

**SEARCHING FOR ENTITIES:
WHEN RETRIEVAL MEETS EXTRACTION**

by

Qi Li

B. S., Peking University, 2000

M. S., Peking University, 2006

Submitted to the Graduate Faculty of
the School of Information Sciences in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2011

UNIVERSITY OF PITTSBURGH
SCHOOL OF INFORMATION SCIENCES

This dissertation was presented

by

Qi Li

It was defended on

October 7th 2011

and approved by

Daqing He, PhD, Associate Professor

Michael Spring, PhD, Associate Professor

Paul Munro, PhD, Associate Professor

Jung Sun Oh, PhD, Assistant Professor

Fu-Chiang (Rich) Tsui, PhD, Assistant Professor

Dissertation Director: Daqing He, PhD, Associate Professor

**SEARCHING FOR ENTITIES:
WHEN RETRIEVAL MEETS EXTRACTION**

Qi Li, PhD

University of Pittsburgh, 2011

Retrieving entities from inside of documents, instead of searching for documents or web pages themselves, has become an active topic in both commercial search systems and academic information retrieval research area. Taking into account information needs about entities represented as descriptions with targeted answer entity types, entity search tasks are to return ranked lists of answer entities from unstructured texts, such as news or web pages. Although it works in the same environment as document retrieval, entity retrieval tasks require finer-grained answers—entities—which need more syntactic and semantic analyses on germane documents than document retrieval. This work proposes a two-layer probability model for addressing this task, which integrates germane document identification and answer entity extraction. Germane document identification retrieves highly related germane documents containing answer entities, while answer entity extraction finds answer entities by utilizing syntactic or linguistic information from those documents. This work theoretically demonstrates the integration of germane document identification and answer entity extraction for the entity retrieval task with the probability model. Moreover, this probability approach helps to reduce the overall retrieval complexity while maintaining high accuracy in locating answer entities.

Serial studies are conducted in this dissertation on both germane document identification and answer entity extraction. The learning to rank method is investigated for germane document identification. This method first constructs a model on the training data set using query features, document features, similarity features and rank features. Then the model

estimates the probability of the germane documents on testing data sets with the learned model. The experiment indicates that the learning to rank method is significantly better than the baseline systems, which treat germane document identification as a conventional document retrieval problem.

The answer entity extraction method aims to correctly extract the answer entities from the germane documents. The methods of answer entity extraction without contexts (such as named entity recognition tools for extraction and knowledge base for extraction) and answer entity extraction with contexts (such as tables/lists as contexts and subject-verb-object structures as contexts) are investigated. These methods individually, however, can extract only parts of answer entities. The method of treating the answer entity extraction problem as a classification problem with the features from the above extraction methods runs significantly better than any of the individual extraction methods.

TABLE OF CONTENTS

1.0 INTRODUCTION TO ENTITY RETRIEVAL	1
1.1 PROBLEM STATEMENT	2
1.2 RESEARCH GOALS	5
1.3 TERM DEFINITIONS	6
1.4 ENTITY RETRIEVALS OUTSIDE UNSTRUCTURED TEXTS	8
1.5 LIMITATIONS AND DELIMITATIONS	11
1.6 CONTRIBUTIONS	13
1.7 OUTLINE	14
2.0 RELATED WORK	16
2.1 DOCUMENT RETRIEVAL	16
2.1.1 Vector Space Model	16
2.1.2 Language Model	18
2.1.3 Link Analyses	20
2.1.4 Topic Detection and Query Construction	21
2.2 ENTITY EXTRACTION	22
2.2.1 Named Entity Recognition Tools	23
2.2.2 Rule-based Methods	25
2.2.3 Supervised Learning	26
2.2.4 Semi-supervised Learning	27
2.2.5 Unsupervised Learning	28
2.3 ENTITY RETRIEVAL IN INEX AND TREC	30

3.0 A TWO-LAYER RETRIEVAL AND EXTRACTION PROBABILITY MODEL	31
3.1 OVERALL ARCHITECTURE	31
3.2 A MODEL COMBINING DOCUMENT RETRIEVAL AND ENTITY EXTRACTION	33
3.3 GERMANE DOCUMENT IDENTIFICATION IN TREPMM	35
3.4 ANSWER ENTITY EXTRACTION IN TREPMM	38
3.5 SUMMARY	41
4.0 EXPERIMENTAL METHODOLOGY	42
4.1 EXPERIMENTAL QUESTIONS AND EVALUATION FRAMEWORK	42
4.2 TEST COLLECTIONS	44
4.2.1 INEX Entity Ranking Track 2007/2008	44
4.2.2 TREC Entity Track 2009/2010	46
4.2.3 RAP Collection	49
4.3 EVALUATION METRICS	50
4.4 INFORMATION RETRIEVAL TOOLS	51
4.5 ENGLISH-LANGUAGE NLP TOOLS	52
5.0 GERMANE DOCUMENT IDENTIFICATION	54
5.1 GERMANE DOCUMENT IDENTIFICATION AS CONVENTIONAL DOCUMENT RETRIEVAL	54
5.2 ENTITY TYPE LANGUAGE MODEL	57
5.2.1 Category Similarity Strategies	59
5.2.2 Document Similarity Strategies	60
5.2.3 Experiments	61
5.2.4 Summary	64
5.3 LEARNING TO RANK	64
5.3.1 Germane Document Identification with a Learning to Rank Approach	65
5.3.2 Variety of Features	68
5.3.2.1 Query features	68
5.3.2.2 Document features	69

5.3.2.3 Rank features	70
5.3.2.4 Similarity features	70
5.3.3 Evaluation	73
5.4 SUMMARY	80
6.0 ANSWER ENTITY EXTRACTION	83
6.1 ANSWER ENTITY EXTRACTION WITHOUT CONTEXTS	87
6.1.1 Answer Entity Extraction with Named Entity Recognition Tools	88
6.1.2 Knowledge Base Entity Type Filtering	92
6.1.3 Discussion	94
6.2 SYMBOLIC CONTEXTS: TABLE/LIST EXTRACTION	95
6.2.1 Answer Entity Extraction from Tables/Lists	96
6.2.2 Experiment on Table/List Extractions from the Web Pages	97
6.2.3 Table/List Extractions from Knowledge Base	99
6.2.4 Discussion	102
6.3 SYNTACTIC CONTEXTS: BOOTSTRAPPING	103
6.3.1 Bootstrapping Algorithm	104
6.3.1.1 Pattern Generation and Pattern Weighting	105
6.3.1.2 Pattern Matching Strategy	108
6.3.2 Experiments on Company-Product and Company-Location	109
6.3.3 Experiments on Twenty Topics of TREC 2009 data set	112
6.3.4 Discussion	115
6.4 ANSWER ENTITY EXTRACTION AS A CLASSIFICATION PROBLEM	116
6.4.1 Answer Entity Extraction: a Binary Classification Problem	118
6.4.2 Evaluation and Results	120
6.4.3 Discussion	123
6.5 EXPERIMENTS ON TREPM MODEL	124
6.5.1 Evaluation on TREC 2009 Task	124
6.5.2 Evaluation on TREC 2010 Task	126
6.5.3 Topic Analysis on TREC entity retrieval	128
6.6 SUMMARY	130

7.0 CONCLUSION AND FUTURE WORK	132
7.1 TREPM MODEL REPRESENTATION	132
7.2 GERMANE DOCUMENT IDENTIFICATION	133
7.3 ANSWER ENTITY EXTRACTION	134
7.4 THE FUTURE OF ENTITY RETRIEVAL AND ITS APPLICATION	135
7.4.1 Future Work	136
7.4.2 Applications on Medical Entity Retrieval	138
8.0 ACKNOWLEDGMENTS	141
BIBLIOGRAPHY	142
APPENDIX A. HOMPAGE DETECTION FOR THE TREC TASK	151
APPENDIX B. ENTITIES OF PRODUCTS EXTRACTED FROM WIKIPEDIA INFOBOX	155
APPENDIX C. A SAMPLE DOCUMENT OF INEX 2007:“NEXT”	157
APPENDIX D. A SAMPLE DOCUMENT OF INEX 2009:“NEXT”	159
APPENDIX E. TWENTY TOPICS IN TREC 2009 ENTITY TRACK	161

LIST OF TABLES

1	The comparison of different retrieval systems	9
2	Methods of germane document identification in TREC and INEX	17
3	Methods of answer entity extraction in TREC	24
4	The testing sets	45
5	The annotation summary of 20 topics in the TREC 2009 data sets	48
6	Results of germane document identification as conventional document retrieval	56
7	Results of document similarity estimations in the entity type language model	63
8	Results of the entity type estimation in the entity type language model	64
9	Results of the learning to rank method for germane document identification .	76
10	The features with their weights in the logistic regression model	79
11	The answer entity context structures for 20 Topics in TREC 2009	85
12	Context structures for 20 topics in TREC 2009	87
13	Results of named entity recognition tools for answer entity extraction	90
14	Results of Wikipedia entity type filtering for answer entity extraction	93
15	Results of table/list detections for answer entity extraction	99
16	Patterns for extracting the company-product pair	114
17	Results of the bootstrapping method for answer entity extraction	115
18	Results of the learning based method for answer entity extraction	122
19	The features with their weights in the learning-based extraction	123
20	Results of entity retrieval with TREPM model	125
21	Results of homepage detection	153
22	Entity homepage sets for the topic of products of MedImmune, Inc.	154

LIST OF FIGURES

1	Entity retrieval model	4
2	Information retrieval tasks	8
3	The Two-Layer Retrieval and Extraction Probability Model (TREPM)	32
4	The evaluation framework for TREMP Model	43
5	A sample topic of INEX entity retrieval task	46
6	A sample topic of TREC entity retrieval task	47
7	The learning to rank framework	66
8	The algorithm extracting answer entities from tables/lists	97
9	A sample of Infobox	100
10	The algorithm extracting entities from knowledge bases	101
11	Bootstrap framework of topic-answer entity pair extraction	106
12	Bootstrap results of company-location extraction with One-Entity matching	110
13	Bootstrap results of company-product extraction	111
14	JRIP rules of entity homepage detection	152

1.0 INTRODUCTION TO ENTITY RETRIEVAL

In answering questions such as “What are Microsoft’s products” and “Who are Microsoft’s competitors,” related information is scattered over different web places, such as companies’ homepages, encyclopedia pages, and news articles which needs to be collected. In order to obtain an answer, users need to identify the names of key entities (e.g., Microsoft), analyze how they are related to each other (e.g., products or competitors), and then piece the assorted bits of information together to formulate an answer. Manually repeating the search process for additional candidates is both tedious and prone to error. In contrast to the Gutenberg Age, when people spent hours in the library gathering information on a certain topic, we now have achieved great improvements in locating information in documents with machine-indexing techniques. However, we now expect to find answers at a finer-grained level (such as passage, entity, or snippet) from various sources (such as news and Web pages) and in several formats (such as books, web pages, emails, blogs, or simply computer files).

Traditional search engines, returning results in a sequentially ranked list of documents or aggregating the results in clusters based on documents or web pages at the smallest unit, may not directly provide answers to users’ information needs in a finer unit. Even though search engines analyze hyper-links and anchor texts, they cannot solve this problem. The deficiency is caused partly by the limitation of the basic assumption in document retrieval that keywords in documents are unordered or a “bag of words.” The co-occurrence of terms at document level make it hard to estimate the answers at the entity level. The deficiency also comes from relevance judgments. If any piece of the document is relevant regardless of how small that piece is in relation to the rest of the document, the current retrieval systems will mark it as relevant to some degree. In this case, a page is retrieved only if it matches the words in the user’s query. This kind of search engine will eschew analyses involving entities

among topics and answers since the identification of entities has not yet occurred. Entity retrieval, on the other hand, assumes the answer entities have some sort of relationship with the topic entities, and is evaluated using a different unit of entities, which will be a useful alternative for document retrieval on large and diverse Web environments.

1.1 PROBLEM STATEMENT

This study focuses on entity retrieval—the search for “objects” of “entities”— on unstructured noisy documents, like HTML pages, in response to users’ information needs which are expressed in the natural languages. Here are some key characteristics for entity retrieval, which makes it different from other retrieval tasks:

Querying about entities , instead of querying on relevant documents, is one of the big differences between document retrieval and entity retrieval. Entity retrieval returns the answer of entities existing in documents as finer units.

Retrieving from un-structured noisy documents is another important factor for the retrieval task. Compared to semantic web retrieval which retrieves on the semantic web, entity retrieval in this study is based on noisy web documents.

Descriptions of users’ information needs with special entity types in entity retrieval differ from the one in document retrieval, which do not specify the entity types. In order to describe users’ information needs, topics will be described narratively, such as the “narrative” field in the TREC entity retrieval task. In most cases, the description of the “narrative” field can be viewed as a topic entity with phrases describing the relations between the topic entity and the query entities. For example, “organizations that award Nobel Prizes” describes a user’s information need, which can be viewed as the topic entity (i.e., Nobel Prizes), the target entity type (i.e., organizations), and the relation (i.e., award) between the topic entity and the target entity.

There are many definitions of entities in the literature. An entity can be a thing that is recognized to have an “independent existence” and can be “uniquely identified”

[Beynon-Davies, 2004]. An entity can be an instance of the pre-definite types/fields in the database or knowledge base. An entity can be an atomic element with categories such as the names of persons, organizations, locations, expressions of times, and quantities in unstructured texts [Tjong Kim Sang, 2002]. In this dissertation, an entity e_t is defined as a named object with a term surface e and an associated type t . For example, “Washington” in the document can be an entity with the type of location or person depending on different contexts. In the sentence of “Washington chopped down a cherry tree”, “Washington” is the entity with the surface of “Washington” and the type of person, while “Washington” in “Washington State” is the entity with the surface of “Washington” and the type of location. This study only experiments on four types of entities—persons, locations, organizations, and products for the purpose of evaluations. Moreover, in this research, entity types, entity classes, and entity categories are treated as equivalent.

The description of topic entities and answer entities is viewed as the relations between two entities, when users describe their information needs in the entity retrieval task [Nardi and Brachman, 2003]. Therefore, for one entity a with the type A and the other entity b with the type B , the relation r with the type of R is defined as $r^R \subseteq a^A \times b^B$. Note that two entities with the same types might have different relations. For example, the relation between “Mary” with the type of persons and “Pittsburgh” with the type of locations can be the relation of “was born in” for the type of *born-in* or the relation of “was studies in” for the type of *study-in*.

A basic entity retrieval problem is set up as follows: we assume that there exists an entity set $E = \{E_{t_1}, \dots, E_{t_i}, \dots, E_{t_m}\}$, where $E_{t_i} = \{e_{1t_i}, e_{2t_i}, \dots, e_{lt_i}\}$ is a group of entity instances with the type of t_i . In the retrieval environment, there is a corpus $C = \{D_1, \dots, D_J, \dots, D_N\}$, where D_J is a document. The corpus C contains an entity set E , which includes the different types of entities, such as locations, companies, persons, et al. Entity retrieval is defined as the matching of some stated user queries about an entity against a set of entities existing in free-form texts. The information needs are represented as descriptions and target types. The matching process identifies the correct entities, not only their surfaces but also their types. The relevancy evaluation is the same as most document retrieval studies, which can be binary relevance (i.e., relevant and non-relevant) or three levels (i.e., non-relevant, relevant,

and highly relevant). The evaluated answer entities may be sent back to the system for either re-constructing the queries or re-building document representations to improve the entity extraction task. Some tasks, such as TREC or INEX, require the URIs/URLs of entities as answers. Because these tasks assume the same entity can be represented in the different names, the URIs/URLs of entities can help to refer the different names of entities to the same entity objects. In this study, however, we focus on the entity retrieval itself, so we treat answer entities, instead of the URIs/URLs of entities, as results. The model of entity retrieval follows the conventional information retrieval process [Manning et al., 2008], as shown in Figure 1.

A typical entity retrieval task in my research, for example, is to find the answers for “products of MedImmune, Inc,” where the query is asking for the entity of “products of MedImmune, Inc” (target entity/answer entity) with the type of products. The answer entities are terms such as “Synagis”, “FluMist”, and “Ethylol”, where their types are products.

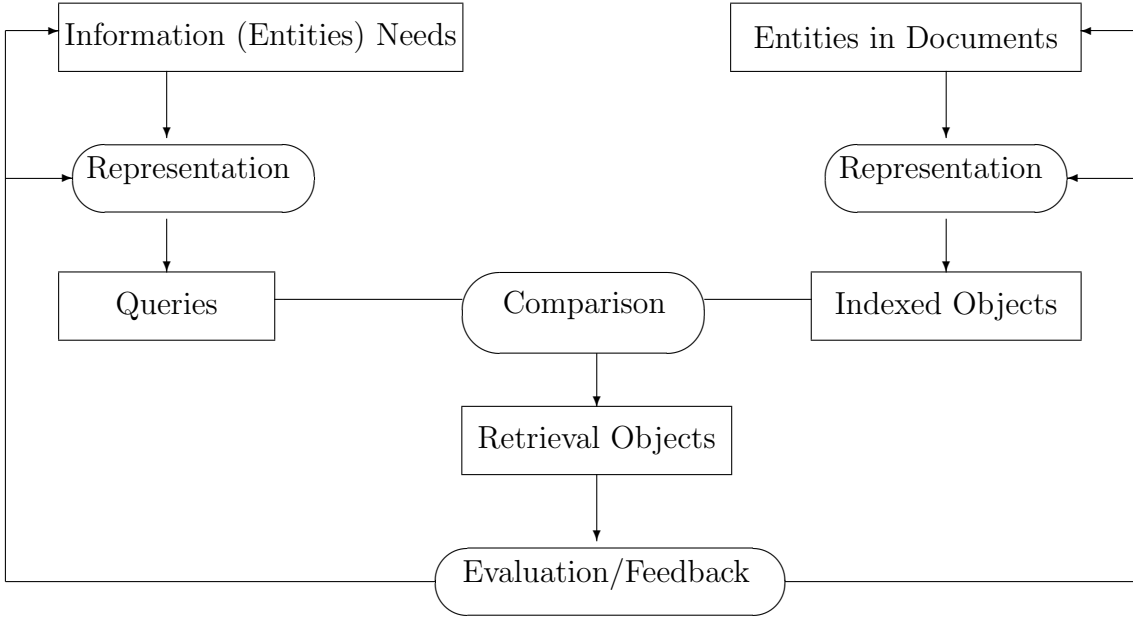


Figure 1: Entity retrieval model

1.2 RESEARCH GOALS

Entity retrieval in this study is driven by several research goals. The most critical one is to find entities in an effective and efficient manner. Entity retrieval, as one type of information retrieval task, shares the same questions raised in information retrieval—performance and efficiency. On the one hand, a retrieval system needs to come up with answers within seconds. Text collections, growing faster than hardware performance, become a challenge for indexing and sorting algorithms. We have to consider the system’s efficiency in the query execution time. On the other hand, the performance of entity retrieval system, including precision and recall, is also important. Given a query, all identified entities should be ranked in a manner where highly relevant entities are ranked above less relevant or non-relevant ones. Although precision has become more important than recall in the Web environment, recall in the entity retrieval task is still important because people will not only consider how many answers are correct but also how the system can answer questions.

In order to find the correct entity answers effectively and efficiently, this study proposes a combined probability model integrating document retrieval and entity extraction. Document retrieval aims to quickly and efficiently find the germane documents containing the answer entities; whereas, entity extraction is in charge of correctly and effectively extracting the answer entities. Modeling entity retrieval problems as a combination of document retrieval and entity extraction helps to reduce the complexity of entity retrieval into two separated sub-tasks. The hypothesis is that the global optimal problem of entity retrieval can be simplified into two local optimization problems. The advantages of this localized optimization are that it not only lowers the computational complexity, but also adapts more state-of-the-art techniques from both document retrieval and entity extraction disciplines into entity retrieval tasks.

By decomposing the entity retrieval problem, this research evaluates whether this probability model can achieve the retrieval goals effectively and efficiently; whether document retrieval can effectively find the germane documents containing the answer entities; whether entity extraction can easily and correctly identify the entities existing in one document or scattered among several documents. Document retrieval explores the methods to improve

germane document identification, such as query generations, entity type searches, and learning to rank. Entity extraction explores methods, such as named entity recognition tools, knowledge base (e.g., Wikipedia) entity extractions, table/list extractions, syntax extraction, or treating named entities identification as a classification task. This research, therefore, will evaluate whether the combination of germane document identification and answer entity extraction can work together effectively to detect answer entities. The results of entity retrieval are entity ranked lists, and are evaluated by the system performance, such as precision and recall.

1.3 TERM DEFINITIONS

Formal definitions of entity retrieval, entities, entity sets, entity types, topic entities, target entities, topics, queries as well as germane documents in this study are listed here.

Entity retrieval is the search task that finds the entity objects in unstructured noisy documents like HTML pages with regards to users' information needs. For example, for the query asking for the product entities of MedImmune, Inc (target entity/answer entity), the answer entities are terms such as Synagis, FluMist, and Ethyol, where their types are products.

Entity (denoted as E_t) is a thing recognized as capable of an independent existence, which can be uniquely identified with a certain type of t . Usually it uses the entity type t to distinguish the different entities. For example, the location entity means the entity with the type of location.

Entity instance (denoted as e_{ti}) is every individual entity with the type t . The entity instances of organization, for example, include Department of Information Sciences, School of Information Sciences, and University of Pittsburgh.

Each entity is assigned a **entity type** t (or, equivalently, category or class). The entity instance of School of Information Sciences, for example, belongs to the type of organization. This work treats the relation between an entity type and its entity instance as the relation of a class and its instance.

Topic entity (or **query entity**), in this thesis, refers to the central entities in the topics describing users' information needs. For example, the topic entity for products of MedImmune, Inc is MedImmune, Inc, which is the central entity in the topic. In TREC entity retrieval task, the topic entities are specially marked up in a separate field with the tag of <entity_name>.

An **answer entity** or a **target entity** refers to the retrieved answers for the entity retrieval system in this study. The answer entities for the query of products of MedImmune, Inc., for example, are the target entities of Synagis, FluMist, and Ethyol.

For our research purpose, **relations**, in this study, refer to the relation (e.g., product of) between the topic entity (e.g., MedImmune Inc.) and the answer entities (e.g., Synagis, FluMist, and Ethyol).

A **topic** represents a user's information needs. This study uses the natural language description to depict users' requirements (e.g., product of MedImmune Inc.), the topic entities to define the subject of information need (e.g., MedImmune Inc.), and the (answer) entity type (e.g., product) to limit the type of the retrieved answers.

A **query** is the texts containing the data or string to be passed to the search system. This study focuses on the Web search, so the queries could be a natural language text string. We should specifically note that with the same information needs and the same search environment, the search queries can vary according to the different criteria or assumptions. For example, the topic of products of MedImmune, Inc. in the Web search environment, the query can be "products of MedImmune Inc", or it can be "MedImmune LLC produces", assuming that MedImmune LLC is a formal name of MedImmune Inc. and the verb "produces" is another way to represent the relation.

A **germane document** is the document which contains the answer entities for answering users' information needs. For example, http://www.medimmune.com/about_us_products.aspx is the germane document for the topic of product of MedImmune Inc because it contains the answer entities of Synagis, Flumist, and Ethyol.

1.4 ENTITY RETRIEVALS OUTSIDE UNSTRUCTURED TEXTS

On the one hand, entity retrieval not only exists in Web search, but also it exists in database and semantic web search. On the other hand, entity search retrieves from the same plain texts as document retrieval, but they also differ in their returned units. Figure 2 shows these retrieval tasks in two dimensions. The horizontal dimension is the structure of retrieval contents. The left part consists of more structured data and the right part contains more un-structured data. The vertical dimension is the returned unit. The higher the vertical dimension, the finer unit the search produces. The closer to the origin the easier the retrieval problem is. Entity retrieval studied in this dissertation is on the upper right part, which means it is among the hardest retrieval tasks.

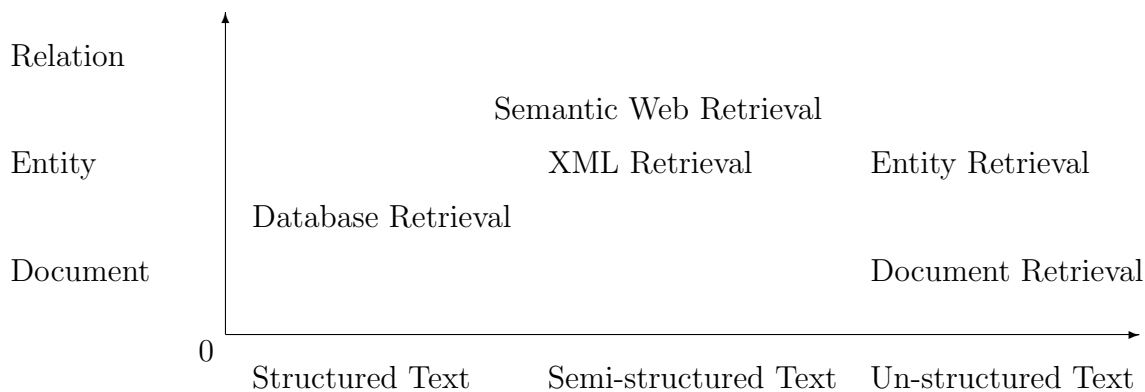


Figure 2: Information retrieval tasks

Table 1 summarizes the differences among the above retrieval systems in terms of retrieval models, data structures, and query languages.

Database searches are designed to find structured data: sets of records that have values for predefined attributes such as product numbers and product prices. For example, some highly structured search problems are best solved with a relational database because the product table contains an attribute for short textual descriptions about each product. The search results from a database are a set of entries without any ranking, which means database search is one kind of boolean search (exist or does not). Another difference between database retrieval and entity retrieval is that database systems build on top of structured

Table 1: The comparison of different retrieval systems

Features	Relational Databases	Semantic (Web) Search	XML retrieval	Web Document Retrieval	Entity Retrieval
Object	Record	RDF Elements	Structured Document (trees structure)	Unstructured Documents	Unstructured Documents
Model	Relational calculus	Metadata data Model	Structured document retrieval	Vector space and others	Document Retrieval + Entity Extraction
Main data structure	Table	RDF Triple	Inverted Index with fields	Inverted Index	Combination
Queries	SQL	SPARQL	Structured Text Queries	Text Queries	Text Queries
Ranking	No	No	Meaningful	Meaningful	Meaningful

information—records—and store all records in the tables. They can use relational calculus not only for retrievals but also inferences and deductions. In contrast to relational databases, however, entity retrieval systems obtain information from unstructured text, which makes retrieval answers more difficult to locate and harder to do inferences in the results.

Semantic search improves search accuracy by adding the understanding of searcher intents, as well as the contextual meanings of terms in the searchable data spaces, on the Web or within a closed system, in order to generate more relevant results [Guha et al., 2003]. Both semantic web retrieval and entity retrieval involve entities associated with their types as well as relations between these entities. Semantic web retrieval, however, includes not only entity search but also other retrievals, such as relation search. In some run-time systems, semantic search relies on the semantic web, which is built on top of RDF documents representing a well-defined ontology or schema with concepts and relations [Guha et al., 2003]. For example, SHOE [Luke and Rager, 1996], Freebase [Bollacker et al., 2008] and DBpedia

[Auer et al., 2007] are some of these semantic systems. Although some systems are based on unstructured document retrieval, they still match the entities to knowledge bases for entity extractions. For example, Balog matches the candidate entities to Freebase for entity retrieval [Balog et al., 2010b]. Entity retrieval in this research differs from these systems in that it is based on searching plain texts, which are the documents without any pre-defined semantic information.

XML Retrieval is the content-based retrieval of documents structured with XML (eXtensible Markup Language). So-called markup languages such as SGML or XML are widely used to annotate text structures in a machine-readable form. XML retrievals are used to search for information in structured documents. The development of structured retrieval has been driven by the INEX evaluation initiative. Because these special structures are assigned to the document, a full-fledged XML retrieval system provides more flexibility and completeness with respect to the formulation and execution of structural queries. Therefore, these searches also benefit from structural indexing and retrieval procedures. Earlier studies on structured retrievals by Navarro [Navarro and Baeza-Yates, 1995] have already considered most of the functionalities that can be expected from current systems working with XML data. Freely composing queries with contents and structure conditions are allowed in XML retrieval. Special query languages are designed to express structural requests on XML like XQuery Full-Text [Amer-Yahia et al., 2007] or NEXI [Trotman, 2004]. Moreover, XML retrieval does not require the users to specify fields of interests at indexing time, but allows them to query the content of any tagged fragment of the collection. These features ask for different index designs. Entity retrieval in this research assumes that all the documents are plain texts, which makes it difficult to index fragments found within collections.

Document retrieval, as always, is important in the history of retrieval. It regards each document as an atomic unit of interests, and does not distinguish whether some parts of a document are relevant to information needs while other parts are not. The user of a document retrieval system will find a sentence snippet if it is considered relevant to his/her query. Also the relevance estimation is based on the content of the entire document. If one sentence in the document is highly relevant but the other parts are not, the final relevance estimation of the entire document is considered lower than those of shorter documents with the same

keywords in the document but is exclusively about the topic of interests. From an indexing perspective, document retrieval allows the construction of efficient inverted document index structures.

1.5 LIMITATIONS AND DELIMITATIONS

While many researchers are working actively in entity retrieval, a developed entity retrieval system is still a long way from completion, especially within the World Wide Web. This study proposes a probability model to decompose the entity retrieval task into two subtasks—germane document identification and answer entity extraction. By doing this, the global optimal problem is turned into two local optimization problems. However, there are some limitations.

First of all, the decomposition cannot guarantee that the whole system always achieves the best results. Although the goal of the two-layer probability model is to effectively and efficiently find the answers according to users' information needs, it is a heuristic rule and does not guarantee the best results. In the most extreme cases, for example, when answers are evenly distributed over documents and within a variety of contexts, the performance of the combination probability model will drop, like "Who have their own websites?" There are a huge number of documents that contain the answer entities and there are also a large number of possible patterns for the answer entities, which makes it harder for the model to achieve the best answers with this localized optimum. However, the advantages of this model, as mentioned, are that it can effectively and efficiently find the possible answers.

The second limitation to this combined model is that germane document identification, with the assumption of bag-of-words searching, can fail at detecting the best documents for further entity extraction. This assumption limits document retrieval models to only consider the co-occurrences of the words but without considering the semantic meanings between the words. With the non-proper queries generated for topics and the ranking strategies, it will cause the retrieval inaccuracy. For example, for the query of "organizations that award Nobel prizes", after stemming and removing stop words, the retrieval task is turned into

finding documents with co-occurrences of the tokens “organization”, “award”, “nobel”, and “prize”, which are equal to the query of “organizations awarded Nobel prizes.” Although some methods such as query reconstructions might fix these problems partially, it cannot be solved completely. For example, if the query “Nobel prizes” instead of “organizations that award Nobel prizes” is used for the retrievals, more semantic analyses work will be applied in the answer entity extraction to further extract the answers. Even so, the query re-writing or entity extraction can only partially compensate for some of the drawbacks from the model assumptions of document retrieval.

The open Web environment is heterogeneous and distributed and entities in this environment are also various and multi-typed. These require significant new progresses in the development of entity retrieval in order to identify the correct entities on the scale of the Web. Some open issues remain.

First of all, entities are quite subjective and flexible concepts and used to describe the existences of things in the world by different people who usually have different viewpoints with various aims. For example, for the same term “apple”, people with different viewpoints can mark it as a “fruit” or a “company”. Moreover, with the same viewpoints and the same aims, different users might use different notation systems to represent them. For example, the type of “company” can be represented as a flat structure or as a hierarchical structure such as “/Business/Company”. Therefore, consistently representing these concepts will be a big challenge to entity extraction systems. This study will not further discuss the possible different representations of the entities.

Secondly, entity identification is a laborious and tedious process. Although many methods can be applied to different domains to extract different entity types, the study will continue to focus on four types of entities—persons, locations, organizations and products.

Finally, although entity retrieval can be the retrievals in various media or data formats, this work focuses only on searches in unstructured texts, especially the noisy HTML web pages, instead of searching in structured data (e.g., databases) or semi-structured data (e.g., semantic web).

1.6 CONTRIBUTIONS

This thesis has studied the problem of entity retrieval represented in a combined probability model. It makes the following four contributions.

First, the probability model proposed in this study decomposes the “black box” of entity retrieval into germane document identification and answer entity extraction. This decomposition, at the same time, separates the word-independence factors from word-dependence factors. In this dissertation, word independence means, in the language model, we assume that in a document all the words are independent to each other, i.e., $p(d) = p(w_1)p(w_2)\dots p(w_n)$. In fact, this is a unigram model. The word-dependence means we assume that in a document the words are related each other. Therefore, we need to analyze the semantic meaning between the words, instead of simply treating them as a unigram model. Information needs for entity retrieval task usually require semantic analyses of topics. Therefore, germane document identification efficiently narrows down document pool into a smaller set based on word-independence factors. Answer entity extraction deeply analyze the sentence structures or document formats of the small set of germane documents in order to extract the answer entities based on word-dependence factors. Because the deep sentence analysis or document format analysis is time-consuming work, narrowing down the pool of documents to be parsed can significantly decrease the time needed for entity retrieval, while increasing the accuracy of entity detections.

Second, this study demonstrates the decomposition process in a theoretical way. Although many groups actively in the entity retrieval competition follow the same ideas of the decomposition the entity retrieval task into the retrieval step and the extraction step, it is the first time to demonstrate it from a theoretical way using a probability model. Chapter 3 proposes a probability model and proves that the entity retrieval problem can be decomposed into germane document identification and answer entity extraction. Moreover, this dissertation proves the calculation time of this model can be significantly decreased using the big-o notation.

Third, with the decomposition, the system performance can be evaluated in two parts, which in turn improves the overall system. Compared to the previous method, especially the

entity retrieval competition task such as TREC or INEX, of treating entity retrieval as one “black box” and evaluating the “black box” as a whole, the decomposition can clearly identify two steps and evaluate them individually in order to detect the sources of the improvements. Additionally, this decomposition can help to find the limitations of sub-steps in the system, which can further improve the entity retrieval.

Lastly, many state-of-the-art methods can be introduced to the system as well as each individual layer, e.g., topic detection. Especially, this thesis introduces the learning to rank method for improving germane document identification, which is the first time using learning-based methods in germane document identification. The answer entity extraction task is treated as the query-dependent extraction method, which is more precise than the previous method of treating this task as query-independent extraction. Based on the query-dependent extraction, the context is introduced for the answer entity extraction. The methods of table/list extraction and sentence syntax extraction are used to improve the entity extractions in the documents.

1.7 OUTLINE

The remaining chapters are organized as follows. Chapter 2 reviews the development of entity retrieval and its related disciplines, such as document retrieval and entity extraction. In Chapter 3, a two-layer probability model combining information retrieval and entity extraction (TREPMP) is proposed as a generalized representation for an entity retrieval system. This chapter describes entity retrieval from a theoretical view and explains how two layers—germane document identification and answer entity extraction—work together and how they symbiotically contribute each other for the benefit of the whole task. Chapter 4 summarizes the research hypotheses and resources used in the experiments including evaluation framework, collections for evaluation, evaluation metrics, and tools for corpus preprocessing and document retrieval as well as entity extraction. Germane document identification, the first layer of TREPMP, is investigated in Chapter 5. It explores the methods of treating germane document identification as a conventional document retrieval problem, the entity type lan-

guage model, and the learning to rank method. Chapter 6 focuses on the second layer of TREPM. The methods of answer entity extraction, such as named entity recognition tools, knowledge bases, table/list extractions, the bootstrapping method, as well as the learning based answer entity extraction, are investigated. Chapter 7 summarizes the whole thesis and discusses my future work.

2.0 RELATED WORK

This chapter reviews the state-of-the-art work on entity retrieval (ER) and some related disciplines. Entity retrieval in this study, to some degree, is regarded as a process combining germane document identification and answer entity extraction. Rather than exhaustively surveying prior work, this chapter reviews the evolution of germane document identification and answer entity extraction in order to analyze alternative or combination approaches that identify entities in the linguistic and semantic levels. At the end of this chapter, we review the entity retrieval task in two famous competitions: TREC and INEX.

2.1 DOCUMENT RETRIEVAL

Document retrieval, sometimes referred to as (or as a branch of) text retrieval, allows users to locate the relevant documents with regards to their information needs. It treats documents as the atomic units of users' interests regardless whether the whole document or just part of the document is relevant to their information needs. The methods for germane document identification are adapted from document retrieval approaches, such as vector space model, language model, link analysis, and query constructions. The germane document identification methods used in TREC and INEX are summarized in Table 2.

2.1.1 Vector Space Model

The vector space model is an algebraic model that represents text documents as term vectors. For example, a document, d , in the corpus is represented as a term vector (i.e.

