Seed-based Generation of Personalized Bio-Ontologies for Information Extraction

Cui Tao⋆ and David W. Embley⋆

Department of Computer Science,
Brigham Young University, Provo, Utah 84602, U.S.A.

Abstract. Biologists usually focus on only a small, individualized, sub-domain of the huge domain of biology. With respect to their sub-domain, they often need data collected from various different web resources. In this research, we provide a tool with which biologists can generate a sub-domain-size, user-specific ontology that can extract data from web resources. The central idea is to let a user provide a seed, which consists of a single data instance embedded within the concepts of interest. Given a seed, the system can generate an extraction ontology, match information with the user's view based on the seed, and collect information from online repositories. Our initial experimentations indicate that our prototype system can successfully match source data with an ontology seed and gather information from different sources with respect to user-specific, personalized views.

1 Introduction

To do activities such as performing background research for a field of study, gaining insights into relationships and interactions among different research discoveries, or building up research strategies inspired by other's hypotheses, biologists often need to search several online databases and gather information of interest. Biologists usually have to traverse different web sources and collect the data of interest manually. This task is a tedious and time-consuming.

It would be beneficial if we could generate a data-extraction ontology specifically for each individual user that would automatically collect the information of interest. But generating an ontology, especially an ontological description for an information repository, is non-trivial; it not only requires domain expertise, but also requires knowledge of specific ontology language. Data heterogeneity and different user objectives makes the task even more daunting.

To illustrate the difficulties biologists encounter in gathering information from a variety of sources and also to illustrate the challenges involved in building an extraction ontology to automatically collect data, consider some examples. For chromosome location of a gene, some users might only care about the chromosome on which this gene is located. Other users might care about a more detailed location like the start and end base pairs. Sources, not knowing user

⋆ Supported in part by the National Science Foundation under Grant #0414644.

objectives, provide their own view of the data. One source could describe a chromosome location of a gene as one concept. Others could describe the location in terms of multiple concepts such as chromosome number, start location, end location, orientation, and size. The size is actually an implicative value which is equal to end minus start, therefore some designer could also choose to omit this concept. As another example, consider the use of different units for the same concept. For example, one site could use kilo-dalton as an unit for molecular weight, one could use dalton, and another could provide both. Still other problems arise because different sites might provide information directly or indirectly. A protein database, for example, could use Gene Ontology (GO) terms to describe molecular functions. In order to obtain information for the description of the definitions of the terms, a user usually needs to go to the GO database.

In this paper, we introduce a system that can automatically build a data-extraction ontology given a user's seed. We call this system SIH (Seed-based Information Harvester, pronounced "sigh"), because once built, we can use the ontology to harvest information from web repositories. A seed consists of a single sample data instance embedded within the concepts of interest. Based on the ontology seed, SIH matches information with the user's view, builds a personalized ontology, and collects information of interest from online repositories. The advantages of this system are (1) it does not require knowledge of conceptual modeling or ontology languages to build ontologies, and (2) it can automatically harvest information from multiple sites and present the information according to a user-specified view.

We present the details of SIH and our contribution to user-specified ontology generation and subsequent information harvesting as follows. Section 2 positions our work within recent work on ontology creation. Section 3 introduces OSM ontologies, the ontology framework we use in this research. Section 4 describes the interface used to create a seed ontology for SIH. Section 5 explains how SIH maps site labels to seed ontology labels and how the generated extraction ontology collects information from various sources. Section 6 reports the results of some initial experiments we conducted with our SIH implementation, makes concluding remarks, and mentions some future directions we wish to pursue in this research.

2 Related Work

In recent years, many researches have tried to facilitate ontology generation. Manual editing tools such as Protege [6] and OntoWeb [8] have been developed to help users create and edit ontologies. It is not trivial, however, to learn ontology modeling languages and complex tools in order to manually create ontological description for information repositories.

Because of the difficulties involved in manual creation, researchers have developed semi-automatic ontology generation tools. Most efforts so far have been devoted to automatic generation of ontologies from text files. Tools such as OntoLT [1], Text2Onto [2], OntoLearn [5], and KASO [11] use machine learning methods to generate an ontology from arbitrary text files. These tools usually require a large training corpus and use various natural language processing algorithms to derive features to learn ontologies. The results, however, are not very satisfactory [7]. Tools such as TANGO [10] and the one developed by Pivk [7] use structured information (HTML tables) as a source for learning ontologies. Structured information makes it easier to interpret new items and relations. These approaches, however, derive concepts and relationships among concepts from source data, not from users. SIH, on the other hand, allows users to provide their own views and generate user-specified ontologies.

