LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-TR-450951

ECO: A Framework for Entity Co-Occurrence Exploration with Faceted Navigation

K. D. Halliday

August 24, 2010

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

ECO: A Framework for Entity Co-Occurrence Exploration with Faceted Navigation

M.S. Final Project
Computer Science Department
California State University, Chico

Kyle Halliday
November 9, 2010

Committee Chair: Dr. Anne Keuneke
Committee Member: Dr. Ben Juliano

LLNL-TR-450951

Abstract

Even as highly structured databases and semantic knowledge bases become more prevalent, a substantial amount of human knowledge is reported as written prose. Typical textual reports, such as news articles, contain information about entities (people, organizations, and locations) and their relationships. Automatically extracting such relationships from large text corpora is a key component of corporate and government knowledge bases. The primary goal of the ECO project is to develop a scalable framework for extracting and presenting these relationships for exploration using an easily navigable faceted user interface. ECO uses entity co-occurrence relationships to identify related entities. The system aggregates and indexes information on each entity pair, allowing the user to rapidly discover and mine relational information.

Table of Contents

Table of Figures

1.0 Introduction

1.1 Purpose

Discovering relationships in text documents is an important problem in many domains, from financial analysis (e.g. company A acquired company B), to biological research (e.g. pathogen A causes disease B), to law enforcement and intelligence analysis (e.g. person A kidnapped person B). For humans, reading and understanding large volumes of text is a time consuming task. For computers, simply identifying entities and their semantic relationships accurately is a challenge. Even more difficult is deriving knowledge from these relationships, something humans can do well, but machines struggle with.

The purpose of this project is to develop a system called ECO that combines the strengths of computers with the strengths of humans, and to build a framework allowing humans to rapidly explore relationships between entities in large volumes of text.

1.2 Problem Statement

Natural language processing (NLP) [1,2], information retrieval (IR) [3,4], applied graph theory [5-7], and human-computer interaction (HCI) [8,9] are all active research areas in computer science. This project combines recent advancements from each of these fields to build a system for humans to explore machine-extracted entity relationships. While many of the research efforts surveyed in this project have tended to focus on solving particular problems more completely with computers (see Chapter 3), the approach of this project is to keep humans in the loop. The ECO framework gives users an opportunity to interact with entity relationship information in an exploratory mode.

1.3 Report Overview

Chapter 2 provides general background information about Natural Language Processing, applicable elements of graph theory, and recent progress in user interface design. Chapter 3 outlines recent research in these areas, providing a comprehensive context for the ECO project. The method and approach are described in detail in Chapter 4, which includes descriptions of the primary software components. The hardware and software infrastructure used to construct the ECO framework is explained in Chapter 5, followed by conclusions and suggestions for future work in Chapter 6 and a list of references in Chapter 7. The ECO software architecture is depicted using UML diagrams in Appendix A.

2.0 Background

This chapter provides some general background in relevant computer science research topics and how they relate to the ECO project. The subsequent chapter details specific research efforts.

2.1 Natural Language Processing

Natural Language Processing (NLP) is an active research field in computer science and computational linguistics that involves using computers to extract meaningful information from written prose. Such text, in the form of a news article, for example, contains inherent grammatical structure. Paragraphs, sentences, parts of speech, and punctuation are all grammatical structural elements. Another form of structure is linguistic semantics, which captures the meaning of the language. A major goal of NLP

in the literature surveyed here and in the ECO project is to extract both grammatical and semantic structure from text for the purposes of storing it in a knowledge base.

To illustrate the differences between written text and a semantically structured representation of the same information contained in the text, consider the Wikipedia [10] and DBpedia [11] projects. Wikipedia is a collaborative, wiki-based encyclopedia with articles written in standard prose. DBpedia extracts structured information from Wikipedia articles and represents the information in a semantic graph using Resource Description Framework (RDF) triples [12]. Entities are categorizing using a structured ontology. Wikipedia is intended for human consumption, while DBpedia is intended for computerized consumption.

From 1987 – 1997, the Defense Advanced Research Projects Agency (DARPA) sponsored a series of Message Understanding Conferences (MUC) [13] to spur research and development in information extraction. Following the MUC series, the National Institute of Standards and Technology (NIST) and Linguistic Data Consortium (LDC) sponsored Automatic Content Extraction (ACE) [14] and Text Analysis Conference (TAC) [15] efforts to further the state-of-the-art. Since 1992, the Text Retrieval Conference (TREC) [16], an annual workshop series currently co sponsored by the NIST and the Intelligence Advanced Research Projects Agency (IARPA), has provided a forum for NLP research activities.

2.2 Named Entity Recognition

Sub-tasks within the field of NLP include named entity recognition (NER), relationship extraction, event extraction, text summarization, textual entailment, and more [17]. Of particular interest to this project is NER, which involves automatically

extracting named entities (NEs), typically proper nouns referencing real-world persons, organizations, or locations.

To illustrate NER, consider the following sentence:

> According to witnesses, John Wilkes Booth was seen near Ford's Theater on the night of Abraham Lincoln's assassination.

This sentence contains mentions of two people and a location. NER algorithms parse the sentence and decide which pieces of text represent these named entities. Given a block of text as input, NER software identifies what it believes to be entities by annotating the text with character offsets and entity types. Here is the same sentence with the entities highlighted. Person entities are highlighted in yellow, and the place entity is highlighted in green.

> According to witnesses, John Wilkes Booth was seen near Ford's Theater on the night of Abraham Lincoln's assassination.

Identifying entities in text is a relatively straightforward task for humans, but even the best automatic entity extractors only achieve around 90% precision [18].

2.3 Entity Co-Occurrences

When two named entities appear in the same block of text, they are said to co-occur within that context. When the context is small enough, there is a good chance that the entity pair will represent an actual semantic relationship. The task is to effectively extract a relationship without specifying the exact nature of the relationship, other than the fact that they appear close together in the source text.

Many more complex approaches to relationship extraction have been proposed and implemented, a few of which are highlighted in the literature review. For the ECO

project the simple approximation described above will suffice for the purposes of displaying related entities in the ECO user interface.

Revisiting the example from the previous section, it is clear that John Wilkes Booth and Abraham Lincoln are related. The ECO framework captures this knowledge and stores it in a search index. A user seeking information about either one of these entities can quickly discover a relationship to the other entity.

When processing a large corpus, the same pair of entities may co-occur in multiple documents. Consider the following two annotated sentences:

Sentence 1, extracted from document A:
According to witnesses, John Wilkes Booth was seen near Ford's Theater on the night of Abraham Lincoln's assassination.

Sentence 2, extracted from document B:
John Wilkes Booth, a stage actor and Confederate sympathizer, had initially plotted with Samuel Arnold and Michael O'Laughlen to kidnap Abraham Lincoln.

The ECO system aggregates multiple co-occurrences for each pair of entities encountered throughout the corpus. Collectively, the contexts from which the pair of entities was extracted can be thought of as the set of text blocks telling us how two entities are connected. Re-assembling these blocks as a "virtual document" allows the ECO user interface to provide additional information about how two entities are connected. Since all of the sentences concerning a particular entity pair are stored together, they can later be queried for key terms summarizing the relationships between the two entities.

2.4 Entity Co-Occurrences as a Graph

Entity co-occurrences can be represented as an undirected, weighted graph, where the nodes are entities, and the edges represent the fact that the two entities co-occurred

within one or more contexts. The edge weights are the total number of times those two entities co-occurred across the corpus. The corpus-wide set of contexts connecting two entities can be considered an attribute of the edge.

After automatically identifying entities, the ECO system arranges the information into what can be conceptually viewed as a co-occurrence graph. The co-occurrence graph generated from the above text is shown in Figure 1. All of the edge weights are 1, except for the edge connection John Wilkes Booth and Abraham Lincoln, since the appeared in two co-occurrences.

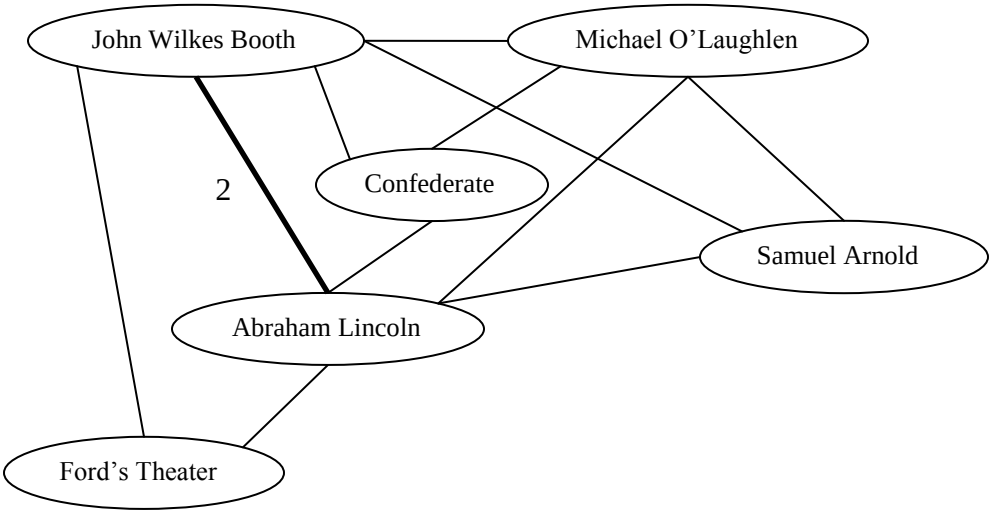

Fig. 1 Co-occurrence graph for the two example sentences

2.5 Faceted Navigation

Faceted navigation is a user-interface technique used for the exploration and discovery of multi-dimensional information. Faceted navigation divides metadata describing a set of records into categories, or facets. Beneath each facet displayed in a user interface, a number of facet values are displayed, along with a count of the number of items that are tagged with that facet. Selecting a facet value reduces the search space in

that dimension, and the display is updated to show only the records having that facet value. In addition, the facet values and counts are updated to reflect the new subset of records being displayed.

