The Fast and the Numerous – Combining Machine and Community Intelligence for Semantic Annotation

Sebastian Blohm, Markus Krötzsch and **Philipp Cimiano**
Institute AIFB, Knowledge Management Research Group
University of Karlsruhe
D-76128 Karlsruhe, Germany
{blohm, kroetzsch, cimiano}@aifb.uni-karlsruhe.de

Abstract

Starting from the observation that certain communities have incentive mechanisms in place to create large amounts of unstructured content, we propose in this paper an original model which we expect to lead to the large number of annotations required to semantically enrich Web content at a large scale. The novelty of our model lies in the combination of two key ingredients: the effort that online communities are making to create content and the capability of machines to detect regular patterns in user annotation to suggest new annotations. Provided that the creation of semantic content is made easy enough and incentives are in place, we can assume that these communities will be willing to provide annotations. However, as human resources are clearly limited, we aim at integrating algorithmic support into our model to bootstrap on existing annotations and learn patterns to be used for suggesting new annotations. As the automatically extracted information needs to be validated, our model presents the extracted knowledge to the user in the form of questions, thus allowing for the validation of the information. In this paper, we describe the requirements on our model, its concrete implementation based on Semantic MediaWiki and an information extraction system and discuss lessons learned from practical experience with real users. These experiences allow us to conclude that our model is a promising approach towards leveraging semantic annotation.

Introduction

With the advent of the so called Web 2.0, a large number of communities with a strong will to provide content have emerged. Essentially, these are the communities behind social tagging and content creation software such as del.icio.us, Flickr, and Wikipedia. Thus, it seems that one way of reaching massive amount of annotated web content is to involve these communities in the endeavour and thus profit from their enthusiasm and effort. This requires in essence two things: semantic annotation functionality seamlessly integrated into the standard software used by the community in order to leverage its usage and, second, an incentive mechanism such that people can immediately profit from the annotations created. This is for example the key idea behind projects such as Semantic MediaWiki (Krötzsch *et al.* 2007) and Bibsonomy (Hotho *et al.* 2006). Direct incentives for creating semantic annotations in a Semantic MediaWiki are for example semantic browsing and querying functionality, but most importantly the fact that queries over structured knowledge can be used to automatically create views on data, e.g. in the form of tables.

However, creating incentives and making annotation easy and intuitive will clearly not be enough to really leverage semantic annotation at a large scale. On the one hand, human resources are limited. In particular, it is well known from Wikipedia and from tagging systems that the number of contributors is relatively small compared to the number of information consumers. On the other hand, we need to use human resources economically and wisely, avoiding that people get bored by annotating the obvious or the same things again and again. This is where standard machine learning techniques which detect regularities in data can help. However, any sort of learning algorithm will produce errors, either because they overgenerate or they overfit the training data. Thus, human verification is still needed. We argue that this verification can be provided by the community behind a certain project if the feedback is properly integrated into the tools they use anyway. This opens the possibility to turn information consumers into "passive annotators" which, in spite of not actively contributing content and annotations, can at least verify existing annotations if it is easy enough.

The idea of semi-automatically supporting the annotation process is certainly not new and has been suggested before. However, we think that it is only the unique combination of large community efforts, learning algorithms and a seamless integration between both that will ultimately lead to the kind of environments needed to make large scale semantic annotation feasible.

In this paper we thus describe a novel paradigm for semantic annotation which combines the effort of communities such as Wikipedia (the *community intelligence* or *"the numerous"* dimension in our model) which contribute to the massive creation of content with the benefits of a machine learning approach. The learned model captures people's annotation behaviour and is thus able to quickly extract new information and suggest corresponding annotations to be verified by the user community (this the *machine intelligence* or *"the fast"* dimension in our model).

The remainder of this paper is organised as follows. In the next section we describe our approach to combining ma-

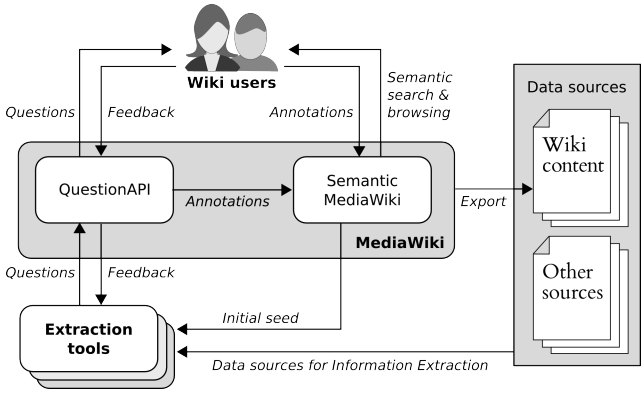

Figure 1: Integrating (semantic) wikis with Information Extraction tools – basic architecture.

chine and human intelligence for semantic annotation in a wiki setting and describe how Semantic MediaWiki can be used for this purpose. Then, we derive requirements for such an integration and describe its corresponding architecture subsequently. We present an implementation based on the English Wikipedia and discuss practical experiences before reviewing related work and concluding.

