Learning to tag and tagging to learn: A case study on Wikipedia

Peter Mika Massimiliano Ciaramita Hugo Zaragoza
Jordi Atserias
Yahoo! Research
Ocata 1, 08003 Barcelona, Spain
pmika,massi,hugoz,jordi@yahoo-inc.com

June 3, 2008

Abstract

Natural language technologies have been long envisioned to play a crucial role in transitioning from the current Web to a more "semantic" Web. If anything, the significance of textual content on the Web has only increased with the rise of Web 2.0 and mass participation in content generation, which comes mostly in the form of text. Yet, natural language technologies face significant challenges in dealing with the heterogeneity of Web content: specifically, the accuracy of systems trained on one corpus for a specific task degrades considerably when either the domain or task changes. In this paper, we consider the problem of semantically annotating Wikipedia. We investigate a method for dealing with domain and task adaptation of semantic taggers in cases where parallel text and metadata are available. By creating a semantic mapping among vocabularies from two sources: Wikipedia and the original annotated corpus, we are able to improve our tagger on the Wikipedia. Moreover, by applying our tagger and mapping between sources we are able to significantly extend the metadata currently available in the DBpedia collection.

1 Introduction

Natural Language technologies are expected to play an important role on the future of the Web. Recent developments such as the huge success of Web 2.0 demonstrate the great potential of annotated data, but human effort alone can not scale to the Web when it comes to annotating documents even at the most primitive levels. Recently the focus within the Semantic Web has shifted away from text and concentrated on explicit annotations provided by users. We believe, however, that these two visions should be developed in parallel. First, we are encouraged by the renewed energy that the Web 2.0 phenomenon brought to annotating content on the Web, as exemplified by large-scale tagging, the

adoption of microformats and the introduction of data structuring mechanism in textual sources such as Wikipedia. Second, while it is true that the majority of Web pages (according to estimates, as much as 80%) is generated from databases, one should not forget that many of those databases store significant amounts of text devoid of machine processable semantics. This is in particular true for one of the most exciting fraction of the Web's content: user-generated content (UGC), the kind of text contributions that populate blogs, wikis, social networks and social media sites such as Yahoo! Answers, YouTube, Flickr, etc.

Unfortunately, providing natural language support for the Web requires addressing one of the most challenging tasks in NLP today, that is *model and task adaptation*. The first problem is that models trained on one source typically under-perform on other, especially if noisier, sources. Training data for machine learning of semantic annotation is limited to a few public datasets, almost exclusively news corpora. Acquiring new training data is often prohibitively costly and thus the problem needs to be solved by implementing some sort of adaptation strategy, a difficult problem which has only recently become the object of systematic investigation. Second, there is typically a mismatch between the requirements of various tasks. A common problem is a mismatch in the semantics one is looking to extract. For example, while an entity tagger trained on news corpora is able to recognize person names in general, in a task-specific application we may need to recognize musical artists, or the other way around. Typically, the two problems compound: we need to process text with no training data and at the same time we might be interested in entities different from the ones our tagger was trained to recognize.

One interesting question then is how to leverage existing human effort in the annotation of user-generated content to provide improved support for machine annotation of the remaining content. The example we will consider in this paper is that of Wikipedia. As a source of generic knowledge covering a range of domains, Wikipedia is one of the most important collections of user-generated content[1]. In Wikipedia —technically, a database— the amount of structured content is significantly overshadowed by the amount of content locked in the text of articles, despite the effort concentrated in the project.

In this work we investigate the task of applying standard Named Entity Recognition (NER) technology to enrich the metadata available in Wikipedia, and, by using this knowledge, how to generate additional training data to improve the NER technology without additional human intervention. The basic approach is based on linking the text of Wikipedia to the structured knowledge found in infoboxes. The process is illustrated in Figure 1. First, we annotate the Wikipedia collection using an off-the-shelf NER tool trained on a standard corpus, in Section 3.[2] Next, in Section 4, we link these semantic annotations with the structured knowledge made available by the DBpedia project[3]. This analysis provides a mapping between the broad categories of the semantic tagger

[1] Involving a significant fraction of Web searches.

[2] We use the word *annotation* interchangeably with the term *(semantic) tagging*, which is in more common use in the NLP community with much the same meaning.