Potentially, it should be possible to derive biologist-specific view ontologies from large biological ontologies such as the Gene Ontology, the NCI Thesaurus, and the SNOMED Ontology. Our own experience in this direction [3], however, has not been very successful, mostly because the existing large biological ontologies are usually more like hierarchial vocabulary lists than the conceptual-model-based ontologies we need for information extraction.

3 OSM Ontologies

We use OSM [4] as the semantic data model for an extraction ontology. Figure 1 shows a graphical view of a sample ontology. The structural components of OSM include object sets, relationship sets, and constraints over these object and relationship sets. An object set in an OSM ontology represents a set of objects which may either be lexical or non-lexical. A dashed box represents a lexical object set and a solid box represents a non-lexical object set. A lexical object set contains concrete values. For example, "T-complex protein 1 subunit theta" is a possible value of the *Name* object set in Figure 1. A non-lexical object describes an abstract concept, such as *Protein* in Figure 1. Lines among object sets represent the relationship sets among them. A small circle at one end of a line indicates optional. For example, a *Protein* can have zero or more *GO Function Definition*s. An arrow indicates functional from domain to range. For example, a *Protein* can only have at most one *Molecular Weight*; the relationship set is therefore functional from *Protein* to *Molecular Weight.* OSM also supports n-ary relationships with multiple lines connecting the object sets involved.

We have found OSM to be more expressive than other standard ontology representations, such as RDF and OWL, which, for instance, only supports binary relationships [10]. In addition, and more important, an OSM ontology can support data extraction from source documents [4].

4 Seed Ontology Creation

In this section, we explains how SIH generates an ontology based on a user's seed. We provide our users with a graphical user interface (GUI) where they can create a seed easily. We adapt the user interface proposed by Zhou [12]. Through this GUI, a user can generate an ontology seed by creating a form and then provides the seed values by filling out the form. The form tells SIH

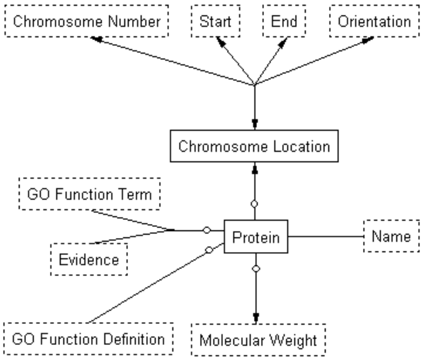


Fig. 1. The Graphical View of a Sample Ontology

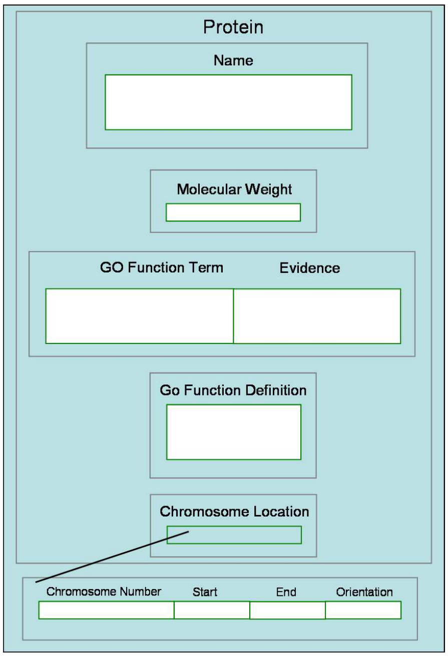


Fig. 2. A Sample Form for Generating the Ontology in Figure 1

how to generate the ontology structure, and the seed values tell SIH something about how to collect information from different sources. We discuss information collection in Section 5.

The GUI provides users with an intuitive method for defining different kind of form features. There are four basic form fields from which users can choose: *single-label/single-entry* fields, *single-label/multiple-entry* fields, *multiple-label/-single-entry* fields, and *multiple-label/multiple-entry* fields. Users can also recursively nest a form inside any basic form field. Nested forms allow users to describe their interests in more structured and meaningful ways.