The currently selected facet values act as breadcrumbs, showing the user which selections they have already made. These breadcrumbs may be deselected in any order, effectively expanding the search space in that dimension. Figure 2 depicts the faceted navigation interface at the website of the city of Raleigh, North Carolina [19].

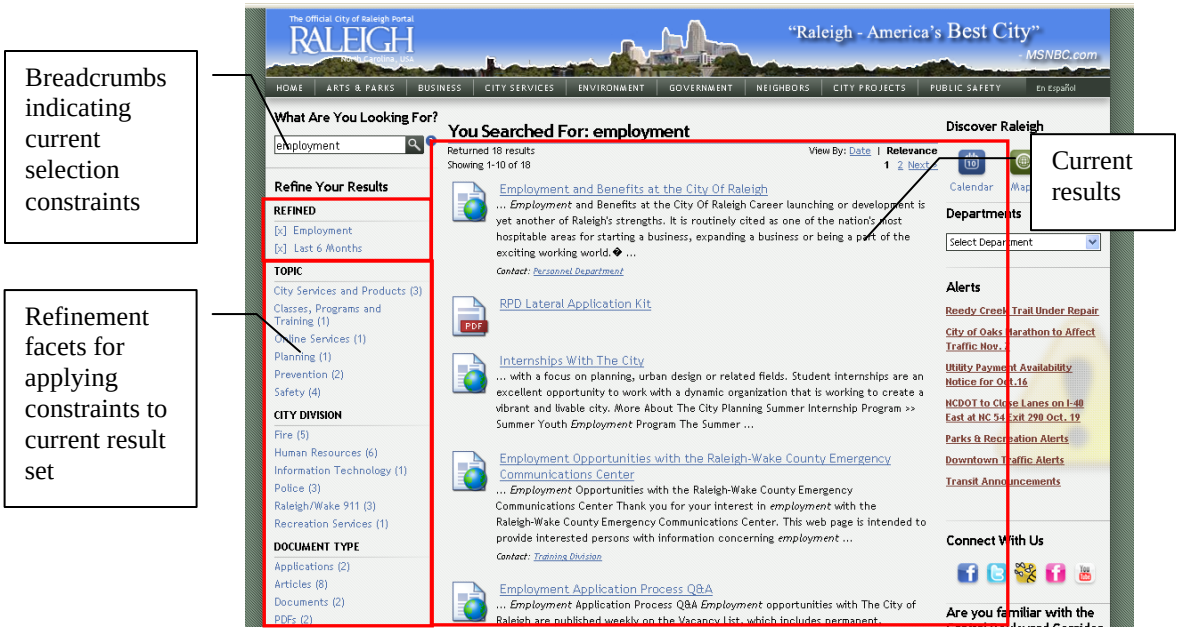

Fig. 2 Faceted navigation interface at the website of the city of Raleigh, North Carolina

Faceted navigation has seen widespread adoption across the web, ranging from commercial websites such as eBay and Amazon to digital library catalogs, and faceted navigation is now supported in both commercial and open-source software packages.

2.6 Entity Co-Occurrences in Faceted Navigation

Representing entity co-occurrence data using faceted navigation means displaying all of the extracted entities under a single "Entity" facet. The entities are listed in descending order of co-occurrence frequency, and the frequency count is displayed alongside each entity. It is possible to either search or browse the list of entities. After selecting an entity from this list, the user is presented with a second list of entities in another column, but this time the only entities displayed are those that co-occur with the initially selected entity. The second column can expand to a third, and so on. Section 4.5 provides a more detailed description of the ECO user interface.

3.0 Literature Review

The importance of co-occurrence data in information retrieval has been well known since the late seventies [20]. Since that time, a number of advances have been made in natural language processing, including the ability to extract named entities and their relationships from text. This survey of related work describes the relevant research efforts and their applicability to the ECO project.

3.1 Entity Co-Occurrence and Relationship Extraction

Co-Occurrence Clustering in Raw Text

Li and Liu [21] described an entity clustering algorithm which uses agglomerative clustering, mutual information, and graph triangle link structure to identify community structures from named entities in free text. Their algorithm differs from many other clustering algorithms in that it allows an entity to be in more than one cluster. Although

ECO does not address clustering, the entity co-occurrence extraction methodology described in this paper forms the basis for the ECO extraction framework.

Entity Relationship Extraction

Hasegawa, Sekine, and Grishman [1] proposed a fully unsupervised algorithm for extracting relationships between named entity pairs by clustering entities with similar context words found between the two entities. Their method uses hierarchical clustering so the number of clusters does not need to be specified in advance. They found related co-occurrences by comparing the cosine similarity of the context word vectors for entity pairs. They then used this information to label clusters using frequently occurring context words. Finally, they propagated the cluster label as the relationship label.

Zhang et al. [22] described a variant of Hasegawa's method which uses tree-similarity metrics over parse trees of the context instead of cosine similarity. Their method does not require the assumption that the same entity pairs in different contexts have the same relationship. This approach yielded better results for instances where the number of co-occurrences was less frequent.

Rather than extract relationships at the time of data ingest, ECO will summarize relationships in an on-demand fashion. As the user navigates the co-occurrence data, no relationship information will be presented. The user will know that entities are related based on co-occurrence, but relationship summary details are only displayed once the user selects two specific entities to examine further.

Entity Relationship Extraction Using Web Search Engines

Using web search engines to measure the strength of relationship between entities, Matsuo et al. developed the POLYPHONET system [23]. They compared multiple co-occurrence metrics and found the overlap coefficient to be the most effective for their purposes.

Jin, Matsuo, and Ishizuka [5] offered improvements to identify relationships based on key terms and tuned the results based on relationship strength thresholds.

Magnini et al. [24] applied co-occurrence statistics to improve the quality of their DIOGENE Question Answering system. Their approach is to validate candidate answers to a question by posing co-occurrence queries to web search engines. They then compare the results for each potential answer to calculate which answer is most likely to be correct.

Since these approaches use web search engines to calculate entity relationship metrics, they effectively require the document content to be pre-indexed. One design goal of ECO, however, is to calculate co-occurrence metrics in a streaming fashion, so computing metrics beyond raw co-occurrence frequency is beyond the scope of the ECO project.

Co-Occurrence Search Engines

In the development of their KIM semantic search engine, Popov et al. [25] present an extension called CORE (Co-Occurrence and Ranking of Entities). They use this module in a mixed-mode IR system, which stores semantic information in an RDF/OWL data store and documents, extracted entities, and other annotations in a relational database. Their approach presents results in a faceted search interface, resulting in a

relationship navigation browser similar to the ECO user interface. Key differences between this system and ECO are:

1) CORE search splits entities by type, while ECO will combine all entities into a single "Entity" class.
2) CORE search uses entire documents as a co-occurrence context, while ECO will use sentence-level co-occurrences.
3) ECO will display the co-occurrence frequency for each entity and rank the entities accordingly.
4) ECO will provide a means for "drilling down" into a co-occurrence relationship.

Entity Language Models

Conrad and Utt [26] described both word distance and context similarity approaches in generating links between named entities. Before creating a relationship link between two entities, they first calculated the strength of association between the entities within a fixed-size context window surrounding each entity. The user interface for their Associations System displays the relationship strength metric in addition to the raw co-occurrence frequency. They also describe the technique of using entity-centric pseudo-documents to enhance the results and display context information about an entity in the user interface. ECO will use a similar technique, but instead of creating a pseudo-document for each entity, it will create one for each entity co-occurrence.

Raghavan, Allan, and McCallum [27] formally described Conrad and Utt's approach as an "entity model", considering it a middle ground between information retrieval for unstructured data and data mining for structured data. They applied this technique to clustering, classification, and question answering.

Petkova and Croft [28] extended this work by describing a method for modeling named entities based on combining nearby text snippets without relying on sentence parsing or other forms of document structure. Their work provides a formal model for describing the dependency between entities and document terms.

ECO follows this general approach, but instead uses an "entity co-occurrence model", capturing information for pairs of entities rather than for individual entities.

3.2 Domain-Specific Text Analysis Applications

Two particular domains where such automated knowledge extraction systems are used are law enforcement and intelligence analysis. Link analysis is technique used to extract and search associations between people, organizations, locations, and other entities of interest. Automated systems enable analysts to examine larger datasets than they would be able to read manually.

