Information Arbitrage Across Multi-lingual Wikipedia

Eytan Adar
University of Washington, CSE
101 Paul G. Allen Center
Seattle, WA 98195
eadar@cs.washington.edu

Michael Skinner
Google
651 N. 34th St.
Seattle, WA 98105
mskinner@google.com

Daniel S. Weld
University of Washington, CSE
101 Paul G. Allen Center
Seattle, WA 98195
weld@cs.washington.edu

ABSTRACT

The rapid globalization of Wikipedia is generating a parallel, multi-lingual corpus of unprecedented scale. Pages for the same topic in many different languages emerge both as a result of manual translation and independent development. Unfortunately, these pages may appear at different times, vary in size, scope, and quality. Furthermore, differential growth rates cause the conceptual mapping between articles in different languages to be both complex and dynamic. These disparities provide the opportunity for a powerful form of *information arbitrage*– leveraging articles in one or more languages to improve the content in another. Analyzing four large language domains (English, Spanish, French, and German), we present *Ziggurat*, an automated system for aligning Wikipedia infoboxes, creating new infoboxes as necessary, filling in missing information, and detecting discrepancies between parallel pages. Our method uses self-supervised learning and our experiments demonstrate the method's feasibility, even in the absence of dictionaries.

Categories and Subject Descriptors

H.4.m Information Systems Applications, Miscellaneous

General Terms: Algorithms, Experimentation

Keywords: Information arbitrage, multi-lingual alignment, Wikipedia, translation

1. INTRODUCTION

Wikipedia is lauded for the millions of authoritative documents created, modified, and linked by a community of volunteer authors and editors. While studies have touted the factual veracity resulting from this process [14][15], fewer people have considered the ramifications of authors' linguistic diversity. Indeed, Wikipedia is becoming not only a repository for a great deal of factual information, but also a parallel, multi-lingual corpus of tremendous scale. Though the English subdomain of Wikipedia is first in page counts, with 2.4 million articles (as of July 1, 2008), this represents only 23% of the factual content. The remaining 77% of effort is distributed among over 250 languages (though principally focused on the top 50) [17]. As Wikipedians rush to translate, extend, and create new articles, there is a significant

WSDM'09, February 9-12, 2009, Barcelona, Spain.

Figure 1: Four different infoboxes from various languages. Figure 1a) and b) for Jerry Seinfeld in English and Spanish and Figure 1c) and d) for Emmanuelle Béart in French and English respectively.

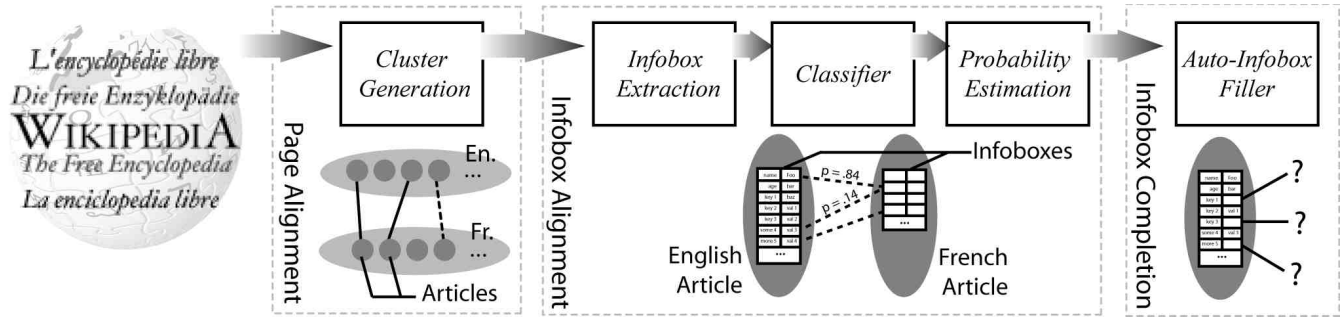

Figure 2: An architectural diagram describing the flow of Ziggurat.

opportunity for automation to amplify this process. In particular, most topics have a specific language which is most commonly used for updating the article. These disparities may have many causes—for example a particularly motivated editor may only write in his native language—but distribution and availability of expertise or inside information may also play a part.

This paper introduces the notion of *information arbitrage* across Wikipedia as a mechanism for detecting and exploiting these linguistic differentials. As in economic arbitrage, information arbitrage attempts to detect inefficiencies. In our case these inefficiencies are due to missing, old, or incorrect information in one language's corpus that can be "fixed" with the data from another. As we later discuss, there are many opportunities for information arbitrage within Wikipedia.

In this paper we focus attention on the differentials between *infoboxes* in different (language) versions of an article. Infoboxes are semi-structured blocks of summary data placed on many Wikipedia pages (Figure 1). In part we have selected infoboxes because their structure allows them to be aligned and evaluated without complex natural language processing. More importantly, they are a "beachhead" from which more complex extractions and alignment can be performed [20].

Figure 1 illustrates four different infoboxes demonstrating various differentials. Figure 1a and 1b, for example, are the infoboxes for the American comedian Jerry Seinfeld. We note the substantial amount of additional information in the English language infobox over the equivalent Spanish page. With information arbitrage, our goal is to deal with such situations and *automatically fill in missing infobox information.* In this example many of the fields in Seinfeld's English infobox can be propagated to the Spanish page. As another example, Figure 1c and 1d show the French and English infoboxes for the French actress Emmanuelle Béart. Although the two contain a substantial number of overlapping infobox fields (e.g. name, birthplace, etc.), the French page: a) has a more detailed birth place, b) disagrees with the English page on her birth name, and c) lists a different number of children. In this situation, even though the infoboxes are in essence equally complete, there are a number of discrepancies that would likely be worth bringing to the attention of the page editors. Because our system automatically aligns infobox data, it could potentially support conflict detection and linked-editing [13].

Figure 2 illustrates the general function of our Ziggurat and also serves as an outline for the remainder of this paper. As input, Ziggurat takes the Wikipedia content in four principal languages (English, Spanish, French, and German). The first phase, *page alignment*, fleshes out the—manually created, and hence incomplete—set of the cross-lingual links that denote equivalence between pages (Section 3.1). Utilizing these links and extracted infobox data, we generate correspondences between infobox attributes (fields), creating an alignment (Section 3.2). This field-by-field alignment provides scores that we can then use to decide which completed attributes (i.e. those with values) are the most likely match for an empty attribute (Section 3.3). Below, we briefly describe the data and opportunity for impact.