For each new ontology to be built, a user creates a form and gives the form a meaningful title. Based on this title, SIH generates a new ontology and a non-lexical object set with this title as the name. Every label represents an object set in the corresponding ontology; the label is the name for the object set. If the label is for a field containing a nested form, its object is non-lexical; otherwise its object set is lexical. SIH generates relationship sets among the object sets as follows. Between the form-title object set and each single-label field, it generates a binary relationship set; between the form-title object set and each multiple-label field, it generates an n-ary relationship set; between each field and a single-label object set nested in side of it, SIH generates a binary relationship set; between each field and a multiple-label object set nested inside of it, SIH generates an n-ary relationship set. Cardinalities for relationship sets depend on whether the form field is single-entry or multiple-entry, which respectively indicates that the relationship set is functional or non-functional from form title or nested form title to field or fields. In the reverse direction, the cardinalities are non-functional except when there is exactly one single-entry field. Thus, for example, we have a one-to-one relationship set between *Chromosome Location* and the quadruple *Chromosome Location*, *Start*, *End*, and *Orientation*.

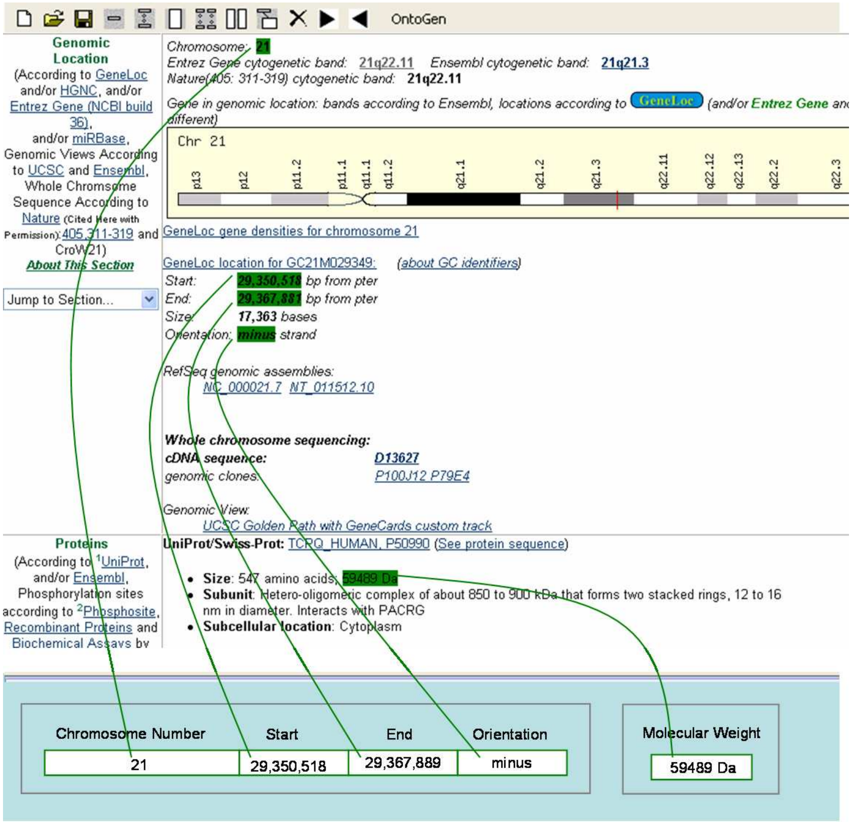
Figure 2 shows an example of form generation. Suppose we are interested in basic information about human proteins (their names, locations, functions, and sizes). In our example, we choose "Protein" as the base-form title. We know each protein can have one or more names, so we choose to add a *single-label/multiple-entry* field to the form and label it *Name.* Since we know there is only one molecular weight for one protein, we choose to use a *single-label/single-entry* form field and label it *Molecular Weight.* We are also interested in the locations of proteins. We know that each protein can have only one location, so generate a *single-label/single-entry* form field and label it *Chromosome Location.* We also know that a chromosome location is composed of four parts: chromosome number, start location, end location, and orientation. In this situation, we choose to create a nested form field. A nested form field is defined in separate panels in the same way as users define basic form fields. Here we choose to use a *multiple-label/single-entry* field as Figure 2 shows. We use the GO (Gene Ontology) to describe a protein function. Each protein has a set of GO function terms to describe their functions. Each GO function term also has an evidence designator associated with it. We thus create a *multiple-label/multiple-entry* field as Figure 2 shows. For each protein, we also want to include the GO function definitions and use a *single-label/multiple-entry* to define it. Overall, SIH generates the ontology in Figure 1.[1]

We complete the creation of a seed by filling in a created form. SIH provides users with a GUI where they can copy values from source pages and paste them into generated forms. Users can browse the online databases with which they are familiar or from which they want to collect information and copy and paste values for one instance into the form. Figure 3 shows an example of copying values for chromosome location and molecular weight from a source page to the corresponding form fields in the form in Figure 2. The highlighted values in the document are the copied values.