Table 2: Methods of germane document identification in TREC and INEX

Conf.	Technique	Method	Reference
INEX	Category	Similarity	[Vercoustre et al., 2008], [Koolen et al., 2010]
INEX	Category	Language model with smoothing	[Jiang et al., 2009], [marie Vercoustre et al.,], [Balog and de Rijke, 2006]
INEX	XML structure index/retrieval		[Rode et al., 2009], [Craswell et al., 2009]
TREC	Links/Anchor	Anchor-based entity	[Serdyukov and de Vries, 2009], [Kaptein and Kamps, 2009]
INEX	Links	Link-based entity authority	[Vercoustre et al., 2009]
TREC	Links/Anchor	Indexing	[Kaptein and Kamps, 2009]
INEX	Links	Language model with smoothing	[Balog and de Rijke, 2006]
INEX	Topic Difficulty Detection		[Vercoustre et al., 2009]
TREC	Structured Retrieval	Document, passage, entity	[Fang et al., 2010]
TREC	Structured Retrieval	Body and title	[McCreadie et al., 2009]
TREC	BM25		[Zhai et al., 2009]
TREC	Language model		[Wu and Kashioka, 2009]
TREC	Google		[Wu and Kashioka, 2009]
TREC	Query construction	Relation: entity + relation	[Vydiswaran et al., 2009]
TREC	Language model, top 1000		[Zheng et al., 2009]
TREC	Query construction with passage retrieval		[Yang et al., 2009]
TREC	Query expansion	Query structure analysis	[Hold et al., 2010]

$d = (w_1, w_2, \dots, w_m)$, and a query, q , is also represented as a term vector (i.e. $q = (w_{1q}, w_{2q}, \dots, w_{tq})$). The relevance between the document d and the query q , therefore, is measured by the cosine similarity between these two vectors (i.e., $\text{sim}(d, q) = \cos \theta = \frac{d \times q}{|d| \times |q|}$). In the classic vector space model proposed by [Salton et al., 1975], the specific weights of terms in the document vector, $w_{t,d}$, are the products of term frequency (TF) and inverse document frequency (IDF). That is, $w_{t,d} = \text{tf}_{t,d} \times \log \frac{N_d}{|\{d \in D | t \in d\}|}$, where $\text{tf}_{t,d}$ is the frequency of a term t in a document d , and $\log \frac{N_d}{|\{d \in D | t \in d\}|}$ is the inverse document frequency with N_d for the total number of documents in the whole document set and $|\{d \in D | t \in d\}|$ for the number of documents containing the term t . Therefore, the final similarity function is $\text{sim}(d, q) = \cos \theta = \frac{d \times q}{|d| \times |q|} = \frac{\sum_i w_{i,d} \times w_{i,q}}{\sqrt{\sum_i w_{i,d}^2 \times \sum_i w_{i,q}^2}}$. More advanced weighting algorithms were created to improve the document-query similarity calculations, such as Okapi BM25. The BM25 method scores the similarity between a document and a query as $\text{sim}(d, q) = \sum_i \log \frac{N - |\{d \in D | t \in d\}| + 0.5}{|\{d \in D | t \in d\}| + 0.5} \times \frac{\text{tf}_{t,d} \times (k_1 + 1)}{\text{tf}_{t,d} + k_1(1 - b + b \frac{N_d}{\text{avg}(d)})}$ proposed by [Robertson et al., 1996], where N , the total number of documents in corpus, and b are constant parameters, and $\text{avg}(d)$ is the average document length in the corpus. Zhai used the BM25 method to identify the germane documents in his work [Zhai et al., 2009].

The advantages of this method are its simplicity and effectiveness, which is capable of computing a continuous degree of the similarity between queries and documents as well as ranking documents according to their relevance scores. But this method has the following limitations. First, long documents will require high-dimensional vector manipulations, which makes similarity measurements expensive to complete. Second, documents with similar contents but different vocabularies may come out with poor inner-product results. These are the limitations of keyword-driven IR systems in general since such systems cannot easily process semantic contents. The research in this study demonstrate that the deficiency of semantic analysis in the document search can be over come by answer entity extraction.

2.1.2 Language Model

Language model approaches to information retrieval are attractive and promising. The language model usually assumes that all terms t_1, t_2, \dots, t_n are independent in a document d .

That is, $p(d) = p(t_1 t_2 \dots t_n) = p(t_1) p(t_2) \dots p(t_n)$. This is a unigram language model. There are many other language models, such as the bigram language model or the trigram language model.

A basic language modeling approach for information retrieval, proposed by Ponte and Croft [Ponte and Croft, 1998], assumes the query is a sample of words drawn from a document according to a language model, i.e. the likelihood of $p(q|d)$. The maximum likelihood estimation is one of the solutions, i.e., $p(q|d) = \prod_t p_{mle}(t|d) = \prod_t \frac{tf_{t,d}}{N_d}$, where t is a term in the query q , and $tf_{t,d}$ is the occurrence of a term t in a document d , and N_d is the total number of terms in the document d . Thus, it is easy to estimate the terms appearing in the documents using the maximum likelihood. In order to estimate the probability of terms that do not appear in the document, called a smoothing problem, many approaches are discussed [Zhai and Lafferty, 2004]. One approach is Jelinek-Mercer smoothing, which is a linear interpolation method combining the maximum likelihood model and the collection model with a coefficient λ to control the influence of each term, i.e., $p(w|d) = (1 - \lambda)p_{mle}(w|d) + \lambda p(w|C)$. Another method is Bayesian smoothing, which uses Dirichlet priors as a prior for each word for smoothing, i.e., $p_\mu(w|d) = \frac{tf_{t,d} + \mu p(w|C)}{N_d}$ [MacKay and Peto, 1994].

Many approaches in entity retrieval use the language modeling for germane document identification. For example, the language modeling approach on structured documents (such as XML files) is applied to improve the search results. Two well-known examples of retrieving the structured texts or text elements are XML retrieval [Gövert and Kazai, 2002] and question answering [Prager et al., 2000]. Both retrieval tasks are based on the hierarchical structures of documents, but the only difference is that XML structures are introduced by authors while structures of question answering are introduced by annotators. Zhao [Zhao and Callan, 2008] summarized the structured retrieval as a generative retrieval model by expanding the basic keyword language model into a structure with hidden variables. Zhao and Callan introduced the fields (which are the snippets in a document) into their model, and emphasized that the major difference between document retrieval and field retrieval was that the surrounding context of a query term shrinks from a document to a field. Later, in the work of McCreadie, they applied the structured retrieval on HTML title and body parts [McCreadie et al., 2009]. The greater weight of the HTML title part than the HTML

body part indicated that they preferred title matches to body matches in germane document identification. Fang proposed a hierarchical relevance language model on three levels: document, passage, and entity [Fang et al., 2010]. The final document ranking score was a linear combination of the relevance scores from these three levels.

Language model approaches are applied not only in the structured document retrieval but also in category/type information for parameter smoothing or model enrichment, especially in INEX. This study does not distinguish categories and types either in entity types or in document categories but treats them equally as the classes for documents or entities. Jiang applied category information from queries into the query model and considers the similarity between queried category and the candidate entity category [Jiang et al., 2009]. They considered the probability of an entity e to be the target type q_cat , that is, $p(q_cat|e) = p(cat_q|cat_e) = \prod_{cat_j \in CAT_q} p(cat_j|CAT_q)$. The probability of the category similarity can also be viewed as Jelinek-Mercer smoothing for complementing the corpus smoothing. Similar approaches also appeared in [Vercoustre et al., 2009] and [Balog et al., 2010a]. This study will expand the language model and apply it on the document types and entity types to improve germane document identification.

2.1.3 Link Analyses

Link analyses apply in information retrieval for finding relevant documents by considering the authority pages and hub pages (such as the HITS algorithm) or measuring the relative importance of a page within a set (such as the PageRank algorithm). The HITS algorithm, developed by Jon Kleinberg, is an iterative algorithm based on two basic steps—the authority update and the hub update [Kleinberg, 1999]. The authority update gives a node its authority score by summing up the hub scores of each node pointing to it; and the hub update gives a node its hub score by summing up the authority scores of each node that it points to. The PageRank algorithm is also an iterative algorithm by adjusting approximate PageRank values to more closely reflect a page’s theoretical true value [Brin and Page, 1998]. Both the HITS algorithm and the PageRank algorithm use a coarse-grained model of the Web, which assumes each page is a node in a graph with a few scores associated with it.

The disadvantage of HITS is that it is sensitive to local topology. Moreover, both algorithms need to handle huge matrix manipulations. Vercoustre et al adapted the similar idea of the document's PageRank into a LinkRank module, which calculated a weight for a page based on the number of links to the entities on this page [Vercoustre et al., 2009]. The assumption is that a good entity page is the one that is referred to form contexts with many occurrences of the entity examples.

Although link analyses are successful in the document retrieval, they are more useful in mining the entities from the linked pages. Therefore, the link information in this thesis would be applied to extract the answer entities from the germane documents.

2.1.4 Topic Detection and Query Construction

With users' information needs, different methods are explored to express the topics as proper queries, such as topic difficulty predictions and query constructions.

The topic difficulty prediction in the research field of the XML entity ranking was studied in [Vercoustre et al., 2009]. Their work generated a topic classifier based on how well the runs submitted by participating systems could answer the topics. Each topic calculated the topic difficulty using the Average Average Precision (AAP) measure: the higher the AAP, the easier the topic. According to the AAP scores, the topics were grouped into two classes (Easy and Difficult). The features for classification were extracted from the topics and run-time results. They detected 32 features. Because some features were correlated, they removed the correlated features and kept nine features for the real-time classification task. These features included the number of sentences in the narrative, the ratio between the number of words in the title and the narrative, the ratio between the intersection and union of words in the title and the narrative, the ratio between the intersection and union of words in the description and the narrative, the ratio between the intersection of words in the title and description part and union of words in the title, description and the narrative part, the ratio between the intersection of words in the description and narrative part and union of words in the title, description and the narrative part, the number of pages in each target categories, the number of intersections of entity categories, and the ratio of the intersection

of entity categories and the union of the entity categories. Their system applies different parameters on the retrieval model according to the topic difficulties. Their results were among the top four best performing in INEX 2008.

Query construction is also applied in germane document identification. Vydiswaran modeled the information need of entity retrieval as a structured query [Vydiswaran et al., 2009]. They identified three parts of the queries: the relation (represented as descriptions of the topic entity and the answer entity), the entity of focus (the topic entity), and the entity of interest (the answer entity). For example, the query of all team-mates of Michael Schumacher could be addressed by two related relation queries: “[Michael Schumacher][drives for] [ORG-1]” and “[PER] [drives for] [ORG-1]”. Finding related entities through ORG-1 (the team), which is as yet unknown, would hopefully get filled in it by the term of “Ferrari” during execution. Once the relation predicates were identified, they augmented them with their synonyms from WordNet as well as similar words from a large text corpus on distributional similarity. The final query formulation with the primary entity, the relation word with its synonyms, optional noun phrases from the narrative, and the type of the desired entity, enforces retrieval system to match the primary entity and one of the relation words in all retrieved documents. Hold also preprocessed the queries using part-of-speech techniques to identify the source entity, target entity, and the relations between them. Then he used synonym dictionaries (exploiting Freebase sources) to find the alternative names for the source entities, and expand the queries [Hold et al., 2010]. The annotations for germane documents in the entity retrieval task also confirmed that different topics had different preferences in germane document identification. This study, therefore, investigates a better way to represent the users’ information needs with proper queries for germane document identification.

2.2 ENTITY EXTRACTION

Answer entity extraction in entity retrieval systems identifies answer entities from germane documents. It is usually simplified as entity extraction, which extracts targeted entities

without considering whether these entities answer the query. Entity extraction involves two subtasks: the identification of proper names in texts and the classification of these names into a set of predefined categories of interests. The entity extraction task was from the Sixth Message Understanding Conference (MUC-6) [Grishman and Sundheim, 1996]. The typical entity extraction task, for example, takes an un-annotated block of text, such as “Jim bought 300 shares of Acme Corp. in 2006,” to produce an annotated block of text, such as `<ENAMEX TYPE=“PERSON”>Jim</ENAMEX> bought<NUMEX TYPE=“QUANTITY”>300</NUMEX> shares of<ENAMEX TYPE=“ORGANIZATION”>Acme Corp.</ENAMEX> in<TIMEX TYPE=“DATE”>2006</TIMEX>`. With 15 years of research (from 1996), many approaches for entity extractions are investigated, which can be categorized into four classes: rule-based methods, supervised learning methods, semi-supervised learning methods, and unsupervised learning methods. Although the studies of named entity extraction have been going on for more than 15 years, the techniques used in the TREC entity retrieval are mainly limited in rules-based methods, which are summarized as in Table 3. This section reviews the methods used not only in the TREC answer entity extraction but also in the general domain of entity extraction.

2.2.1 Named Entity Recognition Tools

The tools of named entity recognition (NER) or entity identification or entity extraction locate and classify atomic elements in texts into predefined categories, such as the names of persons, locations and times. There are many commercial and free NER tools, such as Stanford NER [Finkel et al., 2005], UIUC NER [Ratinov and Roth, 2009], OpenNLP (community), GATE [Cunningham et al., 2002], LingPipe [Alias-i, 2008], OpenCalise [OpenCalaise, 2010], and Inxight [SAP, 2010]. The first three are the free tools, and the last two are the commercial products.

Zhai applied the Stanford NER tool on documents as well as topic descriptions to extract the candidate entities [Zhai et al., 2009]. The final answer entities were decided by filtering out the entities which did not belong to the targeted types and ranking them with a probabilistic model.

Table 3: Methods of answer entity extraction in TREC

Method	Reference
Extract from Tables and Lists	[Fang et al., 2009]
Dictionary-based named entity recognition (named entities from DEPedia)	[McCreadie et al., 2009]
NER tools to extract the target entities from the sentences including the candidate strings	[Zheng et al., 2009]
NER for document (UIUC) Entity re-ranking: similarity between support snippets of entities and input queries: supervised learning, clustering, and an algorithm	[Wu and Kashioka, 2009]
NER: rule; L2R	[Vydiswaran et al., 2009]
Segment with 50/100 words; NER + re-ranking	[Zheng et al., 2009]
NER with some terms from Wikipedia	[Yang et al., 2009]
Four rules for extracting the answer entities	[Hold et al., 2010]
NER from UIUC for candidate entities; and then re-ranking the entities with the similarity to the hyponym relations of the target entity categories	[Vercoustre et al., 2009]

Although Wu also used the NER tool from the Cognitive Computation Group at UIUC [Ratinov and Roth, 2009] to tag persons, organizations, and miscellaneous (as candidates for products), they applied a more complicated algorithm for entity filtering and ranking [Wu and Kashioka, 2009]. First, they gave every entity a score by considering the link between the entity and the query. If there is a hyperlink-to and a hyperlink-from between the entity and the query, the score for the entity is 2; if there is a hyperlink-to or a hyperlink-from between the entity and the query, the score for the entity is 1; if there is a hyperlink-to or a hyperlink-from between the entity and the terms in the query, the score for the entity is 0.5; otherwise, the entity score is 0. Then they chose the top 100 entities (with the descending order of the scores) with their original topics to find the support snippets in the search

engine. The final ranking of the entities was based on the similarities between the input query and support snippets of related entities. Vechtomova applied the NER tools for candidate entity recognitions, then used the similarity of the grammatical dependency between candidate entities and seed entities for re-ranking the candidate entities [Vechtomova, 2010].

Utilizing NER tools to extract target entities as answer entities is simple and straightforward. But it is limited by the entity recognition tool. If the entity type can not be identified by the tool, the system will fail at extraction them. The quality of the entity answers is limited by the tool. What is more important is that the extracted entities can not be guaranteed to answer the questions.

2.2.2 Rule-based Methods

One of the first research papers in the entity extraction field described a system relying on heuristics and handcrafted rules to extract and recognize company names [Rau, 1991]. Collins and Singer added language factors into entity extraction, that is, parsing a complete corpus in search of candidate named entity patterns for the entities of companies, persons, and locations [Collins and Singer, 1999]. The following sample rules are used: rule 1: if the spelling is “New York”, then it is a Location; rule 2: if the spelling contains “Mr.”, then it is a Person; rule 3: if the spelling is all capitalized, then it is an organization.

Fang extracted the entities not only from NERs, but also from the structured data embedded in natural language texts, such as tables, lists or other forms [Fang et al., 2009]. They extracted the attributes from the tables or lists using the rules: if the majority of the elements with the same attribute were the same type or identified as target entities, they treated all these elements as the target entities.

Instead of relying on rules to extract tables and lists, Hold set up four types of rules to identify the source, the target, the context, and the candidate respectively [Hold et al., 2010]. In that way, they identified the sentences with the same structures as queries in order to extract the candidate entities.

The dictionary-based entity extraction approach is a special case of rule-based methods. McCreadie specially used DBpedia as dictionaries [McCreadie et al., 2009] for extractions.

They built a large dictionary of entity names from DBpedia, a structured representation of Wikipedia. This dictionary also comprised all known aliases for each unique entity in DBpedia. For example, “Barack Obama” was represented by the dictionary entries of “Barack Obama” and “44th President of the United States”. They also assigned the entity types with the categories of DBpedia using some heuristics rules. For example, the occurrence of the clue word “company” was likely to be identified as organizations. Entries about people from DBpedia and common proper names derived from US Census data (Census) were used to produce the entities of persons.

Vydiswaran used both knowledge bases, such as gazetteers (derived from Wikipedia), and rules, such as regular expression patterns, to extract the location and product names respectively [Vydiswaran et al., 2009]. For example, in order to extract the product name, the rule of “finding capitalized phrases containing some numbers with length greater than two” was applied on the text of “the Nokia 6600 was one of the oldest models” for tagging “the Nokia 6600” as a potential product name. When multiple regular expression patterns defined to capture the entities of products were triggered in the extractions, the pattern generating the longer phrase was chosen to extract the entities.

The rule-based method is sometimes better than the NER detection approach because each entity type has its own vocabulary for the extraction task. But it still suffers from the following problems. It is limited by the knowledge from the knowledge base. If the knowledge base fails to store some entities, the whole system will not identify these entities in further extractions. Moreover, in order to extract a complete answer entity set, all the possible rules or patterns are required for extractions, but, in fact, it is hard to achieve this level of completeness.

2.2.3 Supervised Learning

Supervised learning is a machine learning technique predicating the output of testing data based on a function deducing from the training data. A small fixed text set is chosen and manually marked up in order to train the function. The trained function produces similar annotations on unseen texts. Therefore, it turns an extraction problem into a learning

problem and extracts the phrases with each argument and the types of entities.

Supervised named entity extraction is a decision on whether a given term in the document is a target entity. Either binary or multi-class classifications can be used. In a binary classification system, the positive samples are labeled as one class and the negative samples are labeled as the other classes. Final classification is performed by passing each instance to be labeled for all of the classifiers and then choosing the label from the classifier with the most confidence. There are many supervised learning based named entity extraction methods, for example, Hidden Markov Models (HMM) [Bikel et al., 1997], Decision Trees [Sekine, 1998], Maximum Entropy Models (ME) [Borthwick et al., 1998], Support Vector Machines (SVM) [Asahara and Matsumoto, 2003], and Conditional Random Fields (CRF) [McCallum and Li, 2003] and [Krishnan and Manning, 2006].

The advantage of supervised learning is its accuracy, but it requires an annotated training data set to learn the function for the entity extraction tasks, which is hard to obtain. For example, the task of extracting the named entity of products is difficult, not only because the innumerable data available on the Web that needs to be annotated, but also because the entity class itself is heterogeneous and, sometimes, mixes multiple classes (such as products).

2.2.4 Semi-supervised Learning

Semi-supervised learning, also called as lightly supervised approaches, or partially supervised learning, includes learning from Labeled and Unlabeled examples (LU learning) and learning from Positive and Unlabeled example learning (PU learning) [Liu, 2006]. Co-training is one of the PU learning methods. Avrim Blum and Tom Mitchell in 1998 introduced for co-training method as assuming each example was described with two different but complementary feature sets. [Blum and Mitchell, 1998]. Ideally, two feature sets, as two views, are conditionally independent; that is, two feature sets for each instance are conditionally independent given the class. Although each of the feature sets is sufficient for learning the target classification function, the co-training method tries to improve the classification by combining them together. The original co-training paper described experiments using co-training to classify whether web pages were “academic course home pages”. The classifier

could be built either on the text appearing on the page itself or on the anchor text attached to hyperlinks pointing to the page from other pages on the Web. Their experiment showed that the classifier categorized 95% of 788 web pages with only 12 labeled web pages as examples. The key part for this method is that it requires two different but complementary feature sets for the extraction.

Another important semi-supervised learning method is called bootstrapping. It either starts from a set of seeds and then generates the patterns for entity extraction, or starts from some patterns and then generates seeds to extract the entities. The idea of bootstrapping is that, with a small degree of supervision, the system starts the learning process, searches for sentences containing entities, and identifies contextual clues common to the samples. Then, the system finds other instances of the entities according to the contextual clues. With the learning process repeating these steps, more entities and more contexts will eventually be gathered. Brin did the pioneer work in 1998 to start with just a handful of seed tuples for the relation of interests, and automatically discovered extraction patterns for the relation extraction task [Brin, 1999]. These patterns, in turn, helped to discover new tuples for the relation, which could be used as new tuple for the next iteration of process. He also discussed the duality between patterns and seeds. Pasca investigated the same techniques inspired by mutual bootstrapping for the entity extraction task, but he innovated more general pattern generation by considering words as members of the same semantic class [Pasca et al., 2006].

Semi-supervised learning has been successfully applied in many extraction tasks. The advantage of this method is that it requires fewer training sets than supervised learning. This research investigates this semi-supervised learning method in answer entity extraction and evaluates whether it can improve entity retrieval task.

2.2.5 Unsupervised Learning

Rule-based entity extraction can be viewed as one type of unsupervised learning. For example, Hearst used rules (e.g., “city such as Paris”) to extract the entities (e.g., the city of Paris) [Hearst, 1992]. Clustering is another kind of important unsupervised learning methods. For example, one can try to gather named entities from clustered groups based on the

similarity of contexts.

The KNOWITALL system used an unsupervised method to extract named entities from the Web [Etzioni et al., 2005]. Inspired by Hearst’s work, the KNOWITALL system utilized eight domain-independent extraction patterns to generate candidate entities. A generic pattern “NP1 such as NPList2”, for example, indicates that the head of each simple noun phrase (NP) in the list NPList2 is a member of the class named in NP1. By instantiating the pattern for the class city, KNOWITALL extracted three candidate cities—Paris, London, and Berlin—from the sentence, “We provide tours to cities such as Paris, London, and Berlin.” Next, KNOWITALL automatically tested the plausibility of the candidate entities it extracted using point-wise mutual information (PMI) statistics computed by treating the Web as a massive corpus of texts. PMI-IR, developed by Turney, measures the dependence between two expressions using web queries [Turney, 2001]. A high PMI-IR means that expressions tend to co-occur. KNOWITALL leverages existing Web search engines to compute these statistics efficiently. In the initial run, they found that KNOWITALL was capable of autonomously extracting high quality information from the Web, so that they used three methods to improve systems’ recall. The first was pattern learning, which learned domain-specific patterns that served both as extraction rules and as validation patterns to assess the accuracy of instances extracted by the rules. The second was subclass extraction, which automatically identified subclasses in order to facilitate extraction. For example, in order to identify more scientists, it might be helpful to determine subclasses of scientists (e.g., physicists, geologists, etc.) and look for instances of these subclasses. The third method was list extraction, which located lists of class instances, learned a “wrapper” for each list, and used the wrapper to extract list elements. Each method dispensed with hand-labeled training examples by bootstrapping from the information extracted by KNOWITALL’s domain-independent patterns. The experiment indicated these three methods greatly improved the recall of the baseline KNOWITALL system, while keeping the high precision, which in turn improved the overall extraction rate.

The unsupervised learning approach has succeeded in the generic entity extraction process, but still needs to be further evaluated for the answer entity extraction task, since answer entity extraction is not only to find the specific entities with target types but also to find

the entities which should be able to answer the questions.

2.3 ENTITY RETRIEVAL IN INEX AND TREC

Research and development in the entity retrieval area is fueled by interests from the INitiative for the Evaluation of XML retrieval (INEX) [de Vries et al., 2007], since 2007, and the annual Text Retrieval Conferences (TREC) from the U.S. National Institute of Standards and Technology (NIST), since 2009 [Balog et al., 2009]. Both tasks require systems to list the answers about persons, products or locations with respects to a given query topic. Although two systems have similar entity retrieval tasks, there are differences between them. INEX focuses on the structured document retrieval, so that it is on the XML files (Wikipedia articles). Document borders should not play any role in this retrieval task. TREC focuses on the unstructured web-based document retrieval, so that the entity retrieval task in TREC is on the HTML sets. Another difference between these two tasks is that every page or document in Wikipedia collection is an entity page with human assigned categories. If we retrieve the Wikipedia collection, the relevant document will be the relevant entities, while the web page with the html format in the ClueWeb09 set cannot be viewed as entities. In this case, in TREC, before any entities can be ranked, they have to be recognized as entities and classified into the correct entity type.

This study investigates the entity retrieval on the Web, unstructured data, so that I choose the TREC task as the main experimental environment. The INEX task with the XML files, however, provides some special features which are useful for some experiments, for example, the retrieval models on evaluating the similarity between the entity types and the document categories.

3.0 A TWO-LAYER RETRIEVAL AND EXTRACTION PROBABILITY MODEL

This chapter gives a formal definition of the *Two-layer Retrieval and Extraction Probability Model* (TREPM), a generalized representation of the entity retrieval problem. This model combines germane document identification and answer entity extraction. The TREPM model is firstly published in the TREC 2010 entity retrieval task [Li and He, 2010].

3.1 OVERALL ARCHITECTURE

The entity retrieval task requires the system effectively and efficiently to return the answer entities from a large unstructured corpus with regard to users' information needs. The inputs of the system include documents (e.g., HTML pages or plain texts) and users' information needs (e.g., the description of a search task with a required entity type). The outputs are answers, ranked lists of entities. This study proposes a Two-layer Retrieval and Extraction Probability Model, short as TREPM, for the entity retrieval task. The TREPM model consists of two major components: germane document identification and answer entity extraction. This structure has been widely adopted in the entity retrieval (ER) systems in recent years.

The overall architecture of the TREPM model is as shown in Figure 3. The first layer is **germane document identification**, which aims at finding a small set of germane documents containing as many answer entities as possible in a short running time. The second layer is **answer entity extraction**, which tries to identify the entities from those germane documents by analyzing contexts. Therefore, the score for ranking answer entities combines

germane document relevance and answer entity relevance according to users' information needs. Furthermore, the output answer entities with their original queries can go back into the component of germane document identification in order to improve the detection of germane documents. For example, the answer entities are incorporated into the queries, and the queries are re-written in order to find the documents with multiple answer entities occurring in the same document, as showed in the link_1 in Figure 3. The output answer entities can go back to the component of answer entity extraction to extract answer entities. For example, the bootstrapping method uses the answer entities for extracting more patterns and further extracting more answer entities, as shown in the link_2 in Figure 3. Although this model contains two loops, this study only investigates germane document identification and answer entity extraction themselves, and leaves the relevant feedback and the pattern learning method for future discussions.

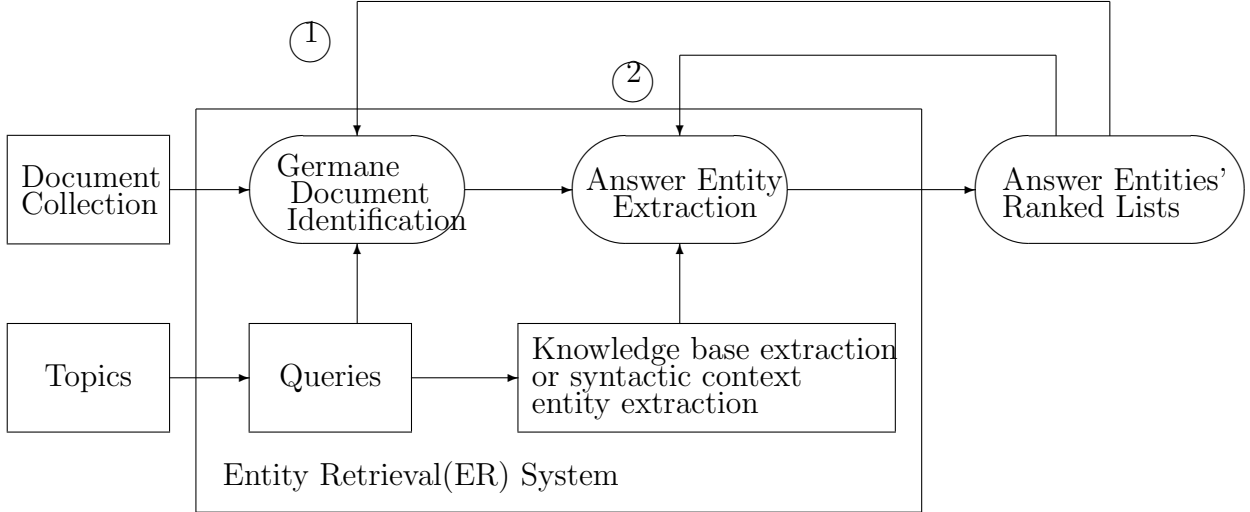


Figure 3: The Two-Layer Retrieval and Extraction Probability Model (TREPM)

Research on the TREPM model is driven by the following questions:

- What is the relation between germane document identification and answer entity extraction?
- How does each individual layer work in practice?
- How does each layer improve overall system performance?

3.2 A MODEL COMBINING DOCUMENT RETRIEVAL AND ENTITY EXTRACTION

An entity retrieval question can be stated as follows: does an entity e answer a query q with the targeted type t ? If we view this problem from the probability aspect, the question answers what is the probability of a candidate entity e being the answer entity given a query q with the target type t ? That is $p(e|q, t)$. The answer lists result entities according to their probabilities. The top k candidates are deemed to be the most probable entities.

Entities exist in the Web pages. Therefore, if we consider all germane documents d , the TREPM will be the following Equation 3.1. If we find the documents containing answer entities, called germane documents $d_{germane}$, to estimate this probability, then the original formula can be estimated as Equation 3.2.

$$\begin{aligned}
 & p(e|q, t) \\
 = & \sum_d p(e, d|q, t) \\
 = & \sum_d p(d|q, t)p(e|d, q, t) \tag{3.1} \\
 \approx & \sum_{d_{germane}} p(d_{germane}|q, t)p(e|d_{germane}, q, t) \tag{3.2}
 \end{aligned}$$

The TREPM model includes the following two parts. The first part of Equation 3.1 is $\sum_d p(d|q, t)$, where $p(d|q, t)$ is the probability of the document d generated by the query q with the target entity type t . This is **germane document identification**, conducted by estimating the similarity between a document and a query. The second part of Equation 3.1 is $p(e|d, q, t)$, i.e., the probability of entity e generation with the target type of t in the document d given query q , called **answer entity extraction**.

If we further consider contexts c for answer entities, answer entity extraction in the TREMP model will be expanded to the following Equation 3.3. The first quantity is $p(c|d, q, t)$, which is the generative probability of the context c in a given the particular document d with the query q and the target entity type t . The second quantity is $p(e|c, d, q, t)$, which is the probability of a candidate entity e to be an answer given a context c in the document d for the query q with the target entity type t . Similar to germane document

identification, if we use the most high probability contexts, called support contexts $c_{support}$, to extract the answer entities, then we have the estimation formula as Equation 3.4.

$$\begin{aligned}
& p(e|q, t) \\
= & \sum_d p(d|q, t)p(e|d, q, t) \\
= & \sum_d p(d|q, t) \sum_c p(c|d, q, t)p(e|c, d, q, t) \tag{3.3} \\
\approx & \sum_{d_{germane}} p(d_{germane}|q, t) \left(\sum_{c_{support}} (p(c_{support}|d_{germane}, q, t)p(e|c_{support}, d_{germane}, q, t)) \right) \tag{3.4}
\end{aligned}$$

There are two reasons for decomposing the entity retrieval problem into germane document identification and answer entity extraction. The first reason is that the decomposition can divide the word-independent factor and the word-dependent factor into two subtasks. The information need of entity retrieval (e.g., what are the products of MedImmune, Inc.) represents the answer entity (e.g., the company’s product) as a description (e.g., products of MedImmune, Inc.) expressing the relation between the topic entity (e.g., MedImmune, Inc.) and the answer entities (e.g, FluMist). The word-independent factor assumes that the words occur in the documents independently, while the word-dependent factor assumes that the meaning of words influences the interpretation of other words in documents. For example, in the word independence assumption, we assume the above query is to find the documents containing the terms of “Products”, “of”, “MedImmune”, and “Inc.” With this assumption, the document can be the one either containing “products of MedImmune Inc” or “MedImmune Inc. buys the computer products from ...” The document retrieval model can provide a good and effective way to retrieve the information, according to the word-independence assumption, in document-level relevancy. It is, to certain degree, to estimate whether the document contains the answers for the query or not. In the word-dependent assumption, the semantic meaning within a document is analyzed to extract the answer entities for the query. For example, we need to treat MedImmune, Inc. as an entity of company, and find the product of MedImmune, Inc from the document. Entity extraction can be a powerful approach for this task.

The second reason for this decomposition is to simplify a globe retrieval problem into two locally optimized problems, which will lower the complexity of the problem and reduce the

execution time. If we assume the number of documents in the corpus is m and the number of contexts is n , then the time requirement for the entity retrieval task is $\Theta(m * n)$, because the system needs to iterate every document and scan all contexts to detect answer entities. If we use document retrieval to find germane documents with the number of m' ($m' \ll m$) and only consider the most effective contexts with the number of n' ($n' \ll n$), then the time requirement for the TREPM to complete the entity retrieval task is $\Theta(m' * n')$, which will significantly reduce the system execute time. The space complexity is similar. Mark Bron and his entity retrieval group formula the entity retrieval task as $p(e|E, T, R)$, i.e., the probability of candidate entities, e , given the source entity, E , the target type, T , and the relation, R , described in the narrative. They calculate the co-occurrence of candidate entities e and source entities E for all documents in the corpus. According to their answer for how long does it take to process the whole corpus, it is around two weeks [Bron et al., 2010]. If they consider germane documents for the candidate entities, this process time will be significantly shorter.

In summary, TREPM considers the relevance between entities and topics on two layers: germane document identification and answer entity extraction. In order to search answer entities, a retrieval system needs to rank all candidate entities by considering all combinations of documents and contexts. In a large-scale information environment or open-ended corpus, such as the Web, however, evaluating all documents and all patterns is an impossible task. Therefore, we find germane documents and support contexts, instead of all documents and all contexts, effectively and efficiently to estimate answer entities.

3.3 GERMANE DOCUMENT IDENTIFICATION IN TREPM

The main role of germane document identification is to retrieve a very small but highly useful subset of documents from the entire collection for further answer entity extraction. The general process of entity retrieval iterates all documents in a corpus, i.e., $\sum_d p(d|q, t)$. In practice, however, it is impossible to parse the whole collection because of the huge data set. For example, the INEX 2007 corpus included about 0.5 million Wikipedia pages, and

the INEX 2009 corpus contained about 2.5 million Wikipedia pages. The number of web pages in the TREC 2009 testing corpus was 50 million (200 Gigabytes) and the number of web page in the 2010 corpus was 500 million (2 Terabytes). Therefore, we consider germane documents to estimate the retrieval as shown in Equation 3.5.

$$\sum_d p(d|q, t) \approx \sum_{d_{germane}} p(d_{germane}|q, t) \quad (3.5)$$

Germane document identification is different from the conventional document retrieval task. On the one hand, germane documents mean the documents containing answer entities for the extraction instead of the ones most relevant to the topics. For example, if we treat the topic of products of MedImmune, Inc as a document retrieval problem, the expected answer lists might be ranked, in the decreasing relevant scores, as follows: <http://www.ethyol.com/>, <http://www.flumist.com/>, and http://www.medimmune.com/about_us_products.aspx, because we expect the pages directly answering the query are ranked higher than the pages with miscellaneous information. Germane document identification, however, expects to reverse the ranked list because germane documents are supposed to contain as many answer entities as possible. On the other hand, we expect the germane document sets as small as possible. In the above example, if we can find the answers in one document, such as http://www.medimmune.com/about_us_products.aspx, then we do not need to process the other two documents. Therefore, I use “germane documents” instead of “relevant documents” to distinguish document retrieval and germane document identification.

Germane documents are also distinct from the homepages of answer entities. TREC or INEX defines the entity retrieval task as finding the homepages of answer entities as results. Therefore, both of them are in forms of URLs/URIs. However, we should notice that they are different. For example, for the topic of products of MedImmune, Inc., one of the answer entities is Ethyol and its homepage of Ethyol is <http://www.ethyol.com/>. This homepage includes a sentence like “ETHYOL is a registered trademark held by MedImmune, LLC, a member of the AstraZeneca group of companies”, which indicates Ethyol is the product of MedImmune, Inc., so that this web page is also the germane document for this topic. This case indicates URLs/URIs of germane documents and the homepages of answer entities can be the same one. They can, however, be different. For example, the answers for the topic of

What countries does Eurail operate in are Austria, Italy, Germany, etc. The homepages for these entities are such as <http://www.germany-tourism.de/> and <http://www.france.com/>, but a germane document for this query would be the web page about how Eurail is introduced and the countries it passes through, such as <http://www.eurail.com/eurail-global-pass?currency=eur>.

Germane document identification approximates the retrieval process. There are many ways to interpret germane document identification. Firstly, if we assume the entity type t is independent from the query q and the document d , then we have the following Equation 3.6. This is the conventional document retrieval ignoring the target entity types.

$$p(d|q, t) = p(d|q) \quad (3.6)$$

In INEX, people assume one document represents one entity, and each document is assigned to some categories because every document is an entry of Wikipedia which has already been categorized. The similar document structure also exists in the Web environment. For example, people like to assign some tags to blog pages when they browse them. If the categories/types of documents $category$ are considered, the first quantity of Equation 3.1 can be revised as Equation 3.7. The first part of this formula is the generated probability of the type given a query and a target type. In fact, it is to calculate the similarity between two types/categories. The second part of the formula calculates the document similarity according to the query, target entity types, and document categories.

$$\begin{aligned} p(d|q, t) &= \sum_{category} p(d, category|q, t) \\ &\approx \sum_{category_{support}} p(category_{support}|q, t)p(d|q, category_{support}, t) \end{aligned} \quad (3.7)$$

With trained data sets, we can treat germane document identification as a learning to rank problem. The condition probability in germane document identification is transferred to the joint probability, since the query and entity type probability are equal for the same query.

$$p(d|q, t) = \frac{p(d, q, t)}{p(q, t)} \propto p(d, q, t) \quad (3.8)$$

The idea for the learning to rank method is that we learn the weights for all features in the model from the germane document training sets, and then use these weights to estimate the joint probability. This method not only uses the terms in the document for the estimation but also extracts the features from the documents and the queries. Therefore, it can represent as a function of feature sets x_1, x_2, \dots, x_n .

$$p(y = 1|d, q, t) = f(x_1, x_2, \dots, x_n) \quad (3.9)$$

3.4 ANSWER ENTITY EXTRACTION IN TREPM

Answer entity extraction in the TREMP model is to extract the answer entities from the germane documents. It estimates the probability of the entity e to be the answer entity, given the document d , the query q , and the target entity t , i.e., $p(e|d, q, t)$.