2. DATA AND OPPORTUNITY

The analysis and system described in this paper makes use of two Wikipedia datasets. First, we utilize the raw data dump from January of 2008 for the English, German, French and Spanish Wikipedia systems (used in the construction of cross-lingual links between articles). This data is in a form of markup known as Wikitext which is parsed by the Wikimedia content management system into HTML which can then be viewed in a browser. Each infobox is of a particular *class* (e.g. infobox_actor, or infobox_city_it) which defined a set of *attributes*. An attribute comprises a *key* and *value*. When an editor creates an infobox inside a page, they define these key/value pairs (e.g., name = "Tom Cruise," birthdate = "July 3, 1962," etc.).

While the raw data contains these infoboxes, the haphazard combination of HTML, Wikitext, templates, and so on make parsing this data extremely difficult. Fortunately, the DBpedia project [3] has processed infoboxes from the same period in a more suitable format. The DBpedia data represents all infobox fields found in the Wikitext. For example, we see data of the form:

Tom_Cruise	birthname	Thomas Cruise Mapother…
Tom_Cruise	spouse	Katie_Holmes
Tom_Cruise	spouse	Mimi_Rogers
	…	

The original data DBpedia data contains 23.2M, 2.9M, 2.7M and 1.4M such rows for English, German, French, and Spanish respectively. After a data cleaning step which collapses multiple rows with the same key (e.g. from the two spouse lines we create a single, set-valued attribute) we are left with 12.8M, 2.1M, 1.5M, and 880k rows for the different languages respectively. By relying on the template content, rather than the rendered data, we may not be able to discover that even though the Wikitext says "*sqarea = 45"* for some field, the user will see "*Square Area: 45 Km*^{2}" when viewing the page. This is an unfortunate but necessary compromise to achieve a higher quality level in the data as attempts to process the rendered HTML directly have proven to be extremely error prone.

2.1 Quantifying the Opportunity

Thus far, we have *assumed* that there are many missing infoboxes and infobox elements, but is this really true? With the data

described above it is possible to quantify this. To measure the number of missing infoboxes we begin by grouping conceptually-equivalent articles into clusters (e.g. the French, Spanish, English and German articles on Tom_Cruise are grouped into a cluster). The particulars of this grouping are described in further detail below. To find the potential number of articles for which we can generate new infoboxes we consider the number of infoboxes present in each cluster and the number that are missing. As long as we have one infobox defined in the cluster, we may be able to propagate this information to the other articles. For example, if only the English article in a cluster has an infobox, there is the potential to create 3 new infoboxes. Figure 3 shows the number of Wikipedia infoboxes which could potentially be created by translation. Note that we may even create a stub article in languages that do not already have an article for a given cluster. For example, if there was no Tom_Cruise article in French, we could automatically generate one and add a partially filled infobox. From this we see that, given 405k clusters with at least one infobox, it should be possible to create over 1 million new infoboxes, 845k of which would be in new stub articles.

Calculating the number of new infobox attributes which could potentially be translated is slightly more difficult. Many infoboxes have duplicate keys that are rarely used, or keys that only make sense in certain contexts (e.g. living actors do not need a *death_date* attribute). However, a baseline approximation of the potentially-translatable data can be calculated by comparing the relative size differences in paired infoboxes. This is done by measuring the average absolute difference between corresponding infoboxes. We find this average to be 6.5 attributes, indicating that there is substantial potential for translating data across attributes. We return to these questions in Section 4.3, when we discuss our experimental results.

3. ZIGGURAT

Ziggurat contains a number of modules (as depicted in Figure 2). Abstractly, Ziggurat attempts to solve the following problem: given a particular article in one language containing an infobox (of some class) that has a missing value for some key, find the most likely value that, when translated, would be an appropriate substitution. This is further complicated by the fact that the infobox class may also be unknown (i.e., from a page with no infobox). Because the replacement value most likely comes from the same article in a different language, Ziggurat attempts to build clusters that group together the same article ("concept") in different languages.

The Ziggurat attribute alignment module attempts to find the most probable mappings between infobox fields. By learning what a "match" is through a simple classifier and ranking possible matches to identify the best one, Ziggurat develops a ranking of the most likely sources for missing data.

3.1 Page Alignment

In fact Wikipedia already provides cross-language links between related articles (e.g., Figure 4a) so it is possible to know that a certain page in English, for example, has a corresponding French equivalent. Though the structure of these links is defined globally by a group of contributors

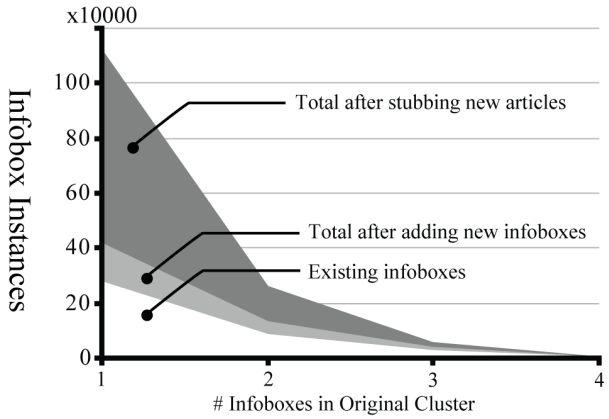

Figure 3: Potential for infobox creation

and policy makers (the Wikipedia Embassy), it is nonetheless voluntary and largely manual (various automated "bots" attempt to repair these links but as we see in our analysis are not entire successful). Figure 4b illustrates the number of existing cross-lingual links in January of 2008. Note that none of the language pairs has an equal number of cross-language links, despite the inherent symmetry—clearly many links are missing.

Thus, as the first phase of Ziggurat, we complete the page-level mapping by computing weakly connected components of the translation graph and assigning a unique concept id to each. As a precaution, we discard any component which contains more than one article in any given language.

3.2 Infobox Alignment

We now address the central task of identifying pairs of corresponding infobox attributes across languages. For example, we wish to predict that the *elevation* attribute of the English *Settlement* infobox, is the same as the *altitude* attribute of the *Ville_des_USA* infobox, but not the same as the *dens* attribute (denoting population density in French).

