an IR model to estimate the relevance of the facts to an IR user query, and a relevance-extended multidimensional model. The rest of this dissertation is devoted to this objective.

In the future, with the Semantic Web widely adopted, companies will be able to gather huge amounts of valuable semantically-related metadata concerning their subjects of interest. All this information will be used to create metadata warehouses for global decision-making. As far as we know, currently there does not exist any approach to build data warehouses for the metadata generated by the Semantic Web.

CHAPTER 3

Warehouses Contextualized by XML Documents

In this chapter an architecture for a contextualized warehouse is proposed. The main components of the architecture are: a traditional corporate data warehouse, an XML document warehouse and the fact extractor module. The XML document warehouse stores the unstructured data that describe the context of the corporate facts. Building a contextualized warehouse mainly means relating each fact of the corporate warehouse to its context. This is the objective of the fact extractor module.

The chapter presents an example application scenario. The rest of examples of the thesis follow this scenario.

The analysis cubes of a contextualized warehouse are called *R-cubes*, since they include two special dimensions: the *relevance* and *context* dimensions. This chapter also studies how the *R-cubes* are materialized in the contextualized warehouse.

Section 3.1 presents the contextualized warehouse architecture. Section 3.2 introduces the example scenario. Section 3.3 discusses the document facts extraction process. Section 3.4 shows how *R-cubes* are materialized. Finally, Section 3.5 gives some conclusions.

3.1 System Definition and Architecture

A contextualized warehouse is a decision support system that allows users to combine all their sources of structured and unstructured data, and to analyze the integrated data under different contexts. Figure 3.1 shows the proposed architecture for the contextualized warehouse. Its main components are a corporate warehouse, an XML document warehouse and the fact extractor module.

The corporate warehouse is a traditional data warehouse that integrates the company structured data sources (e.g., the different department databases). The construction of a corporate warehouse for structured data has been broadly discussed in classical references like [29, 32].

The unstructured data coming from external and internal sources are stored in the document warehouse as XML documents. These documents describe the context of the corporate facts. Thus, the XML document warehouse is a repository of text-rich XML documents containing relevant information for analysis purposes. The document warehouse allows the user to evaluate queries that involve IR conditions and path expressions. IR conditions are used for establishing constraints on the textual contents of the documents (e.g., retrieve documents containing the keywords “financial crisis”), whereas path expressions restrict the structure of the documents (e.g., only consider the articles published in the Economy section of a digital newspaper). The result is a set of document fragments ranked by their relevance to the stated query conditions. This thesis does not address the problems of data acquisition, filtering, change control and schema integration of XML data extracted from heterogeneous sources. They are studied in other works like [87].

The fact extractor module relates the facts of the corporate warehouse with the document fragments that describe their contexts. This module identifies dimension values in the textual contents of the documents and relates each document fragment with those facts that are characterized by the same dimension values. Section 3.3 describes this process in detail by means of an example.

In this way, a contextualized warehouse keeps a historical record of the facts and their contexts as described by the documents. In a contextualized warehouse, the user specifies an analysis context by supplying a sequence of keywords (i.e., an IR condition like “financial crisis”). The analysis is performed on a new type of OLAP cube, called *R-cube*, which is materialized by retrieving the document fragments and facts related to the selected context.

R-cubes have two special dimensions, namely: the *relevance* and the *context* dimensions. Thus, each fact in the *R-cube* will have a numerical value representing its relevance with respect to the specified context (e.g., how important the fact is for “financial crisis”). Thereby the name *R-cube* (Relevance cube). Moreover, each fact will be linked to the set of document fragments that describe its context.

The relevance and context dimensions provide information about facts that can be very useful for analysis tasks. The relevance dimension can be used to explore the most relevant portions of an *R-cube*. For example, it can be used to identify the period of a political crisis, or the regions under economical development. The usefulness of the context dimension is twofold. First, it can be used to restrict the analysis to the facts described in a given subset of documents (e.g., the most relevant documents). And second, the user will be able to gain insight into the circumstances of a fact by retrieving its related document fragments.

Section 3.4 shows the *R-cubes* construction process. The next chapter proposes an IR model to retrieve the document fragments that describe the selected analysis context (IR query) and to estimate the relevance of the facts described by these document fragments to this analysis context. Chapter 5 presents a data model and algebra for the *R-cubes*.

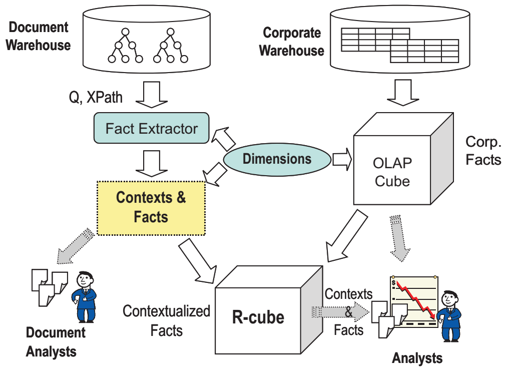


Figure 3.1: Contextualized warehouse architecture

3.2 Example Scenario

This section presents an example contextualized warehouse. More specifically, it describes the example corporate and XML document warehouses. This application scenario will be followed along the rest of examples of the thesis.

Let us consider the corporate warehouse of an international provider of vegetable oil by-products. The main products of this company include: $fo1$, $fo2$ (used as preservatives in the $food$ sector), and $he1$ and $he2$ (used in the elaboration of $healthcare$ products). The company keeps in its corporate warehouse a historical record of its sales, the quantity sold ($Quantity$ measure) and its cost ($Amount$ measure), per product and customer. Thus, the dimensions of the corporate warehouse are $Time$, $Products$ and $Customers$. The $Products$ are classified into $Sectors$ ($food$ and $healthcare$). $Customers$ are organized into $Countries$ and $Regions$ (e.g., $Southeast\ Asia$, $Central\ America$, etc.). Figure 3.2 shows the schema of the example corporate warehouse.

Our example company also maintains a document warehouse of business newspapers gathered from the Internet in XML format. Figure 3.3 shows a fragment of an example document of this warehouse. The document depicts a

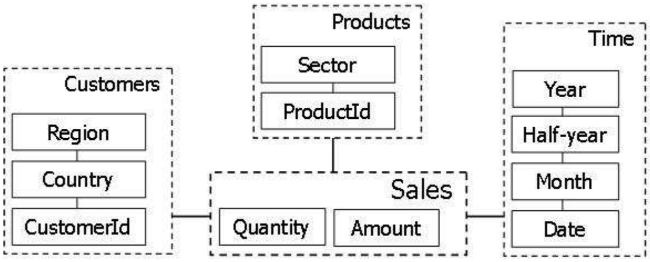


Figure 3.2: Schema of the example corporate warehouse

context for the sales of food sector products to customers of the Southeast Asian region, made during the second half of 1998. Notice that contexts descriptions are very useful for analysis tasks, as they contain detailed information about the facts of the corporate warehouse. For example, the document in Figure 3.3 could help to understand a sales drop.

```
<business_newspaper date=‘‘Dec.1,1998’’>
<economy>
<article>
<headline>Financial Crisis Hits Southeast Asian Market</headline>
<paragraph>
The financial crisis in Southeast Asian countries, has mainly
affected companies in the food market sector. Particularly, Chicken
SPC Inc.  has reduced total exports to $1.3 million during this
half of the year from $10.1 million in 1997.
</paragraph>
<paragraph>...
</article>...
</economy>...
</business_newspaper>
```

Figure 3.3: Example fragment of a business journal

3.3 The Fact Extractor Module

Building a contextualized warehouse mainly means relating each fact of the corporate warehouse with its context. The fact extractor tool uses the dimensions defined in the corporate warehouse to detect the facts described in the documents. This section describes this process in detail by means of an example.

By applying specific information extraction techniques [37, 19, 26], and considering the three analysis dimensions of the corporate warehouse, the dimension values $Southeast\ Asia$, $food$, and $1998/2nd\ half$ can be identified in the paragraph of the document shown in Figure 3.3. It is worth mentioning that the fact extractor module use synonyms (e.g., aliment and food) and other terms semantically related to the dimension values of the corporate warehouse for finding the references to the corporate dimension values in the documents textual contents. The fact extractor tool builds all the valid facts with them, in this case, ($Products.Sector = food$, $Customers.Region = Southeast\ Asia$, $Time.Half\_year = 1998/2nd\ half$). As it can be noticed, some of these dimension values are not completely *precise* and belong to non-base dimension categories. For example, the $SoutheastAsia$ dimension value belongs to the category *Region* of the $Customers$ dimension. We may also find documents where some dimensions are not mentioned, resulting in *incomplete* facts. For each fact, the fact extraction tool also calculates the number of times that its dimension values occur in the document fragment (i.e., the fact dimension values frequency). This frequency value determines the importance of the fact in the document, and will be used to estimate the relevance of a fact in a given context. Notice that a particular dimension value may appear more than once in the text. The frequency of the fact ($Products.Sector = food$, $Customers.Region = Southeast\ Asia$, $Time.Half\_year = 1998/2nd\ half$) in the example paragraph is three.

Let us now consider the second sentence in the paragraph of Figure 3.3. It depicts two facts: ($Company = Chicken\ SPC$, $Time.Year = 1997$, $Exports = \$10,100,000$), ($Company = Chicken\ SPC$, $Time.Half\_year = 1998/2nd\ half$, $Exports = \$1,300,000$). That is, the total exports of the company Chicken SPC Inc. during 1997 and the second half of 1998 were of $10.1 million and $1.3 million, respectively. Chicken SPC Inc. could be a potential customer or competitor of our example oil provider company. In this way, the document warehouse also provides highly valuable strategic information about some facts that are not available in the corporate warehouse nor in external databases. We note that sometimes it is relatively easy to obtain these facts, for example, when they are presented as tables in the documents. However, most times documents contain already aggregated measure values (total exports in the facts of the previous example). The main problem here is to automatically infer the implicit aggregation function that was applied (i.e., average, sum, etc.) Alternatively, the system could ask the user to guess the aggregation function by showing him/her the document contents. In this context, different IR and information extraction-based methods for integrating documents and databases are discussed in [6]. Specifically, [6] proposes a strategy to extract from documents information related to (but not present in) the facts of the warehouse.

This thesis is mainly focused on the fact *dimensions*, leaving for future work the management of measures extracted from texts. Notice that documents extracted measure values are not essential to construct a contextualized warehouse, since the dimension values found in a document fragment are sufficient to relate it with the corporate facts that are characterized by these dimension values.

3.4 Building an *R-cube*

This section explains how the analysis cubes are materialized from the contextualized warehouse. From these cubes, users can study the contextualized facts.

In order to create an *R-cube* the analyst must supply a query of the form $(Q, XPath, MDX)$, which states the following restrictions: Q is an Information Retrieval (IR) condition, consisting of a sequence of keywords that specifies the context under analysis; $XPath$ is a path expression [14] that establishes the document sections under study; finally, MDX are conditions over the dimensions and measures of the corporate warehouse [81]. MDX conditions are used for selecting the subset of facts to analyze from the corporate warehouse. Here, our purpose is not to define a new query language, but to identify the type of conditions needed to build an analysis cube in a contextualized warehouse.

The query process takes place as follows:

1. First, the IR condition Q and the path expression $XPath$ are evaluated in the document warehouse. The result is a set of document fragments satisfying $XPath$ and Q, along with their relevance with respect to Q. More details about this retrieval process will be given in Chapter 4.

2. Second, the fact extractor component parses the document fragments obtained in step (1) and returns the set of facts described by each document fragment, along with their frequency. Notice that we do not parse entire documents, but those document fragments returned by the document warehouse.

3. Third, or in parallel to steps (1) and (2), the MDX conditions are evaluated on the corporate warehouse.

4. Next, we relate the document fragments with those facts of the corporate database whose dimension values can be "rolled-up" or "drilled-down" to some (possibly imprecise or incomplete) fact described by this document fragment.

5. Finally, we calculate the relevance of each fact, resulting in an *R-cube*. Chapter 4 discusses the facts relevance calculus.

By following with the running example, let us consider the analysis of the sales of food products under the context of a financial crisis reported by the business articles of the document warehouse. Thus, given the IR condition $Q =$"$financial, crisis$", $XPath =$"$/business\_newspaper/economy/article//$" and the OLAP expression $MDX = (Products.[food]$, $Customers.Country$, $Time.[1998].Month$, $SUM(Amount) > 0)$, the contextualized warehouse will return the *R-cube* presented in Table 3.1. This *R-cube* includes the set of facts of the corporate warehouse that satisfy the stated MDX conditions, along with their relevance values with respect to the IR condition (relevance dimension, depicted as R), and the set of text fragments where each fact is described (context dimension, represented by $Ctxt$).

F	ProductId	Country	Month	Amount	R	Ctxt
f_1	$fo1$	$Cuba$	1998/03	$4,300,000$	0.05	$d_{23}^{0.005}, d_{47}^{0.005}$
f_2	$fo2$	$Japan$	1998/02	$3,200,000$	0.1	$d_{50}^{0.02}$
f_3	$fo2$	$Korea$	1998/05	$900,000$	0.2	$d_{84}^{0.04}$
f_4	$fo1$	$Japan$	1998/10	$300,000$	0.4	$d_{123}^{0.04}, d_7^{0.08}$
f_5	$fo2$	$Korea$	1998/11	$400,000$	0.25	$d_7^{0.08}, d_{69}^{0.01}$

Table 3.1: Example *R-cube*

In Table 3.1 each row represents a fact. The R and the $Ctxt$ columns (dimensions) depict the relevance value and the context of the facts, respectively. Each d_j^r denotes a document fragment of the collection whose relevance with respect to Q is r.

The relevance is a numeric value. It measures the importance of each fact in the context established by the initial query conditions. The most relevant facts of our example *R-cube* are the facts f_4 and f_5, which involve the sales made to Japanese and Korean customers during the months of October and November 1998. In fact, the sales represented by these facts experimented the sharpest drop. Each document d_j of the context dimension has also associated a relevance value (represented by the superscript) which measures how well this document describes the selected analysis context. Let d_7 depict the paragraph element of the document shown in Figure 3.3. The estimated relevance of this document fragment for "$financial, crisis''$ is 0.08. As discussed in Section 3.3, the fact extractor module found the fact ($Products.Sector = food$, $Customers.Region = Southeast\ Asia$, $Time.Half\_year = 1998/2nd\ half$) described in the paragraph represented by d_7. During the fourth step of the query process we discover that the facts f_4 and f_5 can be "rolled-up" to the fact found in the document fragment d_7, since both f_4 and f_5 depict sales of products of the *food* sector, made to *Southest Asian* customers, during the *second half of 1998*. Then, f_4 and f_5 are related to the document fragment d_7 through the context dimension in the resulting *R-cube*. We could obtain the details of the facts f_4 and f_5, described in the relevant document fragments, by performing a *drill-through* operation on the context dimension [81]. By studying these documents fragments we will find out that d_7, i.e., the Southeast Asian financial crisis reported by the paragraph of Figure 3.3, is a valid explanation for the sales drop.

Unlike OLAP-XML federations like those proposed in [54], *R-cubes* are materialized once, when the query is fetched to the contextualized warehouse, and will be incrementally updated when new relevant documents and data satisfying the original query are added to the system. The main advantage of this approach is that pre-aggregations can be performed over *R-cubes*, enabling fast analysis operations.

3.5 Conclusions

This chapter has presented an architecture for the contextualized warehouse. Its main components are: a traditional corporate data warehouse, an XML document warehouse and the fact extractor module. This architecture allows users to combine their sources of structured an unstructured data (i.e., corporate facts and text-rich XML documents, respectively), and to analyze the integrated data under different contexts (perspectives).

In order to build an analysis cube the user supplies an IR condition, a path expression and a set of MDX conditions. The IR condition and the path expression state the analysis context and are evaluated in the XML document warehouse. The MDX conditions are used for selecting the subset of corporate facts to analyze. The document fragments and facts that satisfy the established conditions are retrieved from the respective warehouse. The fact extractor module relates the facts with the document fragments that mention their dimension values. Each selected fact is placed in an *R-cube* along with its relevance value with respect to the IR condition and the document fragments that describe its context.

The IR condition is a sequence of keywords (i.e., "financial, crisis"). It restricts the analysis to those documents that contain the specified keywords. This kind of condition is necessary when managing large repositories of text-rich documents about many different topics. Moreover, notice that not all the documents sections will be appropriate for answering all types of queries (e.g., the articles in the society section of a newspaper to perform an analysis on financial crisis). In this way, it is also necessary to focus the study in the document sections relevant to the particular analysis. We use the path expressions for this purpose.

In an IR system, the typical result of a query is a list of documents ordered by their estimated relevance to the IR condition. Thus, in a contextualized warehouse, the facts in the resulting *R-cube* may be related to document fragments that have different relevance degrees. Therefore, these facts will also be somehow relevance graded. Intuitively, the relevance of a fact will depend on the relevance index of the document fragments which describe them. A further issue to consider is that the same fact can be presented at different parts of a document fragment, which increases the importance of the fact at this document fragment.

The next chapter proposes a retrieval model for the contextualized warehouse. In the model, the structure of the XML documents is explicitly represented, so that path expressions can be evaluated. The resulting facts are ranked by a relevance measure that considers both the relevance of the textual document contents, and the fact frequencies in these text sections.

CHAPTER 4

An IR Model for the Contextualized Warehouse

This chapters proposes an IR model to retrieve the documents that describe an specific analysis context from the XML document warehouse, and to estimate the relevance of the facts quoted in these documents.

Nowadays, any application required to manipulate large collections of text-rich documents applies IR technology [7]. In an IR system the user supplies a query as a sequence of keywords which describe the contents of the documents to retrieve. The result is a list of documents ordered by their relevance to the established query. Recent proposals in the field of IR include language modeling [69] and relevance modeling [33]. Language modeling represents each document as a language model. Thus, documents are ranked according to the probability of obtaining the query keywords when taking random samples from their corresponding language models. Relevance modeling estimates the joint probability of the query's keywords and the document words over the set of documents deemed relevant for that query. Language and relevance modeling outperform traditional IR models in many cases. One of the current hot topics in IR research is retrieval of XML data [13, 24]. These approaches combine both keyword-based and structural retrieval conditions.

The retrieval model presented in this chapter relies on relevance modeling mainly because of two reasons. First, relevance modeling provides a formal background based on probability theory, which is also well-suited for OLAP operations. Second, relevance modeling deals with sets of relevant documents instead of single documents, which seems more appropriate for representing the contexts of the facts in a data warehouse. We estimate the relevance of a fact by the probability that the fact is described in the set of documents relevant to the specified analysis context. In the retrieval model the usual tree-like representation of XML documents is chosen, so that the previous work concerning the evaluation of path expressions [14, 4] can be easily applied to the model.

This chapter is organized as follows. Section 4.1 reviews the classical IR models and the more recent proposals on language modeling and relevance modeling. Section 4.2 presents the retrieval model for the contextualized warehouse and points out some interesting properties of this model. Section 4.3 evaluates the model with two different test document collections. Finally, Section 4.4 discusses some conclusions.

4.1 IR Models

IR queries are typically expressed as a sequence of keywords that describe the user's information needs. The result is a set of documents ranked by their relevance to the query. This relevance measures how well the document satisfies the user's information needs. In order to rank the documents, IR systems assume some retrieval model. The definition of a retrieval model comprises three elements [38]:

- the representation of the documents,
- the representation of the queries,
- and a function that measures the relevance of the documents with respect to a query

Different frameworks, e.g., set-theory, algebra, or probabilistics, have been proposed in the literature to formalize the previous three elements, leading to different retrieval models. This section summarizes the main ideas behind the classical retrieval models: the boolean, vector and probabilistic models. The reader will find a detailed description of these and other important retrieval models in [7]. The section concludes by sketching some notions of the more recent language modeling and relevance modeling approaches.

Classical IR models usually consider that each document is described by a set of keywords called *index terms*. An *index term* is a document word whose semantics helps in remembering the document's main themes. However, notice that not all the document terms will be equally useful to decide the contents of a document. This effect is captured by assigning a numerical *weight* to each index term of a document. Let k_i be an index term and d_j be a document, then $w_{i,j} \geq 0$ is the weight associated with the pair (k_i, d_j). This weight quantifies the importance of the index term for describing the document contents.

4.1.1 The Boolean Model

The boolean model is based on set theory and boolean algebra. It is one of the simplest retrieval models. It was adopted by many of the early bibliographic systems, and it is still the dominant model in the document database systems.

The boolean model considers that index terms are either present or absent in a document. Documents are represented as binary weighted vectors, i.e., $\overrightarrow{d}_j = (w_{1,j}, w_{2,j}, \ldots)$, where $w_{i,j} \in \{0,1\}$. Let g_i return the weight associated with the index k_i in any vector (i.e., $g_i(\overrightarrow{d}_j) = w_{i,j}$). A query is a boolean expression of index terms, e.g., $Q = k_a \wedge (k_b \vee \neg k_c)$. In order to decide the relevance of a document to a query, queries are represented as a disjunction of conjunctive vectors (i.e., in disjunctive normal form - DNF). For instance, the query Q is represented by $Q_{dnf} = (1,1,1) \vee (1,1,0) \vee (1,0,0)$, where each component is a binary weighted vector associated with the tuple of terms (k_a, k_b, k_c). These binary weighted vectors are called the conjunctive components ($\overrightarrow{q}_{cc}$) of Q_{dnf}. The similarity of a document $\overrightarrow{d}_j$ to a query Q is defined as:

$$sim(\overrightarrow{d}_j, Q) = \begin{cases} 1 & \text{if } \exists \overrightarrow{q}_{cc}\ in\ Q_{dnf} \mid \forall k_i, g_i(\overrightarrow{d}_j) = g_i(\overrightarrow{q}_{cc}) \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

The boolean model predicts that a document is relevant when $sim(\overrightarrow{d}_j, Q) = 1$. That is, only the documents that strictly satisfy the boolean expression are deemed to be relevant. Otherwise, the document is considered to be non-relevant ($sim(\overrightarrow{d}_j, Q) = 0$). This is the major drawback of the model. No partial matching nor relevance ranking is provided. This approach will usually retrieve too few or too many documents. Nowadays, it is well known that (non-binary) index term weighting can lead to substantial improvements in retrieval performance.

