Intelligence in Wikipedia

Daniel S. Weld Fei Wu Eytan Adar Saleema Amershi
James Fogarty Raphael Hoffmann Kayur Patel Michael Skinner
University of Washington, Seattle, WA
weld@cs.washington.edu

Abstract

The *Intelligence in Wikipedia* project at the University of Washington is combining self-supervised information extraction (IE) techniques with a mixed initiative interface designed to encourage communal content creation (CCC). Since IE and CCC are each powerful ways to produce large amounts of structured information, they have been studied extensively — but only in isolation. By combining the two methods in a virtuous feedback cycle, we aim for substantial synergy. While previous papers have described the details of individual aspects of our endeavor [25, 26, 24, 13], this report provides an overview of the project's progress and vision.

Introduction

Recent years have shown that Wikipedia is a powerful resource for the AI community. Numerous papers have appeared, demonstrating how Wikipedia can be used to improve the performance of systems for text classification [9], semantic relatedness and coreference resolution [10, 21], taxonomy construction [20], and many other tasks.

However, it is also interesting to consider how AI techniques can be used to make Wikipedia more valuable to others and to amplify the *communal content creation* process which has driven Wikipedia. For example, one trend in the evolution of Wikipedia has been the introduction of *infoboxes*, tabular summaries of the key attributes of an article's subject. Indeed, infoboxes may be easily converted to semantic form as shown by Auer and Lehmann's DBPEDIA [2]. As evr more Wikipedia content is encoded in infoboxes, the resulting ensemble of schemata will form a knowledge base of oustanding size. Not only will this "semantified Wikipedia" be an even more valuable resource for AI, but it will support faceted browsing [27] and simple forms of inference that may increase the recall of question-answering systems.

The *Intelligence in Wikipedia* (IWP) project aims to develop and deploy AI technology to facilitate the growth, operation and use of Wikipedia. As our first objective we seek to accelerate the construction of infoboxes and their integration into a knowledge base. Our project is driving research on a number of exciting areas, but at the highest level it may be seen as a novel approach to gain synergy from the two predominant techniques for generating structured information:

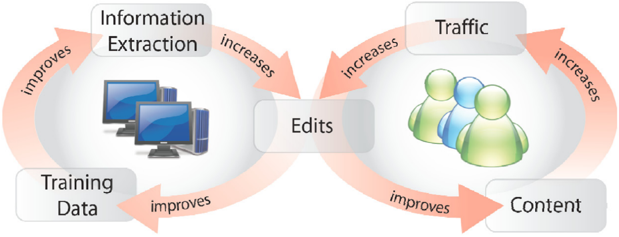


Figure 1: Adding IE to CCC makes a virtuous cycle.

- **Information Extraction (IE)** uses machine learning techniques to scrape tuples from the Web. For example, `zoominfo.com` aggregates information on people, `flipdog.com` gathers data on jobs, Citeseer [12] scrapes bibliographic and citation data, and Google extracts addresses and phone numbers from business Web pages. An advantage of IE is its autonomous nature, but the method has weaknesses as well: it usually requires a large set of expensive training examples for learning and it is error-prone, typically producing data with $80 - 90\%$ precision.
- **Communal Content Creation (CCC)** aggregates the actions of myriad human users to create valuable content. For example, the English version Wikipedia has over 2.3M[1] articles, eBay displays ratings for all sellers, and Netflix recommends movies based on preferences solicited from close to a million users. CCC often generates information which is more accurate than that generated by IE, but has a bootstrapping problem — it is only effective when applied on a high-traffic site. Furthermore, humans often require incentives to motivate their contribution and management to control spam and vandalism.

