# Learning to Tag and Tagging to Learn: A Case Study on Wikipedia

**Peter Mika, Massimiliano Ciaramita, Hugo Zaragoza, and Jordi Atserias,**
*Yahoo! Research*

***The problem of semantically annotating Wikipedia inspires a novel method for dealing with domain and task adaptation of semantic taggers in cases where parallel text and metadata are available.***

Natural language technologies will play an important role in the Web's future. Recent Web developments, such as the huge success of Web 2.0, demonstrate annotated data's great potential. However, when it comes to annotating documents even at the most primitive levels, human effort alone can't scale to the Web. Recently, the focus in the Semantic Web has shifted from text to user-supplied explicit annotations. We believe, though, that the Semantic Web community should develop these two visions in parallel. The Web 2.0 phenomenon brought renewed energy to annotating Web content, as large-scale tagging, adopting microformats, and introducing a data-structuring mechanism in textual sources such as Wikipedia exemplified. At the same time, although databases generate most Web pages (as much as 80 percent, according to estimates), many of those databases store significant amounts of text devoid of machine-processable semantics. This quality is particularly true for one of the most exciting fractions of Web content: user-generated content. These text contributions populate blogs, wikis, social networks, and social media sites such as Yahoo! Answers, YouTube, and Flickr.

Unfortunately, providing natural language support for the Web requires addressing one of the most challenging tasks in natural language processing (NLP): model and task adaptation. First, models trained on one source typically perform much more poorly on other sources. Training data for machine-learning semantic annotation is limited to a few public data sets, almost exclusively news corpora. Acquiring new training data is often prohibitively costly, so adaptation is often the only solution. Second, a mismatch often exists between various tasks' requirements, such as a mismatch in the semantics users want to extract. For example, an entity tagger trained on news corpora can recognize person names in general, but a task-specific application might require recognizing musical artists, or vice versa. Often, the two problems compound: users must process text without training data, and they might be interested in entities different from those our tagger was trained to recognize.

An interesting question, then, is how to leverage existing human effort in annotating user-generated content to provide improved support for machine annotation of the remaining content. The example we consider is Wikipedia. As a source of generic knowledge covering a range of domains, Wikipedia is one of the most important collections of user-generated content (involving a significant fraction of Web searches). In Wikipedia—technically, a


1541-1672/08/$25.00 © 2008 IEEE
*Published by the IEEE Computer Society*

## Related Work in Model Adaptation

Machine-learning-based natural language processing (NLP) approaches rely on the availability of high-quality and costly training data for the particular domain and annotation task at hand. The alternative to acquiring new training data is adapting semantic annotation models from one source with available training data to another where training data is unavailable or scarce.

Model adaptation is still a critical research challenge, particularly when moving from narrower to broader domains, such as moving from news corpora to Wikipedia or the Web. One main approach is *self-training*, which adds automatically annotated data from the target domain to the original training data.[1,2] A second key approach is *structural correspondence learning*, which tries to build a shared feature representation of the data.[3,4] Recently several works have explored the use of Wikipedia in named entity recognition (NER). Jun'ichi Kazama and Kentaro Torisawa propose extracting category labels, of the is-a kind, from definition sentences in Wikipedia articles to use as features in NER systems.[5] An example would be to extract the word *painter* from the first sentence of Pablo Picasso's article. The authors show that these features can improve in-domain NER. Yotaro Watanabe, Masayuki Asahara, and Yuji Matsumoto used anchor text in HTML links to build a graph they used to disambiguate by means of a Conditional Random Field model.[6]

We've also recently seen the first works that exploit or extend the information in Wikipedia available as metadata. In particular, Fei Wu and Daniel S. Weld's Kylin system starts with the same idea that we apply in our work: establishing a correspondence of text and metadata.[7] They use the generated corpus to learn how to extract properties' values from Wikipedia articles. In other words, their task is to automatically fill infoboxes with the correct values. This is an interesting task that produces knowledge that complements the way we enrich DBpedia. As a necessary step in their approach, they also perform document classification based on simple heuristics they apply to Wikipedia category pages.

The work of Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum on YAGO (Yet Another Great Ontology) extends the idea of applying heuristics to extract information from proprietary aspects of Wikipedia.[8] In particular, it targets categories, disambiguation pages, and redirections. Their work also shares the goal of providing a firm class hierarchy for Wikipedia by categorizing resources according to an external (linguistic) ontology, in their case WordNet. The kind of output is thus similar to ours. However, compared to the NLP approach we take, these heuristics seem to provide high precision but low recall. Furthermore, we believe that heuristic-based approaches are inherently overfitting the Wikipedia case. The learning-based approach we present is much less dependent on the particular corpus and the proprietary aspects of Wikipedia articles.