Combining Human and Machine Intelligence

The crucial aspect of our model is that community members and information extraction algorithms interact in such a way that they can benefit from each other. Humans benefit from the fact that information extraction systems can support them in the tedious work of manual annotation, and algorithms exploit human annotations to bootstrap and learn patterns to suggest new annotations. The workflow in our model is thus as follows:

1. Extraction tools use existing high-quality and community-validated human annotations to learn patterns in data, leading to the extraction of new annotations.
2. Users are requested to verify extracted data so as to confirm or reject it. This is done by presenting questions to users.
3. Confirmed extraction results are immediately incorporated into the wiki, if possible.
4. User replies are evaluated by extraction tools to improve future results (learning), and to gather feedback on extraction quality (evaluation), returning to (1) in a bootstrapping fashion.

The model thus is cyclic, but also asynchronous in nature, since learning, annotation, verification, and incorporation into the wiki interact with each other asynchronously and not in a serialised manner. This mode of operation is reflected in the requirements we present below.

Assuming the model above, we present a concrete architecture and implementation that realises the above model in which extraction tools and wiki users interact in a rather asynchronous mode, benefiting from each other. Figure 1 shows the relevant components – *(Semantic) MediaWiki*, the *extraction tools*, a novel *QuestionAPI* as well as their basic interactions. We have selected the wiki-engine MediaWiki as a basis for our work, since this system is widely used on publicly accessible sites (including Wikipedia), such that large amounts of data are available for annotation. Moreover, the free add-on *Semantic MediaWiki* (SMW) extends MediaWiki with means for creating and storing semantic annotations that are then exploited to provide additional functionality to wiki-users (Krötzsch *et al.* 2007). This infrastructure is useful in two ways: first, it allows wiki-users to make direct use of the freshly acquired annotations, and, second, it can support extraction tools by providing initial (user-generated) example annotations as seeds for learning algorithms.

As shown in Figure 1, our general architecture makes little assumptions about the type and number of the employed extraction tools, so that a wide range of existing tools should be useable with the system (see the Related Work section for an overview). As a concrete example for demonstrating and testing our approach, we have selected the *Pronto* information extraction system (Blohm & Cimiano 2007).

Requirements on User Interaction

Successful wiki projects live from vivid user communities that contribute and maintain content, and therefore social processes and established interaction paradigms are often more important than specific technical features. Likewise, any extended functionality that is to be integrated into existing wikis must also take this into account. This has led us to various requirements.

(U1) Simplicity Participating in the annotation process should be extremely simple for typical wiki users, and should ideally not require any prior instruction. The extension must match the given layout, language, and interface design.

(U2) Unobtrusiveness and opt-out In order to seriously support real-world sites an extension must not obscure the actual main functions of the wiki. Especially, it must be acknowledged that many users of a wiki are passive readers who do not wish to contribute to the collaborative annotation process. Registered users should be able to configure the behaviour of the extension where possible.

(U3) User gratification Wiki contributors typically are volunteers, such that it is only their personal motivation which determines the amount of time they are willing to spend for providing feedback. Users should thus be rewarded for contributions (e.g. by giving credit to active contributors), and they should understand how their contribution affects and improves the wiki.

(U4) Entertainment Even if users understand the relevance of contributing feedback, measures must be taken to ensure that this task does not appear monotone or even stupid to them. Problems can arise if the majority of changes proposed by extraction tools are incorrect (and maybe even unintelligible to humans), or if only very narrow topic areas are subject to extraction.

(U5) "Social" control over extraction algorithms Wiki users and contributors take responsibility for the quality of the wiki as a whole. Changes to wiki content are

```
The '''Peugeot 204''' is a
[[class::compact car]] produced
by the [[French]] manufacturer
[[manufacturer::Peugeot]] between [[1965]]
and [[1976]].
```

Figure 2: Annotated wiki source text.

