INTEGRATING UNSTRUCTURED DATA USING PROPERTY PRECEDENCE

TAO CHEN

Integrating Unstructured Data Using Property Precedence

by

© Tao Chen

A thesis submitted to the

School of Graduate Studies

in partial fulfilment of the

requirements for the degree of

(Master of Science)

Computer Science

Memorial University of Newfoundland

August 2008

St. John's Newfoundland

Abstract

Data integration involves combining data from a variety of independent data sources to provide a unified view of data in these sources. One of the challenging problems in data integration is to reconcile the structural and semantic differences among data sources. Many approaches have been introduced to resolve the problem. However, most of these models have difficulties in handling data with less structure and varying granularity. This thesis focuses on developing a novel data integration approach for unstructured data. To identify properties from unstructured data, we adapt a probability model to identify multi-term properties. To address the granularity issue, we use the concept of Property Precedence. Unlike other approaches, Property Precedence does not require that data be class-based and takes 'property' as the basic semantic construct. Considering that unstructured data might contain properties that are not explicitly revealed by the description, we design a model that derives knowledge about a property from the instances known to possess the property. We evaluate this model and the results indicate that it is capable of inferring that an instance possesses a property when this information is not explicit in the data. We build a property precedence schema using the above model to help decide the existence of a property in the instance. We compare the results with property precedence schemas built by other approaches and demonstrate that our approach performs better than the others. Finally, we implement queries based on property precedence and show that these queries overcome the semantic gap between data sources and can retrieve relevant data that cannot be retrieved using other approaches.

Acknowledgments

First I would like to thank my thesis supervisor Dr. Jeffrey Parsons for supporting me academically and financially. I would like to thank him for guiding me into such an interesting and promising research area. I could not finish this thesis without his guidance and inspiration.

I would like to thank my wife Wang Lei. Without her love and support, my life would be less colorful and less happy.

I would like to take this chance to thank my dearest friends, Chen Zheng, Hu Ting, Li Xueming, Lin Haifeng, Grant Strong, Su Jianmin, Wu Xiaonan, Wu Junwei and Zhang Xiaoning. Thanks them for all the stimulating discussions and for all the good time we spent together.

I would like to thank everyone in the Computer Science Department at Memorial University for providing the perfect studying environment, especially, Dr. Wolfgang Banzhaf, Ms. Elaine Boone, Dr. Peter Chen, Ms. Sharon Deir, Dr. Siwei Lu, Dr. Jian Tang, Dr. Krishnamurthy Vidyasankar for their encouragement and support.

Table of Contents

List of Tables

List of Figures

1. Introduction

Data integration is defined as the process of combining a variety of data sources and providing a unified view to data in these sources (Halevy, Rajaraman and Ordille 2006). In management practice, data integration, usually as known as Enterprise Information Integration, is crucial for large enterprises that own multiple independently developed data sources and need to query across these data sources. With the development of Internet, integrating data on the web has become an important branch of data integration. In the era of Web 2.0, blogs and social network services are gaining popularity. A large amount of data is generated daily in the form of blogs, reviews and comments. Much of this data is unstructured. New web applications such as mashups intend to be able to query across these data sources. However, such goals are hard to accomplish. As with other data integration applications, there are two main challenges: 1) structural heterogeneity - different data sources apply different data models or schemas, and 2) semantic heterogeneity - different data sources use different vocabularies (Özsu and Valduriez 1999). Furthermore, as data are unstructured, the approach that satisfies such needs must be able to handle data with less structure and varying granularity.

A significant amount of research has addressed the first two challenges of data integration. A common approach to solve the heterogeneities is to employ a mediated schema in order to bridge the differences among data sources (Halevy, Rajaraman and Ordille 2006). Two major research projects TSIMMIS (Garcia-Molina, et al. 1997) and Information Manifold (Levy, Rajaraman and Ordille 1996a, 1996b) introduce two different approaches to describe the relation between data sources and mediated schema. One is known as global-as-view approach (GAV), in which the mediated schema is described as views of data sources, and the other is known as local-as-view approach (LAV), in which the data sources are presented as views of mediated schema. These two approaches provide well-understood and expressive methods to describe data sources. However, in practice, writing such data source description or schema mappings is very challenging when the number of the data sources is large and the data sources are complex (Halevy, Rajaraman and Ordille 2006). As a result, a considerable amount of research has focused on automatically or semi-automatically generating schema mappings (Chuang, Chang and Zhai 2007, Doan, Domingos and Halevy 2001, Do and Rahm 2002, Kang and Naughton 2003, Madhavan, et al. 2005).

Schema mapping is a process for reconciling semantic heterogeneity. The fundamental problem of schema mapping is schema matching, which is to identify how certain elements in one schema are equivalent to certain elements in another schema (Rahm and Bernstein 2001). Some work matches schemas based on the information in the schemas. For example, they consider linguistic similarities of names and descriptions of attributes, similar data types, and overlapping primary keys and foreign keys. However, the assumption that the schema information is available is generally not valid for data integration over web, as most web applications only provide partial schema information or do not provide any. Recent research (Chuang, Chang and Zhai 2007, Doan, Domingos and Halevy 2001, Do and Rahm 2002, Kang and Naughton 2003, Madhavan, et al. 2005) considers not only the schema information but also the data values of the attributes. They apply machine learning techniques to suggest the similarities between elements of different schemas. Though these approaches are more practical in the web integration scenario, they match elements only when two elements are assumed equivalent in semantics and fail to explore richer semantic relations. This limits their capability since semantic interoperability not only exists in two semantically equivalent elements, but also exists in other forms of semantic relations between elements such as containment. Furthermore, most of these approaches assume data is class-based and this assumption may not hold in web integration as data in the web are in different granularities and are less structured, and schema information is limited.

Parsons and Wand (2003) proposed Property Precedence as a possible way to integrate schemas and overcome some of the difficulties of matching. Unlike data integration models and current schema matching approaches, Property Precedence relaxes the *assumption of inherent classification*, the assumption that data is organized into a class-based schema (Parsons and Wand 2000). Property Precedence is based on the existence of instances and properties independent of any classification. By treating properties as basic semantic constructs, it is possible for the model to handle data with different granularities and less structure. Parsons and Wand suggest a semantics-based mediated schema to accommodate different data sources and to manage semantic relations between properties. To discover interoperable semantic relations between properties, instead of focusing on structural matching, property precedence focuses on the set of instances that possess different properties and the containment relation between them: one property may be a more general representation of another if the instances possessing the first property subsume those possessing the second.

In this thesis we apply Property Precedence to integrate unstructured data sources since: (1) unstructured data are not class-based and do not provide any explicit schema information; (2) unstructured data sources are at the most coarse granularity level and, at that level, semantic reconciliation usually cannot be performed; and (3) integrating unstructured data has a great demand in the era of Web 2.0. In the experiment, we use the Retuers-21578 data set (Reuters-21578, Distribution 1.0) as the unstructured data source, which has 21578 unstructured documents covering business news. We exploit the capability of Property Precedence to reconcile semantic heterogeneity and demonstrate that Property Precedence is capable of handling data in different granularities and with no structure.

We introduce an approach to automatically build a Property Precedence schema on a data set to bridge the semantic gap among documents. We develop a system to integrate these documents and to query them through a unified interface. Our work demonstrates that Property Precedence can successfully contribute to reconciling semantic heterogeneity without assuming data is class-based. The result verifies the effectiveness of our approach to build property precedence schema on unstructured data sources. The specific contributions of the thesis are the following.

- We introduce a method to automatically identify the properties in unstructured data. This method provides the basis for applying Property Precedence.
- We present a novel method to infer a property of an instance that is not explicitly stated in the description of the instance. This method enables Property Precedence to be more accurate in discovering the semantic relations in unstructured data sources.
- We develop an algorithm to discover the precedence relations between properties and to build a property precedence schema. We evaluate the effectiveness of our approach and demonstrate that our approach is more effective than approaches based either on terms appearing in sources or on accessing related terms using WordNet (WordNet, a lexical database for the English language).
- We define and develop querying based on Property Precedence and evaluate the effectiveness of Property Precedence in reconciling semantic differences. The result indicates Property Precedence is capable of resolving the semantic heterogeneity.

The material in this thesis is organized in seven chapters. Chapter 2 reviews related work. Chapter 3 describes the method to identify the properties from unstructured data. Chapter 4 discusses the approach to infer the existence of a property in an instance when the instance does not explicitly indicate it. Chapter 5 presents the algorithm to build property precedence schema and analyzes different building methods. Chapter 6 presents a query system based on Property Precedence and evaluates the effectiveness of Property Precedence in resolving semantic heterogeneity. We summarize the research contributions in Chapter 7.

2. Related Work

To bridge the difference in data sources, a common practice in data integration is to employ a mediated schema. Much research has focused on how the mediated schema maps to the data sources and how the queries on the mediated schema are rewritten to the queries on the data sources. In this chapter we review two major approaches: Global-as-View (GAV) and Local-as-View (LAV). As the scale of data integration becomes larger, manually creating the mappings between the mediated schema and the data sources becomes extremely challenging and limits the application of data integration on a larger scale. A considerable amount of research has been devoted to how to create the mappings automatically. In this chapter we review developments in this area. As research in information retrieval has made significant contributions to querying related unstructured documents from multiple sources, we also review techniques in information retrieval. In addition, we provide a review of Property Precedence.

2.1 GAV and LAV

In the discussion of GAV and LAV, it is common to use datalog notation. Conjunctive queries (Ullman 1988), which are able to express select-join queries, such as SQL, have the following form:

$$q(X) :- p_1(X_1), p_2(X_2), \ldots, p_n(X_n)$$

where q, $p_1, \ldots, p_n$ are predicate names and q refers to a view or a query and $p_1, \ldots, p_n$ refer to tables or relations in databases. q(X) is called head and $p_i(X_i)$ are called subgoals. The tuples X, X_1, … ,X_n contains variables or constants. The duplicates of variables in X_1, … , X_n indicate the equijoin and some predicates are comparison between variables in X_1, … , X_n. The following is an example that expresses an SQL query as a conjunctive query:

```
SQL query: select instructor.name, student.name, course.title
           from instructor, student, course, registration
           where instructor.id = registration.instructor_id
           and student.id = registration.student_id
           and course.id=registration.course_id
           and registration.term >= 'Fall 2005'.
```

```
Conjunctive query: q(instructor_name, student_name, course_title):-
                   instructor(instructor_name, instructor_id),
                   student(student_name, student_id),
                   course(course_title, course_id),
                   registration(instructor_id, student_id, course_id, term),
                   term >= 'Fall 2005'
```

Global-as-View (GAV) is first introduced in the research project TSIMMIS (Hammer, et al. 1995, Garcia-Molina, et al. 1997). The GAV approach describes the mediated schema as views of data sources. In addition, TSIMMIS proposed an OEM (Object Exchange Model) which accommodates different data such as relational data and XML, and also conforms to datalog. An OEM contains 4 parts: ID, label, type and value (which could be an atomic value like a string, an ID or a set). An example using GAV is:

Mediated schema:

instructor(name, id), student(name, id), course(title, id), registration(instructor_id, student_id, course_id, term)

Data source 1:

P1(instructor_name, instructor_id, course_title, course_id)

Data source 2:

P2(instructor_name, instructor_id, student_name, student_id), P3 (student_id, term)

GAV description:

instructor(instructor_name, instructor_id) :- P1(instructor_name, instructor_id, course_title, course_id)
instructor(instructor_name, instructor_id) :- P3(instructor_name, instructor_id, student_name, student_id)
student(student_name, student_id) :- P2(instructor_name, instructor_id, student_name, student_id)
course(course_title, course_id):- P1(instructor_name, instructor_id, course_title, course_id)
registration(instructor_id, student_id, course_id, term):- P1(instructor_name, instructor_id, course_title, course_id), P2(instructor_name, instructor_id, student_name, student_id), P3 (student_id, term)

Query processing in GAV is a process of view unfolding: each subgoal of the query expands until every subgoal in the query corresponds to the relations in data sources. GAV is a straightforward approach and easy to implement but when the data sources increase every GAV description needs to be updated accordingly. This updating process can be overwhelming as hundreds of GAV descriptions may have already been created and updating each of them is prone to errors. The GAV approach may not be very friendly for new data sources. Information Manifold (Levy, Rajaraman and Ordille 1996a, 1996b) suggests a different approach called Local-as-View (LAV). In the LAV, the data sources are described as views of the mediated schema. Thus, new data sources do not need to be aware of the existence of others (Halevy, Rajaraman and Ordille 2006), which allows local changes to remain local. An example of using LAV is:

LAV description:

P1(instructor_name, instructor_id, course_title, course_id) :- instructor(instructor_name, instructor_id), course(course_title, course_id), registration(instructor_id, student_id, course_id, term)

P2(instructor_name, instructor_id, student_name, student_id) :- instructor(instructor_name, instructor_id), student(student_name, student_id), registration(instructor_id, student_id, course_id, term)
P3 (student_id, term) :- registration(instructor_id, student_id, course_id, term)

In exchange for greater scalability, query rewriting in LAV is more complex than in GAV. As the data sources are described as views of the mediated schema, query processing in LAV is to rewrite the query using the given views (Halevy 2001, Levy, Mendelzon and Sagiv 1995). Halevy et al. introduce the bucket algorithm to solve the problem. The MiniCon algorithm (Pottinger and Halevy 2001) further investigates how the variables in the query relate to the views and uses this information to efficiently process queries. Also, research such as (Manolescu, Florescu and Kossmann 2001) translates XML Queries into conjunctive queries such that LAV can be applied to XML data sources.

2.2 Schema Matching

The objective of schema matching is to identify how certain elements in schema S1 are related to certain elements in schema S2 (Rahm and Bernstein 2001). Schema matching is very challenging when the data sources are complex and there are many of them. Automatic or semi-automatic matching can be helpful in large scale data integration projects. Early research (Palopoli, Saccá and Ursino 1999, Palopoli, Terracina and Ursino 2003) singly relies on the schema information. They evaluate the linguistic similarities of names and descriptions of attributes, similar data types and overlapping primary keys and foreign keys, and use this information to derive the matching decision.

Recent research such as LSD (Doan, Domingos and Halevy 2001) introduces approaches that consider not only the schema information but also data values of the attributes. LSD introduces a framework that incorporates decisions that are derived from different types of information and render a more accurate decision. This framework includes base learner, meta-learner, predication converter and constraint handler. Different base learners process different information and compute the confidence for possible matching. For example, a name learner takes the name of an attribute in one data source, computes the similarities with attributes in another data source, and assigns the confidence of matching this attribute with attributes in another data sources according to the computed similarities. As the performance of different base learners may vary when matching different attributes, a meta-learner determines weights of base learners with regard to the attributes that the base learners work on. A predication converter combines the results from meta-learners to derive a final result and a constraint handler ensures the final result does not violate the existing constraints. Do and Rahm (2002) introduce a similar framework with richer base learners and that reuses previous matching results to allow transitive matching. For example, if s1 matches s2 with confidence c1 and s2 matches s3 with confidence c2, the system can derive s1 matches s3 with confidence c3 usually less than c1 and c2. Kang and Naughton (2003)'s work takes the dependency between attributes into consideration and models the dependency between attributes as a graph. Thus, schema matching is reduced to graph matching. As prior knowledge is important to schema matching, Madhavan et al. (2005) suggest using a corpus to help the process. They discover elements similar to a data element in the corpus. The knowledge of the element can be thereby augmented by integrating knowledge of these similar elements. In the case where two elements cannot provide enough information for matching, the matching still can be performed by matching their similar elements in the corpus. Chuang et. al. (2007) also notices the benefit that the corpus can bring. They consider building schema matching for multiple sources as a sequence of tasks. The k-th schema matching task should be able to benefit from the previous k-1 finished tasks. Also each individual schema matching should be consistent with the others.