4.1.2 The Vector Model

The vector model [75] represents both documents and queries as t-dimensional vectors, i.e., $\overrightarrow{d}_j = (w_{1,j}, w_{2,j}, \ldots, w_{t,j})$ and $\overrightarrow{Q} = (w_{1,q}, w_{2,q}, \ldots, w_{t,q})$, where t is the total number of different index terms in the collection. Now the weights are considered positive and non-binary, i.e., $w_{i,j} \geq 0$. In order to compute the degree of similarity between the document and the query vectors, different measures have been proposed. The most widely used is the cosine of the angle formed by these two vectors:

$$sim(\overrightarrow{d}_j, \overrightarrow{Q}) = \frac{\overrightarrow{d}_j \bullet \overrightarrow{Q}}{\mid \overrightarrow{d}_j \mid \times \mid \overrightarrow{Q} \mid} = \frac{\sum_{i=1}^{t} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \times \sqrt{\sum_{i=1}^{t} w_{i,q}^2}} \tag{4.2}$$

In formula (4.2) $sim(\overrightarrow{d}_j, \overrightarrow{Q})$ varies from 0 to 1. Instead of predicting whether a document is relevant or not, the vector model returns a list of documents sorted by their degree of similarity to the query. A document might be retrieved even if it matches the query only partially. A threshold value can be established to discard the documents with a degree of similarity under that threshold.

Index term weights can be estimated in many different ways [76]. Distinct approaches follow different intuitions to determine which are the important terms. The tf/idf weighting is one of the most popular methods. The tf/idf method assigns a high weight to those index terms that occur frequently in the document, but do not appear in many other documents of the collection. The intuition is that the frequent terms within a document are good representatives for the document. However, the terms that occur in many documents are not useful for distinguishing relevant from non-relevant documents.

Formally, let N be the total number of documents in the collection and n_i be the number of documents in which the index term k_i appears. Let $freq_{i,j}$ be the frequency of the term k_i in the document d_j (i.e., the number of times that the term k_i is mentioned in the text of the document d_j). The term frequency factor, *tf-factor*, is the normalized frequency $f_{i,j}$ of the term k_i in the document d_j:

$$f_{i,j} = \frac{freq_{i,j}}{max_l freq_{l,j}} \tag{4.3}$$

In the denominator of formula 4.3, l represents any index term mentioned in the document d_j. The inverse document frequency factor, *idf-factor* is given by:

$$idf_i = \log \frac{N}{n_i} \tag{4.4}$$

Finally, the tf/idf weighting scheme assigns the following weight to the term k_i in the document d_j:

$$w_{i,j} = f_{i,j} \times idf_i \tag{4.5}$$

The tf/idf weighting approach of the vector model improves retrieval performance of the boolean model. The partial matching and ranking strategy allow the retrieval of documents that approximate the query conditions. The model is simple and fast. For these reasons, the vector model is nowadays one of the most popular retrieval models. The main disadvantage of the vector model is that no formal framework is provided to calculate the index term weights. The tf/idf weighting is an empirical model and the different weighting approaches so far proposed have been largely heuristic.

4.1.3 Probabilistic Models

The probabilistic models are based on the *probability ranking principle* [71], which suggests ranking the documents by the ratio:

$$\frac{P(d_j \mid R)}{P(d_j \mid \overline{R})} \tag{4.6}$$

In these models R represents the ideal set of documents relevant to the user's query, and $\overline{R}$ the set of non-relevant documents. Then, $P(d_j \mid R)$ stands for the probability of randomly selecting the document d_j from the ideal set R of relevant documents, and $P(d_j \mid \overline{R})$ for the probability of selecting it from the set of non-relevant documents $\overline{R}$.

The estimation of $P(d_j \mid R)$ differs in various models. The Binary Independence Model [72] represent each document as a binary weighted vector, i.e., $\overrightarrow{d}_j = (w_{1,j}, w_{2,j}, \ldots w_{t,j})$, $w_{i,j} \in \{0,1\}$, and computes the probability $P(\overrightarrow{d}_j \mid R)$ with the following formula:

$$P(\overrightarrow{d}_j \mid R) = \prod_{g_i(\overrightarrow{d}_j)=1} P(k_i \mid R) \prod_{g_i(\overrightarrow{d}_j)=0} (1 - P(k_i \mid R)) \tag{4.7}$$

In formula (4.7), $P(k_i \mid R)$ is the probability that the index term k_i is present in a document randomly selected from the ideal set of relevant documents R. Analogously, the probability $P(\overrightarrow{d}_j \mid \overline{R})$ is calculated by:

$$P(\overrightarrow{d}_j \mid \overline{R}) = \prod_{g_i(\overrightarrow{d}_j)=0} P(k_i \mid \overline{R}) \prod_{g_i(\overrightarrow{d}_j)=0} (1 - P(k_i \mid \overline{R})) \tag{4.8}$$

Since the ideal set of relevant documents R is not known, the probabilities $P(k_i \mid R)$ and $P(k_i \mid \overline{R})$ are estimated using heuristic techniques. For example, consider the following process:

1. We may assume that $P(k_i \mid R)$ is constant for all the index terms k_i (e.g., $P(k_i \mid R) = 0.5$). On the other hand, since for typical queries almost every document in the collection is non-relevant, we can approximate the distribution of index terms among the non-relevant documents by the distribution of index terms among all the documents in the collection (i.e., $P(k_i \mid \overline{R}) = n_i/N$). Given this initial guess, we can retrieve the documents that contain the some of the query terms, and rank the retrieved documents according to formula (4.6).

2. Let RQ be the set composed of the top r ranked documents. RQ stands for documents Relevant to the Query. Let $RQ_i \subseteq RQ$ be the subset of those documents in RQ that contain the index term k_i. That is, $RQ_i = \{\overrightarrow{d}_j \in RQ \mid g_i(\overrightarrow{d}_j) = 1\}$.

3. In order to improve the ranking, we can now approximate $P(k_i \mid R)$ by the distribution of index terms k_i in the documents of RQ, and $P(k_i \mid \overline{R})$ by considering the non-relevant documents as those that are not present in RQ. That is, $P(k_i \mid R) = \frac{|RQ_i|}{|RQ|}$ and $P(k_i \mid \overline{R}) = \frac{n_i - |RQ_i|}{N - |RQ|}$.

4. Next, we can apply formula (4.6) to rank the documents in RQ again.

By repeating recursively the steps 2 to 4, we will improve the estimation of the probabilities $P(k_i \mid R)$ and $P(k_i \mid \overline{R})$.

Like the boolean model, the main disadvantage of the the Binary Independence Model is that it does not model the frequencies of the index terms within the documents (i.e., all the weights are binary). Other probabilistic models go a step further and take into account the index term frequencies [73].

Heuristic estimation differences aside, the common feature of the classical probabilistic models is their notion of an ideal set of relevant documents R, and their attempt to estimate the probabilities $P(k_i \mid R)$ of the index terms in this ideal set. The major drawback of the probabilistic models is the need to initially guess which documents are in RQ. However, sometimes it will be possible to use the assistance of the user for building this initial set of relevant documents. There is not a clear consensus as to whether the probabilistic model outperforms the vector model or not. Anyway, the probabilistic models provide a theoretically well-founded framework for the construction of IR systems.

4.1.4 Language Modeling

The work on language modeling considers each document as a language model d_j. One can see a language model d_j as a black box from which we can repeatly sample index terms. The documents are then ranked according to the probability $P(Q \mid d_j)$ of obtaining the query keywords when randomly sampling from the respective language model.

The calculation of the probability $P(Q \mid d_j)$ differs from model to model. In [69] the query is represented as a binary weighted vector, i.e., $\overrightarrow{Q}_j = (w_{1,q}, w_{2,q}, \ldots w_{t,q})$, $w_{i,q} \in \{0, 1\}$, and the probability $P(\overrightarrow{Q} \mid d_j)$ is calculated by formula (4.9):

$$P(\overrightarrow{Q}_j \mid d_j) = \prod_{g_i(\overrightarrow{Q}_j)=1} P(k_i \mid d_j) \prod_{g_i(\overrightarrow{Q}_j)=0} (1 - P(k_i \mid d_j)) \tag{4.9}$$

In formula (4.9) $P(k_i \mid d_j)$ is the probability of sampling the index term k_i from the language model d_j (i.e., the probability of observing this term in the document). Notice the similarity between formula (4.9) and the Binary Independence Model formula (4.7). However, whereas in the probabilistic models it is hard to estimate the probabilities $P(w \mid R)$, since the ideal set R is a priori unknown, under the language modeling approach an accurate estimation of $P(k_i \mid d_j)$ is possible, since we exactly know the textual contents of the document modeled by d_j.

Other works on language modeling [80] represent the query $Q = q_1 q_2 \ldots q_n$ as a sequence of independent keywords q_i. Let $q_i \in Q$ mean that the keyword q_i appears in the sequence Q. These works compute the probability $P(Q \mid d_j)$ by:

$$P(Q \mid d_j) = \prod_{q_i \in Q} P(q_i \mid d_j) \tag{4.10}$$

The authors of [80] propose to approach the probability $P(q_i \mid d_j)$ by smoothing the relative frequency of the query keyword in the document. The objective of this approach is to avoid probabilities equal to zero in $P(Q \mid d_j)$ when a document does not contain all the query keywords. They make the assumption that finding a keyword in a document might be at least as probable as observing it in the entire collection of documents, and estimate this probability as follows:

$$P(q_i \mid d_j) = \lambda \frac{freq(q_i, d_j)}{\mid d_j \mid_t} + (1 - \lambda) \frac{ctf_{q_i}}{coll\_size_t} \tag{4.11}$$

In formula (4.11) $freq(q_i, d_j)$ is the frequency of the keyword q_i in the document represented by the language model d_j. $|d_j|_t$ denotes the total number of index terms in the document. ctf_{q_i} is the number of times that the query keyword q_i occurs in all the documents of the collection, and $coll\_size_t$ the total number of terms in the collection. The λ factor is the smoothing parameter, and its value is determined empirically, $\lambda \in [0, 1]$.

The retrieval model proposed in this thesis also models the queries as sequences of keywords and follows a similar approach to compute the relevance of the documents.

4.1.5 Relevance-Based Language Models

Many popular IR techniques, such as the relevance feedback, have a very intuitive interpretation in the classical probabilistic models. These techniques require modifying the sample set of relevant documents according to the user's relevance judgments. However, they are difficult to integrate into the language modeling framework where there is not such a notion of relevant set of documents.

The work on relevance modeling returns to the probabilistic models view of the document ranking problem, i.e, the estimation of the probability $P(k_i \mid R)$ of sampling the index term k_i from the ideal relevant set of relevant documents R. They make the assumption that in the absence of training data and given a query $Q = q_i q_2 \ldots q_n$, the probability $P(k_i \mid R)$ can be approximated by the probability $P(k_i \mid q_1 q_2 \ldots q_n)$ of the co-occurrence of between the sequence of query keywords Q and the index term k_i [33], that is:

$$P(k_i \mid R) \approx P(k_i \mid Q) = \frac{P(k_i, Q)}{P(Q)} \tag{4.12}$$

Let $M = \{d_j\}$ be the finite universe of language models d_j that represent the documents in the collection. In order to compute the joint probability $P(k_i, Q)$, they assume independence between the index term k_i and the query keywords Q, and compute the total probability of sampling them from each language model in M:

$$P(k_i, Q) = \sum_{d_j \in M} P(d_j) P(k_i \mid d_j) P(Q \mid d_j) \tag{4.13}$$

Formula (4.13) can be interpreted as follows: $P(d_j)$ is the probability of selecting a language model d_j from the set M, $P(k_i, d_j)$ is the probability of sampling the index term k_i from the language model d_j, and $P(Q \mid d_j)$ the probability of sampling the query keywords Q from the same language model. Like in most modeling approaches [80], the probability $P(k_i \mid d_j)$ can be estimated by the smoothed relative frequency of the index term in the document. See formula (4.11).

By applying the conditional probability formula, the probability $P(Q \mid d_j)$ can be computed by:

$$P(Q \mid d_j) = \frac{P(d_j \mid Q) P(Q)}{P(d_j)} \tag{4.14}$$

Replacing $P(Q \mid d_j)$ by the previous expression in formula (4.13), we obtain:

$$P(k_i, Q) = \sum_{d_j \in M} P(k_i \mid d_j) P(d_j \mid Q) P(Q) \tag{4.15}$$

Finally, by including formula (4.15) in the expression (4.12), the approximation of the probability $P(k_i \mid R)$ results in:

$$P(k_i \mid R) \approx \sum_{d_j \in M} P(k_i \mid d_j) P(d_j \mid Q) \tag{4.16}$$

In oder to implement the ideas behind the relevance models in an IR system, the set M is restricted to only contain the language models of the r top-ranked documents retrieved by the query Q. The systems performs the following process:

1. Retrieve from the document collection the documents that contain all or most of the query keywords and rank the documents according to the probability $P(d_j \mid Q)$ that they are relevant to the query. As formula (4.14) shows, this is equivalent to rank the documents by the probability

$P(Q \mid d_j)$, since the probabilities $P(d_j)$ and $P(Q)$ are constants across queries. The language modeling formula (4.10) proposed in [80] can be used for this purpose. Let RQ be the set composed of the top r ranked documents.

2. Approximate the probability $P(k_i \mid R)$ of sampling an index term k_i from the ideal set of relevant documents R by the probability $P(k_i \mid RQ)$ of sampling it from the set of relevant documents RQ.

$$P(k_i \mid R) \approx P(k_i \mid RQ) \approx \sum_{d_j \in RQ} P(k_i \mid d_j) P(d_j \mid Q) \tag{4.17}$$

The main contribution of relevance modeling is the probabilistic approach discussed above to estimate $P(k_i \mid R)$ using the query alone, which has been done in a heuristic fashion in previous works. This approximation to $P(k_i \mid R)$ can be later used for applying the probability ranking principle of formula (4.6). For instance, the authors of [33] represent the documents as a sequence of independent index terms (let $k_i \in d_j$ be each one of these index terms) and propose to rank the documents by:

$$\frac{P(d_j \mid R)}{P(d_j \mid \overline{R})} \sim \prod_{k_i \in d_j} \frac{P(k_i \mid R)}{P(k_i \mid \overline{R})} \tag{4.18}$$

The models of relevance has been shown to outperform base-line language modeling and tf/idf IR systems in TREC ad-hoc retrieval and TDT topic tracking tasks [33]. Moreover, relevance modeling provides a theoretically-well founded framework where not only is possible to calculate the probability of sampling an index term from the set of documents relevant to an IR query, but also to estimate the probability of observing any arbitrary type of object described in this set of relevant documents. For example, in [34] relevance models are applied in the image retrieval task to compute the joint probability of sampling a set of image features and a set of image annotation index terms.

The notion of the set RQ of documents relevant to an IR query can be used for representing the context of analysis in a contextualized warehouse. This thesis adapts the relevance modeling techniques to estimate the probability of observing a corporate fact described in the set of documents relevant to the context of analysis.

4.2 The Retrieval Model

This section proposes a retrieval model based on relevance modeling techniques [33] specifically intended for the contextualized warehouse. First, it describes how the documents are represented in the XML document warehouse. Afterwards, it studies the document retrieval process (i.e., the computation of the document relevance and the evaluation of queries in the XML document warehouse). Next, it shows how the relevance of the facts described within the retrieved documents is estimated. Finally, the section discusses some interesting properties of the approach.

4.2.1 Documents Representation

We represent the documents of the warehouse as hierarchical trees built by nesting their elements. By choosing the usual tree representation of XML documents, all the previous work concerning the evaluation of path expressions over XML documents [14, 4] can be easily applied to our model.

Definition 4.1. *A document is modeled as a tree of nodes. We denote by d_j a node of the tree (d_j stands for document node).*

There are four different types of document nodes, namely: element, attribute, textual contents and fact collection nodes. Let type(d_j) be a function that returns the type a document node d_j, type : $\{d_j\} \rightarrow \{element, attribute, textual\ contents, fact\ collection\}$.

Example 4.1. The left-side of Figure 4.1 shows the tree representation for the example document of Figure 3.3. Each document node d_j represents a piece of the original document. The parent-child relationship between the document nodes reflects the logical structure of the original document.

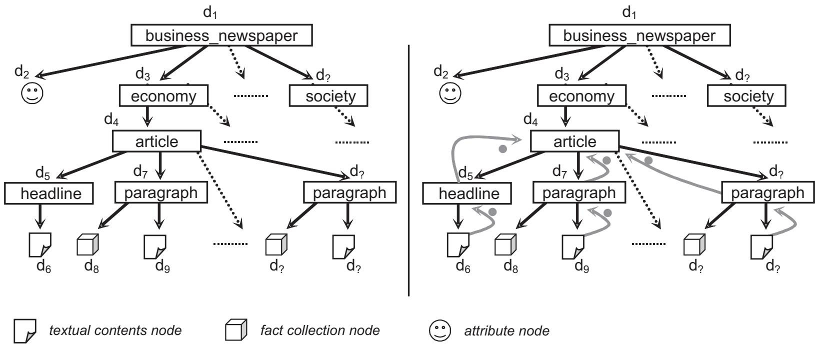


Figure 4.1: Representation of the document of Figure 3.3 (left-side) and aggregated term frequency calculus (right-side)

Next, we formally define the element, attribute, textual contents and fact collection nodes.

Element Nodes

An *element* node depicts an element of the original document, i.e., a piece of the document within a pair of matching tags.

Definition 4.2. *An element node d_j is a document node that represents an element of the original document. Let $name(d_j)$ be a function that returns the tag name of the element represented by the element node d_j.*

Element nodes have an ordered list of child document nodes. These child document nodes can either be attribute, textual contents, fact collection or even element nodes. We write $d'_j \prec d_j$ meaning that the document node d'_j is a child of the element node d_j. A document node d''_j is a descendant of the element node d_j, written $d''_j \prec\prec d_j$, iff $d''_j \prec d_j$ or $d''_j \prec\prec d'_j \wedge d'_j \prec d_j$.

Example 4.2. The element node d_3 depicts the `economy` element of the document, $type(d_3) = element$ and $name(d_3) = economy$. The element node d_7 represents the `paragraph` element of the original document, and $d_7 \prec\prec d_3$ (i.e., this paragraph is found within the economy section). The `article` element is represented by the element node d_4, $d_4 \prec d_3$.

Attribute nodes

We model the tags attributes of the original documents as *attribute* nodes.

Definition 4.3. *An attribute node d_j is a document node that depicts an attribute of an element of the original document. We represent an attribute node as a tuple $d_j = (name, value)$ where: name is the attribute name and value the corresponding value.*

An attribute node d_j is always a leaf node, it has no children node list.

Example 4.3. The element node d_1 depicts the element `business_newspaper` (see Figure 4.1 and Figure 3.3). d_2 is an attribute node, child of the document node d_1, that represents the `date` attribute of the `business_newspaper` element. Then, $type(d_2) = attribute$, $d_2 \prec d_1$ and $d_2 = (date, \text{“}Dec.1,\ 1998'')$.

Textual contents nodes

The text sections contained within an element of the original document are modeled as *textual contents* nodes.

Definition 4.4. *A textual contents node d_j is a document node that represents a character data section of the original document. A textual contents node is a set of 2-tuples $d_j = \{(k_i, tfreq_{k_i,d_j})\}$, where: k_i is an index term (or keyword) appearing in the text section represented by the node d_j and $tfreq_{k_i,d_j}$ is the frequency of the term k_i in this text section (i.e., the number of times that the term occurs in the text section). $tfreq$ stands for term frequency.*

The function $text : \{d_j\} \rightarrow 2^{\{k_i\}}$ returns the set of index terms mentioned in the text represented by a textual contents node d_j, i.e., $text(d_j) = \{k_i \mid \exists (k_i, tfreq_{k_i,d_j}) \in d_j\}$. If the index term k_i does not appear in the text represented by the textual contents node d_j, i.e., $k_i \notin text(d_j)$, then $tfreq_{k_i,d_j} = 0$.

A textual contents node d_j is always a leaf node.

Example 4.4. The character data section of the `paragraph` element of Figure 3.3 is represented by the textual contents node d_9, $type(d_9) = textual\ contents$. Since the term *crisis* is mentioned once in this paragraph ($tfreq_{crisis,d_9} = 1$), the term *financial* also occurs once, and *million* appears twice, we will have $d_9 = \{(financial, 1), (crisis, 1), (million, 2), \ldots\}$ and $text(d_9) = \{financial, crisis, million, \ldots\}$.

The textual contents node d_6 depicts the text section of the `headline` element, then $d_6 = \{(financial, 1), (crisis, 1), (hits, 1), (southeast, 1), (asian, 1), (market, 1)\}$. [1]

Sometimes it will be useful to manage all the text sections included within an arbitrary element node as if they were a single text portion (e.g., all the text of an article or an entire economy section). The *Text* function returns the set of index terms found in the text of an element node or any of its descendant elements. The *aggregated term frequency* function allows computing the frequency of the terms within an element node or any of its descendants. We formally define these functions below.

Definition 4.5. *Let d_j be an element node. Let $Text : \{d_j\} \rightarrow 2^{\{k_i\}}$ be a function that returns the set of index terms k_i mentioned in the text sections of the element represented by the element node d_j or any of its descendant elements.*

$$Text(d_j) = \bigcup_{d'_j \prec\prec d_j,\ type(d'_j) = textual\ contents} text(d'_j) \tag{4.19}$$

Example 4.5. If we evaluate the *Text* function on the element node d_4, we will obtain the index terms that appear in the document subtree under the `article` element. Thus, $Text(d_4) = text(d_6) \cup text(d_9) \cup \ldots = \{financial, crisis, hits, southeast, market, million, \ldots)\}$.

Definition 4.6. *Let d_j be an element node. The aggregated term frequency of a term k_i in the element node d_j is defined as:*

$$TFreq(k_i, d_j) = \begin{cases} tfreq_{k_i,d_j},\ \textit{if } type(d_j) = textual\ context; \\ \sum_{d'_j \in \prec d_j} TFreq(k_i, d'_j),\ otherwise \end{cases} \tag{4.20}$$

Example 4.6. We can use the aggregated term frequency function to calculate the frequency of the index term *crisis* within the `article` element (element node d_4). The right side of Figure 4.1 shows how this function is applied starting from the textual content nodes of the document subtree. The result is $TFreq(crisis, d_4) = tfreq_{crisis,d_6} + tfreq_{crisis,d_9} + \ldots = 1 + 1 + \ldots \geq 2$.

[1] The example assumes that neither stemming nor stop-word processing is performed.

Fact collection nodes

A *fact collection* node represents the facts described in a piece of text of the document. This piece of text can be either a complete text section, part of a text section, all the text included in an element, or a portion of the text contained within an element. The *fact collection* nodes are attached to the document tree as children of the element node that includes the corresponding piece of text. We first define a *fact* in the retrieval model, and then we address the formalization of the *fact collection* nodes.

Definition 4.7. *Let $D_1, D_2, \ldots, D_n$ be the dimensions and measures defined in the corporate warehouse. A fact f_i consists of an n-tuple of dimension and measure values $(e_1, e_2, \ldots e_n)$, where $e_j \in D_j$, meaning that each e_j is a value of the dimension/measure D_j.*

Given that the definition of dimension hierarchies is out of the scope of the retrieval model, here we simply consider a dimension as the flat set that includes all the members of the dimension hierarchy levels, as specified in the corporate warehouse schema. The measures are represented by their domain of possible values. Then, we model the facts as tuples over these flat sets of valid dimension/measure values.