Model	Manufacturer	Class
BMW M1	BMW	Super car
Dodge A100	Chrysler Corporation	
Ferrari F430	Ferrari	Sports car
Honda NSX	Honda Motor Company	Sports car
Porsche Cayman	Porsche	Sports car
Rover Metro	Austin Rover Group	Supermini car

Figure 3: Query result in Semantic MediaWiki: automobiles with mid-engine/rear-wheel drive, their manufacturers, and classes where specified.

frequently discussed and reverted if deemed inappropriate. Credibility and authority play a crucial role here. Frequent inappropriate feedback requests and content modifications by information extraction systems may lead to frustration within the community. Therefore we propose to make the extraction tools identifiable by giving their name, methodology and author so that users can identify the origin of an annotation and contact responsible persons.

Semantic MediaWiki

Semantic MediaWiki (SMW) is an open source semantically enhanced wiki engine that enables users to annotate the wiki's contents with explicit, machine-readable information. This information is then used to offer semantic search and browsing facilities within the wiki, as well as to export semantic data in the standardised OWL/RDF format, thus supporting data reuse in other applications. A brief overview of both aspects is provided here – for further details and related work see (Krötzsch *et al.* 2007).

SMW's main annotation mechanism is the assignment of property-value-pairs to pages. Property values might be other pages (e.g. to express relations like "father of"), or data values of a variety of specialised datatypes (e.g. for describing properties like "birthdate" or "population").

Formally, these annotations are interpreted in the Web Ontology Language OWL DL[1], using the *Semantic Wiki Vocabulary and Terminology SWIVT*[2]. Categories map to OWL classes, and categorised pages map to elements of such a class. Properties are directly interpreted as *object* or *datatype properties* in OWL DL, depending on their datatype as declared in the wiki.

Semantic search and browsing features in SMW are included into the wiki interface. One major feature of this kind are semantic queries formulated in a wiki-like query syntax.

[1] http://www.w3.org/2004/OWL/

[2] http://semantic-mediawiki.org/swivt/

Figure 2 provides a simple example of annotated wiki text, which is the basis for the HTML output of a wiki-page. Square brackets is the standard syntactic notation for hyperlinks, and in SMW these links can be annotated with properties separated by `::` from the link-target. Based on such annotations, SMW can dynamically generate lists of query results, as e.g. the one shown in Figure 3.

Pronto

Pronto is an information extraction system able to extract relations from large collections of text such as the Web on the basis of minimal supervision. The minimal supervision consists of between 5-30 seed examples for the relation in question. Pronto works in a bootstrapping-like fashion by starting from the examples provided and learns new patterns to extract the relation in question by looking at the occurrences of the seed examples in the text collection, generalising these to yield general patterns. These patterns are then used to extract new examples and iterate the process. A pattern extracting the relation *productOf* between products and their manufacturing companies could for example look as follows:

"ARG_1 | is made by ARG_2 and runs *ANY* at"

where ARG_1 and ARG_2 represent the argument slots, "|" marks the separation between title and link context (in the case of applying Pronto to a wiki), and *ANY* is a wildcard that may represent any token. A more detailed description of the Pronto system can be found in (Blohm & Cimiano 2007).

System Design

In this section we discuss the concrete design and implementation of our approach, which realises the basic interactions shown in Figure 1. In order to enable easy integration of many extraction tools in asynchronous operation, all information exchange between wiki and extractors is realised via simple web interfaces, and this web API forms one major part of our *QuestionAPI* extension of MediaWiki developed in the context of the work described here. The other two main components of this module are its internal management of questions and answers, and its user interface extensions in the wiki. All three components will be described in detail below, and it will be explained how the requirements identified are addressed by our particular design. Finally, we explain how contextual information is used to control information requests based on user preferences and content.

User Interface

The main visible component of the QuestionAPI is its extension of the wiki user interface. Requests for feedback on extraction results are presented to the user as multiple-choice questions in a simple web-form, as shown at the bottom of Figure 4. Although we consider further answer formats, the current implementation supports only the answers "yes" and "no", as well as a third option to defer a question. This last option allows users to skip questions without answering them, so that they can continue with other questions instead of accumulating questions that they are unable

Figure 4: Questions to users might be displayed at the bottom of wiki pages.

or unwilling to answer. Details on question scheduling are discussed in the following section.

Providing feedback is thus extremely simple, even for users who are not normally editing wiki-text (U1). Simplicity is also achieved by suitable question construction:

- Questions should be specific and informative, and they should use natural formulations instead of technical terms.
- Questions can contain wiki mark-up, and especially they can contain hyperlinks to relevant wiki-pages. This makes it easier for users to look up information.