2.3 Information Retrieval and Information Extraction

In information retrieval, querying related documents from difference sources is a major task. As documents developed by different people may use different words to express the same idea, queries need to discover shared concepts underlying different wordings in order to find related documents. One approach to solve the problem is to model the document as a collection of concepts and identify the corresponding words in the document related to the concepts. The popular TFIDF model (Spärck Jones 1972) identifies the importance of each concept in the document but this model is incapable of identifying the synonyms of the concept. Latent semantic indexing (Deerwester, et al. 1990) applies singular value decomposition to identify a subspace of the original word-document space. This new subspace, usually considered as the concept-document space, captures the most variance of the document collection. The concept dimension of this subspace is the linear combination of the original word dimension. The different words that have been assigned to the same concept dimension are considered as synonymies. The same word that has been assigned to different concept dimensions is considered as a polysemy. Compared with latent semantic indexing, probabilistic latent semantic indexing (Hofmann 1999) is more comprehensive as it is capable of estimating the joint probability of words and concepts while latent semantic indexing only estimate the probability of words conditional on concepts. In probabilistic latent semantic indexing, each word in a document is generated by a collection of unobserved variables and these unobserved variables are considered as concepts. By applying Expectation Maximization algorithm (Dempster, Laird and Rubin 1977), the probability of the words conditioned on the concept can be determined. For each concept, the word with higher probability is considered as the word reflecting this concept. Latent Dirichlet allocation (Blei, Ng and Jordan 2003) further improves the probabilistic latent semantic indexing by treating the weights of the concepts in a document as hidden variable that can be derived from the document collection. It enables the model to fit the unseen document better and avoids overfitting. Research such as (Ampazis and Perantonis 2004, Li, et al. 2008, Georgakis, Kotropoulos and Pitas 2002, Kurland 2008, Liu, et al. 2008) utilizes these or similar models to map the original document to a concept space. In such a concept space, they further apply clustering algorithms to identify related documents in spite of different wordings.

All of the above models and methods are based on the "bag-of-words" assumption: the order of words in a document is exchangeable. Obviously, such an assumption ignores logical structures in human language. Research such as (Arazy and Woo 2007) indicates the collocation of words within sentences or across sentences is capable of enhancing the performance of information retrieval.

Information extraction, being different from querying related documents, extracts facts from a large collection of documents. The facts are like *St. John's* is a *City* (unary relation) and *St. John's* is a *City of Newfoundland* (n-ary relation). These facts are the instances of given relations and in the above examples the relations are *City* and *CityOf.* The application of information extraction includes automatically building ontology (Soderland and Mandhani 2007). After the instances of a relation are extracted, a document can be identified by a query of the relation even when the words of the relation do not appear in the document. An extracting approach without supervision is discussed in (Etzioni, et al. 2005). The approach does not require any manually identified instances of a relation as training data. By taking the advantage of the huge amount of information on the web, this approach first applies extracting patterns such as *cities such as C1, ..., C2* or *C1 city of C2* to identify the candidate instances. Each candidate is further assessed to verify their validity by calculating the mutual information between the candidate and alternative expressions of the relation, for instance, *Town* is an alternative expression of *City*. It is easy to observe that the instances extracted by this approach are far from comprehensive as the instances can be extracted only if the words of the instances in the document match a certain pattern.

2.4 Property Precedence

Unlike data integration models and current schema matching approaches, Property Precedence relaxes the *assumption of inherent classification*, the assumption that data is organized into a class-based schema (Parsons and Wand 2000). This assumption is best exemplified in the relational data model. All data in a relational database is organized into fixed tables (reflecting classes) and managed through operations on these tables. The class-based data assumption is reasonable when data integration is limited to the data with a well-defined structure. However, as data are frequently unstructured in current data integration contexts, the assumption of inherent classification typically does not hold and approaches based on it may not function well. For example, when matching two XML documents, suppose that one XML document has a text node with value "*Jeffrey Ullman* wrote the book *Principles of database and knowledge-base systems*" and the other document has a text node with value "*Computer science press* published *Principles of database and knowledge-base systems*". Under such a situation, current schema matching approaches treat these two text nodes as two data values of two attributes and therefore face a dilemma: matching these two nodes would not be a reasonable decision as "*Jeffrey Ullman*" does not equal to "*Computer science press*", however, not matching these two nodes would miss an important relation between these two documents as they both refer to the same book.

In contrast to class-based approaches, Property Precedence is based on the existence of instances and properties independent of any classification (Parsons and Wand 2003). Property Precedence regards properties as basic semantic constructs and makes it possible to handle data with different granularities and less structure. In the above situation, Property Precedence can be applied to treat the two text nodes as two instances and identifies the first instance with title and author properties, the second instance with title and publisher properties. Property Precedence then may match the first title property to the second title property.

The precedence relation defined in Property Precedence differs from the mapping relation in current schema matching approaches. The mapping relation only reflects the equivalent semantic relations between properties. The precedence relation is not limited to equivalence relations; it entails equivalence relations, containment relations and other semantic relations. For example, using property precedence, we might say *earning* precedes *depreciation and amortization* in the discussion of corporate income, even though *depreciation and amortization* is neither equivalent to *earning* nor part of *earning*.

The basic idea of Property Precedence is that two properties of different sources are distinct from each other, but may have the same meaning at a more general conceptual level (Parsons and Wand 2003). In the above example, *earning* and *depreciation and amortization* are different, but in the discussion of corporate income, *depreciation and amortization* are reflected in *earning*.

Property Precedence can be understood in terms of several key definitions. First, one property, P_1, is said to *precede* another, P_2, if and only if the set of instances possessing P_2 is subsumed by the set of instances possessing P_1. Second, a *manifestation* of a property P_1 is a set of properties, $P_2, \ldots, P_n$, such that the set of instances possessing any of $P_2, \ldots, P_n$ is a subset of the set of instances possessing P_1. *Full manifestation* means that the union of the sets of instances possessing $P_2, \ldots, P_n$ equals the set of instances possessing P_1. With these definitions, two results can be derived: (1) Given two properties G_1 and G_2 which precede a set of properties S_1 and S_2, respectively, G_1 precedes G_2 if S_2 is a full manifestation of G_2 and S_1 precedes S_2; (2) For every property in S_2, there exists at least one property in S_1 preceding it if S_1 is a full manifestation of G_1 and G_1 precedes G_2. The first result implies a preceding relation between two general properties when the properties that the first general property precedes in turn precede the properties that the second general property precedes. The second result preserves the consistency of the precedence schema by suggesting that two sets of properties that two general properties precede, respectively, have precedence relations when the two general properties do.

2.5 Summary

As GAV and LAV approaches both are based on datalog, goals and subgoals in datalog reflect classes, which indicate that GAV and LAV hold the assumption that data are class-based. Such an assumption may limit their application on unstructured data. Current schema matching approaches match elements only when two elements are assumed equivalent in semantics and fail to explore richer semantic relations. This limits their capability since semantic interoperability is needed not only for semantically equivalent elements, but also for other forms of semantic relations between elements such as containment. The approaches in information retrieval are capable of querying related unstructured documents from multiple sources. However, these approaches treat queries and documents as a unit and do not use the fact that the content of a query or a document can be decomposed into smaller structures. This limits their ability to support expressive querying.

The features that Property Precedence possesses are more suitable for integrating unstructured data. Property Precedence relaxes the assumption of inherent classification. Using properties as the basic construct, Property Precedence is capable of handling data in different granularities and with less structure. It also can support expressive queries. As precedence relations entail richer semantic relations that are not limited to equivalence and containment, it is reasonable to hypothesize that Property Precedence can provide better semantic interoperability. Compared with current schema matching approaches that apply machine learning techniques to determine whether two elements can be matched, the Property Precedence approach only needs to determine the existence of a property in an instance. As this is an easier task than matching, the Property Precedence approach is more likely to achieve better performance. In the following chapters, we will examine how to apply Property Precedence to integrate unstructured data.

3. Extracting Properties from Unstructured Data

Data in data sources represent instances (or things) and reflect properties of instances. Data usually can be grouped into one of three categories: structured, semi-structured and unstructured in data integration. Since identifying properties and instances is the first step for Property Precedence, it is necessary to identify properties of instances from three types of data. Usually identifying properties from structured data is easy. For example, a student record in a university database represents a student. The record is structured data and properties of the student are represented by the fields of the record. For semi-structured data, the structure information helps identify properties. For example, the listing page in eBay represents an item for sale and tags in the pages suggest properties of an item. The unstructured data we are facing are representations embedded in text (e.g., news stories). Unlike the previous two types of data, unstructured data provides no extra information to help identify properties. Consequently the first step is to identify properties in unstructured data.

To apply Property Precedence to the unstructured data, we assume each document of unstructured data as an instance. We begin by considering words in the text as properties of the instance, but single-term words sometimes do not possess enough semantics to decide the content, and they are ambiguous. Compared with single-term words, multiple-term words (phrases) are less ambiguous, more informative and more amenable for semantic relation discovery (Soderland and Mandhani 2007 , Manning and Schütze 1999). Hence our approach intends to identify multiple-term words from the text. Considering that manual identification is too expensive and impractical, we adopt an automatic phrase identification method.

3.1 Introduction

In the computational linguistics literature, several methods have been proposed to identify phrases from input sequences (Samuelsson and Voutilainen 1997). One straightforward method is to match substrings of the input sequence in a dictionary and find the longest matching string which segments the sequence in a way such that the number of segments is minimized. Another popular method that identifies noun phrases is to assign a part-of-speech (POS) tag to every term in the input sequence, and then collect the sequence of terms whose POS tags sequence satisfies that of noun phrases. Besides these two classes of methods, an alternative method that avoids POS tagging uses delimiters such as stop words and verbs to identify phrases.

It is easy to observe that the first method heavily depends on the dictionary used and will fail to identify phrases if they are not in the dictionary. Furthermore, in many cases the longest match does not generate the best result. For example, considering the phrase

Information Processing and Management Science

This phrase can be identified as "Information Processing and Management" and "Science", or as "Information Processing", "and", and "Management Science". It is obvious that the second result is better than the first one but the longest match will match "information processing and management" and will generate the first result if "information processing and management" appears as a phrase in some dictionary.

The second method is based on POS tagged words, which implies a POS tagger plays an important role. One common implementation of the POS tagger is to use the hidden Markov model. The basic assumption of this model is that the POS of the current word is decided by the POS of the previous word. This model regards the POS of each word as a hidden state and each word as the observation in a hidden state. It estimates the probability of transiting from one POS to another POS and the probability observing a word in a POS. By applying Maximum Likelihood Estimation (MLE) on the model, a POS sequence with maximum probability can be identified and this POS sequence is considered as a tagging sequence for the input sequence (Sharman, Jelinek and Mercer 1990). For example:

I have a dog.

According to the model, the probability of tagging the input sequence with a POS sequence "pronoun verb article noun" is

$$
\begin{aligned}
&P(\text{"pronoun verb article noun"} \mid \text{"I have a dog"}) \\
&\qquad \propto P(pronoun) \times P(verb \mid pronoun) \times P(article \mid verb) \\
&\qquad \times P(noun \mid article) \times P(\text{"I"} \mid pronoun) \times P(\text{"have"} \mid verb) \\
&\qquad \times P(\text{"a"} \mid article) \times P(\text{"dog"} \mid noun)
\end{aligned}
$$

where P(pronoun) is the probability that a sentence starts with a pronoun, P("I" | pronoun) is the probability of observing "I" in a word when the word is a pronoun, and P(verb | pronoun) is the probability of transiting from pronoun to verb. The probabilities of other POS sequences for the sentence can be therefore calculated and the POS sequence with the largest probability is considered to be the tagging sequence for the sentence.

After tagging the input sequence, further effort is needed to build a noun phrase identification model which can distinguish the POS sequences of noun phrases from others. In general this approach is plausible but the POS model and the phrase identification model cannot be 100% correct since the model is probabilistic and can introduce errors. A POS tagger with 97% accuracy is very impressive but the chance of getting all tags right in a 15-word sentence is only 63% (0.97^15). The phrase identification model would further increase the chance of error. In addition, probability estimation is not an easy task. Estimating of the probability of transiting from one POS to another POS, such as P(verb | pronoun), requires a large amount of tagged data. So does the estimation of the probability of observing a word in a POS, such as P("I" | pronoun). The estimation may not be accurate and can bring in errors. Furthermore, Feng and Croft (2001) discussed an example in which two sentences have the exactly same POS sequences, but one cannot apply the same noun phrase identification model.

The third method uses stop-words and verbs as delimiters to identify phrases (Bourigault 1992). However stop-words and verbs sometimes do not provide enough information for phrase identification. For example:

Ernst and Young is one of the big four auditors.

Since "and" is a stop-word, this method cannot determine that "Ernst and Young" is a phrase, which results in missing the most important information in this sentence.

To summarize the above discussion, the first and the third method are not robust enough to handle complicated situations in human language. Though the second method is more reliable, the POS model is not error-free and can produce incorrect input to the phrases identification model. The phrases identification model may also produce incorrect results. These uncertainties together make the method prone to errors. Besides, POS tagging sometimes does not provide enough distinction as some literature suggests (Feng and Croft 2001). More important, all three methods fail to bring enough consideration of the relations between words, such as the repetition of word sequences, for example "Ernst and Young" may have occurred several times in business news stories, which may indicate that "Ernst and Young" is a phrase.

The rest of the chapter is organized as follows: Section 3.2 redefines the problem. Section 3.3 introduces a probability model to solve the problem. Section 3.4 discusses the algorithm to solve the problem in the implementation. We conclude in Section 3.5.

3.2 Redefining the Problem

An approach that fits our needs should be simple and robust. By simple we favour a method that does not need to deal with POS tagging. As POS tagging can produce incorrect results and the phrase identification process based on POS tagging would enlarge the error, the performance of the approach can be significantly affected. Most importantly, a method that circumvents POS tagging will free us from needing to acquire a large amount of human tagged data for probability estimation.

By robust we prefer a model with less significant assumptions. We wish to avoid methods using certain words as delimiters since delimiter words cannot adjust to different situations. We do not heavily rely on dictionaries as they may not cover all phrases. Also, as in the "Information Processing and Management Science" example discussed above, dictionaries cannot provide enough information to determine which phrase is more suitable for an input sequence taking into account the context where a phrase occurs.

We first rewrite the phrase identification problem to the following problem: find a partition for a sequence of words such that every segment in this partition is meaningful and understandable to humans. We say that every segment of such a partition forms a phrase. In this way, phrase identification is reduced to a sentence segmentation problem, where a sentence is a sequence of words between punctuations.

3.3 Probability Model

To solve the sentence segmentation problem, we adapt a probability model similar to n-gram model (Manning and Schütze 1999). The n-gram model is used to predict a new word for a word sequence. The n-gram model for a sentence is given as follows:

$$P(ws) = P(ws[1]) \times P(ws[2]|ws[1]) \times P(ws[3]|ws[1 \dots 2]) \times \dots \\ \times P(ws[m]|ws[1 \dots m-1])$$

where *ws* is a word sequence, *ws[i]* is the word at position i of word sequence *ws*, *m* is the number of words in word sequence *ws* and word sequence *ws* can also be expressed as *ws[1]...ws[m]*, *ws[1...m-1]* is the substring of word sequence *ws* that starts at 1 and ends at m-1, *P(ws)* is the probability for the word sequence, and

$P(ws[m]|ws[1...m-1])$ is the probability of the word $ws[m]$ given that the word sequence that precede it is $ws[1...m-1]$. With this model, predicting a new word for a m-word sequence ws is finding a new word w with the maximum probability $P(w|ws[1...m])$ such that the probability of new word sequence ws', $ws[1]...ws[m]w$, is the maximal. The new maximum $P(ws')$ indicates that using word w, the new m+1 word sequence ws' is more likely to exist in human language than using any other word. As estimating $P(w|ws[1...m])$ is not practical, the n-gram model estimates $P(w|ws[m-n+1...m])$ by considering the word at position m+1 is only related to the n words before it which is $ws[m-n+1...m]$. A simplified n-gram model is the unigram model, which assumes words are independent. The unigram model greatly reduces the complexity of computation.

To adapt a similar probability model, we first introduce the assumption that phrases that are syntactically correct and semantically meaningful are repeatedly used in human language. To simplify the computation, we assume that the occurrence of a phrase in a sentence is independent from the occurrence of other phrases, which is similar to the assumption in unigram model. This model has less onerous assumptions compared with the assumptions held by the first and third methods mentioned above. The first method assumes that longer phrases are better than shorter phrases and the second method assumes that certain words separate phrases. Both assumptions do not take other phrases in the sentence into consideration, which implies independence of phrases. Compared with the method using POS tagging, this model allow us to avoid acquiring a large amount of POS tagged data.

We construct a model as follows. For a given sentence, we can obtain a segmentation that divides the sentence and this segmentation can be considered as a model to explain the chance that the sentence exists in human language. If the probability of existence for every segment is known, we can calculate the probability of existence for their combination. By assuming the independence of phrases, the probability of combination is the product of the probability of every segment. As the combination of these segments is the segmentation that is a model to explain the chance that the sentence exists in human language, the probability of the combination can be considered as an estimate of the degree of belief that this sentence exists. Because the sentence does exist, the degree of belief is supposed to be high and the estimation from a good model should be high as well. Next, we will argue that good segmentations are good models because they are more likely to have higher estimation than bad segmentations.