[3] `http://www.dbpedia.org`

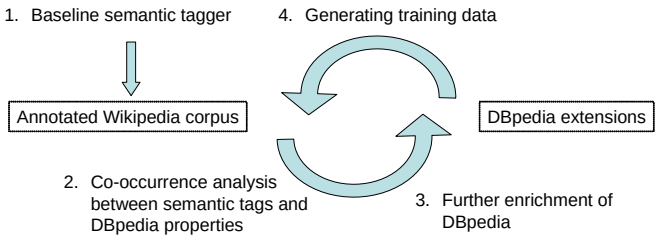

Figure 1: Increasing the level of semantics in Wikipedia and improving semantic tagging are the dual outcomes of our approach.

and the more refined metadata vocabulary of the DBpedia collection. In particular, we enrich DBpedia with additional class hierarchies, type information for resources and range restrictions for properties. This is a versatile resource that we expect to provide valuable background knowledge for many intelligent applications built with DBpedia. In a next step, we apply this mapping to the corpus and thereby generate additional training sentences for the semantic tagger, this time using sentences from the Wikipedia collection. In Section 6 we evaluate several taggers trained according to different strategies; our experiments show that the best solution is to combine professionally created training data with the data generated from Wikipedia. More specifically, we found that it is beneficial to select from the latter resource longer training sequences which provide more context around entity mentions.

The semantically annotated Wikipedia corpus is an improvement over the original version, can be used for improving tagging in other non-specialist domains and is itself a valuable source of knowledge, as the text of Wikipedia contains a large number of entities which have no corresponding Wikipedia articles (and thus are not included in DBpedia). For this reason, we make both our alignment and the semantically annotated Wikipedia corpus publicly available on the Web.[4]

2 Related Work

Machine Learning based 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 the adaptation of semantic annotation models from one source, with available training data, to another where training data is not available or scarce.

Model adaptation is still a critical research challenge, particularly when moving from narrower toward broader domains such as moving from news corpora to Wikipedia or the Web. Main approaches include *self-training*, i.e., adding automatically annotated data from the target domain to the original training

[4] `http://www.yr-bcn.es/semanticWikipedia`

data [11, 1], and *structural correspondence learning* [3, 2], which focuses on building a shared feature representation of the data from several domains. Recently several works have explored the use of Wikipedia in named entity recognition: Kazama and Torisawa have proposed to extract category labels, of the is-a kind, from definition sentences in Wikipedia articles to be used as features in NER systems [9]. For example, the word "painter" from the first sentence of Pablo Picasso's article. They show that these features can improve in-domain NER. Watanabe *et al.* [16] used anchor text in html links to build a graph which is used to disambiguate by means of a Conditional Random Field model. Mihalcea and Csomai [12] use effectively information extracted from Wikipedia for improving keyword extraction and word sense disambiguation, and also identify important concepts in Wikipedia articles in order to link them appropriately. Dakka and Cucerzan [8] propose a method which classifies full Wikipedia articles with named entity categories and exploits for training both labelled data and the Wikipedia category structure. Cucerzan [7] shows that entity detection inside the document can be significantly improved by a system which exploits co-referential coherence and background knowledge extracted from Wikipedia. Bunescu and Pasca [4] show that the category and link structure of Wikipedia can be used successfully to generate features for entity recognition. Our method differs from previous work in that we propose to use a novel source of information (infoboxes) directly to automatically generate additional training data, however our approach could be used in combination with the methods above.

Recently, we have seen the first works appearing that also exploit or extend in some ways the information in Wikipedia that is available as metadata. In particular, the KYLIN system implemented by Wu and Weld starts with the same idea that we apply in our work, i.e. establishing a correspondence of text and metadata [17]. They use the generated corpus for learning how to extract the values of properties from the text of 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 is complementary to the way in which we enrich DBpedia. As a necessary step in their approach they also perform document classification, based on very simple heuristics applied to Wikipedia category pages.

The work of Suchanek et al. on the YAGO ontology extends the idea of applying heuristics to extract information from proprietary aspects of Wikipedia — in particular categories, disambiguation pages and re-directions [15]. 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, although compared to the NLP approach we take, these heuristics seem to provide high precision but low recall. Further, we believe that heuristic-based approaches are inherently over-fitting the Wikipedia case, while the learning-based approach we present is much less dependent on the particular corpus and the proprietary aspects of Wikipedia articles.