### References


1. M. Bacchiani et al., "Map Adaptation of Stochastic Grammars," *Computer Speech and Language*, vol. 20, no. 1, 2006, pp. 41–68.
2. D. McClosky, E. Charniak, and M. Johnson, "Reranking and Self-Training for Parser Adaptation," *Proc. 21st Int'l Conf. Computational Linguistics and 44th Ann. Meeting Assoc. for Computational Linguistics* (Coling-ACL 06), Assoc. for Computational Linguistics, 2006, pp. 337–344.
3. J. Blitzer, M. Dredze, and F. Pereira, "Biographies, Bollywood, Boom-Boxes, and Blenders: Domain Adaptation for Sentiment Classification," *Proc. 45th Ann. Meeting Assoc. for Computational Linguistics* (ACL 07), Assoc. for Computational Linguistics, 2007, pp. 440–447.
4. J. Blitzer, R. McDonald, and F. Pereira, "Domain Adaptation with Structural Correspondence Learning," *Proc. Conf. Empirical Methods for Natural Language Processing* (Emnlp 06), Assoc. for Computational Linguistics, 2006, pp. 120–128.
5. J. Kazama and K. Torisawa, "Exploiting Wikipedia as External Knowledge for Named Entity Recognition," *Proc. Conf. Empirical Methods for Natural Language Processing* (Emnlp 07), Assoc. for Computational Linguistics, 2007, pp. 698–707.
6. Y. Watanabe, M. Asahara, and Y. Matsumoto, "A Graph-Based Approach to Named Entity Categorization in Wikipedia Using Conditional Random Fields," *Proc. Conf. Empirical Methods for Natural Language Processing* (Emnlp 07), Assoc. for Computational Linguistics, 2007, pp. 649–657.
7. F. Wu and D.S. Weld, "Autonomously Semantifying Wikipedia," *Proc. 16th ACM Conf. Information and Knowledge Management* (CIKM 07), ACM Press, 2007, pp. 41–50.
8. F.M. Suchanek, G. Kasneci, and G. Weikum, "YAGO—A Core of Semantic Knowledge," *Proc. 16th Int'l Conf. World Wide Web* (WWW 07), ACM Press, 2007, pp. 697–706.


database—the amount of content locked in the text of articles significantly overshadows the amount of structured content, despite the effort concentrated in the project.

In this article, we investigate how to use standard named-entity recognition (NER) technology to significantly enrich the metadata available in Wikipedia. By using this knowledge, we also examine how to generate additional training data to improve NER technology without additional human intervention. (See the "Related Work in Model Adaptation" sidebar for more additional approaches to this research problem.)

### An overview

Our basic approach is based on linking Wikipedia's text to the structured knowledge found in the project's *infoboxes*, consistently formatted tables that provide summary information. Figure 1 (see p. 28) illustrates the process. First, we annotate the Wikipedia collection using an off-the-shelf NER tool trained on a standard corpus. (We use the word *annotation* interchangeably with [semantic] *tagging*, which the NLP community more commonly uses with much the same meaning.) Next, we link these semantic annotations with the structured knowledge the DBpedia project (www.dbpedia.org) makes available. This analysis provides a mapping between the semantic tagger's broad categories and the DBpedia collection's much more refined metadata vocabulary. In particular, we enrich DBpedia with additional class hierarchies, type information for resources, and range restrictions for properties. This mapping is a versatile resource that we expect to provide valuable background knowledge for many intelligent applications built with DBpedia. We then apply this mapping to the corpus, thereby generating additional

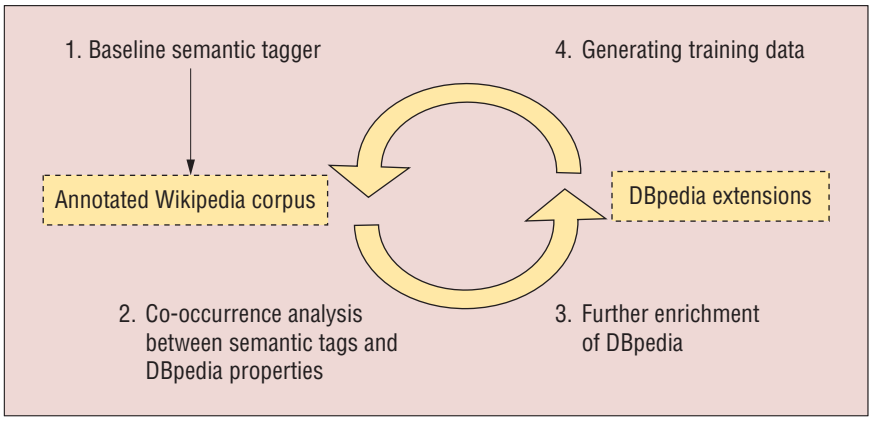


**Figure 1. The enrichment process. Increasing the level of semantics in Wikipedia and improving semantic tagging are the dual outcomes of our approach.**

training sentences for our semantic tagger, this time using sentences from the Wikipedia collection. This step, in turn, improves our original tagger on Wikipedia. The completely automated process is repeatable with the newly obtained corpus in a learning loop, which we plan to evaluate as part of future work.

An improvement over the original version, the semantically annotated Wikipedia corpus can enhance tagging in other nonspecialist domains. Moreover, the corpus itself is a valuable source of knowledge, as Wikipedia's text contains numerous entities that lack corresponding Wikipedia articles (and thus aren't included in DBpedia). For this reason, we make publicly available both our alignment and the semantically annotated Wikipedia corpus on the Web at www.yr-bcn.es/semanticWikipedia.