Example 4.7. As explained along the example of Section 3.4, the text section of the `paragraph` element of Figure 3.3 describes the corporate facts f_4 and f_5 (see Table 3.1). In the retrieval model we represent these facts by $f_4 = (fo1, Japan, 1998/10, 300,000)$ and $f_5 = (fo2, Korea, 1998/11, 400,000)$, both $f_4, f_5 \in Products \times Customers \times Time \times Amount$.

Definition 4.8. *A fact collection node is a document node that represents the facts described in a piece of text the original document. We model a fact collection node as a set of 2-tuples $d_j = \{(f_i, ffreq_{f_i,d_j})\}$, where: f_i is a fact and $ffreq_{f_i,d_j}$ the frequency of its dimension values in the considered piece of text (i.e., the number of times that the dimension values of f_i are mentioned in that text). $ffreq$ stands for fact frequency.*

The function $facts : \{d_j\} \rightarrow 2^{\{f_i\}}$ returns the set of facts represented in the fact collection node d_j, i.e., $facts(d_j) = \{f_i \mid \exists (f_i, ffreq_{f_i,d_j}) \in d_j\}$.

A fact collection node d_j is always a leaf node.

Example 4.8. In Figure 4.1, the fact collection node d_8 represents the facts described within the text section of the `paragraph` element. Since the dimension values of the fact $f_4 = (fo1, Japan, 1998/10, 300,000)$ are mentioned three times in the paragraph (i.e., $ffreq_{f_4,d_8} = 3$) and the frequency of $f_5 = (fo2, Korea, 1998/11, 400,000)$ is also three, we will have $d_8 = \{((fo1, Japan, 1998/10, 300,000), 3), ((fo2, Korea, 1998/11, 400,000), 3)\}$ and $facts(d_8) = \{ (fo2, Korea, 1998/11, 400,000), (fo2, Korea, 1998/11, 400,000)\}$.

Like the *aggregated term frequency* and *Text* functions, the *aggregated fact frequency* and Facts functions can be used to represent the facts described by the subtree under any element node of document.

Definition 4.9. *Let d_j be an element node. Let $Facts : \{d_j\} \rightarrow 2^{\{f_i\}}$ be a function that returns the set of facts f_i described in the element represented by the element node d_j or any of its descendant elements.*

$$Facts(d_j) = \bigcup_{d'_j \prec\prec d_j, type(d'_j)=fact\ \ collection} facts(d'_j) \tag{4.21}$$

Definition 4.10. *Let d_j be an element node. The aggregated fact frequency of a fact f_i in the element node d_j is defined as:*

$$FFreq(f_i, d_j) = \begin{cases} ffreq_{f_i,d_j},\ \textit{if type}(d_j) = fact\ \textit{collection;} \\ \sum_{d'_j \prec d_j} FFreq(f_i, d'_j),\ \textit{otherwise} \end{cases} \tag{4.22}$$

4.2.2 Query Processing and Documents Relevance Calculus

This section details the query process in the XML document warehouse. Given the conditions used for selecting a context of analysis, it first explains how the relevant elements nodes are retrieved. Afterwards, it shows how the relevance of the element nodes to the retrieval conditions is estimated.

Remember Section 3.4. In order to select a context of analysis from the XML document warehouse two conditions are provided: an IR condition Q and a path expression $XPath$ [14]. The IR condition Q restricts the contents (the theme) of the document fragments to retrieve, whereas the path expression $XPath$ states within which document sections these fragments should be found. A path expression is a pattern on the structure of the XML documents. Evaluating the path expression consists in retrieving the document nodes that match the pattern. In general, a path expression can be used for addressing almost any part of an XML document. However, here we only consider the subclass of path expressions that retrieve element nodes.

Definition 4.11. *Let $Col = \{d_j\}$ be the set of all the element nodes d_j of all the documents in the XML document warehouse. Col stands for collection.*

A query in the XML document warehouse is a tuple $(Q, XPath)$, where: $Q = q_1q_2 \ldots q_n$ is an IR condition, consisting of a sequence of keywords q_i; and $XPath$ is a path expression [14].

Let xpath be a boolean function over the set of element nodes of the warehouse, xpath : $Col \rightarrow \{True, False\}$. xpath$(d_j)$ returns $True$ if the element node d_j is selected by the path expression $XPath$, and $False$ otherwise.

The result of evaluating the query $(XPath, Q)$ is the set of element nodes RQ. RQ stands for document fragments Relevant to the Query, and it is formally defined as follows:

$$
\begin{aligned}
RQ = \{d_j \in Col \mid \quad & XPath(d_j) \wedge \mid Text(d_j) \cap Q \mid \geq m \wedge \\
& \nexists d'_j, d'_j \prec\prec d_j \ (P(Q \mid d'_j) \geq P(Q \mid d_j)) \ \}
\end{aligned} \tag{4.23}
$$

Thus, RQ is the set of element nodes d_j that are selected by the path expression (i.e., $XPath(d_j)$), contain at least m query keywords ($|Text(d_j) \cap Q \mid \geq m$) and are more relevant than any of their descendant element nodes ($\forall d'_j, d'_j \prec\prec d_j, P(Q \mid d'_j) < P(Q \mid d_j)$).

Definition 4.12. *Let $Q = q_1 q_2 \dots q_n$ be an IR condition, consisting of a sequence of keywords q_i, and let d_j denote an element node. Let $|d_j \mid_t$ denote the total number of terms in $Text(d_j)$. Let ctf_{q_i} be the number of times that the query keyword q_i occurs in all the documents of the warehouse, and $coll\_size_t$ the total number of terms in all the documents of the warehouse.*

The relevance of the element node d_j to the IR condition Q is calculated by the probability $P(Q \mid d_j)$ of observing the query keywords in the element node:

$$
P(Q \mid d_j) = \prod_{q_i \in Q} P(q_i \mid d_j) \tag{4.24}
$$

$$
P(q_i \mid d_j) = \lambda \frac{TFreq(q_i, d_j)}{\mid d \mid_t} + (1 - \lambda) \frac{ctf_{q_i}}{coll\_size_t} \tag{4.25}
$$

By following a language modeling approach [80], we assume that the query keywords q_i are independent, and use formulas (4.24) and (4.25) for calculating $P(Q \mid d_j)$. Notice the similarity with the language modeling formulas (4.10) and (4.11). Formula (4.25) is based on the one proposed in [80]. However, by including the aggregated term frequency in the formula, we adapt it to work with element nodes (document subtrees) instead of with entire documents. The λ factor is called the smoothing parameter, as it avoids probabilities equal to zero when a document does not contain all the query keywords. The value of λ is determined empirically, $\lambda \in [0, 1]$.

The approach described above ensures that the element nodes in the result have the proper granularity level to describe the IR condition Q. For example, let $(Q, XPath)$ be a query in the document warehouse. First, the path expression $XPath$ selects a subset of the document subtrees of the warehouse. Let d_j be an element node representing an article, and let $d'_j \in\prec\prec d_j$ be an element node depicting the second paragraph of this article. Both d_j and d'_j were selected by $XPath$. Let us consider that d'_j is more relevant than d_j for the given IR condition Q, i.e., $P(Q \mid d_j) < P(Q \mid d'_j)$. This setting could happen, for example, when all the query keywords only occur in the second paragraph of the article. Thus, d'_j and d_j will have the same frequencies for the query keywords. That is, $TFreq(q_i, d_j) = TFreq(q_i, d'_j)$ $\forall q_i \in Q$. However, the article d_j comprises all the terms contained in d'_j plus all the terms of the rest of paragraphs. Then, $|d_j \mid_t > | d'_j \mid_t$, $P(q_i \mid d_j) < P(q_i \mid d'_j)\ \forall q_i \in Q$, see formula (4.25), and $P(Q \mid d_j) < P(Q \mid d'_j)$, see formula (4.24). Since the second paragraph is actually the document portion that better describes the information required by Q (i.e., it obtains the maximum relevance in the subtree), the entire article d_j will not be included in RQ, but we will instead insert the more specific and relevant paragraph d'_j.

4.2.3 Fact Relevance Calculus

Next, we show how to calculate the relevance of a fact with respect to selected context (i.e., to the specified IR condition). Intuitively, a fact will be relevant for the selected context if the fact is found in an element node which is also relevant for this context. We will consider that a fact is important in an element node if its dimension values occur frequently in the text sections of this element.

Given the set of relevant element nodes RQ returned by the XML document warehouse, the relevance of a fact is estimated as the probability of observing it in the text sections of the element nodes in RQ. The probability of finding a fact in an element node is determined by the frequency of the dimension values of this fact in the text sections of the element.

Definition 4.13. *Let $(Q, XPath)$ be a query, and let RQ be the set of element nodes relevant to this query. We estimate the relevance of a fact f_i by calculating the probability $P(f_i \mid RQ)$ of observing this fact in the set of element nodes RQ relevant to the query conditions:*

$$P(f_i \mid RQ) = \frac{\sum_{d_j \in RQ} P(f_i \mid d_j) P(Q \mid d_j)}{\sum_{d_j \in RQ} P(Q \mid d_j)} \tag{4.26}$$

$P(Q \mid d_j)$ is the probability of observing the query keywords in the element node d_j. This probability is computed as discussed in definition (4.12).

$P(f_i \mid d_j)$ is the probability of finding the fact f_i in an element node, which is estimated as follows:

$$P(f_i \mid d_j) = \frac{FFreq(f_i, d_j)}{\mid d_j \mid_f} \tag{4.27}$$

where $|d_j \mid f$ is the total number of dimension values found in the element node d_j.

The approach discussed above to compute the probability $P(f_i \mid RQ)$ is based on the relevance modeling techniques presented in Section 4.1.5. However, we have adapted these techniques to estimate the probability of facts instead of document terms. Next, we point out the major similarities and differences between the two approaches.

The probability $P(Q \mid d_j)$ can be expressed in terms of the probability $P(d_j \mid Q)$ that the element node d_j is relevant to the query Q, by applying the following conditional probability formula:

$$P(Q \mid d_j) = \frac{P(d_j \mid Q)P(Q)}{P(d_j)} \tag{4.28}$$

In formula (4.28) $P(Q)$ is the joint probability of sampling the query keywords from the set RQ of relevant element nodes. $P(d_j)$ denotes the probability of selecting an element node from this set. By including in formula (4.26) the expression of the probability $P(Q \mid d_j)$ of formula (4.28), we have that:

$$P(f_i \mid RQ) = \frac{\sum_{d_j \in RQ} P(f_i \mid d_j)P(d_j \mid Q)P(Q)}{P(d_j)\sum_{d_j \in RQ} P(Q \mid d_j)} \tag{4.29}$$

In order to estimate the probability $P(Q)$, we compute the total probability of observing the query keywords in each element node of the set RQ. See formula (4.30). Notice that the assumption that we make here is equivalent to the one made by the relevance modeling works in formula (4.13) to calculate the joint probability $P(k_i, Q)$.

$$P(Q) = \sum_{d_j \in RQ} P(Q \mid d_j)P(d_j) \tag{4.30}$$

By considering that the probability $P(d_j)$ is constant, and replacing the probability $P(Q)$ by the previous expression, we have that formula (4.29) is equivalent to:

$$P(f_i \mid RQ) = \sum_{d_j \in RQ} P(f_i \mid d_j)P(d_j \mid Q) \tag{4.31}$$

Notice the similarity between formula (4.31) and the relevance modeling formula (4.17) used for computing the probability $P(k_i \mid RQ)$. The difference comes in considering that whereas the ordinary relevance modeling proposals approached the probability $P(k_i \mid R)$ by the probability of sampling the index term k_i, once the query keywords Q have been sampled from the documents, i.e., $P(k_i \mid R) \approx P(k_i \mid Q)$, we approach the probability $P(f_i \mid R)$ by the probability of finding the fact f_i when the query keywords Q have been previously found in the element nodes, that is, $P(f_i \mid R) \approx P(f_i \mid Q)$.

Summarizing, given a query $(Q, XPath)$, and the set of element nodes RQ relevant to these conditions, our model provides the formula (4.26) to estimate the relevance of the facts described in RQ. Since the aggregated term and fact frequency functions have been used in the formulas (4.25) and (4.27), the nodes in RQ can represent any document subtree of the warehouse. The latter means that the document model supports IR-XPath queries, where the user can arbitrarily choose, by using path expressions, which are the documents subtrees under analysis. Then, as discussed at the end of the Section 4.2.2, the construction of RQ ensures that only the most relevant section of each selected subtree will be included in the result.

4.2.4 Properties of the Model and Quality

An interesting property of this approach is that the sum of the relevance values of all the facts is equal to one. This is formally exposed in Theorem 4.1.

Theorem 4.1. *Let $RQ = \{d_j\}$ be the set of relevant element nodes to a particular query (Q, XPath). Let $F = \{f_i\}$ be the set of all the facts of the corporate warehouse related to the element nodes of RQ. Then, $\sum_{f_i \in F} P(f_i \mid RQ) = 1$.*

Proof. Let F_{d_j} be the set of facts related to the element node $d_j \in RQ$. Since in formula (4.27) we use the relative frequency of the fact f_i in the element node d_j to estimate $P(f_i \mid d_j)$, the sum of the probabilities of all the facts related to an element node d_j is 1. That is:

$$\sum_{f_i \in F_{d_j}} P(f_i \mid d_j) = \frac{\sum_{f_i \in F_{d_j}} FFreq(f_i, d_j)}{\mid d_j \mid_f} = \frac{\mid d_j \mid_f}{\mid d_j \mid_f} = 1 \qquad (4.32)$$

Now, by applying formula (4.26) we have that:

$$\begin{aligned} \sum_{f_i \in F} P(f_i \mid RQ) &= \sum_{f_i \in F} \left(\frac{\sum_{d_j \in RQ} P(f_i \mid d_j) P(Q \mid d_j)}{\sum_{d_j \in RQ} P(Q \mid d_j)} \right) \\ &= \frac{\sum_{f_i \in F} \left(\sum_{d_j \in RQ} P(f_i \mid d_j) P(Q \mid d_j) \right)}{\sum_{d_j \in RQ} P(Q \mid d_j)} \qquad (4.33) \end{aligned}$$

Since the set of facts F comprises all the facts related to the nodes $d_j \in RQ$, that is, $F = \cup_{d_j \in RQ} F_{d_j}$, we can write:

$$\sum_{f_i \in F} P(f_i \mid RQ) = \frac{\sum_{d_j \in RQ} \left(\sum_{f_i \in F_d} P(f_i \mid d_j) P(Q \mid d_j) \right)}{\sum_{d_j \in RQ} P(Q \mid d_j)} \qquad (4.34)$$

Finally, by applying the equivalence (4.32), we have that:

$$\sum_{f_i \in F} P(f_i \mid RQ) = \frac{\sum_{d_j \in RQ} P(Q \mid d_j)}{\sum_{d_j \in RQ} P(Q \mid d_j)} = 1 \qquad (4.35)$$

□

It is nice to have such a property in the resulting set of facts. However, this implies that the relevance values of the facts in the result has been normalized. That is, no matter how suitable for the query the result is, the relevance of the facts is scaled to have its sum equal to one. However, not all the collections of documents are appropriate to answer all types of queries (e.g., by using a document collection about products manufacturing processes, we would hardly obtain good answers to queries about a financial crisis).

The denominator of formula (4.26) indicates how good the set of relevant element nodes RQ is for the selected context, since it measures the overall relevance of the element nodes that satisfy the conditions $(Q, XPath)$, that is, the sum of the probabilities of observing the query keywords in each element node of RQ. Thus, we propose the following formula as a quality measure of the query result in the document warehouse.

Definition 4.14. *Let RQ be the set of element nodes relevant for the query $(Q, XPath)$. The quality of the query result in the XML document warehouse is measured by:*

$$Quality = \sum_{d_j \in RQ} P(Q \mid d_j) \tag{4.36}$$

4.3 Experiments and Results

This section evaluates the retrieval model with two different document collections. The experiments in Section 4.3.1 are carried out on an articles database of the Spanish newspaper *El País* [2]. Section 4.3.2 shows the results for a subset of the TREC document collection [27].

4.3.1 *El País* Collection

The objectives of the first set of experiments experiments were: first, building an initial document corpus by following our document model; and second, given an IR query, testing if the proposed fact relevance ranking mechanism assigns the highest relevance index to the most relevant group of facts.

In this case, the document corpus consists of a collection of 30 digital issues of *El País* newspaper published during June 1999. These issues were gathered from the newspaper web site and stored in XML format. In the experiments we considered two different dimensions: *Location* and *Date*. The *Location* dimensions keeps a set of well-known world cities, countries and regions names. The *Date* dimension represent dates of the Gregorian calendar, months and years. By applying the shallow parsing and information extraction techniques presented in [37] and [68], we identified values for these two dimensions in the documents contents. In order to group the dimension values into facts, we used

[2] http://www.elpais.com

a simple heuristics to know, the *Location* and *Date* values found in the same sentence will constitute a fact.

In our experiments we considered the news items elements contained in the International sections of the newspapers (555 news items), and query them with a set of IR queries about different topics. In order to give more importance to the occurrence of the query keywords in the selected news items, rather than in the global collection, we set the parameter λ of the formula (4.25) to 0.9. However, a deeper study of the influence of the λ parameter in the result remains to be done.

For each IR query, we applied the process described in Section 4.2.2 to estimate the relevance of the news items, and chose the news items at the top of the relevance ranking to build RQ. Regarding the number of news items to include in RQ, in our preliminary experiments we observed that a small size of RQ led to imprecise results. In our opinion, this occurs because facts do not use to co-occur in different documents in a small set of samples. In addition, there are a set of important events which are transversally reported by many news items, independently of their main subject. If the size of RQ is too big, these important events may become the main topic of the element nodes at RQ. In our experiments, the best performance was obtained by keeping the size of RQ between 20 and 60.

The Tables 4.2, 4.3 and 4.1 show the top-ranked facts returned by the retrieval model for different IR queries. As it can be noticed, all the returned facts are relevant to the stated queries, and they correspond to the dates and places where they actually occurred. In the results we found a considerable number of incomplete facts (i.e., without the *Location* or the *Date* value). Notice that this problem only appears when the facts are built from the documents alone. As discussed in Section 3.4, in the contextualized warehouse setting, the (possibly incomplete) facts found in the documents are used for relating these documents to the (complete) facts selected from the corporate warehouse.

Facts	**Relev.**	**Explanation**
(/11/1999, $LaHabana$)	0.0583	Summit of the Iberoamerican states in La Habana
(18/06/1999, $Cologne$)	0.0565	G-8 Summit in Cologne
(28/06/1999, $RioJaneiro$)	0.0526	Meeting European Union and Latin American states in Rio Janeiro
(29/06/1999, $RioJaneiro$)	0.0515	Meeting European Union and Latin American states in Rio Janeiro
(16/06/1999, $Cologne$)	0.0512	G-8 Summit in Cologne

Table 4.1: Top-ranked facts for the query $Q = summit$ ($\lambda = 0.9$, size of RQ between 20 and 60)

Facts	Relev.	Explanation
(10/06/1999, *Kosovo*)	0.0893	UNO finishes bombardments in Kosovo
(12/06/1999, *Kosovo*)	0.0882	Multinational NATO forces (KFOR) get into Kosovo
(15/06/1999, *Kosovo*)	0.0863	Serb radical party protests against KFOR
(31/03/1999, *Macedonia*)	0.0504	Three American soldiers were captured in Macedonia *(news items may describe facts from the past)*
(27/06/1999, *Algeria*)	0.0405	On 27th June, Government officials estimated that more than 100,000 Algerians died during the war against the Armed Islamic Group
(06/06/1999, *Guatemala*)	0.0381	Human Rights Organizations claim judgments for the crimes at the Guatemalaś civil war
(16/06/1999, *Kumanovo*)	0.0263	The agreement on the retreat of the Serb forces from Kosovo is signed at Kumanovo
(09/06/1999, *Serbia*)	0.0185	Javier Solanaś declarations about the Serbian situation
(07/06/1999, *Brussels*)	0.0143	NATO declarations at Brussels regarding the bombardments in Kosovo

Table 4.2: Top-ranked facts for the query $Q = civil, conflict$ ($\lambda = 0.9$, size of RQ between 20 and 60)

Facts	Relev.	Explanation
(1994, *Holland*)	0.0770	European Elections
(1994, *Spain*)	0.0753	European Elections
(1994, *Ireland*)	0.0709	European Elections
(2000, *USA*)	0.0599	Next USA Elections
(13/06/1999, *Indonesia*)	0.0411	Elections in Indonesia
(2/06/1999, *SouthAfrica*)	0.0398	Elections in South Africa

Table 4.3: Top-ranked facts for the query $Q = election$ ($\lambda = 0.9$, size of RQ between 20 and 60)

4.3.2 *WSJ* TREC Collection

This section evaluates the retrieval model with a larger document set, the *Wall Street Journal* (WSJ) subcollection of the TREC test collection [27].

The TREC collection is considered to be the reference test collection in IR. TREC stands for Text REtrieval Conference, a yearly conference dedicated to experimentation in IR. For each TREC conference, a set of reference experiments is designed, which are run by the participants for comparing their retrieval systems. The collection is referred as TREC, since it is the corpus used in these experiments.

The TREC collection comprises a set of documents, a set of example information requests (called *topics* in TREC), and an indication of which documents should be retrieved in response to each topic (relevance judgments). The document set includes documents from different sources like Wall Street Journal, Financial Times or US Patents. The primary TREC document collections contain 2 to 3 gigabytes of text and 500,000 to 1,000,000 documents. Each example TREC topic is a textual description of an information need in natural language. The queries, i.e., the actual input of the tested retrieval system, are generated by using these textual descriptions. Three different categories of query construction approaches are considered in the TREC evaluations: automatic (the queries are automatically constructed from the textual descriptions of the topics), manual (manual specification of queries) and interactive (interactive techniques are used for building the queries). For each topic, the ideal subset of documents relevant for the topic is provided.

In our experiments we considered the 1990-WSJ subcollection from TREC disk 2, a total of 21,705 news articles of the Wall Street Journal published during 1990. Here we do not model the structural components the WSJ articles (e.g., headline, paragraphs, etc.), but consider each article as a single piece of text instead. The reason is that TREC evaluations are typically focused on text plain documents, and the relevance judgments (the ideal set of documents to retrieve) are only provided for entire documents, in this case, complete WSJ news articles. However, the experimentation is still interesting, since it allows us to test the IR part of the query process and the facts relevance ranking formula. Evaluating the structural (XML-like) part of the document retrieval strategy is future work.

The news articles of the WSJ subcollection have attached some metadata. These metadata contain, among other information, the date of publication of the article, and the list of companies reported by the news article. By combining the date of publication and company list of each article, we built a $(Date, Company)$ fact database. For each fact, we also kept the news articles were the corresponding $(Date, Company)$ pair was found. Thus, in this case, our experiments involved two dimensions, the *Date* and the *Companies* dimensions. In the *Companies* dimension, the companies described by the WSJ articles are organized into $Industries$, which are in turn classified into $Sectors$.