5 Data Collection

We explain in this section how we generate data-extraction specifications from an ontology seed. SIH collects information from source repositories that present their information in structured/semi-structured ways. SIH first interprets source pages from these online resources. It then maps the labels in a generated user-specific ontology to the labels in the source pages. Once the mappings are defined, SIH can collect source data for the user.

[1] The optional constraints in Figure 1 are not defined by the user's form. Instead, they come from observing source data as explained in Section 5. We may also adjust reverse cardinalities according to our observations of source data.



Genomic Location
(According to GeneLoc and/or HGNC, and/or Entrez Gene (NCBI build 36), and/or miRBase, Genomic Views According to UCSC and Ensembl, Whole Chromsome Sequence According to Nature (Cited Here with Permission):405,311-319 and CroW21)
About This Section
Jump to Section...

Chromosome: 21
Entrez Gene cytogenetic band: 21q22.11 *Ensembl cytogenetic band:* **21q21.3**
Nature(405: 311-319) cytogenetic band: **21q22.11**
Gene in genomic location: bands according to Ensembl, locations according to GeneLoc *(and/or* ***Entrez Gene*** *an*
different)

GeneLoc gene densities for chromosome 21

GeneLoc location for GC21M029349: *(about GC identifiers)*
Start: 29,350,518 *bp from pter*
End: 29,367,881 *bp from pter*
Size: ***17,363*** *bases*
Orientation: minus *strand*

RefSeq genomic assemblies:
NC_000021.7 NT_011512.10

Whole chromosome sequencing:
cDNA sequence: ***D13627***
genomic clones: *P100J12 P79E4*

Genomic View:
UCSC Golden Path with GeneCards custom track

Proteins
(According to [1]UniProt, and/or Ensembl, Phosphorylation sites according to [2]Phosphosite, Recombinant Proteins and Biochemical Assays by

UniProt/Swiss-Prot: TCPQ_HUMAN, P50990 (See protein sequence)

- **Size**: 547 amino acids; 59489 Da
- **Subunit**: Hetero-oligomeric complex of about 850 to 900 kDa that forms two stacked rings, 12 to 16 nm in diameter. Interacts with PACRG
- **Subcellular location**: Cytoplasm

Chromosome Number	Start	End	Orientation
21	29,350,518	29,367,889	minus

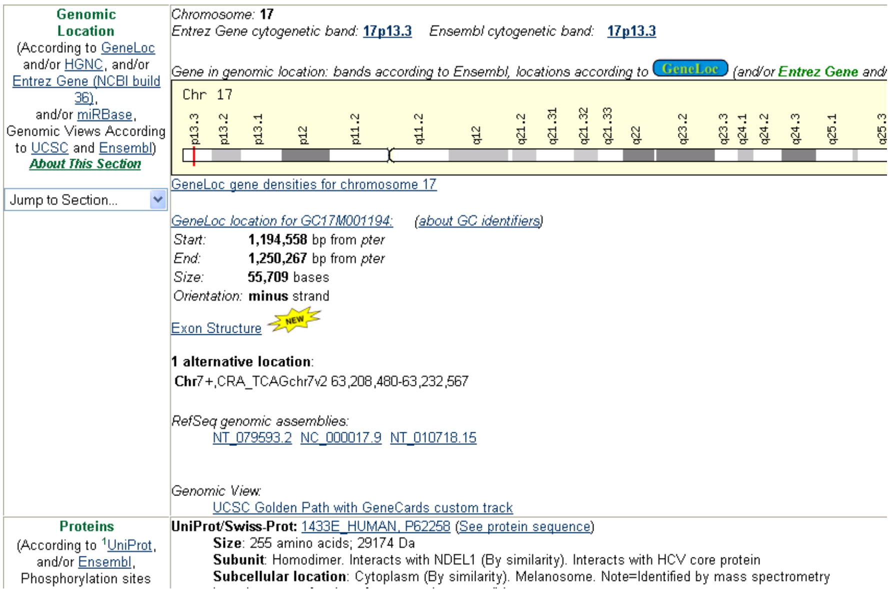
Molecular Weight
59489 Da

Fig. 3. An Sample Ontology Seed with a Source Data Page (Partial)