## Semantic annotation of Wikipedia

Semantic annotation aims to discover all occurrences of the entity classes of interest for an application in a given natural text. To annotate Wikipedia with entity labels, we used a tagger[1] that implements a hidden Markov model (HMM), which is a statistical model of sequential structures. The HMM is trained with the average sequence perceptron algorithm.[2] The tagger uses a generic feature set for NER based on words, lemmas, part-of-speech (PoS) tags, and word shape features. We generated the PoS annotations with the same tagger trained on the *Wall Street Journal* Penn Treebank.[3]

In addition to the choice of machine learning algorithm, a critical issue in learning-based semantic annotation is acquiring appropriate training data. Specifically, the data must comprise sentences completely and consistently annotated with the entity labels an application requires.

We trained the base tagger that we used in these experiments on the 2003 Conference on Natural Language Learning (CoNLL) English NER data set.[4] The set consists of 20,744 English sentences from Reuters data. This corpus is annotated with four category labels that make up our CoNLL tagger's *vocabulary*: PERSON, LOCATION, ORGANIZATION, and MISCELLANEOUS. (Here, we refer to the entity classes a tagger is trained to recognize as the tagger's vocabulary. In a learning-based approach, the vocabulary is a characteristic of the data set and not the tagger itself.) We've also applied our method using the *Wall Street Journal* financial-news collection's training data, which are annotated with 108 hierarchically organized categories. However, we performed our experiments using the smaller CoNLL tag set: increasing the number of entity classes raises the complexity level for performing evaluation with human experts. (The number of choices to consider for marking up a particular term increases with vocabulary size.) Furthermore, agreement among annotators becomes more difficult to maintain as the number of arguably correct choices increases.

The two data sets we mention here are typical of the widely available data sources for learning-based NER: they're focused on a relatively narrow domain (political and financial news), but they offer high-quality annotations in terms of completeness, correctness, and consistency of vocabulary use. The domain's limited scope and the annotations' quality mean it's possible to achieve very good results when training and testing on held-out fractions of the same corpus. For example, the accuracy of the tagger evaluated on held-out CoNLL data is approximately 91 percent F-score (the harmonic mean of precision and recall). The tagger implements an efficient decoding algorithm based on dynamic programming, and its complexity is linear in the length of the sentence. We can therefore use the tagger to tag large amounts of data efficiently. Table 1 shows an output sample, which is in the multitag format NLP tools commonly use. To represent annotations spanning multiple terms, the tags have either a B (beginning of an annotation) or an I (continuation of an annotation) prefix. Our baseline tagger created the first three annotation columns, and our enrichment tool added the last column. The tagger is publicly available at http://sourceforge.net/projects/supersensetag.

## Wikipedia adaptation and metadata enrichment

Our approach to Wikipedia adaptation relies on mapping our existing tagger's annotation vocabulary to Wikipedia's more fine-grained system of templates. Templates provide the schema for a set of infoboxes. The DBpedia project provided template information and other metadata. Using our alignment between the annotation vocabulary and DBpedia templates, we'll also be able to significantly extend the metadata currently extractable from Wikipedia.

To simplify this discussion, we'll use the CoNLL tagger's vocabulary as an example ($V_{conll}$). However, the automated method we'll describe is applicable to any annotation vocabulary.

### Aligning annotation vocabularies and DBpedia properties

DBpedia, which is available for download in RDF, is a lightweight ontology consisting of a straightforward extraction of the information in Wikipedia infoboxes. In DBpedia, a resource represents each page—the underlying assumption being that each Wikipedia page represents a unique entity. (Resources aren't instances of the templates, but rather are related to the templates through the wikiPageUsesTemplate property.) It's merely the entry's name, the template classes used, and the template properties' values that describe an instance. Although the amount of information is significant (close to 100 million triples), important ontological knowledge is missing. In particular, the data set lacks range restriction on properties (rdfs:range);

**Table 1. Semantic annotation example.***

| Token | Part-of-speech | CoNLL | *Wall Street Journal* | Wikipedia |
|---|---|---|---|---|
| Pablo | NNP | B-PERSON | B-E:PERSON | B-persondata_name |
| Picasso | NNP | I-PERSON | I-E:PERSON | I-persondata_name |
| was | VBD | 0 | 0 | 0 |
| born | VBN | 0 | 0 | 0 |
| in | IN | 0 | 0 | 0 |
| Málaga | NN | B-LOCATION | B-E:GPE:CITY | B-persondata-placeOfBirth |
| , | , | 0 | 0 | 0 |
| Spain | NNP | B-LOCATION | B-E:GPE:COUNTRY | B-persondata-placeOfBirth |

* Each column provides annotation for the token according to the given tag sets.

there's no information on the type of values a given property can take.

To make it easier to follow the discussion, we'll introduce the notations $O_i$, $O_p$, and $O_t$ for the sets of instances, properties, and template classes in DBpedia, respectively. We begin our work with the key idea of exploiting parallel corpora of metadata and textual information. In our specific case, we have the DBpedia data set on one hand and the text of the Wikipedia articles on the other. As Figure 2 shows, metadata and text provide two parallel descriptions: the article and the corresponding metadata describe aspects of the same real-world object (in this case, Picasso), but they're kept separately. For example, Picasso's birthplace (Málaga, Spain) is given both in text and in the metadata.