Schroeder, Xu, and Chen address information overload problems in their CrimeLink Explorer application with three techniques: the concept space approach, incorporation of domain knowledge, and shortest-path algorithms [29]. This system uses co-occurrence weights as one metric for associating criminals in crime incident reports and allows users to search for criminal associations over multiple degrees of connectedness.

In another research effort, Xu and Chen evaluate the effectiveness of shortest-path graph search algorithms in criminal networks [6]. They used shortest-path, priority-first-search, two-tree priority-first-search, and breadth-first-search to find the strongest associations amongst entities. They found that priority-first-search returned useful

association paths approximately 70% of the time, while breadth-first-search only returned useful results at a 15-30% rate, depending on the type of criminal network.

3.3 Faceted Navigation

Faceted navigation builds on the concept of faceted classification, a type of information classification proposed by Ranganathan [30] as an alternative to the Dewey Decimal system. The fundamental concept behind his Colon Classification scheme is that a single publication can be classified into multiple categories based on different aspects, or facets, contained within the content of the publication.

Initial user interfaces for dealing with faceted datasets were developed by Pollitt [31] and were called view-based systems. The interface displayed multiple simultaneous views of the data, each from a different perspective. As constraints were applied in one view, the other views updated accordingly.

Modern faceted navigation user interfaces were described in detail by Hearst et al. [32] in the development of the Flamenco project. Hearst has continued to publish research in this area [33,34].

ECO uses a simplified faceted navigation scheme, using only a single facet. As the user explores the co-occurrence relationships, the facet is replicated multiples times, and the facet values displayed differ based on the current navigation state.

4.0 Method

For this project, a corpus of business news articles was assembled. The ECO system extracted the named entities and identified sentence-level entity co-occurrences. The co-occurrence dataset was loaded into two Apache Solr [35] indexes, one to store the

co-occurrence information, and the other to store entity co-occurrence virtual documents for top term identification. Finally, a facet-based web user interface was constructed for rapidly exploring the co-occurrence relationships.

4.1 Text Corpus

The text corpus consisted of a subset of news articles from the New York Times corpus [36]. Lawrence Livermore National Laboratory (LLNL) has an institutional agreement with the Linguistic Data Consortium allowing access to this dataset for research purposes. The full corpus consists of more than 1.8 million articles, but only non-statistical business articles (articles with a business classifier value of "Top/News/Business", and not having a type-of-material value of "Statistics") were selected for processing. In total, this subset corpus contains 227,936 articles.

Although the corpus includes extracted entities in the metadata, entity extraction was performed independently so that the ECO system could identify sentence-level co-occurrences. Performing the extraction separately also makes the software more easily adapted for use with other corpora.

4.2 Entity Extraction

The ECO system utilizes the Stanford Named Entity Recognizer [37] developed by Finkel et al. for named-entity recognition and the Stanford Parser [38,39] developed by Klein and Manning for sentence-level text parsing. Both of these packages are open-source software and freely available for research purposes.

4.3 Faceted Navigation Database

The Apache Solr [35] open-source software package, which supports faceted navigation, was used to store the co-occurrence information. Solr indexes are the primary data storage mechanism used by the ECO framework, and ECO retrieves information from the indexes by submitting queries with the Solr API.

4.4 Data Processing

The ECO data processing code reads news articles from text files. It parses the sentences of each article and performs named entity recognition. It identifies sentence-level entity co-occurrences and writes the co-occurrence data to a series of XML files. The ECO software utilizes the open-source XStream library to serialize co-occurrence objects to XML [40]. The XML files are then loaded into the primary Apache Solr index, which stores the co-occurrence information. Each entity co-occurrence record has the following fields:

Field Name	Description
ID	A unique identifier for each entity co-occurrence record
Entity	The named entity label. Each record has two values for the Entity field
Context	The sentence containing the entity co-occurrence
Article_ID	The ID of the article from which the co-occurrence was extracted
Article_URL	The article URL, which the ECO user interface uses to retrieve the document for display.

Below is a sample XML record:

```
<doc>
    <field name="ID">b6d5d5e5-6d5a-4e7f-adba-cf86bcc2c619</field>
    <field name="Entity">TGV</field>
    <field name="Entity">Amtrak</field>
    <field name="Context">Europe has the bullet-like TGV&amp;apos;s and
tilting trains; the United States has Amtrak.</field>
    <field name="Article_ID">0910125</field>
    <field name="Article_URL">1997/02/20/0910125.txt</field>
</doc>
```

A secondary data processing stage assembles the co-occurrence data from the first index into a set of entity co-occurrence virtual documents. That is, for each pair of co-occurring entities, all of the contexts are concatenated together and stored in another index. This combines all of the information about two entities from across the corpus into a single virtual document, from which relationship summarization information can be gleaned. Each entity co-occurrence virtual document record has the following fields:

Field Name	Description
Key	A key identifying the entity pair. The key is generated by lexicographically sorting the entity pair and concatenating them together with an underscore character.
Contexts	All of the sentences containing the entity pair concatenated together

Below is a sample XML record:

```
<doc>
    <field name=”Key”>Microsoft_Yahoo</field>
    <field name=”Contexts”>But Yahoo and Microsoft have become direct
competitors, and a number of start-up companies are busy developing
search technologies. Yahoo said yesterday that it was negotiating with
Microsoft and hoped it would remain a customer.</field>
</doc>
```

The dataset is made available to the ECO user interface through these two indexes. A conceptual diagram of the ECO processing pipeline is shown in Figure 3.

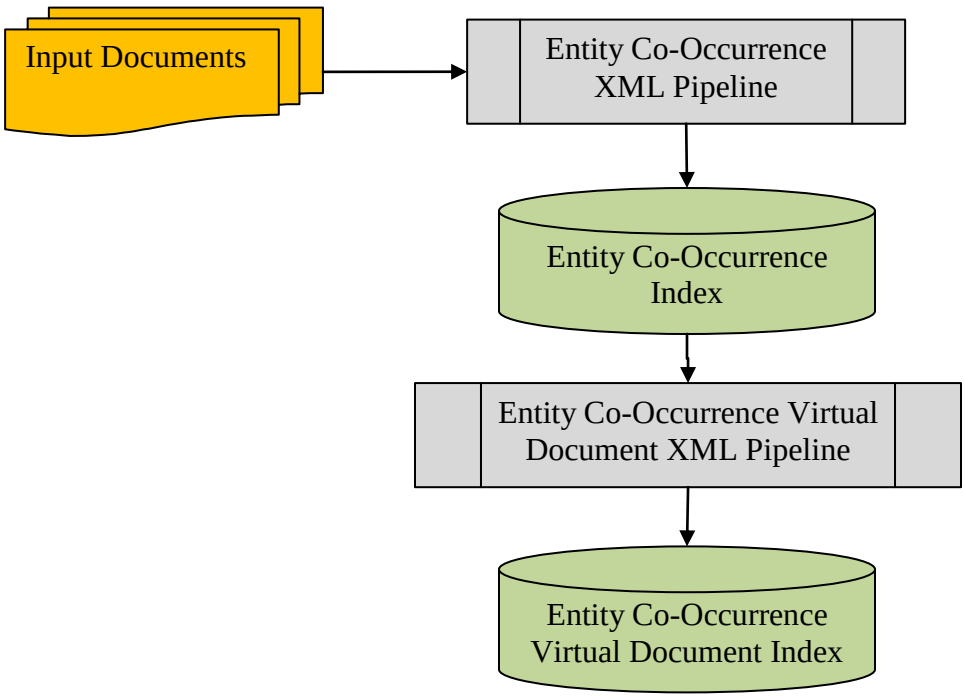

Fig. 3. Conceptual diagram of the ECO processing pipeline

The data processing code is configured to ignore sentences with more than 20 entities, since such an entity-dense block of text is likely to not be a sentence at all. For instance, in initial test runs, the code found a "sentence" which was actually a large table in the original news article, and the sentence parser was not able to differentiate records within the table. This threshold is intended to avoid combinatorial explosions of co-occurrences.

The ECO data processing code is designed to operate in serial or parallel modes. In serial mode, the code runs on a single machine. In parallel mode, the code is executed on multiple nodes of a cluster simultaneously using the SLURM utility.

To distribute the workload in parallel mode, each node of the cluster processes every N^{th} file in the input directory, where N is the total number of nodes used for the

computation. Each node starts with an offset of (n – 1), where n is the node number. This technique is illustrated in Figure 4.