5.1 Source Page Interpretation

Many online repositories present their data in dynamically generated pre-defined templates in response to submitted queries. Pages from this kind of repository usually have the same or similar structure. We call pages that are from the same web site and have similar structures *sibling pages* and the corresponding tables in sibling pages *sibling tables*. Figure 4 shows a sibling page for the page with *Molecular Weight* seed values in Figure 3.

We have developed a system called *TISP* (*Table Interpretation with Sibling Pages*) [9], which can automatically interpret the structure of sibling pages and find the association between labels and values. TISP first decomposes source sibling pages, unnestes all the HTML tables, and finds sibling tables. TISP then compares a pair of sibling tables to identify nonvarying components (category labels) and varying components (data values). After it identifies labels and values, TISP finds the structure pattern of the table. It checks whether a table matches any pre-defined pattern template by testing each template until it finds a match. With a structure pattern for a specific table, TISP can interpret the



Genomic Location
(According to GeneLoc and/or HGNC, and/or Entrez Gene (NCBI build 36), and/or miRBase, Genomic Views According to UCSC and Ensembl)
About This Section

Jump to Section...

Chromosome: **17**
Entrez Gene cytogenetic band: **17p13.3** *Ensembl cytogenetic band:* **17p13.3**

Gene in genomic location: bands according to Ensembl, locations according to GeneLoc *(and/or* ***Entrez Gene*** *and/*

GeneLoc gene densities for chromosome 17

GeneLoc location for GC17M001194: (*about GC identifiers*)

Start:	**1,194,558** bp from *pter*
End:	**1,250,267** bp from *pter*
Size:	**55,709** bases
Orientation:	**minus** strand

Exon Structure NEW

1 alternative location:
Chr7+,CRA_TCAGchr7v2 63,208,480-63,232,567

RefSeq genomic assemblies:
NT_079593.2 NC_000017.9 NT_010718.15

Genomic View:
UCSC Golden Path with GeneCards custom track

Proteins
(According to [1]UniProt, and/or Ensembl, Phosphorylation sites

UniProt/Swiss-Prot: 1433E_HUMAN, P62258 (See protein sequence)
Size: 255 amino acids; 29174 Da
Subunit: Homodimer. Interacts with NDEL1 (By similarity). Interacts with HCV core protein
Subcellular location: Cytoplasm (By similarity). Melanosome. Note=Identified by mass spectrometry

Fig. 4. An Example of a Sibling Page for the Source Page in Figure 3

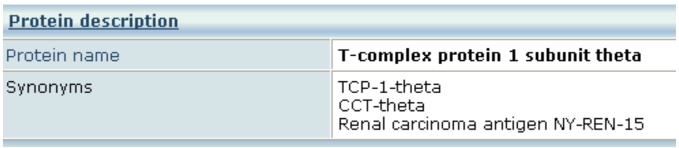
table and all its sibling tables. We assume that values under the same label from different sibling tables are for the same or same set of concepts. Using TISP, we can collect all the values under the same label from a source repository.

5.2 Source-Target Mapping and Data Collection

The next step for SIH is to map source labels to concepts in the generated ontology. Seed values provide the main means of determining these mappings. SIH knows both the label for a seed value and the value's source-page label, and can therefore link the two labels. The basic idea is that source values for source-page labels are values that can fill in the form field for the form labels.

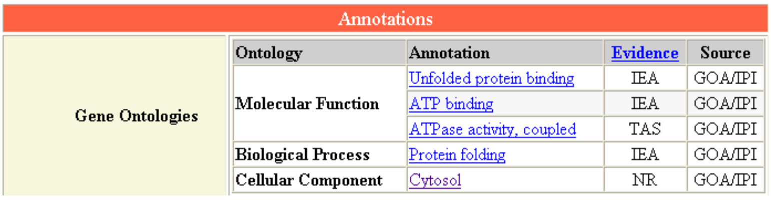
Unfortunately, mapping generation is not quite so simple because the labels may not have exactly the same meaning. Size in Figure 3 and 4, for example, does not have the same meaning as *Molecular Weight* in Figure 1 and 2. Size values include both the number of amino acids and the molecular weight. Thus, SIH must "split" size values and pick up only the part giving the molecular weight. In general, we must handle five different cases we encountered during the mapping process: direct mappings, unions, selections, splits, and merges.