This simple observation gives rise to the idea of annotating the text with the metadata or, more specifically, creating a corpus annotated with DBpedia properties in $O_p$. Our tool achieves this by processing Wikipedia's XML corpus on a per-article basis, looking up for every article the resource in $O_i$ being described. Next, the tool iterates over all of the resource's literal-valued properties, plus the labels of resources connected to the current resource through a resource-valued property. (No separation exists between data type and object properties in DBpedia, which is one reason the ontology isn't OWL compliant.) In the example, the birthplace is given as a link to two resources: one representing the article on Málaga and the other the article on Spain. Next, the tool annotates all occurrences of these values with the name of the corresponding property, in this case, **persondata_placeofbirth**. If a word is part of multiple

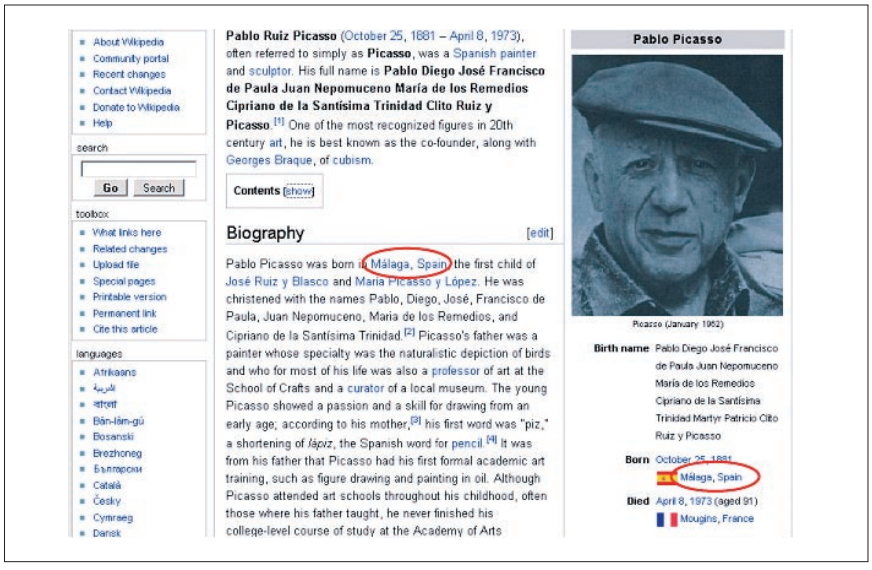


**Figure 2. Wikipedia page on Pablo Picasso. The same information (regarding Picasso's birthplace, in this case) appears in both the text and an infobox. We exploit this feature in our work.**

annotations (nested annotations), we keep only the outermost (longest) annotation.

The results of annotation merge with the semantic tagger's outputs to produce a multitag result format (see Table 1), where the annotations from our tool show up as an additional column. This data set (also available for downloading at www.yr-bcn.es/semanticWikipedia) is useful on its own—for example, to perform learning on Wikipedia properties, as Fei Wu and Daniel S. Weld describe.[5]

The multitag format is also the input for the next processing step, in which we compute the co-occurrence among tags in different tag sets. We analyze co-occurrence among tags on the same terms—that is, co-occurrence of values within the rows of the multitag file (see Table 1). This step's outcome is a co-occurrence graph, where the nodes are tags, and the edges between them represent the association's strength as measured by the number of co-occurrences. The output is in the format of the Pajek network analysis package, which we can use to further process and visualize these graphs (http://vlado.fmf.uni-lj.si/pub/networks/pajek). Next, we extract mappings from this graph as we did in previous work, where we analyzed a co-occurrence network of tags in a folksonomy aiming to find broader and narrower relationships.[6] In our previous work, we relaxed the set subsumption notion by requiring only a significant overlap among the two sets. However, we also applied the constraint that the smaller set is smaller by at least a given factor to exclude cases where the two sets are of similar size.