3 Semantic annotation of Wikipedia

The goal of semantic annotation is to discover all occurrences of the classes of entities that are of interest for a given application in a given natural text. To annotate Wikipedia with entity labels we used a tagger which implements a first-order Hidden Markov Model (HMM), a statistical model of sequential structures. We briefly describe the tagger, discussed in greater detail in [5]. The HMM is trained with the average sequence Perceptron algorithm [6]. The tagger uses a generic feature set for NER (Named Entity Recognition) based on words, lemmas, PoS (part-of-speech) tags, and word shape features. PoS annotations were generated with the same tagger trained on the Wall Street Journal Penn Treebank [10].

Besides the choice of machine learning algorithm, a critical issue in learning-based semantic annotation is the acquisition of appropriate training data, i.e. sentences that are completely and consistently annotated with the entity labels required by an application. The base tagger we used in the experiments described here is trained on the CoNLL 2003 English NER dataset [14] which consists of 20,744 English sentences from Reuters news data. This corpus has been annotated with four category labels (Person, Location, Organization, Miscellaneous), comprising the vocabulary of our CoNLL tagger.[5] We have also applied our method using the training data of the Wall Street Journal financial news collection, which have been annotated with 108 hierarchically organized categories. However, we performed our experiments using the smaller CoNLL tagset: increasing the number of entity classes raises the level of complexity for performing evaluation with human experts. The number of possible choices to consider for marking up a particular term increases with the size of the vocabulary. Further, agreement among annotators becomes more difficult to maintain as the number of combinations of choices increases.

The two data sets mentioned here are typical of the publicly available data sources for learning-based NER in that they are focused on a relatively narrow domain (generic and financial news) but offer high quality annotations in terms of completeness, correctness and consistency in the use of the vocabulary. The limited scope of the domain and the quality of annotations makes it possible to achieve very good results when training and testing on fractions of the same corpus. For example, the accuracy of the tagger evaluated on held-out CoNLL data is approximately 91% F-score (the harmonic mean of precision and recall). The HMM tagger implements Viterbi decoding and thus its complexity is linear in the length of the sentence, and can be used to tag large amounts of data efficiently. The tagger is publicly available[6], and so is the semantically annotated version of Wikipedia. Figure 2 shows a sample of the output.

[5] In the following, we will refer to the classes of entities that a tagger is trained to recognize as the vocabulary of the tagger, although it should be clear from the above that in a learning-based approach the vocabulary is a characteristic of the data set and not the tagger itself.

[6] http://sourceforge.net/projects/supersensetag

Token	POS	CONLL	WSJ	Wikipedia
Pablo	NNP	B-PER	B-E:PERSON	B-persondata_name
Picasso	NNP	I-PER	I-E:PERSON	I-persondata_name
was	VBD	0	0	0
born	VBN	0	0	0
in	IN	0	0	0
Málaga	NN	B-LOC	B-E:GPE:CITY	B-persondata-placeOfBirth
,	,	0	0	0
Spain	NNP	B-LOC	B-E:GPE:COUNTRY	B-persondata-placeOfBirth

Figure 2: Example of the semantic annotation, where each column provides annotation for the token according to a given tagset. In order to represent annotations spanning multiple terms the tags are prefixed with either B (beginning of an annotation) or I (continuation of an annotation). The first three annotation columns are created by our baseline tagger, the last column is added by the our tool introduced in Section 4.

4 Wikipedia adaptation and metadata enrichment

Our approach to Wikipedia adaptation relies on mapping the annotation vocabulary of our existing tagger to the more fine-grained system of templates in Wikipedia. Template information and another metadata is provided by the DBpedia project. Using our alignment between the annotation vocabulary and DBpedia templates, we will also be able to significantly extend the metadata that can be currently extracted from Wikipedia.

In the following, we describe our approach of creating a mapping between the vocabularies used by our natural language taggers and resources in the DBpedia ontology. At a first step we will derive the range of properties (Section 4.1), and then use this information to infer the types of instances and to align DBpedia classes and our annotation vocabularies (Section 4.2). For the simplicity of our discussions we will use the vocabulary of the CoNLL tagger as an example (V_{conll}), which we also use in our evaluation (Section 6). However, the automated method described below could be applied to other annotation vocabularies.