A good segmentation means that most segments are syntactically correct and contain semantically meaningful phrases, and these segments are assumed to have higher probability of existence than segments which are not syntactically correct or semantically meaningful. Thus the probability of their combination is likely to be higher. For a bad segmentation, some segments would be syntactically wrong or semantically meaningless. Their probabilities are assumed to be lower and so is their combination. When a good segmentation is compared to a bad segmentation, the good one is more likely to render a higher probability of occurring than the bad one. For example:

O1: The venture will be called BP/Standard Financial Trading and will be operated by

Standard Oil under the oversight of a joint management committee.

S1: The venture | will be called | BP/Standard Financial Trading | and | will be operated by | Standard Oil | under the oversight of | a joint management committee.

S2: The venture | will be called | BP/Standard Financial | Trading and | will be operated by | Standard Oil | under the oversight of | a joint management committee.

S1 and S2 are two segmentations for the sentence O1 and both have 8 segments, S1(i) and S2(i) where i=1, …, 8. S1 and S2 are almost the same - the only difference between them is in S1(3) and S2(3), and S1(4) and S2(4). S1(4) "and" has much higher probability of existence than S2(4) "Trading and" while S1(3) "BP/Standard Financial Trading" has almost the same probability as S2(3) "BP/Standard Financial". It is reasonable to suggest that S1 has higher estimation than S2 does, which indicates S1 is a better segmentation than S2.

Formally this model can be described as follows. For any given sentence O, there exists a collection of sets, $\mathrm{C} = \{\mathrm{Si}\}$, where every set Si in this collection is a segmentation of O and defined by $\mathrm{Si} = \left\{\mathrm{Si(j)}: \bigcup_{\mathrm{j}=1}^{|\mathrm{Si}|} \mathrm{Si(j)} = O, \mathrm{j} \in \{1 \ldots |\mathrm{Si}|\} \text{ and } \mathrm{Si(j)} \cap \mathrm{Si(k)} = \emptyset, \mathrm{j} \neq \mathrm{k}\right\}$ where Si(j) is a segment in a segmentation. As we discussed above, by the assumption of independence, the probability of the segmentation is the product of the probability of every segment in the segmentation. The probability of segmentation Si is denoted by P(Si) and defined by $\mathrm{P(Si)} = \prod_{\mathrm{j}=1}^{|\mathrm{Si}|} \mathrm{P(Si(j))}$ where P(Si(j)) is the probability of the segment in Si. As P(Si) estimates the probability of existence for sentence O, we are looking for an Si that maximizes the estimation, i.e. $\mathrm{P(Si)} = \mathrm{Max(P(Sj)}: \mathrm{Sj} \in \mathrm{C})$.

3.4 Implementation

The implementation of the model first involves obtaining the probability of every possible segment. As it is impossible to know the exact probability for a segment, the system used Reuters-21578 data set (Reuters-21578, Distribution 1.0 n.d.) as a corpus to estimate the probability because the data set is large (it contains 158224 sentences). Since the probability of a segment describes the chance of seeing the segment in human language, to estimate the probability of a segment we count the number of the sentences containing the segment and divide this number by the total number of sentences in the corpus:

$$\hat{P}(Si(j)) = \frac{\#(Si(j))}{\#(*)}$$

where $\hat{P}(Si(j))$ is the estimated probability for segment $Si(j)$, $\#(*)$ is the total number of the sentences in the corpus and $\#(Si(j))$ is the number of sentences containing $Si(j)$.

Different forms of a word might harm the accuracy of the estimation. For example, "make use of" "made use of" and "making use of" are the same phrase but they are different when counting the occurrence in the corpus. Therefore, the estimate of the probability of "make use of" is less than it should be. To avoid this problem, we used WordNet to convert the different forms of a word to its original form. In the above example, "make", "made" and "making" are replaced by "make".

Finding a segmentation that has the maximum estimation for a sentence incurs the most computation: an n-word sentence has 2^{n-1} segmentations. This is because for an n-word sentence, there are n-1 whitespaces that can be replaced by the segment bar. First, inserting into the sentence zero segment bars, there is one segmentation, the sentence itself. When inserting one segment bar, there are n-1 positions to fit in and so there are $\binom{n-1}{1}$ segmentations. When inserting m segment bars, there are n-1 positions to fit in m bars and this can be done in $\binom{n-1}{m}$ ways, so there are $\binom{n-1}{m}$ segmentations. Thus, the total number of all possible segmentation is $\sum_{m=0}^{n-1}\binom{n-1}{m} = 2^{n-1}$.

If the algorithm needs to do all the comparisons, finding the optimal result is impractical. We observe that finding a segmentation that maximizes the probability in an n-word sentence can be reduced to the problem of finding the shortest path in a graph with n(n+1)/2+1 vertices. To show the reduction, we first construct a graph as following:

Let every possible segment be denoted by a vertex in a graph along with an extra vertex called end state. If segment A and segment B are adjacent in any segmentations and A precedes B, there exists a path from A to B with length $-\ln \widehat{P}(A)$. If a segment B appears as the last segment in any segmentation, there exists a path from B to the vertex end state with length $-\ln \widehat{P}(B)$. If a segment A appears as the first segment in any segmentation, we call the segment start state.

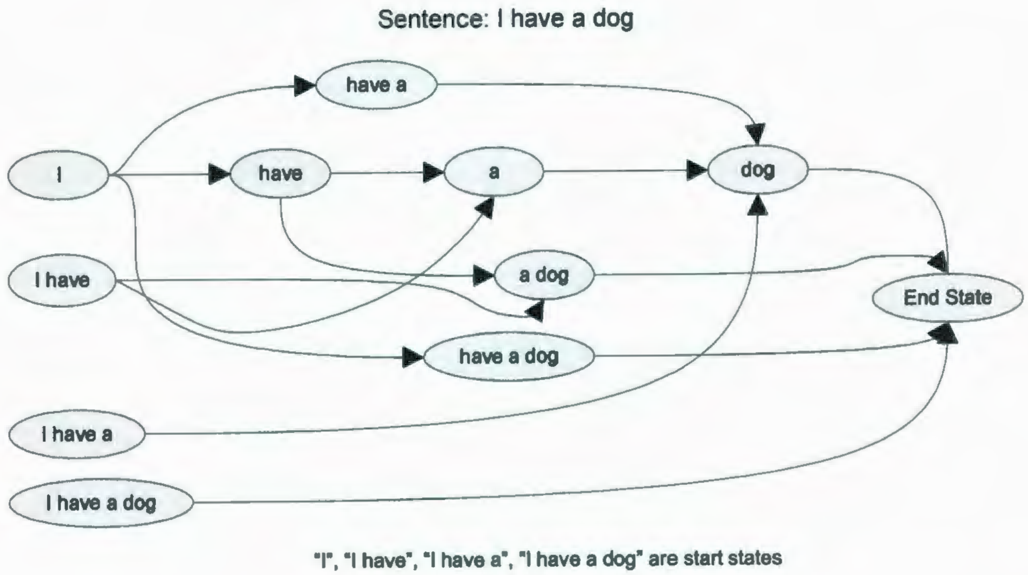
Next we show that the shortest path from start state to end state in this graph corresponds to the segmentation that maximizes the probability. If there exists a path from start state to end state, say A1(start state), A2, …, An(end state), the definition of the graph tells A1 is the first segment for a segmentation, A2 is the adjacent segment to

A1 and A3 is the adjacent segment to A2, and so on. The concatenation of A1, …, An-1 reconstructs the original sentence. If A1, …, An is the shortest path, then $\sum_{i=1}^{n} -\ln \widehat{P}(Ai)$ is the smallest. By the following calculation,

$$e^{\sum_{i=1}^{n} -ln \widehat{P}(Ai)} = \prod_{i=1}^{n} e^{-ln \widehat{P}(Ai)} = \prod_{i=1}^{n} \frac{1}{\widehat{P}(Ai)} = \frac{1}{\prod_{i=1}^{n} \widehat{P}(Ai)}$$

A1, …, An minimizes $1/\prod_{i=1}^{n} \widehat{P}(Ai)$ and thus maximizes $\prod_{i=1}^{n} \widehat{P}(Ai)$. Therefore, A1, …, An is a segmentation that has the maximum probability among other segmentations starting with A1.

Finally we show the number of vertices and edges in the graph and the number of start states. The number of vertices is the number of all possible segments plus end state. For an n-word sentence, the number of all possible m-word segments is n-m. As m ranges from 1 to n, the number of all possible segments is n(n+1)/2. Thus, the total number of vertices is n(n+1)/2+1. The number of edges is the number of possible links. We first consider the segments that start at the first word of the sentence. If the segment has length 1, then it can link to n-1 possible segments and if the segment has length h, it can link to n-h possible segments. The total number of edges for the segments that start at the first word is n(n-1)/2. Next we consider the segments that start at the m word of the sentence where m is between 1 and n. If the segment has length h, it can link to n-m+1-h possible segments. The total number of edges for the segments that start at the m word is (n-m+1)(n-m)/2. Therefore, the total number of edges in the graph is $\sum_{m=1}^{n} \frac{(n-m+1)(n-m)}{2} + n$ where the "+n" indicates the number of edges connecting to the end state. Fig 3.1 is the graph for the sentence "I have a dog".


Sentence: I have a dog

"I", "I have", "I have a", "I have a dog" are start states

Fig 3.1 Segments graph for sentence "I have a dog"

By applying the Viterbi like algorithm using dynamic programming (Viterbi 1967), we can efficiently solve the problem in O(n^2) time. The algorithm is given in Fig 3.2.

```
ws = input sequence // the input sequence with n words
//an array of size n+1 storing the maximal probability of the sub-sequence up to that position
max_probability = {0, ..., 0};
max_probablity[0]=1;
segmentations = {};
for i=1 to n{
  for j=0 to i-1{
    Estimate the probability P(ws[j...i]) //compute the probability of the segment that start at j and ends at i in ws
    if P(ws[j...i])*max_probability[j]>=max_probability[i] then {
      max_probability[i] = P(ws[j...i])*max_probability[j];
      segmentations[i] = j;
    }
  }
}
//output segments
i = n;
while(i>0){
  output ws[segmentation[i]...i];
  i = segmentation[i];
}
```

Fig 3.2 A Viterbi like algorithm to compute the segmentation with maximal probability

Though the complexity is reduced, the corpus is large and the computation is still costly. In addition, estimating the probability for all possible segments takes much space and time. In practice, we only consider the 2-term phrases and 3-term phrases. This approach has been implemented and tested on the Reuters-21578 data set. It works effectively in identifying phrases such as entity name and terminologies. Some phrases are given in the following table.

Table 3.1 Excerpts of identified phrase properties

prudential bache	*corp fnb*	*Producer and consumer*	*remain above*
brazilian export	*case-by-case basis*	*for instance*	*the southern basin*
sao paulo state	*preliminary duty*	*above average*	*advisory committee*
pepsico inc	*in public hand*	*current fiscal year*	*implication of*
expect to decline	*contributor to*	*farm organization*	*lash out*
visible trade	*old rate maturity*	*live cattle future*	*behind schedule*
carryforward gain of	*national corp*	*work population*	*industrial equipment*
payment of capital	*export policy*	*state department spokesman*	*hutton lbo inc*
semi-official anatolian agency	*farm policy*	*gasoline stock*	*golden nugget*
gundy inc	*chairman paul Volcker*	*tax code*	*minister michel noir*
agreement to stabilise	*chase manhattan*	*turkish foreign*	*security repurchase agreement*
canadian wheat export	*significant factor*	*milbank and co*	*takeover off*

3.5 Conclusion

When handling unstructured data, no structure information is available to help identify properties from data. To enable Property Precedence to process unstructured data, we proposed a probabilistic model to identify properties. We first introduced two assumptions, and reasoned the model based on these assumptions and gave the formal definition. The Reuters-21578 corpus allowed us to estimate the probability and implement the model. Observing that using the brute force method to find the optimal solution would cause overhead in computation, we reduced the problem of finding the segmentation with the maximum probability to the problem of finding the shortest path in a graph. Finally we presented some results produced by the model.

In the next chapter, we develop a method to identify the existence of a property in an instance when the property is not explicitly stated in the data.

4. Identifying Implicit Properties from Unstructured Data

As unstructured data usually are intended for human consumption, the data may not explicitly reveal all properties that an instance possesses. Relying only on the properties extracted from the description to determine the existence of a property in an instance is not enough. Recovering properties that are not explicitly revealed by the description, or implicit properties, is critical for property precedence discovery and it requires a good understanding of implicit properties.

Understanding an implicit property nears developing a definition for the property such that any given instance can be tested to determine whether it possesses the property. However, research in knowledge representation claims that a surrogate (such as the definition of a property and the description of an instance) could never be a completely accurate representation of the thing (such as the property and the instance) (Davis, Shrobe and Szolovits 1993). Furthermore, real world data usually only present partial or even distorted reflections of corresponding things. Such inaccurate reflection means the process to understand properties needs to be noise resilient. We apply methods in machine learning to accomplish the task.

4.1 Introduction

One way to find a definition for a property is to look up an existing ontology. Ontologies such as WordNet (2006) or Cyc (2007) usually are able to give a well-formed definition for a property. However, sometimes such well-formed definitions cannot handle incomplete and noisy real world data. The following example demonstrates this idea.

Suppose there are three persons (or instances) called "Anne", "Bob", and "Charles" and the descriptions about them are:

"Anne is a student"

"Bob attends Memorial University of Newfoundland"

"Charles attends Database course, writes Database assignment and loves music"

To determine who possess the property "student", three persons are to be tested on the definition of "student" given by the ontology.

The definition in WordNet: student, pupil, educatee (a learner who is enrolled in an educational institution)

The definition in Cyc: An instance of type of person classified by activity. Each instance of student is a person who studies at some educational institution...

As the description of "Anne" contains "student", it is explicit that "Anne" possesses the property "student". The description of "Bob" satisfies definitions if we know that "Memorial University of Newfoundland" is an educational institute and that "attends" is synonymous with "studies" or "enrol" in the context of education, so, "Bob" possesses the property as well. However, the description of "Charles" does not pass either definition so "Charles" should not possess the property according to these sentences, although human inspection might indicate otherwise.

Furthermore, an ontology may not contain all properties/concepts we are looking for and the effectiveness of the process is determined by the selected ontology.

An alternative method is to use prior knowledge, which is somewhat similar to the process whereby humans learn from past experience. This method first summarizes a model from the prior knowledge about a property and then applies this model to test whether an instance possesses the property. Such a method can be considered as solving a classification problem: the summary part is to learn from the instances that are known to possess the property, and the test part is to classify whether a new instance possess the property.

This method depends on prior knowledge. It is possible to develop a model that summarizes the knowledge of a property while ignoring the noise if the prior knowledge is large enough. Furthermore, the decision process no longer relies on a small piece of definition. It tests an instance from every possible aspect so that an instance without complete information will be accepted as well as long as enough evidence suggests so. Though an ontology provides definitions and relations for a concept, it is still necessary to match a property of an instance to a concept of the ontology. Furthermore, ontologies such as WordNet and Cyc are not domain specific ontologies, and might miss important definitions and relations for domain specific concepts. Comparing what the ontology could offer now, this method is more resilient to incomplete and noisy real world data. We apply this method in the implementation, building models of properties that summarize knowledge of properties (which we called summary model) and using the models to decide whether an instance possesses a property.

The rest of the chapter is organized as follows: Section 4.2 introduces the environment to build summary models and test the models. Section 4.3 discusses the instance model. Section 4.4 introduces the summary model and explains how a summary model of a property is built. Section 4.5 discusses implementation issues in building a summary model. Section 4.6 presents and analyzes results. We conclude in the last section.

4.2 Environment Setting

The Reuters-21578 data set (Reuters-21578, Distribution 1.0 n.d.) has been widely used in research of information retrieval (Dumais, et al. 1998, Yang and Liu 1999, Chai, Chieu and Ng 2002, Georgakis, Kotropoulos and Pitas 2002, Georgakis, Kotropoulos and Xafopoulos, et al. 2004, Debole and Sebastiani 2005, Kim, Han, et al. 2006). The data set has 21578 labeled documents. Every document is about a news story and the labels of a document indicate the topics of the corresponding news story. Topics of news stories are words such as "earn", "corn", "crude" and "money-supply" and the data set does not give further definitions for these words. Considering every news story as an instance, the document of the news story is the data/description of the instance and the topics of the news story are the properties of the instance. Here we assume the words in the document are the properties of the document instance as the meaning of the document are reflected through these words. These word properties of the instance can be extracted from the data, the document describing the news story, by applying the method described in the previous chapter. To distinguish the properties in the topics from the properties in the data, we call the former "topic properties". We also call these properties implicit properties as they may not appear in the news story. In the data set, 2/3 of all documents have been flagged as “TRAIN”, which we call training instances. The other 1/3 newswire have been flagged as “TEST”, which we call testing instances. Unless explicitly indicated below, the topic properties are known to the training instances but are unknown to the testing instances.