This problem can be formulated as a Boolean classification or probability estimation problem, but traditional supervised learning is not obviously applicable, because there is no explicitly-labeled training data. We confront this challenge with *self-supervised* learning. We first generate a training-set with a carefully-chosen set of general heuristics. Next, we apply logistic regression to

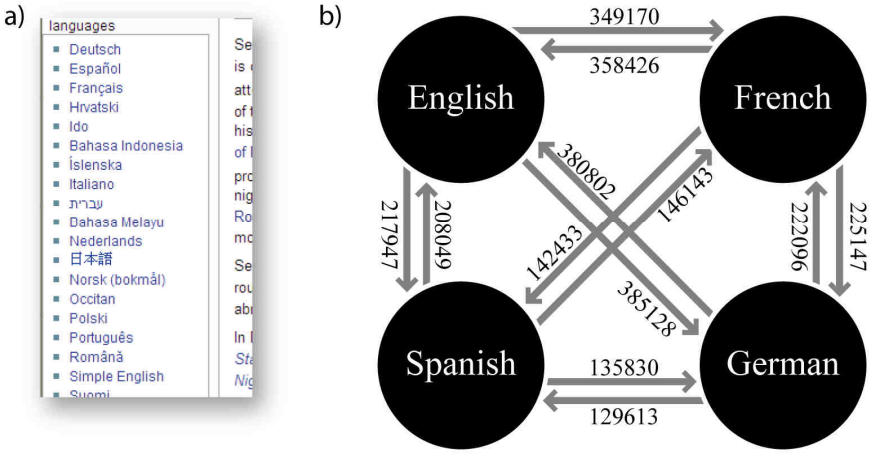

Figure 4: Sample cross-language (a) links and number of existing links between language pairs (b).

train a Boolean classifier over *instances* of infobox attribute pairs (those *without* missing data) to detect whether two *values* are likely to be equivalent. Finally, we use our classifier to determine how often pairs of attributes are found to be equal (e.g., to learn that pairings of *Settlement/elevation* are more likely equal to *Ville_des_USA/altitude* rather than *Ville_des_USA/dens*). As we show, the tremendous amount of Wikipedia data ensures that our method is effective.

In order to find frequently-matched pairings, we first train our Boolean classifier to detect matches. This must be done in a way that is insensitive to various transformations (e.g., "Tom Cruise" is "Cruise, Tom"), translations (e.g., "nombre" is "nom" and 14km is 8.7 miles), abbreviations (e.g., density is dens), and other forms of data mangling.

More formally, our classifier takes two different infobox tuples and outputs 1 if they are likely to be equal or 0 if not. A tuple consists of 4 elements: a language, an infobox class, an infobox attribute, and an infobox value. Each Wikipedia article will contain many such tuples. To model the data we use the following form: $\text{ArticleName}_{\text{Language}}[\text{InfoboxClass, KeyName}] = \text{KeyValue}$ (for example, $\text{Tom\_Cruise}_{\text{English}}[\text{actor,born}] = \text{July 3, 1962}$ to indicate that the actor infobox on the English Tom Cruise page states that he was born on July 3, 1962).

As we will see, it is possible to build such a classifier that performs with a high degree of accuracy. However, it is important to note that this classifier need not be perfectly accurate. Because we will test many pairings of class/key pairs between languages we should be able to generally identify alignment despite individual failures of the classifier.

Before evaluating the classifier in this way, we consider the how the same infobox values can be identified and corresponding features by which we train and test the classifier.

3.2.1 Features

The classification process begins by transforming a potentially matched pair of infobox tuples (article,language,infobox class, and key) into a feature vector that can be used in classification.

Equality Features (6 features) – The simplest test for identifying parallel tuples is to test for equality. Names and other words that remain constant regardless of language are a strong positive indication of a match. Though not as frequent, it is also possible for the attribute names or infobox classes to be equal. This happens when large classes of pages are copied from one language to another. We would expect a more significant amount of matching, for example, in the biological taxonomy infobox class, *taxobox*, which appears in both Spanish and English along with copied attribute names (e.g., *color*, *genus*, *ordo*, etc.). Three indicator variables are used as features to indicate equality. An additional set of three features check the equality of the normalized forms of the infobox values (i.e., lowercasing, removing everything but numbers, removing everything but alphabetical characters).

Word Features (2 features) – In some situations two equal infobox attributes may contain overlapping, but unequal values. This may be caused by, among other things, partial translations (some subset of the value has a unique term in the language) or slightly different lists (e.g., one has an extra element). To calculate similarity we tokenize each value into a set of words and calculate the Dice coefficient: $2 * |X \cap Y| / (|X| + |Y|)$ (where X and Y are sets of tokens). This value indicates on a scale of 0-1 (no match to perfect match) the similarity of the two sets. Additionally, the raw number of overlapping terms is retained as an additional feature.

n-Gram Features (4 features) – Because the languages we are working with frequently have words with similar roots it is possible to find matching substrings that are frequently a feature of matched infobox attributes. For example, in Spanish we may see *nombre* and in French *nom* or *Hamburg* in German or English but *Hambourg* in French. To identify such matches we generate 3 character n-grams (e.g., nombre = {nom,omb,mbr,bre}) and generate features corresponding to the intersection and Dice coefficient (as above). Features are generated both for the pair of values as well as the pair of attribute names.

Cluster ID Features (5 features) – Thus far we have not taken advantage of the hyperlinked nature of Wikipedia. When infoboxes contain links to other Wikipedia articles, it should be possible to utilize this information. For example, Juliette Binoche has the movie "The English Patient" as a value for her English infobox and "Le Patient Anglais" as a value in French. Both phrases are linked to the appropriate page for the movie in their respective language. Because we have previously determined that "Le Patient Anglais" and "The English Patient" are pages that are part of the same cluster and have the same concept ID, we have additional information that the values are equal (and thus the keys may be equal as well). Our feature generation process converts each hyperlinked element into a unique concept ID (generating a concept ID set for each infobox tuple). An indicator feature is used to indicate whether there is exact equality between the two input values. A second feature utilizes the number of intersecting concept IDs in situations where the value contains more than one, and a final feature generates the Dice coefficient for the concept ID set.

One issue with only linking to one concept ID is that there are various situations in which infoboxes point at articles from different places within a hierarchy. For example, Ang Lee, the director, was born in Pingtung, a city in Taiwan. One infobox may point to Pingtung as his birthplace whereas another will point at Taiwan. This will lead to a missed match. To resolve this issue we opted to make use of the fact that Wikipedia articles generally contain high level abstracts in their first paragraph. These abstracts generally mention, and point to, encapsulating topics (e.g. "contained in" or "part of" or "located in"). The Pingtung article, for example, states: "Pingtung City…is the capital of Pingtung County, Taiwan (Republic of China)." To utilize these we create a dataset containing all mentioned concept IDs within the abstract. Any concept ID that fails to be matched directly between the infobox values is tested against this database. Positive matches increase the value of the "containment" feature. For example, Pingtung and Taiwan will not match directly as they have different concept IDs, but the abstract for the Pingtung articles contains Taiwan which *does* match.