This study uses answer entity extraction instead of entity extraction because these two tasks are slightly different. Entity extraction locates atomic elements in texts and classifies them into predefined categories, such as the names of persons, organizations, and locations. Answer entity extraction, however, is to extract the atomic elements not only according to the predefined categories but also answering the queries. Therefore, we use the term of “answer” to emphasis the requirements.

Answer entity extraction can be treated as a entity extraction task on the germane documents, i.e., $p(e|d, q, t)$. Named entity recognizer is used to extract the entities t directly from the document d with the special target entity type t . This case assumes the query q is independent of document e , entity type t , and entity e .

If we consider the contexts c for answer entity extraction, it is useful to analyze whether the entities can answer the queries. We expect to extract the answer entities with high accuracy, as Equation 3.10. If only the most effective support contexts are considered,

$c_{support}$, the second part of the TREPM model will be represented as follows (Equation 3.11).

$$p(e|d, q, t) = \sum_c p(c|d, q, t)p(e|c, d, q, t) \quad (3.10)$$

$$\approx \sum_{c_{support}} p(c_{support}|d, q, t)p(e|c_{support}, d, q, t) \quad (3.11)$$

The contexts can be interpreted into several ways. In the medical domain named entity extraction, the context means the negation, experiencer, and temporal status for the medical findings [Harkema et al., 2009]. The context is also be interpreted as the term co-occurrence in certain window sizes, such as the studies in the word sense identification [Leacock and Chodorow, 1998]. This study focuses on two kinds of contexts—symbolic contexts and syntax contexts.

Symbolic contexts use symbols, *symbol*, to show the nature of contexts. For example, in the Wikipedia homepage, the terms of “InfoBox, product” shows a symbolic context indicating products for a company. It can be represented as Equation 3.12.

$$p(e|d, q, t) = \sum_{symbol} p(symbol|d, q, t)p(e|symbol, d, q, t) \\ \approx \sum_{symbol_{support}} p(symbol_{support}|d, q, t)p(e|symbol_{support}, d, q, t) \quad (3.12)$$

Syntax contexts are culled from the sentence syntax with deep sentence analyses. For example, the syntactic analysis of the sentence “ETHYOL is a registered trademark held by MedImmune, LLC” shows the context of the product (ethyol) and the company (MedImmune, LLC). If every query is represented as a binary relation r_q between the topic entity e_{q1} and the target entity e_{q2} , the content can be represented as the triplet of an entity e_1 with the type t_1 , an entity e_2 with the type t_2 , and their relations r .

This study considers the subject-verb-object structure as syntax contexts for the extraction. The entity extraction in this syntax contexts can be represented as follows, Equation

3.13:

$$\begin{aligned}
& \sum_c p(c|d, q, t)p(e|c, d, q, t) \\
= & \sum_{e_1, t_1, r, e_2, t_2} p(e_1, t_1, r, e_2, t_2|d, r_q, e_{q1}, t_{q1}, t_{q2})p(e_{q2}|e_1, t_1, r, e_2, t_2, d, r_q, e_{q1}, t_{q1}, t_{q2}) \\
\approx & \sum_{e_1=e_{q1}, t_1=t_{q1}, r=r_q, e_2, t_2=t_{q2}} p(e_1, t_1, r, e_2, t_2|d, e_1, t_1, r, e_2, t_2)p(e_{q2}|e_1, t_1, r, e_2, t_2, d) \quad (3.13)
\end{aligned}$$

The first quantifier in Equation 3.13 reflects the association between the query, represented as $r_q, e_{q1}, t_{q1}, t_{q2}$, and the context or the support relation triplet pattern, represented as e_1, t_1, r, e_2, t_2 , in the document. The second quantifier in Equation 3.13 reflects how the component extracts the answer entities from the context, which is to extract the entities e_{q2} when the system matches the relations r_q and r as well as the topic entities e_{q1} and e_1 with the type of t_1 . For example, for the topic of products of MedImmune Inc., the query can be represented as a relation of *relation_surface* (e.g., products), *entity₁* (e.g., MedImmune Inc.), *type₁* (e.g., Company), *entity₂* (e.g., “query”), *type₂* (e.g., Products), as follows:

$$\begin{aligned}
& \textit{Relation}(\textit{Relation_Surface}, \textit{Entity}_1, \textit{Type}_1, \textit{Entity}_2, \textit{Type}_2) \\
& \textit{Query} = \textit{Relation}(\textit{Products_of}, \textit{Apple}, \textit{Company}, ?, \textit{Products})
\end{aligned}$$

Pattern retrieval finds the most relevant patterns, given the query and target entity type. This study only considers binary relations as semantic patterns. For example, the sentence of “Apple launches iPad” can be represented as product_of with relation indicators of “launches”. Therefore, the relations can be represented as (launches, Apple, Company, iPad, Products). Considering the above example, possible answer sentences can be represented as follows:

$$\begin{aligned}
& \text{“Apple launches iPad”} \rightarrow \textit{Relation_instance}=(\textit{launch}, \textit{Apple}, \textit{Company}, \textit{iPad}, \textit{Products}) \\
& \text{“Apple launches iPhone.”} \rightarrow \textit{Relation_instance}=(\textit{launch}, \textit{Apple}, \textit{Company}, \textit{iPhone}, \textit{Products}) \\
& \text{“Apple to produce new Verizon-friendly iPhone.”} \rightarrow \\
& \quad \textit{Relation_instance}=(\textit{to_produce}, \textit{Apple}, \textit{Company}, \textit{iPhone}, \textit{Products})
\end{aligned}$$

The contexts can not only be the symbol context or the syntax context but also be a combination of multiple contexts. With a trained data set, the system can learn a model and then use the model to estimate the further extractions, i.e.,

$$\begin{aligned}
p(e|d, q, t) &= \sum_c p(c|d, q, t)p(e|c, d, q, t) \\
&= \alpha \sum_{c_{symbol}} p(c_{symbol}|d, q, t)p(e|c_{symbol}, d, q, t) \\
&+ \beta \sum_{c_{syntax}} p(c_{syntax}|d, q, t)p(e|c_{syntax}, d, q, t) \\
&+ (1 - \alpha - \beta) \sum_{c_{other}} p(c_{other}|d, q, t)p(e|c_{other}, d, q, t)
\end{aligned}$$

3.5 SUMMARY

This chapter introduces the Two-layer Retrieval and Extraction Probability Model (TREPM) for the entity retrieval task. We theoretically demonstrate that the entity retrieval task can be divided into two subtasks—germane document identification and answer entity extraction—with the TREPM model. The target of the TREPM model is to efficiently and effectively find the answer entities from a huge corpus according to the queries.

Germane document identification is the process of identifying find a small set of documents containing the answer entities. In TREPM model, it is to estimate the probability of a germane document with regard to the query and the target entity type. Answer entity extraction reflects the confidence of an entity to be the target entity given the corresponding evidence, which is the relevance score at the entity level. The entity relevant score is calculated by summing up all the entity instances. In order to retrieve the target entities, this retrieval system ranks all the candidate entities by comparing the combination scores from the most germane documents and the most high-related candidate contexts for the answer entities.

4.0 EXPERIMENTAL METHODOLOGY

This chapter introduces experiment questions, data collections, evaluation metrics, and the tools used in Natural Language Processing (NLP) as well as Information Retrieval (IR). Subsequent chapters detailing experiments will be referred to the descriptions here.

4.1 EXPERIMENTAL QUESTIONS AND EVALUATION FRAMEWORK

As mentioned in the research goals of Section 1.2, this thesis proposes a Two-layer Retrieval and Extraction Probability Model (TREPM) for the entity retrieval task, which decomposes entity retrieval into germane document identification and answer entity extraction. Chapter 3 demonstrates the decomposition process from a theoretical perspective and describes the relations between germane document identification and answer entity extraction. Chapter 3 also clarifies the difference between germane document identification and the conventional document retrieval task as well as the difference between germane document identification and entity retrieval. Answer entity extraction deals with the word-dependent factors and identifies the answer entities in two kinds of contexts—symbolic contexts and syntax contexts. With the theoretical demonstration of the TREPM model, it is important to further evaluate behaviors of each individual layer in the model. With the germane document identification, it has the following questions:

- What methods can be used for germane document identification?
- What factors affect germane document identification?

With germane documents, answer entity extraction further discusses:

- What methods can be used for answer entity extraction?
- What factors affects answer entity extraction?

As part of the development of TREPM, the evaluation framework is as shown in Figure 4. In this evaluation framework, germane document identification (Chapter 5) and answer entity extraction (Chapter 6) are evaluated respectively.

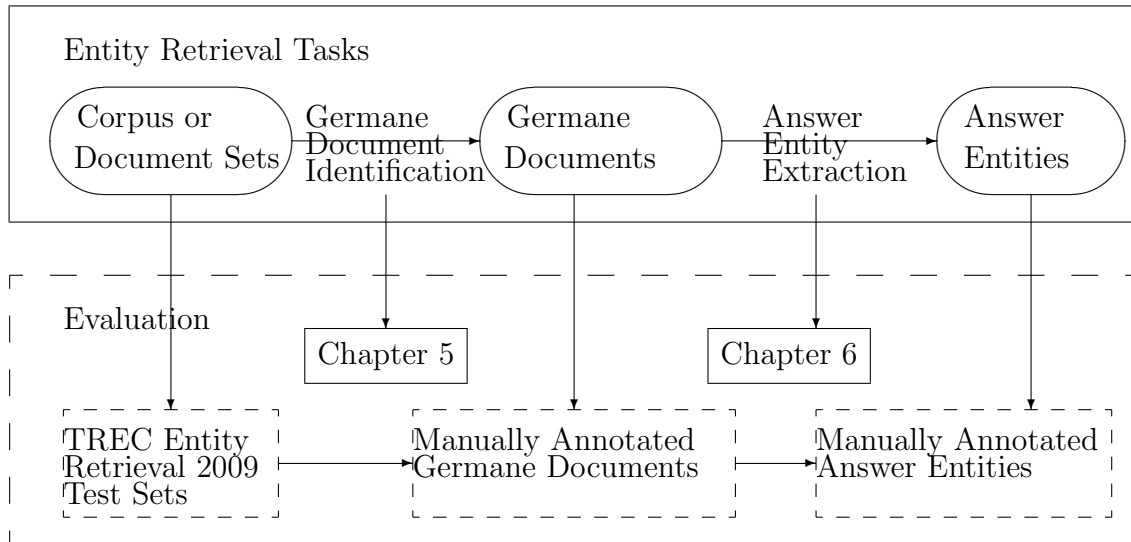


Figure 4: The evaluation framework of TREMP: germane document identification and answer entity extraction

Germane document identification is the process of identifying a small set of germane documents which contains answer entities, so that the evaluation of this component is to test the performance of different approaches in detecting germane documents. The test set is the TREC 2009 data set, which includes the CLUEWEB09B dataset, 20 topics, and the annotated ground truth of germane documents for each topic. The details are in Chapter 5.

Answer entity extraction detects answer entities from germane documents, so the evaluation of this component is testing the methods for extracting the answer entities from germane documents. The test set is the TREC 2009 data set, which includes the CLUEWEB09B dataset, 20 topics, and the annotated answer entities for each topic. The details are in Chapter 6.

The TREC or INEX entity retrieval task requires the entities' URLs/URIs as answers because they intend to use the URLs/URIs to represent the different citations of the same entities. Some work investigates the methods of the URL/URI finding given the answer entities, as in Appendix A. However, this study defines the entity retrieval task as finding the related answer entities, so that we treat the answer entity URL/URI finding as an additional requirement for some special tasks (e.g. TREC or INEX). In the further discussion, we only focus on the entity retrieval task itself, assuming the entity retrieval task requires the entity lists as answers.

4.2 TEST COLLECTIONS

The test collections, summarized in Table 4, are the basis for the following experiments for the TREPm model. Each testing set consists of a text corpus, a set of topics, and a corresponding answer set with relevance judgments. Three collections are used in this dissertation: INEX test set, TREC test set, and RAP test set. The INEX test set contains the semantic structured documents in XML formats with Wikipedia articles used to explore the methods for germane document identification; the RAP collection consists of plain texts from news and Wikipedia documents for exploring entity extraction strategies; and the TREC collection from the unstructured web pages (html pages) evaluates the whole TREPm model as well as its individual layers.

4.2.1 INEX Entity Ranking Track 2007/2008

INEX, which stands for INitiative for the Evaluation of XML retrieval, focuses on the structured document retrieval. It began the comparative evaluations in XML retrieval in 2002 [Gövert and Kazai, 2002] and began entity ranking track in 2007. The test collection are the Wikipedia XML format files. The entity retrieval task in INEX consists of 40 questions in 2007 and 60 questions in 2008. The task of entity ranking defined in INEX 2007 requires to return entities that satisfy a topic described in natural language text [de Vries et al., 2007].

Table 4: The testing sets

	INEX		TREC		RAP
	2007	2008	2009	2010	2008
Name/Notes	Wikipedia 2008 XML data	Wikipedia 2009 data annotated with 2008 w40-2 version of YAGO	Clueweb09b	Clueweb09	Company- related news and Wikipedia articles
Website	Wikipedia 2008 data	Wikipedia 2009 data	ClueWeb09		
# of Documents	659,387	2,666,190	50,220,423	1,040,809,705	530 Wikipedia arti- cles & 20 news ar- ticles
Corpus Size	10G, uncom- pressed	50G, uncom- pressed	200 GB	5TB, com- pressed	1MB
Document Format	XML		HTML pages		Plain Text
# of Top- ics	40	60	20	50	2 relations
Tasks	Entity retrieval		Entity retrieval		Entity extraction
Ground truth	Topics with entities answer sets, and relevance with 3 scales		Topics with entity homepage answer sets, relevance with 3 scales		Company-Product and company- location annota- tions

Given preferred categories/types, relevant entities are assumed to loosely correspond to Wikipedia pages that are labeled with preferred categories/types (or perhaps sub-categories

of these preferred categories). The sample topic is like the following Figure 5.

```
<inex_topic topic_id= "9999" ct_no= "0">
<title>European countries where I can pay with Euros</title>
<description>I want a list of European countries where I can pay with Euros.</description>
<narrative>
Each answer should be the article about a specific European country that uses the Euro as currency.
</narrative>
</categories><category id= "10855" >art museums and galleries</category><categories>
</inex_topic>
```

Figure 5: A sample topic of INEX entity retrieval task

The INEX 2007 test set used in the entity track consists of 659,387 2008 Wikipedia pages, and the 2008 test set includes 2,666,190 2009 Wikipedia pages. As a retrieval test collection, the organizers of the track provided relevant answer documents judged by experts. The same procedures and definitions for entities were used to compile a set of instance-level judgments from the document-level relevance judgments. The relevancy in the INEX corpus is divided into three scales: highly relevant, relevant, and not relevant.

4.2.2 TREC Entity Track 2009/2010

Text REtrieval Conferences (TRECs) is an unstructured document counterpart to INEX. It is held by the U.S. National Institute of Standards and Technology (NIST) as comparative evaluations in retrieval since 1992 [Balog et al., 2010b]. From 2009, they began Entity Retrieval Track as part of their annual competitions. The test collection used in 2009 Entity Retrieval task is limited in three types of target entities: persons, organizations, and products. Figure 6 is an example topic, finding the Airlines (as a type of organization) that currently use Boeing 747 planes. The sample answers to this topic are the organizations like British Airways, Cathay Pacific, Japan Airlines or Korea Air. The complete twenty topics of the TREC 2009 topic sets are in the Appendix E.

The text collection used in the TREC 2009 Entity Track is the Clueweb09B corpus. The corpus consists of 50,220,423 English website pages (approximately 200 GB). The TREC 2010 Entity Track uses the whole English part of Clueweb09 corpus with 2 Terabytes data.


```
<query>
<num>7</num>
<entity_name'>Boeing 747</entity_name>
<entity_URL>clueweb09-en0005-75-02292< /entity_URL>
<target_entity>Organization</target_entity>
<narrative>Airlines that currently use Boeing 747 planes.</narrative>
</query>
```

Figure 6: A sample topic of TREC entity retrieval task

Like the INEX entity track, the relevancy levels of TREC entity track are also the same three scales: highly relevant, relevant, and not relevant.

In order to evaluate the TREPM model, two annotators manually marked up ground truth sets of germane documents and answer entities for the TREC 2009 entity track data set. The annotation requirement for the germane documents is to find at least one germane document which can provide the answers for each topic. If there were corresponding Wikipedia articles existing, they were required to be marked up. The steps for germane document annotations are as follows: first, annotators generate proper queries, and retrieve them on a search engine to find the possible germane documents; secondly, according to the ranked hits from the search engine, two annotators evaluate whether these hits are germane documents. Every topic must find at least one germane document; and if there are more than 10 germane documents found, annotators only judge the first 10 hits.

The annotation requirement for answer entities is to find the answers for each topic from the germane document sets. Entities can be in various surface forms and we keep their original forms without merging them into a standard format. For example, the answer entities for the topic of the campus of Indiana University can be Indiana University East or IU East, which are both seen in the germane documents. The annotators marked all of them as answers without differentiations.

The overall annotations of the TREC 2009 data are summarized in Table 5. This test set covers three types of entities: organizations, persons, and products. The average number of germane documents for each topic is 1.75 and the average number of Wikipedia germane documents for each topic is 0.9. That means we can find the germane documents from

Table 5: The annotation summary of 20 topics in the TREC 2009 data sets

Topic ID	Entity Type	# of Germane Docs	# of Germane Wikipedia Docs	# of Answer Entities	# of Homepages
1	Organization	1	1	4	10
2	Person	1	0	12	1
3	Person	1	0	0	1
4	Organization	3	1	8	5
5	Product	3	1	3	8
6	Organization	2	1	4	4
7	Organization	1	1	4	33
8	Product	2	1	41	41
9	Person	2	1	9	13
10	Organization	2	1	9	2
11	Organization	2	0	8	8
12	Organization	2	1	32	13
13	Product	2	1	2	4
14	Person	1	1	36	4
15	Organization	2	1	12	9
16	Organization	1	1	11	9
17	Person	1	1	61	18
18	Person	2	1	14	3
19	Organization	3	2	4	3
20	Organization	2	1	27	4
Total	3	35	18	301	193
Average		1.75	0.9	15.05	9.65

Wikipedia for most topics and the other parts come from the Web pages. The average number of answer entities for each topic is 15. There are 11 topics with answer entities less than 10; there are 5 topics with answer entities more than 20. The distribution of answer entities is skew.

4.2.3 RAP Collection

RAP (Relation Annotation Platform) collections are used to evaluate the entity extraction by analyzing the relations involving two entities. This collection first appeared in [Li et al., 2009] and mainly consists of topic-answer entity pairs for company-product relations and company-location relations. The ground truth of RAP is manually marked up by two experts with two levels of relevancy: relevant and non-relevant.

Twenty-five target companies are chosen for experiments as short documents testing sets. These companies are large companies listed in Fortune-500 2008. The reason to choose these companies for the experiment is that it is easier for training and testing the article collection. The distribution of companies over the industries is also considered, and five industries are chosen. They are the industries of computer software (Microsoft and Oracle), computer office equipment (HP, DELL, and Apple), Internet services and retailing (Google, Amazon.com, Liberty Media, eBay and Yahoo), petroleum refining (Exxon Mobile, Chevron, Hess, Tesoro and Western Refining), and the telecommunication (AT&T). The details are shown in Appendix B.

Wikipedia articles for these 25 companies are used as long documents testing set (the average length is 16,700 characters), eighty-eight articles from CNET are chosen as short articles for testing set (the average length is 7,278 characters). The difference between the length of Wikipedia articles and news articles comes from the character of the articles. Wikipedia articles about a company are the descriptive articles with overall company information, but news articles about companies usually are connected with product announces or revenue information.

4.3 EVALUATION METRICS

Three metrics—precision, recall, and F-measure—are used to evaluate the performance of germane document identification and answer entity extraction. Both INEX and TREC tasks assume the relevancy of the answer entities with their URIs/URLs are in three levels (non-relevant, relevant, and highly relevant). In the competition, they use nDCG as a major metric to evaluate the overall system performance. This thesis, however, uses the entities as answers, and the relevancy judgment is on two levels, so we do not use nDCG. In order to evaluate the model, each individual component in the TREPM model is also evaluated. These sub-tasks only use two-level relevancy (relevant and non-relevant). Therefore, only precision, recall and F-measure are used. The same approach goes for RAP tasks.

Precision in document retrieval is defined as the number of relevant documents found by a search system divided by the total number of documents found by that search system. In this study, germane document identification uses the same definition as above. In answer entity extraction, the precision definition for answer entities is the percentage of extracted answer entities to the all extracted entities. In entity retrieval, precisions are defined based on the units of entities, i.e., the percentage of retrieved answer entities in all retrieved entities. Here are the definitions of precision in three tasks, seeing Equation 4.1.

$$\begin{aligned}
 precision_{Germane_Document_Identification} &= \frac{|\{Germane_Documents\} \cap |\{Retrieved_Documents\}|}{|\{Retrieved_Documents\}|} & (4.1) \\
 precision_{Answer_Entity_Extraction} &= \frac{|\{Answer_Entities\} \cap |\{Extracted_Entities\}|}{|\{Extracted_Entities\}|} \\
 precision_{Entity_Retrieval} &= \frac{|\{Answers_Entities\} \cap |\{Retrieved_Entities\}|}{|\{Retrieved_Entities\}|}
 \end{aligned}$$

Recall in germane document identification is defined as the number of germane documents retrieved by a search divided by the total number of existing germane documents. In answer entity extraction, the recall is defined as the fraction of the identified answer entities by the algorithm over all answer entities in ground truth sets. In entity retrieval evaluation, the recall is the number of answer entities retrieval by a search system divided by the total number of existing answer entities, as shown in Equation 4.2.

$$\begin{aligned}
recall_{Germane_Document_Identification} &= \frac{|\{Germane_Documents\} \cap |\{Retrieved_Documents\}|}{|\{Germane_Documents\}|} \quad (4.2) \\
recall_{Answer_Entity_Extraction} &= \frac{|\{Answer_Entities\} \cap |\{Extracted_Entities\}|}{|\{Answer_Entities\}|} \\
recall_{Entity_Retrieval} &= \frac{|\{Answer_Entities\} \cap |\{Retrieved_Entities\}|}{|\{Answer_Entities\}|}
\end{aligned}$$

F-measure is a measure of a test’s accuracy. It considers both the precision p (Equation 4.1) and the recall r (Equation 4.2). The F-measure score can be interpreted as a weighted average of the precision and recall, where an F score reaches its best value at 1 and worst score at 0, as shown in Equation 4.3.

$$\begin{aligned}
F_{Germane_Document_Identification} &= \frac{2}{\frac{1}{Precision_{Germane_Document_Identification}} + \frac{1}{Recall_{Germane_Document_Identification}}} \quad (4.3) \\
F_{Answer_Entity_Extraction} &= \frac{2}{\frac{1}{Precision_{Answer_Entity_Extraction}} + \frac{1}{Recall_{Answer_Entity_Extraction}}} \\
F_{Entity_Retrieval} &= \frac{2}{\frac{1}{Precision_{Entity_Retrieval}} + \frac{1}{Recall_{Entity_Retrieval}}}
\end{aligned}$$

4.4 INFORMATION RETRIEVAL TOOLS

The IR tools described in this section form the foundation of germane document identification in this thesis. The tools include the commercial search engine APIs providing the accessing point for user queries, and the indexing and searching tools, allowing users to build their own indexing systems on their own data set.

The **Indri** is the latest search engine using Language Model and Vector Space Model for information retrieval in the open-source Lemur toolkit [Strohman et al., 2005]. The Indri retrieval model is a combination of the language modeling approach and the inference network model. It supports rich structured queries based on the inference network model (for example

InQuery) and the probabilities are estimated using language modeling or the Okapi ranking function. The Indri indexes structures of the documents and the retrieval model of Indri directly supports the fields or concepts which are typed extents defined over a contiguous sequence of tokens. Fields can enclose each other and overlap each other arbitrarily. Fields are commonly used as a means for separating distinct document representation. Examples of using fields indexing include title and body fields indexing for web or Wikipedia documents. In the INEX 2007-2008 entity ranking experiments, entity types annotated by the corpus (such as links, person, etc.) would be fields in the indexing. For ClueWeb09B collection used in TREC 2009 entity ranking experiments, field types derived from NLP analysis including named entity types as well as target verbs and arguments such as arg0 (subject) and arg1 (objects) are indexed.

Search engine API is a particular set of rules and specifications provided by some commercial or non-commercial search engine companies for software programs to communicate with each other. Yahoo!BOSS (Build your Own Search Service) is Yahoo!'s open search web services platform. Yahoo!BOSS is simple for the developers, to foster innovation in the search industry, and also it is easy to build and launch a web-scale search that utilizes the entire Yahoo! Search index. It provides a way to search the relevant documents without considering crawling and indexing, ranking and relevancy algorithms, and powerful infrastructures. Some similar search engine APIs also include Google search API. This study just chooses either of them with no preference. The inputs for search engines APIs are the queries. The outputs of search engines are the ranked hit list with corresponding webpage abstracts according to the queries.

4.5 ENGLISH-LANGUAGE NLP TOOLS

This study relies on some syntax structures and semantic information to extract the target entities, and the NLP tools described in this section are used to pre-process the corpus, as the foundation of answer entity extraction.

Stanford Named Entity Recognizer (Stanford NER) labels sequences of words in a text as the names of things, such as persons, locations, and organizations used in this work. Stanford NER tools implement linear chain Conditional Random Field (CRF) sequence models for Named Entity Extraction task. It not only includes three trained class identifiers for the types of persons, locations, and organizations, but also provides the API for users to train customized entities [Finkel et al., 2005].

Illinois Named Entity Tagger is another state-of-the-art NER tagger that tags plain texts with named entities (persons, organizations, locations, or miscellaneous) [Ratinov and Roth, 2009]. It uses gazetteers extracted Wikipedia, word class models derived from unlabeled texts and expressive non-local features. This NER including the Wikipedia gazetteers can be a good complement for the Stanford NER.

Stanford Part-Of-Speech Tagger (POS tagger) is the software that reads texts in the documents and assigns part-of-speech (such as noun, verb, adjective etc) tags to each word (and other tokens). Stanford POS tagger is a Java implementation of the log-linear POS taggers [Klein and Manning, 2003]. The sample outputs of POS taggers are as follows.

```

$ java -cp stanford-postagger.jar edu.stanford.nlp.tagger.maxent.MaxentTaggerServer -client -host nlp.stanford.edu -port 2020
Input some text and press RETURN to POS tag it, or just RETURN to finish.
I hope this'll show the server working.
I_PRP hope_VBP this_DT 'll_MD show_VB the_DT server_NN working_VBG ._.
```

All the taggers follow the Penn Treebank POS tagset. In the above example, PRP stands for personal pronoun; VBP means it is a non-3rd and present verb; DT means determiner; VB means base form of verb; NN, means singular or mass noun; VBG mean present participle verb. The POS taggers in my study are used to analysis the subject-verb-object structure of sentences.

5.0 GERMANE DOCUMENT IDENTIFICATION

Germane document identification in the entity retrieval task aims at identifying a small set of germane documents from the large corpus for further answer entity extraction. These germane document sets should meet two criteria: they contain as many answer entities as possible; and their sizes are condensed as much as possible, as discussed in Section 3.3. The smaller sets germane documents are, the shorter time entity retrieval executes. Germane document identification is the first quantity of the TREP model as described in Chapter 3, i.e., $\sum_d p(d|q, t)$. It is the probability of a document d is generated by the query q and target entity type t . This chapter discusses three approaches for germane document identification. The first method treats germane document identification as a conventional document retrieval problem. Secondly, we discuss an entity type language model for germane document identification, which considers the similarity between the target entity type and the germane document category. The last one is the learning to rank method for germane document identification.

5.1 GERMANE DOCUMENT IDENTIFICATION AS CONVENTIONAL DOCUMENT RETRIEVAL

If we assume the target entity type t is independent with the query q and the document d , then germane document identification is equal to conventional document search, i.e., $p(d|q, t) = p(d|q)$. Therefore, the first approach treats germane document identification as a conventional document retrieval problem. That is, relevant documents are the same as germane documents with the assumption that these documents contain entities for further

extractions.

In previous studies, various document retrieval methods are applied in germane document identification. Some teams use BM25 for retrieval [Zhai et al., 2009]. More others use language model, such as [Wu and Kashioka, 2009]. Fang applied a language model on the structured retrieval—on document, passage and entity level—to find germane documents [Fang et al., 2009]; McCreadie applied the same idea of structure retrieval but on webpage title and body level [McCreadie et al., 2009]; Zheng used the language model on document and snippet (50-word window size) level [Zheng et al., 2009]. Some other teams consider the query constructions to refine the queries representing users’ information needs. For example, Vydiswaran tried to identify the information need (the narrative part of topic) as a structured query which was represented as a relation including a relation description, an entity of focus, and an entity of interest [Vydiswaran et al., 2009]. Yang also did some query re-constructions by adding the synonym of topic entities into the query for searches [Yang et al., 2009].

In our annotation processes, the assessors find the germane documents for all 20 TREC 2009 topics with some proper queries on search engines. Therefore, we consider to simulate this process: generating the proper queries from the narratives or topics and applying these queries on the search engines, i.e., treating germane document identification as a conventional document retrieval task.

This experiment is based on 20 topics from the TREC 2009 entity retrieval task. Yahoo search engine is used as an indexing and searching system for the corpus. The queries are from the narrative parts of topics. The experiment tests whether the queries from topic narratives can find the germane documents and how to choose the retrieved documents as the germane documents.

The experiment methodology is as follows: the queries generated from the topic narrative part are issued to the Yahoo!Boss search engine API; the top 100 results from Yahoo are evaluated; the performance is evaluated by precision, recall and F-measure. The results are as in Table 6.

According to the F-measure ($F@2=0.22$), the top two documents are the most valuable germane documents. Although the precision at 1 or 2 is 0.2, this approach can only find

Table 6: Results of germane document identification as conventional document retrieval

Rank	P	R	F	Top	P	R	F
100	0.008	0.416667	0.015668	10	0.06	0.3	0.098834
90	0.008333	0.391667	0.016281	9	0.066667	0.3	0.107727
80	0.009375	0.391667	0.018264	8	0.075	0.3	0.118384
70	0.010714	0.391667	0.020796	7	0.085714	0.3	0.131389
60	0.0125	0.391667	0.024144	6	0.1	0.3	0.147619
50	0.015	0.391667	0.028776	5	0.1	0.258333	0.141667
40	0.01875	0.391667	0.035609	4	0.125	0.258333	0.165238
30	0.025	0.391667	0.046698	3	0.166667	0.258333	0.198333
20	0.0375	0.391667	0.067824	2	0.225	0.233333	0.223333
				1	0.25	0.116667	0.158333

a small part of germane documents with regards to the recall at 100 (recall@100=0.41). That means this approach misses more than half of germane documents. Another finding is that almost half of topics have at least a Wikipedia page as the germane document, which means Wikipedia is a good external source for extracting answer entities. Although germane documents can be found for most topics according to the annotators, the number of answer entities for each topic is various. Therefore, it is not proper to use a simple threshold to cut the number of germane documents for each topic.

Further analyses on the query generation are conducted on the TREC 2009 topics. Two kinds of queries can be generated for germane document identification. The first are queries generated by topic entities (e.g., MedImmune, Inc), and the second are queries generated by descriptions (e.g., products of MedImmune, Inc).

Generating queries from descriptions is the most intuitive way. Because the description part provides more information than the topic entity, it is prone to finding documents with answer entities. There are 9 out of 20 cases in the TREC 2009 data sets.

Generating queries from topic entities is that topic entities are better sources as

queries, because the descriptions hurt search results. For example, for the topic of organizations that award Nobel prizes, the description (e.g., organizations that award) for topic entity (e.g., Nobel prizes) can cause the error results from the similar concepts (e.g., Nobel prize awarded organizations), with the assumption of bag-of-words retrieval system, especially when there is no special pages that discuss about the target entities. There are 4 out of 20 cases in the TREC 2009 data sets.

Generating queries from descriptions or topic entities means there are no differences in two kinds of queries because the descriptions fail to provide the additional information than topic entities. A typical case is that the descriptions of topics are the entity type requirements for answer entities, which usually do not appear in the documents, e.g. “students of Claire Cardie”. The “students” is the entity type, which is hard to bring value in the search, especially when there is no special webpage indicating this entity. There are 7 out of 20 cases in the TREC 2009 sets.

Except for the above cases, there are some topics failing at correctly representing the relations between topic entities and answer entities. For example, for the topic of what are some of the spin-off companies from the University of Michigan, the representation of the relation between the topic entity (i.e., the University of Michigan) and the answer entities is “spin-off companies”, which will be more effective if we use their synonyms of “spun of/from/of from”.

These analyses also confirm that germane document identification can only deal with term co-occurrence problems since the retrieval model is built on the assumption of the bag-of-words model(i.e., the independence of terms in a document). Answer entity extraction is an important complement for germane document identification in the entity retrieval task.

5.2 ENTITY TYPE LANGUAGE MODEL

One of the big differences between conventional document retrieval and entity retrieval is that entity retrieval emphasizes the target entity type during the retrieval. Therefore, the entity type language model is considered to integrate the entity type into the retrieval.

The hypothesis is that we can more accurately find germane documents by considering the similarity between document categories and entity types.

The scenario of retrievals on documents associated with document types is not only in germane document identification but also in the documents with tags. Documents in social network, such as Facebook or Twitter, can be treated as this type of documents, where users’ published articles are followed with some comments. The online encyclopedia articles, such as Wikipedia, are another kind of examples, whose entries are attached with their associated categories. Currently the retrieval on these documents is still based on the common document retrieval methods, which ignores all tags and assumes the whole document is the “bag of words”. However, these tags can provide the important hints for some retrieval tasks. For example, in the picture sharing website, such as flickr, the tags of each picture are useful in representing the contents of the pictures. Users’ assigned tags for the academic articles in Citeulike also indicate the related concepts for articles. How to better use these human tagging information to improve the relevant document retrieval will be an interesting topic.

As mentioned in Section 3.3, germane document identification, $\sum_d p(d|q, t)$, is to find the small set of germane documents for further extractions. Previous section assumes the entity type t is independent on the documents d , so that we ignore the type information and treat germane document identification as a conventional document retrieval problem. In this part, we remove this assumption and consider the similarity between document categories and entity types using the entity type language model, as shown in Equation 5.1.

$$\begin{aligned}
 p(d|q, t) &= \sum_{category} p(d, category|q, t) \\
 &= \sum_{category} p(category|q, t)p(d|q, category, t)
 \end{aligned}
 \tag{5.1}$$

1. The first quantity of the entity type language model, $p(category|q, t)$, is the similarity of the target entity type and the document category, called **category similarity**.
2. The second quantity, $p(d|q, category, t)$, is the similarity of the document and the query with the certain type, called **document similarity**. If we neglect the type, it will be the standard language model for document retrieval.

The goal of this section is to demonstrate the dependencies between entity types and documents as well as entity types and queries can improve germane document identification. It also indicates that the entity type language model can improve the retrieval on the documents with associated types.

5.2.1 Category Similarity Strategies

With regards to the entity type language model, the first quantity is category similarity, i.e., $p(\textit{category}|q, t)$, which estimates the similarity between entity types *category* and the target entity type *t* given the query *q*. An entity type likelihood model is proposed for the estimation. Similar to the query likelihood model, a language model M_t for all entity types in the corpus are constructed. The goal of the category similarity is to rank entity types by $p(\textit{category}|q, t)$, where the probability of a document type is interpreted as the likelihood that it is relevant to the target type.

There are two methods for estimating the entity generation probability.

- The first approach is based on the assumption that the query *q* is independent from the document category *category* and the entity type *t*. Therefore, we only care about the similarity between the entity type and the document category. The type generation is as shown in Equation 5.2.

$$\begin{aligned}
 & p(\textit{category}|q, t) && (5.2) \\
 & = p(\textit{category}|t) \\
 & = \prod_{v \in t} \frac{tf_v}{L_t};
 \end{aligned}$$

- The second approach is based on the assumption that all terms in queries and types are independent. The probability of producing query type *category* with entity type *t* using maximum likelihood (MLE) and the unigram assumption, as shown in Equation 5.3, where $tf_{v,t}$ is the frequency of term *v* in the type *t*, and L_d is the number of tokens

in the type t .

$$\begin{aligned}
p(\text{category}|q, t) &= \frac{p(t, q|\text{category})p(\text{category})}{p(q, t)} \\
&\approx p(t, q|\text{category}) \\
&= K \prod_{v \in q, t} \hat{p}_{mle}(v|M_t)^{tf_{v,t}} \\
&= K \prod_{v \in q, t} \frac{tf_{v,t}}{L_t}; \tag{5.3}
\end{aligned}$$

In particular, some words in the queries representing users' information need are not in the types at all. If we estimate the missing terms in document type as 0, we get none generation probability for the target type generation, which will cause errors. The Jelinek-Mercer smoothing (or linear interpolation) is used in the language model for smoothing: discounting non-zero probabilities and giving some probability mass to unseen words. $\hat{p}(v|t) = \lambda \hat{p}_{mle}(v|M_t) + (1 - \lambda) \hat{p}_{mle}(v|M_{ct})$, where $0 < \lambda < 1$ and M_{ct} is a language model built from the entire entity type collection.