The prior knowledge of each topic property is contained in training instances. We use the summary model to summarize the prior knowledge of a topic property and the detail of the summary model is presented in Section 4.4.

To demonstrate the effectiveness of the summary model, we test the summary model on the testing instances. We let the topic properties of testing instances be unknown to the summary model and then test every testing instance on the summary model to decide whether the instance possesses the topic property. Since the topic properties of testing instances are known to us, we can evaluate the decision made by the summary model. If the summary model can effectively recognize the topic properties in the testing instances, we can say the summary model is effective in identifying the implicit properties.

Additionally, we are more interested in the cases such as “Bob” and “Charles” in the above example where the instance does not explicitly possess a property and the description does not directly reveal the existence of the property. To simulate this kind of case, before an instance is to be tested on a summary model, the data of the instance is scanned to remove the words that match with the words of the topic property.

Also, as the training instances of some topics are not large enough for summary, the experiment is conducted on the topics with more than 100 samples. The number of such topics is 16. This setup allows the summary model to have enough positive samples to summarize the prior knowledge of the implicit properties. As these 16 topics account for 82.54% (7926/9603) of the training instances and 91.27% (3011/3299) of the testing instances, they well represent the whole data set.

4.3 Instance Model

The properties and data of instances are text information and they are not suitable for computation. Therefore, it is necessary to map them into a form suitable for computation. One common way of mapping is to use a vector space model that maps the text information into a vector of weights. Each weight of the weight vector measures the importance of a property for the instance. We use TFIDF (Term Frequency/ Inverse Document Frequency) function to calculate the weight of each property for an instance (Spärck Jones 1972), an approach that has been widely used in research of information retrieval e.g., (Yang and Chute 1994, Dumais, et al. 1998, Sebastiani 2002). The following formula is used to calculate TFIDF:

$$tfidf(property\, p, instance\, i) = \#(property\, p, instance\, i) \times log\left(\frac{\#(*)}{\#(property\, p)}\right)$$

where #(property p, instance i), also known as TF, is the number of occurrences of property p in the description of instance i, #(*) is the number of all the training instances, #(property p) is the number of the instances that possess property p, and $\log\left(\frac{\#(*)}{\#(\text{property p})}\right)$ is known as IDF.

TFIDF can effectively measure the importance of a property for an instance. It not only considers the term frequency (TF), the occurrence of a property in the data of an instance, but also takes the inverse document frequency (IDF), the occurrence of a property in the data of other instances, into account. If property p has been repeated mentioned in the description of instance i, p is likely to be important for i. TFIDF reflects this by letting the first term of the formula be large. However, if p also has very high occurrence in other instances, p is more likely to be a common property so p should be less important. TFIDF reflects this by letting the second term be small. If a property is important for the instance, TFIDF of the property in the instance will be large. Otherwise it will be small. Consider the following example:

Suppose there are three instances, say "student", "professor" and "staff". All of them have property "walk" but only "student" has property "take course". By the above formula, the TFIDF of property "walk" for "student" is 0 since the second term is 0 ($3 \times \log\left(\frac{3}{3}\right)$). The TFIDF of "take course" is 1.6. TFIDF measurement suggests that property "take course" is more important for "student" than "walk". This conclusion is consistent with intuition.

Unlike conventional text representation that relies only on the occurrence of single terms in the text (Dumais, et al. 1998), the instance model takes multi-term properties into consideration to avoid ambiguity.

Furthermore, we observe that the properties of an instance are not necessarily independent from each other. The sequence of presenting the properties in the description of the instance reflects the context information of properties and it matters. This observation is consistent with results presented in (Arazy and Woo 2007). The property sequences we discussed here are referred as cross sentence directional collocation. We use this information to capture the relations between sentences by counting the sequence of presenting properties in different sentences and reflecting it in the weight vector of the instance. As statistical significance is important in calculating TFIDF, if property sequences do not have enough statistical significance, their TFIDF weights are trivial and their contributions to deciding the existence of properties are trivial as well. This is not our intention. To ensure statistical significance, we only consider the sequence pair, a sequence with two elements. If property X and property Y appear in difference sentences and the sentence where property X appears is ahead of the sentence where property Y appears in the description of an instance, we denote sequence pair as (X, Y). (X, Y) does not equal to (Y, X). If property X and property Y appear in the same sentence, we do not count as our intention is to capture the relations between sentences. If another Y appears after the first Y, the number of occurrences of (X, Y) is increased by one, which means the importance of (X, Y) to the instance is increased. For example, a news story is about "money foreign exchange". The first sentence of the news story contains the word "foreign investor" and in later sentences "cents" has been mentioned 7 times. ("foreign investor", "cents") is a property sequence of this news story and the number of occurrences is 7. If considering the single property, both of them are not very close with "money foreign exchange": "foreign investor" is more related to "investment" and "cents" is more related to “money”. However, their combination is more likely to suggest “money foreign exchange”.

In this model, we introduce the multi-term properties to capture the context information within sentence and apply the property sequence to represent the context information across sentences. By using such a representation, we relax the “bag of words” (Blei, Ng and Jordan 2003) assumption.

The instance model contains properties and property sequences of an instance and we call properties and property sequences factors of an instance. Applying the instance model, we can map an instance into a weight vector using TFIDF function. The weight vector is normalized such that different instances are comparable. The normalization is given as following:

$$\vec{v}' = \frac{\vec{v}}{||\vec{v}||}$$

where $\vec{\mathrm{v}}$ is the original vector, $\vec{\mathrm{v}}'$ is vector after normalization and $||\vec{\mathrm{v}}||$ the norm of the $\vec{\mathrm{v}}$ calculated by $\sqrt{\mathrm{v}_1{}^2 + \cdots + \mathrm{v}_\mathrm{n}{}^2}$ where $\mathrm{v}_\mathrm{n}{}^2$ is the nth component of the vector $\vec{\mathrm{v}}$.

4.4 Summary Model

The instance model is an abstract view of an instance that describes how different factors imply an instance, since the weight in the instance model for an instance indicates the importance of a factor to the instance. However, it does not describe how these factors imply an implicit property. The summary model is intended to solve this problem. The summary model is a model that suggests the existence of a property in an instance and by using this model we can conclude whether an instance possesses the property. The summary model of a property contains two components: the first part is the factors that are highly correlated with the property, and the second part is a function that describes how these factors are organized to suggest the existence of the property. In other words, it is a function that takes these factors and an instance as input and outputs whether the instance possesses the property. To construct a summary model of a property, we first identify the factors that are highly correlated to the property, and then we examine the instances that possess the property to learn the function. When a new instance is presented to a summary model of a property, the summary model first identifies the factors in the instance that matches the factors of the summary model and then applies the function to determine whether the instance possess the property. In this section, we discuss how to construct these two components of the summary model in detail.

4.4.1 Related factors

As discussed above, TFIDF indicates the importance of a property to an instance. We can also use TFIDF to evaluate the importance of a factor to the property. If a factor has high occurrence in the instances that possess the property, the factor should be related to the property. If a factor also has high occurrence in the instances that do not possess the property, the factor should be less important to the property. The original TFIDF is changed as follows,

$$tfidf(factor\ f, property\ p) = \#(factor\ f, property\ p) \times log\left(\frac{\#(*)}{\#(factor\ f)}\right)$$

where #(factor f, property p) is the number of instances that possess both factor f and property p, and #(*) is the number of all the training instances that do not possess the property p plus 1, and #(factor f) is the number of instances that possess factor f but do not possess the property p plus 1.

The new TFIDF can be interpreted as follows: considering the instances that possess property p as an aggregate, the new TFIDF measures the importance of a factor to the aggregate. If the aggregate implies property p, a factor that is important to the aggregate is necessarily important to property p. We argue the aggregate does imply property p since (1) every instance in the aggregate possesses property p, and (2) the difference among the instances in the aggregate and the large number of instances in the aggregate minimize the chance that the aggregate implies any other thing that is unrelated with property p. The "plus 1" in the above function refers to the aggregate.

Here is an example to explain the idea. Suppose there are five instances and the first three of them possess the property "sports". In these three instances, the first two instances possess the property "hockey" and the third one possesses the property "hockey" and the property "injury". The last two instances do not possess the property "sports", the property "hockey" or the property "injury". It is easy to see the TFIDF of the property "injury" is larger than the property "hockey" for the third instance. The TFIDF of the property "injury" in the third instance is 2.32 ($1 \times \log\left(\frac{5}{1}\right)$) and the TFIDF of the property "hockey" is 0.74. The property "injury" is more important than the property "hockey" for the third instance. However, when considering the instances that possess the property "sports" as a aggregate, the TFIDF of the property "hockey" in the aggregate is 4.75 $(3 \times \log\left(\frac{3}{1}\right))$ and the TFIDF of the property "injury" in the aggregate is 1.58. The new TFIDF suggests the property "hockey" is more important to the property "sports" than the property "injury" and this conclusion agrees with intuition. Though the property "injury" is important to its own instance, it would not have enough occurrences in the aggregate since it is less related to the property "sports" and it becomes less important to the aggregate.

After determining which factor is more related to a property, we further need to determine how many related factors the summary model should use. The most popular way to determine such a parameter is to randomly select some instances from training instances, test the performance of the model and use the number that achieves the best performance (Sebastiani 2002). We choose 1/3 of the training instances as validation instances and build the summary model on the rest of the training instance with different numbers of related factors. By letting the topic properties of the validation instances be unknown to the summary model, we test the summary model using different numbers of related factors on the validation instances. In the experiment, the number of related factors ranges from 100 to 1000 in steps of 100. We measure the performance of the summary model with different number of related factors using the F-measure (discussed in the later section). The experiment indicates the summary model achieve the best performance when it uses the first 600 related factors. In the later experiment, we let the summary model use 600 most related factors.

4.4.2 Learning methods

The second step is to learn how related factors are organized in an instance to imply that the instance that possesses the property. This step involves learning from the training instances and can be considered as the training process in classification. We tested several training methods to evaluate the effectiveness of the selected factors and the training methods as well. These training methods have been well studied and applied in machine learning research. Also these methods are consistent with the design of the instance model and the related factors in the summary model.

Cosine similarity

By the instance model, instances are mapped to weight vectors. The similarity of any two instances can be compared by computing the angle between two weight vectors. The cosine of the angle can be calculated by the inner product of two vectors.

$$cos\theta = \frac{\vec{v}_1 \cdot \vec{v}_2}{\|\vec{v}_1\|\|\vec{v}_2\|} = \frac{(v_{11} \times v_{21} + \cdots + v_{1n} \times v_{2n})}{\|\vec{v}_1\|\|\vec{v}_2\|}$$

The bigger the $\cos\theta$ is, the smaller the θ is and the closer the $\vec{v}_1$ is to $\vec{v}_2$. The cosine method assumes that if an instance is close enough to another instance, then the first instance may possess some properties that the second instance possesses. If one considers all the instances possessing the property as an aggregate, it is possible to generate a weight vector for the aggregate by the instance model. The calculation of the weight vector of the aggregate is the same as the calculation of the new TFIDF in 4.4.1. In this way, we can measure how close an instance is to the aggregate. If they are close enough, the instance is likely to possess the property since the aggregate is the union of instances that possess the property.

Naïve Bayesian

Bayesian method is a widely used probabilistic classifier (Lewis 1992, Sebastiani 2002, Chai, Chieu and Ng 2002, Kim, Han, et al. 2006). It assumes that the belief of hypothesis changes as evidence accumulates. A hypothesis with high degree of belief should be accepted and that with low degree of belief should be rejected. In our problem, the hypothesis is whether an instance possesses a property and the evidence is the properties that an instance possesses and property sequences. This method does not require the vector form of the instance model. The following formula is used to compute the degree of belief.

$$P(property\ p |\ instance\ i) = \frac{P(property\ p) \times P(instance\ i\ |\ property\ p)}{P(instance\ i)}$$

P(property p | instance i) is the probability of instance i possessing the property p, i.e., the degree of belief that the instance i possesses the property p. P(property p) is the priori probability that a randomly selected instance possesses property p. P(instance i | property p) is the probability of seeing the factors of the instance i in the instances possessing property p. P(instance i) is the probability of seeing instance i if randomly selecting an instance.

To determine whether an instance possesses the property or not, P(property p | instance i) is compared with P(no property p | instance i) and the hypothesis with higher probability is accepted.

In the computation, P(property p) is estimated by the ratio of the instances possessing the property out of all the instances. As P(property p | instance i) and P(no property p | instance i) both have the same denominator and the comparison only depends on the numerator, the estimation of P(instance i) is not necessary. Estimating P(instance i | property p) poses the most difficulty. In Naïve Bayesian, which assumes that the probability of any two factors are independent, this probability is estimated by multiplying all the probabilities of seeing each factor of instance i in the instances possessing property p and the probability of seeing a factor of instance i in the instances possessing the property p is estimated by the ratio of the instances possessing the factor and the property p out of the instances possessing the property p. For example, an instance has two properties "take Database course" and "love music". In the instances that possess the property "student", 2% possess the factor "take Database course" and 60% possess the factor "love music". The estimation of P("take Database course"| "student") is 0.02 and the estimation of P("love music" | "student") is 0.6. The estimation of P(instance i | "student") = P("take Database course"| "student")×P("love music"| "student") = 0.012.

Linear regression

Regression estimates a function that approximates the sample data set (Yang and Chute 1994, Yang and Liu 1999). When a new input is given, the function will determine the class label of the new input. Hence the regression method assumes that a function can correctly recognize the new input if it can well approximate the sample data, that is, the new inputs follow the distribution of sample data. In our case, the input of the function is instances and the output of the function is whether the instance possesses property p, which is expressed by the following formula:

$$f\text{:}\{instance\ i\} \rightarrow \{property\ p,\ no\ property\ p\}$$

As an instance can be represented by a set of factors, the above formula is changed into:

$$f\text{:}\{factor\ 1, \cdots, factor\ n\} \rightarrow \{property\ p,\ no\ property\ p\}$$

For linear regression, the above formula can be further expanded into:

$$f\text{:}\{factor\ 1, \dots, factor\ n\} = w_1 \times g(factor\ 1) + \cdots + w_n \times g(factor\ n)$$

As the *g(factor n)* can be a linear function or a non-linear function, say gauss function, the corresponding *f* can be a linear or a non-linear function. Thus, the linear regression is possible to approximate a data set that is non-linearly distributed. Least-squares and gradient-descent methods both can be used to find the suitable weights for the function.

Least-squares method:

$$\overrightarrow{weight} = (X^T X)^{-1} X^T \vec{y}$$

where X is a matrix that describes the factors that each training instance possesses and $\vec{y}$ is a vector specifying whether each training instance possesses the property or not. In the implementation, each row vector of X stands for a model of an instance and the corresponding y is 1 if the instance possesses the property or 0 if not.

Gradient-descent method:

$$\overrightarrow{weight}(n+1) = \overrightarrow{weight}(n) - \frac{1}{2}\mu\nabla\left(f - \hat{f}(n)\right)^2$$

where $\overrightarrow{weight}(n)$ is $\overrightarrow{weight}$ at time n, μ is the learning rate, $\hat{f}(n)$ is the real output, f is the expected output and $\nabla(f - \hat{f}(n))^2$ is the change of error upon the change of weights. Initially $\overrightarrow{weight}(0)$ is set randomly. Every training instance is fed to the function f and the error is computed to update the weight. The process is repeated until certain conditions are satisfied, for example, the weights stop changing.

The gradient-descent method needs a large number of iterations to converge while least-squares method does not. However, the least-squares method consumes more memory than gradient-descent method especially when the input matrix is large. In the implementation, we use the least-squares method.

K-nearest neighbour

The k-nearest neighbour method assumes that the k nearest neighbours of an instance decide whether this instance possesses a property or not. If a certain portion of the k nearest neighbours possesses the property, this instance is likely to possess the property as well. Otherwise this instance is unlikely to possess the property. The literature reports that k between 30 and 45 yields the best result (Sebastiani 2002). Compared with other methods above, this method requires much more computation: deciding k nearest neighbour involves computing the distances from an instance to every training instance and there are more than 9000 training instances in the data set. This high complexity of computation makes this method very unsuitable for solving our problem.

4.5 Implementation Issues

Determine threshold

In the implementation, the cosine similarity method, linear regression method and k-nearest neighbour method incur a problem of determining a threshold such that a testing instance is deemed to possess the property if it surpasses the threshold. Unlike other approaches that use a validation set (Yang and Liu 1999), we determine the threshold by minimizing the entropy, which is faster than the approaches using the validation set.