This is not an entirely satisfactory solution as there are many kinds of hierarchical encapsulation that are not captured by this simple test. A possible solution is utilizing known hierarchies (such as WordNet) or constructing our own. For example, one could mine the category structure of Wikipedia articles or construct more complex heuristics (e.g., understanding ranges of numbers and containment). This is a fairly complex addition that may be worth pursuing as future work, but is only useful in situations where one infobox class *only* lists values at different levels of the hierarchy (a situation we have not observed).

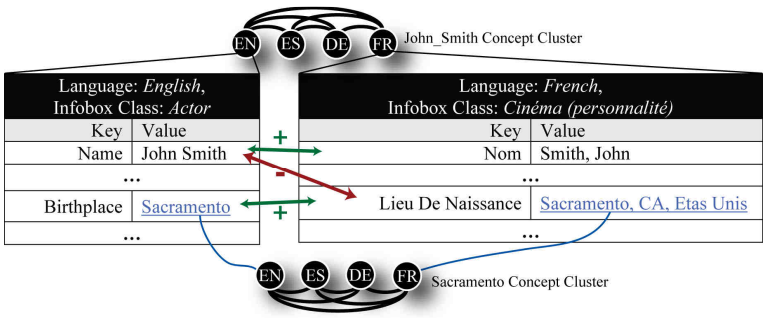

Figure 5: Mapping infoboxes and the construction of training examples

Language Features (1 feature) – A simple indicator variable is used to indicate which type of pairing is being tested (e.g., "German/English" is 1, "English/Spanish" is 2, etc.)

Correlation Features (2 features) – In the case of numerical infobox data, neither exact equality nor simple n-gram features are sufficient for determining value similarity. These features break down in the presence of noisy data (e.g. population estimates from different years) or related values (kilometers vs. miles). With this in mind, we calculate the Pearson product-moment correlation between pairs of numerical attributes across all instances of an infobox class. For example, if we find that {*English, commune_française, hectares*} frequently contains a number, we will compare that to every numerical attribute in the {*Spanish, localidad_de_francia*} class (the two classes are often paired). While most comparisons will not result in a high correlation, when we compare to the "Km^2" value, for example, we will get a nearly perfect fit (and an accurate linear regression that functionally transforms one value to another). This is not a perfect solution as frequently the system will find reasonable correlations that are not due to conversion (e.g. population versus landmass). We therefore cannot overly rely on this feature. However, it is useful for dealing with conversions and noisy data in conjunction with other features. In addition to the correlation we calculate and utilize the significance of each correlation (which may be used to filter directly at some α or after a multiple-comparison correction).

An appealing characteristic of calculating this feature is that it may also be used when suggesting values to fill infoboxes. For example, a value given in Km^2 in a Spanish infobox can be automatically converted into hectares if that is a more suitable unit for the French infobox. Since this is learned, we can bootstrap various numerical translations automatically without any prior knowledge.

Translation Features (6 features) – In situations where there is no textual similarity, we would like to make use of any language resources we have. To do so, we generate translations of each word by querying a sense-disambiguated panlingual translation dictionary, which is a continuation of the work by Etzioni, et al. [4]. This dictionary was created by starting from a collection of existing translation dictionaries, both bilingual and multilingual, and by inferring new translations.

Each word in the key, value, and infobox class name is translated by mapping it to all words in the target language. For example, when an English infobox containing *spouse* is tested against a Spanish infobox the following set is generated by our dictionary: {*consorte, cónyuge, cónyugue, dama, doña, emparejar, esposa, esposo, femenina, hembra, hombre, la mujer, marida, marido, mujer, pareja, señora, varon, varón*}. While the quality of the possible translations may vary, we are unlikely to find an overlap between two highly unrelated terms (e.g., *spouse* will never be translated to *nombre* (name)). For each key, value, and infobox class name we calculate two simple features to indicate a potential map. First, we calculate the number of successful matches (e.g., if a value in one tuple contains two words which are translated and one matches a word in the second tuple's value this score is 1). Second, we measure the ratio of matched terms to the total that can be translated (e.g. what percentage of the words mapped, extending our previous example this is 50%).

3.2.2 Generating a Labeled Training/Test Set

Once we have extracted our features, we generate training data to build our classifier. Recall that we would like our classifier to accept a pair of complete tuples (e.g. {*English, actor, name, "Tom Cruise"*} and {*Spanish, actor, nombre, "Cruise, Tom"*}) and decide if the two should be mapped even though the values are not exactly equal. We would like to generate a significant training set with a minimum of human intervention. To do so, we recognize that infobox pairs that are frequently equivalent are likely to be correspondents. For example, if we find that in the many cases of potential linking we observe, that {*Spanish, actor, nombre*} and {*English, actor, name*} contain exactly the same values we might infer that these two should be mapped. Unfortunately, if these were the only training examples we had, the classifier would learn to predict that only those tuples with equal values are linked. To avoid this, and provide a wider range of training examples, we find pairs of highly equal tuples and then find situations in which the values *are not* equal. This is visualized as the top positive example of Figure 5 (i.e., name and nom).

Our implementation of this is as follows. Each value is hashed (i.e., MD5(Value)) and the output is sorted by the concept ID, hash pairing. All equal (concept ID, hash) pairs that come from a different language are then labeled as a match. For example:

$A\_Person_{English}$[actor,name]= A_Person_{French}[Cinéma…,nom]

will mean that we increment the match counter for the pair {*English, actor, name*} and {*French, Cinéma (personnalité), nom*}. All values that have hyperlinks to some article will be replaced by their concept ID (for example, allowing us to determine that Sacramento and Sacramento, CA États-Unis are the same concept). In our dataset over 1M such "equal" tuple pairs are identified. Sorting these pairs by the number of times they are matched gives us a plausible set of correspondences to start from. Table 1, illustrates the top 6 scoring pairs (of 58k). To complete our selection of positive examples we take the top 4000 high-scoring pairs and find all instances of those pairs whether or not their values are actually equal (1.3M positive examples). Of these we select 20k as positive examples.

Generating negative examples is also relatively straightforward. The general idea is that if we find a positive pair (e.g., {*J_Smith, English, actor, name, "John Smith"*} and {*J_Smith, French, Cinéma (personnalité), nom, "Smith, John"*}), we can randomly select a second element from one of the infoboxes and generate a new, negative, pair. In Figure 5, a negative example is then the replacement of the *nom* tuple with the *lieu de naissance* tuple. In

order to prevent the random selection from generating another positive pairing, we remove from consideration the first 9000 frequently matched pairs in the equality list described above. This eliminates likely positive matches from being included as negative training examples, but does not completely remove pairs with matching values from consideration as negative examples. While this is reasonable (not all pairs of infobox tuples with equal values should be mapped), the end result is a higher number of false negatives. In running this algorithm, we find 3.7M possible negative examples (from which we select 40k for training).