4.1 Aligning annotation vocabularies and DBpedia properties

DBpedia is a lightweight ontology consisting of a straightforward extraction of the information in Wikipedia infoboxes.[7]

In DBpedia each page is represented by a resource, i.e. the underlying assumption is that each Wikipedia page represents a unique entity. (Resources are not instances of the templates, but rather related to them using the property *wikiPageUsesTemplate*.) As Figure 5 shows, an instance is described merely by

[7]DBpedia is available for download in RDF (Resource Description Framework), the most commonly used knowledge representation framework in the Semantic Web domain.

the name of the entry, the template classes used and the values for the template properties. While the amount of information is significant (close to 100 million triples) important ontological knowledge is missing. In particular, the data set is lacking range restriction on properties (*rdfs:range*). In other words, no information is provided on what type of values a given property can take.

To make it easier to follow the discussion below, we 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 articles in Wikipedia on the other hand. As Figure 3 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 kept separate. For example, Picasso's birth place (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, i.e. creating a corpus that is annotated with DBpedia properties in O_p. Our tool achieves this by processing the XML corpus of Wikipedia on a per article basis, looking up for every article the resource in O_i that is being described. Next, it iterates over all literal-valued properties of the resource, plus the labels of resources connected to the current resource through a resource-valued property.[8] In the example, the birth place is given as a link to two resources representing the article on Málaga and the article on Spain, respectively. Next, all occurrences of these values are annotated with the name of the corresponding property, i.e. *persondata_placeofbirth*. In case a word is part of multiple annotations (nested annotations) we only keep the outermost (longest) annotation.

The results of the annotation process are merged with the outputs of the semantic tagger to produce a *multitag* result format (see Figure 2), where the annotations from our tool show up as an additional column. This data set (available online) is useful on its own, for example, to perform learning on Wikipedia properties, as described in [17].

The multitag format is also the input for the next step of processing, where we compute the co-occurrence among tags in different tagsets. We analyze co-occurrence among tags on the same terms, i.e. co-occurrence of values within the rows of the multitag file (see Figure 2). The outcome of this step is a co-occurrence graph, where the nodes are tags and the edges between them represent the strength of the association measured by the number of co-occurrences. The output is in the format of the Pajek network analysis package, which can be used to further process and visualize these graphs [9].

Next, we extract mappings from this graph similar to the way we described in [13], where we analyze a co-occurrence network of tags in a folksonomy with the goal of finding broader-narrower relationships. In our previous work we

[8] There is no separation between datatype and object properties in DBpedia, which is one of the reasons why the ontology is not OWL compliant.

[9] `http://vlado.fmf.uni-lj.si/pub/networks/pajek/`

Figure 3: Wikipedia page describing Pablo Picasso. Note how the same information (regarding Picasso's birthplace, in this case) is described both in text and through an infobox, a feature that we exploit in our work.

have relaxed the notion of set subsumption by requiring only that there is a significant overlap among the two sets. However, we also constrained that the smaller set is smaller by at least a given factor in order to exclude cases where the two sets are of similar size (and therefore we can speak of similarity instead of subsumption). Further, to compute support we considered the size of the broader set.

In our current work, we adapt these measures as the goal is slightly different: we would like to find at most a single mapping for every tag and we are not concerned whether we find a similarity or a subsumption relationship. When finding a mapping for tag A, we are looking at how much a given tag B covers A relative to what is covered by any other tag, i.e. how much of the part that is covered by the tag set is 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 is covered by any tag in the tagset, i.e. for a given threshold k we require that $\frac{\sum_i |A \cap B_i|}{|A|} > k$. Lastly, we require that the absolute size of A is beyond a threshold m, i.e. $|A| > m$.

As an example, the tag *born* may have 100 occurrences and considered a *LOCATION* or a *MISC* in 70% and 20% of the cases, and not tagged with any tag in 10% of the cases. In this case, we consider the mapping of *born* to *LOCATION* 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. Note that if the first threshold is greater than

0.5, we will always have a single mapping.