Information entropy can be used to measure the diversity of a data set and is defined as follows:

$$H(X) = -\sum_{i=1\ or\ 0} p(X = i)\, log\, p(X = i)$$

where p(X=i) is the probability of picking up a piece of data from the data set labelled as i. Suppose data set A has 2 members, both are labelled as 1. By the formula, the entropy of data set A is 0 since logP(X=1) = 0 and p(X=0) = 0. Suppose data set B has 2 members too but one is labelled as 1 and one is labelled as 0. The entropy of data set B is 1 which is larger than 0. Thus B is more diverse than A, which agrees with the observation.

For the cosine similarity method, we compute the cosine value for every training instance and sort the training instances by their cosine values. We label the instances that possess the property as 1 and those that do not as 0. In the best situation, there is a position in the sorted instances list such that all instances above the position has the label 1 and all instances below the position has the label 0. This position separates the instances that possess the property from those do not and we call this position as partition position, the instances above the position as partition 1 and the instance below the position as partition 0. The corresponding cosine value of the partition position can be considered as the threshold. The entropy of partition 1 is 0 as all of the instances in this partition have label 1 and the entropy of partition 0 is 0. We define the combined entropy of partition 1 and partition 0 as follows:

$$H = \frac{\|X\|}{\|X\| + \|Y\|} H(X) + \frac{\|Y\|}{\|X\| + \|Y\|} H(Y)$$

where X is partition 1, Y is partition 0, ||X|| is the number of instances in partition 1 and H(X) is the entropy of partition 1. In this case the combined entropy is 0. As the combined entropy measures the diversity of two partitions, the 0 combined entropy is the smallest combined entropy as entropy is non-negative, which indicates partition 1 and partition 0 have no diversity.

The above discussion is about the best situation but such a situation scarcely exists. In most cases, the instances with label 1 mix with the instances with label 0 but we still hope to find a partition position such that most of the instances in partition 1 have label 1 and most of the instances in partition 0 most have label 0. This means partition 1 and partition 0 should have the least diversity which means their combined entropy should be minimized. We calculate the combined entropy at every position and choose the position with the smallest combined entropy as the partition position and use the corresponding cosine value as the threshold. This method can be similarly applied to the problem of determining thresholds for linear regression and k-nearest neighbour methods.

Pseudo inverse

Using the least-squares method for the linear regression problem involves computing the inverse of X^TX where X is a matrix that describes the property set each training instance possesses. For a matrix to have an inverse, the matrix must not be singular. However it is not possible to assure X^TX is not singular as X may have linear dependent columns. Furthermore round-off errors during the computation may also lead to the singularity problem. To solve this problem, we adopt the pseudo inverse method which utilizes the singular value decomposition.

Singular value decomposition can decompose any matrix into three matrixes:

$$M = U\Sigma V^T$$

where U contains an orthonormal basis of the column vector of M, V contains an orthonormal basis of the row vector of M and $\sum$ is a diagonal matrix where singular values lie at the diagonal.

By singular value decomposition, the inverse of M can be rewritten as:

$$M^{-1} = (U\Sigma V^T)^{-1} = (V^T)^{-1}\Sigma^{-1}U^{-1}$$

As U and V are orthonormal, their inverses equal to their transpose and the above formula can be further written as:

$$M^{-1} = (U\Sigma V^T)^{-1} = V\Sigma^{-1}U^T$$

As $\sum$ is a diagonal matrix, its inverse is also a diagonal matrix such that components in the diagonal are the reciprocal of the corresponding component in $\sum$. If M is a singular matrix, there will be 0 at the diagonal of $\sum$ and in this case the corresponding position of the inverse of $\sum$ is 0 as well. In such a way, any matrix can have a corresponding inverse.

4.6 Evaluation

We employ the four methods described above to learn the summary model from the training instances in Reuters-21578 data set and then test the summary model on the testing instances to evaluate the performance. For comparison, we test the summary model that is derived from the conventional representation, which only uses single terms. To simulate the situation that the data of an instance does not directly reveal the existence of the property in the instance, we intentionally remove the words that match the topic property from the description of the instance.

We use recall and precision to measure performance (Salton and Lesk 1965). Recall and precision are computed by the following formula:

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

where TP is the number of true positive, FN is the number of false negative, FP is the number of false positive.

True positive in our case occurs when the summary model decides that an instance possesses the property and the instance does possess the property. False negative occurs when the summary model decides that an instance does not possess the property but the instance does possess the property and false positive occurs when the summary model decides that an instance possesses the property, but it does not. The effectiveness of the summary model is measured by the F-measure (Sebastiani 2002).

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

We mentioned above that we are more interested in the case when an instance does not explicitly possess a property, and the description does not directly reveal the existence of the property, but the summary model still can successfully identify the instance. We tested two cases: one is "With Topic Property" where every testing instance is presented to the summary model exactly as the original; the other is "Without Topic Property" that the text content is scanned and words that match with the words of the topic property are intentionally removed. The results of the testing are presented in Table 4.1.

Table 4.1 Performance of summary model in identifying implicit properties

	conventional representation		instance model	
	With Topic property	Without Topic property	With Topic property	Without Topic property
Cosine similarity	**69.66%**	**64.37%**	68.80%	63.96%
Naïve Bayesian	64.58%	64.57%	**73.25%**	**73.02%**
Linear Regression	83.72%	79.84%	**84.81%**	**81.56%**
K-nearest neighbor	82.64%	78.85%	**82.98%**	**79.06%**

Though the performance of the summary model varies when using different methods, the results show the performance of the summary model is at least 60%. Using the linear regression method, the performance of the summary model is above 80%. The result demonstrates that the summary model can effectively identify properties from instances even when an instance does not explicitly possess the property. Furthermore, the instance model performs better than the conventional representation in most cases.

To further investigate the statistical significance of the model, we shuffle the training and testing instances in the data set by randomly selecting 2/3 of instances as training instances and letting the remaining 1/3 of the instances serve as testing instances. In such a way, we create 100 samples and each of them has different training and testing instances. For each sample, we use the training instances to build the summary model with conventional representation or instance model, and evaluate the performance (F-measure) of the summary model on testing instances. For each sample, we can calculate the performance difference between conventional representation and instance model and the mean of the performance difference in all samples. We apply the *t-test* to derive the interval of the performance difference at 95% and 99% confidence range respectively. The results are included in Table 4.2

Table 4.2 Performance difference between conventional representation and instance model

	Mean of F-measure difference ($F_{\text{instance model}} - F_{\text{conventional}}$)	
	With Topic property	Without Topic property
Cosine similarity	0.672%	0.715%
Naïve Bayesian	8.443%	8.435%
Linear Regression	1.483%	1.844%

95% confidence	Interval of performance difference ($F_{\text{instance model}} - F_{\text{conventional}}$)	
	With Topic property	Without Topic property

Cosine similarity	(0.553%, 0.791%)	(0.570%, 0.859%)
Naïve Bayesian	(8.372%, 8.515%)	(8.364%, 8.51%)
Linear Regression	(1.417%, 1.567%)	(1.738%, 1.944%)

99% confidence	Interval of performance difference ($F_{\text{instance model}} - F_{\text{conventional}}$)	
	With Topic property	Without Topic property
Cosine similarity	(0.502%, 0.841%)	(0.509%, 0.921%)
Naïve Bayesian	(8.342%, 8.545%)	(8.334%, 8.536%)
Linear Regression	(1.389%, 1.598%)	(1.694%, 1.988%)

The k-nearest neighbour method is not included in this statistical significant analysis because the high complexity. The positive mean of performance difference indicates the instance model is expected to perform better than conventional representation. The interval of performance difference describes an interval that the probability that performance difference would fall in for any given sample is 95% or 99%. The positive lower bound of the interval at 99% confidence range indicates that the probability that the instance model would perform better than the conventional representation is 99%. The results confirm that, by introducing multi-term properties and property sequence the instance model is superior to a conventional representation.

4.7 Conclusion

After introducing a way to identify properties from the description of an instance, we observe that the properties of an instance cannot be fully extracted from the description. To determine whether an instance possesses a property, we need to have a good understanding of the property. As we have discussed, directly applying the definition of a property from an existing ontology cannot achieve the desired result and we suggest using prior knowledge to test the existence of a property in a new instance.

In this chapter, we introduce a novel representation of instances, the instance model. The instance model includes single term properties, multi-term properties and property sequences. All these factors help it to better represent an instance. Based on the instance model, we introduced the summary model of a property, which contains the related factors to the property and a method that uses these factors to determine the existence of the property in an instance. In the experiment over the Retuers-21578 data set, the summary model effectively identified the existence of a property in a new instance after learning, even when the property did not appear in the description of the instance. This experiment validates our proposal for using prior knowledge to test the existence of a property in instances. Also the experiment suggests our novel representation, the instance model, is a better representation and can improve the performance of the summary model. The statistical hypothesis test further confirms our claim. In the next chapter, we propose a method to build a Property Precedence schema with the summary model and evaluate the effectiveness of the schema built with the summary model.

5. Building Property Precedence Schema

5.1 Introduction

Using the method in Chapter 3, we extract the properties of an instance from the description of the instance. Using the summary model discussed in the Chapter 4, we can determine whether an instance possesses an implicit property. Given this information, we can apply the Property Precedence definition (Parsons and Wand 2003) to build the precedence schema.

Other methods may also be able to determine whether an instance possesses a property. We introduce another two methods and let the building process using these two methods to build schemas as well. The schemas built with summary model and these two methods are compared to evaluate the effectiveness of using the summary model to build a property precedence schema.

The rest of the chapter is organized as follows: Section 5.2 introduces the environment setting. Section 5.3 discusses the process of building a property precedence schema. Section 5.4 introduces two alternative methods. Section 5.5 analyzes the schema built with different methods. We conclude in the last section.

5.2 Environment Setting

The setting in this chapter mainly follows the setting in the previous chapter. Every document in the Reuters-21578 data set (Reuters-21578, Distribution 1.0 n.d.) represents an instance of news story. The topics of the news story are properties of the instance, topic properties or implicit properties. Other properties are extracted from the document. As we mentioned before, the topic properties are known to the training instances and are unknown to the testing instances. As the topic properties may or may not be words appearing in the document, the building process needs to infer the topic properties of an instance from its description. To ensure the discovered precedence relations are meaningful, we skip the properties that are possessed by less than three instances. This number can be modified to determine how sensitive results are to the chosen threshold.

By letting the topic properties be known to the testing instances, we can build a property precedence schema which is the best schema we can get. We regard this schema as the "correct" schema and regard the property precedence relations in this schema as the correct precedence relations. We evaluate the effectiveness of the above methods by comparing the schemas built by them with the "correct" schema.

5.3 Building Process

We apply the Property Precedence definition to build the property precedence schema. The definition says that a property precedes another property if the instances that possess the first property include the instances that possess the second property. Intuitively, the process is to select two properties and compare the instances that possess them. The algorithm is given in Fig 5.1.

```
// for every property, find the instance set that possess it
1: for property p in all properties {
2:   initialize the instance set $I_p$ of property p = {};
3:     for instance i in all instances {
4:       if (i possesses property p)
5:         add i to $I_p$;
       }
   }
// identify property precedence between properties
6: initialize property precedence schema S = {};
7:   for property $p_1$ in all properties{
8:     for property $p_2$ in all properties{
9:       if (the instance set $I_{p1}$ of $p_1$  includes the instance set $I_{p2}$ of $p_2$)
10:        add $p_1$ precedes $p_2$ to S;
       }
     }
```

Fig 5.1 The algorithm for building a property precedence schema

In our experimental analysis, we notice that the above algorithm cannot efficiently process the data set because of the large number of properties (more than 100,000, as properties include single-term words and multi-term words). As the number of possible precedence relations is n×(n-1) where n is the number of properties, step 9 is to be repeated for n×(n-1) times to check the containment between instance sets of any two properties. We introduce a new algorithm which avoids the loop in all properties. This algorithm is based on the following observations,

- Though the number of all properties is large in relation to the number of instances, the number of properties in each news story is much smaller.
- If property p_1 precedes property p_2, at least one instance must possess both of them.
- Suppose property p_1 and property p_2 both exist in a subset of instances. If p_1 cannot precede p_2 in the subset, p_1 cannot precede p_2 in whole set of instances.

The first observation suggests that, if the new algorithm instead of looping in all properties only loops in the properties of one instance, the cost of the algorithm will be reduced. The second observation guarantees looping in properties of one instance still can identify all possible precedence relations, that is, the completeness of precedence relations is ensured. It also implies if two properties are not possessed by a common instance, they cannot precede each other, which saves computation. The third observation suggests we may know that one property cannot precede the other property in very early stage.

The new algorithm uses a hash table to store property sequence (p_1, p_2) that property p_1 cannot precede property $p_{2;}$ we call this the non-preceding table. When processing an instance, the algorithm assumes every property of the instance can precede each other unless the non-preceding table indicates otherwise. The following rule governs the correctness of the non-preceding table:

Suppose property p in instance i is being processed. If a property p_n previously assumed to precede p does not appear in instance i, the sequence (p_n, p) is added to the non-preceding table.

Details of the new algorithm are given in Fig 5.2:

```
initialize non-preceding table U = {};
initialize property precedence schema S = {};
for instance i in all instances{
  for property $p_1$ in all properties of i {
    for property $p_2$ in all properties of i{
      if (U does not contain ($p_1$, $p_2$))
        add $p_1$ precedes $p_2$ to S;
      if (U does not contain ($p_2$, $p_1$))
        add $p_2$ precedes $p_1$ to S;
      for property p in all property that precedes property $p_2${
        if ( properties of i does not contain p)
          add (p, $p_2$) to U;
      }
    }
  }
}
```

Fig 5.2 The new algorithm for building a property precedence schema

Two algorithms are tested on a Power Mac with 2.3GHz PowerPC G5 CPU and 1 gigabyte memory. The old algorithm took 2892.156 seconds to process the Reuters-21578 data set while the new algorithm needed only 29.711 seconds.

5.4 Alternative Methods

As instances usually are not presented as a set of well defined properties, the common way to determine whether an instance possesses a property is to analyze the description of the instance. One simple way to analyze the description is to search for the specific word of the property in the description. The instance is considered to possess the property only if the word appears in the description. We call this kind of analysis "surface analysis". The disadvantage of this method is apparent: one property can be expressed in different words due to the existence of synonyms and the surface analysis is not capable to deal with these cases.

Another way is to use a thesaurus or ontology to assist surface analysis. As the concepts in the ontology are well organized and related to each other, it is possible to solve the problem caused by synonyms. For example, WordNet (2006) introduces the synset, such that the words in the same synset are interchangeable. WordNet also provides the relations such as hyponym and meronym: hyponyms of a word X are the words *is a* X and meronyms of a word X are the words *is a part of* X. For example, “St. John’s” is a hyponym of “city” and “St. John’s” is a meronym of “Newfoundland”. When “St. John’s” is in the description of an instance, the instance is inferred to possess the property “city” as “St. John’s” is a city. Also the instance is inferred to possess the property “Newfoundland” as “St. John’s” is part of Newfoundland. The description analysis not only scans for the specific word of the property but also searches for the words that are in the same synset, the hyponyms, and the meronyms from WordNet. Though this method is more effective than surface analysis in handling the synonym case, it cannot handle the polysemy case that a word has different meanings in different context. Furthermore, WordNet assisted surface analysis will fail if a statement that does not contain any synonyms, hyponyms and meronyms of the word of the property still implies the existence of the property. The *“Charles attends Database course, writes Database assignment and loves music*” example in the previous chapter is such a case.

5.5 Results and Analysis

The building process employs these three methods to build the property precedence schemas, respectively. Excerpts of the property precedence schema built with the summary model method are given in Fig 5.3.