Table 1: The 6 most frequent tuples found to be equal

#	language, infobox class, key	language, infobox class, key
8353	en (English), infobox_swiss_town, neighboringMunicipalities	fr (French), infobox_commune_de_suisse, communeslimitrophes (common boundries)
5524	en, infobox_cityit, postalcode	fr, commune_italienne, cp ("code postal")
5054	en, infobox_cityit, name	fr, commune_italienne, nom
4771	de (German), infobox_film, ds (short for "darsteller," or cast)	en, infobox_film, starring
4421	de, personendaten, geburtsort (birthplace)	en, persondata, placeOfBirth
4295	en, infobox_cityit, officialName	fr, commune_italienne, nom

3.2.3 Calculating Pair-Wise Scores

We train an Additive Logistic Regression [6] classifier on the training data described above (10-fold cross validation). Overall, our classifier achieves 90.7% accuracy in labeling pairs correctly (detailed in the experimental section below).

Having constructed our classifier, which is able to detect equivalence, we would now like to find the likelihood that a pair of keys will be equal given many examples. To do this we simply generate up to 100 examples of each possible pairing in the dataset. This number can be varied to generate sufficient significance under multiple-comparison corrections (though we leave this to future work). This process will generate 100 pairs of {*English,actor,name*} and {*French, Cinéma (personnalité), nom*} from existing data, 100 pairs of {*English,actor,name*} and {*French, Cinéma (personnalité), lieuDeNaissance*} and so on. Pairs are selected at random and fed into the classifier (16.9M of them). The classifier determines the number of matched pairs. The ratio between the number pairs found to be matches and the number tested gives us a score, *p*, that the pair is a good match. Running this algorithm identifies 161k pairs with $p > 0$.

Given that we have previously calculated pairings with a high number of exact matches, one might reasonably ask if we could not use these numbers directly for a score. Unfortunately, while effective in situations where there are many examples of a given pairing, edge (i.e., rare) cases do not work nearly as well. For example we see many Italian cities in Wikipedia that that have a *commune_italienne* infobox in the French article and i*nfobox_cityit* in English. In these situations it is easier to find enough matches to convince ourselves that certain values are equal. However, in situations where we do not have enough testable pairs (in the tail of the infobox distribution) we may not be able to find enough *exactly* matching pairs to distinguish between a true positive and a noise. For example, despite 29 potential matchings between the names of Vice Presidents in English and German (e.g. {*infobox_vice_president, name*} and {*personen-daten, name*}) only 1 instance matched exactly.

3.3 COMPLETING INFOBOXES

With weighted mappings between infoboxes, it now becomes possible to find corresponding pages, align infobox attributes and translate missing values. In Ziggurat, this is done by picking the target article, and then using the infoboxes from other articles sharing the same concept ID to complete the target infobox. Toward this end, there are two considerations that need to be made. The first is deciding which attributes should be filled for the target infobox (if the user does not explicitly tell us), and the second is how to transform from the approximate matchings to a more specific and precise case by case matching.

3.3.1 Choosing Potential Attributes

We employ three different methods for choosing target attributes, each increasingly more general. The first requires that the target article have an existing infobox, and works by simply picking the attributes of that infobox that are already present. Although this approach is not capable of generating new attributes, it has the advantage of using more relevant attributes, and could be applied to infobox cleanup and correction.

The second approach also requires that the target article have an existing infobox. However, instead of only using existing attributes, we now expand the potential attributes to include other attributes from the represented classes. For example, if the *actor* class can contain the attributes *name*, *born*, and *movies* and the particular instance only contains *name*, then the potential attributes list is expanded to contain *born* and *movies*. However, because many classes contain infrequently used attributes (e.g. typos or variations), we have implemented a configurable threshold, so that only highly occurring attributes are considered (e.g. attributes that occur at least 1% of the time in the class).

The third and final approach does not require any prior knowledge of the infobox to be completed. Instead, we guess the best set of potential classes, and then generate attributes by filling them out as in the second approach. The guesses are generated by counting infobox class co-occurrences. However, in order to prevent one extremely frequent class from matching many others, we also introduce a weighting mechanism to the co-occurrence count. Instead of using the raw co-occurrence, we weight each by a measure of how related the two classes are. This weight is currently the maximum pair-wise probability over all potential attribute matches, although more sophisticated mechanisms could be used. Once these weighted co-occurrences are found, we save the highest match for each target and source language pair. This also includes pairs where the source and target languages match. Although currently unused, this could be beneficial in generating more potential classes for both the second and third approaches.

3.3.2 Filling Missing Values

Having now determined several potentially-corresponding attributes, we must decide how to select the best match. The first, and simplest, approach is to pick, for each target attribute, the source attribute with the highest pair-wise score. This has been the primary technique employed by our system and, although simple, demonstrates fairly accurate results (see Table 2 for

Table 2: Experimental results showing the best matches for the fields in the English infobox actor class with p > .5 (infobox class names removed and only the top match from each language is retained, probabilities listed in parenthesis).

English	Spanish	German	French
baftaawards	premiosBafta (0.674)		
birthdate	fechaDeNacimiento (0.569)	geburtsdatum (0.712)	
birthname	nombreDeNacimiento (1) ...	imdbNameProperty (0.987) ...	nomDeNaissance (0.979) ...
birthplace	lugarDeNacimiento (0.893) ...	geburtsort (0.990)	lieuDeNaissance (0.946)
caption	nombre (0.8) ...	name (0.663) ...	nom (0.818) ...
cesarawards	premiosCesar (0.923)		
children	hijos (0.857) ...	name (0.552)	enfant (0.818)
deathplace	lugarMuerte (0.857) ...	sterbeort (0.920)	lieuDeDécès (0.846)
emmyawards	premiosEmmy (0.873)		
goldenglobeawards	premiosGloboDeOro (0.737)		
homepage	sitioWeb (1) ...	name (0.833) ...	siteInternet (1) ...
imagesize	tamañoDeFoto (0.921)		imagesize (0.878) ...
imdbId	imdb (1) ...	id (0.982)	imdb (1)
location	location (1) ...	geburtsort (0.900)	lieuDeNaissance (0.965)
name	name (1) ...	name (1) ...	nom (1) ...
notableRole	interpretacionesNotables (0.633)		filmsNotables (0.777)
othername	nombreDeNacimiento (0.604)	imdbNameProperty (0.8) ...	name (0.733)
parents	nombreDeNacimiento (0.703) ...	name (0.764) ...	nom (0.681)
restingplace	lugarDeDefunción (0.8)		
sagawards	lugarDeNacimiento (0.571)		
spouse	cónyuge (0.929) ...		conjoint (0.891)
tonyawards	premiosTony (0.555)		
website	sitioWeb (1) ...	name (0.955) ...	siteInternet (1)

example output). This approach acknowledges that different empty infobox attributes can be filled from the same matched attribute, but reasons about each key-key match independently. In the future, we hope to employ probabilistic joint inference, matching all attributes simultaneously.