5.2.2 Document Similarity Strategies

To estimate the second quantity of the entity type language model, $p(d|q, t, \text{category})$, we use the expanded query likelihood language model, as shown in the following:

$$\begin{aligned}
p(d|q, t, \text{category}) &\propto p(q, t, \text{category}|d) \\
&= p(q, t|d)p(\text{category}|d) \\
&\propto p(q, t|d) \\
&= \prod_{v \in q, t} p(v|d); \tag{5.4}
\end{aligned}$$

Assuming the entity type is independent from the query and the target type, the probability of each entity type is equal. Given documents as sequences of terms and each term as independent, we can estimate the probability of document generation using a language model. This model also faces the data sparse problem. Similar to the previous one, Jelinek-Mercer smoothing is used. We can use the queries extracting terms from different parts of

the topics with and without category information for the estimation. This study uses the following methods for the document similarity strategies:

- **Title** is a baseline system with the title parts of topics as queries.
- **TitleDesc** is a baseline system with the title and description parts of topics as queries.
- **TitleDescNarr** is a baseline system with the title, description and narrative parts of topics as queries.
- **TitleCat** uses the title and target category parts of topics as queries.
- **TitleDescCat** uses the title, description and target category parts of topics as queries.
- **TitleDecNarrCat** uses the title, description, narrative and target category parts of topics as queries.

The experiments are summarized in the following table.

	Title	Title + Description	Title + Description +Narrative
Non-Category	Title	TitleDesc	TitleDescNarr
With-Category	TitleCat	TitleDescCat	TitleDecNarrCat

5.2.3 Experiments

The experiments presented here evaluate whether the entity type language model can improve germane document identification in the documents with their associated types. Because this model assumes the documents are assigned categories, the experiment requires the documents with their categories. Therefore, we chose the INEX2007 entity retrieval task, whose corpus is from Wikipedia articles with corresponding category information, instead of the TREC task. The original documents set are in the HTML format, as shown in Appendix C. In order to use the high quality of human markups, the semantic annotated version of Wikipedia entries for the whole corpus is extracted, as shown in Appendix D. The corpus is pre-processed by transforming the HTML format into the XML format with semantic markups, removing stop-words, and indexing the XML files with Indri tool.

Since this corpus is consisted of Wikipedia articles with human annotated categories, we rely on those markups to identify the named entities instead of employing named entity

identification tools to identify them. Moreover, candidate entities considered in this experiment are those with corresponding Wikipedia articles. An entity in an article is the one with a link to the corresponding article. Therefore, the distinctions between articles and entities are abandoned. Since each entity has a corresponding Wikipedia article, we have the following hypotheses: a good entity page answers the query; and a good entity page is associated with a category close to the target entity type. In the context of Wikipedia, the type of an entity is defined by the categories assigned to the entity’s article. One entity can have multiple types, and Wikipedia categories are hierarchically organized. Therefore, an entity assigned to a category also belongs to its ancestor categories. However, the hierarchy of Wikipedia categories is not a strict tree structure. That is, there exists a loop in the structure. Moreover, if the link path between two categories is too long, then it can lead to unexpected type assignments. However, this experiment ignores the hierarchical structure in the Wikipedia categories and only considers the directly assigned categories. Similarity measures between two concepts in a hierarchical structure have been studied in the ontology, such as tree-based similarity [Blanchard et al., 2006].

The experiments for the entity type language model are divided into two parts. The first one evaluates the document similarity, and the second one evaluates the entity type similarity.

The experiment on document similarity strategies evaluates whether category information applied on the document similarity estimate in the entity type language model can improve entity retrieval results. The experiment evaluates the six groups described in Section 5.2.2, i.e., Title, TitleDesc, TitleDescNarr, TitleCat, TitleDescCat, and TitleDescNarrCat. All queries have removed stop-words. Table 7 shows the performance scores on these six runs.

Although there are slight improvements on the three runs of Title, TitleDesc, and TitleDescNarr, there are no significant differences among them (two tails T-test on MAP values). Same as the non-category group, there is no significant difference in the results with category information (TitleCat, TitleDescCat, and TitleDescNarrCat). Comparing the results with and without category information as part of query, there are no significant differences too. Therefore, we conclude that category information adds few value on the document

Table 7: Results of document similarity estimations in the entity type language model

	MAP	REL-RET	P@5	P@10	P@15
Title	0.1446	331	0.19	0.16	0.15
TitleDesc	0.1503	330	0.176	0.172	0.149
TitleDescNarr	0.1538	324	0.184	0.176	0.1547
TitleCat	0.1387	343	0.168	0.168	0.152
TitleDescCat	0.1448	352	0.168	0.184	0.152
TitleDescNarrCat	0.1559	342	0.184	0.176	0.1627

similarity estimation on the entity type language model.

The experiment on entity type similarity strategies evaluates whether category information in the entity type similarity estimation can improve germane document identification on the entity type language model. Because there are no significant differences on the various groups of the document similarity, the following experiment only evaluates two entity type similarity strategies as well as the base line system of the title part as query. The experimental systems are:

- **Title** is a baseline with the title part of topic as query.
- **Title_CatLM** is the model combining the similarity of a query and a document by using the title part of topic as query, and the similarity between document categories and target entity types uses Equation 5.2.
- **Title_CatTitleLM** is the model combining the similarity of a query and a document by using the title part of topic as query, and the similarity between the document category and the target entity type as well as query (Equation 5.3).

Table 8 shows the results. The result of the Title_CatLM group is significantly better than the result of the Title group (two pairs T-test with p-value of 0.01). The Title_CatTitleLM result is significantly better than the Title result (p-value is 0.016). But there is no significant difference between Title_CatLM and Title_CatTitleLM. Therefore, we can conclude that

entity type information is efficient on the type similarity estimation with the entity type language model.

Table 8: Results of the entity type estimation in the entity type language model

	MAP	REL-RET	P@5	P@10	P@15
Title	0.1446	331	0.19	0.16	0.15
Title_CatLM	0.2124	331	0.312	0.24	0.2
Title_CatTitleLM	0.2276	342	0.2260	0.2200	0.1867

5.2.4 Summary

This section investigates the methods of the entity type language model for germane document identification by considering the relation between entity types and document categories. The experiment is based on INEX 2007 data. The reason for using INEX task as testing set instead of TREC data set is that INEX corpus composes XML files with much more information than HTML pages, which can provides us an easy way to explore the useful factors for document retrieval. Moreover the Wikipedia articles contain the category information which meets the requirement of this experiment for documents with categories. The results shows that although category information has few effects on the document retrieval, it can significantly improve the entity type similarity estimation which, in turn, improve germane document identification in the entity type language model.

5.3 LEARNING TO RANK

The main goal of germane document identification is to retrieve a document set as small as possible but containing answer entities as many as possible. Section 5.1 treating germane document identification as a conventional document retrieval task can only find part of germane documents, with regard to $R@100=0.42$ in Table 6, and achieve low accuracy, with

regard to $P@1=0.25$ in Table 6. There are some limitations of this approach. First, it is hard for a system to decide how to generate a proper query for a topic. For example, it is hard to decide whether it is better using topic entities as queries (e.g., “Claire Cardie”) or it is better using descriptions as queries (e.g., “students of Claire Cardie”) for a particular topic, especially when the topic is tricky. The query, such as “organizations that award Nobel prizes”, is easily confused with some similar query, such as “organizations awarded Nobel prizes”. Secondly, the conventional document retrieval approach highly relies on the ranking, so that a proper threshold is required for cutting out the germane documents. However, how to find the proper threshold is hard. If the threshold is too high, it will bring a big germane document set; if the threshold is too low, it will miss the low ranked germane documents. Furthermore, the entity type is also important factor for finding germane document, and how to integrate the type information in the retrieval, especially in the documents without category information, is also a challenge.

To tackle the problems mentioned above, we propose a learning to rank method for germane document identification. That is, with a model learned from the training data sets, the system can predict the probability of a germane document. This method can combine features from various considerations for germane document identification task. To learn a model, a variety of features representing documents are generated, and then the machine learning method—logistic regression—is applied to estimate the probability of a germane document. This work is originally published in the [Li and He, 2011b].

5.3.1 Germane Document Identification with a Learning to Rank Approach

Learning to rank or machine-learned ranking is a type of supervised machine learning method to automatically construct a ranking model from training data, so that the model can sort documents according to their degrees of relevance, preference, or importance [Liu, 2009]. In this section, we delineate germane document identification as a learning to rank problem, that is, a learning task to predict the germane document according to the training data.

In recent years, more and more machine learning technologies have been used to information retrieval task for learning the ranking model, such as the work on relevance feed-

back [Drucker et al., 2002] and automatically tuning the parameters of existing IR models [Taylor et al., 2006]. Most of the state-of-the-art learning to rank methods learn on the combining features extracted from query-document pairs through discriminative training as Figure 7. germane document identification in this section adapts the general learning to rank structures and summarizes the framework as follows:

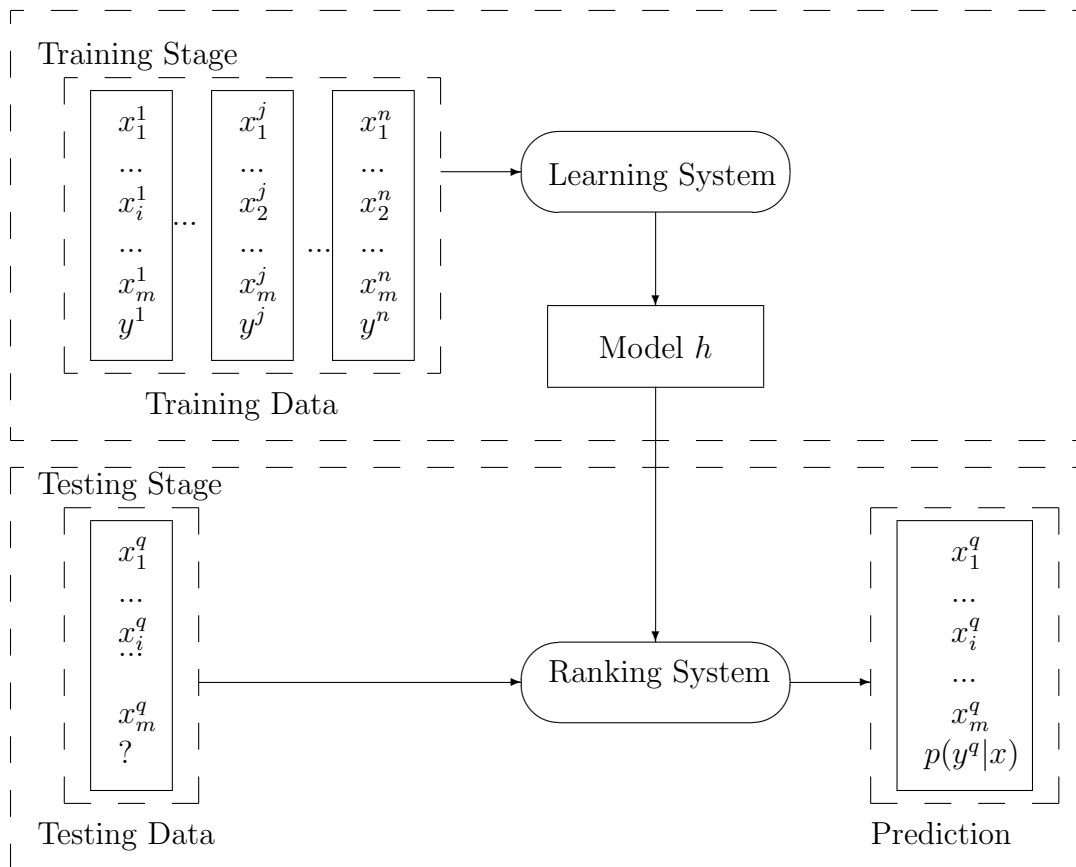


Figure 7: The learning to rank framework

- The input space in the training stage is composed of feature vectors and their corresponding labels. Features in a feature vector (denoted as $(x_1^j, \dots, x_i^j, \dots, x_m^j)$) are the ones extracted from each single document d_j for the corresponding topic q_j . The label y^j indicates whether the documents are the germane documents or not. If $y^j = 1$, then this document is a germane document for a particular topic; if $y^j = 0$, then it is not. Therefore, the input space is denoted as $\{(x_1^1, \dots, x_i^1, \dots, x_m^1, y^1), \dots, (x_1^j, \dots, x_i^j, \dots, x_m^j, y^j), \dots, (x_1^n, \dots, x_i^n, \dots, x_m^n, y^n)\}$.

Here, we should note that the document features, $(x_1^j, \dots, x_i^j, \dots, x_m^j)$, not only come from the documents, but also from the queries, and the relationships between documents and queries (e.g., the document ranking according to the query in the conventional retrieval model). The input space in the testing stage is only composed of feature vectors which represent the documents and the corresponding topics.

- The output space in testing stage contains the probability prediction of each single document to be the germane document according to the query q , that is, $p(y^q = 1 | x_1^q, \dots, x_i^q, \dots, x_m^q)$.
- The hypothesis space contains functions that take the feature vectors as inputs, and predict the probability of a document to be the germane document. The function will be learned from the training data set. Logistic regression is a generalized linear model used for binomial regression. It was first used in the TREC-2 conference by Berkeley researchers [Cooper et al., 1992], and then it was extensively in medical and social sciences fields. In this study, we use logistic regression for germane document identification. Logistic regression is a sigmoid linear function of data. That is,

$$p(y^q = 1 | x_1^q, \dots, x_i^q, \dots, x_m^q) = \frac{1}{1 + e^{\sum_i w_i x_i}}$$

- The optimal function examines the accurate prediction of the ground truth label for each single document. With the logistic regression model, the prediction function directly learns the probability of a document to be the germane document with the given features. Therefore, the training data are used to estimate the parameters of η . It will be calculated as following:

$$w_0^{t+1} \leftarrow w_0^t + \eta \sum_j (y^j - p(y^q = 1 | x_1^j, \dots, x_i^j, \dots, x_m^j, w^t))$$

For $i = 1, \dots, m$

$$w_i^{t+1} \leftarrow w_i^t + \eta \sum_j x_i^j (y^j - p(y^q = 1 | x_1^j, \dots, x_i^j, \dots, x_m^j, w^t))$$

Here, η is the step size. The iteration calculates until the parameter converges.

5.3.2 Variety of Features

Applying the learning method to germane document identification raises the questions of what kinds of information should be used in the learning process. Although many different types of information can contribute toward deciding germane documents, the two principles are followed in the process of feature selections:

1. The feature should not be limited by the instances.
2. The feature should be general enough and domain independent so that the model could be generalized to other topics regardless of the domain.

Four types of features are generated for germane document identification: query features, document features, rank features, and similarity features.

5.3.2.1 Query features Query features are selected according to the principle described in [Jones et al., 2006]. It is the isolated characteristics of elements in queries (e.g., the length of query and the length of narrative). There are five features considered. Firstly, we consider the source of queries, and this feature indicates whether it is from topics or from narratives. With the previous experiment, the queries used for germane document identification can be various and from different sources, so that the system needs one feature to indicate the source of queries. Secondly, we consider the query entity types. Germane documents are easier to be found for certain entity types, such as persons and locations, than the others, such as products. Therefore, we use one feature to indicate the entity types. Thirdly, the length of queries and their different parts are also detected as features. The assumption here is that the longer topic entities or narratives or relations are, the more information they carry and the better sources they are to generate the queries. Fourthly, we consider whether the entity mention has different form. For example, for the query of Journals published by the AVMA, the topic entity is American Veterinary Medical Associations, which is the full name for the acronym form of AVMA in the narrative. The assumption is that if the topic entity has different forms then the query with a acronym form might not be a good one. Last, we consider the hit information. The hits indicate how many relevant documents returned from the search engines for a topic. The more the relevant documents are retrieved by a search

engine with a query, the more chance this query is the proper query to generate germane documents. These features are summarized and defined as follows:

EntityNarrative indicates if the query is generated from the topic entity or the narrative part of the topic. In the pilot study, we find that both query generation methods are useful. Therefore, in the learning to rank method, we choose both methods to generate queries: the topic entities as queries and the narratives as queries.

EntityType indicates the target answer entity types required by the topics. Its value includes persons, locations, products, and organizations.

EntityLength is the character length of the topic entity without stop words.

NarrativeLength is the character length of the topic narrative without stop words.

RelationLength is the absolute character length difference between topic entity and the narrative without stop words. $RelationLength = |NarrativeLength - EntityLength|$.

EntityTokenLength is the token length of topic entity without stop words.

NarrativeTokenLength is the token length of the narrative without stop words.

RelationTokenLength is the absolute token length difference between the topic entity and the narrative without stop words. $RelationTokenLength = |NarrativeTokenLength - EntityTokenLength|$.

IsSameEntity is to indicate whether the surface name of the topic entity is different from its surface name in the topic entity field. If it is different, then the score is 1, and the else is 0.

Hits is the numbers of relevant documents retrieved by the search engine.

Hitstrend is a binary value feature of (1, -1), which compares the hits of the topic entities as queries and the narratives as queries. If the number of hits from the topic entity queries is larger than the number of hits from the narrative queries, then $Histrend = 1$. Otherwise, $Histrend = -1$.

5.3.2.2 Document features Document features describe the characteristics of documents. The Wikipedia pages are supposed to have more authoritative information, so they are more likely to be the germane documents. In this study, we especially detect Wikipedia

as an important source for germane documents. In the future, other sources with high quality pages as germane documents can be included, such as the entity’s homepage. We define the following features:

IsWikipedia is a binary feature (1 or 0) indicating whether this hit is from the Wikipedia.

IsEntityWikipedia is a binary feature (0 or 1) to indicate whether this hit refers to a Wikipedia page, whose entry name is the same as the topic entity itself. For example, for , the value of IsEntityWikipedia is equal to 1, when the query terms are “MedImmune, Inc.” and the its hit is <http://en.wikipedia.org/wiki/MedImmune>.

5.3.2.3 Rank features Rank related features are based on the rank information to indicate the popularity of the documents. These features can also give useful hints for germane document identifications. For example, we assume that the higher rank of a document, the more possible it is to be the germane documents. We list the following features:

DocRank is the rank of a returned URL from the search engine for each query.

RankScore is the normalized ranking score for each hit. It is calculated by summing up the reverse of rank for the same URL in the same topic. This score will merge the results on both the entities as queries and the narratives as queries. It is denoted as follows:

$$RankScore(URL) = \sum_{URL} \frac{1}{rank_{URL}}$$

NewRank is the new rank list according to the RankScore, which considers the same URL in the same topic but retrieved by different queries.

5.3.2.4 Similarity features Similarity features are the measurements of the similarity between the query and its retrieved document. We assumes that the shorter of the semantic distances (measured by the semantic similarity) between a query and a document, the higher chance it is a germane document. For example, for the query of products of MedImmune Inc., if the document title is also ‘products of MedImmune Inc, then there is a high probability to be the germane document for this query. We design some term distance measures to estimate the similarity, such as TitlePrecision, TitleRecall, ContentPrecision and ContentRecall.

However, term distance measures suffer some drawbacks, such as hardness to measure the similarity between the entities and their corresponding synonym sets or abbreviation forms. For example, “AVMA” is the acronym of “American Veterinary Medical Associations”. If we use the term distance measures, we will find no similarities between these two entities. Therefore, semantic measurements are introduced. Some systems use thesaurus to map the synonyms or abbreviations, e.g., WordNet or Wikipedia. Because it is hard to find their corresponding entries in thesaurus for all queries narrated in sentences, an alternative, the WebDice coefficient, is introduced for this problem. The similarity features are defined as follows:

TitlePrecision is the rate of the number of the overlapping terms between the hit’s title and query to the number of terms in the query, which represents the similarity between a query and its hit. It is defined as follows:

$$TitlePrecision = \frac{num_of_terms_in_ (query \cap title)}{num_of_terms_in_ (query)}$$

Here, the terms exclude the stop words (e.g., the, a, an). For example, the TitlePrecision score of the topic “Products of MedImmune, Inc.” for the document, <http://www.medimmune.com/>, with the title of “MedImmune, Inc.” is 0.667, because the $num_of_terms_in_ (query \cap title)$ is 2 (only the terms of “MedImmune” and “Inc” are counted), and the $num_of_terms_in_ (query)$ is 3 (only the terms of “products”, “MedImmune” and “Inc” are counted), and the term of “of” is the stop word.

TitleRecall is the rate of the number of the overlapping terms in the query and in its hit’s title to the number of terms in the title, which represents the similarity between a query and its hit. It is defined as follows:

$$TitleRecall = \frac{num_of_terms_in_ (query \cap title)}{num_of_terms_in_ (title)}$$

Here, the terms exclude the stop words (e.g., the, a, an). For example, the TitleRecall score of the topic of Products of MedImmune, Inc. and the document of <http://www.medimmune.com/> with the title of MedImmune, Inc. is 1, where the $num_of_terms_in_ (query \cap title)$ is 2 (only the terms of “MedImmune” and “Inc” are counted), and the $num_of_terms_in_ (title)$ is 2 (only the terms of “MedImmune” and “Inc” are counted).

TitleDistance is the feature to measure whether the query terms are close to each other in the title part. We assume that the documents with the titles containing the query phrases are more relevant than the one with the titles containing the query keywords. TitleDistance is the rate of query length to the scope of query terms in the title, as follows:

$$TitleDistance = \frac{num_of_terms_in(query)}{num_of_terms_in(scope_of_query_terms_in_title)}$$

ContentPrecision is similar to TitlePrecision, but replaces the title part to the hit’s content. We want to cover the features from both titles and contents. That is,

$$ContentPrecision = \frac{num_of_terms_in(query \cap content)}{num_of_terms_in(query)}$$

ContentRecall is similar to TitleRecall, but replaces the title part to the hit’s content part.

That is,

$$ContentRecall = \frac{num_of_terms_in(query \cap content)}{num_of_terms_in(content)}$$

ContentDistance is similar to TitleDistance, which measures the query terms in the content part. That is,

$$ContentDistance = \frac{num_of_terms_in(query)}{num_of_terms_in(scope_of_query_terms_in_Content)}$$

WebDiceOrg is to define the similarity between two queries by measuring the Web space similarity of two relevant document sets retrieved by the two queries. It is the approximation of F-measure in the web [Bollegala et al., 2007]. Page counts of the query P and Q can be considered as the co-occurrence of two words P and Q on the web. For example, the page count of the query of “Journals published by the AVMA” is 145,000. The page count for the document of “AVMA Journals” is 245,000, and the page count for the document of “AVMA Journals - Reprints, ePrints, Permissions” is 159. From the page count similarity, “Journals published by the AVMA” is closer to “AVMA Journals” than “AVMA Journals - Reprints, ePrints, Permissions”. The WebDiceOrg coefficient

has been demonstrated to outperform the other three modified co-occurrences (i.e. WebJaccard, WebOverlap, and WebPMI) in [Bollegala et al., 2007]. Therefore, in this work, we only use WebDiceOrg. The WebDiceOrg is defined as follows.

$$WebDiceOrg(query, title) = \begin{cases} 0 & \text{if } H(query \cap title) \leq c \\ \frac{2H(query \cap title)}{H(query) + H(title)} & \text{otherwise} \end{cases}$$

where $H(query)$ denotes the page counts for the query of “query” in a search engine, and D denotes the page counts for the query of “query and title”. c is a predefined threshold (e.g., $c=5$) to reduce the adverse effects caused by random co-occurrence.

WebDice is the normalized WebDiceOrg score with the maximum value of WebDiceOrg, so that its value is between 0 and 1.

$$WebDice(query, title) = \frac{WebDiceOrg(query, title)}{\max\{WebDiceOrg(query, *)\}}$$

5.3.3 Evaluation

Experiments on the TREC Entity Extraction Task (2009 and 2010) data sets evaluate whether the learning to rank method can improve germane document identification. The test data of TREC entity retrieval 2009 and 2010 topics is described in Section 4.2.2. The total topics are 70. The evaluation criteria are precision, recall and F-measure, as described in Section 4.3. The evaluation systems are as follows:

- **Baseline System I** uses the topic entities as queries for germane document identification. In the experiment, we use Google search engine, and only consider the top 16 documents as germane documents for the evaluation.
- **Baseline System II** uses the narratives as queries for germane document identification. Google search engine is used to collect the germane documents, and only top 16 documents are considered as germane documents for the evaluation.

- **Baseline System III** uses the mixture germane document rank list from the topic entities as queries and the narrative as queries. Mixture germane document list ranks the documents from Baseline System I and Baseline System II with the following score:

$$ds(doc) = \sum_{query} \frac{1}{OriginalRank(doc, query)}$$

- **Experiment System:** the learning to rank method for germane document identification.

The methodology of this experiment is:

1. Collect the candidate germane documents as a pool:

For each topic, various queries are generated. This experiment considers the queries generating from the narratives (i.e., the narratives as queries), and the queries generating from topic entities (i.e., the topic entities as queries).

Issue each query to the search engines, and return a ranked list for each query. In the experiment, we use Google search engine, and only consider the top 16 documents as candidate germane documents. The documents include their rankings, hit's URLs, hit's titles, hit's summary, and query's page count. The results of topic entities as queries are Baseline System I, and the results of the narratives as queries are Baseline System II.

There are total 1116 documents (16 documents per topic; 70 topics total) in Baseline System I, and there are 40 germane documents in it. There are another 1115 documents (16 documents per topic; 70 topics total) in Baseline System II, and there are 64 germane documents in it. The total number of pooling documents for the mixture model, Baseline System III, and the learning to rank method, Experiment System, are 2107. There are 74 supporting documents in this pool.

The mixture germane document list, as Baseline System III, combines the lists of Baseline System I and Baseline System II by ranking the documents in these two baseline systems with the following score:

$$ds(doc) = \sum_{query} \frac{1}{OriginalRank(doc, query)}$$

2. Extract the features for documents:

Various features (i.e., linguistic, structures, web, and queries) discussed in the last section are extracted as feature vectors.

3. Annotate the supporting documents according to the returned documents:

For each returned hit, we mark whether it is the supporting document according to the ground truth. If this page contains the answer, it will be labeled as 1, otherwise it will be labeled as 0.

4. Evaluate the systems:

For the three baseline systems, we will calculate the precision, recall, and F-measure at each ranking.

For the learning to rank algorithm, the ten-fold cross evaluation will be conducted.

- a. The whole corpus are randomly divided into 10 folds.
- b. Every time, we train on the 9 fold and test on the last fold. The logistic regression can estimate the probability of a document to be the germane documents.
- c. We choose the top 16 high probability documents as germane documents according to the probability score.
- d. The final precision, recall, f-measure of learning to rank method are the average of the 10 results.

Table 9 shows the precision (Figure 8(a)), recall (Figure 8(b)) and F-measure (Figure 8(c)) of the baseline systems and experiment system for germane document identifications. The top 16 documents are evaluated. Comparing two baseline systems, the narratives as queries (Baseline System II) is significantly better than the entities as queries (Baseline System I) (for the two-tail t-test, $p < 0.0001$). The mixture model (Baseline III) is significantly better than the entities as queries (Baseline I). However, there is no significant difference in the narratives as queries (Baseline II) and the mixture model (Baseline III). The learning to rank method (Experiment system) runs significantly better than three base systems in precision and F-measure, but not recall.

We find that no matter what kinds of methods use for germane document identification, the topics asking for products are the hardest, because its error rate in the learning to rank method is 0.1, which is higher than the other three (0.01, 0.03, and 0.04 for locations,

Table 9: Results of the learning to rank method for germane document identification

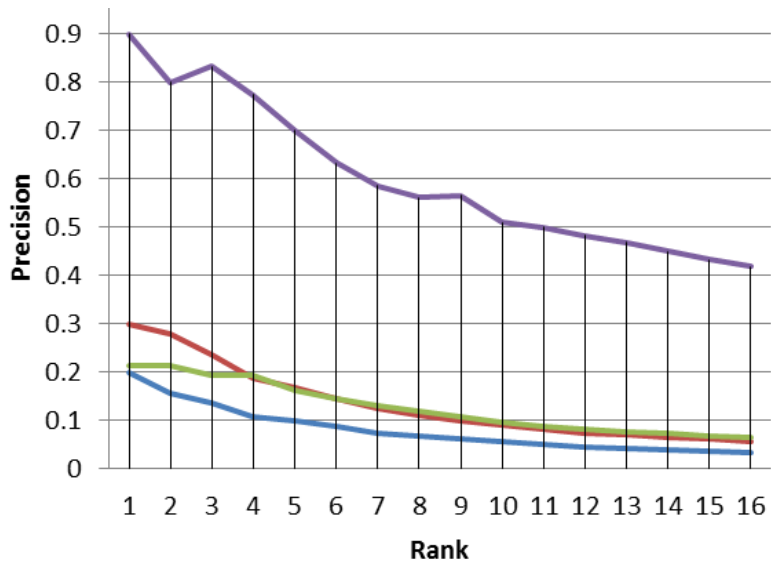
Method	Rank	Precision	Recall	F	Method	Rank	Precision	Recall	F
Baseline I	1	0.2	0.1369	0.1557	Baseline II	1	0.3	0.1940	0.2271
	2	0.1571	0.2300	0.1795		2	0.2786	0.375	0.3076
	3	0.1381	0.3155	0.1860		3	0.2381	0.4905	0.3086
	4	0.1071	0.3226	0.1562		4	0.1893	0.5190	0.2682
	5	0.1	0.3774	0.1539		5	0.1686	0.5631	0.2511
	6	0.0881	0.3988	0.1408		6	0.1452	0.5917	0.2265
	7	0.0755	0.3988	0.1242		7	0.1265	0.6060	0.2038
	8	0.0696	0.4202	0.1171		8	0.1125	0.6095	0.1851
	9	0.06191	0.4202	0.1059		9	0.1	0.6095	0.1678
	10	0.056	0.4202	0.0967		10	0.09	0.6095	0.1534
	11	0.0506	0.4202	0.0890		11	0.0817	0.6095	0.1411
	12	0.0464	0.4202	0.0824		12	0.075	0.6095	0.1310
	13	0.0429	0.4202	0.0767		13	0.0703	0.6167	0.1240
	14	0.0408	0.4274	0.0735		14	0.0653	0.6167	0.1161
	15	0.0381	0.4274	0.0691		15	0.0636	0.6310	0.1135
	16	0.0335	0.4007	0.0612		16	0.0577	0.6231	0.1040
Baseline III	1	0.2143	0.1524	0.1714	LTR	1	0.9	0.0863	0.1576
	2	0.2143	0.3167	0.2471		2	0.8000	0.1536	0.2577
	3	0.1952	0.4452	0.2638		3	0.8333	0.2400	0.3725
	4	0.1929	0.5548	0.2782		4	0.7750	0.2973	0.4295
	5	0.1629	0.5905	0.2490		5	0.7000	0.3355	0.4533
	6	0.1452	0.6310	0.2307		6	0.6333	0.3645	0.4625
	7	0.1327	0.6738	0.2171		7	0.5857	0.3936	0.4706
	8	0.1196	0.7024	0.2006		8	0.5625	0.4318	0.4883
	9	0.1095	0.7214	0.1868		9	0.5667	0.4882	0.5242

Method	Rank	Precision	Recall	F	Method	Rank	Precision	Recall	F
Baseline III	10	0.0984	0.7214	0.1703	LTR	10	0.5100	0.4882	0.4986
	11	0.0896	0.7214	0.1569		11	0.5000	0.5273	0.5130
	12	0.0821	0.7214	0.1453		12	0.4833	0.5573	0.5174
	13	0.0780	0.7357	0.1392		13	0.4692	0.5864	0.5210
	14	0.0735	0.75	0.1321		14	0.4500	0.6055	0.5160
	15	0.0686	0.75	0.1241		15	0.4333	0.6236	0.5111
	16	0.0643	0.75	0.1171		16	0.4188	0.6427	0.5068

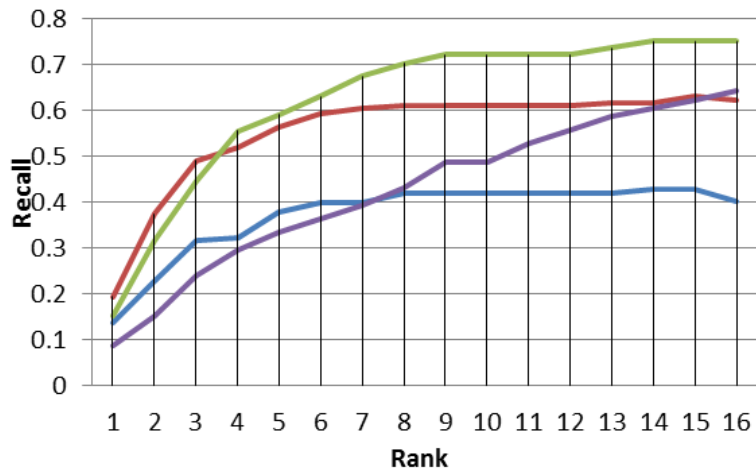
organizations and persons respectively). One reason for the low accuracy of products is that the product usually is general category name, which need to be further clarified in the special retrieval task. For example, CDs, and software are assigned as products. Another reason is that the training sets for the products are too small.

We calculate the co-efficiency for all features used in the learning to rank method in Table 10. The higher absolute value of the weight, the more important it is in the model. The positive value of the feature means it contributes to the non germane documents, and the negative value of the feature means it contributes to the germane documents. From this result, we can summarize the following findings.

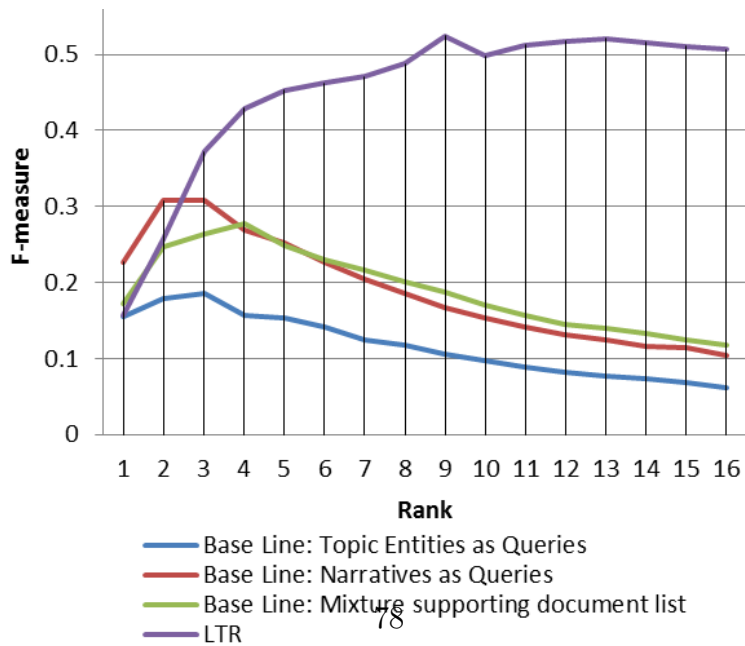
1. Wikipedia entity page is one of the most important features. If the document from Wikipedia page with the same entry name as the entity name, it is more valuable than the other Wikipedia pages, according to the weight score of isWikipedia and isEntity-Wikipedia.
2. The ranking of documents in the ranked list is another factor in the learning method, especially the normalized ranking score which merges the multiple query results. It can be explained as if the document keeps appearing in the returned lists of different queries from the same topic, it should be more chance to be a germane document.
3. Except for the type of products, entity types have low effects on the learning. It is a hint that the complicated entity type (e.g., products) can be an important factor to detect germane document identification.



(a) Precision of the Learning to Rank Method



(b) Recall of the Learning to Rank Method



(c) F-measure of the Learning to Rank Method

Table 10: The features with their weights in the logistic regression model

Feature	Weight	Feature	Weight
isEntityWikipedia	-3.57	TitlePrecision	-2.76
EntityNarrative	-2.28	ContentRecall	-1.85
TitleRecall	-1.24	RankScore	-0.96
WebDice	-0.77	RelationTermLength	0.72
NarrativeTermLengh	-0.5856	isWikipedia	-0.57
EntityType=product	-0.5328	HitTrend	0.4343
NewRank	0.4282	EntityTermLength	0.4223
EntityType=location	0.1737	ContentDistance	-0.1561
Rank	-0.12	Narrativelength	0.0864
RelationLength	-0.0841	ContentPrecision	-0.0505
EntityLength	-0.0429	TitleDistance	0.0366
Entitytype=person	0.0339	Entitytype=organization	0.0313
WebDiceOrignal	-0.0004	Hits	0

4. Term length measures are better than the character length measures, which can be concluded from comparisons between the weights of NarrativeTermLength and NarrativeLength as well as EntityTermLength and EntityLength.
5. ContentRecall, TitlePrecision, and TitleRecall are more important than ContentPrecision. The title parts of documents are more valuable than the content parts.
6. Webdice does help to recognize the germane documents, but the various hits, such as query hits, has no affects on finding germane documents.