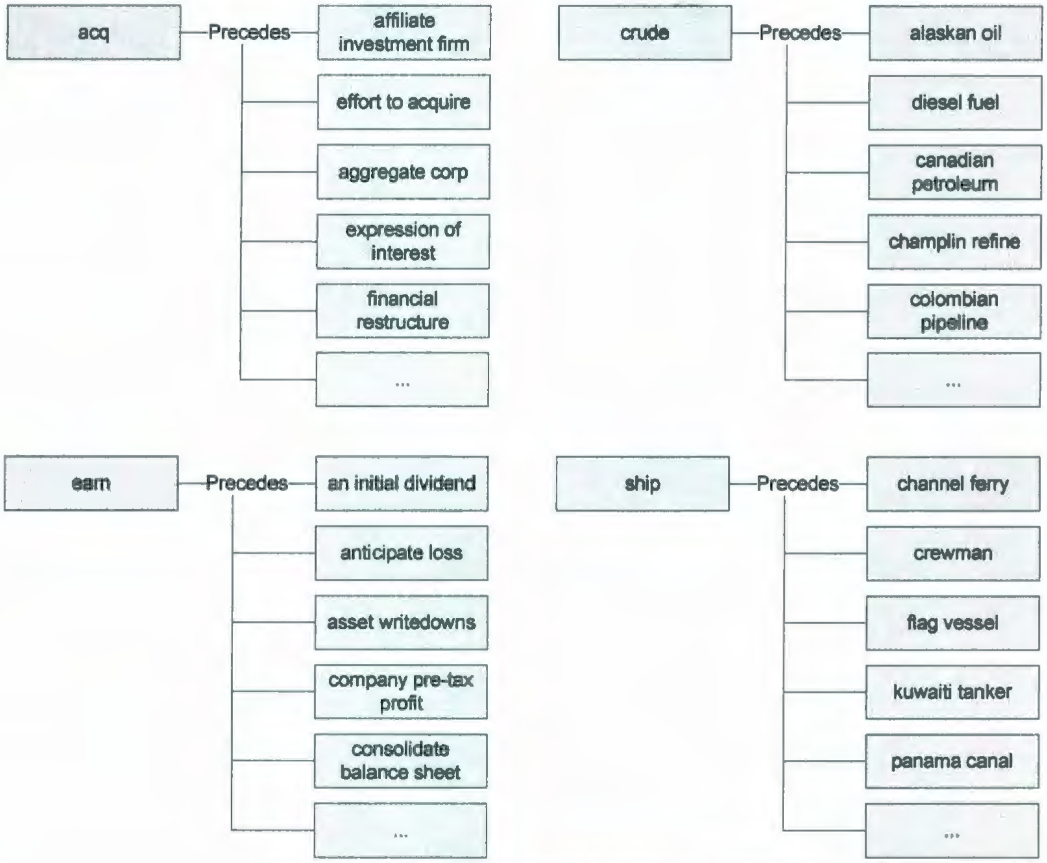


Fig 5.3 Excerpts of the property precedence schema

These schemas are compared with the correct schema to evaluate the effectiveness. The effectiveness is measured by the number of incorrect precedence relations. There are two kinds of incorrect precedence relations: (1) the schema does not have precedence relations that it should have, or false negatives (FN), and (2) the schema has precedence relations that it should not have, or false positives (FP). False negatives occur when the method fails to recognize that an instance possesses a property and incorrectly concludes that other properties in this instance cannot be preceded by this property. This results in correct precedence relations missing from the schema. False positive occur when the method incorrectly determines that an instance possesses a property and other properties in this instance may have chance to be preceded by this property. This can result in adding incorrect precedence relations to the schema. The result is presented in Table 5.1. We denote the correct precedence relations as true positive (TP). In the table, we calculate the precision, the recall, and F-measure and use them to measure the performance.

Table 5.1 Effectiveness of surface analysis, WordNet assisted surface analysis and summary model

Method	FN	FP	TP	Precision	Recall	F-measure
Surface Analysis	3036	**368**	42098	**0.9913**	0.9327	0.9611
WordNet assisted Surface Analysis	1482	1232	43652	0.9627	0.9674	0.9650
Summary Model	**1210**	593	**43924**	0.9867	**0.9732**	**0.9799**

As the F-measure measures the overall effectiveness of a method, the building process that employs summary model builds the best property precedence schema. Surface analysis cause a large number of false negatives, which is result of the ineffectiveness in determining the existence of properties in instances. As we expect, WordNet assisted surface analysis and summary model are more capable of determining the existence of properties and both successfully reduce the number of false negatives. Summary model generates the best result.

Though surface analysis results in the least false positive, this is because surface analysis only takes the exact word expressing a property into consideration, which helps to minimize the false positives. This is at the cost of a large number of false negatives and a small number of true positive.

WordNet assisted surface analysis has a very high number of false positives. This is because, although the synonyms, hyponyms and meronyms solve the problem that a property can be expressed in different words, they may have different meanings in different context. WordNet assisted surface analysis cannot differentiate these contexts, with the result that some instances are mistakenly determined to possess the properties. As a result, incorrect precedence relations are introduced.

Our approach using summary model is also not perfect, as it introduces some incorrect precedence relations. However, taking the improvement in false negative into consideration, it has the highest number of true positives and the best overall performance.

5.6 Conclusion

In this chapter, we discussed how to build a property precedence schema. The new algorithm for the building process greatly reduced the complexity of the computation. We further discussed two alternative methods that can determine the existence of properties in instances and compared the schemas built by different methods. The result indicated that summary model have the best overall performance. In the next chapter, we will discuss query processing based on a property precedence schema.

6. Querying on Property Precedence

6.1 Introduction

When we discuss querying across the Reuters-21578 data set (Reuters-21578, Distribution 1.0 n.d.), it is reasonable to consider every news story in the data set as a data source with a different schema as different news stories may have some properties in common but use very different words. For example, one document may use the word "ship" and the other document may use the word "ferry". Because of the semantic difference, data sources cannot totally understand queries and directly querying across all data sources may not produce the expected result. To resolve semantic differences, Property Precedence is introduced to provide a model to capture the semantic relationships between properties, which are able to bridge the semantic gap. When utilizing Property Precedence, queries are posed on a property precedence schema instead of directly on data sources. The property precedence schema translates the original queries into queries that data sources understand and then data sources take over to process the new queries.

Similar to other data integration models such as GLAV (Friedman, Levy and Millstein 1999), a property precedence schema also acts as a mediated schema to resolve the semantic difference between data sources. However, querying on a property precedence schema is different from querying on other data integration models. Most data integration models are class-based models that assume data in data sources have been organized by classes, that is, the data is either structured or semi-structured.

Property Precedence does not hold such an assumption; in contrast it is based on properties and instances. This difference offers Property Precedence extra flexibility in handling different types of data. It allows us to use Property Precedence to handle unstructured data in the Reuters-21578 data set. The difference also implies that the natural way to query a property precedence schema is to query by instances and properties.

The rest of the chapter is organized as follows: Section 6.2 defines property precedence query. Section 6.3 introduces the architecture and querying process on a property precedence schema. Section 6.4 presents some sample queries and analyzes the query result. Section 6.5 provides a conclusion.

6.2 Defining Property Precedence Query

As Property Precedence is a property-instance model, properties and instances form the basic constructs for querying the property precedence schema. The basic construct of querying is instances possessing a property, denoted as I(P = "p"), where I are instances that possess property P having name "p". For example, suppose the data set has three instances: one is "Anne is a student", one is "Bob is a teacher", and the other is "Charles is a high school teacher and attend Memorial University for higher degree". I(P="student") in this data set is "Anne" and "Charles" since "Anne" possesses property "student" and "Charles" possesses property "attending university" which is preceded by "student".

Since I(P= "p") stands for a set of instances, querying can be further expanded by introducing the following set operators to the basic construct. They are,

- NOT I(P= "p") is the instances that do not possess a property. In the previous example, NOT I(P= "student") only includes "Bob" since "Bob" does not possess property "student" and other properties preceded by property "student". Though "Charles" does not have property "student", he possesses property "attending university" which is preceded by property "student".
- I(P= "p_1") INTERSECT I(P= "p_2") is the instances that possess property "p_1" and property "p_2". In the example, I(P= "student") INTERSECT I(P= "teacher") only includes "Charles" since only "Charles" possesses both the properties since property "attending university" is preceded by "student".
- I(P= "p_1") UNION I(P= "p_2") is the instances that possess either one of property "p_1" and property "p_2". In the example, I(P= "student") UNION I(P= "teacher") includes "Anne" since "Anne" has property "student", includes "Bob" since "Bob" has property "teacher" and includes "Charles" as well since "Charles" has both properties. Also, union allows accessing combined information from different data sources. Suppose one data source has Anne's student number and academic record and another data source has Anne's user name and resume at a job seeking site, given "Anne" precedes "Anne's student number" and "Anne" precedes "Anne's username at a job seeking site", with Anne's consent, a recruiter can use a query (I(P= "Anne") INTERSECT I(P= "academic record"))

UNION (I(P= "Anne") INTERSECT I(P= "resume")) to review Anne's academic record and resume at the same time.

- I(P= "p_1") MINUS I(P= "p_2") is the instances that possess property "p_1" but do not possess property "p_2". In the example, I(P= "student") MINUS I(P= "teacher") only includes "Anne" since though "Charles" possesses property "student" he also possess property "teacher".

By combining the basic construct and these operators, we can form more expressive queries.

6.3 Query Processing Architecture

The architecture of query processing is given in Fig 6.1. Queries are posed on the global query processing unit and the global property precedence schema enriches the semantic meaning of queries by adding properties that are preceded by the properties that appear in the queries (Semantic Enrichment at Global Query Processing). The enriched queries are passed to the local query processing unit. Since the local property precedence schema is not necessary to understand every property in the global property precedence schema, the unknown properties are filtered to facilitate the next step processing (Property Filter at Local Query Processing). Also as the local schema may contain properties and precedence relations that are invisible to the global schema, it is necessary to employ the local schema to further enrich the semantics of queries (Semantic Enrichment at Local Query Processing). After this, the queries are passed to the data sources and the corresponding instances are selected and passed back to the global query processing unit to produce the results for queries.

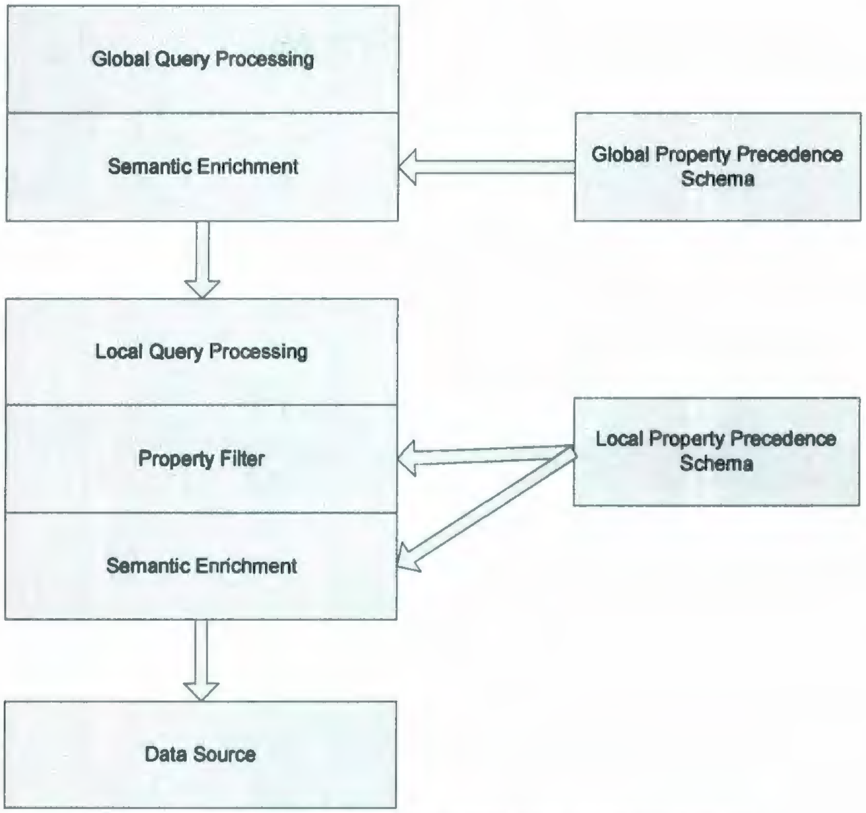


Fig 6.1 Architecture of property precedence query processing

Suppose the global property precedence schema has “student” precedes “attending school” and the local property precedence schema has “attending school” precedes “taking courses”. Two data sources are under the local property precedence schema. One is “Anne attends Memorial University of Newfoundland” and the other is “Bob takes Database course”. The query is given as I(P = “student”) posed on the global query processing unit. The global schema enriches the query as I(P= “student”) UNION I(P= “attending school”). When the enriched query is passed to the local query processing unit, the local schema filter the query I(P= “student”) UNION I(P= “attending school”) as I(P= “attending school”) since local schema does not understand property “student”. Next the local schema further enriches the query I(P= “attending school”) as I(P= “attending school”) UNION I(P = “taking course”) since property "attending school" precedes property "taking course". When the data sources receive the query, the first one returns "Anne" and the second one returns "Bob". The global query processing unit combines the returned results and produces the final results for the query. The detail algorithm of query processing is given in Fig 6.2.

```
// Global Processing Unit
global_processing(){
        read the input query Q;
        initialize property collection C = {};

        for every property p in Q {
                // return all properties that p precedes
                property set S = find_preceded_property(p, global schema GPS);
                add S to C;
        }

        pass C to local processing unit;
        receive results from local processing unit;
        produce final results according to Q;
}

// Local Processing Unit
local_processing(){
        read property collection C
        initialize property collection C' = {}

        for every property set S in C {
                remove properties in S that do not appear in local schema LPS;
                for every property p in S {
                        property set S' = find_preceded_property(p, local schema LPS);
                        add S' to C';
                }
        }
        pass C' to data sources;
        receive results from data sources;

}

// find all properties that a property precedes
find_preceded_property(property p, property precedence schem PS){
        initialize property set S = {};

        for(all properties p' that property p precedes in PS){
                if (p' is not p){
                        property set S' = find_preceded_property(p', PS)
                        add S' to S;
                }
                add p' to S;
        }

        return S;
}
```

Fig 6.2 The algorithm for query processing

6.4 Query Result Analysis

The query system is tested with some queries to evaluate the effectiveness. For example, the query I(P = "acq") produces some interesting results. One result is produced by the query because the system recognizes "acq" precedes "investor Asher Edelman", who is a former corporate raider (Asher Edelman - Wikipedia, the free encyclopedia 2008). A snippet of the result is given:

> *Burlington's stock rose sharply this morning on the report, which said Dominion Textile had joined with U.S.* ***investor Asher Edelman*** *to buy a stake in the company and to consider making a takeover offer.*

Though this instance can also be recognized by the system through other ways since the description contains word "acquisition", it demonstrates that the property precedence schema built by our approach can identify semantic relations between properties. Also as "investor Asher Edelman" not equal to or contained by "acq", this result indicates that property precedence is capable of capturing much richer semantic relations.

Another result is produced because the system recognizes "acq" precedes "definitive merger agreement". The full text of the news story is given:

> *Computer Associates International Inc and UCCEL Corp <UCE> said they have signed a* ***definitive merger agreement*** *under which Computer Associates will pay about 800 mln dlrs in stock for all outstanding UCCEL shares.*
>
> *The companies said under the terms of the agreement, all UCCEL shareholders will receive about 1.69 shares of Computer common stock for each of the approximately 17 mln UCCEL shares outstanding.*
>
> *According to the companies, this would amount to about 47.50 dlrs per UCCEL share, based on May 29 New York Stock Exchange closing prices.*
>
> *Closing of the transaction is anticipated in August, the companies said. The companies said the resulting company wil retain the name Computer Associates International Inc.*
>
> *Additionally, the companies said Charles Wang, currently Computer Associates chairman and chief executive, will continue as chairman of the new company.*

We notice the description of the newswire does not contain words such as "acquire" or "acquisition". Without the precedence relation that "acq" precedes "definitive merger agreement", the query would not be able to produce a result like this one. It shows the schema built by the system has a deep understanding of the instance and the properties. Furthermore, as "definitive", "merger" and "agreement" are not preceded by "acq", it also shows the process of identifying phrase properties effectively avoid semantic ambiguity.

To evaluate the overall effectiveness of the property precedence schema, we tested queries involving the topic properties over all testing instances (e.g., one of these queries is I(P = "acq")). In total, there are 90 queries querying 90 topic properties. We compare the case where property precedence schema is enabled with the case where the property precedence schema is disabled. The property precedence schema used here is the schema built with summary model. We let the querying processing unit process these queries in both situations and count the number of correct results. In the situation where property precedence schema is disabled, the number of correct results is 2639. In the situation where property precedence schema is enabled, the number of correct results is 3961. We also observe 107 incorrect query results when property precedence schema is enabled. This is because the property precedence schema built with summarized model may introduce incorrect precedence relations that lead to incorrect query results.

We examine the incorrect results and notice some incorrect results (listed in Appendix II) are produced by the precedence relation such as "money-fx" precedes “dollar/yen rate”, “money-supply” precedes “reserve projection”, “interest” precedes “easy monetary policy”, “wheat” precedes “agricultural produce”, “crude” precedes “mln barrel”, and “ship” precedes “freight cost”. By further investigating the corresponding news stories, incorrect results produced by the precedence relations such as “money-fx” precedes “dollar/yen rate”, “money-supply” precedes “reserve projection” and “interest” precedes “easy monetary policy” can be considered as correct results because these topics instead of being a major topic of the news stories are subtopics.