The correspondence between companies, industries and sectors is based on the Yahoo Finance [3] companies classification.

We selected 16 topics from the TREC-2 and TREC-3 conferences. We chose the topics that have at least 20 documents in the provided ideal set of documents relevant for the topic. We make such a restriction to ensure that the set of relevant documents is big enough to find several samples of the dimension values relevant for the query, as discussed in the previous section. Furthermore, we examined the textual description of each selected topic and determined the industry that is most likely related to the thematic of the topic. That is, the industry of the companies that are expected to be found at the top-ranked the facts for each topic.

Topic #	**Title**	**Industry**
109	Find Innovative Companies	*Software & Computer Services*
112	Funding Biotechnology	*Biotechnology*
124	Alternatives to Traditional Cancer Therapies	*Health Care Equipment & Services*
133	Hubble Space Telescope	*Aerospace & Defense*
135	Possible Contributions of Gene Mapping to Medicine	*Biotechnology*
137	Expansion in the U.S. Theme Park Industry	*Media*
143	Why Protect U.S. Farmers?	*Food Producers*
152	Accusations of Cheating by Contractors on U.S. Defense Projects	*Aerospace & Defense*
154	Oil Spills	*Oil & Gas Producers*
162	Automobile Recalls	*Automobiles & Parts*
165	Tobacco company advertising and the young	*Tobacco*
173	Smoking Bans	*Tobacco*
179	U. S. Restaurants in Foreign Lands	*Restaurants & Bars*
183	Asbestos Related Lawsuits	*Construction & Materials*
187	Signs of the Demise of Independent Publishing	*Media*
198	Gene Therapy and Its Benefits to Humankind	*Biotechnology*

Table 4.4: Topic number, title and expected top-ranked industry of the TREC topics selected for the experiment

Table 4.4 shows the topic number, title and expected top-ranked industry of the TREC topic set considered in our experiments. For example, as this table

[3] http://finance.yahoo.com

shows, the expected most relevant industry for the TREC topic number 198, entitled "Gene Therapy and Its Benefits to Humankind", is *Biotechnology*.

Next, we show how we constructed the context of analysis (i.e., the set RQ of relevant documents) for the test topics. For each topic, we specified an IR query, and then we retrieved the set RQ of documents relevant for this query, as discussed in Section 4.2.2. Like in the experiments of the previous section, we set the λ parameter of formula (4.25) to 0.9. The query keywords were interactively selected to eventually reach an acceptable precision versus recall figure [7].

Let R be the ideal set of documents judged to be relevant. Precision measures the fraction of the retrieved documents that are actually relevant, i.e., $\frac{|RQ \cap R|}{|RQ|}$. Recall depicts the coverage of result, in terms of the fraction of documents retrieved from the ideal relevant set, $\frac{|RQ \cap R|}{|R|}$. Finding a good compromise between precision and recall implies a good document retrieval performance. Typically, an "acceptable" retrieval performance is considered to be achieved when the precision is over 40% at low recall values, e.g, 20%; greater than 30% for a recall of 50%; and no lower than 10% for high recall percentages like 80%. See for example the evaluations of [80], [33] and [27].

Figure 4.2 illustrates the average precision values obtained at the 11 standard recall levels for the selected topics. The percentages are over the acceptable margins quoted above. Table 4.5 details these precision values, as well as the resulting average R-Precision.

Recall level	**Avg. Precision**
0.0	0.8403
0.1	0.6671
0.2	0.5690
0.3	0.5283
0.4	0.4472
0.5	0.4167
0.6	0.3728
0.7	0.3525
0.8	0.2556
0.9	0.1697
1.0	0.0057
Avg.	0.4205
Avg. R-Pr.	0.4558

Table 4.5: Average precision versus recall and average R-Precision obtained for the selected TREC topics

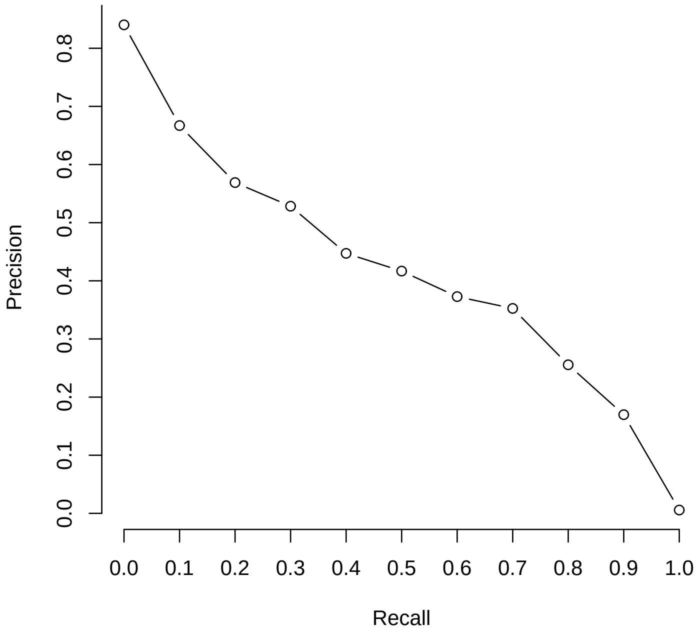


Figure 4.2: Average precision versus recall obtained for the selected TREC topics

One may argue that this interactive selection of query keywords will blur the test results. However, it is important to emphasize that the objective here is not to evaluate the document retrieval performance. The document relevance ranking formulas presented in Section 4.2.2 are based on those of language modeling, that have already been shown to obtain good performance results [80]. The final objective is to evaluate the proposed facts relevance ranking approach. For this purpose, we need an acceptable description of each selected topic in the corresponding set of relevant documents RQ. Even more, a real (out of the laboratory) scenario is mostly interactive, where user is able to successively improve the IR queries to accurately describe the contents of analysis.

The R-Precision is a useful parameter for measuring the quality of the result set RQ for each individual topic, when the ideal set R of documents judged to be relevant is known [7]. Given $|R\,|$, the number of documents in the ideal set R, it calculates the precision for the $|R\,|$ top-ranked documents in RQ. Table 4.6 shows the R-Precision values obtained for each topic. Figure 4.3 depicts the corresponding the R-Precision histogram.

Topic #	R-Precision
109	0.2727
112	0.2500
124	0.5000
133	0.4762
135	0.7500
137	0.7083
143	0.4615
152	0.1852
154	0.5294
162	0.3333
165	0.3500
173	0.5526
179	0.1250
183	0.6842
187	0.3718
198	0.7419

Table 4.6: R-Precision obtained for each TREC topic

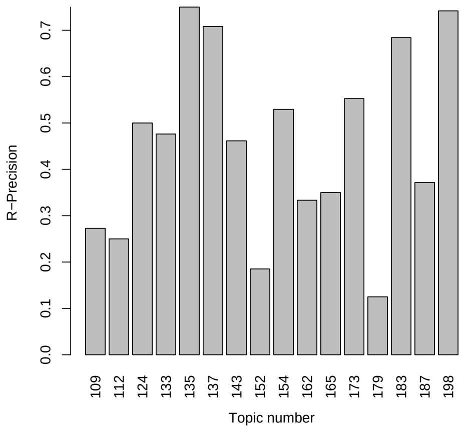


Figure 4.3: R-Precision histogram for the selected TREC topics

We now turn our attention to tuning the size of the *RQ* sets. That is, we attempt to determine the number of top-ranked documents to be included in *RQ* that maximizes the retrieval performance. For this purpose, we use a different performance measure, called F-measure [7], that calculates the harmonic mean of precision and recall. Maximizing the F-measure means finding the best possible combination of precision and recall. We computed the average F-measure for the selected TREC topics with different sizes of *RQ*. As Figure 4.4 shows, the maximum value is 0,4534, reached at the 36th top-ranked document.

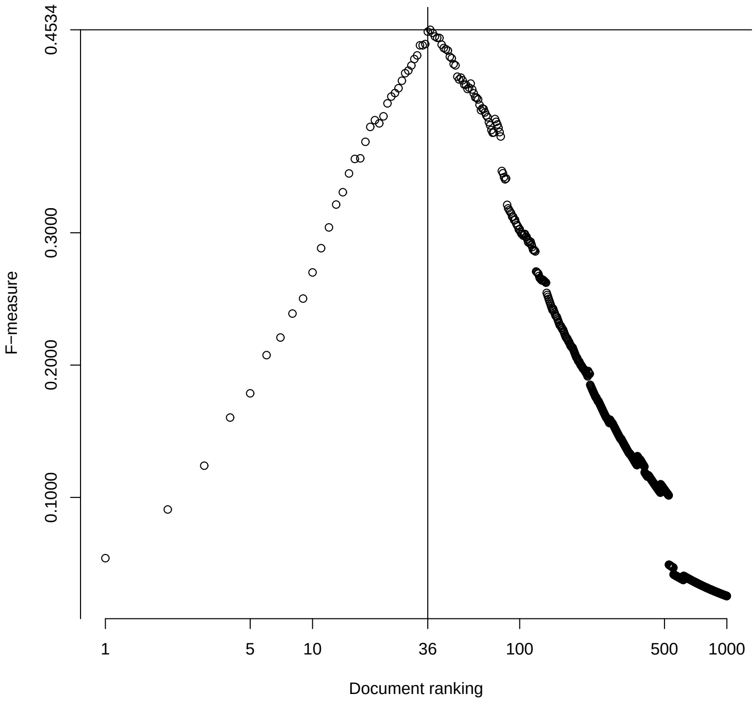


Figure 4.4: Average F-measure for the selected TREC topics with different sizes of *RQ*

Finally, we evaluate the fact retrieval performance of our model. For each topic, we considered the facts described by the 36 top-ranked documents in the corresponding set *RQ*. We grouped the facts by industry, and calculated their relevance to the IR query following the approach discussed in Section 4.2.3. Table 4.8 and Table 4.8 show the industries, along with their relevance, at the top of the ranking for each topic.

Topic 109, expected industry = Software & Computer Services	
Industry	**Relev.**
Software & Computer Services	0.6772
Technology Hardware & Equipment	0.2510
Fixed Line Telecommunications	0.0297
Chemicals	0.0211

Topic 112, expected industry = Biotechnology	
Industry	**Relev.**
Biotechnology	0.7565
Pharmaceuticals	0.0981
Aerospace & Defense	0.0426

Topic 124, expected industry = Health Care Equipment & Services	
Industry	**Relev.**
Biotechnology	0.6496
Health Care Equipment & Services	0.1778
Pharmaceuticals	0.1439
Food & Drug Retailers	0.0249
Technology Hardware & Equipment	0.0038

Topic 133, expected industry = Aerospace & Defense	
Industry	**Relev.**
Aerospace & Defense	0.9793
General Retailers	0.0207

Topic 135, expected industry = Biotechnology	
Industry	**Relev.**
Biotechnology	0.8870
Pharmaceuticals	0.0460
Chemicals	0.0385
Health Care Equipment & Services	0.0213

Topic 137, expected industry = Media	
Industry	**Relev.**
Media	0.6262
Industrial Metals	0.3019
Food Producers	0.0234
General Retailers	0.0151

Topic 143, expected industry = Food Producers	
Industry	**Relev.**
Food Producers	0.9999
Chemicals	3.2e-5

Topic 152, expected industry = Aerospace & Defense	
Industry	**Relev.**
Aerospace & Defense	0.9881
Technology Hardware & Equipment	0.0083
Electronic & Electrical Equipment	0.0016

Table 4.7: Top-ranked industries for the TREC topics 109 to 152

Topic 154, expected industry = Oil & Gas Producers	
Industry	**Relev.**
Oil & Gas Producers	0.6348
Oil Equipment, Services & Distribution	0.3192
Industrial Transportation	0.0460

Topic 162, expected industry = Automobiles & Parts	
Industry	**Relev.**
Aerospace & Defense	0.5426
Automobiles & Parts	0.4562
Oil & Gas Producers	0.0008
Chemicals	0.0002

Topic 165, expected industry = Tobacco	
Industry	**Relev.**
Tobacco	0.6356
Media	0.1473
Airlines	0.1263
Industrial Transportation	0.0877
Aerospace & Defense	0.0029

Topic 173, expected industry = Tobacco	
Industry	**Relev.**
Airlines	0.4585
Tobacco	0.3525
Media	0.0701
Industrial Transportation	0.0431

Topic 179, expected industry = Restaurants & Bars	
Industry	**Relev.**
Restaurants & Bars	0.8930
Beverages	0.0476
Travel & Leisure	0.0229

Topic 183, expected industry = Construction & Materials	
Industry	**Relev.**
Construction & Materials	0.7459
Media	0.0591
Chemicals	0.0579

Topic 187, expected industry = Media	
Industry	**Relev.**
Media	0.9375
Technology Hardware & Equipment	0.0283
General Retailers	0.0225

Topic 198, expected industry = Biotechnology	
Industry	**Relev.**
Biotechnology	0.8858
Chemicals	0.0718
Pharmaceuticals	0.0249

Table 4.8: Top-ranked industries for the TREC topics 154 to 198

The results obtained are satisfactory. For all the topics, even for those where the R-Precision was low (see for example the topics 152 and 197), the expected industry is found at the first (81% of the topics) or the second (19%) position of the ranking.

Furthermore, the relevance value assigned to facts clearly differentiates the industries that are directly related to the topic of analysis from those that are not so relevant. In almost all cases, the relevance value approximately decreases by one order. For example, in the topic number 154, the first (*Oil & Gas Producers*) and second (*Oil Equipment, Services & Distribution*) ranked industries are clearly related to the thematic of the topic (“Oil spills”). The relevance values assigned to these industries (0.6348 and 0.3192, respectively) are significantly greater than the relevance value of the next industry in the ranking (*Industrial Transportation*, 0.0460).

We also found an explanation for some of the topics where the ranking was not completely accurate. The top-raked industry for the topic number 173 is *Airlines*, whereas the expected industry *Tobacco* is found at the second position of the ranking. The reason is that a number of the documents judged to be relevant for this topic report smoking bans on flights. The industry at the top of the ranking for the topic 137 is *Media*, since many media companies also own theme parks (e.g., Time Warner / Warner Bros. Entertainment). In fact, in our *Companies* dimension, the *Media* industry also comprises these recreation and entertainment companies. The second top-ranked industry for this topic is *Industrial Metals*, which still has a relative high relevance value. Although, this industry initially seemed irrelevant for the topic 137, after reading some of the documents retrieved for this topic, we discovered some news relating the vanguardist Japanese company Nippon Steel’s diversification strategy on the amusement-park sector.

4.4 Conclusions

This chapter has proposed a retrieval model to represent and query the documents stored in the XML document warehouse.

The model uses the traditional tree representation of XML documents and maps the elements of the original documents into nodes of the corresponding document trees. Four different types of document nodes are defined, namely: *element*, *attribute*, *textual contents* and *fact collection nodes*. Each *element* node depicts an element of the original document (i.e., the portion of an XML document between a pair of matching tags); the *attribute* nodes represent the elements attributes; the *textual contents* nodes represent the text sections of the documents; and each *fact collection* nodes depicts the facts described within a piece of text.

The queries of the XML documents warehouse comprise two different types of conditions: an IR condition and a path expression. These conditions establish the context of analysis in the contextualized warehouse. The path expression selects a subset of document subtrees from the XML document warehouse. The IR condition is represented as a sequence of query keywords, and language modeling techniques [80] are applied to rank the document subtrees by the probability of finding the query keywords in their text sections. The result is the set RQ (document fragments Relevant to the Query) that comprises the most relevant element node of each selected subtree.

This notion of a set of documents relevant to the query is also used in the probabilistic and relevance modeling-based IR systems to rank the documents according to the probability of sampling their index terms from a document in the set of relevant documents. The work on relevance modeling approachs this probability by the the joint probability of finding the query's keywords and the document index terms together. The retrieval model proposed in this chapter follows a similar approach and adapts the relevance modeling techniques to rank the facts described in the element nodes of RQ, according to their relevance to the query conditions. In this case, the relevance is estimated by the joint probability of sampling the fact and the query keywords. An interesting property is that the sum of the relevance values of all the facts is always equal to one. We propose to measure the quality of the query result by the overall relevance of the element nodes in the set RQ.

In the evaluations of the model we have obtained satisfactory results. However, the structural (XML) part of the query processing strategy remains to be tested. In order to directly analyze the facts found within the textual contents of the documents (without contextualizing a corporate cube), it would be interesting to enrich the retrieval model with some fact fusion operations able to derive initially incomplete facts.

Each element node in the query result set RQ depicts a document fragment that describes the context under analysis. In the rest of the thesis we will use the more general term "document" to mean "element node".

Chapter 5

A Relevance-Extended Multidimensional Model

As earlier discussed in Chapter 3, in a contextualized warehouse, the user specifies the context of analysis by supplying a sequence of keywords. Then, the analysis is performed on an *R-cube*, which is materialized by retrieving the documents and facts related to the selected context. *R-cubes* have two special dimensions: the *relevance* and the *context* dimensions. Thus, each fact in the *R-cube* has a numerical value representing its relevance with respect to the specified context, thereby the name *R-cube* (Relevance cube). Each fact is linked to the set of documents that describe its circumstances by the context dimension.

This chapter extends an existing multidimensional model [62] to represent these two new dimensions, and studies how the traditional OLAP operations are affected by them. Section 5.1 formally defines the *R-cubes* data model. Section 5.2 provides *R-cubes* with a set of unary algebra operations. Finally, Section 5.3 presents some conclusions.

5.1 *R-cubes* Data Model

This section defines a formal data model for the *R-cubes*. It extends an existing multidimensional model [62] with two new special dimensions to represent both the relevance of the facts and their context. For each component of the extended data model, we show its definition and give some examples. The examples follow the scenario presented in Chapter 3. Table 3.1 showed the example *R-cube*.

5.1.1 Dimensions

A *dimension* D is a two-tuple $D = (C_D, \sqsubseteq_D)$, where $C_D = \{C_j\}$ is a set of categories C_j.

Example 5.1. In [62] everything that characterizes a fact is considered to be a dimension, even those attributes modeled as measures in other approaches. Figure 5.1 shows the dimensions for the running example.

Each category $C_j = \{e\}$ is a set of dimension values. $\sqsubseteq_D$ is a partial order on $\cup_j C_j$ (the union of all dimension values in the individual categories). Given two values $e_1, e_2 \in \cup_j C_j$, then $e_1 \sqsubseteq_D e_2$ if e_1 is logically contained in e_2. The intuition is that each category represents the values of a specific granularity level. We will write $e \in D$, meaning that e is a dimension value of D, if $e \in \cup_j C_j$.

There are two special categories present in all dimensions: $\top_D$ and $\bot_D \in C_D$ (the top and bottom categories). The category $\bot_D$ has the values with the finest granularity. All these values do not logically contain other category values and are logically contained by the values of other coarser categories. The category $\top_D = \{\top\}$ represents the coarsest granularity. For all $e \in D, e \sqsubseteq_D \top$.

The partial order $\sqsubseteq_D$ on dimension values is generalized to relate dimension categories as follows: given $C_1, C_2 \in C_D$, then $C_1 \leq_D C_2$ if $\exists e_1 \in C_1, e_2 \in C_2, e_1 \sqsubseteq_D e_2$. We will write $\sqsubseteq$ and $\leq$ instead of $\sqsubseteq_D$ and $\leq_D$ when it is clear that $\sqsubseteq$ and $\leq$ represent the partial order of the dimension D.

Example 5.2. The *Customers* dimension has the categories $\bot_{Customers} = Country \leq Region \leq \top_{Customers}$, with the dimension values $Country = \{Japan, Korea, Cuba, \ldots\}$ and $Region = \{Southeast\ Asia, Central\ America, \ldots\}$. The partial order on category values is: $Japan \sqsubseteq Southeast\ Asia \sqsubseteq \top$, $Korea \sqsubseteq Southeast\ Asia$, $Cuba \sqsubseteq Central\ America \sqsubseteq \top$, etc.

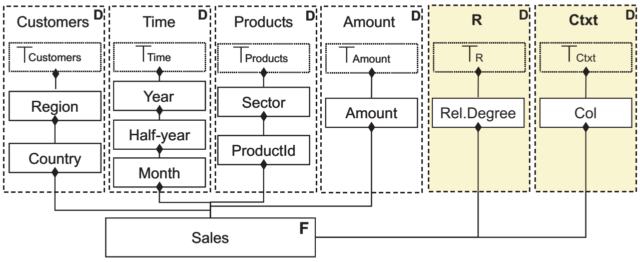


Figure 5.1: Dimensions of the example case of study

The Relevance Dimension

The *relevance* dimension depicts the importance of each fact of the *R-cube* in the selected context (i.e., the IR condition Q). Therefore, it can be used to identify the portions of an *R-cube* that are more interesting for the context of analysis.

Different approaches can be followed to state the *relevance* dimension R. The simplest one is to define it just with the bottom and top categories: $\perp_R = Relevance \leq_R \top_R$. Since we model the relevance as a probability value, the values of the *Relevance* category are real numbers in the interval [0,1]. Like in [45], we propose to introduce an intermediate category to study relevance values from a higher qualitative abstraction level. In this new category, the relevance values will be classified into groups (*Relevance Degrees*) like *irrelevant*, *relevant* or *very relevant*.

As the relevance values are normalized to sum to one, a relevance index of 0.02 may be *irrelevant* if the rest of relevance values are significantly greater, or *relevant* if the maximum value of relevance obtained was, for example, 0.03. Thus, we need to define a dynamic partial order $\sqsubseteq_R^{\gamma}$ to map the values r of the base *Relevance* category to values of the *Relevance Degree* category depending on the value of r/γ. We will use γ as a normalization factor. Note that γ should measure the global relevance of a particular result. Typical measures are $\gamma = MAX(r)$, $\gamma = AVG(r)$ or $\gamma = Quality$.

Definition 5.1. *The relevance dimension is a two-tuple $R = (C_R, \sqsubseteq_R^{\gamma})$ where: $C_R = \{Relevance, Relevance\ Degree, \top_R\}$ is the set of categories; $Relevance = [0,1]$ is the base category $\perp_R$; $Relevance\ Degree \in \wp([a,b])$ is a partitioning of the interval of Real numbers $[a,b]$; and $\sqsubseteq_R^{\gamma}$ is the partial order $r \sqsubseteq_R^{\gamma} rd$, if $r \in Relevance$, $rd \in Relevance\ Degree$ and $r/\gamma \in rd$.*

Example 5.3. Let us consider $\gamma = MAX(r)$ (the maximum value of relevance obtained in the *R-cube*), and five different degrees of relevance, $Relevance\ Degree = \{very\ irrelevant = [0, 0.25),\ irrelevant = [0.25, 0.45),\ neutral = [0.45, 0.55),\ relevant = [0.55, 0.75),\ very\ relevant = [0.75, 1]\}$, which define a partitioning of [0,1]. In the example of Table 3.1 $MAX(r) = 0.4$, then $0.05 \sqsubseteq_R^{0.4}$ *very irrelevant*, $0.1 \sqsubseteq_R^{0.4}$ *irrelevant*, $0.2 \sqsubseteq_R^{0.4}$ *neutral*, $0.4 \sqsubseteq_R^{0.4}$ *very relevant* and $0.25 \sqsubseteq_R^{0.4}$ *relevant*.