With removing the features with their absolute weights smaller than 0.1 (i.e., $-0.1 < \text{feature weight} < 0.1$), we re-run the experiment using the logistic regression based learning to rank method. There are no significant differences on the two results.

With the further consideration about the evaluation of germane document identification, we argue that evaluating germane document identification based on the annotated ground truth of germane document set may be unfair or inaccurate for some testing set because the final task is to detect answer entities instead of germane documents. When the system identify the part of germane documents but covering all answer entities, although the recall of germane document identification is low, it still can support to find all answer entities. For example, for the topic of “products of MedImmune Inc”, we assume the germane document ground truth set only includes two documents. One is the wikipedia page of “MedImmune Inc” with the answer entities of Synagis and FluMist, and the other is the homepage of “MedImmune Inc” with answer entities of Synagis, FluMist, and Ethyol. If germane document identification only finds one document of the wikipedia page, then the recall of germane document identification is 0.5. However, if we consider how much the document contains the answer entities, it is 0.667, which is not so bad comparing with recall of 0.5 based on germane documents. Therefore, the evaluation is not very accurate. In the future research, we will consider to use the number of answer entities existing in the germane documents as the weight for the germane documents, which will be the adjust the precision and recall as well as F-measure evaluation.

5.4 SUMMARY

This chapter discusses different methods for germane document identification. It only deals with word-independence factors by considering the term co-occurrences (i.e. the assumption of the term independence in the document). All the semantic related analyses, therefore, should be postponed into answer entity extraction. We investigate three approaches for germane document identification: conventional document retrieval, the entity type language model, and the learning to rank method.

The first study is about treating germane document identification as a conventional document retrieval task, and the query generation method is discussed. This approach assumes the entity type is independent from the query and the document. In most cases, the

narrative part is the best source for query generation. In some cases, however, it destroys germane document identification. For example, when the answer entities are only part of the Web pages (e.g., “students of Claire Cardie”), the topic entity is a better choice for the query generation. How to correctly represent the relation between the topic entities and the answer entities is another difficulty for query generation. For example, the query of “organizations that award Nobel prizes” presents the relation between the answer entity (organization) and the topic entity (Nobel prizes) as “award”. The other query of “organizations that were awarded Nobel prizes” presents the relation between answer entity (organization) and topic entity (Nobel prizes) as “be-award”. Although they are different semantic meaning, in the retrieval with the assumption of bag-of-words, they get the same results. The topics using some terms seldom in the corpus will be also hard for the retrieval. For example, the topic is like “What are some of the spin-off companies from the University of Michigan?”

Secondly, the entity type language model evaluates germane document identification by considering the document similarity and the entity type similarity with entity types. The experiment demonstrates that document categories have few effects on document similarity, but it does improve the results in entity type similarity which in turn improves germane document identification. But the limitation of this method is that it requires the document has category information.

The learning to rank method for germane document identification estimates the probability of a document to be the germane document. It is a learning-based ranking method. That is, a ranking model is learned from the training set, and then the system applies the model to predict the probability. Multiple features including the different query generation strategies, the lists of candidate germane documents from search queries, and the entity types are applied in this learning to rank method. In the evaluation, 28 features are identified and used for the learning to rank method. The results indicate the learning to rank method is significantly better than the baseline systems.

Although the learning to rank method can improve the precision of the germane document identification, the recall is still low. In future studies, we will investigate methods to improve the recall of germane document identification. For example, since Wikipedia is one of important source for the germane documents, the Wikipedia page with the same entry

name as answer entities should be further processed to separate out more relevant features for germane document identification.

The current evaluation method is based on the germane documents. As discussed, it might be not accurate. Even for the low recall of germane documents, if they cover all answer entities, the system still has a good chance to detect answer entities. Therefore, in the future, the evaluation of germane document identification will be based on the answer entities.

6.0 ANSWER ENTITY EXTRACTION

This chapter investigates answer entity extraction, which extracts the answer entities from germane documents for the entity retrieval task. With the TREPM model, answer entity extraction estimates the probability of a candidate entity e being an the answer entity, given the document d , the query q and the targeted entity type t , i.e., $p(e|d, q, t)$, which is the second term in the TREPM model (Equation 3.2).

Since germane document identification considers the similarity between the query and the document, answer entity extraction can assume the entity is independent from the query given the germane documents. Therefore, it can be simply treated as an entity extraction from germane documents, extracting entities, e , from the germane documents, d , with regards to entity types, t , i.e., $p(e|d, t)$. It relies on the named entity recognizer tools to achieve this goal. This method is used by most current competition groups in TREC.

In order to accurately extract answer entities, the contexts of entities c are considered for the extraction, i.e. $p(e|d, q, t) = \sum_c p(c|d, q, t)p(e|c, d, q, t)$. The first quantity, $p(c|d, q, t)$, is the probability of the context c being detected within a germane document according to the query as well as the answer entity type. The second quantity, $p(e|c, d, q, t)$, is the probability of the answer entity extracted from the context c . In most cases, iterating all contexts is not efficient. Therefore, in order to efficiently extract the answer entities, we consider the most useful contexts $c_{support}$, instead of all possible contexts, to approximate the extraction process, i.e.,

$$p(e|d, q, t) = \sum_c p(c|d, q, t)p(e|c, d, q, t) \approx \sum_{c_{support}} p(c_{support}|d, q, t)p(e|c_{support}, d, q, t)$$

The key point of answer entity extraction with contexts is how to detect these support contexts $c_{support}$ and how to efficiently extract the answer entities from these contexts. As

mentioned in Section 3.4, the contexts of the entities can be interpreted in multiple ways. In the medical domain named entity extraction, the context means the negation, experiencer, and temporal status for the medical findings [Harkema et al., 2009]. In such studies as the word sense identification, the context is interpreted as the term co-occurrence in certain window sizes [Leacock and Chodorow, 1998]. In my dissertation, the context only refers to the text environment surrounding around the answer entities.

According to the topics in TREC 2009 and the ground truth annotation process, the answer entity contexts in germane documents are viewed from two aspects: the physical structures of contexts and the logical structures of contexts. The physical structures of contexts illustrate what media or physical techniques are used to represent the contents. Most of the contexts in the web environment are in web pages, which follow the html or xml script rules for the format representations, while some other contexts use the PDF or DOC files and even pictures for the content representations. The logical structures of the contexts mean how these contents are logically organized. For example, the contexts can be the sentences narrating the relation between two objects using syntax, or the lists or tables with the header rows and columns to indicate the answer entities. We should note that the same logical structure of contents can be represented in the different physical structure contexts. For example, the products of MedImmune Inc can be stored in tables, such as Wikipedia Infobox, or in the sentences, such as “the website contains information concerning MedImmune and its products and services including Synagis, FluMist”, or even presented in the pictures in the webpage. For example, the webpage of http://www.medimmune.com/about_us_products.aspx uses the images to present the products.

According to the physical structures and logical structures, we analysis the answer entity contexts for the TREC 2009 topics. The details are shown as in Table 11.

There are total 36 germane documents for 20 topics, 21 of them from web pages, 14 of them from Wikipedia pages, and one of them from PDF files. From the context structure analyses of the 2009 topics (Table 12), there are 12 cases where answer entities are in the sentences, and there are 24 cases where answer entities are in the tables or lists. For example, in the Wikipedia page of MedImmune, Inc., the table of Infobox contains the answer for the products of MedImmune, Inc. In the homepage of MedImmune Inc., there is a product

Table 11: The answer entity context structures for 20 Topics in TREC 2009

Id	Topic	Url	Logical Context	Physical Context	Notes
2	1	http://en.wikipedia.org/wiki/BlackBerry	Sentence	Wikipedia	
33	2	http://awards.acm.org/homepage.cfm	Table	Webpage	
65	3	http://www.cs.cornell.edu/home/cardie/cv.pdf	Table	PDF	
110	4	http://en.wikipedia.org/wiki/Sports_in_Philadelphia	Sentence	Wikipedia	
g 111	4	http://philadelphia.about.com/.../Philadelphia_Professional_Sports.htm	Sentence	Webpage	
112	4	http://www.yelp.com/.../Philadelphia	Sentence	Webpage	
143	5	http://en.wikipedia.org/wiki/MedImmune	Table	Wikipedia	
146	5	http://www.medimmune.com/about_us_products.aspx	Table	Pictures	
163	6	http://en.wikipedia.org/wiki/Nobel_Prize	Sentence	Wikipedia	
190	7	http://en.wikipedia.org/wiki/Boeing_747	Sentence	Wikipedia	primary users
222	8	http://www.kingsingers.com/c/kings-singers/cds-aand-dvds.html	Table	Wikipedia	
225	8	http://en.wikipedia.org/wiki/King%27s_Singers	Table	Wikipedia	
254	9	http://en.wikipedia.org/wiki/Beaux_Arts_Trio	Sentence	Wikipedia	
256	9	http://www.beauxartstrio.org/about.html	Table	Webpage	
304	10	http://www.indiana.edu/campuses/index.shtml	Table	Webpage	
334	11	http://www.homedepotfoundation.org/donors.html	Table	Picture	
335	11	http://www.homedepotfoundation.org/.../2010-complete-donor-list.htm	Table	Webpage	change lines as lists

Id	Topic	Url	Logical Context	Physical Context	Notes
366	12	http://en.wikipedia.org/wiki/Air_Canada	Table	Webpage	share code vs code-share
352	12	http://www.aircanada.com/en/.../codeshare.html	Sentence	Webpage	
382	13	http://www.avma.org/	Table	Webpage	
399	13	http://en.wikipedia.org/wiki/American_Veterinary_Medical_Association	Sentence	Wikipedia	
386	13	http://www.avma.org/journals/default.asp	Sentence	Webpage	
430	14	http://www.bouchercon.info/history.html	Table	Webpage	Hierarchical lists
441	14	http://en.wikipedia.org/wiki/Anthony_Award	Sentence	Wikipedia	Part of Lists
445	15	http://en.wikipedia.org/wiki/Southeastern_Conference	Table	Wikipedia	
449	15	http://sec12.com/	Table	Webpage	
498	16	http://www.quiltfest.com/sponsor.asp	Table	Picture	
509	17	http://www.foodnetwork.com/chefs/index.htm	Table	Webpage	Part of Lists
540	18	http://en.wikipedia.org/wiki/Jefferson_Airplane	Sentence	Wikipedia	
571	18	http://www.classicbands.com/jefferson.html	Table	Webpage	
557	18	http://www.last.fm/music/Jefferson%2BAirplane	Table	Webpage	
562	18	http://www.facebook.com/.../List-of-Jefferson-Airplane-band-members/...	Table	Webpage	
571	19	http://en.wikipedia.org/wiki/John_L._Hennessy	Table	Wikipedia	
572	19	http://people.forbes.com/profile/john-l-hennessy/8214	Table	Webpage	
619	20	http://en.wikipedia.org/wiki/Islay_whisky	Table	Wikipedia	
622	20	http://www.whisky-distilleries.info/Regions-de-production_EN.shtml	Table	Webpage	

list presenting the answer for this topic. Some sentences, such as “it (MedImmune, LLC) produces Synagis, a drug for ...”, semantically present the product entity. We can see that most answer entities are in the tables or lists (24 out of 36), and only few of them (12 out of 36) are in the sentences.

Table 12: Context structures for 20 topics in TREC 2009

Number of cases	Wikipedia	Web	PDF or Word	Pictures	Total
Sentence	7	5	0	0	12
Tables/Lists	7	13	1	3	24
Total	14	18	1	3	36

Although the Web environment is heterogeneous and the physical contexts for these Web pages are various, this study focuses only on the HTML pages and Wikipedia pages and does not further discuss the other physical structures, such as pictures and PDF files. Entity extractions from HTML pages can demonstrate the similar questions in the entity retrieval task since the same algorithm can be applied to the other media with correct format transformations.

The following study mainly investigates two kind of context logic analyses and extraction: symbolic contexts and syntax contexts. The symbolic contexts are the ones that use symbols to show the nature of the context. For example, the tables and lists are frequently used to concisely display the company’s products or papers cited by an author. The syntactic contexts are culled from sentences’ structures by using shallow sentence analyses. For example, “Apple launches iPad” indicates the company of Apple has the product of iPad.

6.1 ANSWER ENTITY EXTRACTION WITHOUT CONTEXTS

We firstly consider answer entity extraction without contexts. Answer entity extraction in TREPM model is the following form: $p(e|d, q, t)$, which is the probability of an entity e being an answer entities given the document d and the query q with the target type of t . If we

consider the similarity between queries and documents in germane document identification, we assume the probability of answer entities e are independent from queries q given the germane documents d and the target entity type t , i.e., $p(e|d, q, t) = p(e|d, t)$. Therefore, answer entity extraction can be viewed as entity extraction. There are two approaches to extract answer entities without contexts—named entity recognition tools for extractions and the knowledge base for entity type detection.

6.1.1 Answer Entity Extraction with Named Entity Recognition Tools

Most of the researchers in the TREC 2009 entity retrieval task applied named entity recognition (NER) tools for answer entity extraction. The TREP model can be interpreted as an answer entity extraction without contexts, i.e., $p(e|d, q, t)$. In practices, the Stanford NER tool is the most popular tool for this task. Unfortunately, it can only identify the named entities of persons, organizations, and locations, but not products. Therefore, teams like [Zheng et al., 2009] treated proper nouns as candidate product entities. Teams such as [Yang et al., 2009] and [Serdyukov and de Vries, 2009] used external knowledge base (e.g., Wikipedia) to train a named entity tool for products. Similar approaches were done by [Vydiswaran et al., 2009] and [McCreadie et al., 2009]) relying on a dictionary of company names and a pre-defined set of patterns for the product recognition. Most of these researchers did further entity re-ranking since the results directly from the named entity recognition are not promising. Wu specifically evaluated the re-ranking process by calculating the similarities between input query, support snippets, and related entities [Wu and Kashioka, 2009].

We follow the same idea for answer entity extraction without contexts, i.e., the named entity recognition (NER) tool for extractions. The research question is whether the NER tools can extract the answer entities from germane documents. A special parser is designed for the Wikipedia page extraction, because we expect a better parser to pre-process the webpages can improve the extraction. Therefore, the experiment also tests whether the html parser will affect the results of named entity extractions. The answer entities should consider all extractions from germane documents for each topics. Therefore, we compare the results before and after this sum in order to evaluate whether entities extracted from

one document and entities from multiple documents can complete each other. Moreover, the evaluation results are reported according to the entity types (such as products, persons, and organizations) and page types (such as Web pages and Wikipedia pages) in order to test whether these factors affect answer entity extraction.

The experiment is based on the TREC 2009 entity retrieval tasks. All germane documents are preprocessed as plain texts, removing all tags from HTML pages. Stanford NER tool identifies the entity of organizations and persons, and the noun phrase extractor extracts noun phrases as products. Three groups of experiments are evaluated.

- Experiment 1: Stanford NER extracts the entities from the germane documents. The top 10 results are evaluated.
- Experiment 2: With the special parser for the Wikipedia pages, further cleaned up by removing the header and footer tags, answer entities are extracted from Wikipedia germane documents. Because there are a lot of non-relevant contents in the Wikipedia page, e.g., category information in the bottom and language information in the left, a simple parser is introduced to remove the header and footer parts of the Wikipedia to reduce this noise. The experiment evaluates whether removing the noise in this context can improve the results significantly. The top 10 results are evaluated.
- Experiment 3: With the answer entities extracted from every germane document, the algorithm summarizes the results by topics. Different from the previous two experiments by extracting the answer entities by documents, this experiment summarizes the entities across the documents within the same topic. This experiment evaluates whether the real answers for the same topic from the same document or multiple documents can complete the answer sets for each other. The top 10 results are evaluated.

The results of precision, recall and F-measure are as shown in Table 13. With the answer entities within the same topic, the precision and the F-measure significantly improve from 0.103 to 0.17 and from 0.144 to 0.16 respectively (two-tail t-test, $p < 0.001$), but the recall drops from 0.419 to 0.37, according to Experiment 1 and Experiment 3. This result indicates that answer entities from different documents for the same topic can complement each other and improve the precision which in turn improves the overall performance (F-measure). The

Table 13: Results of named entity recognition tools for answer entity extraction

	Precision	Recall	F-measure
Experiment 1: based line, evaluated by documents			
Overall	0.1030	0.4190	0.1440
Product	0.0120	0.2959	0.0230
Person	0.2480	0.5460	0.3370
Organization	0.0770	0.4110	0.1110
Web page	0.1148	0.3693	0.1551
Wiki page	0.0830	0.5204	0.1269
Experiment 2: Special parser for Wikipedia, evaluated by documents			
Overall	0.1083	0.4400	0.1500
Product	0.0127	0.3639	0.0240
Person	0.2588	0.5463	0.3454
Organization	0.0829	0.4241	0.1179
Web page	0.1148	0.3693	0.1551
Wiki page	0.0982	0.5501	0.1426
Experiment 3: evaluated by topics			
Overall	0.1700	0.3700	0.1600
Product	0.0293	0.3163	0.0495
Person	0.4449	0.4236	0.3555
Organization	0.1117	0.3646	0.1237
Web page	0.2060	0.2783	0.1704
Wiki page	0.1055	0.5275	0.1439

answer entities extracted from different documents do co-reference each other and improve the accuracy of the extraction. However, the recall drops because merging the results from different documents by topics reduces some rare but relevant answer entities with low scores.

Therefore, further work is to investigate how to improve extracting answer entities with rare existing.

With the special Wikipedia parser to remove some noise, the results of Wikipedia page are improved significantly (two-tail t-test, $p < 0.001$). Precision rises from 0.08 in Experiment 1 to 0.10 in Experiment 2, recall rises from 0.52 in Experiment 1 to 0.55 in Experiment 2, and F-measure rises from 0.13 in Experiment 1 to 0.14 in Experiment 2. That means narrowing down the context and removing the noise does help to improve the results. However, the results of the web page extraction (F-measure of Web page is 0.1551 in Experiment 1 and 0.17 in Experiment 3) are better than the ones from Wikipedia pages (F-measure of Wikipedia page is 0.1269 in Experiment 1 and 0.14 in Experiment 3). Especially, the precision in Experiment 1 (0.12 vs. 0.08) and in Experiment 3 (0.2 vs. 0.1) is higher but the recall in Experiment 1 (0.37 vs. 0.52) and in Experiment 3 (0.28 vs. 0.53) is lower. It is because the Wikipedia germane documents cover more information than the webpage germane documents, which can bring in more answer entities so recall is improved, but also bring the noises which cause precision drops.

Comparing the performance of three experiments according to different entity types, the results indicate that the extractions of organizations and persons (directly extracted from NER) are significantly better than products extracted from noun phrases. This means named entity tools are critical in this step. The approach of treating noun phrases as the products brings too much noise (the precision of products is only 0.01). For the entity type of organizations, even with some trained data and rules for extraction, the precision is still very low (0.08). Therefore, further work is needed to investigate answer entity extraction for the named entity recognizer non identifiable entities, such as products.

The precision of NER method is 0.17 (overall precision in Experiment 3), which needs to be further improved. In the next section, we use knowledge base method to improve the extraction precision by filtering the candidate entities with the entity categories from knowledge bases. The recall of NER is less than 0.5, which means this method misses half of important answer entities. In the next section, we will extract more answer entities from knowledge base, which is independent from the corpus, to improve the recall of the extraction.

6.1.2 Knowledge Base Entity Type Filtering

Another way to find the answer entities with target types without considering contexts is by relying on a knowledge base to detect the answer entity types. The named entity recognition tools only identify the high level categories, such as person, locations, and products. The same high level categories are used in the TREC task. However, the detail categories can be analyzed from the queries. For example, in the topic of airlines that Air Canada has code share flights with, the entity type is company, but in fact the detail entity type is the airlines. Although airlines can be treated as companies, they are a better description for the answer entity type. Therefore, it is better to take advantage of this description to extract the proper entities.

A knowledge base, such as Wikipedia, contains the category information which is useful to filter the unrelated entities and keep the answer entities with target types, which in turn improves precision and recall of the extraction. Wikipedia categories are applied as an example for knowledge base entity type filtering tasks. Wikipedia categories are human mark-up types or classes. For example, Antonio Meneses who is one of members of The Beaux Arts Trio is assigned such categories as 1957 births, living people, and Brazilian classical cellists in the Wikipedia, which indicates this entity is the type of person. The research question is how to detect the nominated entity type information from both the queries and the knowledge base categories. In order to match the entity types in TREC task to the categories in the Wikipedia, the following strategies are used.

- The entity type of person in TREC is mapped to the categories with “*** births”, or “living people”, or “*** deaths”.
- If the narrative is the structure of noun phrase followed by a noun clause, then choose noun phrase as the TREC entity types. For example, the term of “airlines” is detected as the entity type in the topic of airlines that Air Canada has code share flights with);
- If the narrative is the noun phrase with prepositions, then keep the noun phrase as the TREC entity types. For example, the term of “chefs” is detected as the entity type in the topic of Chefs with a show on the Food Network;
- If the narrative is the noun phrase with a modifier, then keep the noun phrase as the

TREC entity types. For example, the term of “journals” is detected as the entity type in the topic of journals published by the AVMA).

The experiment of the Wikipedia categories for entity type filtering evaluates whether the precision of answer entity extraction can be improved with the aid of knowledge base type filtering. It is performed on the 20 TREC 2009 topics and the related entity types are persons, products, and organizations. The candidate answer entities are extracted from Wikipedia germane documents. The assumption is that entries on the Wikipedia page with the links linking to the other Wikipedia entries are treated as candidate entities. The algorithm uses Wikipedia categories to filter out the non-related entries, and then the rest are the answer entities.

Table 14: Results of Wikipedia entity type filtering for answer entity extraction

Topic ID	Target Entity Type	# of Identified Entities	# of Correctly Identified Entities	# of Answers	Precision	Recall	F
9	Person	6	6	10	1	0.6	0.75
14	Person	42	0	37	0	0	0
17	Person	16	9	71	0.562	0.127	0.207
18	Person	38	6	14	0.158	0.429	0.231
4	Organization	34	7	11	0.206	0.636	0.311
6	Organization	8	1	7	0.125	0.143	0.133
7	Organization	17	4	4	0.235	1	0.381
12	Organization	56	20	33	0.357	0.606	0.449
15	Organization	16	12	12	0.75	1	0.857
Avg. of the extracted topics					0.377	0.505	0.369
Avg. of 20 topics					0.120	0.227	0.166

The results are as shown in the above Table 14. The precision and f-measure for the Wikipedia type filtering method are significantly better than the NER method, and the scores rises from 0.08 to 0.12 and from 0.127 to 0.166 respectively. But the recall drops from 0.4 to 0.2. If we only consider the topics with entities in Wikipedia, precision, recall

and f-measure are all significantly improved to 0.38, 0.5, and 0.37 respectively. The recall averaging 20 topics drops because the method only considers the Wikipedia pages as germane documents and misses the Web pages. This method works well when the germane documents are Wikipedia pages and the target types are marked by the Wikipedia too, such as the topic of airlines that currently use Boeing 747 air planes.

Currently, with the limitation of the system implement, this method can only deal with target entity type of persons and organizations but not products, which means this method highly relies on the matching algorithms between entity types and Wikipedia categorizes. If the algorithm fails at mapping, the result of extraction will be bad. For example, in the Wikipedia, the categories of Vaccines, Influenza vaccines, 2009 flu pandemic, and AstraZeneca are assigned to the entity of FluMist, which are hard to be matched to products.

6.1.3 Discussion

This section explores the methods for answer entity extraction without contexts. Two approaches are investigated. The first is answer entity extraction from NER tools, and the second is using knowledge base for the entity type filtering. According to the experiment results of NER tool extraction, the precision of this method is only 0.1, which is low accuracy for the extraction, and the recall of answer extraction is less than 0.5, which means this method still misses half of the important answer entities.

Filtering out the answer entities using Wikipedia categories improves the precision of the extraction. The current algorithm can only identify the persons and organizations, but not products. Although this method can significantly improve the extraction precision, it has little effect on recall because it is limited by the knowledge base. In the future, we will further investigate the methods on entity type detection from knowledge base in order to improve precision and recall of entity extraction.

Answer entity extraction without contexts assumes the query and the entity type are independent given the germane document. In fact, this assumption is not accurate. In the following section, we will remove this assumption and introduce the context for answer entities to improve the extraction.

6.2 SYMBOLIC CONTEXTS: TABLE/LIST EXTRACTION

Answer entity extraction in TREPM model is represented as $p(e|d, q, t)$. If we consider contexts c for the entities, then it is represented as $\sum_c p(e, c|d, q, t)p(e|c, d, q, t)$. It is not efficient to consider all contexts c , so we use the most relevant support contexts $c_{support}$ to approximate this estimate. Therefore, it is

$$p(e|d, q, t) = \sum_c p(e, c|d, q, t)p(e|c, d, q, t) \approx \sum_{c_{support}} p(c_{support}|d, q, t)p(e|c_{support}, d, q, t)$$

As we discuss in Chapter 3, contexts can be interpreted in several ways. This section considers the tables as entities contexts. Tables are ubiquitous in the Web environment. They are often used to present the structural data, such as the latest experiments results, the statistical data, and the TV show schedule, in a condensed and concise way. With the richness of information in the tables, it has been an importance source of answer entities. As discussed in Section 6.1, almost half of the answer entities in the TREC entity retrieval task exists in the tables or lists of the Web pages, as shown in Table 11. Moreover, because the named entity recognition tool usually is trained on the sentence syntax, it might fail at detecting entities from tables or lists. Therefore, in this section, we will consider the tables/lists as the contexts and focus on the answer entities extraction from tables and lists. That is,

$$p(e|d, q, t) = \sum_c p(e, c|d, q, t)p(e|c, d, q, t) \approx \sum_{c_{table}} p(c_{table}|d, q, t)p(e|c_{table}, d, q, t)$$

Although lists present information in a sequential order, they are considered as a concise means of arranging data without syntax surroundings. Therefore, this thesis treats lists as a special case of one-column tables, and the following discussions only mention about tables. Tables appear in print media, handwritten notes, computer software, architectural ornamentation, traffic signs and many other places. Some researchers work on the table extractions from plain texts [Ng, 1999], while some others extract tables from PDF files [Liu et al., 2006] and images [Pinto et al., 2003]. This thesis, as mentioned in previous section, only focuses on extracting tables from the HTML pages. For example, <http://www.foodnetwork.com>

/chefs/index.html is a germane document for the topic of chefs with a show on the Food Network, which covers answer entities in a list presented by the HTML “li” tags. The same idea of extracting entities from tables or lists also appears in [Fang et al., 2009].

6.2.1 Answer Entity Extraction from Tables/Lists

Extracting tables and their elements is a challenging task. The challenge comes from diverse physical structures representing the tables, no formal table designing rules/standards, distinct presentation schemes in different mediums, various table logistic contexts, diverse table cell types, many affiliated elements, etc. In order to characterize table extraction, Liu classifies the six types of table meta data [Liu et al., 2007]. It has the following three types of features with regards to the entity retrieval task in the Web:

- **Table Position Feature.** It records the medium types (HTML, PDF, image, PS, text, email, etc), the URL of table (the URL indicating where the table is located), the page title (the web page title shown in the web page, usually in a large font size), the web author, the web origination (the name of the website), the table starting position (the X and Y-axis coordinates of the starting place of the table), and the table frame meta data (left, right, top, bottom, all, none, etc.) These features can facilitate the table searching if users only know pieces of the document information or wish to restrict the search to certain types of documents.
- **Table Affiliated Feature.** It contains the table caption (the caption sentences appearing along with the table), the table caption position (the position of the caption with the body of the table: above or below), the table footnote (explaining the information in the table and usually appears below the table body), and the table reference text (the text referring to the table and discusses the content of the table).
- **Table Content Data.** It refers to the values as well as their data types in each cell of a table with the cell position information. It enables people to search tables based on the contents of their cells. For example, the content in $Cell(i, j)$ is the content in the cell that is located in the i th row and the j th column of a table. The data type in the cell can be numerical and/or symbolic. Because this thesis investigates entity retrieval, the

text strings will be our major extractions.

We adopt Liu’s algorithm of automatic table extraction for the answer entity extraction task [Liu et al., 2007]. The algorithm is composed of two steps: detecting the table candidates with table position meta data and its affiliated mete data; recognizing the table structures with table content data. Figure 8 shows the detailed pseudo-codes for these two steps.

```
Input: the original text
Output: table position metadata and affiliated metadata
Begin
  for each page do
    for each line do
      if symbols ∈ (the predefined table symbol list)
        collect the table position metadata and affiliated metadata
      end if
    end for each line
  end for each page
End

Input: the starting position of a table candidate and table position and affiliated metadata
Output: table content metadata
Begin
  nline ← read the first text piece in the candidate table
  Extract the row indexes from nline
  While (NOT at the end of a table)
    nline ← read the next line
    if (nline == new column in the same row)
      column++; adjust startX[column] and endX[column]
    end if
    if (nline == new line in the same cell)
      combining with previous lines in the same cell; adjust startX[column] and endX[column]
    end if
    if (nline == text pieces in the next row)
      row++; adjust startY[row] and endY[row]
    end if
    if (nline is special characters)
      combining with previous text line; adjust startX[column] and endX[column]
    end if
    else if
      table adds this cell
    end While
  End
```

Figure 8: The algorithm extracting answer entities from tables/lists

6.2.2 Experiment on Table/List Extractions from the Web Pages

The experiment evaluates the table/list answer entity extraction algorithm using the 20 topics from the TREC 2009 entity retrieval task (Appendix E). This experiment only works

on the HTML files. The goal of this experiment is to test whether the table/list extraction can help detect answer entities for the entity retrieval task. The experiment compares two approaches:

- Baseline system: named entity recognition (NER) tools for answer entity extraction. This experiment has done in previous section, so it will not repeat the results here.
- Experiment system: the table/list extraction for answer entity extraction. The table/list extraction algorithm is applied on germane documents to detect the answer entities.

The methodology of this experiment is as follows:

1. For each germane document, the algorithm is applied to extract the candidate entities from the tables/lists.
2. For all the candidate terms in the cells of tables/lists, a named entity recognition tool (e.g., Stanford NER) is used to identify the entities with their types. In this study, Stanford NER is applied and only identifies the types of persons and organizations.
3. Entities should be the ones starting with numbers or English letters, and the characters of “.” or “&” or “-” are only allowed to appear in the middle of terms.
4. The system removes some high-frequency entities. For example, the term of “sitemap” frequently appears in several different topics which, in fact, is not the answer entity. Similar to the idea of inverted document frequency, the common entities will tend to be over-emphasized, because it is low chance that one entity can be the answer for several different topics. Therefore, we will remove these common entities. In this study, the system removes candidate entities with the topic frequency larger than 3.

Table 15 lists the precision, recall and f-measure of the table/list entity extractions from germane documents based on the 20 topics of 2009 TREC entity retrieval task. The topics that are not in this table are those do not contain the tables or cannot extract any related entities.

Comparing answer entity extraction from tables/lists (Table 15) with answer entity extraction from NER tools (Table 13), the precision is significantly improved from 0.1 to 0.17 for averaging 20 topics (with two-tail t-test, $p < 0.001$). If we only consider the documents containing tables and related entities, the precision is even higher (0.69). This indicates the

Table 15: Results of table/list detections for answer entity extraction

Topic ID	Target Entity Type	# of Identified Entities	# of Correct Identified Entities	# of Answers	Precision	Recall	F
9	person	3	3	9	1	0.333	0.5
10	organization	6	4	13	0.667	0.308	0.421
12	organization	18	3	33	0.167	0.09	0.118
17	person	68	43	71	0.632	0.606	0.619
18	person	5	5	14	1	0.357	0.526
Avg of found entities					0.693	0.339	0.437
Avg of 20 topics					0.173	0.085	0.109

answers in the tables are good sources for the extractions, and the algorithms can achieve higher accuracy than the NER methods. The recall drops significantly from 0.4 to 0.08, the F-measure also drops from 0.14 to 0.11. This means this algorithm misses more answer entities than the NER method. Moreover, we find that only 5 topics can be extracted, instead of 18 topics which are expected containing the tables in the germane documents. One reason is that there are some tables/lists represented in images or pdfs or some other formats, which will be our future researches. The other reason is the complicated table structures make the extraction harder, and the current algorithm and its implement can not detect them.

6.2.3 Table/List Extractions from Knowledge Base

A knowledge base can be independent from the original corpus. Therefore, the extraction from a knowledge base only relies on the knowledge base itself. The knowledge base also organizes the information in the tables or lists, but with a more standard way. Although the tables in knowledge base can use the same extraction method mentioned in the previous section, it can be more precise in the knowledge base, which is described as the relation context. The relation context refers to the relation r and the associated topic entity e_1 and

the answer entity e_2 . According to the relation contexts, the query can be interpreted as the topic entity e_{q1} and the relation r_q . Therefore, with the TREPM model, answer entity extraction with relation contexts can be represented as following:

$$\begin{aligned}
 p(e|d, q, t) &= \sum_c p(c|d, q, t)p(e|c, d, q, t) \\
 &\approx \sum_{e_1, r, e_2} p(e_1, r, e_2|d, r_q, e_{q1})p(e_{q2}|e_1, r, e_2, d, r_q, e_{q1}) \\
 &\approx \sum_{e_1=e_{q1}, r=r_q, e_2} p(e_1, t_1, r, e_2|d, e_1, r, e_2)p(e_{q2}|e_1, r, e_2, d)
 \end{aligned}$$

Wikipedia Infobox is one of the knowledge bases and is used to demonstrate the extraction process. It extracts high accuracy answer entities but is not limited by corpus and increases the recall of answer entity extraction. Figure 9 is a sample of a Wikipedia Infobox. Noisy knowledge is one of the problems in using knowledge bases for answer entity extraction. The study of Wu illustrates that the knowledge base like Infobox need to be further cleaned for extractions [Wu and Weld, 2008]. In the pilot study of extracting company-product pairs from Infobox, there are two of twenty company cases (10%) where the “product” fields in the Infobox pages contain links to other pages instead of the product information itself. Another problem is the incompleteness of its knowledge. For example, three of the twenty company pages do not contain information of products (15%), where one case has no product field in its Infobox and the other two have not Infobox fields at all.



Type	Public (NASDAQ: GOOG ↗ , FWB: GQQ1 ↗)
Industry	Internet, Computer software
Founded	Menlo Park, California (September 4, 1998) ^{[1][2]}
Founder(s)	Sergey M. Brin Lawrence E. Page
Headquarters	1600 Amphitheatre Parkway, Mountain View, California, United States
Area served	Worldwide
Key people	Eric E. Schmidt (Chairman & CEO) Sergey M. Brin (Technology President) Lawrence E. Page (Products President)
Products	See list of Google products.
Revenue	▲ US\$23.651 billion (2009) ^{[3][4]}
Operating income	▲ US\$8.312 billion (2009) ^{[3][4]}

Figure 9: A sample of Infobox

Because of the incompleteness and the complexity of knowledge base, the algorithm of answer entity extraction from a knowledge base first has to detect the related topic entities. For example, for the query of products of Medimmue Inc, the algorithm needs to find the

correct entry of the topic entity, i.e., MedImmune Inc. It is to implement the detection of $e_1 = e_{q1}$ in the formula. Secondly, in the topic entities related attributes, the algorithm identifies whether this targeted entity has attributes associated with the queried relations, e.g. products. This is the detection of $r = r_q$. The last step is to identify the entity instances acting as the attributes of topic entities, e.g., the value of FluMist for the attribute of products. If the attribute fields in Infobox are not directly extractable, further mining steps need to associate to related pages in order to extract the answer entity information. The overall algorithm of mining the Wikipedia Infobox for extracting answer entities is in Figure 10.

```

For each target type (e.g., company) in the knowledge base (e.g., Wikipedia Infobox){
  Get related part (e.g., InfoBox) {
    Get target field (e.g., location) {
      Extract target information {
        If (field is terms), then extract terms as they are (e.g., products)
        If (field needs to be further extracted, e.g., "List of Google Products" or "Yahoo Products")
          Further extraction method{
            e.g., crawl_this_page{
              If the page contains LIST information, then extract them as products;
              If the page contains the links to another page, crawl_this_page
            } //end of further_extraction_method
          } //end of extract_target_information
        } //end of get_target_field
      } //end of get_related_part
    } //end of all
  }
}

```

Figure 10: The algorithm extracting entities from knowledge bases

The first experiment evaluating answer entity extraction of the tables/lists from knowledge base is on RAP sets. Topics are the companies, and the targeted entities are the products and locations of those companies. There are 265 entities of products extracted for 30 companies from Wikipedia 2008 version. The results are in Appendix B. The knowledge base entity extraction can effectively extract most answer entities with high precision for the experimental entities.

The second experiment is the answer extraction for the TREC 2009 20 topics. Because there are only 3 topics related to product retrieval, this experiment only uses these three topics. There is only one topic out of three to be extracted. That is, Synagis and FluMist are extracted as products for the topic of products of MedImmune, Inc.

Although this method can extract the high accuracy answer entities, which are independent on the noisy corpus, the extraction still relies on the knowledge base itself and the representation of knowledge. For example, the topic of airlines that currently use Boeing 747 planes uses the term of “primary users” in the Wikipedia Infobox of Boeing 747 pages to represent the relation between the topic entity and answer entity. The various representation causes difficulty in the matching of topic entities and answer entities. How to expand the algorithm to extract more entities will be the future work.

6.2.4 Discussion

The list/table extraction method can successfully detect answer entities for such topics as chefs with a show on the Food Network. However, there are some topics that are still hard for extractions because of the complicated structures of lists and tables.