In Figure 4 we list the results of mapping the very simple CoNLL tagset to Wikipedia properties, using hand-picked parameters ($n > 0.8$, $k > 0.8$, $m > 25$). As a reminder, these mappings provide a type for the ranges of properties, a crucial piece of knowledge that is missing in DBpedia. While quality is generally high, some of the mappings suffer from the liberties afforded by Wikipedia. In particular, when filling in templates users are also unconstrained in the values they can provide and the resulting heterogeneity goes mostly unnoticed as values are only used for display purposes.

Immediately apparent, for example, that Wikipedia users sometimes put more information than is needed and often put in some text just to indicate that a property is not applicable or unknown. Further, many properties such as *highlandercharacter_born* indeed suffer from ambiguity in that some users fill in locations while others add dates: in such cases even with the simple vocabulary that we have it is not possible to assign a single range restriction to the property.

Wikipedia property	CONLL	Examples of property values
infoboxNbaPlayer_name	PER	Alex Groza, Elgin Baylor, Jerry West, David Thompson, Glen Rice, Christian Laettner, Richard Hamilton, Juan Dixon, Sean May, Joakim Noah, ...
infoboxSerialKiller_alias	ORG	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, ...
cluster_name	ORG	AM-2, Christmas Tree Cluster, Coma Star Cluster, Double Cluster, Jewel Box, NGC 581, Messier 18, Messier 21, Messier 25, Messier 26, ...
highlanderCharacter_born	LOC	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, ...
infoboxSuperbowl_stadium	LOC	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, ...
infoboxNationalFootballTeam_name	LOC	Netherlands, Italy, Israel, United States, Mexico, Russia, Cuba, United Kingdom, Burkina Faso, France, ...
infoboxWeapon_usedBy	LOC	USA, None, One, none, Italy, United States, Mexico, UK, Russia, Under development, ...
minorLeagueTeam_league	MISC	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, ...
infoboxTea_teaOrigin	LOC	Nuwara Eliya, Sri Lanka near Adam's Peak between 2200 - 2500 metres, Japan, India, Vietnam, Taiwan, Turkey, China, Anhui, Guangdong, Jiangxi, ...
infoboxPrimeMinister_name	PER	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, ...

Figure 4: A sample of ten mappings with the first ten property values for each Wikipedia property, out of which seven is correct, one is partially correct (*highlanderCharacter_born*) and two are incorrect (*cluster_name*, *infoboxSerialKiller_alias*).

4.2 Enrichment of DBpedia

The reader may have noted that the co-occurrence analysis described above can also be applied between other columns of the multitag format. In particular, we can find mappings between terms or phrases and the annotation vocabulary. This gives a mechanism to extract those entities from Wikipedia that are con-

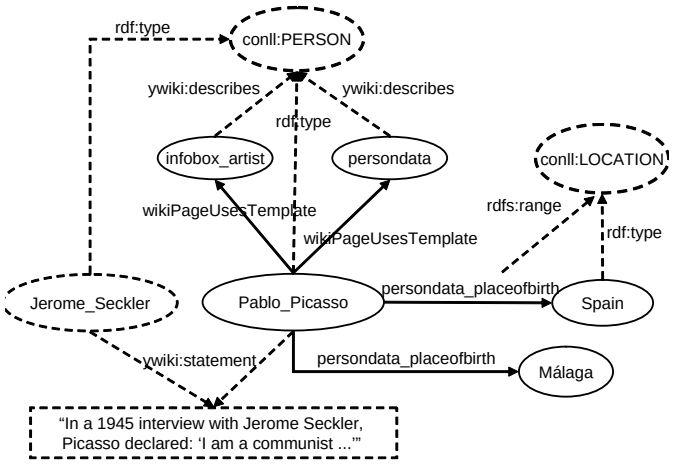

Figure 5: An example of the kind of knowledge learned through co-occurrence analysis and further processing of the results. Resources with a solid boundary are part of the original description of the Pablo Picasso resource in DBpedia. Resources with a dashed boundary are added to the ontology through our method.

sistently tagged with the same annotation class. In other words, we filter out terms and phrases that are ambiguous or are not always recognized as entities. We again obtain significant new knowledge compared to Wikipedia: while it is considered by many as complete in some sense, Wikipedia articles contain significant numbers of entities that don't have a Wikipedia page yet. In the case of the Picasso article this includes the names of persons who were important in Picasso's life, schools he attended, and even the name of a journalist who interviewed Picasso.