**Table 2. Mapping the CoNLL tag set to Wikipedia properties.**

| Wikipedia property | CoNLL tag | First 10 property values | Accuracy |
|---|---|---|---|
| infoboxNbaPlayer_name | PERSON | Alex Groza, Elgin Baylor, Jerry West, David Thompson, Glen Rice, Christian Laettner, Richard Hamilton, Juan Dixon, Sean May, Joakim Noah | Correct |
| infoboxSerialKiller_alias | ORGANIZATION | Gray Man, the Werewolf of Wysteria, Brooklyn Vampire, Sister, Brian Stewart, Bloody Benders, The ProstishooterThe Rest Stop KillerThe Truck Stop Killer, The Sunset Strip Killer, Cincinnati Strangler, Son of Sam, Plainfield Ghoul, Ed "Psycho" Gein, The Co-ed Killer | Incorrect |
| cluster_name | ORGANIZATION | AM-2, Christmas Tree Cluster, Coma Star Cluster, Double Cluster, Jewel Box, NGC 581, Messier 18, Messier 21, Messier 25, Messier 26 | Incorrect |
| highlanderCharacter_born | LOCATION | Unknown, unknown, 1659, 1945, 802, 1887, 1950, Glenfinnan, Scotland, (original birth date unknown) ("Highlander II"), 896 BC, Ancient Egypt (original birth date unknown) ("Highlander II"), California | Partially correct |
| infoboxSuperbowl_stadium | LOCATION | Sun Devil Stadium, Georgia Dome, Miami Orange Bowl, Hubert H. Humphrey Metrodome, Dolphin Stadium, Raymond James Stadium, Louisiana Superdome, Joe Robbie Stadium, Ford Field, Los Angeles Memorial Coliseum | Correct |
| infoboxNationalFootballTeam_name | LOCATION | Netherlands, Italy, Israel, United States, Mexico, Russia, Cuba, United Kingdom, Burkina Faso, France | Correct |
| infoboxWeapon_usedBy | LOCATION | USA, none, one, none, Italy, United States, Mexico, UK, Russia, Under development | Correct |
| minorLeagueTeam_league | MISCELLANEOUS | Eastern League (1923–37, 1940–63, 1967–68, 1992–), Pacific Coast League, Arizona League, Texas League, South Atlantic League, California League, Midwest League, Northwest League, International League, Carolina League | Correct |
| infoboxTea_teaOrigin | LOCATION | Nuwara Eliya, Sri Lanka near Adam's Peak between 2,200–2,500 metres, Japan, India, Vietnam, Taiwan, Turkey, China, Anhui, Guangdong, Jiangxi | Correct |
| infoboxPrimeMinister_name | PERSON | Abdallah El-Yafi, Umar al-Muntasir, Dr. Abdellatif Filali, Abderrahmane Youssoufi, Abdessalam Jalloud, Abdul Ati al-Obeidi, Abdul Hamid al-Bakkoush, Abdul Majid Kubar, Abdul Majid al-Qaud, Abd al-Qadir al-Badri | Correct |

Therefore, we can speak of similarity instead of subsumption. Furthermore, to compute support, we considered the broader set's size.

In our current work, we adapt these measures because the goal is slightly different: we want to find at most a single mapping for every tag, and we aren't concerned whether we find a similarity or a subsumption relationship. When finding a mapping for tag $A$, we're looking at how much a given tag $B$ covers $A$ relative to what any other tag covers. In other words, how much of the part that the tag set covers is also covered by that particular tag? Given a threshold $n$, we include in the results a mapping between sets $A$ and $B$ if

$$\frac{|A \cap B|}{\sum_i |A \cap B_i|} > n.$$

Similarly, when measuring support, we compute how much of $A$ any tag in the tag set covers—that is, for a given threshold $k$ we require that

$$\frac{\sum_i |A \cap B_i|}{|A|} > k.$$

Lastly, we require that $A$'s absolute size is beyond a threshold $m$, or $|A| > m$.

As an example, the tag born might have 100 occurrences, being considered a LOCATION in 70 percent of the cases, MISCELLANEOUS in 20 percent, and not tagged in 10 percent. In this case, we consider mapping born to LOCATION under three conditions:

- if the ratio of locations to all tags (70/90) is greater than the first threshold parameter,
- if the number of co-occurring instances (90/100) is greater than the second threshold parameter, and
- if the number of occurrences (100) is greater than the third threshold.

If the first threshold is greater than 0.5, we'll always have a single mapping.

In Table 2, we list the results of mapping the simple CoNLL tag set to Wikipedia properties, using hand-picked parameters ($n > 0.8$, $k > 0.8$, $m > 25$). As a reminder, these mappings provide a type for property ranges, a crucial piece of knowledge missing in DBpedia. Although quality is generally high, some of the mappings suffer from the liberties Wikipedia affords. In particular, when filling in templates, users are unconstrained in the values they can provide. The resulting heterogeneity goes mostly unnoticed because the values serve only display purposes.

Immediately apparent, for example, is that Wikipedia users sometimes put more information than necessary and often enter text just to indicate that a property is not applicable or unknown. Furthermore, many properties suffer from ambiguity. For example, for highlanderCharacter_born, some users filled in locations, while others added

dates. In such cases, even with the simple vocabulary that we have, we can't assign a single range restriction to the property.

Finding the optimal parameter settings, and thus the optimal mapping, depends on the task at hand. The optimal mapping depends on the kind of costs associated with false positives and false negatives.

## DBpedia enrichment

We can also apply the co-occurrence analysis between other columns of the multitag format. In particular, we can find mappings between terms or phrases and the annotation vocabulary. This provides a mechanism to extract those entities from Wikipedia that are consistently tagged with the same annotation class. In other words, we filter out terms and phrases that are ambiguous or aren't always recognized as entities. Through this process, we again obtain significant new knowledge compared to Wikipedia. Although many consider Wikipedia complete in some sense, its articles contain significant numbers of entities that don't yet have a Wikipedia page. In the case of the Picasso article, this includes the names of people who were important in Picasso's life, schools he attended, and even the name of a journalist who interviewed Picasso.