1. Tables/lists can be embedded into the pictures or photos. For example, for the topics of sponsors of the Mancuso quilt festivals, the lists on the Web page are the logos for these companies with the links pointing to these companies. This kind of representation is popular on the Web to avoid robots mining the web contents, but it also causes our difficulties in the entity extraction. Similar cases include the topic of donors to the Home Depot Foundation, whose answer entities are in one picture, which cannot be extracted by the text extraction.
2. The hierarchical or mixture structures of lists or tables also cause difficulty in extraction. The answer entities for the topic of authors awarded an Anthony at Bouchercon in 2007, for example, are listed in lines combining the authors and the title of a work of fiction together, which causes the difficulty of extractions.
3. Various formats to present the list structure also cause the failure of extraction. For example, for the topic of sport teams in Philadelphia, the Wikipedia page, http://en.wikipedia.org/wiki/Sports_in_Philadelphia.html, uses HTML heading to represent the answer entities. For the topic of donors to the Home Depot, the Webpage of the Home Depot, <http://www.homedepotfoundation.org/donors/2010-complete-donor-list.html>, uses the way of each line per donor to present the answer entities. Therefore, the algorithm

should further consider the various presentation of the table/list structure for the extraction.

4. Sometimes, there are multiple tables in the document, but not all the tables or lists in the germane documents discuss the answer entities and only tables or lists in some sections present the answer entities. For example, for the topic of the journals published by AVMA, the germane document of <http://www.avma.org/journals/default.asp> contains the answer entities in the “journals” section which is mixture with others. Therefore, how to identify the answer sections in the germane documents will be our future work.
5. Various names are needed to represent the relations between answer entities and the topic entities. The researches on matching the relation names in the ceiling header of the tables and the queries will be our future work.

6.3 SYNTACTIC CONTEXTS: BOOTSTRAPPING

The studies on the bootstrapping method for answer entity extraction were originally published in [Li et al., 2009]. Answer entity extraction with contexts of the subject-verb-object structure, which exists in the sentence syntax and is called patterns in this study, is investigated. Therefore, we discuss syntax contexts for answer entity extraction.

As we discuss in Section 3.4, in syntax contexts, every query can be represented as a binary relation r_q between the topic entity e_{q1} with the type of t_{q1} and the target entity e_{q2} with the type of t_{q2} . Therefore, the entity retrieval task is to retrieve the e_{q2} given $e_{q1}, t_{q1}, r_q, t_{q2}$. Similarly, the context can also be represented as the triplet of an entity e_1 with the type t_1 , an entity e_2 with the type t_2 , and their relations r , that is, $c = (e_1, t_1, r, e_2, t_2)$. Therefore, answer entity extraction estimates the probability of e_2 to be the e_{q2} as the

following equation.

$$\begin{aligned}
p(e|c, d, q, t) &= \sum_c p(c|d, q, t)p(e|c, d, q, t) \\
&\approx \sum_{e_1, t_1, r, e_2, t_2} p(e_1, t_1, r, e_2, t_2|d, r_q, e_{q1}, t_{q1}, t_{q2})p(e_{q2}|e_1, t_1, r, e_2, d, r_q, e_{q1}, t_{q1}, t_{q2}) \\
&\approx \sum_{e_1=e_{q1}, t_1=t_{q1}, r=r_q, e_2, t_2=t_{q2}} p(e_1, t_1, r, e_2, t_2|d, t_1, r, e_2, t_2)p(e_{q2}|e_1, t_1, r, e_2, t_2, d)
\end{aligned}$$

The first quantifier, $p(e_1, t_1, r, e_2, t_2|d, e_1, t_1, r, e_2, t_2)$, reflects given the document, the probability of existing the context c containing the requirement of query q . And the second quantifier $p(e_{q2}|e_1, t_1, r, e_2, t_2, d)$ is to estimate the probability of answer entity extracted from the contexts.

The syntax context is hard to detect not only because it is hard to fully understand the sentences or list complete semantic contexts for the answer entity extraction task, but also because it is hard to have huge human resources annotating training sets for supervised learning method. Syntax contexts can be extracted using the deep sentence analysis. For example, Li uses the sentence dependency analysis of relation extractions for the image retrieval [Li and He, 2011a]. However, the sentence structure in the Web environment is more complex than the image meta data. Therefore, a semi-supervised method—a bootstrapping algorithm—is introduced for the entity extraction task with aid of the syntactic contexts. The hypothesis of bootstrapping method is: with the high quality topic-answer entity pairs (e.g. Apple Inc. and iPad), the system can detect high qualified contexts (e.g., the pattern of “launch” for the sentence of “Apple Inc. launches iPad”) containing these entities; then the high qualified contexts are used for further answer entity detection (e.g., “Apple Inc. launches iPhone).

6.3.1 Bootstrapping Algorithm

The bootstrapping algorithm learning the syntactic contexts for entity extraction tasks concentrates on the extraction of the topic-answer entity pair according to the pattern of the subject-verb-object (SVO) structure.

Figure 11 shows an example work flow of bootstrapping method for entity extraction. The inputs of the bootstrapping algorithm are some well qualified topic-answer entities seeds. These seeds can be obtained from knowledge bases, which take advantages of the well defined structures or the schema of knowledge bases, as in Section 6.2.3. The core of the bootstrapping algorithm is pattern generation and ranking, which is the basis of building a trained model for entity extraction. One of key problems in pattern generation is how to accurately identify the patterns that can effectively predict the topic-answer entity pairs for the later entity extraction task. Another is how to choose the most effective one from large numbers of learned potential patterns. This idea of bootstrapping method—beginning from a good seed then generating the patterns and further extracting more seeds with the aid of the patterns—has been widely shared in web mining communities. For example, KNOWITALL used a bootstrap method to extract generic named entities by using the Web as the source for training, and avoided hand-labeled training examples [Etzioni et al., 2005]. Pasca extracted facts and named entities from the Web using patterns learning from training sets [Pasca et al., 2006]. In this section, we use the same method on answer entity extraction.

6.3.1.1 Pattern Generation and Pattern Weighting Based on the extracted seeds, the bootstrapping method tries to infer patterns that cover the extracted seeds. It uses the Web as the corpus for generating patterns by querying (e.g., Yahoo!BOSS) the Web (as Step_1 in Figure 11). The results returned from the search engine usually cover several pieces of information such as a page title, the URL of the page, and a short summary of the page content (as Step_2 in Figure 11).

The bootstrapping method in this work uses the subject-verb-object (SVO) structure for patterns extractions from search results. Patterns are identified by the key verb with two entities in the sentence. For example, in the sentence of “EBay launches Kijiji”, the system first identifies two important entities (the company named EBay and the product named Kijiji) as well as the verb (launches) in the sentence, and then marks the sentence as “Ebay.COMPANY launches Kijiji.PRODUCTS”. Therefore, the pattern is extracted as “COMPANY launch PRODUCT”, which is Step_3 in Figure 11.

As Brin points out, the quality of extracted entities highly correlates to the quality of the

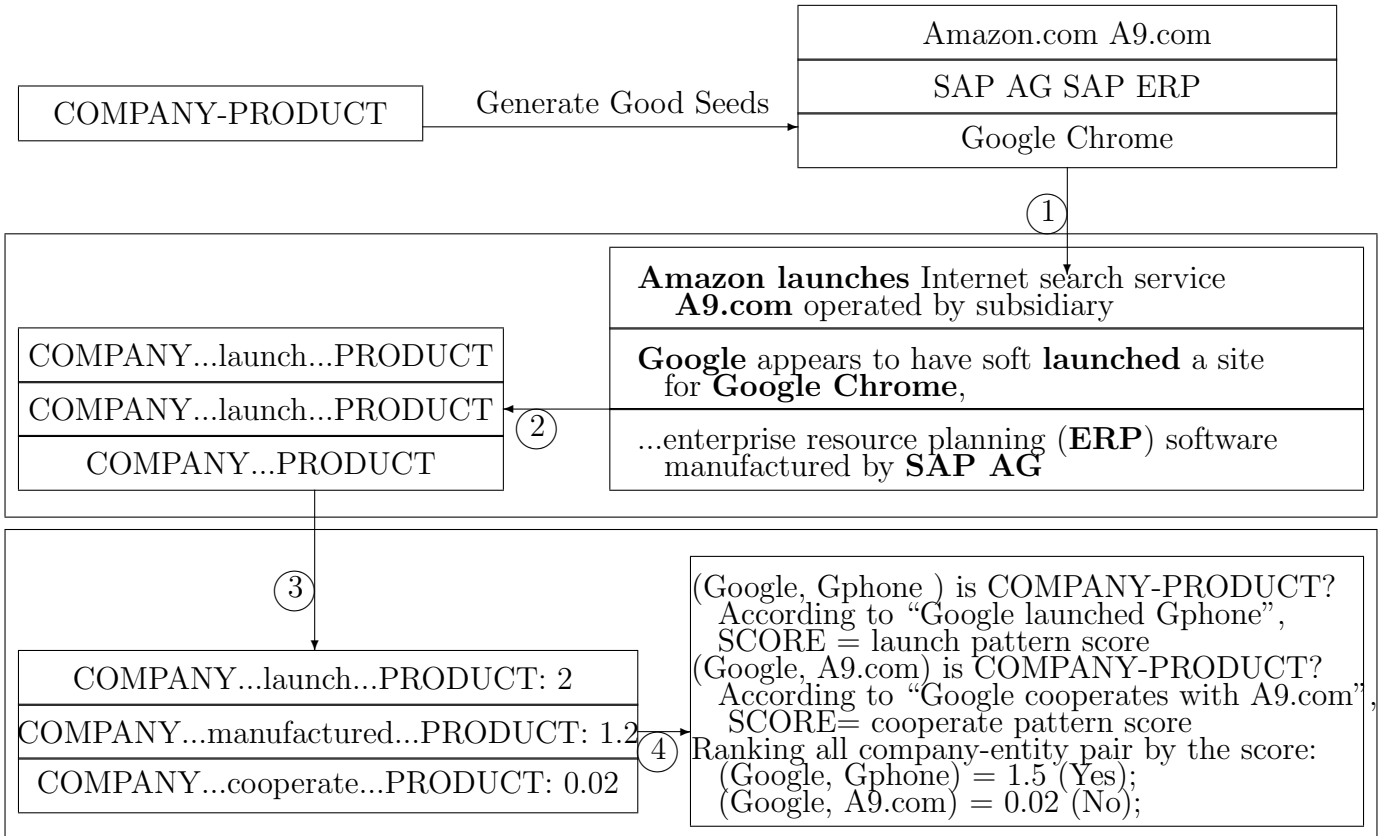


Figure 11: Bootstrap framework of topic-answer entity pair extraction

extracted patterns [Brin, 1999]. Some researchers investigate in the good quality patterns. Ravichandran uses frequency threshold to select the patterns with the assumption that high frequency patterns are correlates to good entities [Ravichandran and Hovy, 2002]. However, it is not necessary. Low frequency patterns could also be useful. For example, the frequency of the “produce” pattern (e.g., in the sentence of “shell holds acreage with potential to produce shale gas”) is lower than the frequency of “be” pattern (e.g., in the sentence of “royal Dutch Shell, commonly known as Shell, is a global oil and gas company”), but the “produce” pattern is better than the “be” pattern for its high precision, because the “be” pattern can also be such sentence as “a shell is a piece of software that provides.” Therefore, ranking the relevance between patterns and topic-answer entity pairs is an complicate but important task here. We propose five different weighting schemes to rank patterns are

evaluated as follows. This is shown as Step_4 in Figure 11.

1. Frequency Weight.

Frequency Weight (FW) assumes that the higher the frequency of a pattern is on the Web, the better its quality is [Ravichandran and Hovy, 2002]. Stemming is used to improve the coverage of the method. This work defines FW as Equation 6.1.

$$FW(v) = \frac{|x, v, y|}{\sum_v |x, v, y|} \quad (6.1)$$

where $|x, v, y|$ denotes the frequency of the pattern with term x , term y and pattern verb v in the same window size. In this work, the window size is within the same sentence.

2. Distance Weight

Distance Weight (DW) denotes the word distance between two entities, as shown in Equation 6.2.

$$DW(V) = \frac{1}{\text{number_of_word_between_}(x, y)} \quad (6.2)$$

3. Verb Distance Weight Verb Distance Weight (VDW) represents a special case of DW which examines the distance between verb and target/answer entity (Equation 6.3).

$$VDW(V) = \frac{1}{\text{number_of_word_between_}(v, y)} \quad (6.3)$$

4. Frequency-Distance Weight

Frequency Distance Weight (FDW) combines the distance weight and the frequency weight, which is defined in Equation 6.4:

$$FDW(V) = DW(V) \times FW(V) \quad (6.4)$$

5. PMI

Pointwise Mutual Information (PMI) is a commonly used metric for measuring the connections between two events. Pantel and Penacchiotti used PMI to evaluate the reliability between patterns and instances [Pantel and Pennacchiotti, 2008]. We adopt PMI as a weight for the pattern ranking, and at the same time, PMI is also used as a baseline for evaluating the weights mentioned above. PMI is defined as Equation 6.5.

$$PMI(v) = \text{avg}_e \frac{\text{pmi}(e, v)}{\text{Max}(\text{pmi})} \quad (6.5)$$

where $Max(pmi)$ is the maximum pmi of all patterns v and all instances e . And pmi is defined as follows:

$$pmi(e, v) = \log \frac{|x, v, e|}{|x, e||v|}$$

whereas x, v, e is the frequency of the pattern v instantiated with term x and term e . $|x, e|$ is the frequency of x and term e co-occurrence together; $|v|$ is the frequency of term verb v .

6.3.1.2 Pattern Matching Strategy The output of the bootstrapping method is a ranked list of answer entity instance pairs. For example, the extraction output of company-location can be the pair of “Google, Menlo Park.” The assumption is that the system has named entity tool to help identify the two entities in each pair. The bootstrapping method would then annotate whether two entities have the relationships according to the pattern, as shown in Step_4 in Figure 11.

One problem for entity identification in entity extractions is co-reference, which is when two different mentions could co-refer to the same entity. Failure of identifying the co-reference entities directly cause the failure of identifying entity pairs. For example, at the beginning of the documents, the author mentions “MedImmune, LLC, headquartered in Gaithersburg, Maryland”. Later, the document said “It produces Synagis”. In fact, “MedImmune, LLC” and “it” co-reference the same entity of “MedImmune, LLC”. However, due to the lack of a co-reference tool, the current method can not handle co-reference. In order to overcome this problem, a matching strategy that relies on matching to just one entity is used in this work. This approach is motivated by Yarowsky’s work in word sense disambiguation that stated “one sense per collocation” [Yarowsky, 1993]. It assumes that there is only one topic entity in the window size of one document. Therefore, the topic entity x with the answer entity y as well as the pattern verb p denoted as $|x, p, y|$ will be the topic entity throughout a document; and the matching process only concerns the finding of the verb p and entity y . I ran pilot studies to test the assumption that the most frequently appearing entities (e.g., the company) in a document could be the topic entity for the whole documents. There are eighty-eight articles extracted from CNET news for topic entities of companies. It showed that this method only failed at three of eighty-eight articles at identifying the topic entities

of companies (95.5%). The two matching strategies are summarized as follows:

- **One-entity matching:** to only match verb pattern p with the answer entity y , and use the default topic entity x in the same document.
- **Two-entity matching:** to match verb pattern p as well as the topic entity x and the answer entity y .

6.3.2 Experiments on Company-Product and Company-Location

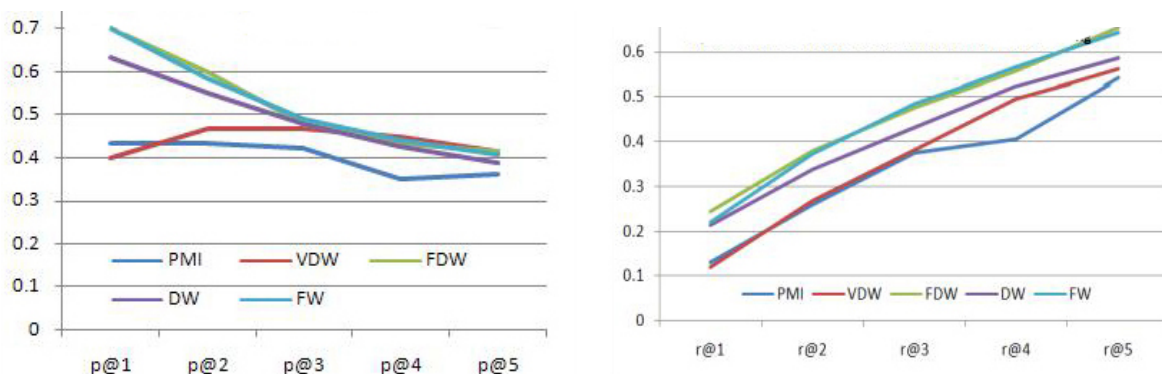
Two topic-answer entity pairs, company-location and company-product, are considered to evaluate the performance of the bootstrapping method for answer entity extraction. The company-location entity pair is chosen to represent the topic-answer entity pairs with the identifiable answer entity (e.g. location), and the company-product entity pair is used to represent the entity pairs with the NER-non-identifiable answer entity (e.g. product). The performance of five weights (frequency weight, distance weight, verb-distance weight, frequency-distance weight and PMI) and two matching methods (one-entity matching and two-entity matching) are evaluated. In the experiment, the high-qualified seeds from knowledge bases are used as the inputs for the bootstrapping methods. The experiment focuses on the evaluation of pattern generation to build up models for topic-answer entity pairs.

Named entity extraction tool from Inight LinguistX Platform developed by Business Object is used for the entity identification. Yahoo!Boss Search (short as Boss) is used for querying the Web in the experiment. Twenty-five target company articles from Wikipedia distributed in five industries (according to Fortune 500, 2008) are chosen for experiments as testing sets for both company-location and company-product entity extractions. In Nasdaq100 index, thirty one companies with company-product and company-location pairs are extracted from Infobox as seeds for training. Ground truth is manually marked up by two experts. Precision and recall are used for the evaluation.

The first experiment is the extraction of the company-location pair, which represents the entity pairs with identifiable entities. There are 251 company-location pairs extracted as the seed pairs. These pairs as queries are issued to Yahoo!Boss, and 12,103 search results (hits) are retrieved with 50 results per query. Relying on the SVO pattern extractions, there are

2,203 SVO patterns identified. Twenty-five Wikipedia articles are used for the evaluation.

Figure 12(a) & 12(b) are the precision and recall graph for the company-location pairs with one-entity matching respectively. Since the average number of company-location pairs in the documents is about 3, only the top 5 locations are evaluated. For one-entity matching (Figure 12), there is no significant difference between PMI and verb-distance weight as well as frequency-distance weight and frequency weight in either precision or recall. Distance weight is significantly better than verb-distance weight in both precision and recall. The recall of frequency-distance weight is significantly better than distance weight; and precision and recall of PMI is worse than the other four weights. Similar experiments are conducted on two-entity matching, and there is no significant difference between one-entity and two-entity matching for five groups in precision and recall by running a T-test.



(a) Precision of Company-Location Extraction

(b) Recall of Company-Location Extraction

Figure 12: Bootstrap results for company-location extraction with One-Entity matching

Therefore, frequency weight and frequency-distance weight are better than distance weight and verb-distance weight for the topic-answer entity pairs with an identifiable entity. Moreover, all four weights (frequency weight, distance weight, verb-distance weight, and frequency-distance weight) are better than PMI. Matching methods—one-entity and two-entity matching—have no effects for the entity identification with an identifiable entity.

The second is the extraction of company-product pair, which represents the entity pairs with non-identifiable entities. There are 265 company-product pairs extracted from Infobox. These pairs as queries are issued to Yahoo!Boss with 50 results per query. There are 13,250

hits returned for pattern analyses. There are 3,653 SVO patterns extracted for training. The same twenty-five Wikipedia articles about companies are the testing sets for company-product entity extractions. Figure Figure13 shows the results. The precision of pairs with one-entity and two-entity matching are in Figure13(a) and Figure 13(b) respectively, while the recall of pairs with one-entity and two-entity matching are in Figure13(c) and Figure 13(d) respectively.

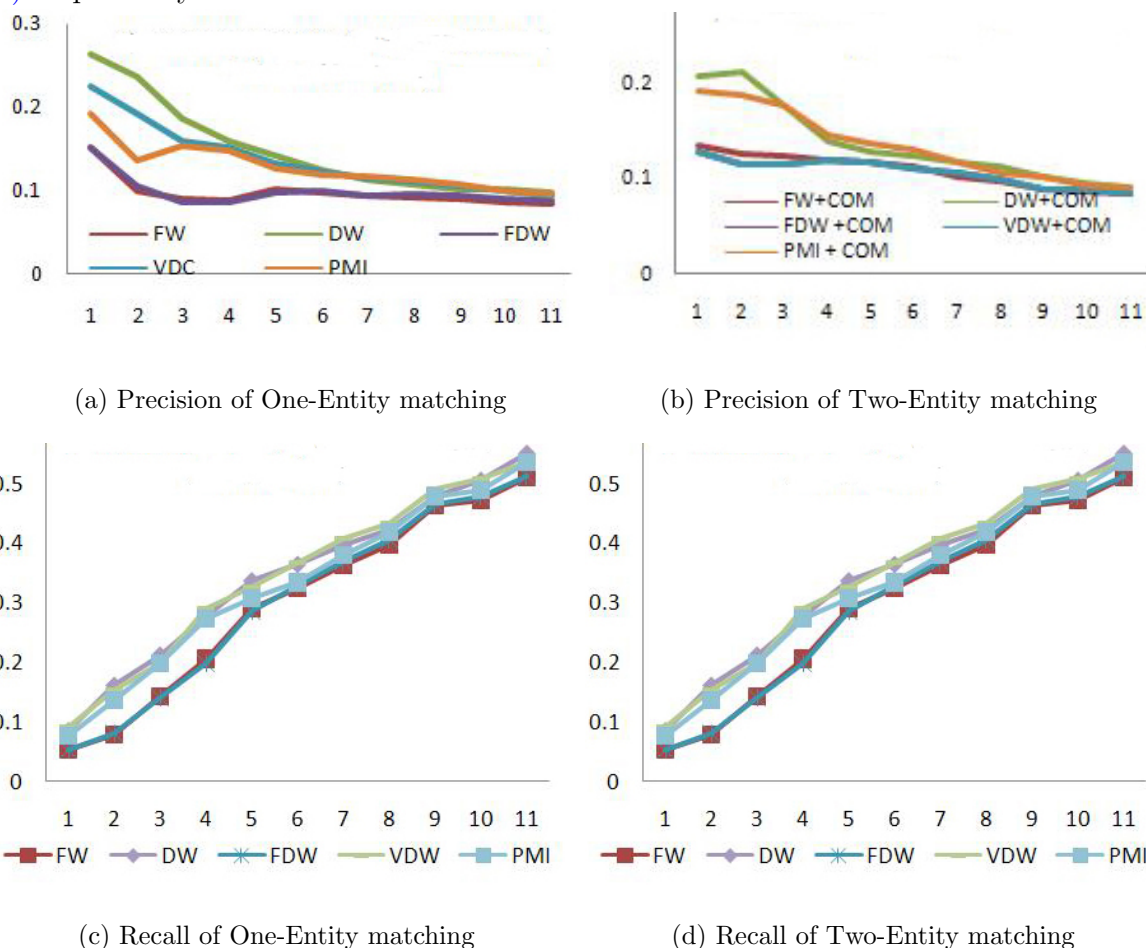


Figure 13: Bootstrap results of company-product extraction

For the one-entity matching, there is no significant difference between frequency weight and frequency-distance weight as well as PMI and verb-distance weight for both precision and recall in the extraction of Company-Product pairs. Verb-distance weight is significantly better than frequency-distance weight for both precision and recall. Distance weight is significantly better than verb-distance weight in precision but not in recall. For the two-entity matching, frequency-distance weight is significantly better than frequency weight; and

verb-distance weight and frequency-distance weight have no significant difference; PMI and distance weight have no significant difference also, but both are better than verb-distance weight.

Both frequency weight and frequency-distance weight of two-entity matching are significantly better than one-entity matching in precision. For distance weight and verb-distance weight, however, the precision of one-entity matching is significantly better than two-entity matching. The difference between frequency weighting and distance weight is that distance weighting considered the distance of verb and the other entity, that is, it is including the sentence syntactic information, while frequency weight only considered the frequency of the verb without any syntactic information at all. Therefore, syntactic information is very useful for entity extractions with a non-identifiable entity.

6.3.3 Experiments on Twenty Topics of TREC 2009 data set

This experiment is to evaluate the performance of the bootstrapping method on the twenty topics of TREC 2009 data for answer entity extraction. The assumption of the method is that given the germane document, if we have good quality seed pairs (topic-answer entities), then we can find the similar answers (answer entities) for particular topics. We assume the topic-answer entities extracted from Wikipedia Infobox are the high quality seeds, which can use bootstrapping method for further answer entities extraction.

In the 20 topics from the TREC 2009 entity search set, only the topic of products of MedImmune Inc has the answer of Synagis and FluMist extracted from the Wikipedia Infobox. Therefore, in this experiment, we use the pairs of “MedImmune Inc. FluMist”, “MedImmune Inc. Synagis”, and “MedImmune Inc. FluMist Synagis” for the extraction.

The experiment steps of the bootstrapping method for answer entity extraction are as follows:

1. Generating good quality seeds. The pairs of “MedImmune Inc. FluMist”, “MedImmune Inc. Synagis”, and “MedImmune Inc. FluMist Synagis” are used to search on the Google search engine, and the top 80 hit results per topic are collected as a pool for pattern generation. Germane documents are processed with part-of-speech, and the

verb between the subject (MedImmune Inc.) and the object (FluMist or Synagis) are extracted. Two kinds of verb structures are considered. One is the single verb in the verb structures, e.g., the single verb of “make” in the sentence of “COMPANY makes a drug called PRODUCTS”. The other one is the terms between the subject and object, e.g., “makes a drug called” for the same sentence mentioned above. Moreover, the verb pattern weights are calculated using the frequency-distance weight, because in the previous experiment, the frequency-distance weight is best for the entity pairs with non-identifiable entities. Nineteen verb patterns are generated, as shown in Table 16.

2. With 38 verb patterns and the subject (i.e., MedImmune, Inc.), our system generates another 38 queries issued to the Google engines in order to pool the relevant answer sets. Two strategies are applied to generate the queries. One is from the single verb structure, which is only the subject and single verb, e.g., “MedImmune, Inc. make”. The other is the subject and the verb structures between the subject and object, e.g., “MedImmune Inc. makes a drug called” The 80 hits are returned from each query, which generate the pool of sentences with candidate answer entities.
3. According to the subject-verb-object structure, the sentences in the pools with candidate answer entities are pre-processed with POS taggers. Two types of structures are considered. The first noun phrase after the subject-verb structure is extracted as candidate answer entities with their scores according to the verb weight. For example, the term of “FluMist” in the sentence of “MedImmune, Inc. make FluMist” should be detected as products. The second structure is last noun phrase before the verb-subject structure. For example, the term of “Synagis” in the sentence of “MedImmune Inc. makes a drug called Synagis” should be detected as products.
4. The candidate answer entity list is generated and ranked according to their weight, which combines the pattern weight and the entity’s verb-frequency weight.

The top ten candidate answer entities extracted for this topic are listed in Table 17.

The extracted top five results are the answer entities. It indicates that with the high quality seeds as well as the pattern generate model and weighting system, the bootstrap method can extract the highly accurate answer entities. For the topic of “products of MedImmune Inc.”, the system extracts not only the products but also some variations of product

Table 16: Patterns for extracting the company-product pair

Pattern	Weight
COMPANY/NNP acquired/VBD PRODUCTS/NNP	2
COMPANY/NNP produces/VBZ PRODUCTS/NNP	2
COMPANY/NNP Says/VBZ PRODUCTS/NNP	1
PRODUCTS/NNP helps/VBZ lead/VB COMPANY/NN	1
PRODUCTS/NNP impacts/VBZ COMPANY/NN	1
PRODUCTS/NNP is/VBZ COMPANY/NN	1
PRODUCTS/NNP is/VBZ a/DT registered/JJ trademark/NN of/IN COMPANY/NN	0.8
COMPANY/NNP makes/VBZ injectable/JJ PRODUCTS/NNP	0.5
PRODUCTS/NNP ,/, made/VBN by/IN COMPANY/NN	0.5
PRODUCTS/NNP is/VBZ a/DT nasal/JJ spray/NN influenza/NN vaccine/NN manufactured/VBN by/IN COMPANY/NN	0.5
PRODUCTS/NNP is/VBZ produced/VBN at/IN a/DT /NNP COMPANY/NN	0.5
PRODUCTS/NNP -LRB-/-LRB- palivizumab/NN -RRB-/-RRB- is/VBZ a/DT registered/JJ trademark/NN of/IN COMPANY/NN	0.4
COMPANY/NNP expects/VBZ peak/JJ annual/JJ PRODUCTS/NNP	0.33
PRODUCTS/NNP ,/, made/VBN by/IN jbi/NNP COMPANY/NN	0.33
COMPANY/NNP ”/” -RRB-/-RRB- ,/, the/DT petitioner/NN ,/, manufactures/VBZ the/DT drug/NN PRODUCTS/NNP	0.25
COMPANY/NNP is/VBZ best/RB known/VBN for/IN two/CD products/NNS -/: PRODUCTS/NNP	0.2
COMPANY/NNP ,/, which/WDT already/RB makes/VBZ the/DT nasal/JJ spray/NN vaccine/NN PRODUCTS/NNP	0.167
COMPANY/NNP makes/VBZ a/DT drug/NN called/VBN PRODUCTS/NNP	0.167
COMPANY/NNP markets/VBZ four/CD products/NNS PRODUCTS/NNP ,/, Ethiol/NNP ,/, PRODUCTS/NNP	0.167

Table 17: Results of the bootstrapping method for answer entity extraction

Candidate Entities	Score	Candidate Entities	Score
FluMist®	8.5	Vitaxin	1.14
Synagis	8.08	Aprimo	0.90
FluMist	3.64	Ethyol	0.57
Synagis®	3	Flumist	0.5
Ethyol®	2.86	FlumistÅ	0.5

names, and some medicines under development, such as Vitaxin. The only error of the top 10 results comes from the Aprimo, which is the company helps to market MedImmune Inc.

6.3.4 Discussion

This section investigates the syntax contexts for answer entity extraction. The bootstrapping mining method on the web is used for syntactic context identification. High quality seeds extracted from knowledge bases are the input to the bootstrapping method. Patterns based on the subject-verb-object structures generate the syntactic contexts for topic-answer entity pair extraction.

In order to find patterns, five weight schemes (frequency weight, frequency-distance weight, distance weight, verb-distance weight, and PMI) and two matching strategies (one-entity and two-entity matching) are investigated. The experiments evaluate two types of answer entity pairs: pairs with identifiable entities (e.g., locations) and pairs with a non-identifiable entity (e.g., products). The experiment shows that frequency weight and frequency-distance weight are better for identifiable entity extraction (e.g., locations), while distance weight and verb-distance weight are better for non-identifiable entity extraction (e.g., products). For the matching strategies, although one-entity matching can compensate the problems caused by the lack of tools to detect the entities co-referring each other, as shown in the results, this matching strategy does not work for the non-identifiable entity

extraction, because one-entity and two-entity matching has no significant difference on extracting entity pairs with a non-identifiable entity. One-entity matching, however, is better in frequency weight, verb-distance weight, and PMI for the NER-identifiable entity extraction.

The capability of the bootstrap method in the answer entity extraction task is limited. It can achieve high accuracy for the entities with good quality seeds, for example, the topic of products of MedImmune Inc. However, it closely relies on whether the topics have good quality seeds. Only one out of twenty TREC 2009 topics can be found the good seeds. Therefore, its capability is limited. Another difficulty for answer entity extraction with syntax contexts is that some topics in the entity retrieval task have unique germane documents containing the answer entities. Therefore, it is hard to directly extract the answers from knowledge base, such as Wikipedia Infobox. For example, it is hard to find good seeds for the topic of donors to the Home Depot Foundation. In the future, more work will be done on finding the good quality seeds for answer entity extraction.

6.4 ANSWER ENTITY EXTRACTION AS A CLASSIFICATION PROBLEM

The main role of answer entity extraction is to extract the correct answer entities from the germane documents with high accuracy for entity retrieval. The named entity recognition tools as a common approach to extract answer entities are investigated in Section 6.1. However, the precision and recall for this approach are low (overall $P=0.1$, $R=0.4$, $F=0.144$ in Table 13). The knowledge-based entity extraction which is independent from the search corpus is introduced to improve the extraction. It can achieve high precision on the existing answer entity detection, but the overall recall is low (extracted topics $P=0.4$, $R=0.5$, $F=0.4$ in Table 14), especially when the information asked by the topics is limited by the knowledge bases (overall $P=0.12$, $R=0.22$, $F=0.17$ in Table 14). For example, the topic, such as students of Claire Cardie, which has no records in the knowledge base, such as Wikipedia, will fail at this approach. Considering the contexts, the table/list extraction as symbolic contexts are investigated for answer entity extraction. It can improve the precision of the

extraction, but the recall of the extraction drops down (extracted entities P=0.7, R=0.3, F=0.4 in Table 15). The bootstrapping method can help to solve some problems from the limitation of knowledge base by learning the patterns from the Web to extract more answer entities from the corpus. For example, for the topic of products of MedImmune, Inc, the products of Syngas can be extracted from knowledge bases correctly, which can be the good seeds for bootstrapping method. Then, relying on these good seeds, bootstrapping methods extract more products of FluMist according to the similar structure of “MedImmune, Inc. produces FluMist.” However, sometimes, it is hard to get the good sample seeds and collect high quality patterns.

Although each method mentioned above has its own disadvantages, they also have their own advantages. In order to take advantage of these methods, and complement the limitations, we aim at generating a generic entity extraction model, which can combine the above methods together for answer entity extraction, i.e., treating answer entity extraction as a binary classification problem. With the candidate answer entity e , the system decides whether the entity e is the answer entity to the query q in germane documents d with target entity type t . The classifier will learn the parameters α , β , and θ for extractors. Later, the learned model will be applied to detect answer entities on the testing set.

$$\begin{aligned}
p(e|d, q, t) &= \sum_c p(c|d, q, t)p(e|c, d, q, t) \\
&= \alpha \sum_{c_{table}} p(c_{table}|d, q, t)p(e|c_{table}, d, q, t) \\
&+ \beta \sum_{c_{relation}} p(c_{relation}|d, q, t)p(e|c_{relation}, d, q, t) \\
&+ \theta \sum_{c_{syntax}} p(c|d, q, t)p(e|c, d, q, t) \\
&+ (1 - \alpha - \beta - \theta)p(e|c, d, q, t)
\end{aligned} \tag{6.6}$$

In order to learn a system, a variety of features that reflect the characteristics of entity extractions are generated, and then the SVM algorithm are applied.

6.4.1 Answer Entity Extraction: a Binary Classification Problem

Applying machine learning methods to answer entity extraction raises the questions of what types of information should be used in the learning process. Many different types of information can contribute toward deciding the answer entities. Two principles are followed in the process of feature selections:

- The feature should not be limited by the instances.
- The feature should be general enough and domain independent so that the model could be generalized to other topics regardless of the domain.

There are 15 features generated for answer entity extraction, which includes the basic features (like document id and topic id and target entity types), and the features indicating whether the entities are extracted from Wikipedia, whether the entities are extracted from the tables/lists of the Web or the knowledge base, and whether they are from the bootstrapping method. The following is the list of named entity features used in the classification of answer entity extraction.

Eid is the entity id, which has a corresponding entity from the documents for each topic.

Total has 43494 unique candidate entities extracted from the various methods including entities extracted from NER tools, Wikipedia, tables/lists, and the bootstrapping method.

Did is the document id, which is to identify the document. Total has 633 documents collected for the 20 topics. For each topic, two kinds of queries are generated: narrative as queries and topic entities as queries. For each query, top 16 hits are chosen as candidate germane documents, which also include the real germane documents annotated by our annotators.

Tid is the topic id, which is to identify the topics. Total topics are 20, which are from the TREC 2009 entity retrieval task.

TargetType is the target entity type for this topic. The possible values are organizations, persons, products, and locations.

Wiki indicates whether this entity is extracted from the Wikipedia. For the target entity of organizations, the Wiki value is organizations; for the target entity of persons, the

Wiki value is persons; for the target entity of products, the Wiki value is products; for the others, the Wiki value is others. This value indicates the entity extraction from knowledge base (i.e., Wikipedia) results.

HpLinkentity indicates whether this entity is extracted from the webpages with links.

We assume the terms with links linking to another pages are the entities. Therefore, HpLinkentity is the terms in these links. Because this feature is useless in the classification, it is removed in the final evaluation.

HpLinkentityWiki indicates whether this entity is extracted from Wikipedia pages. If the entity belongs to HpLinkentity, and the entity also appears in the Wikipedia, then its score is 1; otherwise its score is 0.

HpLinkentityWikiRedirect indicates whether this entity is extracted from Wikipedia and redirected from other pages. If the entity belongs to HpLinkentityWiki, and the entity is redirected by the original one in the Wikipedia, then its score is 1; otherwise its score is 0.

StanfordNERfreq is the frequency of the entity extracted from the documents using Stanford Named Entity Identification Tools. Here, we treat the web page as a plain text for extraction. Therefore, the pre-processing of removing HTML tagger is applied to them. This value indicates the method of named entity recognition tools for entity extractions.

UIUCNERfreq is the frequency of the entity extracted from the documents using UIUC Named Entity Identification Tools. Here, we treat the web page as a plain text for extraction. Therefore, the pre-processing of removing HTML tagger is applied to them. This value indicates the method of named entity recognition tools for entity extractions. The reason we used two named entity recognition tool is that UIUC NER includes the thesaurus extraction while Stanford NER is good at sentence structure analysis.