For precedence relations such as “wheat” precedes “agricultural produce”, it is obvious that two properties are related: the news story generated by “wheat” precedes “agricultural produce” actually has topic “grain”, the story generated by “crude” precedes “mln barrel” has topic “heating oil” and the story generated by “ship” precedes “freight cost” has topic “trade”. If the topic properties instead of being as specific as “wheat”, “crude”, and “ship”, are more general properties such as “farming”, “oil products” and “transport”, these precedence relations will be correct and produce the correct results. Considering the increased number of correct results and the precision of the query results (precision > 97.37% ($\frac{3961}{3961+107}$)), the incorrect results are acceptable. Property precedence can significantly increase the number of correct results by bridging the semantic difference between data sources and the number of incorrect results brought by the property precedence schema built with summary model is in a reasonable range. Some extra correct results retrieved by property precedence query are listed in Appendix I.

6.5 Conclusion

In this chapter, we discussed querying on a property precedence schema. First we defined property precedence query. Then we introduced the architecture and querying processing of property precedence query. At the end we analyzed the result that property precedence query produced and the result demonstrates the effectiveness of Property Precedence, the way we built the property precedence schema, and the way we identify properties.

7. Conclusion

Unlike data integration models and current schema matching approaches, Property Precedence relaxes the assumption of inherent classification, the assumption that data is organized into a class-based schema. It allows us to handle data in different granularities and with less structure. In this thesis, we presented a system that applies the concept of Property Precedence to integrate unstructured data sources. Specifically, we introduced an approach to identify multi-term phrase properties from unstructured data, which is capable of avoiding ambiguousness and are amenable for semantic discovery. Considering the unstructured data are intended for human consumption, a property may exist in an instance without appearing in the description of the instance. We introduced the summary model to determine the existence of these implicit properties in an instance. Our experiment results show the summary model is effective. We applied the definition of Property Precedence to build a property precedence schema. By introducing a new algorithm, we can build the property precedence schema efficiently. To evaluate the effectiveness of the property precedence schema, we compared the built schema with other schemas built by other approaches. The results indicate that our approach can build the most effective property precedence schema. Finally we defined and implemented the property precedence query. The experiment shows property precedence query can bridge the semantic difference between data sources. The evaluation of property precedence query shows property precedence query is capable of retrieving results that cannot be retrieved by other querying approaches.

Reference

Ampazis, N., and S. J. Perantonis. "LSISOM – A Latent Semantic Indexing Approach to Self-Organizing Maps of Document Collections." *Neural Process. Lett. 19, 2 (Apr. 2004)*, 2004: 157-173.

Arazy, O, and C. Woo. "Enhancing Information Retrieval through Statistical Natural Language Processing: A Study of Collocation Indexing." *MIS Quarterly Vol. 31 No. 3*, 2007: 525-546.

"Asher Edelman - Wikipedia, the free encyclopedia." *Wikipedia, the free encyclopedia.* 2008. http://en.wikipedia.org/wiki/Asher_Edelman.

Blei, D. M., A. Y. Ng, and M. I. Jordan. "Latent Dirichlet Allocation." *Journal of Machine Learning Research 3*, 2003: 993-1022.

Bourigault, D. "Surface grammatical analysis for the extraction of terminological noun phrases." *Proceedings of the 14th Conference on Computational Linguistics - Volume 3 (Nantes, France, August 23 - 28, 1992).* 1992. 977-981.

Chai, K. M., H. L. Chieu, and H. T Ng. "Bayesian online classifiers for text classification and filtering." *Proceedings of the 25th Annual international ACM SIGIR Conference on Research and Development in information Retrieval (Tampere, Finland, August 11 - 15, 2002).* 2002. 97-104.

Chuang, S., K. C. Chang, and C. Zhai. "Context-aware wrapping: synchronized data extraction." *Proceedings of the 33rd international Conference on Very Large Data Bases (Vienna, Austria, September 23 - 27, 2007).* 2007. 699-710.

Davis, R., H. Shrobe, and P. Szolovits. "What is a Knowledge Representation?" *AI Magazine, 14(1)*, 1993: 17-33.

Debole, F., and F Sebastiani. "An analysis of the relative hardness of Reuters-21578 subsets: Research Articles." *J. Am. Soc. Inf. Sci. Technol. 56, 6 (Apr. 2005)*, 2005: 584-596.

Deerwester, S., S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. "Indexing by latent semantic analysis." *Journal of the American Society for Information Science (1990)*, 1990.

Dempster, A. P., N. M. Laird, and D. B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm." *Journal of the Royal Statistical Society*, 1977: 1–38.

Do, H., and E. Rahm. "COMA: a system for flexible combination of schema matching approaches." *Proceedings of the 28th international Conference on Very Large Data Bases (Hong Kong, China, August 20 - 23, 2002).* 2002. 610-621.

Doan, A., P. Domingos, and A. Y. Halevy. "Reconciling schemas of disparate data sources: a machine-learning approach." *Proceedings of the 2001 ACM SIGMOD international Conference on Management of Data (Santa Barbara, California, United States, May 21 - 24, 2001).* 2001. 509-520.

Dumais, S., J. Platt, D. Heckerman, and M. Sahami. "Inductive learning algorithms and representations for text categorization." *Proceedings of the Seventh international Conference on information and Knowledge Management (Bethesda, Maryland, United States, November 02 - 07, 1998).* 1998. 148-155.

Etzioni, O., et al. "Unsupervised named-entity extraction from the Web: An experimental study." *Artificial Intelligence, 165(1)*, 2005: 91-134.

Feng, F., and W. B. Croft. "Probabilistic techniques for phrase extraction." *Inf. Process. Manage. 37, 2 (Mar. 2001)*, 2001: 199-220.

Friedman, M., A. Levy, and T. Millstein. "Navigational plans for data integration." *Proceedings of* *the Sixteenth National Conference on Artificial intelligence and the Eleventh innovative Applications of Artificial intelligence Conference innovative Applications of Artificial intelligence (Orlando, Florida, United States, July 18 - 22, 1.* 1999. 67-73.

Garcia-Molina, H., et al. "The TSIMMIS Approach to Mediation: Data Models and Languages." *J. Intell. Inf. Syst. 8, 2 (Mar. 1997)*, 1997: 117-132.

Georgakis, A., C. Kotropoulos, A. Xafopoulos, and I. Pitas. "Marginal median SOM for document organization and retrieval." *Neural Netw. 17, 3 (Apr. 2004).* 2004. 365-377.

Georgakis, A., C. Kotropoulos, and I Pitas. "A SOM Variant Based on the Wilcoxon Test for Document Organization and Retrieval." *Proceedings of the international Conference on Artificial Neural Networks (August 28 - 30, 2002).* 2002. 993-998.

Halevy, A. "Answering queries using views: A survey." *The VLDB Journal 10, 4 (Dec. 2001)*, 2001: 270-294.

Halevy, A., A. Rajaraman, and J. J. Ordille. "Data integration: the teenage years." *Proceedings of the 32nd international Conference on Very Large Data Bases (Seoul, Korea, September 12 - 15, 2006).* 2006. 9-16.

Hammer, J., H. García-Molina, K. Ireland, Y. Papakonstantinou, J. Ullman, and J. Widom. "Information translation, mediation, and mosaic-based browsing in the TSIMMIS system." *Proceedings of the 1995 ACM SIGMOD international Conference on Management of Data.* 1995.

Hofmann, T. "Probabilistic latent semantic indexing." *Proceedings of the 22nd Annual international ACM SIGIR Conference on Research and Development in information Retrieval (Berkeley, California, United States, August 15 - 19, 1999).* 1999. 50-57.

Kang, J., and J. F. Naughton. "On schema matching with opaque column names and data values." *Proceedings of the 2003 ACM SIGMOD international Conference on Management of Data (San Diego, California, June 09 - 12, 2003).* 2003. 205-216.

Kim, S., K. Han, H. Rim, and S. H Myaeng. "Some Effective Techniques for Naive Bayes Text Classification." *IEEE Transactions on Knowledge and Data Engineering 18, 11 (Nov. 2006)*, 2006: 1457-1466.

Kim, S., K. Han, H. Rim, and S. H. Myaeng. "Some Effective Techniques for Naive Bayes Text Classification." *IEEE Transactions on Knowledge and Data Engineering 18, 11 (Nov. 2006)*, 2006: 1457-1466.

Kurland, O. "The opposite of smoothing: a language model approach to ranking query-specific document clusters." *Proceedings of the 31st Annual international ACM SIGIR Conference on Research and Development in information Retrieval (Singapore, Singapore, July 20 - 24, 2008).* 2008. 171-178.

Levy, A. Y., A. O. Mendelzon, and Y. Sagiv. "Answering queries using views." *Proceedings of the Fourteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (San Jose, California, United States, May 22 - 25, 1995).* 1995. 95-104.

Levy, A. Y., A. Rajaraman, and J. J. Ordille. "Query-Answering Algorithms for Information Agents." *AAAI/IAAI, Vol. 1 1996*, 1996a: 40-47.

—. "Querying Heterogeneous Information Sources Using Source Descriptions." *Proceedings of the 22th international Conference on Very Large Data Bases (September 03 - 06, 1996).* 1996b. 251-262.

Lewis, D. D. "An evaluation of phrasal and clustered representations on a text categorization task." *Proceedings of the 15th Annual international ACM SIGIR Conference on Research and*

Development in information Retrieval (Copenhagen, Denmark, June 21 - 24, 1992). 1992. 37-50.

Li, T., C. Ding, Y. Zhang, and B. Shao. "Knowledge transformation from word space to document space." *Proceedings of the 31st Annual international ACM SIGIR Conference on Research and Development in information Retrieval (Singapore, Singapore, July 20 - 24, 2008).* 2008. 187-194.

Liu, Y., ·W. Li, Y. Lin, and L. Jing. "Spectral geometry for simultaneously clustering and ranking query search results." *Proceedings of the 31st Annual international ACM SIGIR Conference on Research and Development in information Retrieval (Singapore, Singapore, July 20 - 24, 2008).* 2008. 539-546.

Madhavan, J., P. A. Bernstein, A. Doan, and A. Halevy. "Corpus-Based Schema Matching." *Proceedings of the 21st international Conference on Data Engineering (April 05 - 08, 2005).* 2005. 57-68.

Manning, C. D., and H Schütze. *Foundations of Statistical Natural Language Processing.* MIT Press, 1999.

Manolescu, I., D. Florescu, and D Kossmann. "Answering XML Queries on Heterogeneous Data Sources." *Proceedings of the 27th international Conference on Very Large Data Bases (September 11 - 14, 2001).* 2001. 241-250.

Özsu, M. T., and P. Valduriez. *Principles of Distributed Database Systems (2nd Ed.).* Prentice-Hall, Inc., 1999.

Palopoli, L., D. Saccá, and D. Ursino. "Semi-automatic techniques for deriving interscheme properties from database schemes." *Data Knowl. Eng. 30(3) (Jul. 1999)*, 1999: 239-273.

Palopoli, L., G. Terracina, and D. Ursino. "Experiences using DIKE, a system for supporting cooperative information system and data warehouse design." *Inf. Syst. 28, 7 (Oct. 2003)*, 2003: 835-865.

Parsons, J., and Y Wand. "Emancipating instances from the tyranny of classes in information modeling." *ACM Trans. Database Syst. 25, 2 (Jun. 2000)*, 2000: 228-268.

Parsons, J., and Y. Wand. "Attribute-Based Semantic Reconciliation of Multiple Data Sources." *J. Data Semantics 1 (2003)*, 2003: 21-47.

Parsons, J., and Y. Wand. "Emancipating instances from the tyranny of classes in information modeling." *ACM Trans. Database Syst. 25, 2 (Jun. 2000)*, 2000: 228-268.

Pottinger, R., and A. Halevy. "MiniCon: A scalable algorithm for answering queries using views." *The VLDB Journal 10, 2-3 (Sep. 2001)*, 2001: 182-198.

Rahm, E., and P. Bernstein. "A survey of approaches to automatic schema matching." *The VLDB Journal 10, 4 (Dec. 2001)*, 2001: 334-350.

"Reuters-21578, Distribution 1.0." *Reuters-21578.* http://www.daviddlewis.com/resources/testcollections/reuters21578/.

Salton, G., and M. E Lesk. "The SMART automatic document retrieval systems—an illustration." *Commun. ACM 8, 6 (Jun. 1965)*, 1965: 391-39.

Samuelsson, C., and A Voutilainen. "Comparing a linguistic and a stochastic tagger." *Proceedings of the Eighth Conference on European Chapter of the Association For Computational Linguistics (Madrid, Spain, July 07 - 12, 1997).* 1997.

Sebastiani, F. "Machine learning in automated text categorization." *ACM Comput. Surv. 34, 1 (Mar. 2002)*, 2002: 1-47.

Sharman, R. A., F. Jelinek, and R Mercer. "Generating a grammar for statistical training." *Proceedings of the Workshop on Speech and Natural Language (Hidden Valley, Pennsylvania,* *June 24 - 27, 1990).* 1990. 267-274.

Soderland, S., and B. Mandhani. "Moving from Textual Relations to Ontologized Relations." *AAAI Spring Symposium on Machine Reading.* 2007 .

Spärck Jones, Karen. "A statistical interpretation of term specificity and its application in retrieval." *Journal of Documentation 28 (1)*, 1972: 11-21.

The Cyc Foundation, Computable Common Sense. 2007. http://www.cycfoundation.org/concepts.

Ullman, J. D. *Principles of Database and Knowledge-Base Systems, Vol. I.* Computer Science Press, Inc., 1988.

Viterbi, Andrew J. "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm." *IEEE Transactions on Information Theory, IT-13*, 1967: 260 -269.

"WordNet, a lexical database for the English language." *WordNet.* 2006. http://wordnet.princeton.edu/.

Yang, Y., and C. G Chute. "An example-based mapping method for text categorization and retrieval." *ACM Trans. Inf. Syst. 12, 3 (Jul. 1994)*, 1994: 252-277.

Yang, Y., and X Liu. "A re-examination of text categorization methods." *Proceedings of the 22nd Annual international ACM SIGIR Conference on Research and Development in information Retrieval (Berkeley, California, United States, August 15 - 19, 1999).* 1999.

Appendix I

Property precedence query retrieve the following news story because "acq" precedes "merger take place" and words such as "acquisition" and "acquire" do not appear in this news story.

<Hoechst Celanese Corp> said it sent propsective customers a confidential report describing its polyester textile fiber facilities in North Carolina and South Carolina.

The company did not disclose any prices.

The report describes the facilities in Darlington County, S.C., and Fayetteville, N.C., the company said. The report also decribes related manufacturing, marketing, administrative and technical resources that could be made avialable to a buyer.

Hoechst Celanese was formed Feb 27 by the merger of Celanese Corp and American Hoechst Corp. The ***merger took place*** *after an agreement was reached with the Federal Trade Commission that certain domestic polyester textile fiber assets of the combined companies would be divested, it said.*

Hoechst Celanese said it has the option of divesting either the South Carolina facilities of the former American Hoechst or a package of polyester textile fiber facilities of the former Celanese.

Property precedence query retrieve the following news story because "acq" precedes "takeover proposal" and words such as "acquisition" and "acquire" do not appear in this news story.

British press magnate Robert Maxwell said his British Printing and Communication Corp Plc would not renew its bid for Harcourt Brace Jovanovich Inc <HBJ> if the lawsuit filed against Harcourt in New York today fails.

Speaking at a press conference, Maxwell denied market rumors that British Printing had approached British institutions to arrange a rights issue with a view to relaunching its bid for the U.S. publishing concern.

"I don't believe in chasing mirages," maxwell said.

British Printing filed suit in U.S. District Court in Manhattan to block what Maxwell called a fraudulent recapitalization announced by Harcourt last week.

Harcourt, in response to a hostile two billion dlr ***takeover proposal*** *from Maxwell, planned a recapitalization that would pay shareholders 40 dlrs per share. Under the plan, it also said 40 pct of its shares will be controlled by its employees, management, and its financial adviser, First Boston Corp <FBC>.*

Property precedence query retrieve the following news story because "acq" precedes "propose takeover" and words such as "acquisition" and "acquire" do not appear in this news story.

Northair Mines Ltd said it would oppose Nor-Quest Resources Inc's earlier reported ***proposed takeover*** *bid "with every means at its disposal," saying "this attempt at a property grab is an insult to the intelligence of our shareholders."*

It said Nor-Quest's offer to swap one Nor-Quest share plus one dlr for two Northair shares would seriously dilute Northair's equity in its Willa mine in British Columbia.

"Our company is in sound financial position and production financing can be readily arranged when required. We're not looking for a partner and if we were, it certainly wouldn't be these guys," Northair said.