These techniques are complementary. For example, IE can be used to bootstrap content on a site attracting traffic, and CCC can be used to correct errors, improving training data and enabling a virtuous cycle as shown in Figure 1. But surprisingly there has been almost no prior research aimed at combining the methods, looking for additional synergies or

[1] As of April 2008

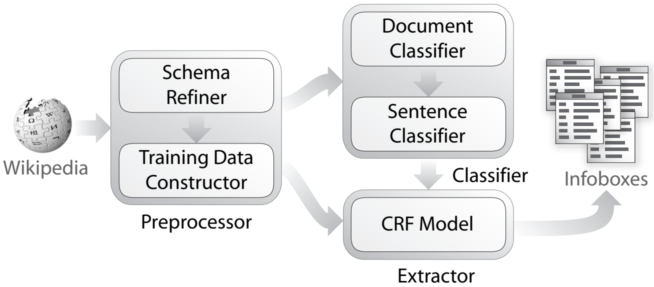


Figure 2: Architecture of Kylin's basic, self-supervised information-extraction system, before shrinkage is applied.

discerning the principles of combination. Many IE systems, e.g. *Citeseer, Rexa* [1], and *DBlife* [5] allow users to correct errors, but the interaction is rarely used and doesn't improve future learning accuracy.

The rest of this paper describes the architecture of our system, IWP, highlights some of the interesting AI techniques required, and sketches our ongoing vision. Space constraints preclude an ample discussion of related research; see our primary research papers (e.g., [25, 26, 24, 13]) for a detailed comparison.

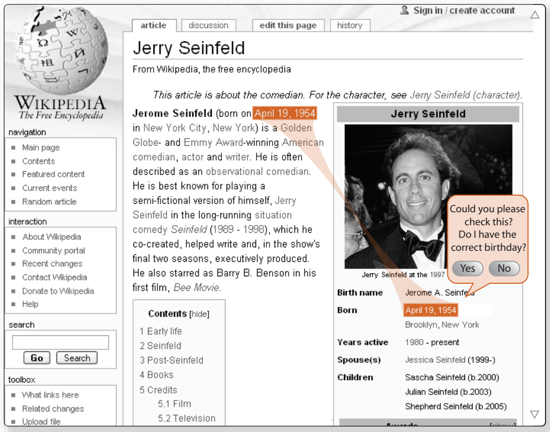
Self-Supervised Information Extraction

Automated information extraction is the key to bootstraping IWP's virtuous cycle of structured-information production. IWP starts with our Kylin system's self-supervised approach to extracting semantic relations from Wikipedia articles [25]. For a list of possible relations to extract and the training data for learning how to extract these relations, Kylin relies on infoboxes. A Wikipedia infobox is a relational summary of an article: a set of attribute / value pairs describing the article's subject (see [25] for an example). Not every article has an infobox and some infoboxes are only partially instantiated with values. Kylin seeks to create or complete infoboxes whenever possible. In this section, we review the architecture (Figure 2) of Kylin and discuss its main components.

Preprocessor The preprocessor selects and refines infobox schemata, choosing relevant attributes; it then generates machine-learning datasets for training sentence classifiers and extractors. Refinement is necessary for several reasons. For example, *schema drift* occurs when authors create an infobox by copying one from a similar article and changing attribute values. If a new attribute is needed, they just make up a name, leading to schema and attribute duplication.

Next, the preprocessor constructs two types of training datasets — those for sentence classifiers, and CRF attribute extractors. For each article with an infobox mentioning one or more target attributes, Kylin tries to find a unique sentence in the article that mentions that attribute's value. The resulting labelled sentences form positive training examples for each attribute; other sentences form negative training examples. If the attribute value is mentioned in several sentences, then one is selected heuristically.

Generating Classifiers Kylin learns two types of classifiers. For each class of article being processed, a heuristic *document classifier* is used to recognize members of the infobox class. For each target attribute within a class a *sentence classifier* is trained in order to predict whether a given sentence is likely to contain the attribute's value. For this, Kylin uses a maximum entropy model [19] with bagging. Features include a bag of words, augmented with part of speech tags.



Jerry Seinfeld

From Wikipedia, the free encyclopedia

This article is about the comedian. For the character, see Jerry Seinfeld (character).