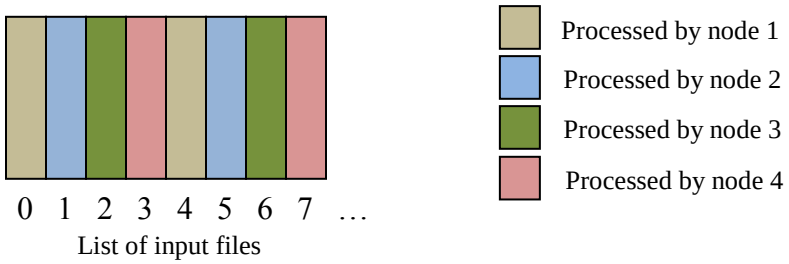

Fig. 4. ECO parallel processing workload distribution with four nodes

In serial mode, the entity co-occurrence XML generation code is run as a standard Java program:

```
java kdh.eco.pipeline.ECOXMLPipeline <input-dir> <output-dir>
```

and accepts the following input parameters:

1) **<input-dir>**: directory containing news articles
2) **<output-dir>**: directory where the XML output files are written

In parallel mode, the code is run using the SLRUM srun command:

```
srun –l –w <node-selector-expression> java kdh.eco.pipeline.ECOXMLPipeline
  <input-dir> <output-dir> <node-selector-expression>
```

and accepts the following input parameters:

1) **<input-dir>**: directory containing news articles
2) **<output-dir>**: directory where the XML output files are written
3) **<node-selector-expression>**: an expression specifying which nodes to use to run the job. This value can either be a hostname (e.g. node1) or an expression of the form node[1-2,4-5,7], where there may be an arbitrary number of comma-separated node numbers or numeric ranges within the square brackets.

The entity co-occurrence virtual document XML generation code is run in much the same way. In serial mode:

```
java kdh.eco.pipeline.ECOVDXMLPipeline <index-url> <index-dir> <output-dir>
```

where the input parameters are:

1) **<index-url>**: URL of the entity co-occurrence Solr index
2) **<index-dir>**: directory containing the entity co-occurrence index
3) **<output-dir>**: directory where the XML output files are written

In parallel mode, the code is run using the SLRUM srun command:

```
srun -l -w <node-selector-expression> java kdh.eco.pipeline.ECOVDXMLPipeline
<index-url> <index-dir> <output-dir> <node-selector-expression>
```

where the input parameters are:

1) **<index-url>**: URL of the entity co-occurrence Solr index
2) **<index-dir>**: directory containing the entity co-occurrence index
3) **<output-dir>**: directory where the XML output files are written
4) **<node-selector-expression>**: an expression specifying which nodes to use to run the job. This value can either be a hostname (e.g. node1) or an expression of the form node[1-2,4-5,7], where there may be an arbitrary number of comma-separated node numbers or numeric ranges within the square brackets.

4.5 ECO User Interface

The ECO web user interface is divided into three panels. The top half of the interface is dedicated to exploring and navigating entity co-occurrences. The lower-left panel is used to display co-occurrence details for a selected pair of entities, and the lower-right panel is used to display the original document text of a selected document. The three-panel interface is shown in Figure 5.

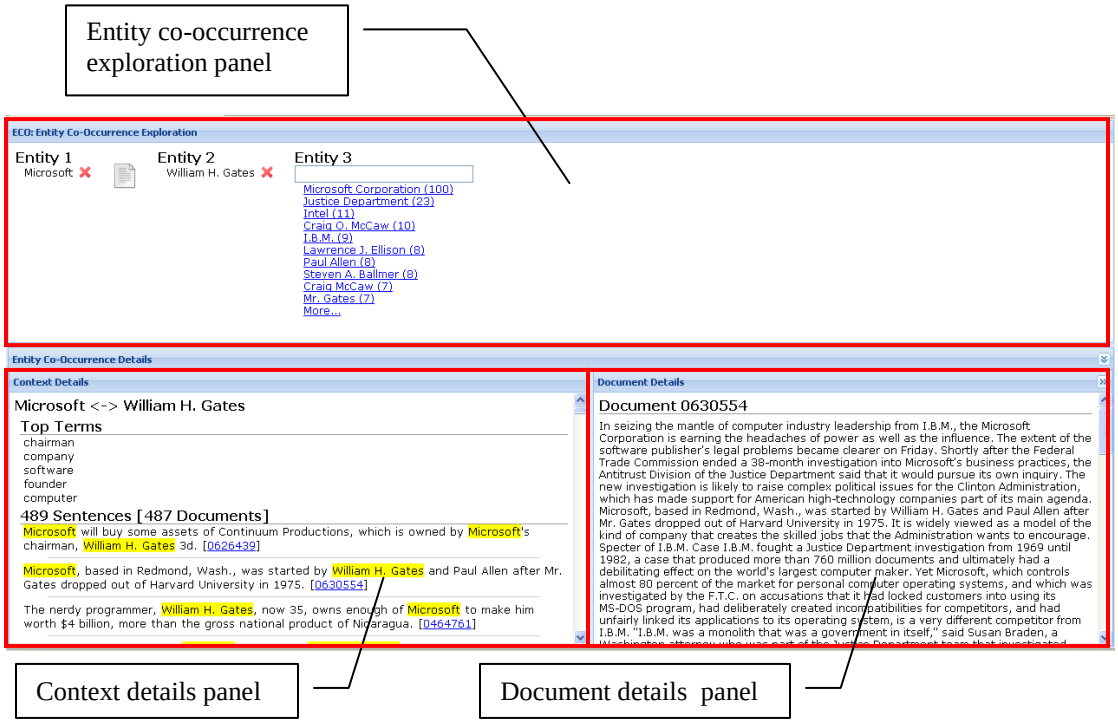

Fig. 5. ECO three-panel user interface

Entity Co-Occurrence Exploration Panel

The entity co-occurrence exploration panel provides a column-oriented faceted navigation view of the co-occurrence data. In the initial state, the panel presents a single facet entitled “Entity 1” beneath which is displayed a list of extracted entities ranked by co-occurrence frequency. Only the top n entities are initially displayed. A close-up view of this panel in its initial state is shown in Figure 6.

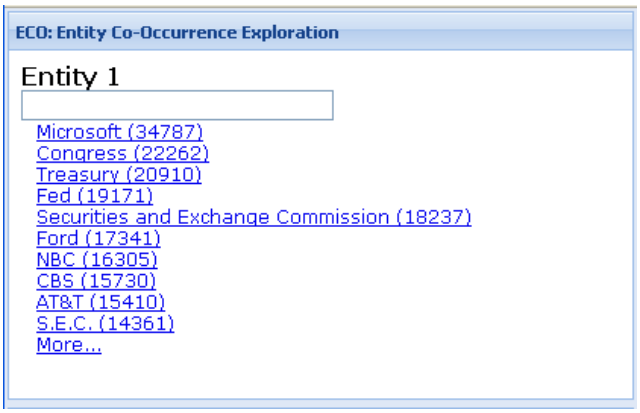

Fig. 6. ECO co-occurrence exploration panel in its initial state. Entities are displayed in descending order of co-occurrence frequency across the corpus.

The user is able to select an entity, expand the list of entities by clicking the "more…" link, or search the entity list using the search box atop the list. When an entity is selected, the selection in the "Entity 1" column becomes fixed and a new "Entity 2" column appears to the right of the first column. The second column displays another list of entities, but the entities displayed are only those participating in a co-occurrence relationship with the entity selected from the "Entity 1" column.

Alongside the selected entity in the "Entity 1" column, an "X" button is displayed. Clicking this button deselects the entity, causes the "Entity 2" column to disappear, and resets the user interface to its initial state. Figure 7 shows how the interface appears after the user selects an entity from the first column.

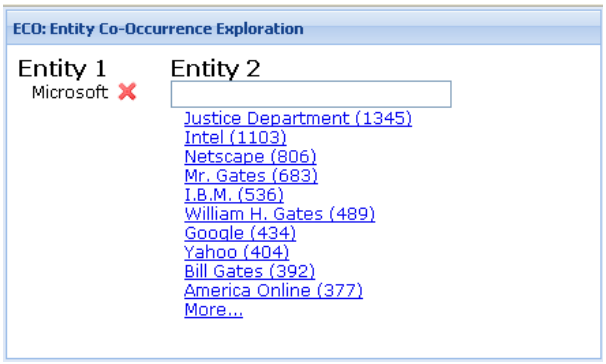

Fig. 7. ECO user interface after user has selected an entity from the "Entity 1" column. Only entities co-occurring with that selected entity are displayed in the new "Entity 2" column.

After the user selects an entity in the second column, the entity values from the first two columns are fixed. A new column entitled "Entity 3" appears to the right of the "Entity 2" column. This column contains all entities co-occurring with the entity selected in the previous column except the entity from the first column.

At this point, the user has selected a pair of entities and may be interested in exploring how those entities are related. A document icon is displayed between the two entities. Clicking this icon displays the co-occurrence details in the lower-left panel of the user interface. The co-occurrence details include the top terms and context fragments from the "virtual document" composed of all contexts (i.e. sentences) from which these two entities were extracted. Figures 8 and 9 depict the ECO user interface in these states.