The Context Dimension

The context of each fact is detailed by the documents of the warehouse. We represent these documents in the *context* dimension.

Definition 5.2. *The context dimension is a two-tuple $Ctxt = (C_{Ctxt}, \sqsubseteq_{Ctxt})$, where $C_{Ctxt} = \{Col, \top_{Ctxt}\}$ is the set of categories. The category $\perp_{Ctxt} = Col = \{d\}$ is the set of the documents d of the warehouse.*

Example 5.4. In our example, $\{d_{123}^{0.04}, d_{7}^{0.08}, d_{23}^{0.005}, d_{84}^{0.04}, d_{50}^{0.02}, d_{69}^{0.01}, d_{47}^{0.005}\} \subset$ *Ctxt* are the documents of the warehouse which describe the context of the facts presented in the *R-cube*. The superscript denotes the relevance $P(Q \mid d)$ of the document d to the context of analysis (the IR condition Q).

The context dimension as defined in Definition 5.2 is flat, i.e., it has no hierarchies. It would be possible to define a hierarchy for the context dimension by considering the hierarchical structure of the XML documents. Then, we could classify the different documents nodes into the category which represents their element type (i.e., tag name). Such a hierarchy would be useful for browsing the documents of the context dimension at different levels.

5.1.2 Fact-Dimension Relations

The fact-dimension relations link facts with dimension values. Following [62], given a set of facts $F = \{f\}$ and a dimension D, the *fact-dimension relation* between F and D is the set $FD = \{(f, e)\}$, where $f \in F$ and $e \in D$.

A fact f is *characterized* by the dimension value e, written $f \rightsquigarrow_D e$, if $\exists\, e' \in D, (f, e') \in FD \wedge e' \sqsubseteq_D e$. In order to avoid missing values it is required that $\forall f \in F, \exists\, e \in D, (f, e) \in FD$. If the dimension value that characterizes a fact is not known, the pair $(f, \top)$ is added to FD.

Example 5.5. In the example of Table 3.1, we have the facts $F = \{f_1, f_2, f_3, f_4, f_5\}$. $FD_{Customers}$ is the fact-dimension relation that links each fact with its value in the dimension *Customers*. Thus, $FD_{Customers} = \{(f_1,\ Cuba), (f_2,\ Japan), (f_3,\ Korea), (f_4,\ Japan), (f_5,\ Korea)\}$, and for f_3, $f_3 \rightsquigarrow_{Customers}$ *Korea* and $f_3 \rightsquigarrow_{Customers}$ *Southeast Asia*, the fact f_3 depicts trades with Southeast Asian customers.

Example 5.6. The fact-dimension relation FD_{Amount} links each fact with its value in the dimension *Amount*. Then, $FD_{Amount} = \{(f_1, 4,300,000\$), (f_2, 3,200,000\$), (f_3, 900,000\$), (f_4,\ 300,000\$), (f_5, 400,000\$)\}$.

The Relevance Fact-Dimension Relation

The relevance fact-dimension relation links each fact with its relevance value.

Definition 5.3. *The relevance fact-dimension relation is the set* $FR = \{(f, r)\}$ *where* $f \in F$ *is a fact and* $r \in R$ *its relevance. We require each fact to have a unique relevance value,* $\forall f \in F, \exists!\ r \in R, (f, r) \in FR$. *The sum of the relevance values of all the facts in* F *is equal to one,* $\sum_{(f,r)\in FR} r = 1$.

Let $rd \in RelevanceDegree$ *and* γ*, we will write* $f \rightsquigarrow_R^{\gamma} rd$*, meaning that the relevance degree of the fact* f *is* rd *when global relevance measure* γ *is applied, if* $\exists r \in R, (f, r) \in FR$ *and* $r \sqsubseteq_R^{\gamma} rd$.

Example 5.7. For the running example we have $FR = \{(f_1,\, 0.05),\, (f_2, 0.1),\, (f_3,\, 0.2),\, (f_4, 0.4),\, (f_5, 0.25)\}$, and by taking $\gamma = MAX(r) = 0.4$, $f_1 \rightsquigarrow_R^{0.4}$ *very irrelevant*, $f_2 \rightsquigarrow_R^{0.4}$ *irrelevant*, $f_3 \rightsquigarrow_R^{0.4}$ *neutral*, $f_4 \rightsquigarrow_R^{0.4}$ *very relevant* and $f_5 \rightsquigarrow_R^{0.4}$ *relevant*. That is, f_5 is *relevant*, but f_2 may be *irrelevant* for the selected context.

The Context Fact-Dimension Relation

The context fact-dimension relation links each fact with the documents that describe its context.

Definition 5.4. *We define the context fact-dimension relation as the set* $FCtxt = \{(f, d)\}$ *where* $f \in F$ *is a fact described by the document* $d \in Ctxt$*, also written* $f \rightsquigarrow_{Ctxt} d$*. We denote by* RQ *the set of documents relevant for the analysis that describe the facts in* F*,* $RQ = \cup_{(f,d)\in Ctxt}\{d\}$*.*

Example 5.8. In the example, $FCtxt = \{(f_1,\, d_{23}^{0.005}),\, (f_1,\, d_{47}^{0.005}),\, (f_2,\, d_{50}^{0.02}),\, (f_3,\, d_{84}^{0.04}),\, (f_4,\, d_{123}1^{0.04}),\, (f_4,\, d_7^{0.08}),\, (f_5,\, d_7^{0.08}),\, (f_5,\, d_{69}^{0.01})\}$. Thus, the set of documents relevant for the analysis is $RQ = \{d_{123}^{0.04},\, d_7^{0.08},\, d_{23}^{0.005},\, d_{84}^{0.04},\, d_{50}^{0.02},\, d_{69}^{0.01},\, d_{47}^{0.005}\}$. The documents $d_{123}^{0.04}$, $d_7^{0.08}$ depict the context of the fact f_4, then $f_4 \rightsquigarrow_{Ctxt} d_{123}^{0.04}$ and $f_4 \rightsquigarrow_{Ctxt} d_7^{0.08}$.

5.1.3 *R-cubes*: Relevance-Extended Multidimensional Objects

We extend the definition of *multi-dimensional object* [62] to include the relevance and context dimensions discussed before.

Definition 5.5. *A relevance-extended multidimensional object (or R-cube) is a four-tuple* $RM = (F, D, FD, Q)$*, where:* $F = \{f\}$ *is a set of facts;* $D = \{D_i, i = 1, \ldots, n\} \cup \{R, Ctxt\}$ *is a set of dimensions,* $R, Ctxt \in D$ *are the relevance and context dimensions previously defined;* $FD = \{FD_i, i = 1, ..., n\} \cup \{FR, FCtxt\}$ *is a set of fact-dimension relations, one for each dimension* $D_i \in D$*;* FR*,* $FCtxt \in FD$ *are the relevance and context fact-dimension relations defined above; and* Q *is an IR condition. In the model, we represent the relevance of each fact with respect to the context established by the IR condition* Q*.*

We measure the analysis quality of an R-cube for the selected context by $Quality = \sum_{d \in RQ} P(Q \mid d)$*. That is, the overall relevance to the IR condition* Q *of the documents that describe the facts of the R-cube.*

Example 5.9. The sales shown in Table 3.1 constitute the set of facts F of the *R-cube*. The set of dimensions is $D = \{\mathit{Products}, \mathit{Customers}, \mathit{Time}, \mathit{Amount}\} \cup \{R, Ctxt\}$. In the previous examples we have shown the definition of some of these dimensions along with their corresponding fact-dimension relations. The IR condition used for stating the context of analysis was $Q =$“$financial, crisis$”. The quality of the *R-cube* is $Quality = 0.2$.

5.2 *R-cubes* Algebra

In this section we present an algebra for the *R-cubes* by extending the definition of the unary operators presented in [62] to regard the relevance and context of the facts. For each operator, we show its definition, discuss how the relevance and context are updated in the result, and give some examples.

Along the definitions we will assume an *R-cube* $RM = (F, D, FD, Q)$, where $D = \{D_i, i = 1, \ldots, n\} \cup \{R, Ctxt\}$, $FD = \{FD_i, i = 1, \ldots, n\} \cup \{FR, FCtxt\}$ and whose quality is $Quality$. The set of documents relevant for the analysis query Q in the *R-cube* is denoted by RQ.

5.2.1 The Relevance-Extended Selection Operator

The *selection* operator restricts the facts in the cube to the subset of facts that satisfy some given conditions (a predicate). We extend the definition of the selection operator for *R-cubes*, as follows:

Definition 5.6. *Let* $p : D_1 \times \ldots \times D_n \times R \times Ctxt \rightarrow \{true,\ false\}$ *be a predicate on the dimensions in* D*. The relevance-extended selection operator,* σ_R*, is defined as* $\sigma_R[p](RM) = (F', D', FD', Q')$*, where:*

$$\begin{aligned}
F' &= \{f \in F \mid \exists (e_1, \ldots, e_n, r, d) \in D_1 \times \ldots \times D_n \times R \times Ctxt \\
&\qquad \big(p(e_1, \ldots, e_n, r, d) \wedge f \rightsquigarrow_1 e_1 \wedge \ldots \wedge f \rightsquigarrow_n e_n \wedge \\
&\qquad \wedge f \rightsquigarrow_R r \wedge f \rightsquigarrow_{Ctxt} d\big)\}, \\
D' &= D, \\
FD' &= \{FD'_i, i = 1 \ldots n\} \cup \{FR', FCtxt'\}, \\
FD'_i &= \{(f', e) \in FD_i \mid f' \in F'\}, \\
FCtxt' &= \{(f', d) \in FCtxt \mid f' \in F'\}, \quad RQ' = \{d \mid \exists (f', d) \in FCtxt'\}, \\
FR' &= \{(f', r') \mid \exists (f', r) \in FR \wedge f' \in F' \wedge r' = \beta r + \delta(f')\}, \\
\beta &= \frac{Quality}{Quality'} \geq 1,\ Quality' = \sum_{d \in RQ'} P(Q \mid d), \\
\delta(f') &= \sum_{\{d \in RQ' \mid \exists (f,d) \in FCtxt \setminus FCtxt'\}} \frac{P(f' \mid d)' - P(f' \mid d)}{Quality'} P(Q \mid d) \geq 0, \\
P(f' \mid d)' &= \frac{FFreq(f', d)}{\sum_{(f,d) \in FCtxt'} FFreq(f, d)}, \\
P(f' \mid d) &= \frac{FFreq(f', d)}{\sum_{(f,d) \in FCtxt} FFreq(f, d)}, \quad Q' = Q
\end{aligned}$$

The set of facts in the resulting *R-cube* is restricted to those facts characterized by the dimension values where p is true. The fact-dimension relations are restricted accordingly. In particular, the documents that do not describe selected facts are removed from the $FCtxt$ fact-dimension relation and from RQ. In this way, the quality of the *R-cube* will decrease if a relevant document is discarded.

As formally discussed in Theorem 5.1, the relevance values of the facts after the selection are increased by a factor of β. The β factor represents the relative increment of importance of the selected documents when other documents of the warehouse are discarded. In addition, if a fact f' is described in documents which also describe non-selected facts, its relevance is also incremented by $\delta(f')$. This increment represents the increase of importance of the selected fact f' in the documents, when the non-selected facts are no longer taken into account. Thus, it is ensured that the sum of the relevance values of the facts in the resulting *R-cube* remains equal to one.

Theorem 5.1. *Let $RM = (F, D, FD, Q)$ be an R-cube and $RM' = (F', D', FD', Q')$ the R-cube obtained after applying the selection operation $\sigma_R[p]$ over RM, $\sigma_R[p](RM) = RM'$. The relevance of the facts $f' \in F'$ can be calculated as $P(f' \mid RQ') = \beta P(f' \mid RQ) + \delta(f')$, where:*

$$\begin{aligned}
\beta &= \frac{Quality}{Quality'} \geq 1,\\
\delta(f') &= \sum_{\{d \in RQ' \mid \exists (f,d) \in FCtxt \setminus FCtxt'\}} \frac{P(f' \mid d)' - P(f' \mid d)}{Quality'} P(Q \mid d) \geq 0,\\
P(f' \mid d)' &= \frac{FFreq(f', d)}{\sum_{(f,d) \in FCtxt'} FFreq(f, d)},\\
P(f' \mid d) &= \frac{FFreq(f', d)}{\sum_{(f,d) \in FCtxt} FFreq(f, d)}
\end{aligned}$$

Proof. Let $f' \in F'$, as discussed in Section 4.2.3, we estimate its relevance $P(f' \mid RQ')$ by:

$$P(f' \mid RQ') = \frac{\sum_{d \in RQ'} P(f' \mid d)' P(Q \mid d)}{\sum_{d \in RQ'} P(Q \mid d)}$$

Notice that the probability $P(f' \mid d)'$ of observing the fact f' in a document d when considering the restricted set of facts F', is different from the probability $P(f' \mid d)$ of observing the fact f' in d when considering the super-set F.

Since the documents $d \in RQ \setminus RQ'$ do not describe any fact of F', the probability of observing a fact $f' \in F'$ in a document $d \in RQ \setminus RQ'$ is $P(f' \mid d)' = 0$. Thus, we can write:

$$P(f' \mid RQ') = \frac{\sum_{d \in RQ} P(f' \mid d)' P(Q \mid d)}{\sum_{d \in RQ'} P(Q \mid d)}$$

Let RQ_1 be the subset of documents that only describe facts in F', $RQ_1 = \{d \in RQ \mid \nexists (f, d) \in FCtxt \setminus FCtxt'\}$; and RQ_2 the subset of document that at least describe a fact that was in F but not in F', $RQ_2 = \{d \in RQ \mid \exists (f,d) \in FCtxt \setminus FCtxt'\}$. The subsets RQ_1 and RQ_2 as defined above constitute a partitioning of RQ, i.e. $RQ_1 \cap RQ_2 = \emptyset$ and $RQ_1 \cup RQ_2 = RQ$, then:

$$P(f' \mid RQ') = \frac{\sum_{d \in RQ_1} P(f' \mid d)' P(Q \mid d)}{\sum_{d \in RQ'} P(Q \mid d)} + \frac{\sum_{d \in RQ_2} P(f' \mid d)' P(Q \mid d)}{\sum_{d \in RQ'} P(Q \mid d)}$$

Since the documents in RQ_1 only describe facts in F', we have that $\forall d \in RQ_1, P(f' \mid d)' = \frac{FFreq(f',d)}{\sum_{(f,d) \in Ctxt'} FFreq(f,d)} = \frac{FFreq(f',d)}{\sum_{(f,d) \in Ctxt} FFreq(f,d)} = P(f' \mid d)$, and consequently:

$$P(f' \mid RQ') = \frac{\sum_{d \in RQ_1} P(f' \mid d) P(Q \mid d)}{\sum_{d \in RQ'} P(Q \mid d)} + \frac{\sum_{d \in RQ_2} P(f' \mid d)' P(Q \mid d)}{\sum_{d \in RQ'} P(Q \mid d)}$$

The previous formula can be rewritten as follows:

$$\begin{aligned} P(f' \mid RQ') \quad = \quad & \frac{\sum_{d \in RQ} P(Q \mid d)}{\sum_{d \in RQ} P(Q \mid d)} \Big(\frac{\sum_{d \in RQ_1} P(f' \mid d) P(Q \mid d)}{\sum_{d \in RQ'} P(Q \mid d)} \\ & + \frac{\sum_{d \in RQ_2} P(f' \mid d) P(Q \mid d)}{\sum_{d \in RQ'} P(Q \mid d)} + \frac{\sum_{d \in RQ_2} P(f' \mid d)' P(Q \mid d)}{\sum_{d \in RQ'} P(Q \mid d)} \\ & - \frac{\sum_{d \in RQ_2} P(f' \mid d) P(Q \mid d)}{\sum_{d \in RQ'} P(Q \mid d)} \Big) \end{aligned}$$

Since $RQ_1 \cup RQ_2 = RQ$, we have that:

$$\begin{aligned} P(f' \mid RQ') \quad = \quad & \frac{\sum_{d \in RQ} P(f' \mid d) P(Q \mid d)}{\sum_{d \in RQ} P(Q \mid d)} \frac{\sum_{d \in RQ} P(Q \mid d)}{\sum_{d \in RQ'} P(Q \mid d)} + \\ & + \frac{\sum_{d \in RQ_2} (P(f' \mid d)' - P(f' \mid d)) P(Q \mid d)}{\sum_{d \in RQ'} P(Q \mid d)} \end{aligned}$$

The relevance $P(Q \mid d)$ of the documents $d \in RQ \supseteq RQ'$ do not change because the IR condition Q is maintained. In this way, $\beta = \frac{Quality}{Quality'} = \frac{\sum_{d \in RQ} P(Q|d)}{\sum_{d \in RQ'} P(Q|d)} \geq 1$ (notice that $|RQ \mid \geq \mid RQ' \mid$). On the other hand, $P(f' \mid d)' \geq P(f' \mid d)$ because $|\{(f,d) \in FCtxt'\} \mid \leq \mid \{(f,d) \in FCtxt\} \mid$ and $\sum_{(f,d) \in FCtxt'} FFreq(f,d) \leq \sum_{(f,d) \in FCtxt} FFreq(f,d)$. Finally, the previous formula can be expressed as:

$$\begin{aligned} P(f' \mid RQ') \quad &= \quad \beta P(f' \mid RQ) + \delta(f'), \\ \delta(f') \quad &= \quad \sum_{\{d \in RQ' \mid \exists (f,d) \in FCtxt \setminus FCtxt'\}} \frac{P(f' \mid d)' - P(f' \mid d)}{Quality'} P(Q \mid d) \geq 0 \end{aligned}$$

□

Example 5.10. We can apply the relevance-extended selection operator to dice the *R-cube* to study the sales made to *Southeast Asian* customers. Since conditions on the relevance dimension are supported, we could also restrict the analysis to those facts considered as *relevant* or *very relevant*. Thus, $p =$ ($Customers.Region$ $=$ $Southeast\ Asia$, $R.Relevance\ Degree = very\ relevant$ $or\ relevant$). Table 5.1 shows the resulting *R-cube*. The set of facts is restricted to $F' = \{f_4, f_5\}$. The resulting fact-dimension relations are: $FProducts' = \{(f_4, fo1), (f_5, fo2)\}$, $FCustomers' = \{(f_4, Japan), (f_5,\ Korea)\}$, $FTime' = \{(f_4,\ 1998/10), (f_5,\ 1998/11)\}$, $FAmount' = \{(f_4, 300,000\$), (f_5, 400,000\$)\}$. The stated restriction also affects the set of documents that describe the facts of the *R-cube*, $FCtxt' =\ \{(f_4,\ d_{123}^{0.04}),\ (f_4, d_7^{0.08}),\ (f_5, d_7^{0.08}),\ (f_5, d_{69}^{0.01})\}$, and then $RQ' = \{d_{123}^{0.04},\ d_7^{0.08},\ d_{69}^{0.01}\} \subset RQ$. Since some documents relevant for the analysis context are discarded, the quality of the resulting *R-cube* decreases to $Quality' = 0.13$ ($< Quality = 0.2$). Notice that all the facts related to d_{123}, d_7 and d_{69} in $FCtxt$ were selected by the operation. That is, in this case, all the documents in the resulting *R-cube* only describe selected facts. Consequently, the relevance of the facts is increased by a factor of β, $\beta = Quality/Quality' = 1.54$. The resulting relevance fact-dimension relation is $FR' = \{(f_4, 0.615), (f_5, 0.385)\}$.

F’	ProductId	Country	Month	Amount	R	Ctxt
f_4	$fo1$	$Japan$	1998/10	$300,000\$$	0.615	$d_{123}^{0.04}, d_7^{0.08}$
f_5	$fo2$	$Korea$	1998/11	$400,000\$$	0.385	$d_7^{0.08}, d_{69}^{0.01}$

Table 5.1: Result of applying σ_R on the example *R-cube* of Table 3.1, p = (*Customers.Region* = *Southeast Asia, R.Relevance Degree* = *very relevant or relevant*). *Quality’* = 1.54.

Example 5.11. Let us now consider the result of the previous example. If we select the sales made during the month of November 1998 (i.e., let $p =$ $(Time.Month = 1998/11)$) from the *R-cube* shown in Table 5.1, the new set of facts is $F' = \{f_5\}$, and the context fact-dimension relation becomes $FCtxt' = \{(f_5, d_7^{0.08}),\ (f_5, d_{69}^{0.01})\}$, resulting $RQ' = \{d_7^{0.08}, d_{69}^{0.01}\}$ and $Quality' = 0.09$, then $\beta = 0.13/0.09 = 1.444$. Document d_{69} only describes f_5, the selected fact. However, document d_7 describes both the selected fact, f_5, and the discarded one, f_4, since in the input *R-cube* we had that $\{(f_4, d_7^{0.08}), (f_5, d_7^{0.08})\} \subset FCtxt$. The relevance of f_5 in the input *R-cube* was 0.385, $(f_5, 0.385) \in FR$. Then, in the resulting *R-cube*, the relevance of f_5 will be recalculated as $\beta 0.385 + \delta(f_5)$. Let $FFreq(f_4, d_7) = FFreq(f_5, d_7) = 3$, the dimension values of the fact f_4 appear three times in document d_7, likewise, the frequency of the dimensions values of the fact f_5 in d_7 is three. Thus, we have that $P(f_5 \mid d_7) = 3/(3+3) = 0.5$ and $P(f_5 \mid d_7)' = 3/3 = 1$. The relevance of document d_7 to the IR condition is $P(Q \mid d_7) = 0.08$. Then, $\delta(f_5) = (P(f_5 \mid d_7)' - P(f_5 \mid d_7))P(Q \mid$

$d_7)/Quality' = 0.444$. In this way, we finally have that $\beta 0.385 + \delta(f_5) = 1$, and the resulting relevance fact-dimension relation is $FR' = \{(f_5, 1)\}$.

The β factor measures the quality lost in the resulting *R-cube*. Good restrictions will result in low β values, since they preserve the relevant facts of the *R-cube* and discard the non-relevant ones. However, sometimes, we may be interested in a particular region of the cube. A high β value (a low $Quality'$) will warn the user of a meaningless result.