Jerome Seinfeld (born on April 19, 1954 in New York City, New York) is a Golden Globe- and Emmy Award-winning American comedian, actor and writer. He is often described as an observational comedian. He is best known for playing a semi-fictional version of himself, Jerry Seinfeld in the long-running situation comedy *Seinfeld* (1989 - 1998), which he co-created, helped write and, in the show's final two seasons, executively produced. He also starred as Barry B. Benson in his first film, *Bee Movie*.

Contents [hide]
1 Early life
2 Seinfeld
3 Post-Seinfeld
4 Books
5 Credits
5.1 Film
5.2 Television

Jerry Seinfeld

Jerry Seinfeld at the 1997

Could you please check this? Do I have the correct birthday?

Yes No

Birth name	Jerome A. Seinfeld
Born	April 19, 1954 Brooklyn, New York
Years active	1980 - present
Spouse(s)	Jessica Seinfeld (1999-)
Children	Sascha Seinfeld (b.2000) Julian Seinfeld (b.2003) Shepherd Seinfeld (b.2005)

Figure 3: User interface mockup. Casual users are presented with a standard Wikipedia page highlighting a single attribute value; an ignorable popup window allows the user to verify the extraction if she wishes.

Learning Extractors Extracting attribute values from a sentence is best viewed as a sequential data-labelling problem. Kylin uses conditional random fields (CRFs) [17] with a wide variety of features (e.g., POS tags, position in the sentence, capitalization, presence of digits or special characters, relation to anchor text, etc.). Instead of training a single master extractor to clip all attributes, IWP trains a different CRF extractor for each attribute, ensuring simplicity and fast retraining.

Mixed Initiative Operation

While some of the CRF extractors learned by Kylin have extremely high precision, in most cases precision is well below that of humans. Thus, it is clear that a fully automated process for adding new infobox entries to Wikipedia is untenable. Instead, IWP aims to amplify human effort towards this task via a user interface. Figure 3 shows our first mockup design of this interface. In a series of interviews with members of the Wikipedia community, informal think-aloud design reviews, and a final online user study we refined, explored and evaluated the space of interfaces, focussing on several key design dimensions.

Contributing as a Non Primary Task Although tools already exist to help expert Wikipedia editors quickly make large numbers of edits [4], we instead want to enable contributions by the long tail of users *not yet contributing* [23]. In other words, we believe that pairing IIE with CCC will be most effective if it encourages contributions by people who had not otherwise planned to contribute. The next subsection discusses appropriate ways of drawing the attention of people, but an important aspect of treating contributing as a non primary task is the fact that many people will never even notice the potential to contribute. A design principle that therefore emerged early in our process is that unverified information should never be presented in such a way that it might be mistakenly interpreted as a part of the page.

Announcing the Potential for Contribution Any system based in community content creation must provide an incentive for people to contribute. Bryant et al. report that newcomers become members of the Wikipedia community by participating in peripheral, yet productive, tasks that contribute to the overall goal of the community [1]. Given this community, our goal is to make the ability to contribute sufficiently visible that people will choose to contribute, but not so visible that people feel an interface is obtrusive and attempting to coerce contribution. We designed three interfaces (Popup, Highlight, and Icon) to explore this tradeoff, evaluating their effectiveness in stimulating edits as well as users' perception of intrusivness in a study which used Google Adwords to recruit subjects [13].

Presenting Ambiguity Resolution in Context As shown in Figure 3 there are two plausible locations in an article for presenting each potential extraction: at the article text from which the value was extracted by Kylin or at the infobox where the data will be added. Presenting information at the latter location can be tricky becuase the user cannot verify information without knowing the context and varying the way of presenting this context can dramatically affect user participation.

While we developed some interface designs which yielded even higher participation rates, the "winner" of our study was an icon design which was deemed relatively unobtrusive and yet which led people to voluntarily contribute an average of one fact validation for every eight page visits, with the provided validations having a precision of 90%. By validating facts multiple times, we believe we can achieve very high precision on extracted tuples. By using these new tuples as additional training examples, the overall performance of IWP will keep increasing.