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

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: 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.[10]

[10] As templates are not represented as classes in DBpedia, we opted not to represent this information using the *rdfs:subClassOf* relationship, but rather to introduce a new instance

	PER	ORG	LOC	MISC	$\sum$
rdfs:range	805	515	480	124	1924
ywiki:describes	184	18	13	4	219
rdf:type (from text)	285873	140885	46079	32550	505387
rdf:type (through rdfs:range)	307874	58768	31383	8108	406133
rdf:type (through ywiki:describes)	47702	489	611	74	48876

Table 1: Summary of the number of statements learned through mapping, per class and in total. Note that the counts depend primarily on the thresholds chosen (here $n > 0.8$, $k > 0.8$, $m > 25$), although the distribution is likely to be typical.

The classification of instances is an important input, among others, for merging template information: for example, we can now find all the templates describing persons in Wikipedia, which are obvious candidates for merging, at least with respect to the basic properties relating to persons. The knowledge we learn 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 could significantly improve the quality of the information. By manually revising the mapping for just the most commonly occurring templates one can very effectively revise the classification of a significant number of resources.

In summary, we have enriched the DBpedia ontology with significant new knowledge as shown on our Picasso example in Figure 5. We have extracted range information for properties, which by inference also provides type information for the values of the properties. We have introduced new instances into the ontology based on NER. Lastly, we have extracted type information for templates which again provides type information for the resources that use them. Table 1 provides statistics on the number of statements we extract with reasonable, but handpicked settings for the parameters. These numbers do not include inferred statements.

5 Training data and re-tagging Wikipedia

In this section we investigate how the information extracted as described in the previous Section can be used to improve the baseline semantic tagger. The key idea here 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 the mention of "North Brabant" has been annotated with property *infobox_city_subdivisionname* from O_p:

- Boxmeer, is a village in the Netherlands in the province of **North Brabant**.

relationship linking templates to the types they describe.

We note that we can avoid problems of ambiguity as this annotation (like in the process described in Section 4.1) is performed on a per-article basis. While common words may change meaning within an article, this is unlikely to be the case with named entities.

From the learned mapping this mention can be identified as an instance of *LOCATION* in V_{conll}. The box identifies the extent of the data we can directly use for training: the remaining terms in the sentence may also be entities — in fact, there are two other location mentions in the sentence—, which we have no knowledge of. A limitation of this approach is that only positive examples are identified, while entities are a minority of the terms in the text. To generate better training data we built a "nostop" list of words which in the original gold-standard training data are tagged as non-entity labels. Then the context to use for training around the identified entity is extended as long as the words are in the nostop list. Thus in Example 5 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 LOC and all other words as "NULL". We call these stretches of text *fragments*, since they are large than entities but not necessarily sentences.

The set of training sentences we generate are not a random sample of Wikipedia text since is biased by the narrower scope of the source used to train the base tagger and the subsequent filtering process imposed. Additionally, since we extend the reliable context around an entity in a very conservative manner short fragments of text with minimal context are very frequent. As a result, in the automatically generated data the specific entity categories are over-represented which lead to trained taggers which tend to over-detect entities. To alleviate this problem we attempted a simple solution consisting in ranking the training sentences based on the number of terms with null labels. The intuition being that these instances provide more context and provide more informative, e.g., sentence-like, stretches of text. Then we selected a number of training instances for each category in such a way that they yield a distribution of categories close to the training data. One disadvantage of such a scheme is that part of the Wikipedia training data is not used for training and better solutions might be devised in the future.

6 Evaluation

Evaluation of semantic annotations is a difficult and expensive process, traditionally done by asking experts to read the text and annotate it by hand. This means identifying the entity boundaries and their type. Many kinds of difficult "semantic controversies" can arise even for simple tag sets, which can lead to low inter-annotator agreement and slow annotation speeds. Finally, evaluation efforts are often dominated by tagging the most frequent tags which are often the easiest to locate automatically and, therefore, the least interesting.

For this investigation we developed an evaluation framework which could be implemented in practice, in a short time and with few resources, as follows. We asked human judges to mark the *overall* quality of a tagged sentence, instead of annotating themselves the sentences with tags. The feedback is a binary decision: the sentence is judged correct (Ok) if it doesn't contain any mistakes, otherwise it is incorrect (Bad). We allowed a third outcome if the text was unintelligible (Unsure). Since tagging is to some degree subjective we ask several judges to look at each sentence and used their judgments as votes, taking the majority label vote as true (sentences resulting in ties were removed from the evaluation, see bellow).