There are a number of situations where this flexibility causes problems. For example, there are many "name" attributes for different attributes (birth name, alternative name, alias, etc.). Because "name" is similar to all these, it will be considered a high quality match to all of them. By filling in these fields with the name value we frequently make mistakes. A solution we have experimented with is enforcing a one-to-one mapping between two infoboxes. While this assumption is not entirely correct, all attributes within any given infobox should represent distinct pieces of information, so a one-to-one mapping is acceptable. Thus, if one assumes that two sets of infobox attributes have a one-to-one mapping, known algorithms for maximum weight matching can be used to determine a mapping. Preliminary experiments using the Kuhn-Munkres maximum weight bipartite matching algorithm on pairs of infoboxes show promising results.

Ziggurat will attempt to fill in the missing value in the language of the target article. In many situations for Wikipedia this is not necessary as infobox values are frequently personal names or numerical values that can be directly copied. However, there are situations where some translation would be useful. For example, if we are completing a French infobox using English infobox data, we would prefer to use Étas-Unis as the birthplace rather than United States. This is easiest in situations where the value is a hyperlink to another Wikipedia article in which case we might use the title for that article in the language of the missing article. In situations where there is no link, we must rely on either manual translation or automated dictionaries. It is here where use of the pan-lingual dictionary [4] can pay off by proposing a plausible set of translations which can be manually corrected.

4. EXPERIMENTAL RESULTS

Because Ziggurat's modules each depend on learning and are sensitive to the peculiarities of the data, we evaluate each independently.

4.1 Classifier Accuracy

As described earlier, the classifier accuracy for all features is 90.7% (10-fold cross validation). This result is biased towards more false negatives (17%) over false positives (5%). As alluded to earlier, we were interested in how much translation based features add to the precision of the classifier. Removing these features from consideration we find our precision goes down very slightly (90.6%) with most new mistakes being categorized as false negative (18% false negative rate). This slight difference may indicate that complex translation infrastructure or large dictionaries are not necessary for this task. That being said, we do believe that translation may be a highly useful feature when comparing very dissimilar languages. Because we are only considering western languages in this work, we can frequently rely on features such as character n-gram similarity to detect related words. This may not be the case when comparing English to Chinese, for example. Thus, while translation features do not add much to the accuracy of our results in this instance, they are likely worth retaining and considering in a more global scenario.

4.2 Scores and Attribute Matching

After generating the score, *p*, as described above, we wanted to make sure that both our intuition for the interpretation of these values as well as our mechanism for selecting training data was plausible. To test this we group the ranked list of infobox attribute pairs by the number of exact matches (as in Table 1) into quartiles (see Figure 6). Thus, the first quartile contains the pairs with the most number of exact matches, and so on. Plotting the average calculated score for each quartile we see a significant difference (Kruskal-Wallis, $p < 0.05$) between the first quartile and other groupings, and a general downward slope. This is a

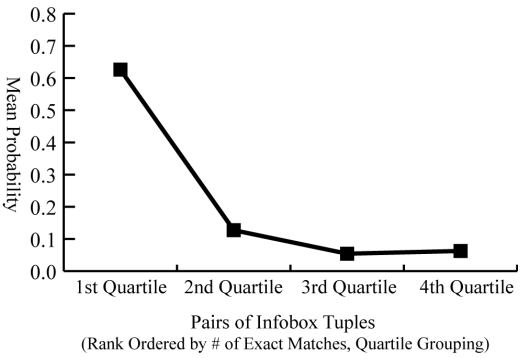

Figure 6: # of Exact matching versus probabilities

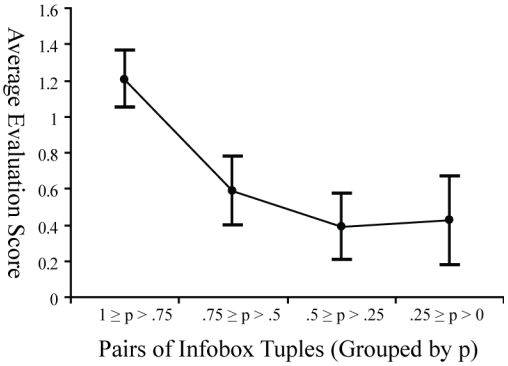

Figure 7: Probabilities versus Evaluation Scores

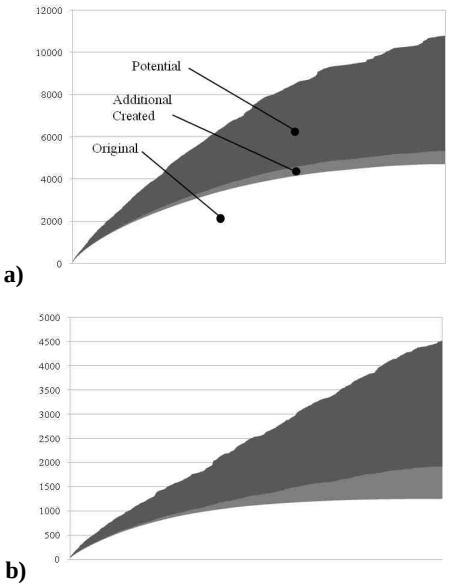

Figure 8: Original, content created by Ziggurat, and Potential for English (a) and Spanish (b) domains.

positive indication that pairings that are frequently equal also have a higher *p*.

To further test our measure we generated a list of 285 attribute pairings selected with a broad range of *p*. These were manually labeled by 2-4 participants on a scale of 0 (not a match) to 2 (a perfect match), with 1 indicating a possible, but non-ideal match (i.e., if one had a value from the given attribute one might infer the value for the other). Figure 7 illustrates the results. Again, all groupings are significantly different (Kruskal-Wallis, $p < 0.05$) and the higher the score the higher the average evaluation score. The disproportionately higher quality of pairs with $p > .75$ informed our decision to utilize this number as a threshold. We also found the average rank for each pair in each other's lists of best matches (i.e. for a pair of infobox attributes A and B, if we list all matches for A, where does B fall?). Calculating the correlation between average evaluation score and rank (for those up to rank 14 as data is sparse after this), we find a negative Kendall's tau correlation (-.384) indicating a degradation in evaluation score the lower the rank (i.e., worse the match).