Generating an Ontology for Shrinkage

We also pushed on ways of improving the performance of IWP's autonomous extraction performance beyond that of our initial Kylin implementation. Our analysis showed that Kylin could learn accurate extractors if the class of articles already has numerous infoboxes for training[2]. However, it floundered on sparsely-populated classes — the majority of the cases. For example, the 7/17/07 snapshot of the English subset of Wikipedia contains 1756 classes, 1442 (82%) of which have fewer than 100 instances and 709 (40%) have 10 or fewer instances. This section explains how we mitigate this problem by employing shrinkage, a statistical technique for improving estimators based on limited training data [18].

Intuitively, instances of closely-related classes have similar attributes which are described using similar language. For example, knowing that `performer IS-A person`, and `performer.loc=person.birth_plc`, we can use data from `person.birth_plc` values to help train an extractor for `performer.loc`. The trick is automatically generating a good subsumption hierarchy which relates attributes between parent and child classes.

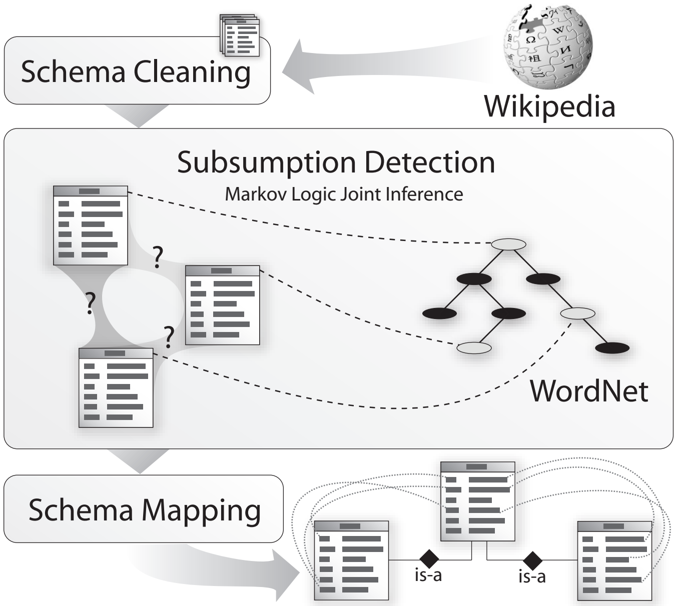


Figure 4: Architecture of the ontology generation system which powers shrinkage (hierarchical smoothing).

IWP's ontology generation system (named KOG) is described in [26], which automatically builds a rich ontology by combining Wikipedia infoboxes with WordNet using statistical-relational machine learning. At the highest level IWP computes six different kinds of features, some metric and some Boolean: *similarity measures*, *edit history patterns*, *class-name string inclusion*, *category tags*, *Hearst patterns* search-engine statistics, and *WordNet* mappings. These features are combined using statistical-relational machine learning, specifically joint inference over Markov logic networks, in a manner which extends [22].

Figure 4 shows KOG's architecture. First, its schema cleaner scans the infobox system to merge duplicate classes and attributes, and infer the type signature of each attribute. Then, the subsumption detector identifies the subsumption relations between infobox classes, and maps the classes to WordNet nodes. Finally, the schema mapper builds attribute mappings between related classes, especially between parent-child pairs in the subsumption hierarchy. KOG's taxonomy provides an ideal base for the shrinkage technique, as described below.

Given a sparse target infobox class, IWP's shrinkage module searches both up and down through the KOG ontology to aggregate data from parent and child classes. Appropriate weights are assigned to the aggregate data before augmenting the training dataset used to learn for new extractors. Shrinkage improves IWP's precision, but more importantly, it enormously increases recall on the long tail of sparse infobox classes [24]. For example, on the "performer" and "Irish newspapers" classes, the recall improvements are 57% and 457% respectively; and the area under the precision and recall curve improves 63% and 1430% respectively.