Direct Mapping. When a seed value matches a source value exactly, SIH infers a direct mapping. For example, the highlighted value "21" under the source label *Chromosome* in Figure 3 matches the seed value "21" in the form under seed label *Chromosome Number*; thus SIH infers a direct mapping from *Chromosome* to *Chromosome Number*. For information harvesting, SIH just collects all the information under the same label from all the sibling pages.



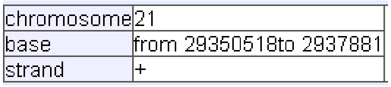
Protein description	
Protein name	**T-complex protein 1 subunit theta**
Synonyms	TCP-1-theta CCT-theta Renal carcinoma antigen NY-REN-15

Fig. 5. An Example of Union



Annotations				
Gene Ontologies	**Ontology**	**Annotation**	**Evidence**	**Source**
	Molecular Function	Unfolded protein binding	IEA	GOA/IPI
		ATP binding	IEA	GOA/IPI
		ATPase activity, coupled	TAS	GOA/IPI
	Biological Process	Protein folding	IEA	GOA/IPI
	Cellular Component	Cytosol	NR	GOA/IPI

Fig. 6. An Example of Selection



chromosome	21
base	from 29350518to 2937881
strand	+

Fig. 7. An Example of Split

Union/Selection. When individual source values under different source labels match individual seed values under one label, SIH infers a union mapping. For example, suppose a user creates seed values for *Name* (of *Protein*) by copy-and-paste of "T-complex protein 1 subunit theta", "TCP-1-theta", "CCT-theta", and "Renal carcinoma antigen NY-REN-15" from Figure 5. Then, in this case, SIH detects a union mapping from *Protein name* and *Synonyms* to *Name*. When individual seed values under a seed label match a subset of individual source values under only one source label, SIH infers a selection mapping. For example, in Figure 6, suppose that the seed values only include the first three values under *Annotation* because the user only cares about protein functions. SIH then can infer a selection mapping from the source label *Annotation* for only *Molecular Function* annotations. For information harvesting for a union mapping, SIH collects information from all source fields. For selection, if the desired values are labeled in the source pages as they are in Figure 6, SIH collects the information under the proper, restricted label. If not, the user needs to provide a selection expression to filter the values (see future work).

Split/Merge. When part of a seed value matches an individual source value, SIH detects a split mapping. For example, in Figure 3, only part of the source value "29,350,518 bp from pter" under the source label *Start* matches the seed value "29,350,518". In this case, SIH detects a split mapping. Sometimes, one source value could be split into multiple seed values. For example, the value under the label *base* in Figure 7 matches seed values for labels *Start* and *End* and thus should be split and mapped to two ontology concepts. To do the split, SIH stores and uses patterns. The pattern "<start> bp from pter" works for extracting *Start* values for the site of Figures 3 and 4, and the pattern "from <Start>to <End>" works for extracting *Start* and *End* values from the site from which Figure 7 was taken. A merge mapping is the opposite of a split mapping. If the user's form had a *single-label/single-entry* field for base that expected values like the base value in Figure 7, then for the site of Figure 3 and 4, SIH would need to merge *Start* and *End* values. For information harvesting, SIH collects values to be split by filtering them through generated patterns and collects values to be merged by obtaining all of them and concatenating them, separated by a delimiter (e.g. " - ").

To harvest information from multiple sites, the user specifies multiple seeds. The user does not specify another form, but does fill in the form with seed values from each new site. If, however, SIH can find a match in a new site with either the original seed values, or with any seed values it has already collected, the user need not even specify new seed values for the new site.

After SIH collects information, it checks cardinality constraints. For example, if SIH finds that each value under a single-label/single-entry field is unique, it marks this concept as a unique identifier for the base form or subform concept. If SIH observes that values for some fields are not available, it marks the field as optional. For example, if SIH finds that some web site does not provide information about *Molecular Weight* for a protein, it marks this concept as optional. If data collected contradicts user-provided constraints, SIH warns the user and allows the user to determine if any adjustment needs to be made.