Using a threshold of .75 to discard low quality matches, we "hid" existing infobox values and generated the most likely match using the simple ("best match") algorithm described above for 200 pairs. These were manually labeled by the authors, revealing an overall precision of 86%.

4.3 Ziggurat "Recall"

In Figure 8 we see a graphical representation of the gains made by our algorithm. These plots were generated by calculating the average number of entries for each infobox class before and after applying our system for infobox classes with at least 10 occurrences. We have also included a measure of potential, which is simply the 99th percentile of sizes for encountered instances of each infobox class. Classes are then sorted in decreasing order of average number of entries before plotting. However, a plot of the raw data is fairly noisy, since neither gains nor potential is directly proportional to average size, so we have instead plotted a cumulative version of this same data for ease of reading.

Particularly noteworthy about these plots is the continued growth of infobox sizes after applying our system even as the growth of existing entries begins to slow. This is due in large part to the ability of our system to generate a filled infobox even when the target article has no infobox or is missing altogether. Also interesting is the noticeably larger growth in Spanish and French (not shown) infoboxes as compared to English and German (not shown). This is strong evidence that our system is able to leverage size differentials to boost infobox sizes. Even more important is that these gains can be realized between any linked articles with such a size differential. The plots illustrate this trend on the per-language scale.

4.4 Generating Missing Infoboxes

In order to measure the quality of our infobox creation mechanism, we introduce two quality metrics. First we measure the quality of the guessed classes using the traditional measures of precision and recall. To create these numbers, we ran our infobox creation algorithm using existing infoboxes as a target (i.e. attempting to recreate an existing infobox from scratch), and then measuring the overlap between the classes in the existing infobox and the classes in the guessed infobox. Summing these overlaps, the number of guessed classes, and the number of existing classes over all target infoboxes gives us the overall precision of 54% and recall of 40%. German was at a high of 80.7% precision and English at a low of 45.7%. This is likely due to the fact that there are far fewer potential German infobox classes (e.g. *personnendaten* is a popular box for any type of person).

Keeping in mind that many infobox classes are applicable to any given article and the large number of potential classes, these numbers are quite acceptable. More than half of the classes

guessed are applicable, and around 40% of the original classes are "re-guessed." However, this only tells us the quality of the classes guessed, and not how the selected attributes are distributed within them. Although only 54% of the classes found by our algorithm already exist in the target infobox, it is possible that a disproportionate number of the found attributes lie within those classes. For this reason, we introduce our second quality measure, the percentage of guessed attributes that belong to the overlapping classes. This is a measure of the overlap between the attributes selected by the infobox creation and completion algorithms. The overall result is 71.8%.

This indicates that our algorithm is indeed finding high quality attributes to fill, even when nothing is known about the target infobox. Furthermore, this second measurement allows us to estimate a lower bound on the quality of created infoboxes. Since almost 72% of keys are shared with the completion algorithm, which has a precision of 86%, we can conclude that the creation algorithm has an estimated precision of at least 62%.

5. APPLICATIONS AND FUTURE WORK

In addition to the application described here, the ability to align pages and infobox attributes in multi-lingual Wikipedia has many other uses, several of which we are starting to explore.

5.1 Information Extraction

Wu and Weld [20] have shown that by heuristically matching infobox attributes with sentences containing identical values, their Kylin system can create a dataset for self-supervised training of a CRF extractor. When applied to an appropriate page, which *doesn't* yet have an infobox, the Kylin extractor can often find the correct attribute value, thus creating or completing an infobox for a page. In contrast, this paper has shown an additional way to obtain missing infobox values—by translating them from a language whose page *does* have the value in an infobox. But there are many possible improvements to our scheme.

Voting across Languages: Instead of picking a single language and translating the value from the page in that language, one could read the infobox value in multiple languages, translate into the target language and (if the candidate values differed) vote to find the most likely value.

Parallel Extraction: The same self-supervised methods pioneered by Kylin can be applied independently in each language, training CRF extractors from pairs of infobox values and the corresponding natural language sentences. After running these extractors on Spanish, French and other pages, there will be a much larger set of multi-lingual infobox attribute values; these can now be voted to create even higher-precision translations.

Joint Extraction: Instead of learning separate extractors for each language and then voting, a more sophisticated approach might be to learn a *single, joint* extractor which takes as input aligned pages from several languages. A single finite-state machine would be trained using bag of words, capitalization and part-of speech information in each language simultaneously.

Stacked Extraction: Instead of using voting to resolve a conflict when two languages disagree on the translated value of an infobox attribute, a better approach might be to train a meta-learner to learn to predict which language is more likely to have the correct answer. This stacked learner might learn rules of the form ``When German and French disagree, German is more likely correct—unless Spanish agrees with French."

Shrinkage across Languages: Wu *et al.* [18] showed that one could train more accurate Wikipedia CRF extractors by using a statistical technique, called *shrinkage*, to increase the number of training examples. Following, [9], they used a taxonomy to identify the correspond attribute A' for the parent, I', of I and the analogous attributes for subclasses of I. By treating the values of I'.A' (along with their matching natural-language text) as training examples for I.A, a much larger training set was obtained and both precision and recall improved. The same mechanism can be applied in multi-lingual Wikipedia, assuming that for any infobox class, I, different languages describe different sets of entities. Thus if the English version didn't have an infobox for an actor while the French version did, one could translate the French values to English and use the result for training examples. This approach can be improved using voting or stacked extraction. Furthermore, the process may be run iteratively, as with co-training: shrinking examples from L_1 to train extractors for L_2, extracting values for L_2, and then using shrinkage to learn a better classifier for L_1.

Page Classification: If a page has even a partial infobox, then it is clear what type of extractors should be applied to find values for additional attributes; however, if no infobox exists, one must use a classifier to determine which class of infobox is appropriate. Wu and Weld [20] used a simple heuristic classifier, which has high precision but low recall. Several machine-learning methods could be used to train a more versatile classifier—an important topic for future work. But this raises the question of which features should be fed to that classifier. Clearly, one might use a bag of words as well as list and category information. But our multi-lingual techniques suggest an even larger set of features. Rather than just using a bag of L_1 words when classifying an article, it seems most likely that including words from aligned pages in different languages will result in improved performance.