Example 5.12. When the selection operator is applied to the example *R-cube* of Table 3.1, with the predicate $p = (Customers.Region = Central\ America)$, the set of facts in the resulting *R-cube* is restricted to $F' = \{f_1\}$, the context fact-dimension relation becomes $FCtxt' = \{(f_1, d_{23}^{0.005}),\ (f_1, d_{47}^{0.005})\}$ and $RQ' = \{d_{23}^{0.005}, d_{47}^{0.005}\}$. Consequently, the quality is reduced to $Quality' = 0.01$, resulting $\beta = 0.2/0.01 = 20$. The high β value points to a considerable lost quality, meaning that the analysis result is not significant in the selected context (as the financial crisis mainly affected the Southeast Asian countries).

5.2.2 The Relevance-Extended Aggregate Formation Operator

The *aggregate formation* operator evaluates an aggregation function on the *R-cube*. Following [62], we assume the existence of a family of functions $g : 2^F \rightarrow D_{n+1}$ that receive a set of facts and compute an aggregation by taking the data from the requested fact-dimension relation (e.g. SUM_i takes the data from FD_i, and performs the sum).

The *Group* operator defined in [62] groups the facts characterized by the same dimension values. Given the dimension values $(e_1, \ldots, e_n) \in D_1 \times \ldots \times D_n$, $Group(e_1, \ldots, e_n) = \{f \in F \mid f \rightsquigarrow_1\ e_1 \wedge \ldots \wedge f \rightsquigarrow_n\ e_n\}$.

Example 5.13. In the example *R-cube* of Table 3.1, we can group those sales made to Southeast Asian customers during the second half of 1998 as follows: given the dimension values $(\top, Southeast\ Asia,\ 1998/2nd\ half, \top) \in \top_{Products} \times Region \times Half\_year \times \top_{Amount}$, we have that $Group(\top,\ Southeast\ Asia,\ 2nd\ half\ 1998, \top) = \{f_4, f_5\}$, since $f_4 \rightsquigarrow_{Products} \top$, $f_4 \rightsquigarrow_{Customers} Southeast\ Asia$, $f_4 \rightsquigarrow_{Time} 2nd\ half$ 1998 and $f_4 \rightsquigarrow_{Amount} \top$; $f_5 \rightsquigarrow_{Products} \top$, $f_5 \rightsquigarrow_{Customers} Southeast\ Asia$, $f_5 \rightsquigarrow_{Time} 2nd\ half$ 1998, and $f_5 \rightsquigarrow_{Amount} \top$.

Definition 5.7. *Given a new dimension D_{n+1}, an aggregation function $g : 2^F \rightarrow D_{n+1}$, and a set of grouping categories $\{C_i \in C_{D_i}, i = 1 \ldots n, C_{D_i} \neq C_R, C_{Ctxt}\}$, the relevance-extended aggregate formation operator, α_R, is defined* *as* $\alpha_R[D_{n+1}, g, C_1, \ldots, C_n](RM) = (F', D', FD', Q')$*, where:*

$$
\begin{array}{rcl}
F' & = & \{Group(e_1, \ldots, e_n) \mid (e_1, \ldots, e_n) \in C_1 \times \ldots \times C_n \wedge \\
 & & \qquad\qquad\qquad\qquad \wedge\ Group(e_1, \ldots, e_n) \neq \emptyset\}, \\
D' & = & \{D'_i, i = 1 \ldots n\} \cup \{D_{n+1}\} \cup \{R, Ctxt\}, \\
D'_i & = & (C'_{D_i}, \sqsubseteq'_{D_i}), C'_{D_i} = \{C_{ij} \in C_{D_i} \mid C_i \leq_{D_i} C_{ij}\}, \sqsubseteq'_{D_i} = \sqsubseteq_{D_i}|_{C'_{D_i}}, \\
FD' & = & \{FD'_i, i = 1 \ldots n\} \cup \{FD_{n+1}\} \cup \{FR', FCtxt'\}, \\
FD'_i & = & \{(f', e'_i) \mid \exists (e_1, \ldots, e_n) \in C_1 \times \ldots \times C_n, \\
 & & \qquad f' = Group(e_1, \ldots, e_n) \in F' \wedge e_i = e'_i\}, \\
FD_{n+1} & = & \bigcup\limits^{(e_1, \ldots, e_n) \in C_1 \times \ldots \times C_n} \{(Group(e_1, \ldots, e_n), g(Group(e_1, \ldots, e_n))) \mid \\
 & & \qquad\qquad\qquad Group(e_1, \ldots, e_n) \neq \emptyset\}, \\
FR' & = & \{(f', r') \mid \exists (e_1, \ldots, e_n) \in C_1 \times \ldots \times C_n \wedge \\
 & & \qquad \wedge\ f' = Group(e_1, \ldots, e_n) \in F' \wedge \\
 & & \qquad \wedge\ r' = \sum\limits_{(f,r) \in FR, f \in Group(e_1, \ldots, e_n)} r\}, \\
FCtxt' & = & \{(f', d') \mid \exists (e_1, \ldots, e_n) \in C_1 \times \ldots \times C_n \wedge \\
 & & \qquad \wedge\ f' = Group(e_1, \ldots, e_n) \in F' \wedge \\
 & & \qquad \wedge\ d' \in \bigcup\limits_{(f,d) \in FCtxt, f \in Group(e_1, \ldots, e_n)} \{d\}\}, \\
RQ' & = & RQ, \quad Quality' = Quality, \\
Q' & = & Q
\end{array}
$$

Each fact in the resulting *R-cube* represents a group of facts of the original *R-cube* (those characterized by the same values in the grouping category). The aggregation function is evaluated over each group of facts and the result is stored in the new dimension D_{n+1}. The dimensions $D_i, \ldots D_n$ are restricted to the ancestor categories of the corresponding grouping category. The $FCtxt$ fact-dimension relation now relates each new fact with the documents that were associated with any of the original facts of the corresponding group. Notice that the set of documents relevant to the analysis query RQ does not change. Likewise, the quality of the *R-cube* is not modified. As discussed in Section 4.2.3, we estimate the relevance of the facts by the frequency of their dimension values in the relevant documents. Consequently, the relevance of each group is the sum of the relevance values of the original facts in the group (see Theorem 5.2). We update the FR fact-dimension relation accordingly. Thus, the sum of the relevance values of the facts in the resulting *R-cube* remains equal to one.

Theorem 5.2. *Let* $\{C_i \in C_{D_i}, i = 1 \ldots n\}$ *be a set of grouping categories, and let* $Group(e_1, \ldots, e_n)$ *be the group of facts of the cube characterized by the* *category values* $(e_1, \ldots, e_n) \in C_1 \times \ldots \times C_n$*. The relevance value of the group* $P(Group(e_1, \ldots, e_n) \mid RQ)$ *is determined by the following formula:*

$$P(Group(e_1, \ldots, e_n) \mid RQ) = \sum_{f_i \in Group(e_1,\ldots,e_n)} P(f_i \mid RQ)$$

Proof. Consider the fact f characterized by the dimension values $(e_1, \ldots, e_2)$. By applying the formula (4.27), the probability $P(f \mid d)$ of finding the fact f in the document d can be estimated as follows:

$$\begin{aligned} P(f \mid d) &= \frac{FFreq(f,d)}{\mid d \mid_f} = \sum_{f_i \in Group(e_1,\ldots,e_n)} \frac{FFreq(f_i,d)}{\mid d \mid_f} \\ &= \sum_{f_i \in Group(e_1,\ldots,e_n)} P(f_i \mid d) \end{aligned}$$

That is, $P(f \mid d)$ can be calculated by adding the dimension values frequency of each fact of $Group(e_1, \ldots, e_n)$ in the document d. Notice that $\forall f_i \in Group(e_1, \ldots, e_n), f_i \rightsquigarrow_1 e_1 \wedge \ldots \wedge f_i \rightsquigarrow_n e_n$.

Thus, with the previous result, the fact relevance calculus formula (4.26) can be expressed as:

$$\begin{aligned} P(f \mid RQ) &= \frac{\sum_{d \in RQ} P(f \mid d) P(Q \mid d)}{\sum_{d \in RQ} P(Q \mid d)} \\ &= \sum_{d \in RQ} \frac{\left(\sum_{f_i \in Group(e_1,\ldots,e_n)} P(f_i \mid d)\right) P(Q \mid d)}{\sum_{d \in RQ} P(Q \mid d)} \\ &= \sum_{f_i \in Group(e_1,\ldots,e_n)} \left(\frac{\sum_{d \in RQ} P(f_i \mid d) P(Q \mid d)}{\sum_{d \in RQ} P(Q \mid d)} \right) \\ &= \sum_{f_i \in Group(e_1,\ldots,e_n)} P(f_i \mid RQ) \end{aligned}$$

□

Example 5.14. In the example *R-cube* of Table 3.1, we can compute the total amount of sales per *Region* and *Half_year* by applying the aggregate formation operator as follows:

Let $Total = (C_{Total}, \sqsubseteq_{Total})$ be a new dimension to store the result of the sum, with the categories $C_{Total} = \{Total\ Amount, \top_{Total}\}$, $\perp_{Total} = Total\ Amount \leq \top_{Total}$. Let SUM_{Amount} be the aggregation function that performs the sum of the values of the *Amount* dimension. Since we want to evaluate the sum per *Region* and $Half\_year$, the grouping categories are $\{\top_{Products}$, *Region*, $Half\_year$, $\top_{Amount}\}$. Table 5.2 shows the result of applying the aggregate formation operator $\alpha_R[Total,\ SUM_{Amount},\ \top_{Products},$ *Region*, $Half\_year,\ \top_{Amount}]$ to the *R-cube* of Table 3.1.

In the resulting *R-cube*, there is a new fact for each combination $(e_1, \ldots, e_2)$ of dimension values in the given grouping categories, $(e_1, \ldots, e_2) \in \top_{Products} \times Region \times Half\_year \times \top_{Amount}$. In the example, the possible combinations are ($\top$, *Central America*, 1998/1*st half*, $\top$), ($\top$, *Southeast Asia*, 1998/1*st half*, $\top$) and ($\top$, *Southeast Asia*, 1998/2*nd half*, $\top$). Each new fact represents the group of original facts characterized by the corresponding combination of grouping categories values. Thus, in the resulting *R-cube*, we have the facts $\{f_1\} = Group(\top, Central\ America, 1998/1st\ half, \top)$, $\{f_2, f_3\} = Group(\top,$ *Southeast Asia*, 1998/1*st half*, $\top)$ and $\{f_4, f_5\} = Group(\top, Southeast\ Asia,$ 1998/2*nd half*, $\top)$, obtaining $F' = \{\{f_1\}, \{f_2, f_3\}, \{f_4, f_5\}\}$.

The resulting *R-cube* has seven dimensions. The *Ctxt* and R dimensions are not modified. The dimension $Products'$ and $Amount'$ have been restricted to their top categories, $\top_{Products}$ and $\top_{Amount}$, respectively. The dimension $Customers'$ is reduced, so that only the categories $Region \leq \top_{Customers}$ are kept. The $Time'$ dimension is also reduced to the categories $Half\_year \leq Year \leq \top_{Time}$. The new dimension $Total$ stores the result of the aggregation.

In the result, $FProducts'$, $FCustomers'$, $FTime'$ and *FAmount'*are the fact-dimension relations that now link each new fact with the dimension values that characterize the corresponding group of original facts. For example, for the new fact $\{f_4, f_5\}$, we have that $(\{f_4, f_5\}, \top) \in FProducts'$, $(\{f_4, f_5\}, Southeast\ Asia) \in FRegion'$, $(\{f_4, f_5\}, 1998/2nd\ half) \in FTime'$ and $(\{f_4, f_5\}, \top) \in Amount'$. The $FCtxt'$ fact-dimension relation links each new fact with the documents that were related to the original facts of the corresponding group. For example, in the original *R-cube* we had $\{(f_4, d_{123}^{0.04}), (f_4,\ d_7^{0.08}), (f_5, d_7^{0.08}), (f_5, d_{69}^{0.01})\} \subset FCtxt$, then, in the resulting *R-cube* we have $\{(\{f_4, f_5\}, d_{123}^{0.04}), (\{f_4, f_5\}, d_7^{0.08}), (\{f_4, f_5\},\ d_{69}^{0.01})\} \subset FCtxt'$. Thus, the aggregate formation operation never modifies the set of documents relevant for the analysis, i.e., $RQ' = RQ$. Then, the quality of the *R-cube* remains, i.e., $Quality' = Quality = 0.2$. The relevance of the new facts is the sum of the relevance values of the original facts in the corresponding group. In the example, we have that $(\{f_4, f_5\}, 0.65) \in FR'$, since $\{(f_4, 0.4), (f_5, 0.25)\} \subset FR$. Finally, the new $FTotal$ fact-dimension relation links each new fact with the result of applying the aggregation function SUM_{Amount} to the corresponding group of facts. Since $\{(f_4, 300,000\$), (f_5, 400,000\$)\} \subset FAmount$, then $(\{f_4, f_5\}, 700,000\$) \in FTotal$.

The resulting *R-cube* clearly shows that the most relevant fact is $\{f_4, f_5\}$. That is, the financial crisis had the strongest impact in the Southeast Asian region during the second half of the year, which would explain the corresponding sales fall. We could gain insight into the context of this fact by performing a *drill-through* operation [81], thus retrieving the textual contents of the documents that explain the details of the crisis.

F'	$\top_{Products}$	$Customers'.Region$	$Time'.Half\_year$	$\top_{Amount}$
$\{f_1\}$	$\top$	$Central\ America$	$1998/1st\ half$	$\top$
$\{f_2, f_3\}$	$\top$	$Southeast\ Asia$	$1998/1st\ half$	$\top$
$\{f_4, f_5\}$	$\top$	$Southeast\ Asia$	$1998/2nd\ half$	$\top$

F'	$Total$	R	$Ctxt$
$\{f_1\}$	$4,300,000\$$	0.05	$d_{23}^{0.005}, d_{47}^{0.005}$
$\{f_2, f_3\}$	$4,100,000\$$	0.3	$d_{50}^{0.02}, d_{84}^{0.04}$
$\{f_4, f_5\}$	$700,000\$$	0.65	$d_{123}^{0.04}, d_7^{0.08}, d_{69}^{0.01}$

Table 5.2: Result of applying $\alpha_R[$*Total, SUM*$_{Amount}$, $\top_{Products}$, *Region, Half_year,* $\top_{Amount}]$ on the example *R-cube* of Table 3.1.

5.2.3 The Relevance-Extended Projection Operator

The *projection operator* removes some of the cube dimensions. Next, we give the formal definition of the *relevance-extended projection operator*. It is basically the projection operator defined in [62], but restricted to avoid the removal of the *relevance* and the *context* dimensions. In this way, we can conclude that since the result of the three operations over *R-cubes* is always an *R-cube*, the *R-cubes* algebra presented here is closed.

Definition 5.8. *Given the dimensions* $D_1, \ldots, D_k \in D \setminus \{R, Ctxt\}$*, we define the relevance-extended projection operator as* $\pi_R[D_1, \ldots, D_k](RM) = (F', D', FD', Q')$*:* $F = F'$*,* $D' = \{D_1, \ldots, D_k\} \cup \{R, Ctxt\}$*,* $FD' = \{FD_1, \ldots, FD_k\} \cup \{FR, FCtxt\}$*, and* $Q' = Q$*.* $RQ' = RQ$ *and* $Quality' = Quality$*.*

Example 5.15. By following with the Example 5.14, we can apply the relevance-extended projection operator to remove the $Products$ and $Amount$ dimensions. The result is equivalent to the one that would be obtained with the traditional *roll-up* operation.

Thus, by applying $\pi_R[Customers, Time, Total\ Amount]$ on the *R-cube* of Table 5.2, we obtain a new *R-cube* with the same set of facts, the dimensions $Customers$, $Time$, $Total$, R and $Ctxt$ as returned by the aggregation operator, along with their corresponding fact-dimension relations. Since the $FCtxt$ fact-dimension relation is not modified, $RQ' = RQ$ and the quality of the resulting *R-cube* remains, i.e., $Quality' = Quality = 0.2$.

The *drill-down* operation is equivalent to evaluating an *aggregate formation* on lower categories [62]. Since more detailed data is required, a reference to the original *R-cube* is needed.

5.3 Conclusions

This chapter has presented a multidimensional data model and algebra for the *R-cubes*. The model extends that proposed in [62] with the relevance and the context dimensions. The relevance dimension can be used to explore the most relevant portions of an *R-cube*. The user will be able to gain insight into the circumstances of a fact by retrieving its related documents represented in the context dimension.

An interesting property of the *R-cubes* is that the sum of the relevance values of all the facts in the cube is equal to one. However, as discussed in Section 4.2.4, although all document collections are not suitable for every analysis tasks, the sum of the facts relevance values will be kept equal to one. We propose to measure the quality of an *R-cube* by the overall relevance of the documents that describe the facts in the cube, i.e., the sum of the relevance values of all the documents in the context dimension.

The chapter redefines the unary algebra operators of [62] to regard the relevance and context of the facts:

- The *relevance-extended selection* operator restricts the facts in the *R-cube* to the subset of facts that satisfy some given conditions. Thus, the selection operator can be applied to dice and only study a region of the *R-cube*. We note that the quality of the *R-cube* will decrease if a relevant document is discarded. The relevance values of the facts after the selection are increased by a factor of β which measures the quality lost in the resulting *R-cube*. Good restrictions will result in low β values, since they preserve the relevant facts of the *R-cube* and discard the non-relevant ones. However, sometimes, we may be interested in a particular region of the cube. A high β value will warn the user of a meaningless result. In addition, if a fact f' is described in documents which also describe non-selected facts, its relevance is also incremented by $\delta(f')$. This increment represents the increase of importance of the selected fact f' in the documents, when the non-selected facts are no longer taken into account.

- The *relevance-extended aggregate formation* operator evaluates an aggregation function on the *R-cube*. This operator receives a set of dimension categories (grouping categories). In the resulting *R-cube*, each fact represents a group of facts of the original R-cube (those characterized by the same values in the grouping categories). The aggregation function is evaluated over each group of facts and the result is stored in a new dimension. Each new fact is related to the documents that were associated with any of the original facts of the corresponding group. Thus, the quality of the *R-cube* is not modified, since no documents are discarded. The relevance of each group is calculated by adding up the relevance values of the original facts in the group.

- The *relevance-extended projection* operator removes some of the cube dimensions. The traditional roll-up operation is equivalent to the aggregate formation operator, followed by a projection operation, to remove the non-grouping categories from the cube.

To our best knowledge, the unique work directly related to this chapter is [70]. This paper proposes to annotate external documents by means of an ontology in RDFs format that comprises all the dimension values defined in the DW. In this way, the results of OLAP queries are presented next to the documents annotated with the same dimension values. However, unlike the work proposed in this thesis it does not provide a formal framework for calculating fact relevance with respect to user queries. The analysis capabilities of the contextualized warehouse outperform those of the system proposed in [70]. The *R-cubes* allow users to analyze the corporate facts under different contexts, whereas this option is not possible in [70]. That is, in a contextualized warehouse the analysis process starts by selecting the set of documents that constitute the analysis context, then the facts in the *R-cube* are ranked by their importance to this context. In [70] the user directly interact with the OLAP cube and the system shows the documents related to the selected facts. In an *R-cube* it is also possible to perform a similar analysis by considering the entire document collection as the context of analysis (i.e., $Q = NULL$ and $RQ = Col$), and by assuming that all the documents are equally relevant for the IR condition (e.g., $P(Q \mid d) = 1$). Notice that when the user selects a fact in the *R-cube*, we still can rank the documents related to this fact by the probability $P(f \mid d)$ of finding the fact in each document.

The *R-cube's* algebra remains to be completed with binary operators. For this purpose, data fusion mechanisms [17] can be applied to combine the relevance of the involved facts.

Finally, note that an *R-cube* is a special *multi-dimensional object* [62]. Thus, an *R-cube* can also be queried by using the algebra proposed in the base model. In this case, the result may no longer be an *R-cube*, as the relevance or the context dimension may be projected away, or the facts relevance may not be updated. However, these operators could be applied to perform interesting analysis. For example, the context dimension may be used as a grouping category to calculate aggregations over the facts described in each document.

Chapter 6
The Prototype

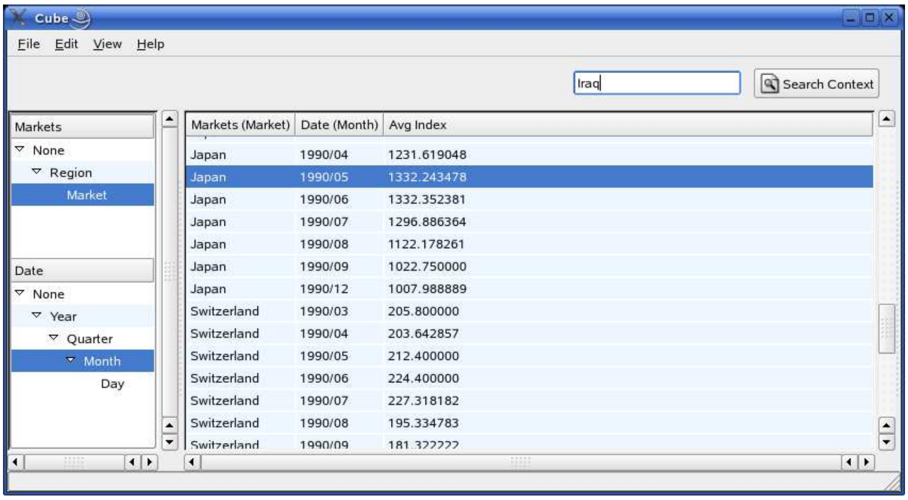
Given the novelty of the *R-cubes*, it is difficult to find a proper testbed to prove its utility. In this chapter a prototypical contextualized warehouse is presented. The new analysis capabilities introduced in the *R-cubes* are tested in a real-world analysis scenario. In the prototype a corporate warehouse with data of the World major stock indices is contextualized with a collection of digital business newspapers. In this scenario *R-cubes* are used for analyzing how the Middle East conflict of 1990 influenced the different market indices. The chapter gives an overview of the main aspects involved in the design of the system.

Section 6.1 presents the document and corporate warehouses of the prototype. Section 6.2 shows the usefulness of the prototype by means of an example usage case, and describes the analysis process by means of a sequence of screen-shots. Section 6.3 summarises some implementation issues.

6.1 The Document and the Corporate Warehouse

The document warehouse consists of a digital collection of some well-known international business newspapers. We inserted in the prototype a total of 132 articles from the issues published during 1990. Among other things, these articles report the trends of markets during that period. It is usual to find news explaining how stock markets are affected by some financial circumstances, e.g.: *"The reaction of German market to the rise of interest rates is expected to be . . . "*.