6 Experimental Results, Conclusion, and Future Work

We have conducted some preliminary tests for SIH by creating the sample ontology in Figure 1 and extracting information form seven different web sites. Although there were seven web sites, we only needed to create three sets of seed values. Some web site had various values or values in different formats for the same concept. For example, some web sites used "minus" for *Orientation*, whereas other sites used "-". There were a total of 31 concept mappings. Among these 31 mappings, 11 were direct mappings, which SIH was able to handle 100% correctly. SIH also successfully detected and correctly processed all 4 union mappings it encountered. There were 15 split mappings, SIH detected and found the correct patterns for 12 of them. One error was due to the use of different delimiters in the same site, and two errors were due to the seed value being only a small subset of the source field. (Additional work on finding delimiters is needed.) SIH did not encounter any merge mapping—our specified fields were always at the lowest granularities. For selection mappings, we only tested the part where a source page uses a label to mark the selection. SIH encountered one selection mapping and was able to detect it successfully.

As a conclusion, we can say that seed-based harvesting of information via bio-ontologies appears to be both possible and reasonable. SIH can match source data with an ontology seed and gather information from different sources with respect to user-specific, personalized views.

Several directions remain to be pursued. First, we would like to support additional form features such as allowing the user to specify filter functions or desired units for form fields. Second, we want to integrate the data, not just harvest it. Finally, we want to improve SIH, so that the users do not need to create and fill our forms. We plan to have the users just highlight the values they want from sample pages; our future system would generate ontologies directly from these highlighted values.

References

1. P. Buitelaar, D. Olejnik, and M. Sintek. Ontolt: A protege plug-in for ontology extraction from text. In *Proceedings of the International Semantic Web Conference (ISWC'03): Demo Session*, October 2003.
2. P. Cimiano and J. Völker. Text2Onto—a framework for ontology learning and data-driven change discovery. In *Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB'05)*, pages 227–238, Alicante, Spain, June 2005.
3. Y. Ding, D.W. Lonsdale, D.W. Embley, M. Hepp, and L. Xu. Generating ontologies via language components and ontology reuse. In *Proceedings of 12th International Conference on Applications of Natural Language to Information Systems (NLDB'07)*, Paris, France, June 2007. (in press).
4. D.W. Embley, D.M. Campbell, Y.S. Jiang, S.W. Liddle, D.W. Lonsdale, Y.-K. Ng, and R.D. Smith. Conceptual-model-based data extraction from multiple-record Web pages. *Data & Knowledge Engineering*, 31(3):227–251, November 1999.
5. R. Navigli, P. Velardi, A. Cucchiarelli, and F. Neri. Quantitative and qualitative evaluation of the OntoLearn ontology learning system. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 1043–1050, Geneva, Switzerland, August 2004.
6. N.F. Noy, M. Sintek, S. Decker, M. Crubezy, R.W. Fergerson, and M. Musen. Creating semantic web contents with Protege-2000. *IEEE Intelligent Systems*, 16(2):60–71, March/April 2001.
7. A. Pivk. Automatic ontology generation from web tabular structures. *AI Communnications*, 19(1):83–85, 2006.
8. P. Spyns, D. Oberle, R. Volz, J. Zheng, M. Jarrar, Y. Sure, R. Studer, and R. Meersman. OntoWeb—a semantic web community portal. In *Proceedings of the 5th International Conference on Practical Aspects of Knowledge Management (PAKM'02)*, pages 189–200, Vienna, Austria, December 2002.
9. C. Tao and D.W. Embley. Automatic hidden-web table interpretation by sibling page comparison. 2007. (submitted for publication).
10. Y.A. Tijerino, D.W. Embley, D.W. Lonsdale, Y. Ding, and G. Nagy. Toward ontology generation from tables. *World Wide Web: Internet and Web Information Systems*, 8(3):251–285, September 2004.
11. Y. Wang, J. Völker, and P. Haase. Towards semi-automatic ontology building supported by large-scale knowledge acquisition. In *AAAI Fall Symposium On Semantic Web for Collaborative Knowledge Acquisition*, volume FS-06-06, pages 70–77, Arlington, Virginia, October 2006.
12. Y. Zhou. Generating data-extraction ontologies by example. Master's thesis, Brigham Young University, December 2005.