Furthermore, we also obtain type information for DBpedia instances ($O_i$) by mapping entities to DBpedia instances with the same label. This acquisition obviously results in noisy information, as a step of disambiguation would be necessary to make sure that the entity the text mentions and the article's subject are the same. However, we can skip this task for our purposes and use this knowledge in a statistical fashion in the next processing step.

Given this—somewhat noisy—classification of instances (articles) according to the annotation types, we can now try to align Wikipedia templates ($O_t$) and our annotation vocabulary. We can look at this as a case of instance-based ontology mapping:

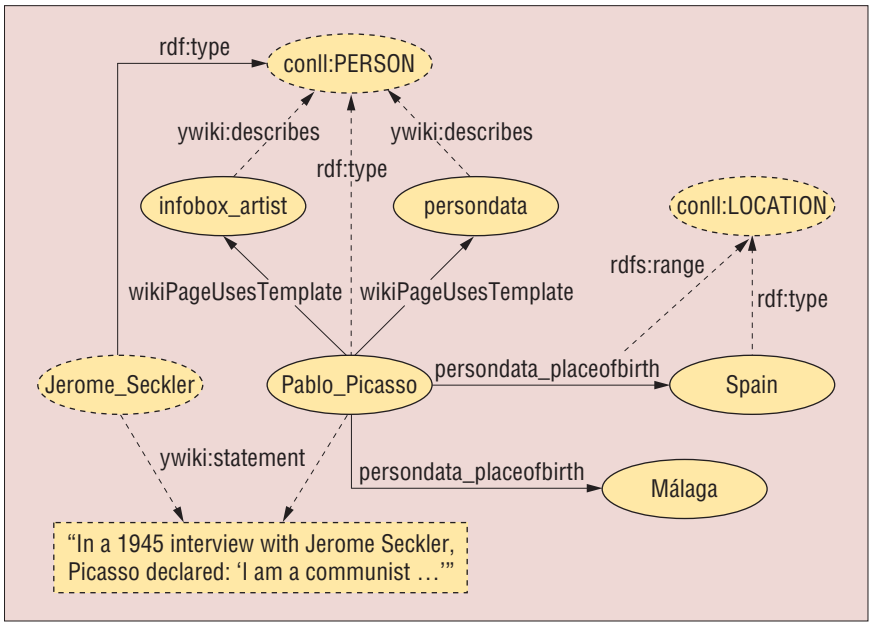


**Figure 3. Co-occurrence analysis. This flowchart is an example of knowledge gained through co-occurrence analysis and further processing the results. Resources with solid outlines were part of DBpedia's original description of the Picasso resource. Resources with dashed outlines were added to the ontology through our method.**

given a number of articles classified using both templates and annotation classes, we can now derive mappings between the two ontologies. Again, what we gain is significant new knowledge in the form of an "upper ontology" for Wikipedia templates. (Because DBpedia doesn't represent templates as classes, we opted not to represent this information using the rdfs:subClassOf relationship. Rather, we introduce a new instance relationship linking templates to the types they describe.) Using our Picasso example again, Figure 3 shows how our tool can enrich the DBpedia ontology.

Instance classification is an important input, among others, for merging template information. For example, we can now find all the templates describing people in Wikipedia who are obvious candidates for merging (at least the basic properties relating to people). Such knowledge is useful for building simplified browsing and search interfaces for Wikipedia and also for performing question answering on the Wikipedia corpus.

Our mapping has another important characteristic, namely that minimal human effort can significantly improve information quality. By manually revising the mapping for only the most commonly occurring templates, we can effectively revise the classification of a significant number of resources.

Table 3 shows statistics on statements we extracted with reasonable, but handpicked, settings for the parameters. Although the distribution is typical, the counts depend primarily on the chosen thresholds (here, $n > 0.8$, $k > 0.8$, $m > 25$). Also, the numbers don't include inferred statements.

## Generating training data and retagging Wikipedia

We apply the knowledge we learned in the

**Table 3. The number of statements extracted through mapping.**

| | PERSON | ORGANIZATION | LOCATION | MISCELLANEOUS | Total |
|---|---|---|---|---|---|
| rdfs:range | 805 | 515 | 480 | 124 | 1,924 |
| ywiki:describes | 184 | 18 | 13 | 4 | 219 |
| rdf:type (from text) | 285,873 | 140,885 | 46,079 | 32,550 | 505,387 |
| rdf:type (through rdfs:range) | 307,874 | 58,768 | 31,383 | 8,108 | 406,133 |
| rdf:type (through ywiki:describes) | 47,702 | 489 | 611 | 74 | 48,876 |

Table 4. Tagger model performance.

| Model | Sentences labeled Ok (%) | Relative increase over the baseline model (%) | *p* | Sentences tagged identically as the baseline model (%) |
|---|---|---|---|---|
| CoNLL (baseline) | .660 | — | — | — |
| Wikipedia | .584 | –11 | 1E-5 | 47 |
| Wikipedia-B | .583 | –12 | 1E-6 | 47 |
| Mixed | .668 | +1.3 | .956 | 69 |
| Mixed-B | .696 | +5.5 | .003 | 74 |
| Spain | NNP | B-LOCATION | B-E:GPE:COUNTRY | B-persondata-placeOfBirth |

previous section to improve the baseline semantic tagger. The key idea is to use the alignment between the properties in $O_p$ and the annotation vocabulary $V_{conll}$ to generate new, in-domain training data. For example, in the following statement, we annotated "North Brabant" with the property infobox_city_subdivisionname from $O_p$: Boxmeer, is a village in the Netherlands in the province of [*North Brabant*].