Property precedence query retrieve the following news story because "acq" precedes "negotiate transaction" and words such as "acquisition" and "acquire" do not appear in this news story.

Atlantis Group Inc said it bought 100,000 shares of Charter-Crellin Inc common stock, or 6.3 pct of the total outstanding, and may seek control in a ***negotiated transaction****.*

In a filing with the Securities and Exchange Commission, Atlantis said it has informally discussed a business combination with Charter-Crellin management.

But the company said it has not held negotiations with Charter-Crellin and does not intend to initiate further discussions.

Pending development of specific proposals, Atlantis said it will continue to purchase additional Charter-Crellin shares in private or open market transactions depending on a range of factors including the market price of the stock.

Atlantis said it bought its Charter-Crellin common stock in open market transactions between September 22 and October 7 at 14.91 dlrs to 15.62 dlrs a share, or for a total of about 1.51 mln dlrs.

Property precedence query retrieve the following news story because "acq" precedes "the merger plan" and words such as "acquisition" and "acquire" do not appear in this news story.

Japan's little-known Ministry of Posts and Telecommunications (MPT) has emerged as an international force to be reckoned with, political analysts said.

MPT, thrust into the spotlight by trade rows with the U.S. And Britain, is in a position of *strength due to its control of a lucrative industry and its ties with important politicians, they said.*

"The ministry is standing athwart the regulatory control of a key industrial sector, telecommunications and information," said one diplomatic source.

"They are a potent political force," the diplomatic source said.

But MPT is finding domestic political prowess does not always help when it comes to trade friction diplomacy, analysts said.

"The ministry was a minor ministry and its people were not so internationalized," said Waseda University professor Mitsuru Uchida. "Suddenly they're standing at the centre of the world community and in that sense, they're at a loss (as to) how to face the situation."

Most recently the ministry has been embroiled in a row with London over efforts by Britain's Cable and Wireless Plc to keep a major stake in one of two consortia trying to compete in Japan's lucrative overseas telephone business.

The ministry has favoured the merger of the two rival groups, arguing the market cannot support more than one competitor to Kokusai Denshin Denwa Co Ltd, which now monopolizes the business.

It has also opposed a major management role in the planned merger for any non-Japanese overseas telecommunications firm on the grounds that no such international precedent exists.

The ministry's stance has outraged both London, which has threatened to retaliate, and Washington, which says ***the merger plan*** *is evidence of Japan's failure to honour pledges to open its telecommunications market.*

Washington is also angry over other ministry moves which it says have limited access for U.S. Firms to Japan's car telephone and satellite communications market.

Much of MPT's new prominence stems from the growth of the sector it regulates.

"What has been happening is an important shift in the economy which makes the ministry a very important place," said James Abegglen, head of the consulting firm Asia Advisory Service Inc.

A decision to open the telecommunications industry to competition under a new set of laws passed in 1985 has boosted rather than lessened MPT's authority, analysts said.

"With the legal framework eased, they became the de facto legal framework," said Bache Securities (Japan) analyst Darrell Whitten.

Close links with the powerful political faction of the ruling Liberal Democratic Party (LDP) nurtured by former Prime Minister Kakuei Tanaka are another key to MPT's influence, the analysts said.

"Other factions ignored MPT (in the 1970s), but the Tanaka faction was forward looking and ... Recognized the importance of MPT," Uchida said. Many former bureaucrats became members of the influential political group, he added.

The ministry also has power in the financial sector due to the more than 100,000 billion yen worth of deposits in the Postal Savings System, analysts said.

MPT has helped block Finance Ministry plans to deregulate interest rates on small deposits, a key element in financial liberalisation, since the change would remove the Postal Savings System's ability to offer slightly higher rates than banks, they said.

Diplomatic sources, frustrated with what they see as MPT's obstructionist and protectionist posture, have characterized the ministry as feudal.

Critics charge MPT with protecting its own turf, limiting competition and sheltering the former monopolies under its wing. Providing consumers with the best service at the lowest price takes a back seat to such considerations, they said.

But many of the ministry's actions are not unlike those of its bureaucratic counterparts in much of the Western world including Britain, several analysts said.

"The United States is really the odd man out," Abegglen said. "For a government to take the view that it wants to keep order in utilities markets is not an unusual and/or unreasonable view," he said.

Appendix II

Property precedence query retrieve the following news story because “money-fx” precedes “dollar/yen rate”. Though the Reuters-21578 data set considers this story does not have topic “money-fx”, we consider this story has the topic.

The yen is likely to start another uneven rise against the dollar and other major currencies because the Group of Seven communique contained nothing new, currency and bond analysts here said.

"Is that it? I was expecting something more than that," said one trader at a major Wall Street securities company.

Marc Cohen of Republic National Bank of New York said: "The market now has the impetus to drive the dollar lower again."

The dollar hovered between 145.50 and 147 yen in the days just before the talks. Dealers restrained their underlying bearishness and squared positions ahead of Wednesday's meeting of the finance ministers and central bankers of the top seven industrialized nations in Washington.

After more than four hours of talks, the G-7 issued a communique which merely reaffirmed the recent Paris agreement's view that prevailing currency levels were broadly consistent with economic fundamentals and that exchange rate stability should be fostered around these levels.

The dollar sank to 144.75 yen in early Tokyo trading.

*"They said that the **dollar/yen rate** was broadly in line with fundamentals when it was 154. Now they are saying it's in line when it's at 146. Will this still be so at 138 or 130?," asked Republic's Cohen.*

Japanese Finance Minister Kiichi Miyazawa fuelled speculation about the amount of fluctuation the authorities are prepared to tolerate by saying that the current yen level is still inside the range agreed on in Paris in late February.

Official statements in recent weeks had indicated that the key psychological level of 150 yen was at the lower end of the authorities' permissible range.

Dealers and analysts warned that the dollar's decline would probably be uneven. They anticipated a concerted effort to prop up the dollar and restrain the yen via a mixture of open market intervention and public comments.

Shortly after the Tokyo market opened today the Bank of Japan was detected by local dealers buying moderate amounts of dollars. The dollar rebounded to about 145.20 yen.

The sources said the market may also be wary of aggressively selling dollars for yen before Tuesday's February U.S. Trade data. The figures are expected to show a deficit of 13 billion dlrs, from a provisional 14.8 billion in January.

Property precedence query retrieve the following news story because “interest” precedes “easy monetary policy”. Though the Reuters-21578 data set considers this story does not have topic “interest”, we consider this story has the topic.

New U.S. Banking data suggest the Federal Reserve is guiding monetary policy along a steady path and is not signalling any imminent change of course, economists said.

But they also said that if money supply growth remains weak, as this week's unexpected eight billion dlr M-1 decline suggests it may, this could influence the Fed to loosen its credit reins and move toward a more accommodative monetary policy.

A Reuter survey of 17 money market economists produced a forecast of a 600 mln dlr M-1 decline for the week ended June 8, with estimates ranging from a gain of one billion dlrs to a decline of four billion. Instead, M-1 fell eight billion dlrs to 745.7 billion dlrs at a seasonally adjusted annual rate.

Coming on the heels of a 4.3 billion decrease in M-1 for the week ended June 1, this means the nation's money supply has fallen more than 12 billion dlrs in the past two weeks, economists said.

"M-1 has hit an air pocket of weakness," said Bill Sullivan of Dean Witter Reynolds Inc.

While M-1 may have lost its significance as an indicator of economic growth, Sullivan said Fed officials might be concerned the latest drop in M-1 means another month of sluggish growth in the broader monetary aggregates, M-2 and M-3, which are seen as better gauges of economic growth.

Latest monthly M-2 and M-3 data showed that as of May, both measures were growing at rates below the bottom of the Fed's 5-1/2 to 8-1/2 pct target ranges.

If money growth does not accelerate, Fed officials, concerned that this indicates economic growth is flagging, could turn toward easier monetary policy, economists said.

"Does this mean that the Fed abandons its current open market position? No," Sullivan said. "But does this mean the end of tightening for the time being? Definitely yes."

Economists said average adjusted discount window borrowings of 385 mln dlrs for the latest two-week bank statement period were lower than they had expected. Most believed the Fed had targetted a two-week borrowings average of around 500 mln dlrs.

But they said that if it had not been for a large one-day net miss in the Fed's reserve projections, the higher borrowings target would probably have been reached.

A drop in May U.S. Housing starts and continued weakness in auto sales show key sectors of the U.S. Economy are lagging, while a recent modest 0.3 pct gain in May producer prices has helped dispel inflation fears, Slifer said.

"If this continues, we can entertain the notion of Fed easing at some point," he said.

Other economists said the Fed would probably pay little attention to weak money supply growth. "It has been a number of years since M-1 has given good signs of what's going on in the economy," one said. "I don't think M-1 shows that the economy is falling apart and the Fed should ease."

Economists agreed a stable dollar will continue to be a prerequisite for any move by the Fed toward ***easier monetary policy****.*

They said the Fed is reluctant to lower short-term rates for fear this would spur expectations of a weaker dollar and higher inflation which would push up long-term yields and choke off econmomic growth.

But Sullivan said the dollar has been steady since late April. "The Fed has to determine if this represents a fundamental change for the dollar. If it does, then this gives them more room to ease," he said.

Property precedence query retrieve the following news story because “money-supply” precedes “reserve projection”. Though the Reuters-21578 data set considers this story does not have topic “money-supply”, we consider this story has the topic.

Economists said that they doubt the Federal Reserve is firming policy to aid the dollar, despite higher discount window borrowings in the latest two-week statement period and very heavy borrowings Wednesday.

Data out today show net borrowings from the Fed averaged 393 mln dlrs in the two weeks to Wednesday, up from 265 mln dlrs in the prior statement period. Wednesday borrowings were 1.4 billion dlrs as Federal funds averaged a high 6.45 pct.

"One could make a case that the Fed is firming, but it probably isn't," said William Sullivan of Dean Witter Reynolds.

Sullivan said some may assume the Fed has firmed policy modestly to support the dollar because net borrowings in the two-weeks to Wednesday were nearly 400 mln dlrs after averaging around 250 mln dlrs over the previous two months.

However, the Dean Witter economist noted that the latest two-week period included a quarter end when seasonal demand often pushes up borrrowings.

"Some might argue that the Fed was firming policy, but it looks like it tried to play catchup with reserve provisions late in the statement period and didn't quite make it," said Ward McCarthy of Merrill Lynch Capital Markets.

A Fed spokesman told a press press conference today that the Fed had no large net one-day miss of two billion dlrs or more in its ***reserve projections*** *in the week ended Wednesday.*

Still, McCarthy said it may have had a cumulative miss in its estimates over the week that caused it to add fewer reserves earlier in the week than were actually needed.

The Fed took no market reserve management action last Thursday and Friday, the first two days of the week. It added temporary reserves indirectly on Monday via two billion dlrs of customer repurchase agreements and then supplied reserves directly via System repurchases on Tuesday and Wednesday.

Based on Fed data out today, economists calculated that the two-day System repurchase agreements the Fed arrranged on Tuesday totaled around 5.9 billion dlrs. They put Wednesday's overnight System repos at approximately 3.4 billion dlrs.

"It is quite clear that the Fed is not firming policy at this time," said Larry Leuzzi of S.G. Warburg and Co Inc.

Citing the view shared by the other two economists, Leuzzi said the Fed cannot really *afford to seriously lift interest rates to help the dollar because that would harm already weak economies in the United States and abroad and add to the financial stress of developing countries and their lenders.*

"Those who believe the Fed tightened policy in the latest statement period have to explain why it acted before the dollar tumbled," said McCarthy of Merrill Lynch.

He said the dollar staged a precipitous drop as a new statement period began today on disappointment yesterday's Washington meetings of international monetary officials failed to produce anything that would offer substantive dollar aid.

In fact, currency dealers said there was nothing in Wednesday's G-7 communique to alter the prevailing view that the yen needs to rise further to redress the huge trade imbalance between the United States and Japan.

The economists generally agreed that the Fed is aiming for steady policy now that should correspond to a weekly average Fed funds rate between six and 6-1/8 pct. This is about where the rate has been since early November.

"I'm not so sure that the Fed is engineering a tighter policy to help the dollar, as some suspect," said Sullivan of Dean Witter.

If it is, however, he said that Fed probably has just nudged up its funds rate goal to around 6.25 to 6.35 pct from six to 6.10 pct previously.

Property precedence query retrieve the following news story because "wheat" precedes "agricultural produce". Instead of having topic "wheat", this story has topic "grain".

A prolonged dry spell has damaged 111,350 hectares of rice and corn plantations in 10 provinces in the central and southern Philippines, agriculture officials said.

They said some 71,070 tonnes of ***agricultural produce*** *estimated at about 250 mln pesos was lost to the lack of rainfall. They warned of a severe drought if the prevailing conditions continued until next month.*

Agriculture Secretary Carlos Dominguez said he hoped the losses would be offset by the expected increase in output in other, normally more productive areas not affected by the dry spell.

Affected were 14,030 hectares of palay (unmilled rice), representing a production loss of 22,250 tonnes valued at 77.8 mln pesos, Department of Agriculture reports said.

About 48,820 tonnes of corn from 97,320 hectares valued at 170.8 mln pesos have also been lost, they said.

Officials said the hectarage planted to palay that has been hit by the drought accounted for only one pct of national total thus the damage is considered negligible.

In the case of corn, they said the loss can be filled by production from non-traditional corn farms which diversified into the cash crop from sugar two years ago.

The Philippine Coconut Authority said coconut production in the major producing region of Bicol might drop by 25 pct to 320,000 tonnes if the dry spell continued. There were no reports of actual damage.

Property precedence query retrieve the following news story because “crude” precedes “mln barrel”. Instead of having topic “crude”, this story has topic “heating oil”.

The U.S. Court of Appeals for the Second Circuit upheld a lower court decision dismissing a suit by Apex Oil Co against the New York Mercantile Exchange and several oil companies.

The Court, however, ruled that Apex Oil could pursue anititrust and commodities market manipulation allegations against Belcher Oil Co, a unit of Coastal Corp <CGP>.

Apex Oil, primarily a trading company, charged that several companies, including Belcher, and NYMEX conspired to force it to deliver heating oil it had sold on the mercantile exchange, knowing Apex could not make full delivery.

*The NYMEX ordered Apex to deliver four **mln barrels** of heating oil sold via a February 1982 heating oil contract. Apex eventually fulfilled this obligation but claimed damages.*

Richard Wiener, attorney for Apex at Cadwalader Wickersham and Taft, said the company has not yet decided whether to pursue its case against Belcher Oil.

The NYMEX, meanwhile, has a counterclaim pending against Apex Oil, seeking an unspecified amount of attorney's fees and 15 mln dlrs in punitive damages, according to a NYMEX spokeswoman.

Property precedence query retrieve the following news story because “ship” precedes “freight cost”. Instead of having topic “ship”, this story has topic “trade”.

*The Commerce Department said on that insurance and **freight costs** for imported goods of 1.45 billion dlrs were included in the February trade deficit of 15.1 billion dlrs reported on Tuesday.*

The department is required by law to wait 48 hours after the initial trade report to issue a second report on a "customs value" basis, which eliminates the freight and insurance charges from the cost of imports.

Private-sector economists emphasized that the Commerce Department was not revising down the deficit by 1.45 billion dlrs but simply presenting the figures on a different basis.

A report in the Washington Post caused a stir in the foreign exchanges today because it gave the impression, dealers said, that the underlying trade deficit for February had been revised downward.

The Commerce department would like to have the law changed to permit it to report both sets of figures simultaneously.

"My feeling is the second one is a better report but there's legislation that requires us to delay it two days," said Robert Ortner, Commerce undersecretary for economic affairs.

"But this has been going on for a long time and no one pays any attention to the second figure."

The 15.1 billion dlr February trade deficit compared with a revised January deficit of 12.3 billion dlrs.

The law requiring a 48-hour delay in publishing the monthly trade figure excluding freight and insurance was passed in 1979.

Reportedly the feeling was the first figure, which includes customs, freight and insurance, allowed a better comparison with other countries that reported their trade balances on the same basis.

The second figure, which would always be lower by deducting freight and insurance, presents the deficit in a more favorable light for the Reagan administration.

Ortner said he would like to see the law changed to eliminate the 48-hour delay in reporting the two figures.

"We're considering it," he said, "It's one of those dinosaur laws and I think it's time has come."

The second figure, which would always be lower by deducting freight and insurance, presents the deficit in a more favorable light for the Reagan administration.

Ortner said he would like to see the law changed to eliminate the 48-hour delay in reporting the two figures.

"We're considering it," he said, "It's one of those dinosaur laws and I think its time has come."