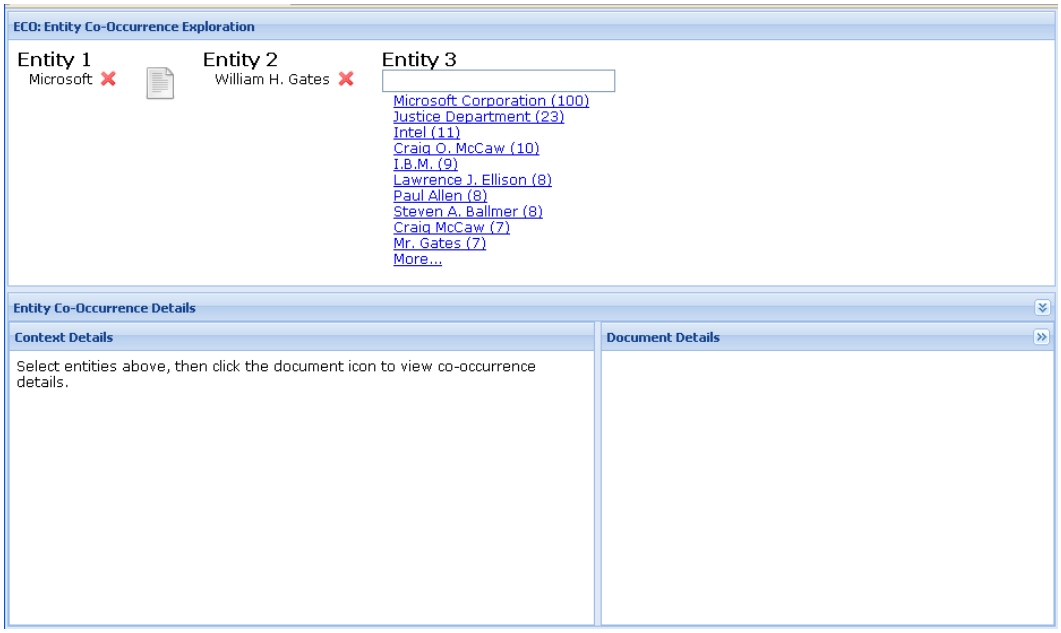

Fig. 8. The ECO user interface after the first two entities have been selected. Clicking the document icon between the two selected entities provides additional information about how the entities are related.

Fig. 9. The ECO user interface after the user has clicked the document icon between the first two selected entities.

The top terms are calculated using term frequency – inverse-document-frequency (TF-IDF) measures calculated from the entity co-occurrence virtual document index. TF-IDF compares the ratio of the term frequency of terms within one document with the inverse document frequency of those terms corpus-wide. If a particular term appears frequently in one document but infrequently throughout the rest of the corpus, that term has a higher importance to that document.

The top terms are also filtered based on some simple criteria. For a term to be considered, it must not be a number or appear in a configurable stop word list. The term must also have at least three characters and must not be a substring of either of the two co-occurring entities.

Beneath the top terms, the list of sentences where the two entities co-occurred is shown. Each sentence is followed by a hyperlinked document identifier, which when clicked displays the full text of the document from which that sentence was extracted in the document details panel. Figure 10 shows the user interface after the user has clicked on a document identifier link.

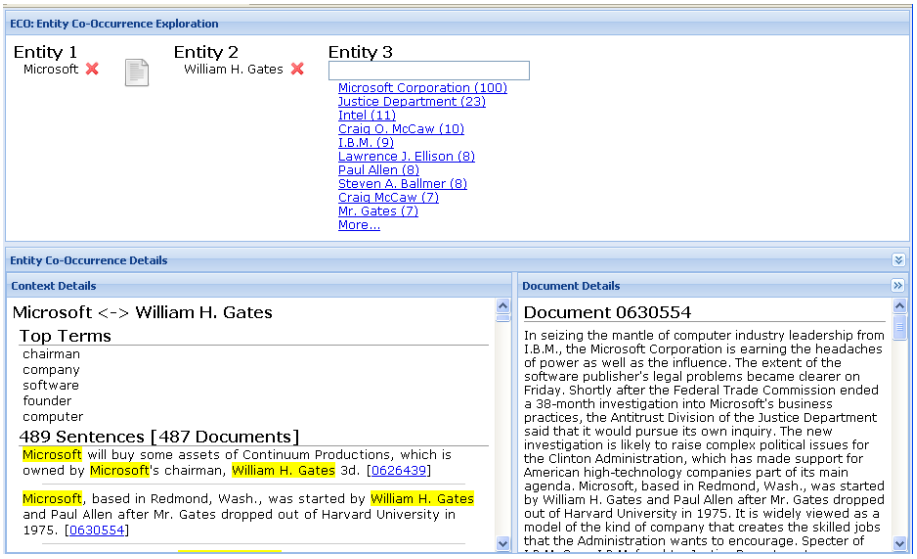

Fig. 10. The ECO user interface after the user has clicked on a document identifier link

The user can continue selecting entities and investigating relationships as the user interface expands. In theory, the number of columns could be unlimited, but extending the degree of connectedness beyond four or five does not prove useful in practice due to small world effects [41].

4.6 Processing Performance Evaluation

To evaluate the performance of the data processing pipeline, the ECO pipeline ingested a subset of the overall dataset consisting of all non-statistical business articles from the year 1987. This set contains 15,273 articles, from which the ECO data processing code extracted 180,591 entity co-occurrences.

The entity co-occurrence XML generation code was run in parallel mode, using successively more nodes of the cluster. All runs were performed with an allocation of 3GB of RAM (`-Xms3g -Xmx3g`) to the Java Virtual Machine per node. The results of these runs are summarized in Figure 11. Performance scales in proportion to the number of nodes utilized, but the scale factor is not linear, most likely due to the fact that as the number of nodes increases, there is increased I/O contention on the NFS storage system. Running the data ingest on 22 nodes of the cluster took 55 seconds, a speedup of 18.9 over the single-node ingest.

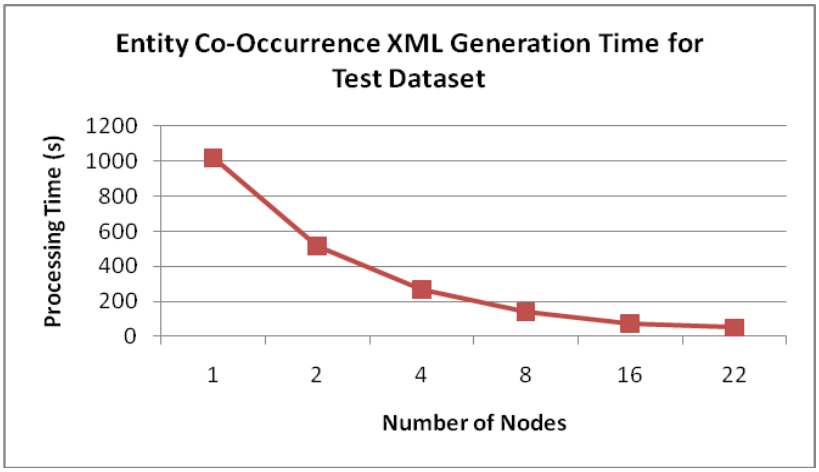

Fig. 11. Entity co-occurrence XML generation time for test dataset

4.7 Co-Occurrence Extraction Performance Evaluation

To evaluate the quality of the co-occurrence extraction, the entire NYT corpus consisting of 227,936 non-statistical business articles published between 1987 and 2007 was loaded into the ECO system. From this corpus, the ECO processing pipeline identified 562,241 unique entities participating in 3,143,187 co-occurrences. As shown in Figure 12, the distribution of entities across the co-occurrences is highly skewed. Over half of the entities participate in only one or two co-occurrences. At the same time, however, a small number of entities are very frequently mentioned and thus appear in a large number of co-occurrences.

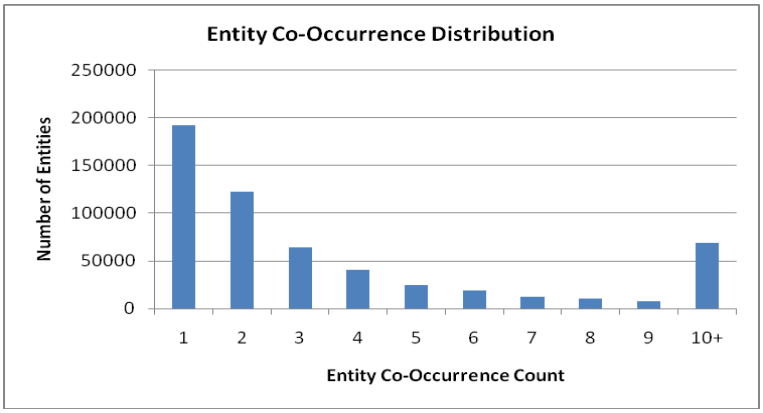

Fig. 12. Distribution of entities by co-occurrence count

The most frequently co-occurring entity is "Microsoft", appearing in 34,787 co-occurrences. The most common entity pair is "Microsoft" and "Justice Department". These two entities were extracted from 1,345 separate sentences.

To evaluate the quality of the extracted relationships, the top terms generated from the ECO virtual document index were examined for several well-known pairs of related entities. The results are summarized in Table 1.