The corporate warehouse keeps a historical record of market indices as measured by Morgan Stanley Capital International Perspective [2]. As Figure 6.1 shows, the corporate cube has two dimensions. The $Markets$ dimension is organized into two categories: $Market$ ($U.S.$, $Japan$, etc.) and $Region$ ($North\ America$, $Asia$, etc.). The $Date$ dimension is organized into Day, $Month$, $Quarter$ and $Year$. The $Avg\ Index$ dimension measures the average price index per market and date. In order to make the price indices directly comparable, each index is based on the close of 1969 equaling 100. The facts $(Japan, 1990/05, 1332.24)$ and $(Germany, 1990/05, 297.92)$ depict that the average index of the Japanese and German markets during the month of May 1990 was of 1332.24 and 297.92, respectively. In the experiments we have only considered the index trends of the year 1990, resulting, at the lowest dimension categories, in 1396 facts. It is worth to mention than this small data set was selected, since the objective of this experiment is to illustrate the usefulness of a contextualized warehouse. Testing the performance of the system with larger data sets and studying query evaluation optimization techniques is future work.

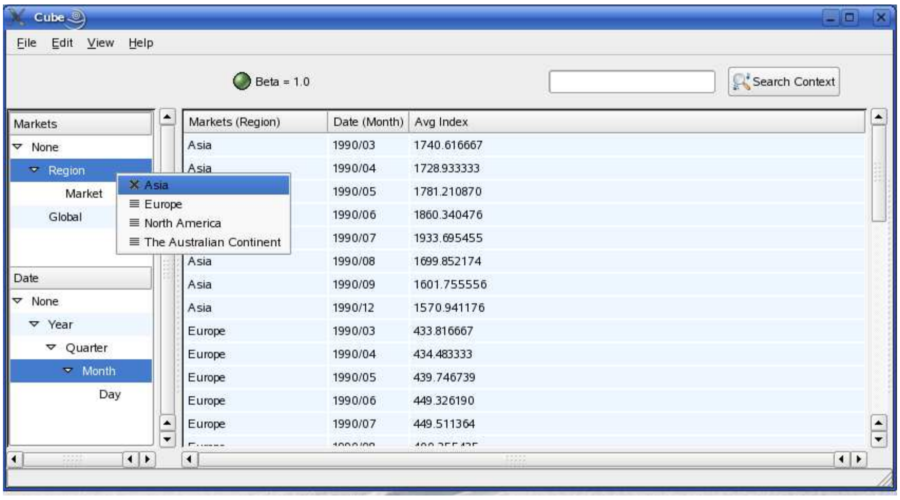


Iraq

Markets (Market)	Date (Month)	Avg Index
Japan	1990/04	1231.619048
Japan	1990/05	1332.243478
Japan	1990/06	1332.352381
Japan	1990/07	1296.886364
Japan	1990/08	1122.178261
Japan	1990/09	1022.750000
Japan	1990/12	1007.988889
Switzerland	1990/03	205.800000
Switzerland	1990/04	203.642857
Switzerland	1990/05	212.400000
Switzerland	1990/06	224.400000
Switzerland	1990/07	227.318182
Switzerland	1990/08	195.334783
Switzerland	1990/09	181.322222

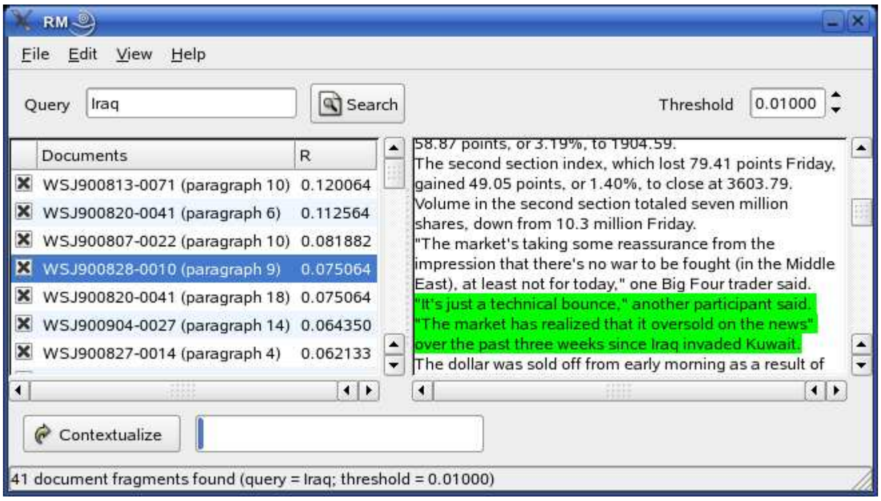
Figure 6.1: OLAP window

Like in traditional data warehouses, an OLAP interface allows analysts to query the corporate warehouse. Among other things, it is possible to study the average index of the different markets, pivot to order by date, roll-up to calculate the average per region, or dice the cube to select the index values of the second quarter of 1990 in Germany (see Figure 6.1 and Figure 6.2).



Markets (Region)	Date (Month)	Avg Index
Asia	1990/03	1740.616667
Asia	1990/04	1728.933333
	1990/05	1781.210870
	1990/06	1860.340476
	1990/07	1933.695455
Asia	1990/08	1699.852174
Asia	1990/09	1601.755556
Asia	1990/12	1570.941176
Europe	1990/03	433.816667
Europe	1990/04	434.483333
Europe	1990/05	439.746739
Europe	1990/06	449.326190
Europe	1990/07	449.511364

Figure 6.2: Dicing in the OLAP window



Query Iraq

Threshold 0.01000

	Documents	R
☒	WSJ900813-0071 (paragraph 10)	0.120064
☒	WSJ900820-0041 (paragraph 6)	0.112564
☒	WSJ900807-0022 (paragraph 10)	0.081882
☒	WSJ900828-0010 (paragraph 9)	0.075064
☒	WSJ900820-0041 (paragraph 18)	0.075064
☒	WSJ900904-0027 (paragraph 14)	0.064350
☒	WSJ900827-0014 (paragraph 4)	0.062133

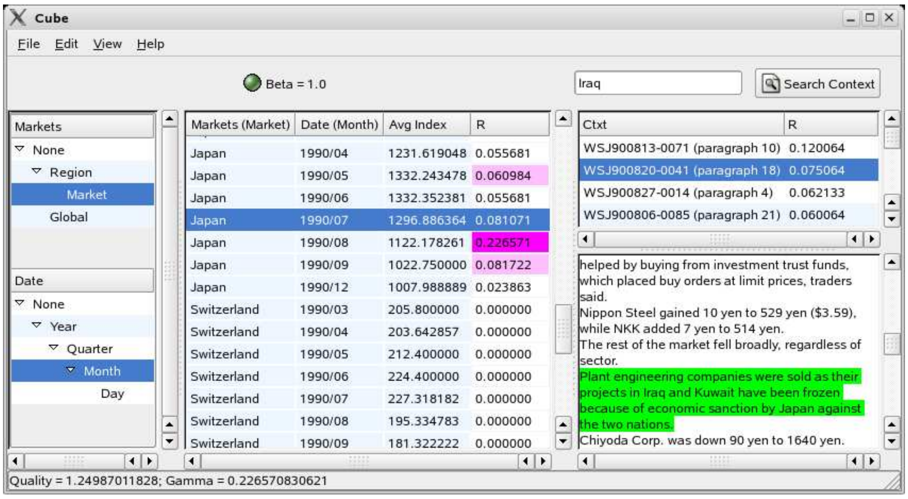
58.87 points, or 3.19%, to 1904.59.
The second section index, which lost 79.41 points Friday, gained 49.05 points, or 1.40%, to close at 3603.79.
Volume in the second section totaled seven million shares, down from 10.3 million Friday.
"The market's taking some reassurance from the impression that there's no war to be fought (in the Middle East), at least not for today," one Big Four trader said.
"It's just a technical bounce," another participant said.
"The market has realized that it oversold on the news" over the past three weeks since Iraq invaded Kuwait.
The dollar was sold off from early morning as a result of

41 document fragments found (query = Iraq; threshold = 0.01000)

Figure 6.3: IR window

6.2 An Analysis Example

Let us suppose that there are recent news about a conflict happening in the Middle East. During the last decades conflicts have been frequent in this area, so the analyst decides to use the prototype to study the reaction of the stock markets to the Iraq war of 1990. After entering the keyword “Iraq” in the toolbar and clicking on “Search Context”, the system presents to the user a list of documents about Iraq ranked by relevance (see the left side of Figure 6.3). By selecting a document, its contents appear in the right part of the window, and the paragraph that contains the keyword is highlighted. Then, the analyst can refine the query by adding or removing keywords, and by specifying a minimum relevance threshold. It is also possible to provide some user feedback by clicking on the check boxes associated to the documents that better describe the Iraq conflict of 1990. Once the set of documents that describe the context under analysis has been obtained, the “Contextualize” button is used to continue the analysis in the OLAP window shown in Figure 6.4.



Markets (Market)	Date (Month)	Avg Index	R
Japan	1990/04	1231.619048	0.055681
Japan	1990/05	1332.243478	0.060984
Japan	1990/06	1332.352381	0.055681
Japan	1990/07	1296.886364	0.081071
Japan	1990/08	1122.178261	0.226571
Japan	1990/09	1022.750000	0.081722
Japan	1990/12	1007.988889	0.023863
Switzerland	1990/03	205.800000	0.000000
Switzerland	1990/04	203.642857	0.000000
Switzerland	1990/05	212.400000	0.000000
Switzerland	1990/06	224.400000	0.000000
Switzerland	1990/07	227.318182	0.000000
Switzerland	1990/08	195.334783	0.000000
Switzerland	1990/09	181.322222	0.000000

Ctxt	R
WSJ900813-0071 (paragraph 10)	0.120064
WSJ900820-0041 (paragraph 18)	0.075064
WSJ900827-0014 (paragraph 4)	0.062133
WSJ900806-0085 (paragraph 21)	0.060064

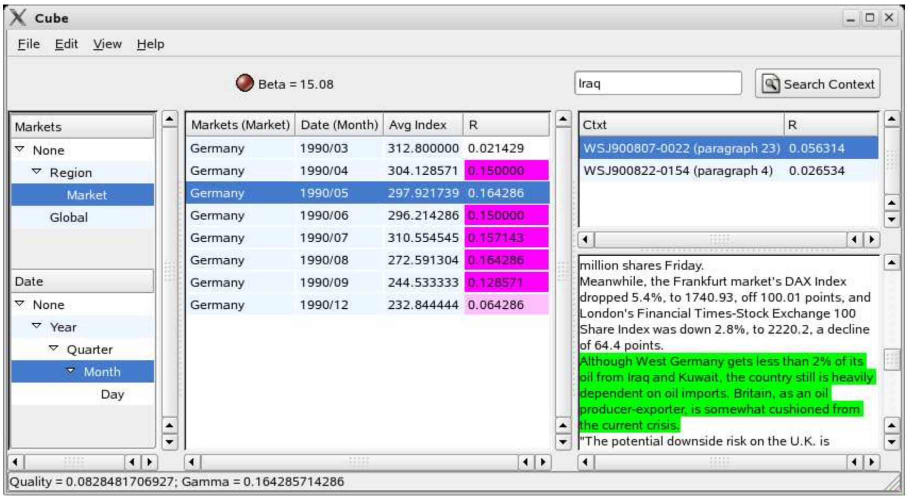
helped by buying from investment trust funds, which placed buy orders at limit prices, traders said.
Nippon Steel gained 10 yen to 529 yen ($3.59), while NKK added 7 yen to 514 yen.
The rest of the market fell broadly, regardless of sector.
Plant engineering companies were sold as their projects in Iraq and Kuwait have been frozen because of economic sanction by Japan against the two nations.
Chiyoda Corp. was down 90 yen to 1640 yen.

Quality = 1.24987011828; Gamma = 0.226570830621

Figure 6.4: OLAP window showing an *R-cube*

Now this window presents an *R-cube* that includes the relevance and the context values assigned to each fact of the original cube. Dark colours depict very relevant facts, whereas light colours mark the irrelevant ones. By rolling up to the *Region* and *Quarter* levels and ordering the facts by relevance, the analyst discovers that the most relevant facts involve the Asian markets and the third quarter of 1990. Then, the analyst decides to execute a drill-down operation to study the average index per month in the Asian countries. The most relevant facts correspond to Japan and the months of August and September. As can be seen in Figure 6.4, the Japanese market index had a sharp fall during these months, a fall of 100 points, whereas the average falls in the rest of markets were of about 10 points. By selecting the fact that represents the average index of the Japanese market in August, the system presents the documents that describe the context of this fact (see the right side of the window shown in Figure 6.4). In the highlighted paragraph of the first document, the analyst discovers that *"... plant engineering companies fell as their projects in Iraq and Kuwait were frozen because of the economic sanction of Japan against Iraq"*. Thus, the analyst concludes that it could be a good idea to watch Japanese investments now that there is a new conflict in the Middle East which could be as important for the financial markets as the Iraq war of 1990.

Finally, our user is particularly interested in the German market, since he/she has some business in this country. Figure 6.5 shows the result of dicing the *R-cube* of Figure 6.3 to study the German index values. The high β value points to a considerable loss of quality, meaning that the analysis results are not significant in the selected context (e.g., the Iraq war did not influence the German market too much). The analyst finds the explanation in the documents related to the selected facts: *"... West Germany gets less than 2% of its oil from Iraq and Kuwait..."*



Beta = 15.08

Iraq

Markets (Market)	Date (Month)	Avg Index	R
Germany	1990/03	312.800000	0.021429
Germany	1990/04	304.128571	0.150000
Germany	1990/05	297.921739	0.164286
Germany	1990/06	296.214286	0.150000
Germany	1990/07	310.554545	0.157143
Germany	1990/08	272.591304	0.164286
Germany	1990/09	244.533333	0.128571
Germany	1990/12	232.844444	0.064286

Ctxt	R
WSJ900807-0022 (paragraph 23)	0.056314
WSJ900822-0154 (paragraph 4)	0.026534

million shares Friday.
Meanwhile, the Frankfurt market's DAX Index dropped 5.4%, to 1740.93, off 100.01 points, and London's Financial Times-Stock Exchange 100 Share Index was down 2.8%, to 2220.2, a decline of 64.4 points.
Although West Germany gets less than 2% of its oil from Iraq and Kuwait, the country still is heavily dependent on oil imports. Britain, as an oil producer-exporter, is somewhat cushioned from the current crisis.
"The potential downside risk on the U.K. is

Quality = 0.0828481706927; Gamma = 0.164285714286

Figure 6.5: R-cube after the selection operation

6.3 Prototype Implementation

The prototype has been developed as a set of Python modules, and its interface in Glade and GTK+ [1]. Persistence has been provided by the cPickle Python module. In order to evaluate keyword-based searches over the XML collection, the document warehouse keeps a inverted file index [7] and implements the relevance modelling logic of the IR model presented in Section 4.2. Stemming and proper noun recognition tasks are executed by the Tree Tagger tool [78]. The corporate cubes and OLAP operations have been supported by implementing the data model and algebra operators of the base multidimensional model [62]. The fact extractor module provides the methods to build the *R-cube* by looking for date, stock market, and region references in the paragraphs of the documents. The occurrences of the major stock indices (e.g., Nikkei) and company names (e.g., General Motors) found in the text are also used to relate paragraphs and markets (Japan and U.S., respectively). A trie-like data structure is included within the fact extractor module for searching both actual and alternative (semantically related) dimension values in the text sections. Finally, analysis capabilities over *R-cubes* have been provided by implementing the data model and algebra operators discussed in Chapter 5.

CHAPTER 7

Conclusions

This last chapter presents the main results of the thesis and outline the future research lines. The chapter concludes by listing the publications resulted from this thesis work. Section 7.1 surveys the results of the thesis. Section 7.2 discusses the future work. Section 7.3 lists the main published contributions of the thesis.

7.1 Summary of Results

The work of this thesis links two historically separated research fields: the Data Warehouse and OLAP technologies, and the Information Retrieval systems.

In the last years, the organizations have successfully applied the DW and OLAP technologies to build decision support systems able to organize and analyze the huge amounts of structured data that companies store in their databases. On the other hand, the digital libraries, and more recently the web search engines, have based their discovery services on IR techniques. These applications deal with the massive storage and retrieval of documents where large text sections predominate.

Companies and organizations also circulate a considerable amount of information as text-rich documents. Nowadays, the Web has become the greatest source of information ever known. Organizations can now find highly valuable information about their business environment on the Internet. This situation opens a novel and interesting range of possibilities for the combination of DW and OLAP technology with IR systems. Imagine a decision support system able to obtain strategic information by combining the company sources of structured data with the text-rich documents available in either internal sources or external web repositories. This is the particular setting considered in this thesis and its major contribution: the construction of a DW contextualized with documents.

The advent of XML and related technologies is playing an important role in the future evolution of the Web. Most documents are now published in the Web in XML-like formats. Moreover, existing XML tagging techniques can be applied to give some structure to plain documents by identifying the visual styles that characterize the different document sections. The standardization of XML for Web data exchange has brought us to assume XML as the common format of the contextualized warehouse documents.

DW and OLAP tools are also affected by the irruption of XML and the Web revolution. This report has summarized the most relevant research on combining DW and XML-based Web technologies. As far as we know there does not exist any similar survey. The thesis has studied the advantages of XML as an integration tool for heterogeneous and distributed DW systems. It has also reviewed the work on the construction of web data repositories. These papers address the efficient storage, query processing, data acquisition, change control and schema integration of data gathered from web sources. The design of multidimensional databases for XML data, and the extension of OLAP techniques for analyzing XML external data have been also studied. However, these approaches only deal with highly structured XML data, and they are not suitable for exploiting the information described by text-rich documents during the OLAP analysis. Finally, the thesis has also surveyed some works on unstructured data and OLAP. These proposals are focused on applying multidimensional databases to build IR systems, but do not address the analysis of the factual data described in the textual contents of the documents.

In a contextualized warehouse, the facts of a traditional corporate DW are related to the text-rich documents that describe their circumstances. These text-rich documents are stored in an XML document warehouse. This thesis has shown how the dimension values mentioned in the textual contents of the documents can be used to relate documents and facts. The contextualized warehouse analysis cubes are called *R-cubes*. In order to materialize an *R-cube* the user supplies an IR condition (i.e., as sequence of keywords), a path expression and a set of MDX conditions. The IR condition and the path expression state the analysis context and are evaluated in the XML document warehouse. The MDX conditions are used for selecting the subset of corporate facts to analyze. The document fragments and facts that satisfy the established conditions are retrieved from the respective warehouse. Each selected fact is placed in the *R-cube* along with its relevance value with respect to the IR condition and the document fragments that describe its context. Thus, the *R-cubes* have two special dimensions: the context and the relevance dimension. The relevance dimension can be used to explore the most relevant portions of an *R-cube*. The user will be able to gain insight into the circumstances of a fact by retrieving its related documents in the context dimension.

This thesis has provided the contextualized warehouse with a formal framework that comprises:

- a novel IR model to retrieve the documents fragments that describe the analysis context and to estimate the relevance of the facts quoted in these documents,
- and a multidimensional data model and algebra for the *R-cubes*.

The IR model represents the XML documents as trees, so that the previous work concerning the evaluation of path expressions [14, 4] can be easily applied to the model. Given the IR condition and path expression used for establishing the context of analysis, we show how to retrieve the document subtrees that satisfy these conditions. In order to estimate the relevance of the document subtrees to the IR condition we follow [69] and calculate the smoothed relative frequency of the query keywords in the text sections of the document subtree. The query process discussed in the thesis ensures that only the most relevant elements of each selected subtree will be included in the result. The retrieval model adapts relevance modeling techniques [33] to measure the relevance of a fact by the probability of observing the fact in the set documents that are relevant to the IR condition. The probability of finding a fact in a document is computed by the relative frequency of its dimension values in the textual contents of the document. An interesting property of this approach is that the sum of the relevance values of the facts is always equal to one. Since not all document collections are suitable for every analysis, we propose to measure the quality of the result by computing the overall relevance of the retrieved documents.

The *R-cubes* multidimensional model extends the data model proposed in [62] with the relevance and the context dimensions. The unary algebra operators of [62] are also redefined to manage these special dimensions. The relevance values of the facts after a selection operation are increased by two factors. The first one represents the quality lost in the resulting *R-cube*. The second one measures the increment of importance of the selected facts in the documents, when the non-selected facts are no longer taken into account. In the result of the aggregate formation operation each new fact represents a group of facts of the original *R-cube*. The new facts are related to the documents that were associated with any of the original facts of the corresponding group. The relevance of each group is calculated by adding up the relevance values of the original facts in the group.

Finally, the thesis has presented a prototypical contextualized warehouse where the new analysis capabilities introduced in the *R-cubes* are tested in a real analysis scenario.

7.2 Future Work

This thesis work can be continued following different directions. These research lines include the extension of the *R-cubes* algebra with binary operators, testing the performance of the prototype system with larger data sets, the use of multidimensional techniques for exploring the XML document warehouse, or directly analyzing the factual data described by the documents without contextualizing a traditional corporate cube.

The *R-cubes* algebra remains to be completed with the binary operators. The main problem here is to compute the relevance of the facts in the result, since the input *R-cubes* may represent two distinct analysis contexts. For this purpose, IR data fusion mechanisms [17] can be applied to estimate the relevance of the involved facts. A different interesting issue is to use the binary algebra operators of the base multidimensional model [62] for combining an ordinary cube with an *R-cube*. Even more, the *R-cubes* can also be queried with the unary algebra operators proposed in the base model. In this case, the result may no longer be an *R-cube*, as the relevance or the context dimension may be projected away, or the facts relevance may not be updated. However, these operators could be applied to perform interesting analysis. For example, the context dimension may be used as a grouping category to calculate aggregations over the facts described in each document.

The prototype presented in the thesis shows the potential usefulness of a contextualized warehouse. Testing the performance of the system with larger data sets, and studying query evaluation techniques for *R-cubes*, like pre-aggregation strategies, is also future work. We are planning to develop a different prototype based on a commercial multidimensional database system. Other interesting topic for future research is to integrate the prototypes with existing web search engines, thus providing better scalability as well as the possibility to contextualize the data cubes with on-line WWW documents.

As previously said, some works propose to implement IR systems over multidimensional databases [42, 46]. There are a number of advantages in this setting. The OLAP cube dimensions provide an intuitive general-to-specific method for the analysis of the document contents, and the optimized evaluation of aggregation functions in multidimensional databases can be applied to efficiently compute the relevance formulas of the IR systems. Inspired by these works, in [65] we sketched an alternative architecture for the contextualized warehouse. In this architecture the user specifies the context of analysis by querying an OLAP cube that represents the document collection. The IR conditions are specified by navigating on concept dimensions that classify the documents by theme. In parallel, the facts described by the selected documents can be analyzed in an *R-cube*. The interactions between the documents cube and the *R-cubes* still require a deeper study. A prototype system is under construction.