Because we perform this annotation on a per-article basis, we can avoid ambiguity. Although common words might change meaning within an article, it's unlikely to be the case with named entities.

From the learned mapping, we can identify this mention as an instance of LOCATION in $V_{conll}$. The box identifies the extent of the data we can use directly for training. The remaining terms in the sentence might also be entities that we have no knowledge of—in fact, the sentence has two other location mentions. A limitation of this approach is that it identifies only positive examples, while entities are a minority of the terms in the text. To generate better training data, we built a "nostop" list of words that are tagged as nonentity labels in the original gold-standard training. This step extends the context for training around the identified entity as long as the words are in the nostop list. So, in our previous example, we would extend the box as follows: Boxmeer, is a village in the Netherlands [in the province of *North Brabant*.]

Where "North Brabant" is tagged as LOCATION and all other words as 0, we call these stretches of text *fragments* because they're longer than entities but not necessarily sentences.

The set of training sentences we generate are not a random sample of Wikipedia text. The distribution of entity categories can be considerably skewed. As a result, we can overgenerate specific categories while tagging. To alleviate this problem, we attempted a simple solution of ranking the training sentences on the basis of the number of terms with 0 labels. The logic is that these instances have more context and are more sentence-like. We then selected numerous training instances for each category in such a way that they yield a category distribution similar to the training data. One disadvantage of such a scheme is that we won't use part of the Wikipedia training data for training, which leaves room to devise better solutions.

## Evaluation

Evaluating semantic annotations is a difficult and expensive process, traditionally involving experts who read the text and annotate it by hand. This process means identifying the entity boundaries and their type. Many semantic controversies can arise, even for simple tag sets, which can lead to low interannotator agreement and slow annotation speeds. Moreover, tagging the most frequent tags often dominates evaluation efforts. Unfortunately, these tags are often the easiest to locate automatically and, therefore, the least interesting.

For this investigation, we developed an evaluation framework to implement in practice, in a short time, and with few resources. We asked human judges to mark the overall quality of a tagged sentence, rather than annotate the sentences with tags. The feedback is a binary decision: the sentence is judged as correct (Ok) if it doesn't contain any mistakes; otherwise, it's judged incorrect (Bad). We allowed a third outcome if the text was unintelligible (Unsure). Given that tagging is to some degree subjective, we asked several judges to look at each sentence and used their judgments as votes, taking the majority label vote as true (we removed from the evaluation sentences resulting in ties).

We evaluate our work by considering the CoNLL tagger as a baseline. We used a fully automated method combining automatic tagging (using the baseline model) and co-occurrence analysis.

Our judges evaluated sentences sampled at random from Wikipedia. To accomplish this, we randomly sorted all articles (disregarding entries used for training), then we sampled approximately one in every 100 sentences. This process resulted in 989 sentences total, which we tagged with the baseline and evaluated. Next, the adapted models, which we'll describe in detail in a moment, tagged the data. Our judges then evaluated them again (except for sentences that hadn't changed). Altogether, we used five human judges, with at least two judges (2.4 on average) evaluating each sentence. The judges saw sentences only once, except when there was a difference between the models.

The process produced a total of 6,956 judgments, averaging 1,400 clicks per judge and roughly 20 evaluation staff-hours. Of the 989 sentences, 200 received an Unsure tag by at least one judge and were discarded, leaving 789 sentences for evaluation. We evaluated these sentences for each model, removing sentences (for that model only) that resulted in tied judgments.

Table 4 reports the models' performance. First, consider the CoNLL tagger, used out-of-the-box without any form of adaptation to Wikipedia. This model tagged 66 percent of sentences accurately, a result much lower than the performance it had with news stories (above 80 percent). Nevertheless, 66 percent is still high and shows that state-of-the-art taggers might be successful on corpora such as Wikipedia. Worth noting is that the 34 percent tagged incorrectly contain many correct entities, but a single incorrect entity is sufficient to deem a whole sentence Bad.

The performance of the adapted Wikipedia model is 58 percent, which is 7.6 percent worse than the baseline in absolute terms

(11 percent worse in relative terms). We computed the *p* value using the two-sided, paired Wilcoxon signed rank test, as implemented in *R*. (See http://sekhon.berkeley.edu/stats/html/wilcox.test.html for more information.) Furthermore, the Wikipedia model tagged 47 percent of sentences identically as the baseline had tagged them. Considering that the data used to train this model comes from Wikipedia and infoboxes alone, this performance is remarkable. We also note that this performance is a lower bound to the performance that we could obtain if a mapping of types (from templates tags to tagger tags) were available. Indeed, the Wikipedia model's quality depends strongly on the quality of the mapping used. We must translate the many idiosyncratic templates into a small unified set of types of interest.