5.2 Ontology Learning

Wu and Weld [19] demonstrated an autonomous system for generating ontology (including parent/child mappings for corresponding attributes) over infobox classes in English, and shrinkage along this taxonomy was later shown to greatly improve the precision and recall of extraction [18] as we mentioned above. Ontological shrinkage will likely prove even more effective with a better taxonomy, so how can one improve the accuracy of ontology construction? Not surprisingly, multi-lingual Wikipedia again promises to help. Wu and Weld's approach leverages a number of features when predicting subsumption relations; for example, the revision history of a page. By aligning pages together and tracking the revision history of each, one would likely get several times more feature data in this regard alone.

6. RELATED WORK

Though there is a great deal of research on Wikipedia and its uses, there is only a limited amount on its multi-lingual properties. A number of systems have begun to apply this data for various tasks including question answering [5], thesaurus building [8], disambiguation and named entity extraction [11][16], and topic identification [7]. To our knowledge, Ziggurat is the first attempt at infobox alignment in the multi-lingual Wikipedia. However, other systems have recently emerged supporting other types of correspondences [2].

Creating cross-language links is a problem recognized both within the Wikipedia community—as evidenced by the creation of various "bots" to automatically fix these links—as well as in a

number of recent research projects [1][12]. In our work we have opted to utilize a fairly simple technique for completing missing links. However, we believe the output of this work, in particular the infobox alignment, can be used to feed back into link creation algorithms by identifying potential connections present in infoboxes but not in the link structure.

The task of infobox alignment is related to the automated schema matching/alignment techniques that is a popular topic in the database community [10]. We utilize a number of these techniques in our own work and hope to augment our system further with these mechanisms.

7. CONCLUSIONS

The globalization of Wikipedia shows no apparent slowdown and there is a unique opportunity to utilize the parallel work of editors versed in different languages. As content is created at different rates in different languages, and the quality of that content is highly variable, there is a huge opportunity to resolve differences and inconsistencies. In this paper we introduce Ziggurat, a system to automatically resolve differentials in infobox completeness. The system provides a unique mechanism that allows the content in one language to benefit from parallel content in others. By utilizing the notion that this differential is exploitable (an arbitrage opportunity), we develop an accurate system for filling in missing infobox data. We additionally discuss a number of other applications that leverage the multi-lingual Wikipedia and the alignment generated by Ziggurat.

8. ACKNOWLEDGEMENTS

We would like to thank Oren Etzioni and the Turing Center for feedback and support of this work as well as providing access to the PanDictionary. Additional thanks to Ivan Beschastnikh, Travis Kriplean, Raphael Hoffman, Fei Wu and Mausam for their feedback, advice, and labeling. This work is funded by the ARCS and NSF Fellowship and by the WRF / TJ Cable Professorship.

9. REFERENCES

[1] Adafre, S. F., and M. de Rijke, "Discovering Missing Links in Wikipedia," LinkKDD'05, Chicago, IL, August, 2005.

[2] Adafre, S. F., and M. de Rijke, "Finding Similar Sentences across Multiple Languages in Wikipedia," EACL '06, Trento, Italy, April 2006.

[3] DBpedia, www.dbpedia.org, last retrieved Aug. 10, 2008.

[4] Etzioni, O., K. Reiter, S. Soderland, and M. Sammer, "Lexical translation with application to image searching on the web." MT Summit XI, Copenhagen, Denmark, September, 2007.

[5] Ferrández, S., A. Toral , Ó. Ferrández, A. Ferrández, and R. Muñoz, "Applying Wikipedia's Multilingual Knowledge to Cross–Lingual Question Answering," *Lecture Notes in Computer Science*, vol. 4592, Springer, 2007.

[6] Friedman, J., T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting." *Annals of Statistics, 28(20),* 337-407, 2000

[7] Kawaba, M., H. Nakasaki, T. Utsuro, and T. Fukuhara, "Cross-Lingual Blog Analysis based on Multilingual Blog Distillation from Multilingual Wikipedia Entries," ICWSM'08, Seattle, WA, March 2008.

[8] Kinzler, D., "Automatischer Aufbau eines multilingualen Thesaurus durch Extraktion semantischer und lexikalischer Relationen aus der Wikipedia" Thesis, 2008.

[9] McCallum, A., R. Rosenfeld, T. M. Mitchell, and A. Y Ng, "Improving text classification by shrinkage in a hierarchy of classes," ICML '98, Madison, WI, July 1998.

[10] Rahm, E., and P. A. Bernstein, "A survey of approaches to automatic schema matching," *VLDB Journal*, 10:334-350, 2001.

[11] Richman, A.E., and P Schone, "Mining Wiki Resources for Multilingual Named Entity Recognition," ACL'08, Columbus, Ohio, June 2008.

[12] Sorg, P., and P. Cimiano, "Enriching the Crosslingual Link Structure of Wikipedia - A Classification-Based Approach," AAAI'08 Wikipedia and Artificial Intelligence Workshop, Chicago, IL, July 2008.

[13] Toomim, M., A. Begel, and S. L. Graham, "Managing Duplicated Code with Linked Editing," VL/HCC '04, Rome, Italy, Sep. 2004.

[14] Voss, J, "Measuring Wikipedia." 10th International Conference of the International Society for Scientometrics and Informetrics, Stockholm, Sweden. 2005.

[15] Weld, Daniel S., F. Wu, E. Adar, S. Amershi, J. Fogarty, R. Hoffman, K. Patel, and M. Skinner, "Intelligence in Wikipedia," AAAI'08, Chicago, IL, July 2008.

[16] Wentald, W., J. Knopp, C. Silberer, and M. Hartung, "Building a Multilingual Lexical Resource for Named Entity Disambiguation, Translation and Transliteration," LREC '08, Marrakech, Morocco, May 2008.

[17] "Wikipedia: MultiLingual Statistics," Aug. 10, 2008 en.wikipedia.org/wiki/Wikipedia:Multilingual_statistics,

[18] Wu, F., R. Hoffmann, and D. S. Weld, "Information Extaction from Wikipedia: Moving Down the Long Tail," KDD'08, Las Vegas, NV, Aug. 2008.

[19] Wu, F., and D.S. Weld, "Automatically Refining the Wikipedia Infobox Ontology," WWW '08, Beijing, China, Apr. 2008.

[20] Wu, F., and D. S., Weld, "Automatically Semantifying Wikipedia," CIKM '07, Lisbon, Portugal, Nov. 2007