We evaluate our work by considering the CoNLL tagger as a baseline. We used a fully automatic method combining automatic tagging (using the baseline model) and correlation analysis, as described in Section 4.1. As explained, this mapping could be improved by humans, either by curating it or by allowing users to specify them in the infoboxes.

Evaluated sentences were sampled at random from Wikipedia: after sorting all articles in a random order (disregarding entries used for training), we sampled approximately one every hundred sentences resulting in 989 sentences total, which were tagged with the baseline and evaluated. Then, the data was tagged by the adapted models and again evaluated[11]. We used 5 human judges. Sentences were evaluated by at least 2 judges (2.4 on average). Each sentence was seen only once by each judge, except when there was a difference between the models. A total of 6,956 judgments were produced in total, averaging 1,400 clicks per judge and a total of roughly 20 evaluation man-hours. Of the 989 sentences, 200 were tagged as Unsure by at least one judge and were discarded, leaving 789 sentences for evaluation. We evaluate these sentences for each model, removing sentences (for that model only) which resulted in tied judgments. The agreement between annotators was quite high considering the vagueness of the evaluation setting. If we consider all pairs of judgments on the same sentences by two different judges, the highest label disagreement was on Unsure labels; of these, 5% of the time the other label was also Unsure, 68% was Bad and 27% was Ok. These sentences were removed from the evaluation. Agreement was much higher on the remaining sentences for the Ok and Bad labels: 86% of pairs were consistently labeled.

Table 2 reports the performance of the different models. First let us consider the performance of the CoNLL tagger, used "out-of-the-box" without any form of adaptation to Wikipedia. This model tags 66% of sentences accurately; this is much less than the performance obtained by this tagger on news stories (above 80%), but nevertheless high, and shows that state-of-the-art taggers may be used on corpora such as Wikipedia with some success. Note that the remaining 34% incorrect sentences contain many correct entities, since a single incorrect entity is sufficient to make a whole sentence bad.

The performance of the adapted *Wikipedia* model is 58%, that is 7.6% worse

[11] except for sentences which had not changed.

Model	PGS	Δ%	p	%Identical
CoNLL	.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%

Table 2: Performance of the different models. PGS: percentage of sentences labelled `Ok`. Δ%: relative increase over the baseline (CoNLL). *%Identical*: the percentage of sentences which were tagged exactly as by the baseline model.

than the baseline in absolute terms (11% worse in relative terms).[12] Furthermore, 47% of sentences were tagged identically as by the baseline. Considering that the data used to train this model comes from Wikipedia and infoboxes alone, this performance is remarkable. Furthermore, we note that this performance is a lower-bound to the performance that could be obtained if a mapping of types (from templates tags to tagger tags) was available. Indeed, the quality of the Wikipedia model depends strongly on the quality of the mapping used. We need to translate the many idiosyncratic templates into a small unified set of types of interest.

The type of training data that is generated by infoboxes is biased towards the types of templates available. For example, it generates many more person names than organizations (see Table 1). We attempt to correct for this by re-sampling the training instances with respect to the more unbiased CoNLL distribution: model *Wikipedia-B* in Table 2. Its performance is similar to that of the unbalanced Wikipedia training set. We hypothesize that the distribution is so skewed that it cannot be easily unbiased by simply removing instances; it requires data from the under-represented types, which is simply not available. For this reason we experiment 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 it roughly that of the original CoNLL model. The difference in PGS (percentage of sentences labelled `Ok`) is only 1.3% relative, and it is not statistically significant. As expected, the number of sentences tagged identically to the CoNLL baseline increases (to 69%). We hypothesize that the lack of performance increase is due to the fact that the two distributions being combined are too different. To test this hypothesis we carried out the same re-sampling experiment as before, leading to the Mixed-B experiment setting. Indeed, the performance of this model is significantly better than the baseline: 5.5% relative, reaching 70% accuracy at the sentence level. It is interesting to note that the number of identical sentences is even higher than for the Mixed model, meaning that it has changed less sentences, but in the right way. This provides evidence

[12] The p value reported is computed using the two-sided Wilcoxon paired signed rank test, as implemented in *R*, see `http://sekhon.berkeley.edu/stats/html/wilcox.test.html`

that correcting for the distribution of entities is important and needs more investigation. Furthermore, it shows that combining hand labeled collections with user generated content can bring an increase in tagging performance.