[2] Kylin's precision ranges from mid 70th to high 90th percent, and it's recall ranges from low 50th to mid 90th, depending on the attribute type and infobox class

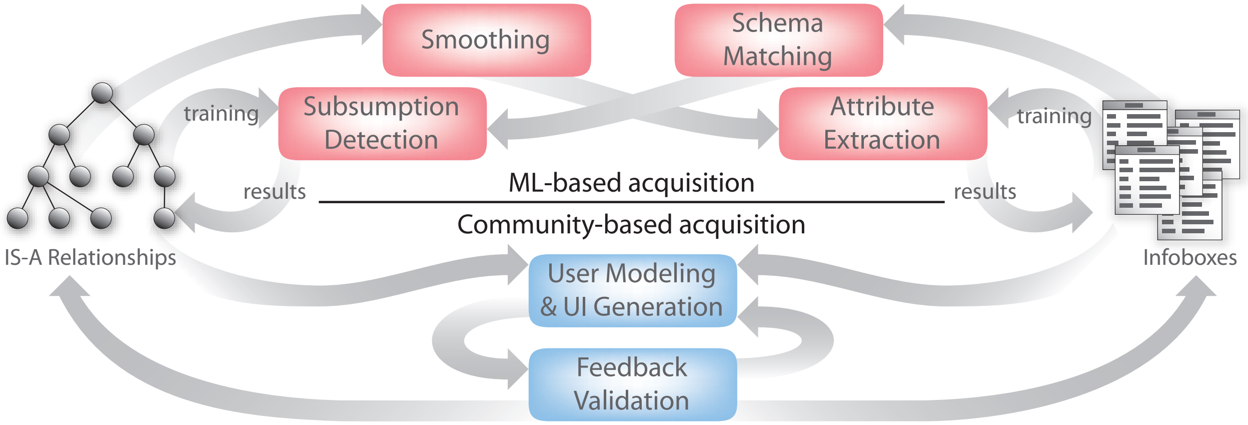


Figure 5: IWP interleaves information extraction with communal corrections and additions, creating the virtuous feedback cycle symbolized abstractly in Figure 1.

Retraining

Shrinkage enables IWP to find extra data within *Wikipedia* to help train extractors for sparse classes. A complementary idea is the notion of harvesting additional training data even from the *outside* Web. Here the problem becomes determining how to automatically identify relevant sentences given the sea of Web data. For this purpose, IWP utilizes TextRunner, an open information extraction system [3], which extracts semantic relations $\{r|r = \langle obj_1, predicate, obj_2\rangle\}$ from a crawl of about 10 million Web pages. Importantly for our purposes, TextRunner's crawl includes the top ten pages returned by Google when queried on the title of every Wikipedia article.

For each target attribute within an infobox, Kylin queries to identify and verify relevant sentences that mention the attribute's value. Sentences passing the verification are labeled as extra positive training examples. Kylin also identifies the phrases (predicates) which are harbingers of each target attribute. For example, ”was married to” and ”married” are identified for the ”person.spouse” attribute. These harbingers are used to eliminate potential false negative training examples generated from the Wikipedia data.

By adding new positive examples and excluding potential false negative sentences, retraining generates a cleaned and augmented training dataset which improves Kylin's performance. When used together with shrinkage, the improvement is dramatic, especially in terms of recall. For example, on the “performer” and “Irish newspapers” classes, the recall improvements are 73% and 585% respectively; and the area under the precision and recall curve improves 79% and 1755% respectively.

Furthermore, we note that the wording of text from the greater Web are more diverse than the relatively strict expressions used in many places in Wikipeidia[3]. By adding a wider variety of sentences to augment the training dataset, it would improve the robustness of IWP's extractors, which would potentially improve the recall, especially when we apply the extractors on the greater Web to harvest more semantic relations, as described in the following section.