The architecture assumes that the information extractors implementing the question API will provide their questions in natural language. Note that the right formulation of a question can not be meaningfully automated by using generic templates. Thus, we assume that every information extraction system is responsible to deliver an appropriate formulation of their questions in natural language.

All questions are associated with the extraction tool that requested the feedback, and this information is displayed with each question. A wiki page is maintained for each extraction tool, so that users can find additional information or provide comments (U5).

Besides the general form of the request interface, an important question is *where* to display questions in the wiki. Following our requirement for unobtrusiveness and opt-out (U2), the QuestionAPI can be configured to display a variable number of questions either at the bottom of all wiki pages, or only via a specific web interface ("special page") of the wiki.

After answering one or more questions, users are shown a summary of the submitted answers, as well as the option to answer further questions. The QuestionAPI supports *direct changes* based on answers to questions such that if a user has confirmed a certain semantic information, the QuestionAPI directly adds this fact as an annotation to the wiki. If this is enabled, changes will be done immediately when submitting an answer, and the answering user will get credit for the change just as if she would have edited the wiki manually. While this helps to increase user motivation (U3), it may also seem somewhat risky. But direct changes only *simplify* the editing process – the question whether or not a single user may modify a page still depends on the wiki's settings.

The Web API

The *QuestionAPI* extends MediaWiki with a simple web-based API that extraction tools can use to exchange information with the wiki. The API is protected by a permission control system based on MediaWiki's user permission management. Authentication works by associating to every extraction tool a wiki user-account that is then granted permission to access the question API. Other than being an indispensable feature for preventing abuse of the QuestionAPI, this mechanism also facilitates the management of requests and answers by extraction tools, such that extractors can access only data related to their own requests. Besides the simple use of session cookies for this purpose, all communication is completely stateless.

The QuestionAPI enables extraction systems to pose questions, to request gathered user feedback, and to remove questions from the system. Questions are added by supplying the question text as a parameter (possibly with wiki markup), after which a numerical question ID is returned by the wiki (or 0 if the question was denied). Lists of answers are provided in a simple XML format, and extraction tools may request either all available answers (to their questions), or specify a single question directly. A question is deleted from the system by supplying its ID, and this will also cause all answers to that question to be dropped from the system (though it is possible to have both archived by QuestionAPI as well, e.g. for later statistical evaluation).

The specification of direct changes currently works by specifying a string replacement *and* the page context of that replacement. The latter ensures that replacements happen only if the page still (locally) corresponds to the version inspected by the extraction tool. If other changes occurred, modifications need to be done manually by users.

Practical Experiences

We now present experiences gathered with the implementation of our collaborative semantic annotation framework. We have set up an integrated system based on Wikipedia data[3] which we presented to community members in order to collect feedback and usage data.

The observations discussed here are not meant to be a formal evaluation – information extraction with Pronto on SMW-like annotations on Wikipedia has been formally evaluated in (Blohm & Cimiano 2007), and performance and usage statistics for SMW have been published in (Krötzsch *et al.* 2007). What remains to be investigated is community uptake of the feedback extension as such, and the utility of the derived information. While extensive studies of these aspects must be left to future research, our initial tests have provided us with important insights for improving the current design.

[3] `http://testserver.semantic-mediawiki.org`

We created a mirror of the English Wikipedia based on a Wikipedia database dump from December 17th 2006. The server runs current versions of MediaWiki (1.12alpha) and SMW (1.0RC1), as well as our new extension QuestionAPI. For maintenance and performance reasons, software components were distributed over three server-sized computers: one running the PHP server for MediaWiki and its extension, one providing the database, and one running the Pronto extraction system. The systems were able to serve pages at below 1 second response time, and to run Pronto at its regular extraction rate of 0.3 facts per second.

Experienced wiki users and developers were asked to test the system via wiki-related mailing lists, and during a time of 5 days, 40 users (estimated from the number of distinct IPs) provided a total of 511 answers to the QuestionAPI.

Of the 511 questions answered, 51% were answered with "no", 34% were deferred, and the remaining 15% were answered with "yes" which in our setup led to automatic knowledge insertion. All users reacted positively to the interaction paradigm. The general purpose of the questions was quickly understood and appreciated, and no concerns were expressed with respect to obstructiveness or lack of simplicity. Several users mentioned that the questions reminded them of a quiz game, and suggested further uses of this extension beyond information extraction. We interpret this as positive effect with respect to the entertainment requirement (U4).