HpListStanford indicates whether this entity is extracted from Stanford NER tool. There are lots of entities listed in the tables or lists without the sentence structure. Therefore, we also extract the terms in the lists, and then apply the Stanford NER to recognize the target entities. If the terms are from the lists and extracted by the Stanford NER, then the score for them are 1; otherwise is 0.

Bootstrapping is the score from the Bootstrapping method, which is investigated in previous section.

ClassLabel indicates whether the entity is the answer entity or not.

6.4.2 Evaluation and Results

The goal of the experiment investigates whether the learning-based method, which integrates multiple extraction methods, can improve answer entity extraction comparing to those individual extraction methods. The test sets are 20 topics in the TREC 2009 data set (Section 4.2.2). The method is evaluated on the germane documents for these 20 topics. The evaluation uses the criteria of precision, recall and f-measure, as described in Section 4.3. The experiment is based on the following groups.

- Baseline I: the named entity recognition (NER) tools for answer entity extraction (Section 6.1.1).
- Baseline II: the knowledge bases for answer entity extraction (Section 6.1.2).
- Baseline III: the Web table/list answer entity extractions (Section 6.2)
- Baseline IV: the bootstrapping method for answer entity extraction (Section 6.3)
- Experiment system: the learning based method for answer entity extraction.

The methodology of this experiment is:

1. In order to extract the UIUCNERfreq and StanfordNERfreq features, we treat each document as a plain text by removing all the HTML tags, and then extract the entities using Stanford NER and UIUC NER tools respectively. The entities extracted from Stanford NER from the germane documents are evaluated as Baseline I.
2. Extract the company-product pairs from the Wikipedia Infobox. All the entities from the Wikipedia are filtered by the Wikipedia categories. The entities extracted from this part are evaluated as Baseline II.
3. All entities extracted from the table/list method are as Baseline III.
4. Entities extracted from the bootstrapping method are as Baseline IV.
5. In order to evaluate the experiment system, all entities extracted the above method are collected and treated as candidate entities. The candidate entities are randomly divided

into 10 folds. Every time, we train a SVM model on 9-folds set, and then test the last folds with the trained model. The final precision, recall, f-measure is the average of these 10 results.

There are total 8318 distinct candidate entities, and 1447 out of them are the answer entities. There are total 3889 candidate entities extracted from the Stanford NER tools, and 1210 out of them are the answer entities. There are total 4779 candidate entities extracted from UIUC NER tools, and 1194 out of them are the answer entities. There are total 1105 candidate entities extracted from the tables/lists, and 720 out of them are the answer entities. There are total 4 candidate entities extracted from the Wikipedia Infobox, and all of them are the answer entities. There are the 229 candidate entities filtering out by the Wikipedia categories, and 65 out of them are the answer entities. There are total 192 candidate entities extracted from the Bootstrapping method, and 8 out of them are the answer entities (for the top 8 ranked entities).

Table 18 shows the performances of four baseline systems and the experiment system. The experiment system using the learning-based method will classify an entity as the answer entity or not. Since most of the candidate entities are non answer entities (6871 out of 8318), we will pay more attention on the positive answer entities (i.e., Class label=1). Comparing the performance of the learning-based entity extraction with the baseline systems, the precision is 0.95, which is better than the four baseline systems (0.1, 0.1, 0.2, and 0.05 respectively). Recall is 0.5, which is better than the four baseline systems (0.4, 0.2, 0.1, and 0.1 respectively). F-measure value is 0.5, which is also better than the four baseline systems (0.1, 0.2, 0.1, 0.1). The improvement in the precision is much higher than in the recall. Therefore, we conclude that the learning-based method can integrate multiple extraction methods for the answer entity extraction.

Table 19 summarizes the features used in the learning-based method for the extraction. The positive value of the weights contribute to the answer entities, while the negative value of the weights contribute to the non-answer entities. The higher absolute value of the weight, the more important the feature is to the classification model. According to these features' weights, we find that the answer entities extracted from the Wikipedia Infobox are the most valuable answers. The knowledge bases for the answer entity type filtering are also useful

Table 18: Results of the learning based method for answer entity extraction

Topic ID	Precision	Recall	F	
Baseline I: NER Tools for Answer Entity Extraction				
Overall	0.103	0.419	0.144	
Product	0.012	0.2959	0.023	
Person	0.248	0.546	0.337	
Organization	0.077	0.411	0.111	
Homepage	0.114787	0.369279	0.155062	
Wikipage	0.083	0.5204	0.1269	
Baseline II: Knowledge Based for Answer Entity Extraction				
Avg. of found entities	0.377078	0.504512	0.368849	
Avg. of 20 topics	0.119685	0.227031	0.165982	
Baseline III: Tables/Lists for Answer Entity Extraction				
Avg of found entities	0.693137	0.338942	0.436744	
Avg of 20 topics	0.173284	0.084736	0.109186	
Baseline IV: Bootstrapping method				
Avg. of extracted topics	1	1	1	
Avg. of 20 topics	0.05	0.05	0.05	
Experiment System: the Learning Based method for Answer Entity Extraction				
SVM	Class label = 0	0.899	0.995	0.945
	Class label = 1	0.95	0.471	0.629
	Weighted Avg.	0.908	0.904	0.89

for the answer entity extraction, especially when the entity type is simple and easy to be detected. The bootstrapping method, which extracts the answer entities according to the sentence structure of subject-verb-object, is also valuable method for the extraction. Other methods, such as answer entities extracted from the tables/lists or using the NER tools for the answer entity extraction contribute trivial efforts on the classification.

Table 19: The features with their weights in the learning-based extraction

Feature	Weight	Feature	Weight
wikitypeperorginfo=Infobox	1.4919	WikiTypeFiltering=0	-1.0004
WikiTypeFilteringRedirect	1.0003	WikiTypeFiltering=PERSON	1.0001
Bootstrapping	1	HpTable	0.0045
hplinkentity=PRODUCT	-0.0008	TargetType=PERSON	-0.0008
hplinkentity=PERSON	0.0006	TargetType=PRODUCT	0.0006
hplinkentity=0	0.0005	hplinkentity=ORGANIZATION	-0.0003
WikiTypeFiltering=ORGANIZATION	0.0003	stanfordNERfreq	0.0003
TargetType=ORGANIZATION	0.0001	UIUCNERfreq	0

6.4.3 Discussion

Answer entity extraction by considering the contexts, physical contexts and logical contexts, treats the extraction as a query-dependent task instead of a query-independent task as most current competition groups in TREC used. For the logical contexts, we consider the tables/lists and sentence syntax. Experimental results show that the learning based answer entity extraction method performs well in the 2009 TREC entity retrieval data set. Treating answer entity extraction as a classification problem can improve the answer entity extraction. This method integrates the several approaches into one model by treating the previous results as the features in the model. As the results show, this method can successfully find the answer entities with high accuracy. However, the learning-based method can find less than half of answer entities, even with multiple extraction methods. As we discussed, the failure of answer entity extraction is caused not only by the complicated structure of tables/lists containing the answer entities, but also by the physical contexts of answer entities, such as PDF files or images. How to improve the recall of the answer entity extraction will be an issue for future work.

6.5 EXPERIMENTS ON TREPM MODEL

Although the TREPM model can decouple the entity retrieval task into two separating components and be evaluated at two individual layers, we would like to know how two layers affect each other. We evaluate the TREPM model on the TREC 2009 entity retrieval set, i.e., evaluating germane document identification and answer entity extraction as a whole. Moreover, in order to compare this model with the state-of-the-art entity retrieval techniques, I participated the TREC 2010 entity retrieval competition, and report the results in this section.

6.5.1 Evaluation on TREC 2009 Task

In order to test the performance of two layers, the evaluation on the whole TREPM model is conducted on the 20 topics of the TREC 2009 entity retrieval task. Three groups are evaluated, one baseline systems and two experiment systems.

- **Baseline system:** The top 16 relevant documents from the topic entity as query and the top 16 relevant documents from the narrative as query are collected and treated as the germane documents, called pseudo germane documents. The table/list answer extraction method based on the pseudo germane documents is used for answer entity extraction. In this experiment, we use the table/list extraction as a baseline extraction method because it runs the best comparing with other extraction methods, NER extraction, table/list extraction, bootstrapping method, knowledge base extraction and knowledge base entity type filtering. We use the pseudo germane documents and the table/list extraction as the baseline system to evaluate the TREPM model.
- **Experiment system I:** The top 16 relevant documents queried from the topic entity as query and the top 16 relevant documents from the narrative as query as pseudo germane documents are collected. The learning-based method based on the pseudo germane documents is used for answer entity extraction. This experiment system fixing the component of germane document finding tries to compare the answer entity extraction component in the TREPM system based on the baseline system and this experiment system.

- Experiment system II: the learning to rank method is used for germane document identification and the learning-based method is used for the answer entity extraction in the TREPM model. Experiment system II will be compared with Experiment system I on the component of germane document identification by fixing the component of answer entity extraction.

The results are as shown in Table 20. In both Experiment system I and Experiment system II, the classification label of 0 is the non-answer-entity, and the classification label of 1 is the answer entity. Since we are aim to extract the answer entities, we only discuss the classification label of 1 here.

Table 20: Results of entity retrieval with TREPM model

Class Label		Precision	Recall	F
Baseline System: Table/List Extraction on Pseudo Germane Documents				
Avg of found entities		0.107893	0.143435	0.123151
Avg of 20 topics		0.086315	0.114748	0.07951
Experiment System I: Learning Based Entity Extraction on Pseudo Germane Documents				
SVM	Class label = 0	0.964	0.992	0.978
	Class label = 1	0.694	0.334	0.451
	Weighted Avg.	0.95	0.957	0.95
Experiment System II: Learning to Rank + Learning Based Extraction				
SVM	Class label = 0	0.964	0.992	0.978
	Class label = 1	0.697	0.337	0.454
	Weighted Avg.	0.95	0.957	0.95

The precision on Baseline System is 0.08, the recall is 0.11, and the F-measure is 0.07, while the precision on Experiment System I is 0.69, the recall is 0.33, and the f-measure is 0.45. Experiment System I is significantly better than Baseline System on precision, recall, and F-measure (two tailed t-test, $p < 0.001$). Because Baseline System and Experiment System I are on the same condition of germane document identification, i.e., pseudo support

documents, we conclude that the component of answer entity extraction in TREPM model has significantly effects on the entity retrieval. There are no significant differences between Experiment System I and Experiment System II. These two experiment systems have the same condition on the component of answer entity extraction but different condition on the component of germane document identification, Therefore, we conclude that germane document identification in the TREPM model has no effects on entity retrieval. According to the above conclusions, we argue that answer entity extraction is more important than the germane document identification in the TREPM model.

6.5.2 Evaluation on TREC 2010 Task

In order to compare the TREPM model with the state-of-the-art techniques in the domain of entity retrieval, I jointed 2010 TREC entity retrieval task, and had submitted the results. Because the TREC task requires entities' URLs/URIs as answers, an additional step is added to the model in order to meet their requirement, i.e., matching the answer entities to their homepages.

According to the experiment results from the APPENDIX A for entity homepage detection, the result of the learning-based method for homepage detection has no significant difference with the result of treating the first relevant document retrieved by search engines as the home page for the answer entities. Therefore, we use the search engine (Google API) to find the first URL as the home page for the answer entity. Then, we match the URLs of the answer entities to the URIs in the ClueWeb09B collection. The results averaged over 70 topics. The result of NDCG@R is 0.2884, and the MAP is 0.164, and rPrec is 0.2258, which is ranked 9 out of total 48 submissions. The best result is 0.369 for NDCG@R, 0.273 for MAP, and 0.308 for rPrec from the BIT group. The eight higher rank runs come from three groups: BIT, Fudan, and Purdue.

The BIT group considers the structures of the Webpages. They employ a logical sitemap constructor, which extracts hierarchical structures in order to enrich the anchor text model for finding more relevant pages. Those hierarchical structures, such as menus or navigational bars or breadcrumbs, indicate the logical relations between pages in the same site and the

concise summary of pages in some sense. Under the assumption that items in similar visual presentations are probably similar in nature and to be classified in a group, they discriminate extracted entities by their locations in DOM tree and give more preference weights to multiple entities in tables and lists. The better understanding for the documents allows them to have their outstanding performs [Yang et al., 2010]. In my future research, in order to improve germane document identification, the structure analysis will be introduced to further detect the germane documents.

The Fudan group proposes a multiple-stage retrieval framework for the task of related entity finding. In germane document identification, search engine is used to improve the retrieval accuracy. In answer entity extraction, they extract entity with NER tools, Wikipedia and text pattern recognition. Then a stop list and other rules are employed for filtering entity. Specifically, deep mining of the authority pages in germane document identification is also conducted by their group. In answer entity detection, many factors including keywords from narrative, page rank, combined results of corpus-based association rules and search engine are considered in their implementation [Wang et al., 2010]. These will be also included into our future research.

The Purdue group generally follows their probability retrieval model proposed last year to estimate the similarity between the queries and the entities. They also investigate the structures of tables and lists to extract related target entities from them. Moreover, they infer the types of target entities from the query and infer the types of candidate entities from their profiles, and then estimate the similarity between target entity types and candidate entity types [Fang et al., 2010]. In the future, we would like to improve our table/list extract algorithms to correctly detect the relevant tables and understand the complicated structure tables.

The entity retrieval task this thesis is slightly different from the entity retrieval task in TREC in that TREC requires the return of homepages of answer entities as results but my task is to return the answer entities as results. Therefore, this study only uses the deal with retrieving answer entities and ignores homepage detection for the answer entities. With the further investigation on the method of the homepage detection for the answer entities, we could expect the better results on the TREC task.

6.5.3 Topic Analysis on TREC entity retrieval

This dissertation discusses about the general entity retrieval task, although we based on entity retrieval task in TREC for evaluation. The current TREC entity retrieval is on the Web pages and asks the general domain questions. Therefore, it still can demonstrate the general entity retrieval task. Furthermore, these methods can be applied in the special domain, such as the medical domain, because the current TREPM model is domain independent. All methods and the model can be transferred or applied into these domains.

With reviewing the topics in TREC 2009, we identify two types of topics with regard to whether the answers are uniquely existing in a document. The first type is asking the general knowledge or information, e.g., “products of MedImmune Inc.” The answers for this type of topics can exist in the multiple germane documents repeatedly. They are either scattering in several germane documents and require people to summarize the answers for the topics complementally, e.g., “carriers that Blackberry makes phones for”, or accumulatively appear in some documents which require people to extract them as a whole, e.g., “products of MedImmune Inc.” The second type of topics is asking the questions whose answers are uniquely existing in one document, e.g., “students of Claire Cardie” or “Donors to the Home Depot Foundation.” This type of topics is sensitive at the germane document identification and answer entity extraction. If the system fails at detecting the germane document for this type of topics or detecting the answer entities, the system will fail at collecting the answer entities. This type of topic is tougher one than previous one.

There are seven out of twenty topics in the TREC 2009 entity retrieval tasks, whose answers are uniquely existing in the Web.

- The topics, such as “Students of Claire Cardie”, only can find the answer hint from the topic entity’s homepage, e.g., “Clair Cardie”. If the web pages are removed or the web pages are composed with PDF files or photos, the component of answer entity extraction will be hard to detect the answer entities. Therefore, this case is critical for the germane document identification.
- The topic of “Chefs with a show on the Food Network” has the unique answers in their website about the TV show schedule. However, the representation of the table structure

uses embed HTML lists, so that the current extraction method can not fully extract all answer entities.

- The topic of “Winners of the ACM Athena award” has the answers in the ACM webpage. The difficulty to detect the answer entities for this topics is that the winners’ names mixture with other awards. Therefore, how to detect the related tables within a germane document will be a challenge task.
- The topic of “Authors awarded an Anthony Award at Bouchercon in 2007” is tough because of the year limitation of the query. The topic needs to find the answers in the exact year of 2007. Therefore, the answer entity extraction component should differentiate the answers from others with regards to years.
- The topic of “Sponsors of the Mancuso quilt festivals” has the unique answer sets in their website. Especially, in order to protect their sponsors to be maliciously crawled, these sponsors are embedded in the Web page using the images of logos. Therefore, although we can easily detect the germane document, it is still hard to extract the answer entities. Similar case is the topic of “Donors to the Home Depot Foundation.”

There are thirteen out of twenty topics in the TREC 2009 entity retrieval tasks asking the questions whose answers exist in the multiple documents. For example, for the topic of “carriers that Blackberry makes phones for,” the answers are scattering in the multiple documents, and the system is required to crawl them and summarize them as the answer set.

- The topic of “professional sports teams in Philadelphia” has the answers in multiple documents. Some of them cover all answer sets, while some others are not. Some answers are in the sentences, while others are in the tables/lists. Similar cases include “products of MedImmune Inc”, “Scotch whisky distilleries on the island of Islay”, “Campuses of Indiana University”, “Members of the band Jefferson Airplane”, “CDs released by the King’s Singers”, “Airlines that currently use Boeing 747 planes”, “Members of The Beaux Arts Trio”, and “Airlines that Air Canada has code share fights with”.
- The topic of “organizations that award Nobel prizes” can be easily be confused with the topic of “organizations awarded Nobel prizes”. Therefore, we use the topic entity as

queries “Nobel prizes” to find the germane documents for answer entity extraction.

- The topic of “Journals published by the AVMA” includes the abbreviation of “AVMA” for “American Veterinary Medical Association (AVMA)”. Similar case is the topic of “Universities that are members of the SEC conference for football”.
- The topic of “Companies that John Hennessy serves on the board of” has the answers scattering multiple documents. However, it is also really not obvious webpages indicating the information. It is also a tough topic.

6.6 SUMMARY

This chapter examined answer entity extraction, whose target is to identify answer entities from germane documents for the entity retrieval task in an effective way. We considered several ways of entity extraction: named entity recognition tools, knowledge base (Wikipedia) extraction and entity filtering, table/list extraction, bootstrapping methods, and classification methods.

Named entity recognition tools (NER) for answer entity extraction can only work on grammatical sentences. It treats the documents as plain texts, so the corpus containing noise web pages should be preprocessed by removing the HTML tags. With the pre-processing, many non-grammatical sentences are generated in the corpus, which causes some errors in extraction. For example, many entities are listed as items in the Web page. The simple parsing is hard to extract answer entities according to the queries from the germane document. This is the reason why the recall for the NER entity extraction is high (about 0.4 on the extraction from germane documents) but the precision (about 0.1 on the extraction from germane documents) and the F-measure (about 0.1 on the extraction from germane documents) is low. Moreover, this method also depends on whether NER can identify the type. If the NER tool could not identify the types, it will fail to extract them. For example, the extraction results on the entity type of product are worse than the ones on the entity types of person and organization.

Two approaches using knowledge base to facilitate entity extraction are investigated to

improve the precision and recall of answer entity extraction. One is to mine the entity answers from a knowledge base (e.g., Wikipedia Infobox). The other one is using the knowledge base to filter the non-relevant entities out. The results of knowledge base answer entity extraction show that the approach can extract high accuracy entities but only for a small set of those topics. The method of knowledge base filtering can significantly improve the accuracy of answer entity extraction. But both methods are limited by the knowledge base and the representation in the knowledge base.

Tables/lists are considered as the symbolic contexts for the entity extraction. As the analyses on the entity contexts, we find that most entities are in tables or lists. Therefore, an algorithm to extract the entities from the tables/lists in the Web is investigated and implemented for answer entity extraction. The results show that this approach is more accurate than the NER system, but also it can find 30% entities. This is because part of answers are in the different media, such as images or PDF files. The complicate representation of the tables/lists in the web page is another reason for extraction failure.

A semi-supervised learning method, bootstrapping, is considered as the syntactic context for answer entity extraction. The experiment shows this approach can achieve high recall results for some topics. But it is also highly dependent on the entity seeds and patterns. In this experiment, the method could only extract the answers for one topic (out of 20). In the future work, I will investigate the impact of more seeds and better patterns for the extraction.

In order to complement the extraction disadvantages from the above methods, we treated the entity extraction as a binary classification problem and the extraction results from the above methods as features. The experiment compares this method with the other answer entity extraction methods is conducted. The results indicate that this method is significantly better than all the individual extraction methods by themselves. However, because the low recall of the above extraction method, the learning-based method could only find half of the answer entities. The reason for the low recall is that the current system only treats the noun phrases as the candidate answer entities. Therefore, it will miss some answer entities with special characters, such as FluMist®. In the future, more methods should be introduced to improve the recall of the answer entity extraction.

7.0 CONCLUSION AND FUTURE WORK

This dissertation has studied the problem of entity retrieval in the unstructured data environment. Guided by the user’s information need about relevant information and relevancy verification in the entity level, this thesis sets out to develop a Two-layer Retrieval and Extraction Probability Model (TREPМ) capable of integrating document retrieval (germane document identification) and entity extraction (answer entity extraction) in order to efficiently and effectively detect the answer entities from the corpus.

7.1 TREPМ MODEL REPRESENTATION

Germane document identification efficiently finds germane documents with the assumption of bag-of-words; while answer entity extraction effectively extracts the answer entities from the germane documents by considering the semantic relations between words. The TREPМ model delineates whole entity retrieval problem. Chapter 3 theoretically demonstrates that entity retrieval can be interpreted as the TREPМ model, which decomposes the problem into document retrieval and entity extraction, using a probability model. That is, $p(e|q, t) = \sum_d p(d|q, t)p(e|d, q, t)$. The TREPМ model provides a method to retrieve information in a finer granularity but with low system workload.

This decomposition helps to break the black box of entity retrieval into document retrieval and entity extraction. It not only allows the evaluations on each individual layer, which further improves the overall system performance, but also helps to bring the state-of-the-art techniques in the document retrieval and entity extraction into the entity retrieval task.

Although the entity retrieval task in TREC and INEX requires entities’ URLs/URIs as

answers, this study focuses on entity retrieval task itself and treats the entities as answers instead of entities' URLs/URIs as answers. This model summaries the general problems of entity retrieval, which can be applied to TREC can INEX task also.

7.2 GERMANE DOCUMENT IDENTIFICATION

Germane document identification (Chapter 5) discusses how to effectively locate the highly relevant germane documents, which contain as many answer entities as possible.

Some methods are investigated for germane document identification. First, we study how to generate the proper queries in order to collect germane documents and how to set up the threshold to choose germane documents. Both the narratives and topic entities could be the source of queries for searches. The experiment indicates that in most cases the narratives are a better source for the queries. However, when the narratives are sensitive in representing the relation between the topic entity and the target entity (e.g., “organizations awarded Nobel Prizes” vs “organizations that award Nobel Prizes”), the topic entity is better to be the queries.

Second, the entity type language model is investigated to evaluate whether the similarity between entity types and document categories can improve germane document identification. The documents with associated categories widely exist in the Web environments. The entries in the knowledge base, assigned with some categorizes or the posts in the social network with their tags, can be viewed as one of this type of documents. The experiment indicates that entity types or document categories are helpful for germane document identifications. The entity type language model can significantly improve the entity search result in the documents with their categories.

Last, we investigate the “learning to rank” method for germane document identification. The learning to rank approach treats germane document identification as a binary classification problem. Twenty-eight features are generated from queries, the hits, and the linguistic features used for the classification. The evaluation indicates that the learning to rank method can achieve high accuracy on germane document identification. With the anal-

yses on the annotations of germane documents, there is a germane document including all answers to the topic for most topics. But there are still some topics whose answers scatter in several documents. Wikipedia is an important source for the answer sets because we find the germane documents from the Wikipedia for about half of the topics.

Current evaluation on germane document is based on the comparison with the ground truth germane document sets. Therefore, precision and recall is also based on the germane documents. However, in fact, it is not so accurate to estimate the degree of these germane documents covering the answer entities. With a germane document with all answer entity set for a topic, it can still be possible to be extracted all answers although the recall of this germane document may be very low. Therefore, in the future, we should consider using the number of the answer entities as the weight for evaluating the germane documents.

7.3 ANSWER ENTITY EXTRACTION

Answer entity extraction (Chapter 6) discusses different approaches for answer entity detection in the entity extraction task. Entities in the germane documents can be in various contexts, which can be interpreted in multiple ways. From the physical context view, it includes html pages, plain texts, pdf files or image files. In this study, we only focus on plain texts and html pages. From the logical context view, the answer entities exist in tables/lists or the sentences. Therefore, in this thesis, I focus on answer entity extraction from these two resources.

Most of the current work on entity retrieval rely on NER tools to extract the entities with target types. This answer entity extraction method does not consider the contexts and treats the extraction as a query-independent extraction. In our study, we find that the precision of this method is low. Because the corpus is the noisy web page and the NER is trained by the grammatical corpus, NER could not correctly identify the entities for this corpora and the results from this method are not promising.

The second method uses the knowledge base (Wikipedia) for entity extraction, which hopes to extract the answer entities from the ungrammatical documents with the aim of

knowledge base. The algorithm for Wikipedia Infobox extraction is proposed and the Wikipedia entry category information for entity type filtering is discussed. Although the Wikipedia Infobox extraction can achieve high accuracy result, the recall is rather low. The entity type filtering using Wikipedia information is limited by the knowledge in the Wikipedia. With the analyses of the contexts of answer entity in the germane documents, we find that most entities are from tables and lists, which need some efficient methods for detections.

The answer entities are scattering across several HTML pages with symbolic contexts. Therefore, answer entity extraction with wrappers is introduced to extract the entities from tables or lists. This wrapper method also only works for some topics, but fails for the others. One of the reasons for the failure of the extraction is that answers are put into the pictures which cannot use text mining way to extract them. Another is the complicated table/list structure and the representation way, which can not be well extracted by the current system.

Semi-supervised learning method, bootstrapping, is conducted for entity extractions. The idea of bootstrapping is that, by identifying the reliable patterns from the good seeds, the model can extract more result entities with these patterns. Although the precision of bootstrapping is high, the recall is still low because this method is limited by the quality seeds and good patterns. For the topics whose answers uniquely exist in the Web, it will be difficult to find the good quality seeds and pattens.

With the above extraction methods for answer entity extraction, the last method treats answer entity extraction as a learning problem, which is to learning the above methods as features for entity extraction. The results show that the learning based method significantly better than all the above methods individually.

7.4 THE FUTURE OF ENTITY RETRIEVAL AND ITS APPLICATION

This thesis is definitely not the complete work of entity retrieval. There are many research questions and implementation questions which are needed to be further investigated. Moreover, entity retrieval task is not only limited in the general domain web retrieval but also

applied into other domains, such as medical text mining.

7.4.1 Future Work

The TREPM model for entity retrieval achieves the best results or retrieves the answer entities with heuristic methods. If the number of germane documents m' is large (i.e., $m' \approx m$, where m is the number of documents in corpus), this method will fail at detecting the germane documents and further fail at extracting the answer entities. One of future work will investigate the method to find the answer entities for these type of entity retrieval.

For some topics, the germane document is unique in the corpus, e.g., the germane document for the topic of sponsors of the Mancuso quilt festival. In this case, the study will focus on the accurately detect the germane document. Although the learning to rank method which can achieve high precision is investigated in this dissertation, the recall of the current method is around 0.5, which means it misses half of the germane documents. Therefore, the more features need to be further studied to improve the recall of germane document identification.

Although knowledge base extraction, table/list extraction, bootstrapping extraction and the learning-based method for answer entity extraction are evaluated in entity retrieval, several researches should be further done on the extraction.

First, in answer entity extraction without context, we find that the types of entities which can be detected by the named entity recognition tool and the accuracy of the detection are critical for this task. Therefore, in future work, the tool accurately detecting the entities with the target types, such as products, will be studied.

Second, we can build a tool or method to support the relation mapping, which describes the relation between topic entities and answer entities. In the relation context extraction, the various representation of relations in the knowledge base attribute causes the difficulties on answer entity extraction. For example, the relation of “use” in the topic of “airline that currently use Boeing 747 planes” can be represented as “users” in the Wikipedia Infobox under the entry of “Boeing 747.” How to improve the relation mapping between different representations will be our future work.

Third, although this dissertation investigates the table/list extraction for the answer entity extraction, the implementation should be further improved in several points. The current system treats the germane documents as the units for extracting the answer entities for the table/list extraction. However, in fact, all tables in the same germane document are not necessary to be containers for the answer entities. Therefore, the future implementation should find the relevant tables in the germane document for answer entity extraction. Another point is that the current system can not deal with the complicated table/list structures, such as the embedded or hierarchical structure. Therefore, it will also be our future work.

Fourth, the bootstrapping method can extract the high accurate answer entities. However, this method is limited by the sample seeds. Currently it only works well for those topics with good sample seeds. On the one hand, for those topics with the answers existing in the multiple documents, in the further work, more sources need to be discovered for the seed detections. On the other hand, for those topics with the answer topics uniquely existing in the Web, some methods should be investigated for syntax analysis in order to extract the answers.

Last method treats answer entity extraction as a learning problem. Although it can significantly improve answer entity extraction than all above methods individually, the recall of this method is still not very high. The failure of Some entities comes from the different media such as pdf or image, which are beyond the discussion of this dissertation. They will be our future work.

The current entity retrieval task is designed as finding the URLs/URIs for the entities either in TREC or INEX task. Therefore, the evaluation system as well as the ground truth is annotated based on the answer entities' URLs/URIs. I would like to argue that it is far from enough. On the one hand, the URLs/URIs of answer entities are different from the answer entities themselves or sometime, it is hard to tell which should be the proper URL/URI for an answer entity. For example, for one answer of the topic of "products of MedImmune Inc.", Ethyol, both the webpage of <http://www.ethyol.com/> and the webpage of <http://www.medimmune.com/products/ethyol/index.asp> are annotated. If you miss any one, the performance will be dropped. However, the answer entity itself is correct. On the other hand, the evaluation is on the final step of the whole "black box." It is not enough.

Moreover, although this dissertation tries to decompose it as two steps and evaluate them in details, the current system evaluates germane document identification at the germane document level. However, we would like to argue that it should also be evaluated at the answer entity level because the final task for the system is to detect the answer entities. Even though the low precision and recall of germane document identification, it still can achieve high performance of answer entity extraction. In the future, the evaluation should be also based on the answer entities for germane document identification.

The current evaluation is based on the general domain web entity retrieval task from TREC. Moreover, we analysis the topics provided by TREC with the framework of TREPM. We find that answer entity retrieval has more important effects on the germane document identification. However, the further detail relations between these two layers will be our future research.

On February 2011, IBM's Watson computer facing off against two former Jeopardy! champion in a two-game match played over three shows. Dr. Watson locked the first game and won 1 million. This is a great successfulness of entity retrieval, which will also be our future research direction.

7.4.2 Applications on Medical Entity Retrieval

This dissertation has elaborated a TREPM model for the entity retrieval task. It is not only applied in the competition tasks, such as TREC or INEX, but also promising in some domain problems, such as medical ontology learning.

The task of medical ontology learning is to mine the knowledge from the community experts or the publications or related sources to form well-represented ontologies. Current approaches for medical ontology building are relying on the experts to point out the medical diseases with their findings. Although this method is accurate, it is tedious with heavy human labor cost. The medical ontology can be learn from multiple resources: structured data (such as medical database), semi-structured data (such as Wikipedia), and unstructured data (such as the Web or the medical reports).

According to the six layer definition of ontology, the ontology learning also includes the

six layers, which are the terms learning, synonyms learning, concepts learning, taxonomy learning, relation learning, and rule learning. The first four layers can be treated as concepts and their structure learning. The fifth layer is the relation learning. The sixth layer is the inference and reasoning.

The medical concepts and structure learning is to mine the medical publications or authorized medical resources for diseases' related medical findings, for example, shigellosis findings' mining. The authorized resources, such as the shigellosis page on the Wikipedia page (<http://en.wikipedia.org/wiki/Shigellosis>) clearly states that "Symptoms may range from mild abdominal discomfort to full-blown dysentery characterized by cramps, diarrhea, fever, vomiting, blood, pus, or mucus in stools or tenesmus. Onset time is 12 to 50 hours." The mining system automatic detects the symptoms for this disease as abdominal discomfort, dysentery, cramps, etc. With the multiple publications about the shigellosis as well as its findings, the algorithm would estimate the probability of each mined medical finding to be the disease's related finding. Furthermore, I would like to mine the probability of the medical findings for certain disease.

The detected medical diseases and their associated findings would need to detect the relation between them. Many medical findings are related to different types, such as symptoms or medicines, and some medical findings are hierarchical related. Therefore, we are eager to detect the relation between the diseases and their medical findings. The supervised or semi-supervised methods are used for the extraction and detection. The relation is important for the medical ontology construction because it helps to disambiguate the relations involving two entities. For example, the position of the tumor in the body is an important factor for the diagnosis and sensitive for the treatment.

The application of the TREPM model is not only in the medical domain but also in other domains because the TREPM model aims at the general domain. For example, expert findings in the social network can use the TREPM model for quickly narrowing down the germane documents and then detecting the experts.

With the development of retrieval system, users are no longer satisfied with finding the relevant documents. They would like to find the relevant "nuggets" smaller than documents, such as entities. Therefore, the research on entity retrieval problem will be more and more

critical in the future. The work in this dissertation is definitely not my ending of the research on this topic, but will be my starting point.

8.0 ACKNOWLEDGMENTS

I would like to thank the members of my dissertation committee, Dr. Micheal Spring, Dr. Paul Munro, Dr. Jung Sun Oh, Dr. Fu-Chiang Tsui, and Dr. Daqing He for their efforts and time in guiding me to fulfill the requirement of the degree.

In particular, I would like to acknowledge Dr. He who gives me chances to work in the domain of information retrieval.

I would like to acknowledge Dr. Peter Brusilovsky, Dr. Wendy Chapman, and Dr. Fu-Chiang Tsui who open the door of Biomedical Informatics for me.

I would like to acknowledge Dr. Ming Mao and Dr. Yefei Peng who help me to apply my knowledge to the daily life.

I would like to acknowledge Vicky Chen and Jon Walker who help proofreading my paper. I learn a lot from you.

I would like to acknowledge my friends in various labs, Dan Wu, Jongdo Park, Sung-Min Kim, Zhen Yue, Yiling Lin, Sharon, Jiepu Jiang, Shuguang Han, Jialan Que, Wei Wei, Ming Li, Ren Ming, Hua Li, Cui Jie

I would like to acknowledge my parents, who have most generous love and allow me to do whatever I like.

Thank you for all people who always support and encourage me.

BIBLIOGRAPHY

- [Alias-i, 2008] Alias-i (2008). Lingpipe 4.0.1. <http://alias-i.com/lingpipe>.
- [Amer-Yahia et al., 2007] Amer-Yahia, S., Botev, C., Buxton, S., Case, P., Doerre, J., Holstege, M., Melton, J., Rys, M., and Shanmugasundaram, J. (2007). Xquery 1.0 and xpath 2.0 full-text 1.0 working draft.
- [Asahara and Matsumoto, 2003] Asahara, M. and Matsumoto, Y. (2003). Japanese named entity extraction with redundant morphological analysis. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 8–15, Stroudsburg, PA, USA. Association for Computational Linguistics.
- [Auer et al., 2007] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). Dbpedia: a nucleus for a web of open data. In *Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference, ISWC'07/ASWC'07*, pages 722–735, Berlin, Heidelberg. Springer-Verlag.
- [Balog et al., 2010a] Balog, K., Bron, M., De Rijke, M., and Weerkamp, W. (2010a). Combining term-based and category-based representations for entity search. In *Proceedings of the Focused retrieval and evaluation, and 8th international conference on Initiative for the evaluation of XML retrieval, INEX'09*, pages 265–272, Berlin, Heidelberg. Springer-Verlag.
- [Balog and de Rijke, 2006] Balog, K. and de Rijke, M. (2006). Finding Experts and their Details in E-mail Corpora. In *15th International World Wide Web Conference (WWW2006)*.
- [Balog et al., 2009] Balog, K., de Vries, A. P., Serdyukov, P., Thomas, P., and Westerveld, T. (2009). Overview of the trec 2009 entity track. In *TREC 2009 Working Notes*. NIST.
- [Balog et al., 2010b] Balog, K., Serdyukov, P., and de Vries, A. P. (2010b). Overview of the trec 2010 entity track. In *TREC 2010 Working Notes*. NIST.
- [Beynon-Davies, 2004] Beynon-Davies, P. (2004). *Database System*. Basingtoke, UK.
- [Bikel et al., 1997] Bikel, D. M., Miller, S., Schwartz, R., and Weischedel, R. (1997). Nymble: a high-performance learning name-finder. In *In Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 194–201.

- [Blanchard et al., 2006] Blanchard, E., Kuntz, P., Harzallah, M., and Briand, H. (2006). A tree-based similarity for evaluating concept proximities in an ontology. In Batagelj, V., Bock, H.-H., Ferligoj, A., and Iberna, A., editors, *Data Science and Classification*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 3–11. Springer Berlin Heidelberg. 10.1007/3-540-34416-0_1.
- [Blum and Mitchell, 1998] Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, COLT' 98, pages 92–100, New York, NY, USA. ACM.
- [Bollacker et al., 2008] Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, SIGMOD '08, pages 1247–1250, New York, NY, USA. ACM.
- [Bollegala et al., 2007] Bollegala, D., Matsuo, Y., and Ishizuka, M. (2007). Measuring semantic similarity between words using web search engines. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 757–766, New York, NY, USA. ACM.
- [Borthwick et al., 1998] Borthwick, A., Sterling, J., Agichtein, E., and Grishman, R. (1998). Exploiting diverse knowledge sources via maximum entropy in named entity recognition. In *IN PROCEEDINGS OF THE SIXTH WORKSHOP ON VERY LARGE CORPORA*, pages 152–160.
- [Brin, 1999] Brin, S. (1999). Extracting patterns and relations from the world wide web. In *Selected papers from the International Workshop on The World Wide Web and Databases*, pages 172–183, London, UK. Springer-Verlag.
- [Brin and Page, 1998] Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.*, 30:107–117.
- [Bron et al., 2010] Bron, M., He, J., Hofmann, K., Meij, E., de Rijke, M., Tsagkias, M., and Weerkamp, W. (2010). The university of amsterdam at trec 2010 session, entity, and relevance feedback. In *Proceedings of the 19th Text Retrieval Conference (TREC 2010)*.
- [Collins and Singer, 1999] Collins, M. and Singer, Y. (1999). Unsupervised models for named entity classification. In *In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 100–110.
- [Cooper et al., 1992] Cooper, W. S., Gey, F. C., and Dabney, D. P. (1992). Probabilistic retrieval based on staged logistic regression. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '92, pages 198–210, New York, NY, USA. ACM.