Extracting from the General Web

Even when IWP does learn an effective extractor there are numerous cases where Wikipedia has an article on a topic, but the article simply doesn't have much information to be extracted. Indeed, a long-tailed distribution governs the length of articles in Wikipedia — around 44% of articles are marked as stub pages — indicating that much-needed information is missing. Additionally, facts that are stated using uncommon or ambiguous sentence structures also hide from the extractors. IWP exploits the general Web to retrieve facts which can't be extracted from Wikipedia.

The challenge for this approach — as one might expect — is maintaining high precision. Since the extractors have been trained on a very selective corpus, they are unlikely to discriminate irrelevant information. For example, an IWP extractor for a person's birthdate has been trained on a set of pages all of which have as their primary subject that person's life. Such extractors become inaccurate when applied to a page which compares the lives of several people — even if the person in question is one of those mentioned.

To ensure extraction quality, it is thus crucial to carefully select and weight content that is to be processed by IWP's extractors. We view this as an information retrieval problem, which IWP's web extraction module solves in the following steps: 1) It generates a set of queries and utilizes a general Web search engine, namely Google, to identify a set of pages which are likely to contain the desired information. 2) The top-k pages are then downloaded, and the text on each page is split into sentences, which are processed by IWP. 3) Each extraction is then weighted using a combination of factors, including the rank of the page, the extraction confidence, and the distance between the current sentence and the closest sentence containing the name of the concept/article.

When combining the extraction results from Wikipedia and the general Web, IWP gives Wikipedia higher weight, because it is likely that extractions from Wikipedia will be more precise. That is, in Wikipedia we can be more certain that a given page is highly relevant, is of higher quality, has

[3] It is possible that Wikipedia's replication stems from a pattern where one article is copied and modified to form another. A general desire for stylistic consistency is another explanation.

a more consistent structure, and for which IWP's extractors have been particularly trained.

Extracting from the Web further helps IWP's performance. For example, on the "performer" and "Irish newspapers" classes, the recall improvements are 90% and 743% respectively; and the area under the P/R curve improves 102% and 1771%.

Multi-Lingual Extraction

We have already described how certain forms of structure (e.g., infoboxes and edit histories) in Wikipedia facilitate content "semantification." But another important feature is the presence of parallel articles in different languages. While the English sub-domain of Wikipedia is largest with 2.3M articles, these comprise only 23% of the pages.[1] The remaining effort is distributed among over 250 languages, many growing at rates far exceeding their English counterparts. Certain pages represent direct translations as multilingual users build and maintain a parallel corpus in different languages. While parallel, the pure text of these articles lacks any alignment annotations. Furthermore, different levels of interest and expertise will drive certain versions of a page to evolve more quickly. For example, the French Wikipedia page for a French pop star might include all the newest albums while pages in other languages lag in these updates. This disparity in article details and quality is an opportunity to further leverage the virtuous feedback cycle by utilizing IE methods to add or update missing information by copying from one language to another, and utilizing CCC to validate and improve these updates.

In IWP, we are devising automated text-alignment methods to link between corresponding pages. Extractors can be trained in multiple languages, and joint inference will allow us to compensate for individual differences. The use of linked-editing techniques can ensure that once aligned, two pieces of information can be persistently tied. Updates in one language (e.g. the most recent population statistics for a Spanish city on the Spanish page) can be instantly propagated to other languages. Our initial efforts in this space have led to a mechanism for aligning semi-structured information such as infoboxes and other tables and lists. By using the occasions in which editors in different languages have generated the same structured mapping (e.g. nombre = "Jerry Seinfeld" and name = "Jerry Seinfeld"), we are able to train classifiers to recognize equivalence in which the mapping is less obvious (e.g. cónyuge = "Jessica Sklar" and spouse = "Jessica Seinfeld"). These classifiers, which in initial testing approach 86% accuracy, allow us to automatically build a translation dictionary and to support alignment and "patching" of missing information between languages. From these simple dictionaries and alignments we are developing more sophisticated techniques in IWP that both encourage multi-lingual CCC and provide a rich data source for automated translation and "downstream" applications.