Entity 1	Entity 2	Top Terms
Sergey Brin	Larry Page	founders, google, cofounders, company, schmidt
Monica Lewinsky	Clinton	president, jury, grand, intern, testimony
PeopleSoft	Oracle	bid, offer, takeover, software, hostile
Boeing	Air Force	tankers, tanker, druyun, pentagon, refueling
Jack Welch	General Electric	chief, executive, nbc, chairman, former

Table 1. Well-known entity pairs and the top terms extracted from the ECO virtual document index

One factor that plays a role in co-occurrence extraction is entity disambiguation. Named entity extraction identifies named entities within text, but it does not resolve the extracted entities to real-world entities. For example, the entities "Securities and Exchange Commission" and "S.E.C" both appear near the top of the most frequently co-occurring entities, and they very likely refer to the same real-world organization. Entity

disambiguation is a well-known problem in the field of text analysis, and ECO does not attempt to address this issue.

Another important issue is entity co-reference resolution, which involves associating pronouns with the proper nouns they reference. For example, in the two sentences "Microsoft is a massive software corporation. It develops the Windows operating system," the equivalence between "Microsoft" in the first sentence and "It" in the second sentence, can be a non-trivial association to make, depending on the complexity and formulation of the sentences.

5.0 Hardware and Software Infrastructure

An initial prototype of the system was developed on a Microsoft Windows PC. A Linux cluster at LLNL, along with the SLURM utility [42] for parallel processing, was used for large-scale text processing, entity extraction, and indexing.

The Linux cluster used in this project is a 24-node system with a shared NFS disk. Each node has a 4-core Intel Xeon CPU operating at 3.4 GHz and 4 GB of RAM. This cluster was utilized because it was readily available and offers good performance for the corpus sizes used in this project.

The data ingest software is written in Java using the Sun Microsystems Java Runtime version 1.6. The user interface was written in Java, using Java Server Pages (JSP), servlets, and JavaScript. The ExtJS JavaScript library [43] was utilized for layout management and AJAX communication. The user interface is packaged as a Java Web Application Archive (.war) file, which runs using the Apache Tomcat server [44]. The

resulting software is platform and browser-independent. The software architecture is detailed in Appendix A.

6.0 Conclusions and Future Work

The ECO framework is an end-to-end system for extracting and analyzing entity co-occurrences from a large text corpus. The system integrates many freely available, open-source software components and offers scalability for ingesting large datasets. By utilizing faceted navigation concepts, the ECO user interface presents a novel method for navigating a co-occurrence graph and for identifying and understanding co-occurrence relationships in a large document set.

A few notable challenges encountered during development of the ECO system included parallelizing the data processing code and refining the nature of the top terms describing a co-occurrence relationship. Initial versions of the ingest code attempted to use multi-threading to parallelize the data processing pipeline so it could utilize multiple processor cores on a single compute node. However, it was discovered that the Stanford Named Entity Recognizer is not thread-safe, as it yielded erroneous results when run in a multi-threaded manner.

During development of the TF-IDF top terms component, the top terms occasionally included very common words, such as pronouns or numbers. Even though these words were highly relevant to the relationship, they were not necessarily descriptive or particularly useful. To resolve this issue, a stop word list was used to discard commonly used terms and numbers, and the terms were required to be two or more characters in length.

Many opportunities exist for extending the application with new functionality. Adding article dates to the index, for example, would allow the user to restrict the co-occurrences displayed to only those within a certain time range. The co-occurrence and co-occurrence virtual document indexes have good potential to contribute to entity disambiguation problems, as entities with different spellings but referencing the same real-world entity most likely co-occur with similar entities.

Another interesting research direction would be to examine frequently co-occurring groups of entities. This work focuses on pairs of entities, but indexing information about entity groups could help to locate and describe multi-way associations. The co-occurrence graph could also be examined with graph algorithms and social network analysis to discover entity clusters, cliques, or other meaningful structure.

7.0 Reference List

[1] T. Hasegawa, S. Sekine, and R. Grishman, "Discovering relations among named entities from large corpora," *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, Morristown, NJ, USA: Association for Computational Linguistics, 2004, p. 415.

[2] C. Manning and H. Schuetze, *Foundations of Statistical Natural Language Processing*, MIT Press, .

[3] C. Manning, R. Prabhakar, and S. Hinrich, *Introduction to Information Retrieval*, Cambridge University Press, 2008.

[4] "ACM SIGIR Special Interest Group on Information Retrieval Home Page" Available: http://www.sigir.org/.

[5] Y. Jin, Y. Matsuo, and M. Ishizuka, "Extracting social networks among various entities on the web," *ESWC '07: Proceedings of the 4th European conference on The Semantic Web*, Berlin, Heidelberg: Springer-Verlag, 2007, pp. 251–266.

[6] J. Xu and H. Chen, "Fighting organized crimes: using shortest-path algorithms to identify associations in criminal networks," *Decision Support Systems*, vol. 38, 2004, pp. 473-487.

[7] T. Lewis, *Network Science: Theory and Applications*, Wiley, 2009.

[8] "ACM SIGCHI Special Interest Group on Computer Human Interaction Home Page" Available: http://www.sigchi.org/.

[9] A. Dix, J. Finlay, G. Abowd, and R. Beale, *Human-Computer Interaction*, Prentice

Hall, 2003.
[10] “Wikipedia” Available: http://www.wikipedia.org/.
[11] “DBpedia” Available: http://dbpedia.org/About.
[12] “RDF - Semantic Web Standards” Available: http://www.w3.org/RDF/.
[13] “NIST Message Understanding Conference (MUC) Website” Available: http://www.itl.nist.gov/iad/894.02/related_projects/muc/.
[14] “NIST Automatic Content Extraction (ACE) Website” Available: http://www.itl.nist.gov/iad/mig/tests/ace/.
[15] “NIST Text Analysis Conference (TAC) Website” Available: http://www.nist.gov/tac/.
[16] “NIST Text REtrieval Conference (TREC) Website” Available: http://trec.nist.gov/.
[17] A. Kao and S. Poteet, *Natural Language Processing and Text Mining*, Springer, 2006.
[18] D. Nadeau and S. Sekine, “A survey of named entity recognition and classification,” *Lingvisticae Investigationes*, vol. 30, 2007, pp. 3-26.
[19] “The Official City of Raleigh, NC Portal” Available: http://raleighnc.gov/.
[20] C. Van Rijsbergen, “A theoretical basis for the use of co-occurrence data in information retrieval,” *Journal of Documentation*, vol. 33, 1977, pp. 106-119.
[21] X. Li and B. Liu, “Mining community structure of named entities from free text,” *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, New York, NY, USA: ACM, 2005, pp. 275–276.
[22] M. Zhang, J. Su, D. Wang, G. Zhou, and C.L. Tan, “Discovering relations between named entities from a large raw corpus using tree similarity-based clustering,” *IJCNLP*, Jeju Island, Korea: Springer-Verlag Berlin Heidelberg New York, 2005, pp. 378-389.
[23] Y. Matsuo, J. Mori, M. Hamasaki, K. Ishida, T. Nishimura, H. Takeda, K. Hasida, and M. Ishizuka, “POLYPHONET: an advanced social network extraction system from the web,” *WWW '06: Proceedings of the 15th international conference on World Wide Web*, New York, NY, USA: ACM, 2006, pp. 397–406.
[24] B. Magnini, M. Negri, R. Prevete, and H. Tanev, “Mining knowledge from repeated co-occurrences: DIOGENE at TREC-2002.,” Gaithersburg, MD: 2003, pp. 349-357.
[25] B. Popov, I. Kitchukov, K. Angelov, and A. Kiryakov, “Co-occurrence and ranking of entities,” May. 2006.
[26] J.G. Conrad and M.H. Utt, “A system for discovering relationships by feature extraction from text databases,” *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA: Springer-Verlag New York, Inc., 1994, pp. 260–270.
[27] H. Raghavan, J. Allan, and A. Mccallum, “An exploration of entity models, collective classification and relation description,” *In Proceedings of KDD Workshop on Link Analysis and Group Detection*, 2004, pp. 33–3.
[28] D. Petkova and W.B. Croft, “Proximity-based document representation for named entity retrieval,” *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, New York, NY, USA: ACM, 2007, pp. 731–740.