The document warehouse may provide highly valuable strategic information about some facts that are not available in the corporate warehouse nor in external databases. We note that sometimes it is relatively easy to obtain these facts, for example, when they are presented as tables in the documents. However, many times documents contain already aggregated measure values. The main problem here is to automatically infer the implicit aggregation function that was applied (i.e., average, sum, etc.) Alternatively, the system could ask the user to guess the aggregation function by showing him/her the document contents. Different IR and information extraction-based methods for integrating documents and databases are discussed in [6]. Specifically, [6] proposes a strategy to extract from documents information related to (but not present in) the facts of the warehouse. This thesis has shown how the dimension values found in documents can be applied in the process of relating them with the corporate facts that have the same dimension values. Trying to directly analyze the facts extracted from the documents without considering the corresponding corporate facts is an even more challenging task. In this case, the analysis may involve facts that are incomplete (not all the dimensions may be quoted in the documents contents) and/or imprecise (if the dimension values found belong to non-base granularity levels). The *R-cubes* base model supports incompleteness and imprecision [62]. For the future, we plan to exploit these features to analyze the facts described in the documents that are not available in the corporate warehouse.

7.3 List of Publications

This section enumerates the publications that produced this thesis. For each paper we show a brief summary and point out the chapters of the thesis that mainly influenced it.

- Juan Manuel Pérez, Rafael Berlanga and María José Aramburu. Semi-Structured Information Warehouses: Requirements and Definition. In Proceedings of the *6th International Conference on Enterprise Information Systems (ICEIS' 2004).* Porto (Portugal), April 2004. Conference Proceedings, Vol.1. pp. 579-582. ISBN 972-1-8865-00-7.

  This paper states a set of requirements for the XML document warehouse. It discusses the need to support path expressions and IR conditions to establish the context of analysis, and sketches the main intuitions behind the facts relevance calculus. This paper is mainly related to the third and fourth chapters of the thesis.

- Juan Manuel Pérez, Rafael Berlanga and María José Aramburu. Técnicas de Análisis en Almacenes de Información Semi-estructurada: Limitaciones y Requerimientos. In Proceedings of the *VIII Jornadas de Ingeniería del Software y Bases de Datos (JISBD' 2003).* Alicante (Spain), November 2003. Ed. Universitat d'Alacant. pp. 727-736. ISBN 84-688-3836-5.

This is an extended version of the previous paper.

- Juan Manuel Pérez, Rafael Berlanga, María José Aramburu and Torben Bach Pedersen. Integrating Data Warehouses with Web Data: A Survey. DB Technical Report TR-18. Department of Computer Science (Aalborg University), November, 2006. 19 pages.

  This technical report surveys the research done on combining DW, OLAP and Web-based technologies. The report provided the contents of the second chapter of the thesis.

- Juan Manuel Pérez, Rafael Berlanga and María José Aramburu. A Document Model Based on Relevance Modeling Techniques for Semi-Structured Information Warehouses. In Proceedings of the *15th International Conference on Database and Expert Systems Applications (DEXA' 2004)*. Zaragoza (Spain), September 2004. Lecture Notes in Computer Science Vol. 3180. pp. 318-327. Springer, 2004. ISSN 0302-9743, ISBN 3-540-22936-1.

  This paper presented the retrieval model discussed in the fourth chapter of the thesis.

- Juan Manuel Pérez, Torben Bach Pedersen, Rafael Berlanga and María José Aramburu. IR and OLAP in XML Document Warehouses. In Proceedings of *Advances in Information Retrieval, 27th European Conference on IR Research (ECIR' 2005)*. Santiago de Compostela (Spain), March 2005. David E. Losada and Juan M. Fernández-Luna (Eds.) Lecture Notes in Computer Science, Vol. 3408. pp.536-539. Springer, 2005. ISSN 0302-9743, ISBN 3-540-25295-9.

  In this paper an alternative architecture for the contextualized warehouse is proposed. The architecture comprises an OLAP cube that represents document collection, and an *R-cube*. The user specifies the context of analysis by querying the documents cube. In parallel, the facts described by the selected documents can be analyzed in the *R-cube*. . This paper is related to the third chapter of the thesis.

- Juan Manuel Pérez, Rafael Berlanga, María José Aramburu and Torben Bach Pedersen. A relevance-extended multi-dimensional model for a data warehouse contextualized with documents. In Proceedings of the *8th ACM international workshop on Data warehousing and OLAP (DOLAP' 2005)*. Bremen (Germany), November 2005. pp. 19-28. ACM Press, 2005. ISBN 1-59593-162-7.

  This paper presented the architecture of the contextualized warehouse and the *R-cubes* data model and algebra. The paper is related to the third and fifth chapters of the thesis.

- Juan Manuel Pérez, Rafael Berlanga, María José Aramburu and Torben Bach Pedersen. Contextualizing Data Warehouses with Documents. To be published in *Decision Support Systems*. 2007. 18 pages.

  This paper is an extended version of the previous paper.

- Juan Manuel Pérez, Rafael Berlanga, María José Aramburu and Torben Bach Pedersen. R-Cubes: OLAP Cubes Contextualized with Documents. To be published in the Proceedings of the *IEEE 23rd International Conference on Data Engineering (ICDE' 2007).* Istanbul (Turkey), April 2007. 2 pages.

  This paper shows the prototype of a contextualized warehouse as presented in Chapter 6.

Bibliography

[1] Glade - a User Interface Builder for GTK+ and GNOME. http://glade.gnome.org.

[2] Morgan Stanley Capital International Inc. http://www.msci.com.

[3] Abelló, A. *YAM^2: A Multidimensional Conceptual Model*. PhD thesis, Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya (Spain), 2002.

[4] Aramburu, M. J., and Berlanga, R. A Temporal Object-Oriented Model for Digital Libraries of Documents. *Concurrency: Practice and Experience 13*, 11 (2001), 987–1011.

[5] Avnur, R., and Hellerstein, J. M. Eddies: Continuously Adaptive Query Processing. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data* (2000), ACM Press, New York, NY, pp. 261–272.

[6] Badia, A. Text warehousing: Present and future. In *Processing and Managing Complex Data for Decision Support*, J. Darmont and O. Boussaïd, Eds. Idea Group Publishing, 2006, pp. 96–121.

[7] Baeza-Yates, R. A., and Ribeiro-Neto, B. A. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.

[8] Beyer, K., Chambérlin, D., Colby, L. S., Özcan, F., Pirahesh, H., and Xu, Y. Extending XQuery for analytics. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data* (2005), ACM Press, New York, NY, pp. 503–514.

[9] Bhowmick, S. S. *WHOM: A Data Model and Algebra for a Web Warehouse*. PhD thesis, School of Computer Engineering, Nanyang Technological University (Singapore), 2001.

[10] Bhowmick, S. S., Mandria, S., and Ng, W. K. Detecting and Representing Relevant Web Deltas in Whoweda. *IEEE Transactions on Knowledge and Data Engineering 15*, 2 (2003), 423 – 441.

[11] Blaschka, M., Sapia, C., Hofling, G., and Dinter, B. Finding your way through multidimensional data models. In *Proceedings of the 9th International Workshop on Database and Expert Systems Applications* (1998), IEEE Computer Society, Washington, DC, pp. 198–203.

[12] Bruckner, R. M., Ling, T. M., Mangisengi, O., and Tjoa, A. M. A Framework for a Multidimensional OLAP Model using Topic Maps. In *Proceedings of the 2nd International Conference on Web Information Systems Engineering* (2001), IEEE Computer Society, Washington, DC, pp. 109–118.

[13] Chinenyanga, T. T., and Kushmerick, N. An expressive and efficient language for XML information retrieval. In *Proceedings of WebDB* (2001), ACM Press, New York, NY, pp. 1–6.

[14] Clark, J., and DeRose, S. XML path language (XPath) version 1.0. W3C recommendation, W3C, Nov. 1999. http://www.w3.org/TR/1999/REC-xpath-19991116.

[15] Cóbena, G., Abiteboul, S., and Marian, A. Detecting changes in XML documents. In *Proceedings of the 18th International Conference on Data Engineering* (2002), IEEE Computer Society, Washington, DC, pp. 41–52.

[16] Codd, E. F. Providing OLAP to user-analysts: An IT mandate, 1993.

[17] Croft, W. B. Combining approaches to information retrieval. In *Advances in Information Retrieval*. Kluwer, 2000, pp. 1–36.

[18] Daconta, M. C., Obrst, L. J., and Smith, K. T. *The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management*. Wiley Publishing Inc., 2003.

[19] Danger, R., Berlanga, R., and Ruiz-Shulcloper, J. CRISOL: An Approach for Automatically Populating Semantic Web from Unstructured Text Collections. In *Proceedings of 15th International Conference on Database and Expert Systems Applications* (2004), Springer, Berlin, pp. 243–252.

[20] Deach, S., Graham, T., Berglund, A., Grosso, P., Caruso, J., Richman, J., Adler, S., Milowski, R. A., Gutentag, E., Zilles, S., and Parnell, S. Extensible stylesheet language (XSL) version 1.0. W3C recommendation, W3C, Oct. 2001. http://www.w3.org/TR/2001/REC-xsl-20011015/.

[21] DeRose, S., Maler, E., and Orchard, D. XML linking language (XLink) version 1.0. W3C recommendation, W3C, June 2001. http://www.w3.org/TR/2001/REC-xlink-20010627/.

[22] Fallside, D. C., and Walmsley, P. XML schema part 0: Primer second edition. W3C recommendation, W3C, Oct. 2004. http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/.

[23] Foster, I., and Kesselman, C. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, 1998.

[24] Fuhr, N., and Grojohann, K. XIRQL: A query language for information retrieval in XML documents. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval* (2001), ACM Press, New York, NY, pp. 172–180.

[25] Golfarelli, M., Rizzi, S., and Vrdoljak, B. Data warehouse design from XML sources. In *Proceedings of the 4th ACM international conference on Data warehousing and OLAP* (2001), ACM Press, New York, NY, pp. 40–47.

[26] Grishman, R. Information Extraction: Techniques and Challenges. In *SCIE '97: International Summer School on Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology* (1997), Springer, Berlin, pp. 10–27.

[27] Harman, D. K. Overview of the Third REtrieval Conference (TREC-3). In *Overview of The Third Text REtrieval Conference (TREC-3)*, D. K. Harman, Ed. NIST Special Publication 500-225, 1995, pp. 1–19.

[28] Hümmer, W., Bauer, A., and Harde, G. XCube - XML For Data Warehouses. In *Proceedings of the 6th ACM international workshop on Data warehousing and OLAP* (2003), ACM Press, New York, NY, pp. 33–40.

[29] Inmon, W. H. *Building the Data Warehouse*. John Wiley & Sons, 2005.

[30] Jensen, M. R., Møller, T. H., and Pedersen, T. B. Specifying OLAP Cubes on XML Data. *Journal of Intelligent Information Systems 17*, 2/3 (2001), 255 – 280.

[31] Jensen, M. R., Møller, T. H., and Pedersen, T. B. Converting XML DTDs to UML diagrams for conceptual data integration. *Data & Knowledge Engineering 44*, 3 (2003), 323 – 346.

[32] Kimball, R., and Ross, M. *The Data Warehouse Toolkit*. John Wiley & Sons, 2002.

[33] Lavrenko, V., and Croft, W. B. Relevance-Based Language Models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval* (2001), ACM Press, New York, NY, pp. 120–127.

[34] Lavrenko, V., Feng, S. L., and Manmatha, R. Statistical Models for Automatic Video Annotation and Retrieval. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing* (2003), pp. 17–21.

[35] Lee, J., Grossman, D., and Orlandic, R. MIRE: A Multidimensional Information Retrieval Engine for Structured Data and Text. In *Proceedings of the International Conference on Information Technology: Coding and Computing* (2002), IEEE Computer Society, Washington, DC, pp. 224–229.

[36] Lee, J., Grossman, D., and Orlandic, R. An Evaluation of the Incorporation of a Semantic Network into a Multidimensional Retrieval Engine. In *Proceedings of the 12th international conference on Information and knowledge management* (2003), ACM Press, New York, NY, pp. 572–575.

[37] Llidó, D. M., Berlanga, R., and Aramburu, M. J. Extracting Temporal References to Assign Document Event-Time Periods. In *Proceedings of the 12th International Conference on Database and Expert Systems Applications* (2001), Springer, Berlin, pp. 62–71.

[38] Losada, D. E. *A Logical Model of Information Retrieval based on Propositional Logic and Belief Revision.* PhD thesis, Departamento de Computación, Universidad de A Coruña (Spain), 2001.

[39] Maler, E., Bray, T., Paoli, J., Yergeau, F., and Sperberg-McQueen, C. M. Extensible markup language (XML) 1.0 (fourth edition). W3C recommendation, W3C, Aug. 2006. http://www.w3.org/TR/2006/REC-xml-20060816.

[40] Mangisengi, O., Huber, J., Hawel, C., and Essmayr, W. A Framework for Supporting Interoperability of Data Warehouse Islands Using XML. In *In Proceedings of the 3rd International Conference on Data Warehousing and Knowledge Discovery* (2001), Springer, Berlin, pp. 328–338.

[41] Marian, A., Abiteboul, S., Cóbena, G., and Mignet, L. Change-centric management of versions in an XML warehouse. In *Proceedings of the 27th International Conference on Very Large Data Bases* (2001), Morgan Kaufmann Publishers Inc., San Francisco, CA, pp. 581–590.

[42] McCabe, M. C., Lee, J., Chowdhury, A., Grossman, D., and Frieder, O. On the design and evaluation of a multi-dimensional approach to information retrieval. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval* (2000), ACM Press, New York, NY, pp. 363–365.

[43] Microsoft Corp. and Hyperion Solutions Corp. XML for Analysis Specification. http://xmla.org, 2001.

[44] Miller, E., and Manola, F. RDF primer. W3C recommendation, W3C, Feb. 2004. http://www.w3.org/TR/2004/REC-rdf-primer-20040210/.

[45] Moole, B. R. A Probabilistic Multidimensional Data Model and Algebra for OLAP in Decision Support Systems. In *Proceedings of IEEE SoutheastCon* (2003), pp. 18–30.

[46] Mothe, J., Chrisment, C., Dousset, B., and Alaux, J. Doccube: Multi-dimensional visualisation and exploration of large document sets. *Journal of the American Society for Information Science and Technology 54*, 7 (2003), 650–659.

[47] Nguyen, B., Abiteboul, S., Cóbena, G., and Preda, M. Monitoring XML data on the web. In *Proceedings of the 2001 ACM SIGMOD international conference on Management of data* (2001), ACM Press, New York, NY, pp. 437–448.

[48] Nguyen, T. B., Tjoa, A. M., and Mangisengi, O. Meta Cube-X: An XML Metadata Foundation of Interoperability Search among Web Data Warehouses. In *Proceedings of the Third International Workshop on Design and Management of Data Warehouses* (2001), CEUR-WS.org., pp. 8.1–8.8.

[49] Nguyen, T. B., Tjoa, A. M., and Wagner, R. Conceptual Multidimensional Data Model Based on MetaCube. In *Proceedings of the First International Conference on Advances in Information Systems* (2000), Springer, Berlin, pp. 24–33.

[50] Niemi, T., Niinimäki, M., Nummenmaa, J., and Thanisch, P. Constructing an OLAP Cube from Distributed XML Data. In *Proceedings of the 5th ACM international workshop on Data Warehousing and OLAP* (2002), ACM Press, New York, NY, pp. 22–37.

[51] Niemi, T., Niinimäki, M., Nummenmaa, J., and Thanisch, P. Applying Grid Technologies to XML Based OLAP Cube Construction. In *Proceedings of the 5th International Workshop on Design and Management of Data Warehouses* (2003), CEUR-WS.org., pp. 4.1–4.13.

[52] OMG – Object Management Group. Unified Modeling Language (UML). http://www.uml.org, 2004.

[53] Park, B.-K., Han, H., and Song, I.-Y. XML-OLAP: A Multidimensional Analysis Framework for XML Warehouses. In *Proceedings of the 6th International Conference on Data Warehousing and Knowledge Discovery* (2005), Springer, Berlin, pp. 32–42.

[54] Pedersen, D., Pedersen, J., and Pedersen, T. B. Integrating XML Data in the TARGIT OLAP System. In *Proceedings of the 20th International Conference on Data Engineering* (2004), IEEE Computer Society, Washington, DC, pp. 778–781.

[55] Pedersen, D., and Pedersen, T. B. Achieving Adaptivity for OLAP-XML Federations. In *Proceedings of the 6th ACM international conference on Data warehousing and OLAP* (2003), ACM Press, New York, NY, pp. 25–32.

[56] Pedersen, D., and Pedersen, T. B. Synchronizing XPath Views. In *Proceedings of the 8th International Database Engineering and Application Symposium* (2004), IEEE Computer Society, Washington, DC, pp. 149–160.

[57] Pedersen, D., Pedersen, T. B., and Riis, K. The Decoration Operator: A Foundation for On-Line Dimensional Data Integration. In *Proceedings of the International Database Engineering and Applications Symposium* (2004), IEEE Computer Society, Washington, DC, pp. 357–366.

[58] Pedersen, D., Riis, K., and Pedersen, T. B. Cost Modeling and Estimation for OLAP-XML Federations. In *Proceedings of the 6th International Conference on Data Warehousing and Knowledge Discovery* (2002), Springer, Berlin, pp. 245–223.

[59] Pedersen, D., Riis, K., and Pedersen, T. B. Query Optimization for OLAP-XML Federations. In *Proceedings of the 5th ACM international workshop on Data Warehousing and OLAP* (2002), ACM Press, New York, NY, pp. 57–64.

[60] Pedersen, D., Riis, K., and Pedersen, T. B. XML-Extended OLAP Querying. In *Proceedings of the 14th International Conference on Scientific and Statistical Database Management* (2002), IEEE Computer Society, Washington, DC, pp. 195–206.

[61] Pedersen, T. B., and Jensen, C. S. Multidimensional databases. In *The Industrial Information Technology Handbook*, R. Zurawski, Ed. CRC Press, 2005, pp. 1–13.

[62] Pedersen, T. B., Jensen, C. S., and Dyreson, C. E. A foundation for capturing and querying complex multidimensional data. *Information Systems 26*, 5 (2001), 383–423.

[63] Pepper, S., and Moore, G. XML Topic Maps (XTM) 1.0. TopicMaps.Org Specification, Aug. 2001. http://www.topicmaps.org/xtm/1.0/xtm1-20010806.html.

[64] Pérez, J. M., Berlanga, R., and Aramburu, M. J. Semi-structured Information Warehouses: An approach to a document model to support their construction. In *Proceedings of the 6th International Conference on Enterprise Information Systems* (2004), pp. 579–582.

[65] Pérez, J. M., Pedersen, T. B., Berlanga, R., and Aramburu, M. J. IR and OLAP in XML Document Warehouses. In *Proceedings of Advances in Information Retrieval: 27th European Conference on IR Research* (2005), Springer, Berlin, pp. 536–539.

[66] Pokorný, J. Modelling Stars Using XML. In *Proceedings of the 4th ACM international conference on Data warehousing and OLAP* (2001), ACM Press, New York, NY, pp. 24–31.

[67] Ponniah, P. *Data Warehousing Fundamentals: A Comprehensive Guide for IT Processionals.* Wiley, 2001.

[68] Pons, A., Berlanga, R., and Ruíz-Shulcloper, J. Building a Hierarchy of Events and Topics for Newspaper Digital Libraries. In *Proceedings of Advances in Information Retrieval: 25th European Conference on IR Research* (2003), Springer, Berlin, pp. 588–596.

[69] Ponte, J. M., and Croft, W. B. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval* (1998), ACM Press, New York, NY, pp. 275–281.

[70] Priebe, T., and Pernul, G. Towards Integrative Enterprise Knowledge Portals. In *Proceedings of the 12th international Conference of Information and Knowledge Management* (2003), ACM Press, New York, NY, pp. 216–223.

[71] Robertson, S. The probability ranking principle in IR. In *Readings in information retrieval.* Morgan Kaufmann Publishers Inc., 1997, pp. 281–286.

[72] Robertson, S., and Jones, K. S. Relevance weighting of search terms. *Journal of the American Society of Information Science 27*, 3 (1976), 129–146.

[73] Robertson, S., and Walker, S. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval* (1994), Springer-Verlag New York, Inc., New York, NY, pp. 232–241.

[74] Robie, J., Fernández, M. F., Chamberlin, D., Boag, S., Florescu, D., and Siméon, J. XQuery 1.0: An XML query language. Candidate recommendation, W3C, June 2006. http://www.w3.org/TR/2006/CR-xquery-20060608/.

[75] Salton, G. *The SMART Retrieval System – Experiments in Automatic Document Processing*. Prentice Hall Inc., 1971.

[76] Salton, G., and McGill, M. J. *Introduction to Modern Information Retrieval*. McGraw-Hill Book Co., 1983.

[77] Sanz, I., Berlanga, R., and Aramburu, M. J. Gathering metadata from web-based repositories of historical publications. In *Proceedings of the 9th International Workshop on Database Applications and Expert Systems* (1998), IEEE Computer Society, Washington, DC, pp. 473–478.

[78] Schimd, H. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing* (1994).

[79] Singahl, A., Buckley, C., and Mitra, M. Pivoted document length normalization. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval* (1996), ACM Press, New York, NY, pp. 21–29.

[80] Song, F., and Croft, W. B. A general language model for information retrieval. In *Proceedings of the eighth international conference on Information and knowledge management* (1999), ACM Press, New York, NY, pp. 316–321.

[81] Spofford, G. *MDX Solutions with Microsft SQL Server Analysis Services*. John Wiley & Sons, 2001.

[82] The Web Warehousing & Mining Group. Whoweda. http://www.cais.ntu.edu.sg:8000/˜whoweda.

[83] Trujillo, J., Luján-Mora, S., and Song, I. Applying UML and XML for Designing and Interchanging Information for Data Warehouses and OLAP Applications. *Journal of Database Management 14*, 1 (2004), 41 – 72.

[84] Tseng, F., and Chen, C. Integrating Heterogeneous Data Warehouses Using XML Technologies. *Journal of Information Science 31*, 3 (2005), 209 – 229.

[85] van Harmelen, F., and McGuinness, D. L. OWL web ontology language overview. W3C recommendation, W3C, Feb. 2004. http://www.w3.org/TR/2004/REC-owl-features-20040210/.

[86] Widom, J. Research problems in data warehousing. In *Proceedings of the fourth international conference on Information and knowledge management* (1995), ACM Press, New York, NY, pp. 25–30.

[87] Xyleme, L. A dynamic warehouse for XML data of the Web. *IEEE Data Engineering Bulleting 24*, 2 (2001), 40 – 47.

[88] Yinyan, C., Lim, E. P., and Ng, W. K. Storage Management of a Historical Web Warehousing System. In *Proceedings of 11th International Conference on Database and Expert Systems Applications* (2000), Springer, Berlin, pp. 457–466.

[89] Zhuge, Y., and Garcia-Molina, H. Graph Structured Views and their Incremental Maintenance. In *Proceedings of the 14th International Conference on Data Engineering* (1998), IEEE Computer Society, Washington, DC, pp. 116–125.