Conclusion

The *Intelligence in Wikipedia* (IWP) project is developing AI methods to facilitate the growth, operation and use of Wikipedia. Our initial goal is the extraction of a giant knowledge base of semantic triples, which can be used for faceted browsing or as input to a probabilistic-reasoning-based question-answering system. We believe such a knowledge based is best created by synergistically combing information extraction (IE) and communal content creation (CCC) paradigms in a mixed initiative interface. This paper summarizes our overall architecture, aspects of self-supervised information extraction, the generation of an ontology for shrinkage, retraining, extraction from the general Web, multi-lingual extraction, and our initial study of mixed-initiative interfaces. While we have made considerable progress, there is much room for improvement in all our components and the IWP project is just beginning.

Abbreviated Vita

Daniel S. Weld is Thomas J. Cable / WRF Professor of Computer Science and Engineering at the University of Washington. After formative education at Phillips Academy, he received bachelor's degrees in both Computer Science and Biochemistry at Yale University in 1982. He landed a Ph.D. from the MIT Artificial Intelligence Lab in 1988, received a Presidential Young Investigator's award in 1989, an Office of Naval Research Young Investigator's award in 1990, was named AAAI Fellow in 1999 and deemed ACM Fellow in 2005. Prof. Weld is an area editor for the Journal of the ACM, on the editorial board of Artificial Intelligence, was a founding editor and member of the advisory board for the Journal of AI Research, was guest editor for Computational Intelligence and for Artificial Intelligence, and was Program Chair for AAAI-96.

In addition to his academic activities, Prof. Weld is an active entrepreneur with several patents and technology licenses. In May 1996, he co-founded Netbot Incorporated, creator of Jango Shopping Search and later acquired by Excite. In October 1998, he co-founded AdRelevance, a revolutionary monitoring service for internet advertising which was acquired by Media Metrix and subsequently by Nielsen NetRatings. In June 1999, Weld co-founded data integration company Nimble Technology which was acquired by the Actuate Corporation. In January 2001, he joined the Madrona Venture Group as a Venture Partner and member of the Technical Advisory Board.

Prof. Weld's work on the *Intelligence from Wikipedia* project [25, 26, 24, 13] follows considerable related work on information extraction, e.g. [6, 15, 16, 8] and intelligent interfaces, e.g. [14, 11].

Acknowledgments

We thank Ivan Bechastnikh, Oren Etzioni, Krzysztof Gajos, Travis Kriplean, Stefan Schoenmackers, members of the KnowItAll group and students of CSE574 for useful conversations and ideas. We also thank anonymous reviewers for valuable suggestions and comments. This work was supported in part by NSF grant IIS-0307906, ONR grant N00014-06-1-0147, SRI CALO grant 03-000225 and the WRF / TJ Cable Professorship.

References

[1] http://rexa.info.

[2] Sören Auer and Jens Lehmann. What have Innsbruck and Leipzig in common? Extracting semantics from wiki content. In *Proc. ESWC*, 2007.

[3] Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. Open information extraction from the Web. In *Proc. of the 20th IJCAI*, 2007.

[4] D. Cosley, D. Frankowski, L. Terveen, and J. Riedl. Suggestbot: Using intelligent task routing to help people find work in wikipedia. In *Proc. of the 2007 Conf. on Intelligent User Interfaces*, January 2007.

[5] P. DeRose, W. Shen, F. Chen, A. Doan, and R. Ramakrishnan. Building structured web community portals: A top-down, compositional, and incremental approach. In *Proc. of the 33rd Int'l Conf. on Very Large Databases (VLDB-07)*, Vienna, Austria, 2007.

[6] R. Doorenbos, O. Etzioni, and D. Weld. A scalable comparison-shopping agent for the World-Wide Web. In *Proc. of the First Int'l Conf. on Autonomous Agents*, pages 39–48, Marina del Rey, CA, 1997.

[7] D. Downey, S. Schoenmackers, and O. Etzioni. Sparse information extraction: Unsupervised language models to the rescue. In *Proc. of ACL*, 2007.