[29] J. Schroeder, J. Xu, and H. Chen, "CrimeLink explorer: using domain knowledge to facilitate automated crime association analysis," *ISI'03: Proceedings of the 1st NSF/NIJ conference on Intelligence and security informatics*, Tucson, AZ, USA: 2003, pp. 168-180.
[30] S. Ranganathan, *A descriptive account of Colon Classification*, Bangalore, India: Sarada Ranganathan Endowment for Library Science, 1965.
[31] S.A. Pollitt, "The key role of classification and indexing in view-based searching," Copenhagen, Denmark: 1997.
[32] M. Hearst, A. Elliott, J. English, R. Sinha, K. Swearingen, and K. Yee, "Finding the flow in web site search," *Commun. ACM*, vol. 45, 2002, pp. 42–49.
[33] M.A. Hearst, "Clustering versus faceted categories for information exploration," *Commun. ACM*, vol. 49, 2006, pp. 59–61.
[34] M. Hearst, "UIs for faceted navigation: recent advances and remaining open problems," *HCIR 2008*, Redmond, WA: 2008.
[35] "Apache Solr" Available: http://lucene.apache.org/solr/.
[36] E. Sandhaus, "The New York Times Annotated Corpus," 2008.
[37] J.R. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by Gibbs sampling," *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, Morristown, NJ, USA: Association for Computational Linguistics, 2005, pp. 363–370.
[38] D. Klein and C. Manning, "Accurate unlexicalized parsing," *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, 2003, pp. 423-430.
[39] D. Klein and C. Manning, "Fast exact inference with a factored model for natural language parsing," *Advances in Neural Information Processing Systems 15*, Cambridge, MA: MIT Press, 2002, pp. 3-10.
[40] "XStream" Available: http://xstream.codehaus.org/.
[41] J. Travers and S. Milgram, "An experimental study of the small world problem," *Sociometry*, vol. 32, 1969, pp. 425-443.
[42] "Simple Linux Utility for Resource Management" Available: https://computing.llnl.gov/linux/slurm/.
[43] "Sencha - Ext JS - Client-side JavaScript Framework" Available: http://www.sencha.com/products/js/.
[44] "Apache Tomcat" Available: http://tomcat.apache.org/.

Appendix A: Software Design Diagrams

A.1. Architectural Layers

The ECO software consists of the following architectural layers, illustrated in Figure 13.

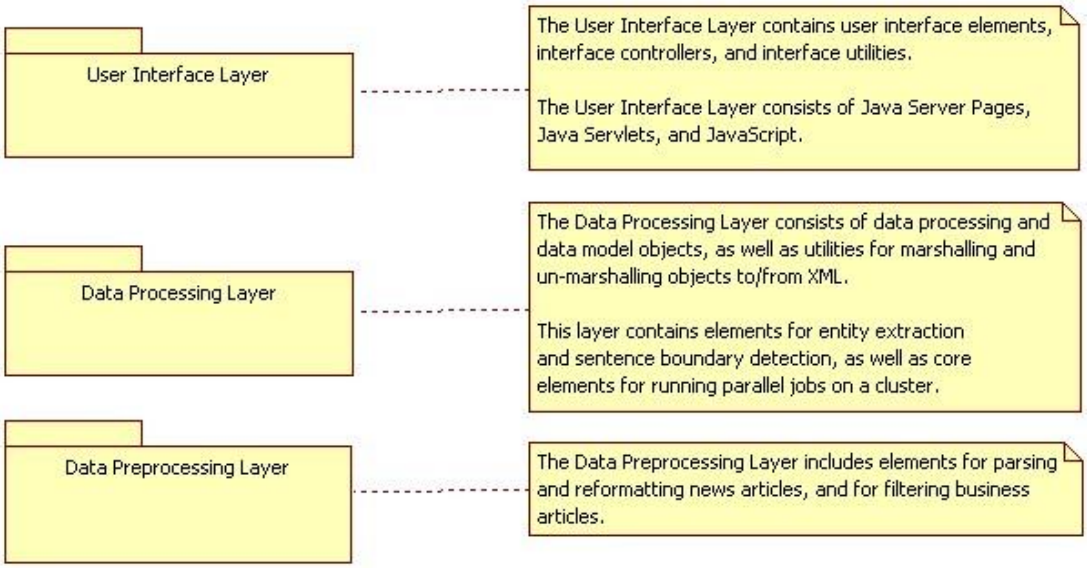

Fig. 13 ECO architectural layers

A.2. Subsystems

Each layer is further divided into logical cohesive subsystems, as illustrated in Figure 14.

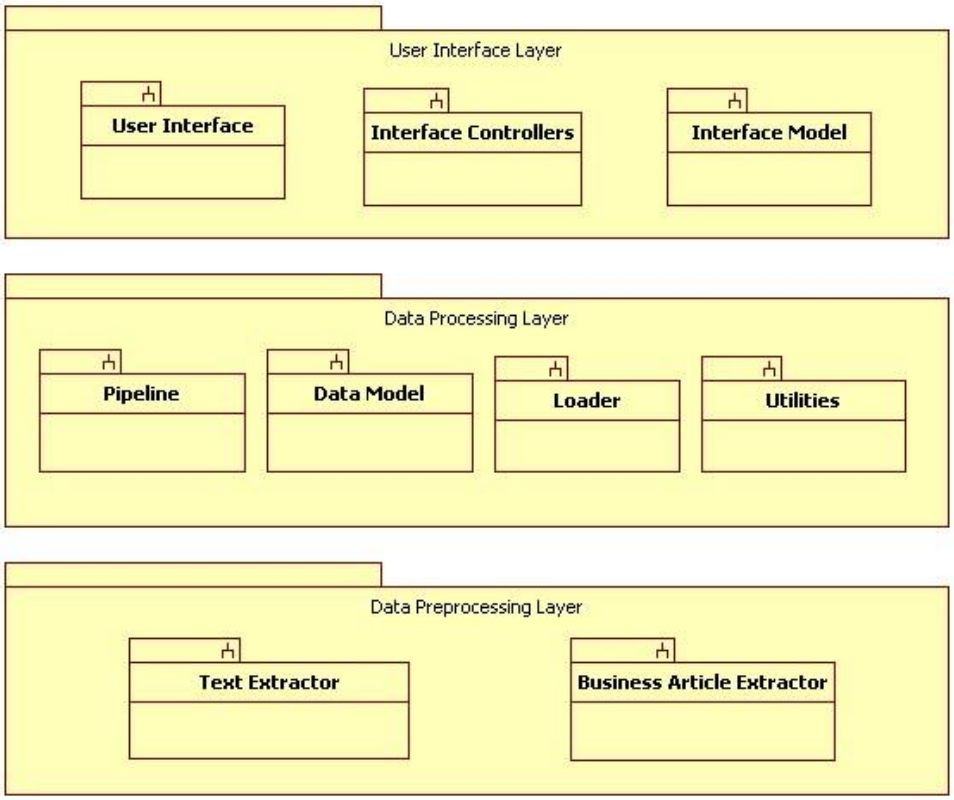

Fig. 14 ECO subsystems

A.3. Data Preprocessing Layer

The Data Preprocessing Layer includes elements for parsing and reformatting news articles and for filtering business articles. Figures 14 and 15 depict the Text Extractor and Business Article Extractor subsystems.

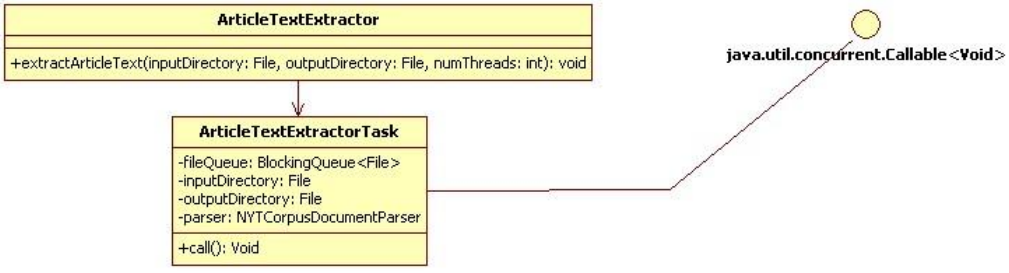

Fig. 15 Text Extractor subsystem

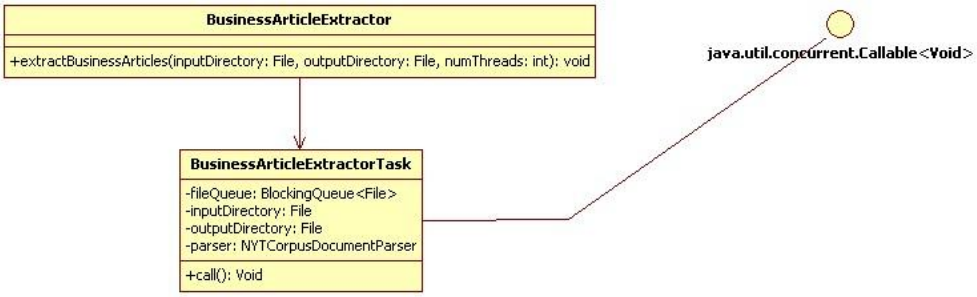

Fig. 16 Business Article Extractor subsystem

A.4. Data Processing Layer

The Data Processing Layer consists of data processing and data model objects, as well as utilities for marshalling and un-marshalling objects to/from XML. This layer contains elements for entity extraction, sentence boundary detection, and the core framework for running parallel jobs on a cluster. These subsystems are illustrated in Figures 16 – 19.