7 Discussion and Future Work

In this paper we have investigated the task of enriching the DBpedia metadata collection through the use of a named entity tagger and information extracted from Wikipedia infoboxes. The amount of information generated is significant and automatically adds useful missing knowledge to DBpedia. We expect that the results will be useful for a number of Wikipedia-specific tasks such as mapping templates, cleaning up infobox data, providing better search and browsing experiences. As the domain of Wikipedia is broad, we also expect that our data sets will serve as useful background knowledge in other applications. As an example, we have shown how to apply the data toward the problem of improving our baseline tagger used for semantic annotation. In terms of semantic annotation, a natural question to ask if it is possible with this type of methods to closes the loop, as suggested in Figure 1, and create a feedback virtuous cycle between implicit user feedback and learning algorithm. We plan to explore this issue in the future. Our evaluation shows that significant improvements can be obtained compared to a baseline tagger on a task that is hard even for human evaluators due the breadth of human knowledge encompassed in Wikipedia.

Our work is ongoing in that there are still clear targets for technical improvements, such as better sentence detection, more precise tagging with Wikipedia metadata, and better balancing of the training data, among others. 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. Last, but not least we see the possibility to generalize our approach to other situations where semantic annotations are given or parallel text and metadata are available. In particular, Web pages annotated with microformats or RDFa would provide an interesting testing ground with larger scale, but certainly noisier metadata than in the case of Wikipedia.

References

[1] Michiel Bacchiani, Michael Riley, Brian Roark, and Richard Sproat. Map adaptation of stochastic grammars. *Computer Speech and Language*, 20(1):41–68, 2006.

[2] John Blitzer, Mark Dredzde, and Fernando Pereira. Biographies, bollywood, boom-boxes, and blenders: Domain adaptation for sentiment classification. In *Proceedings of ACL 2007*, 2007.

[3] John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of EMNLP 2006*, 2006.

[4] R. Bunescu and M. Pasca. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of the European Conference of the Association for Computational Linguistics*, Trento, Italy, 2006.

[5] Massimiliano Ciaramita and Yasemin Altun. Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger. In *Proceedings of EMNLP 2006*, 2006.

[6] Michael Collins. Discriminative training methods for Hidden Markov Models: Theory and experiments with perceptron algorithms. In *Proceedings of EMNLP 2002*, pages 1–8, 2002.

[7] S. Cucerzan. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of EMNLP-CoNLL 2007*, pages 708–716, 2007.

[8] W. Dakka and S. Cucerzan. Augmenting wikipedia with named entity tags. In *Proceedings of IJCNLP 2008*, 2008.

[9] Jun'ichi Kazama and Kentaro Torisawa. Exploiting wikipedia as external knowledge for named entity recognition. In *Proceedings of EMNLP 2007*, 2007.

[10] M.P. Marcus, B. Santorini, and M.A. Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2), 1993.

[11] David McClosky, Eugene Charniak, and Mark Johnson. Reranking and self-training for parser adaptation. In *Proceedings of COLING-ACL 2006*, pages 337–344, 2006.

[12] R Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management*, 2007.

[13] Peter Mika. Ontologies are us: A unified model of social networks and semantics. *Journal of Web Semantics*, 5(1):5–15, 2007.

[14] Erik F. Tjong Kim Sang and Fien De Muelder. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of CoNLL 2003 Shared Task*, pages 142–147, 2003.

[15] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago - A Core of Semantic Knowledge. In *Proceedings of the 16th International World Wide Web Conference (WWW2007)*. ACM Press, 2007.

[16] Yotaro Watanabe, Masayuki Asahara, and Yuji Matsumoto. A graph-based approach to named entity categorization in wikipedia using conditional random fields. In *Proceedings of EMNLP 2007*, 2007.

[17] Fei Wu and Daniel S. Weld. Autonomously Semantifying Wikipedia. In *Proceedings of the 16th ACM Conference on Information and Knowledge (CIKM 2007)*, 2007.