[8] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates. Unsupervised named-entity extraction from the Web: An experimental study. *Artificial Intelligence*, 165(1):91–134, 2005.

[9] E. Gabrilovich and S. Markovitch. Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge. In *Proc. of the 21st Nat'l Conf. on Artificial Intelligence (AAAI)-06*, pages 1301–1306, 2006.

[10] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proc. of the 20th Int'l Joint Conf. on Artificial Intelligence (IJCAI-07)*, Hyderabad, India, January 2007.

[11] K. Z. Gajos, J. O. Wobbrock, and D. S. Weld. Decision-theoretic user interface generation. In *Proc. of the 22^{nd} AAAI Conf. on Artificial Intelligence (AAAI-08)*, 2008.

[12] C. Lee Giles, Kurt Bollacker, and Steve Lawrence. CiteSeer: An automatic citation indexing system. In Ian Witten, Rob Akscyn, and Frank M. Shipman III, editors, *Digital Libraries 98 - The Third ACM Conference on Digital Libraries*, pages 89–98, Pittsburgh, PA, June 23–26 1998. ACM Press.

[13] R. Hoffmann, S. Amershi, K. Patel, F. Wu, J. Fogarty, and D. S. Weld. Amplifying community content creation with mixed-initiative information extraction. *Submitted for publication*, 2008.

[14] R. Hoffmann, J. Fogarty, and D. S. Weld. Assieme: finding and leveraging implicit references in a web search interface for programmers. In *Proc. of UIST*, pages 13–22, 2007.

[15] N. Kushmerick, D. Weld, and R. Doorenbos. Wrapper Induction for Information Extraction. In *Proc. of the 15^{th} Int'l Joint Conf. on Artificial Intelligence (IJCAI-97)*, pages 729–737, 1997.

[16] C. T. Kwok, O. Etzioni, and D. Weld. Scaling question answering to the Web. *ACM Transactions on Information Systems (TOIS)*, 19(3):242–262, 2001.

[17] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of the ICML01*, Edinburgh, Scotland, May 2001.

[18] A. McCallum, R. Rosenfeld, T. Mitchell, and A. Ng. Improving text classification by shrinkage in a hierarchy of classes. In *Proc. of ICML-98, 15^{th} Int'l Conf. on Machine Learning (ICML-98)*, pages 359–367, 1998.

[19] K. Nigam, J. Lafferty, and A. McCallum. Using maximum entropy for text classification. In *Proc. of the IJCAI-99 Workshop on Machine Learning for Information Filtering*, 1999.

[20] S. P. Ponzetto and M. Strube. Deriving a large scale taxonomy from wikipedia. In *Proc. of the 22^{nd} Nat'l Conf. on Artificial Intelligence (AAAI-07)*, pages 1440–1445, 2007.

[21] S. P. Ponzetto and M. Strube. Knowledge derived from wikipedia for computing semantic relatedness. *Journal of AI Research (JAIR)*, 30:181–212, 2007.

[22] R. Snow, D. Jurafsky, and A. Ng. Semantic taxonomy induction from heterogenous evidence. In *Proc. of ACL06*, 2006.

[23] J. Voss. Measuring Wikipedia. In *Int'l Conf. of the Int'l Society for Scientometrics and Informetrics*, 2005.

[24] F. Wu, R. Hoffmann, and D. S. Weld. Information extraction from Wikipedia: Moving down the long tail. *Submitted for publication*, 2008.

[25] F. Wu and D. Weld. Autonomouslly semantifying Wikipedia. In *Proceedings of CIKM07*, Lisbon, Porgugal, 2007.

[26] F. Wu and D. S. Weld. Automatically refining the wikipedia infobox ontology. In *Proceedings of WWW08*, 2008.

[27] K. Yee, K. Swearingen, K. Li, and M. Hearst. Faceted metadata for image search and browsing. In *Proceedings of SIGCHI03*, 2003.