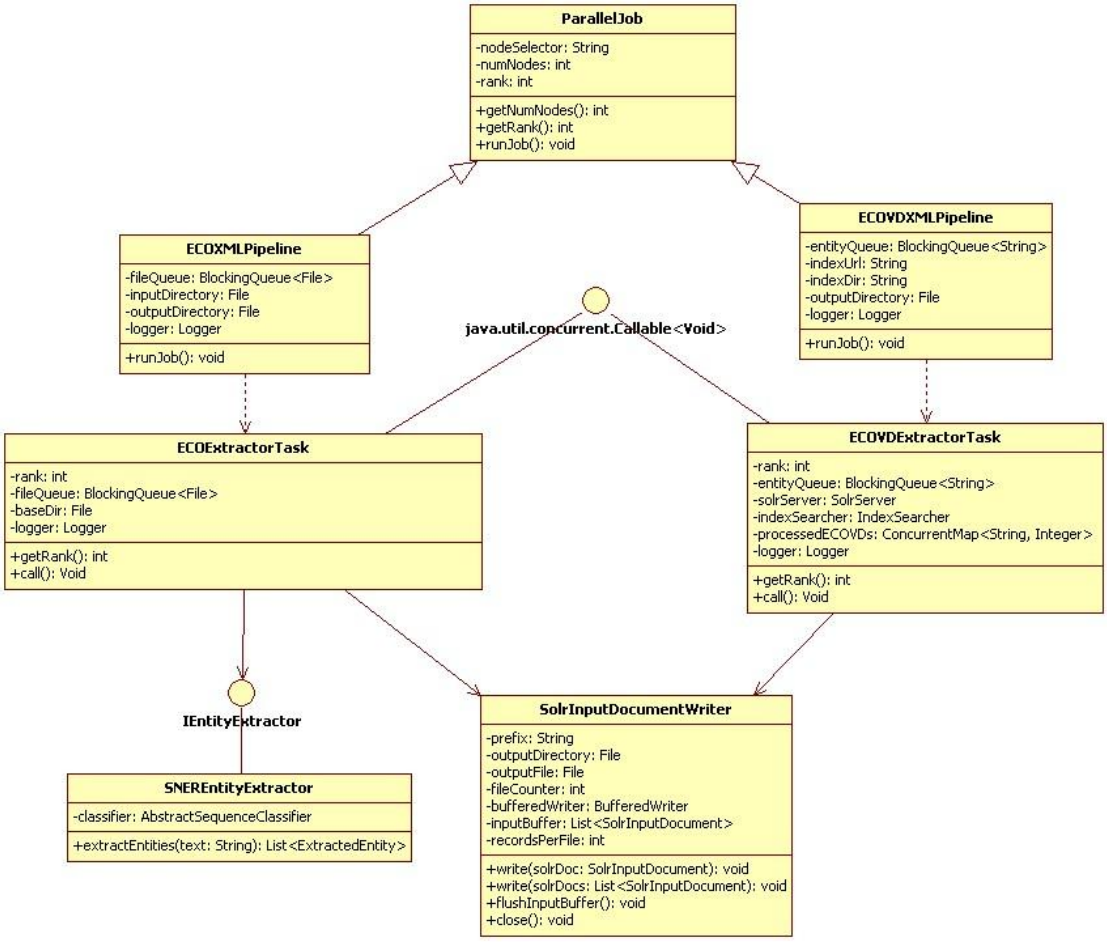

Fig. 17 ECO Pipeline subsystem

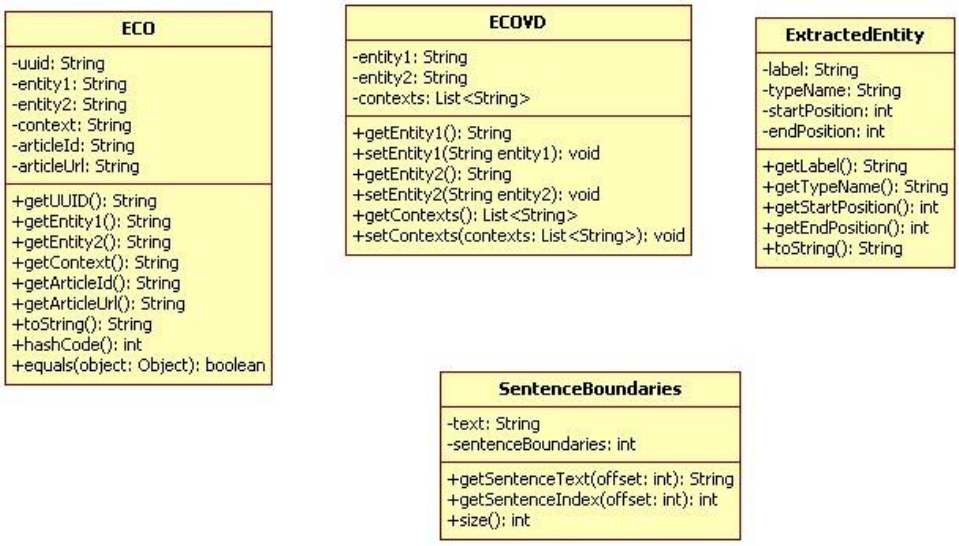

Fig. 18 ECO Data Model subsystem

Fig. 19 ECO Loader subsystem

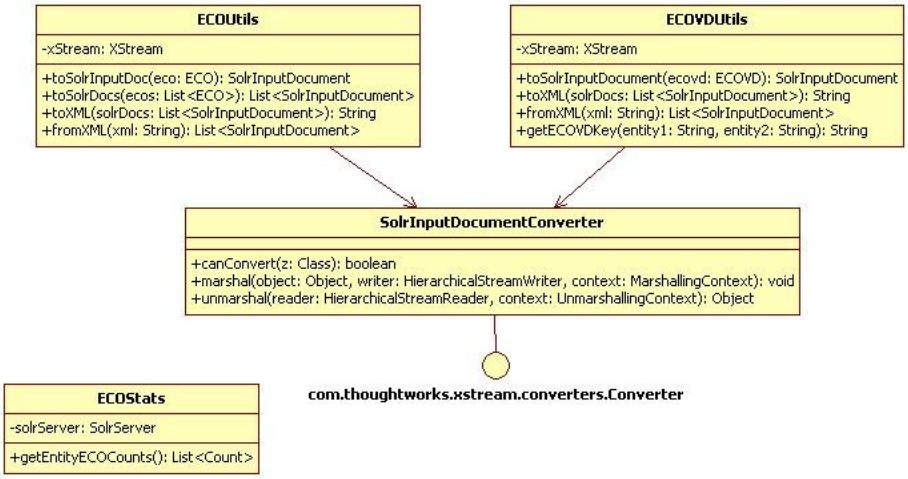

Fig. 20 ECO Utilities subsystem

A.5. User Interface Layer

The User Interface Layer contains user interface elements, interface controllers, and interface model elements. The User Interface Layer is implemented using Java Server Pages, Java Servlets, and JavaScript. The User Interface subsystems are shown in Figures 20 – 22.

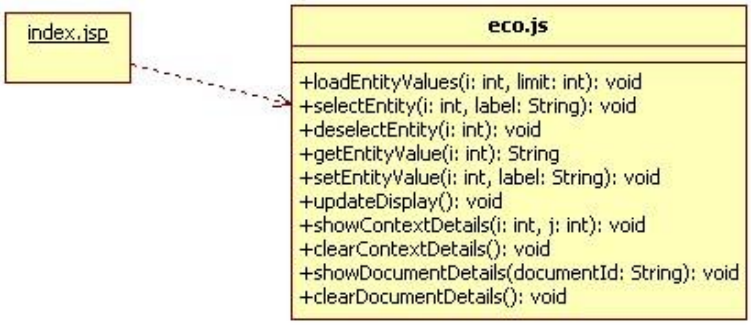

Fig. 21 User Interface subsystem

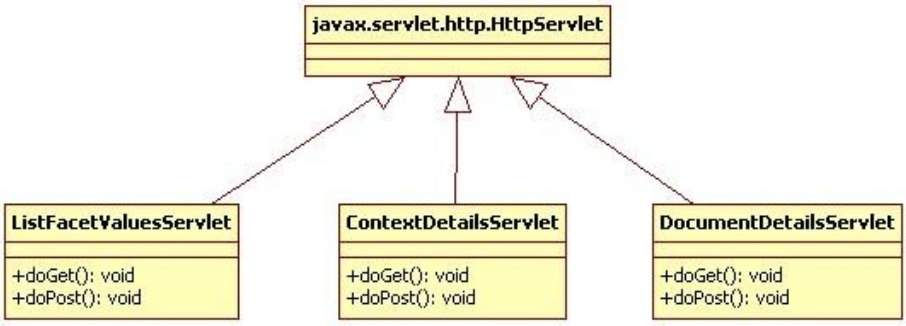

Fig. 22 Interface Controllers subsystem

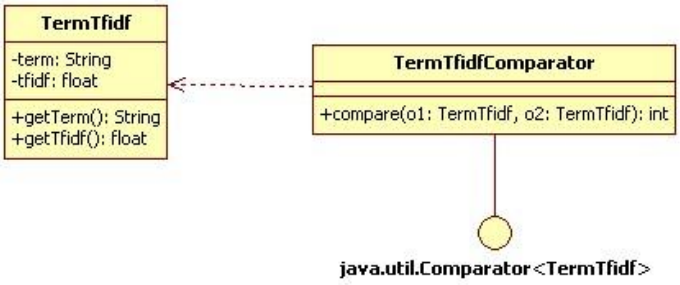

Fig. 23 Interface Model subsystem