During the experiment, the option for deferring a question had been labelled "don't know" which was changed to "ask someone else" only later. This labelling is assumed to be responsible for the large portion of "don't know" answers: users who considered the questions as a kind of quiz mentioned that they perceived it as "cheating" to look up an answer that they were not sure about, such that "don't know" was considered more appropriate. This indicates that some introduction and/or clearer labelling is still needed to better convey the purpose of the questions. One consequence of this insight was the relabelling of "don't know" to "ask someone else" so as to communicate that personal knowledge is not to be tested, while still encouraging an answer by reminding the user that the task will otherwise be left to other users.

Besides some bug reports about character encoding, the only actual complains from users were related to the content of some types of questions, especially in cases where systematic errors occurred. This also produced some suggestions for filtering Wikipedia-specific extraction errors, e.g. caused by special kinds of frequent summary articles ("List of ...") that can normally not be involved in any relation.

In order to account for these observations, we formulate an extension of the *entertainment* requirement (U4): It is important to ensure that systematic errors in suggested relations are minimised beforehand, and excluded from verification through collaborative annotation. One interesting approach to do this automatically could be the use of unsupervised clustering methods that detect regularities, and to exclude questions belonging to large clusters for which only "no" answers have been provided so far. For this purpose, an additional answer option can be introduced to allow the users to mark individual relation instances as "unreasonable" suggestions.

Related Work

Annotation of web content has become very popular in particular as *tagging* of various kinds of media resources. Cameron Marlow et al. (Marlow *et al.* 2006) give an overview of tagging systems, and discuss dimensions in which they can differ. While not a tagging system in the stricter sense, the setup presented here would thereby be classified as a *free-for-all set model* system with high *resource connectivity* and a special form of *tag support*. The paper discusses various forms of incentives ranging from future retrieval to opinion expression. As Wikipedia already has a vivid community, we did not consider incentives for this study, and assume that our architecture helps to involve a larger user community by providing a low-entry threshold for contribution. An innovative approach with respect to incentives and human-machine collaboration in tagging is the ESP game (von Ahn & Dabbish 2004) which asks pairs of users to come up with common tags for images by guessing what the other user might tag. Further related work is done in the field of assisted semantic annotation of websites (e.g. (Dzbor, Domingue, & Motta 2003)). While our approach is largely tailored to semantifying sources like Wikipedia, other projects have studied the interaction between human input of facts and data mining technology. The Open Mind initiative studies the interaction of Internet users and knowledge bases. Their Common Sense (Pentney *et al.* 2007) system prompts users for natural language statements on a given entity. In a similar way, the Knowledge Base of the True Knowledge^{TM} question answering system can be extended by users.

Unlike in classical tagging, annotations in Semantic MediaWiki are structured statements that establish relationships between entities, or describe properties of these. This is possible because each page is assumed to describe an ontological element, and links are assumed to express relations between them. As described above, annotations in SMW have a formal semantics suitable for exchanging them via the Web. Some tagging systems are also working towards a more formal interpretability of tags. Flickr (`http://www.flickr.com`) introduced "machine tags" which allow unambiguous expression of facts about the annotated media. Bibsonomy (Hotho *et al.* 2006) provides the possibility to organise tags by asserting relations among them. The Spock person search engine (`http://www.spock.com`) provides the possibility to mark existing tags as correct and incorrect, which is not completely unlike the question based interaction in our setting.

While in our implementation we use information extraction from text to automatically derive suggested annotations of Wikipedia hyperlinks, our architecture is not limited to that setting. As reviewed and discussed in (Hendler & Golbeck 2008), much potential lies in the links and network structure as well as in social connections between users. The authors argue that the social interactions enabled by annotation constitute an important incentive for producing them.

Wikipedia is currently widely used for information extraction from text. Suchanek et al. (Suchanek, Kasneci, & Weikum 2007) have focussed on high-precision ontology learning and population with methods specifically tailored to Wikipedia. Wikipedia's category system is exploited assuming typical namings and composition of categories that allow to deduce semantic relations from category membership. In (Ruiz-Casado, Alfonseca, & Castells 2005) information extraction from Wikipedia text is done using hyperlinks as indicators for relations just like in the present study. As opposed to the work presented here it relies on WordNet as a hand-crafted formal taxonomy and is thus limited to relations for which such sources exist. Strube and Ponzetto use the taxonomy of the Wikipedia categories to define a measure for the semantic relatedness between words (Strube & Ponzetto 2006).