- [Craswell et al., 2009] Craswell, N., Demartini, G., Gaugaz, J., and Iofciu, T. (2009). *L3S at INEX 2008: Retrieving Entities Using Structured Information*, pages 253–263. Springer-Verlag, Berlin, Heidelberg.
- [Cunningham et al., 2002] Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V. (2002). GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Annual Meeting of the ACL*.
- [de Vries et al., 2007] de Vries, A. P., Vercoustre, A.-M., Thom, J. A., Craswell, N., and Lalmas, M. (2007). Overview of the inex 2007 entity ranking track. In *INEX*, pages 245–251.
- [Drucker et al., 2002] Drucker, H., Shahraray, B., and Gibbon, D. C. (2002). Support vector machines: relevance feedback and information retrieval. *Inf. Process. Manage.*, 38:305–323.
- [Etzioni et al., 2005] Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S., and Yates, A. (2005). Unsupervised named-entity extraction from the web: an experimental study. *Artif. Intell.*, 165:91–134.
- [Fang et al., 2010] Fang, Y., Si, L., Somasundara, N., Yu, Z., and Xian, Y. (2010). Purdue at TREC 2010 Entity Track. In *Proceedings of the Niteenth Text REtrieval Conference (TREC 2010)*.
- [Fang et al., 2009] Fang, Y., Si, L., Yu, Z., and Xu, Y. (2009). Entity Retrieval with Hierarchical Relevance Model, Exploiting the Structure of Tables and Learning Homepage Classifiers. In *Proceedings of the Eighteenth Text REtrieval Conference (TREC 2009)*.
- [Finkel et al., 2005] Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ACL '05*, pages 363–370, Stroudsburg, PA, USA. Association for Computational Linguistics.
- [Gövert and Kazai, 2002] Gövert, N. and Kazai, G. (2002). Overview of the initiative for the evaluation of xml retrieval (inex) 2002. In *INEX Workshop*, pages 1–17.
- [Grishman and Sundheim, 1996] Grishman, R. and Sundheim, B. (1996). Message understanding conference-6: a brief history. In *Proceedings of the 16th conference on Computational linguistics - Volume 1, COLING '96*, pages 466–471, Stroudsburg, PA, USA. Association for Computational Linguistics.
- [Guha et al., 2003] Guha, R., McCool, R., and Miller, E. (2003). Semantic search. In *Proceedings of the 12th international conference on World Wide Web, WWW '03*, pages 700–709, New York, NY, USA. ACM.
- [Harkema et al., 2009] Harkema, H., Dowling, J. N., Thornblade, T., and Chapman, W. W. (2009). Context: An algorithm for determining negation, experiencer, and temporal status

from clinical reports. *Journal of Biomedical Informatics*, 42(5):839 – 851. Biomedical Natural Language Processing.

- [Hearst, 1992] Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics - Volume 2*, COLING '92, pages 539–545, Stroudsburg, PA, USA. Association for Computational Linguistics.
- [Hold et al., 2010] Hold, A., Leban, M., Emde, B., Thiele, C., Naumann, F., Barczynski, W., and Brauer, F. (2010). ECIR - a lightweight approach for entity-centric information retrieval. In *Proceedings of the Nineteenth Text REtrieval Conference (TREC 2010)*.
- [Jiang et al., 2009] Jiang, J., Lu, W., Rong, X., and Gao, Y. (2009). Adapting language modeling methods for expert search to rank wikipedia entities. In Geva, S., Kamps, J., and Trotman, A., editors, *Advances in Focused Retrieval*, volume 5631 of *Lecture Notes in Computer Science*, pages 264–272. Springer Berlin / Heidelberg. 10.1007/978-3-642-03761-0_27.
- [Jones et al., 2006] Jones, R., Rey, B., Madani, O., and Greiner, W. (2006). Generating query substitutions. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 387–396, New York, NY, USA. ACM.
- [Kaptein and Kamps, 2009] Kaptein, R. and Kamps, J. (2009). *Finding Entities in Wikipedia Using Links and Categories*, pages 273–279. Springer-Verlag, Berlin, Heidelberg.
- [Klein and Manning, 2003] Klein, D. and Manning, C. D. (2003). Fast exact inference with a factored model for natural language parsing. In *In Advances in Neural Information Processing Systems 15 (NIPS)*, pages 3–10. MIT Press.
- [Kleinberg, 1999] Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *J. ACM*, 46:604–632.
- [Koolen et al., 2010] Koolen, M., Kaptein, R., and Kamps, J. (2010). Focused search in books and wikipedia: categories, links and relevance feedback. In *Proceedings of the Focused retrieval and evaluation, and 8th international conference on Initiative for the evaluation of XML retrieval*, INEX'09, pages 273–291, Berlin, Heidelberg. Springer-Verlag.
- [Krishnan and Manning, 2006] Krishnan, V. and Manning, C. D. (2006). An effective two-stage model for exploiting non-local dependencies in named entity recognition. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 1121–1128, Stroudsburg, PA, USA. Association for Computational Linguistics.
- [Leacock and Chodorow, 1998] Leacock, C. and Chodorow, M. (1998). *Combining local context and WordNet similarity for word sense identification*, pages 305–332. In C. Fellbaum (Ed.), MIT Press.

- [Li and He, 2010] Li, Q. and He, D. (2010). Searching for entities: When retrieval meets extractions. In *Proceedings of the 19th Text Retrieval Conference (TREC 2010)*.
- [Li and He, 2011a] Li, Q. and He, D. (2011a). Facilitating exploratory search on image metadata with relations. In *ASIST 2011*.
- [Li and He, 2011b] Li, Q. and He, D. (2011b). Finding support documents with a learning to rank approach. In *SIGIR 2011 Workshop on Entity-Oriented Search (EOS 2011)*.
- [Li et al., 2009] Li, Q., He, D., and Mao, M. (2009). A study of relation annotation in business environments using web mining. In *Proceedings of the 2009 IEEE International Conference on Semantic Computing, ICSC '09*, pages 203–208, Washington, DC, USA. IEEE Computer Society.
- [Liu, 2006] Liu, B. (2006). *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data (Data-Centric Systems and Applications)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- [Liu, 2009] Liu, T.-Y. (2009). Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, 3:225–331.
- [Liu et al., 2007] Liu, Y., Bai, K., Mitra, P., and Giles, C. L. (2007). Tableseer: automatic table metadata extraction and searching in digital libraries. In *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries, JCDL '07*, pages 91–100, New York, NY, USA. ACM.
- [Liu et al., 2006] Liu, Y., Mitra, P., Giles, C. L., and Bai, K. (2006). Automatic extraction of table metadata from digital documents. In *In JCDL*, pages 339–340.
- [Luke and Rager, 1996] Luke, S. and Rager, D. (1996). Ontology-based knowledge discovery on the world-wide web. In *Working Notes of the Workshop on Internet-Based Information Systems at the 13th National Conference on Artificial Intelligence (AAAI96)*, pages 96–102. AAAI Press.
- [MacKay and Peto, 1994] MacKay, D. J. and Peto, L. C. B. (1994). A hierarchical dirichlet language model. *Natural Language Engineering*, 1:1–19.
- [Manning et al., 2008] Manning, C. D., Raghavan, P., and Schtze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.
- [marie Vercoustre et al.,] marie Vercoustre, A., Pehcevski, J., and Naumovski, V. Topic difficulty prediction in entity ranking. In *In Geva et al*, pages 280–291.
- [McCallum and Li, 2003] McCallum, A. and Li, W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4, CONLL '03*, pages 188–191, Stroudsburg, PA, USA. Association for Computational Linguistics.

- [McCreadie et al., 2009] McCreadie, R., Macdonald, C., Ounis, I., Peng, J., and Santos, R. L. T. (2009). University of Glasgow at TREC 2009: Experiments with Terrier. In *the Eighteenth Text REtrieval Conference (TREC 2009)*.
- [Nardi and Brachman, 2003] Nardi, D. and Brachman, R. J. (2003). The description logic handbook. chapter An introduction to description logics, pages 1–40. Cambridge University Press, New York, NY, USA.
- [Navarro and Baeza-Yates, 1995] Navarro, G. and Baeza-Yates, R. (1995). A language for queries on structure and contents of textual databases. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '95, pages 93–101, New York, NY, USA. ACM.
- [Ng, 1999] Ng, H. T. (1999). Learning to recognize tables in free text.
- [OpenCalaise, 2010] OpenCalaise (2010). Opencalaise. <http://www.opencalaise.com>.
- [Pantel and Pennacchiotti, 2008] Pantel, P. and Pennacchiotti, M. (2008). Automatically harvesting and ontologizing semantic relations. In *Proceeding of the 2008 conference on Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, pages 171–195, Amsterdam, The Netherlands, The Netherlands. IOS Press.
- [Pasca et al., 2006] Pasca, M., Lin, D., Bigham, J., Lifchits, A., and Jain, A. (2006). Organizing and searching the world wide web of facts - step one: the one-million fact extraction challenge. In *proceedings of the 21st national conference on Artificial intelligence - Volume 2*, pages 1400–1405. AAAI Press.
- [Pinto et al., 2003] Pinto, D., McCallum, A., Wei, X., and Croft, W. B. (2003). Table extraction using conditional random fields. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, pages 235–242, New York, NY, USA. ACM.
- [Ponte and Croft, 1998] Ponte, J. M. and Croft, W. B. (1998). A language modeling approach to information retrieval. pages 275–281.
- [Prager et al., 2000] Prager, J., Brown, E., Coden, A., and Radev, D. (2000). Question-answering by predictive annotation. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '00, pages 184–191, New York, NY, USA. ACM.
- [Ratinov and Roth, 2009] Ratinov, L. and Roth, D. (2009). Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, CoNLL '09, pages 147–155, Stroudsburg, PA, USA. Association for Computational Linguistics.

- [Rau, 1991] Rau, L. F. (1991). Extracting company names from text. In *Artificial Intelligence Applications, 1991. Proceedings., Seventh IEEE Conference on*, volume i, pages 29–32.
- [Ravichandran and Hovy, 2002] Ravichandran, D. and Hovy, E. (2002). Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 41–47, Stroudsburg, PA, USA. Association for Computational Linguistics.
- [Robertson et al., 1996] Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M., and Gatford, M. (1996). Okapi at trec-3. pages 109–126.
- [Rode et al., 2009] Rode, H., Hiemstra, D., Vries, A., and Serdyukov, P. (2009). *Efficient XML and Entity Retrieval with PF/Tijah: CWI and University of Twente at INEX'08*, pages 207–217. Springer-Verlag, Berlin, Heidelberg.
- [Salton et al., 1975] Salton, G., Wong, A., and Yang, C. S. (1975). A vector space model for automatic indexing. *Commun. ACM*, 18:613–620.
- [SAP, 2010] SAP (2010). Inxight. <http://www.sap.com>.
- [Sekine, 1998] Sekine, S. (1998). Nyu: Description of the japanese ne system used for met-2. In *Proc. of the Seventh Message Understanding Conference (MUC-7)*.
- [Serdyukov and de Vries, 2009] Serdyukov, P. and de Vries, A. (2009). Delft University at the TREC 2009 Entity Track: Ranking Wikipedia Entities. In *Proceedings of the Eighteenth Text REtrieval Conference (TREC 2009)*.
- [Strohman et al., 2005] Strohman, T., Metzler, D., Turtle, H., and Croft, W. B. (2005). Indri: a language-model based search engine for complex queries. Technical report, in Proceedings of the International Conference on Intelligent Analysis.
- [Taylor et al., 2006] Taylor, M., Zaragoza, H., Craswell, N., Robertson, S., and Burges, C. (2006). Optimisation methods for ranking functions with multiple parameters. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, CIKM '06, pages 585–593, New York, NY, USA. ACM.
- [Tjong Kim Sang, 2002] Tjong Kim Sang, E. F. (2002). Introduction to the conll-2002 shared task: language-independent named entity recognition. In *proceedings of the 6th conference on Natural language learning - Volume 20*, COLING-02, pages 1–4, Stroudsburg, PA, USA. Association for Computational Linguistics.
- [Trotman, 2004] Trotman, A. (2004). Narrowed extended xpath i (nexi). In *In Proceedings of the INEX 2004 Workshop*, pages 16–40. Springer-Verlag GmbH.
- [Turney, 2001] Turney, P. D. (2001). Mining the web for synonyms: Pmi-ir versus lsa on toefl. In *Proceedings of the 12th European Conference on Machine Learning*, EMCL '01, pages 491–502, London, UK. Springer-Verlag.

- [Vechtomova, 2010] Vechtomova, O. (2010). Related entity finding: University of waterloo at trec 2010 entity track. In *Proceedings of the 19th Text Retrieval Conference (TREC 2010)*.
- [Vercoustre et al., 2009] Vercoustre, A.-M., Pehcevski, J., and Naumovski, V. (2009). Topic Difficulty Prediction in Entity Ranking. In *Proceedings of the Eighteenth Text REtrieval Conference (TREC 2009)*.
- [Vercoustre et al., 2008] Vercoustre, A.-M., Pehcevski, J., and Thom, J. A. (2008). Focused access to xml documents. chapter Using Wikipedia Categories and Links in Entity Ranking, pages 321–335. Springer-Verlag, Berlin, Heidelberg.
- [Vydiswaran et al., 2009] Vydiswaran, V., Ganesan, K., Lv, Y., He, J., and Zhai, C. (2009). Finding Related Entities by Retrieving Relations: UIUC at TREC 2009 Entity Track. In *Proceedings of the Eighteenth Text REtrieval Conference (TREC 2009)*.
- [Wang et al., 2010] Wang, D., Wu, Q., Chen, H., and Niu, J. (2010). A Multiple-Stage Framework for Related Entity Finding: FDWIM at TREC 2010 Entity Track. In *Proceedings of the Niteenth Text REtrieval Conference (TREC 2010)*.
- [Wu and Weld, 2008] Wu, F. and Weld, D. S. (2008). Automatically refining the wikipedia infobox ontology. In *Proceeding of the 17th international conference on World Wide Web, WWW '08*, pages 635–644, New York, NY, USA. ACM.
- [Wu and Kashioka, 2009] Wu, Y. and Kashioka, H. (2009). NiCT at TREC 2009: Employing Three Models for Entity Ranking Track. In *Proceedings of the Eighteenth Text REtrieval Conference (TREC 2009)*.
- [Yang et al., 2009] Yang, Q., Jiang, P., Zhang, C., and Niu, Z. (2009). Experiments on Related Entity Finding Track at TREC 2009. In *Proceedings of the Eighteenth Text REtrieval Conference (TREC 2009)*.
- [Yang et al., 2010] Yang, Q., Jiang, P., Zhang, C., and Niu, Z. (2010). Reconstruct Logical Hierarchical Sitemap for Related Entity Finding. In *Proceedings of the Niteenth Text REtrieval Conference (TREC 2010)*.
- [Yarowsky, 1993] Yarowsky, D. (1993). One sense per collocation. In *Proceedings of the workshop on Human Language Technology, HLT '93*, pages 266–271, Stroudsburg, PA, USA. Association for Computational Linguistics.
- [Zhai and Lafferty, 2004] Zhai, C. and Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22:179–214.
- [Zhai et al., 2009] Zhai, H., Cheng, X., Guo, J., Xu, H., and Liu, Y. (2009). A Novel Framework for Related Entities Finding: ICTNET at TREC 2009 Entity Track. In *Proceedings of the Eighteenth Text REtrieval Conference (TREC 2009)*.

- [Zhao and Callan, 2008] Zhao, L. and Callan, J. (2008). A generative retrieval model for structured documents. In *Proceeding of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 1163–1172, New York, NY, USA. ACM.
- [Zheng et al., 2009] Zheng, W., Gottipati, S., Jiang, J., and Fang, H. (2009). UDEL/SMU at TREC 2009 Entity Track. In *Proceedings of the Eighteenth Text REtrieval Conference (TREC 2009)*.

APPENDIX A

HOMPAGE DETECTION FOR THE TREC TASK

The goal of homepage detection is to identify the corresponding homepages for answer entities. The entity retrieval tasks, such as TREC and INEX, define the entity’s homepages as the answers for the retrieval tasks. Therefore, in our TREC competition task, we have an additional step to match the answer entities to their homepages. There are three approaches used in previous TREC for homepage detection: relying on search engines, training a classifier for entity homepage detection, and relying on knowledge base to query entity homepages. The work of Vydiswaran relies on search engines, by building up the structured index with more weights on title and headline fields to find the most relevant documents as the entity’s homepages [Vydiswaran et al., 2009]. Some groups, such as [McCreadie et al., 2009] and [Kaptein and Kamps, 2009], use knowledge bases, like Wikipedia or DBpedia, to extract homepages for the target entities. The third method is to build a classifier for homepage identification, such as logistic regression in [Yang et al., 2009] and [Fang et al., 2009].

We adopted the classification method in entity homepage detection. Features listed in Fang’s work are used to train a classifier for homepage identification [Fang et al., 2009]. The features chosen for the classification are as follows. It includes the features of `isWebSite`, `type`, `isWiki`, `separator`, `urlContainsEntities`, `partInURL`, `hasAbout`, `hasIndex`, `hasWWW`. The details are as follows.

`isWebSite` indicates whether this website is official website or not,

`type` indicates what is the type of the entities, persons or organizations or products.

isWiki indicates whether this page is Wikipedia page or not.

separators indicates how many separators in the URL. The assumption is if the page is the homepage, then it should have few separators.

urlContainsEntities indicates whether the URL of this page include answer entities.

partInURL indicates whether the URL of this page include the part of answer entities.

hasAbout indicates whether the URL of this page include the term of “about”.

hasIndex indicates whether the URL of this page include the term of “index”.

hasWWW indicates whether the URL of this page starts with “WWW”.

The classification results using JRIP method are as shown in Figure 14.

```
Test mode: 10-fold cross-validation
JRIP rules:
=====
(isWebSite = official website) => cls=1 (100.0/0.0)
=> cls=0 (1473.0/155.0)
Number of Rules : 2
Time taken to build model: 0.16 seconds
=== Summary ===
Correctly Classified Instances   1418      90.1462 %
Incorrectly Classified Instances  155       9.8538 %
Kappa statistic                  0.5195
Mean absolute error              0.1764
Root mean squared error          0.297
Relative absolute error          64.8439 %
Root relative squared error      80.591 %
Total Number of Instances       1573
=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC
Area Class
      1      0.608  0.895  1      0.944  0.67  0
      0.392  0      1      0.392  0.563  0.67  1
Weighted Avg. 0.901  0.509  0.912  0.901  0.883
0.67
=== Confusion Matrix ===
  a  b  <-- classified as
1318  0 | a = 0
155 100 | b = 1
```

Figure 14: JRIP rules of entity homepage detection

The results indicates that the rules are similar to choose the top results from search engines as entity homepages. Therefore, the follow-up experiment focuses on how many results from search engines can be the homepages of entities. Yahoo!Boss is still used to find the homepage of entities. The results are in Table 21. The evaluation indicates that the

Table 21: Results of homepage detection

	# of correct entity	# of gth	# of entities hits	Precision	Recall	F
Top 5	53	167	5760	0.058	0.386	0.076
Top 4	50	167	4554	0.07	0.35	0.090
Top 3	50	167	3401	0.094	0.348	0.082
Top 2	50	167	2286	0.1376	0.348	0.1103
Top 1	45	167	1168	0.21	0.3	0.13

commercial search engines usually return the homepages, at the top, for the entity queries. In the final entity retrieval task, entity homepage detection uses the heuristic rule: if the homepage link from corresponding Wikipedia entity homepage are existing, then the answers in Wikipedia are as homepages; otherwise, the first hit from the search engine (Yahoo!Boss) is treated as homepages.

Entity homepage detection by searching on search engine can only find one fifth homepage. Although knowledge bases such as Wikipedia can also provide the answer for another one third, it is still a hard topic. One of the reasons for the failures of entity homepage detection is that the identical entities can be represented in different text surfaces. For example, both “Indiana University East” and “IU East” can be represented as the same entity, which can be referred to the same homepage (i.e., <http://www.iue.edu>). But in some cases, the abbreviation format of the entities will cause the difficulty of homepage identification. Another difficult is from the definition of the homepage. Some entities only have some webpages or webpage snippets to describe them. For example, the homepage sets for the topic of products of MedImmune, Inc. are in Table 22. The homepage of a product can be news, or product-related company’s homepage, or the product introduction page from its company, or the products homepage. In this case, it will be hard to define the homepage for some entities.

Table 22: Entity homepage sets for the topic of products of MedImmune, Inc.

Docno	URL	Type of the URL
5-HPclueweb09-en0000-27-129352	http://baltimore.bizjournals.com/baltimore/stories/2009/01/05/daily20.html	News
5-HPclueweb09-en0006-42-198412	http://www.ethyol.com/	Products Homepage
5-HPclueweb09-en0006-41-111382	http://www.flumist.com/	Products Homepage
5-HPclueweb09-en0008-26-393002	http://www.medimmune.com	Company Homepage
5-HPclueweb09-en0008-26-393062	http://www.medimmune.com/about/history.asp	Company Introduction page
5-HPclueweb09-en0008-26-393262	http://www.medimmune.com/products/ethyol/index.asp	Company Introduction page
5-HPclueweb09-en0008-26-393282	http://www.medimmune.com/products/flumist/index.asp	Company Introduction page
5-HPclueweb09-en0008-26-393302	http://www.medimmune.com/products/synagis/index.asp	Company Introduction page

APPENDIX B

ENTITIES OF PRODUCTS EXTRACTED FROM WIKIPEDIA INFOBOX

Company Name	# of Prod	Products
Google	139	AdWords Editor ; Google Chrome—Chrome ; Google Desktop—Desktop ; Google Earth—Earth ; Gmail/Google Notifier ; Google.Lively—Lively ;
Amazon.com	3	A9.com ; Alexa Internet ; Internet Movie Database—IMDb ;
Liberty Media	0	
eBay	7	online auction business model—Online auction hosting ; Electronic commerce ; Shopping mall ; PayPal ; Skype ; Gumtree ; Kijiji ;
Yahoo!	56	Bix ; blo.gs ; del.icio.us ; Dialpad ; Flickr ; Fire Eagle ; Kelkoo ; upcoming.org ; Jumpcut.com ; Zimbra ; Yahoo! 360
Microsoft	11	Microsoft Windows ; Microsoft Office ; Microsoft Servers ; Microsoft Visual Studio—Developer Tools ; Microsoft Expression Studio—Microsoft Expression ; Microsoft Dynamics—Business Solutions ; Microsoft Game Studios—Games ; Xbox 360 ; Windows Live ; Windows Mobile ; Zune ;

Company Name	# of Prod	Products
Oracle Corporation	14	Oracle Database ; Oracle Rdb ; Oracle eBusiness Suite ; Oracle Application Server ; JDeveloper—Oracle JDeveloper ; Oracle Application Framework ; Oracle Application Development Framework—Oracle ADF ; Oracle Beehive ; TimesTen ; Oracle Collaboration Suite ; Oracle Enterprise Manager ; Oracle Application Express ; Oracle Designer ; Oracle Developer Suite ;
Symantec	8	Symantec Endpoint Protection;Network Access Control 11.0; Control Compliance Suite; Security Information Manager; Brightmail;
SAP AG	9	AP Business Suite ; SAP ERP ; SAP Customer Relationship Management (SAP CRM) ; SAP Supply Chain Management (SAP SCM) ; SAP Supplier Relationship Management (SAP SRM) ; SAP Product Lifecycle Management (SAP PLM) ; SAP NetWeaver ; SAP Business One ; SAP Business All-in-One ;
ExxonMobil	3	Fuels ; Lubricants ; Petrochemicals ;
Chevron Corporation	7	Oil ; Petroleum ; Natural Gas ; Petrochemical ; Fuel ; Lubricant ;
ConocoPhillips	6	Oil ; Natural Gas ; Petroleum ; Lubricant ; Petrochemical ;
Valero Energy Corporation	1	Petrochemical
Marathon Oil	1	Petrochemical
Sunoco	1	Petrochemical
Hess Corporation	1	Petrochemical
Tesoro	1	Petroleum products
Frontier Oil	1	Petrochemical
Total	265	

APPENDIX C

A SAMPLE DOCUMENT OF INEX 2007: “NEXT”

```
<article>
<name id=“40642”>NEXTSTEP</name>
<conversionwarning>0</conversionwarning><body>
<figure>
<image xmlns:xlink=“http://www.w3.org/1999/xlink” xlink:type=“simple”
xlink:href=“../pictures/NeXTSTEP_desktop.jpg” id=“60698” xlink:actuate=“onLoad”
xlink:show=“embed”>
NeXTSTEP_desktop.jpg
</image><caption>
NeXTSTEP Desktop
</caption></figure><emph3>
NEXTSTEP
</emph3> is the original
<collectionlink xmlns:xlink=“http://www.w3.org/1999/xlink” xlink:type=“simple”
xlink:href=“22757.xml”>
object-oriented
</collectionlink>,
<collectionlink xmlns:xlink=“http://www.w3.org/1999/xlink” xlink:type=“simple”
xlink:href=“6857.xml”>
multitasking
</collectionlink><collectionlink xmlns:xlink=“http://www.w3.org/1999/xlink”
xlink:type=“simple” xlink:href=“22194.xml”>
operating system
</collectionlink> that
<collectionlink xmlns:xlink=“http://www.w3.org/1999/xlink” xlink:type=“simple”
xlink:href=“21694.xml”>
NeXT Computer
</collectionlink>, Inc. developed to run on its proprietary
NeXT computers (informally known as “black boxes”).
NeXTSTEP 1.0 was released on
<unknownlink src=“18 September”>
18 September
</unknownlink><collectionlink xmlns:xlink=“http://www.w3.org/1999/xlink”
xlink:type=“simple”
xlink:href=“34847.xml”>
1989
</collectionlink> after several previews starting in
<collectionlink xmlns:xlink=“http://www.w3.org/1999/xlink” xlink:type=“simple”
xlink:href=“34761.xml”>
1986
```

```

</collectionlink>, and the last release 3.3 in early
<collectionlink xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
xlink:href="34658.xml">
1995
</collectionlink>, by which time it ran not only on
<collectionlink xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
xlink:href="20319.xml">
Motorola
</collectionlink><collectionlink xmlns:xlink="http://www.w3.org/1999/xlink"
xlink:type="simple" xlink:href="64826.xml">
68000 family
</collectionlink> processors (specifically the original black boxes), but also generic IBM compatible
x86/Intel, Sun
.....
<language link lang="de">
NeXTStep
</language link><language link lang="es">
NEXTSTEP
</language link><language link lang="fr">
NeXTSTEP
</language link><language link lang="it">
NeXTSTEP
</language link><language link lang="ja">
NEXTSTEP
</language link><language link lang="no">
NeXTSTEP
</language link><language link lang="pl">
NeXTStep
</language link></section>
</body>
</article>

```


APPENDIX D

A SAMPLE DOCUMENT OF INEX 2009:“NEXT”

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- generated by CLiX/Wiki2XML [MPI-Inf, MMCI@UdS] LastChangedRevision : 92 on
16.04.2009 15:41:13[mciao0826] -->
<!DOCTYPE article SYSTEM "../article.dtd">
<article xmlns:xlink="http://www.w3.org/1999/xlink">
<O confidence="0.9508927676800064" wordnetid="106832680">
<header>
<title>Nextstep</title>
<id>40642</id>
<revision>
<id>239398792</id>
<timestamp>2008-09-18T23:05:07Z</timestamp>
<contributor>
<username>Aldie</username>
<id>901</id>
</contributor>
</revision>
<categories>
<category>Window-based operating systems</category>
<category>BSD</category>
<category>Discontinued software</category>
<category>NeXT</category>
<category>Mach</category>
</categories>
</header>
<bdy>
<template>
<name>Infobox_OS</name>
<parameters>
<screenshot>
<image width="230px" src="NeXTSTEP_desktop.jpg">
</image>
</screenshot>
<supported_platforms>
<chip wordnetid="103020034" confidence="0.8">
<artifact wordnetid="100021939" confidence="0.8">
<instrumentality wordnetid="103575240" confidence="0.8">
<microprocessor wordnetid="103760310" confidence="0.8">
<conductor wordnetid="103088707" confidence="0.8">
<device wordnetid="103183080" confidence="0.8">
<semiconductor_device wordnetid="104171831" confidence="0.8">
<link xlink:type="simple" xlink:href="../270/20270.xml">
Motorola 68000</link></semiconductor_device>
</device>
</conductor>
</microprocessor>
</instrumentality>
</artifact>
```

```

</chip>
.....
</template>
<b>Nextstep</b> was the original <link xlink:type="simple"
xlink:href=" ../109/230109.xml">
object-oriented</link>, <link xlink:type="simple" xlink:href=" ../857/6857.xml">
multitasking</link> <link xlink:type="simple" xlink:href=" ../194/22194.xml">
operating system</link> that <company wordnetid="108058098"
confidence="0.9508927676800064">
<link xlink:type="simple" xlink:href=" ../694/21694.xml">
NeXT Computer</link></company>
developed to run on its range of proprietary computers, such as the <computer
wordnetid="103082979" confidence="0.8">
<occupation wordnetid="100582388" confidence="0.8">
<artifact wordnetid="100021939" confidence="0.8">
<instrumentality wordnetid="103575240" confidence="0.8">
<event wordnetid="100029378" confidence="0.8">
<device wordnetid="103183080" confidence="0.8">
<machine wordnetid="103699975" confidence="0.8">
<digital_computer wordnetid="103196324" confidence="0.8">
<act wordnetid="100030358" confidence="0.8">
<psychological_feature wordnetid="100023100" confidence="0.8">
<activity wordnetid="100407535" confidence="0.8">
<workstation wordnetid="104603399" confidence="0.8">
<link xlink:type="simple" xlink:href=" ../717/2886717.xml">
NeXTcube</link></workstation>
</activity>
</psychologica_feature>
</act>
</digital_computer>
</machine>
</device>
</event>
</instrumentality>
</artifact>
</occupation>
</computer>
. Nextstep 1.0 was released on <link xlink:type="simple" xlink:href=" ../146/28146.xml">
September 18</link>, <link xlink:type="simple" xlink:href=" ../847/34847.xml">
1989</link> after several previews starting in <link xlink:type="simple"
xlink:href=" ../761/34761.xml">
1986</link>. The last version, 3.3, was released in early <link xlink:type="simple"
xlink:href=" ../658/34658.xml">
1995</link>, by which time it ran not only on <company wordnetid="108058098"
confidence="0.9508927676800064">
<link xlink:type="simple" xlink:href=" ../319/20319.xml">
Motorola</link></company>
<link xlink:type="simple" xlink:href=" ../826/64826.xml">
68000 family</link> processors, but also <link xlink:type="simple"
xlink:href=" ../803/49803.xml">
IBM PC compatible</link> <link xlink:type="simple" xlink:href=" ../198/34198.xml">
x86</link>, Sun <link xlink:type="simple" xlink:href=" ../954/36954.xml">
SPARC</link>, and HP <link xlink:type="simple" xlink:href=" ../970/24970.xml">
PA-RISC</link>. <company wordnetid="108058098" confidence="0.9508927676800064">
<link xlink:type="simple" xlink:href=" ../856/856.xml">
Apple Inc.</link></company>
's <link xlink:type="simple" xlink:href=" ../640/20640.xml">
Mac OS X</link> is a direct descendant of Nextstep.
<sec>
</p>

</sec>
</bdy>
</O>
</article>

```

APPENDIX E

TWENTY TOPICS IN TREC 2009 ENTITY TRACK

```
<query>
<num>1</num>
<entity_name>Blackberry</entity_name>
<entity_URL>clueweb09-en0004-50-39593</entity_URL>
<target_entity>organization</target_entity>
<narrative>Carriers that Blackberry makes phones for.</narrative>
</query>
```

```
<query>
<num>2</num>
<entity_name>ACM Athena award</entity_name>
<entity_URL>clueweb09-en0004-21-12770</entity_URL>
<target_entity>person</target_entity>
<narrative>Winners of the ACM Athena award.</narrative>
</query>
```

```
<query>
<num>3</num>
<entity_name>Claire Cardie</entity_name>
<entity_URL>clueweb09-en0009-89-01791</entity_URL>
<target_entity>person</target_entity>
<narrative>Students of Claire Cardie.</narrative>
</query>
```

```
<query>
<num>4</num>
<entity_name>Philadelphia, PA</entity_name>
<entity_URL>clueweb09-en0011-13-07330</entity_URL>
<target_entity>organization</target_entity>
<narrative>Professional sports teams in Philadelphia.</narrative>
</query>
```

```
<query>
<num>5</num>
<entity_name>MedImmune, Inc.</entity_name>
<entity_URL>clueweb09-en0008-26-39300</entity_URL>
<target_entity>product</target_entity>
<narrative>Products of MedImmune, Inc.</narrative>
</query>
```

```
<query>
<num>6</num>
<entity_name>Nobel Prize</entity_name>
<entity_URL>clueweb09-en0002-23-19459</entity_URL>
<target_entity>organization</target_entity>
<narrative>Organizations that award Nobel prizes.</narrative>
</query>
```

<query>
<num>7</num>
<entity_name>Boeing 747</entity_name>
<entity_URL>clueweb09-en0005-75-02292</entity_URL>
<target_entity>organization</target_entity>
<narrative>Airlines that currently use Boeing 747 planes.</narrative>
</query>

<query>
<num>8</num>
<entity_name>The King's Singers</entity_name>
<entity_URL>clueweb09-en0002-63-29621</entity_URL>
<target_entity>product</target_entity>
<narrative>CDs released by the King's Singers.</narrative>
</query>

<query>
<num>9</num>
<entity_name>The Beaux Arts Trio</entity_name>
<entity_URL>clueweb09-en0005-08-02741</entity_URL>
<target_entity>person</target_entity>
<narrative>Members of The Beaux Arts Trio.</narrative>
</query>

<query>
<num>10</num>
<entity_name>Indiana University</entity_name>
<entity_URL>clueweb09-en0007-37-37513</entity_URL>
<target_entity>organization</target_entity>
<narrative>Campuses of Indiana University.</narrative>
</query>

<query>
<num>11</num>
<entity_name>Home Depot Foundation</entity_name>
<entity_URL>clueweb09-en0009-23-04855</entity_URL>
<target_entity>organization</target_entity>
<narrative>Donors to the Home Depot Foundation.</narrative>
</query>

<query>
<num>12</num>
<entity_name>Air Canada</entity_name>
<entity_URL>clueweb09-en0004-24-03450</entity_URL>
<target_entity>organization</target_entity>
<narrative>Airlines that Air Canada has code share flights with.</narrative>
</query>

<query>
<num>13</num>
<entity_name>American Veterinary Medical Association (AVMA)</entity_name>
<entity_URL>clueweb09-en0004-39-32528</entity_URL>
<target_entity>product</target_entity>
<narrative>Journals published by the AVMA.</narrative>
</query>

<query>
<num>14</num>
<entity_name>Bouchercon 2007</entity_name>
<entity_URL>clueweb09-en000508-25203</entity_URL>
<target_entity>person</target_entity>
<narrative>Authors awarded an Anthony Award at Bouchercon in 2007.</narrative>
</query>

<query>
<num>15</num>
<entity_name>SEC conference</entity_name>
<entity_URL>clueweb09-en0010-56-11826</entity_URL>
<target_entity>organization</target_entity>
<narrative>Universities that are members of the SEC conference for football.</narrative>
</query>

<query>
<num>16</num>
<entity_name>Mancuso Quilt Festivals</entity_name>
<entity_URL>clueweb09-en0011-22-08631</entity_URL>
<target_entity>organization</target_entity>
<narrative>Sponsors of the Mancuso quilt festivals.</narrative>
</query>

<query>
<num>17</num>
<entity_name>The Food Network</entity_name>
<entity_URL>clueweb09-en0006-55-17239</entity_URL>
<target_entity>person</target_entity>
<narrative>Chefs with a show on the Food Network.</narrative>
</query>

<query>
<num>18</num>
<entity_name>Jefferson Airplane</entity_name>
<entity_URL>clueweb09-en0009-25-04698</entity_URL>
<target_entity>person</target_entity>
<narrative>Members of the band Jefferson Airplane.</narrative>
</query>

<query>
<num>19</num>
<entity_name>John L. Hennessy</entity_name>
<entity_URL>clueweb09-en0011-14-04774</entity_URL>
<target_entity>organization</target_entity>
<narrative>Companies that John Hennessy serves on the board of.</narrative>
</query>

<query>
<num>20</num>
<entity_name>Isle of Islay</entity_name>
<entity_URL>clueweb09-en0008-96-25389</entity_URL>
<target_entity>organization</target_entity>
<narrative>Scotch whisky distilleries on the island of Islay.</narrative>
</query>