The type of training data infoboxes generate is biased toward the types of templates available. For example, it generates many more PERSON names than ORGANIZATIONS (see Table 2). We attempted to correct for this by resampling the training instances with respect to the more unbiased CoNLL distribution: model *Wikipedia-B* in Table 4. This model's performance is similar to that of the unbalanced Wikipedia training set. We hypothesize that the distribution is so skewed that it can't easily be made unbiased by removing instances; it requires data from the underrepresented types, which is simply unavailable. For this reason, we experimented next with combining the Wikipedia data with wide-coverage data, in particular the CoNLL collection.

The *Mixed* model combines the human-annotated CoNLL collection with the Wikipedia collection by concatenating the training sets. Unfortunately, its performance is roughly that of the original CoNLL model. The difference in percentage of sentences labeled Ok is only 1.3 percent relative, which isn't statistically significant. As expected, the number of sentences tagged identically to the CoNLL baseline increases (to 69 percent). We hypothesize that the lack of performance increase is because the two distributions being combined are too different, a typical domain adaptation problem in NLP. To test this theory, we carried out the same resampling experiment, leading to the *Mixed-B* experiment. Indeed, this model's performance is significantly better than the baseline: 5.5 percent relative, reaching 70 percent accuracy at the sentence level. It's interesting to note that the number of identical sentences is even higher than for the Mixed model, meaning that the Mixed-B model changed fewer sentences, but correctly. This result proves that correcting for entity distribution is important and needs more investigation. Furthermore, it shows that combining hand-labeled collections with user-generated content can increase tagging performance.

Our investigation of enriching the DBpedia metadata collection through the use of an NLP tagger and statistical analysis provided significant information. The results undoubtedly will be useful for many Wikipedia-specific tasks, such as mapping templates, cleaning up infobox data, and providing better searching and browsing experiences. Because Wikipedia's domain is broad, we expect that our data sets will serve as useful background knowledge in other applications. For example, we've shown how to apply the data toward the problem of improving our baseline tagger used for semantic annotation. This application closes the loop in Figure 1 and creates a feedback cycle that we plan to explore in the future.

Our targets for technical improvements include better sentence detection, more precise tagging with Wikipedia metadata, and better balancing of the training data. We also plan to experiment with applying minimal manual effort in cleaning up the mapping to observe how much we gain compared to a completely automated approach. Finally, we see the possibility of generalizing our approach to other situations with semantic annotations or where parallel text and metadata are available. In particular, Web pages annotated with microformats or Resource Description Framework attributes (RDFa) would provide an interesting testing ground with a larger scale, but certainly noisier metadata, than Wikipedia. ■

## The Authors


**Peter Mika** is a researcher at Yahoo! Research. His research interests include the field of social networks and the Semantic Web. Mika received his PhD in computer science from Vrije Universiteit Amsterdam. Contact him at pmika@yahoo-inc.com.

**Massimiliano Ciaramita** is a researcher at Yahoo! Research. His research interests include natural language processing and information retrieval. Ciaramita received his PhD from Brown University. Contact him at massi@yahoo-inc.com.

**Hugo Zaragoza** is a senior researcher at Yahoo! Research. His research interests include applications of machine learning and natural language processing for information retrieval. Zaragoza received his PhD from the University of Paris 6. He's a member of the ACM Special Interest Group on Information Retrieval. Contact him at hugoz@yahoo-inc.com.

**Jordi Atserias** is a research engineer at Yahoo! Research. His research interests include natural language processing, primarily morphological analysis, parsing, word sense disambiguation, semantic role labeling, and information extraction. Atserias received his PhD in computer science from the University of the Basque Country. Contact him at jordi@yahoo-inc.com.


## References


1. M. Ciaramita and Y. Altun, "Broad-Coverage Sense Disambiguation and Information Extraction with a Supersense Sequence Tagger," *Proc. Conf. Empirical Methods Natural Language Processing* (EMNLP 06), Assoc. for Computational Linguistics, 2006, pp. 594–602.
2. M. Collins, "Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms," *Proc. Conf. Empirical Methods Natural Language Processing* (EMNLP 02), Assoc. for Computational Linguistics, 2002, pp. 1–8.
3. M.P. Marcus, B. Santorini, and M.A. Marcinkiewicz, "Building a Large Annotated Corpus of English: The Penn Treebank," *Computational Linguistics*, vol. 19, no. 2, 1993, pp. 313–330.
4. E.F. Tjong Kim Sang and F. De Muelder, "Introduction to the CoNLL2003 Shared Task: Language-Independent Named Entity Recognition," *Proc. 7th Conf. Natural Language Learning* (CoNLL 03), 2003, pp. 142–147.
5. F. Wu and D.S. Weld, "Autonomously Semantifying Wikipedia," *Proc. 16th ACM Conf. Information and Knowledge Management* (CIKM 07), ACM Press, 2007, pp. 41–50.
6. P. Mika, "Ontologies Are Us: A Unified Model of Social Networks and Semantics," *J. Web Semantics*, vol. 5, no. 1, 2007, pp. 5–15.