Conclusion and Next Steps

We have presented a new approach for facilitating semantic annotation of wikis by means of community-supervised information extraction, and we have presented a concrete practical realisation of this idea based on Semantic MediaWiki and an extraction system. Our robust and flexible design enables the loose, web-based integration of a wide range of extraction tools into existing community portals – thus tapping a large application field for information extraction on the one hand, and new content-creation solutions for community platforms on the other.

Our contribution removes the major barrier between two vibrant fields of research and application, and thus opens up a multitude of new opportunities for both areas. The first step certainly is to apply and evaluate information extraction tools on real-world community platforms. Our approach has been designed to be completely open, such that existing extraction tools can use our system with very little effort. We will open our Wikipedia-mirror to help developers of extraction tools to conduct tests in large scale real-world contexts, and to solicit user-feedback. We also consider a similar setup for conducting a "Wikipedia extraction challenge" where various types of extraction tools can demonstrate their utility in a kind of annotation contest. Further future work includes putting questions in contexts where visitors can be assumed to have the knowledge to answer them, integrating more question types. Additionally aggregating multiple user answers (e.g. by majority vote) could increase annotation quality.

On the other hand, there is a very real need for high quality and high coverage annotations in modern community sites. Many users of our Semantic MediaWiki system have made this request, both in community portals and in intranet applications.

Thus, when practical experiments have shown the maturity of extraction tools, there is also a clear path towards wide adoption and exploitation (economic or otherwise, e.g. in semantifying Wikipedia). In this way, information extraction – currently still mostly a mere consumer of Web-content – can take its proper place as a key technology for modern community platforms, and a major enabler of the Semantic Web.

Acknowledgements

This work was funded by the X-Media project (www.x-media-project.org) sponsored by the European Commission as part of the Information Society Technologies (IST) program under EC grant number IST-FP6-026978.

References

Blohm, S., and Cimiano, P. 2007. Using the web to reduce data sparseness in pattern-based information extraction. In *Proceedings of the ECML PKDD*. Springer.

Dzbor, M.; Domingue, J.; and Motta, E. 2003. Magpie – Towards a Semantic Web browser. In *Proc. 2nd International Semantic Web Conference (ISWC-03)*, volume 2870 of *Lecture Notes in Computer Science*, 690–705.

Hendler, J., and Golbeck, J. 2008. Metcalfe's law, Web 2.0, and the Semantic Web. *Journal of Web Semantics* 6(1):14–20.

Hotho, A.; Jäschke, R.; Schmitz, C.; and Stumme, G. 2006. BibSonomy: A social bookmark and publication sharing system. In *Proc. 2006 Conceptual Structures Tool Interoperability Workshop*, 87–102.

Krötzsch, M.; Vrandečić, D.; Völkel, M.; Haller, H.; and Studer, R. 2007. Semantic Wikipedia. *Journal of Web Semantics* 5(4):251–261.

Marlow, C.; Naaman, M.; Boyd, D.; and Davis, M. 2006. HT06, tagging paper, taxonomy, Flickr, academic article, to read. In *Proc. 17th Conf. on Hypertext and Hypermedia (HYPERTEXT-06)*, 31–40. New York, NY, USA: ACM.

Pentney, W.; Philipose, M.; Bilmes, J. A.; and Kautz, H. A. 2007. Learning large scale common sense models of everyday life. In *Proc. 22nd Nat. Conf. on Artificial Intelligence (AAAI-07)*, 465–470.

Ruiz-Casado, M.; Alfonseca, E.; and Castells, P. 2005. Automatic extraction of semantic relationships for WordNet by means of pattern learning from Wikipedia. In *Natural Language Processing and Information Systems*. Berlin/Heidelberg: Springer.

Strube, M., and Ponzetto, S. P. 2006. WikiRelate! Computing semantic relatedness using Wikipedia. In *Proc. 21st Nat. Conf. on Artificial Intelligence (AAAI-06)*, 1419–1424.

Suchanek, F. M.; Kasneci, G.; and Weikum, G. 2007. Yago: A core of semantic knowledge. In *Proc. 16th Int. Conf. on World Wide Web (WWW-07)*, 697–706. ACM Press.

von Ahn, L., and Dabbish, L. 2004. Labeling images with a computer game. In *Proc. SIGCHI Conf. on Human Factors in Computing Systems (CHI-04)*, 319–326. New York, NY, USA: ACM Press.