Towards an Automatic Creation of Localized Versions of DBpedia

Alessio Palmero Aprosio[1], Claudio Giuliano[2], and Alberto Lavelli[2]

[1] Università degli Studi di Milano, via Comelico 39/41, 20135 Milano, Italy
alessio.palmero@unimi.it
[2] Fondazione Bruno Kessler, Via Sommarive 18, 38123 Trento, Italy
{giuliano,lavelli}@fbk.eu

Abstract. DBpedia is a large-scale knowledge base that exploits Wikipedia as primary data source. The extraction procedure requires to manually map Wikipedia infoboxes into the DBpedia ontology. Thanks to crowdsourcing, a large number of infoboxes has been mapped in the English DBpedia. Consequently, the same procedure has been applied to other languages to create the localized versions of DBpedia. However, the number of accomplished mappings is still small and limited to most frequent infoboxes. Furthermore, mappings need maintenance due to the constant and quick changes of Wikipedia articles. In this paper, we focus on the problem of automatically mapping infobox attributes to properties into the DBpedia ontology for extending the coverage of the existing localized versions or building from scratch versions for languages not covered in the current version. The evaluation has been performed on the Italian mappings. We compared our results with the current mappings on a random sample re-annotated by the authors. We report results comparable to the ones obtained by a human annotator in term of precision, but our approach leads to a significant improvement in recall and speed. Specifically, we mapped 45,978 Wikipedia infobox attributes to DBpedia properties in 14 different languages for which mappings were not yet available. The resource is made available in an open format.

1 Introduction

DBpedia is a community project[1] aiming to develop a large-scale knowledge base that exploits Wikipedia as primary data source. Wikipedia represents a practical choice as it is freely available under Creative Commons License, covers an extremely large part of human knowledge in different languages (45 out of 285 have more than 100,000 articles), and is populated by more than 100,000 active contributors, ensuring that the information contained is constantly updated and verified. At the time of starting this paper, the English DBpedia contained about 3.77 million entities, out of which 2.35 millions are classified in the DBpedia Ontology, available as Linked Data,[2] and via DBpedia's main SPARQL endpoint.[3] Due to the large and constantly increasing number

[1] http://dbpedia.org/
[2] http://wiki.dbpedia.org/Downloads
[3] http://dbpedia.org/sparql

H. Alani et al. (Eds.): ISWC 2013, Part I, LNCS 8218, pp. 494–509, 2013.

of links from and to other data sources, DBpedia continues to gain popularity and today it plays a central role in the development of the Web of Data.

The DBpedia ontology, consisting of 359 classes (e.g., person, city, organization) – organized in a subsumption hierarchy – and 1,775 properties (e.g., birth place, latitude, family name), is populated using a semi-automatic rule-based approach that relies prominently on Wikipedia *infoboxes*, a set of attribute-value pairs that represent a summary of the most important characteristics Wikipedia articles have in common. For example, country pages in the English Wikipedia typically contain the infobox `Infobox_country` containing specific attributes such as `currency`, `population`, `area`, etc. Specifically, the DBpedia project provides an information extraction framework[4] used, first, to extract the structured information contained in the infoboxes and, second, to convert it into RDF triples. Then, crowdsourcing is extensively used to map infoboxes and their attributes to the classes and properties of the DBpedia Ontology, respectively. For example, the `Infobox_country` is mapped to the class `Country` and its attribute `area` is mapped to the property `areaTotal`. Finally, all Wikipedia articles (instances) containing mapped infoboxes are automatically added to the DBpedia ontology, and mapped properties are used to add facts (statements) describing these instances. There are three main problems to solve. First, infoboxes do not have a common vocabulary, as the collaborative nature of Wikipedia leads to a proliferation of variants for the same concept. This problem is addressed using crowdsourcing, a public wiki for writing infobox mappings: editing existing ones, as well as editing the ontology, is available since DBpedia 3.5. Second, the number of infoboxes is very large, and consequently the mapping process is time consuming. To mitigate this problem, the mapping process follows an approach based on the frequency of infobox usage in Wikipedia articles. Most frequent elements are mapped first, ensuring a good coverage as infobox utilization follows the Zipf's distribution [17]. In this way, even though the number of mappings is small, a large number of Wikipedia articles can be added to the knowledge base. Third, mappings need maintenance due to the constant and quick changes of Wikipedia articles. For example, the Italian template `Cardinale_della_chiesa_cattolica` (Cardinal of the Catholic Church) has been replaced by a more generic `Cardinale` (Cardinal). In this particular case, the Wikipedia editors decided to delete the template, without creating a redirect link, therefore the mapping[5] between the template and the DBpedia class `Cardinal` becomes orphan, and the DBpedia extraction framework is no longer able to extract the corresponding entities.

At the early stages of the project, the construction of DBpedia was solely based on the English Wikipedia. More recently, other contributors around the world have joined the project to create localized and interconnected versions of the knowledge base. The goal is to populate the same ontology used in the English project, extracting articles from editions of Wikipedia in different languages. In its current version 3.8, DBpedia contains 16 different localized datasets and the information extraction framework has been extended to provide internationalization and multilingual support [7].

[4] `http://dbpedia.org/documentation`

[5] `http://mappings.dbpedia.org/index.php/Mapping_it:Cardinale_della_chiesa_cattolica`

However, the inclusion of more languages has emphasized the problems described above. Furthermore, the DBpedia ontology needs frequent extensions and modifications as it has been created on the English Wikipedia, while each edition of Wikipedia is managed by different groups of volunteers with different guidelines.

In this paper, we focus on the problem of automatically mapping infobox attributes to properties into the DBpedia ontology for extending the coverage of the existing localized versions (e.g., Italian, Spanish) or building from scratch versions for languages not yet covered (e.g., Swedish, Norwegian, Ukranian). This task is currently performed using crowdsourcing and there are no published attempts to perform it automatically. Related work has exclusively focused on developing automatic approaches to attribute mapping between different Wikipedia editions; these results can be used to automatize the mapping process, though this solution is highly prone to changes in Wikipedia, a noticeable drawback considering how fast edits are made. This study is complementary to previous investigations in which we studied the mapping of infoboxes to classes in the DBpedia ontology [10,11]. The above problem can be classified as schema matching, limited to alignment as we do not perform any successive merging or trasforming.

We propose an instance-based approach, that exploits the redundancy of Wikipedia in different editions (languages), assuming that attributes and properties are equivalent if their values are similar. Specifically, the mapping is cast as a binary classification task in which instances are infobox attribute/ontology property pairs extracted from versions of Wikipedia and DBpedia in different languages and cross-language links are used to represent the instances in a unified space. This allows us to learn the mapping function, for example, from existing mappings in English and German and predict Swedish instances. Attributes and properties are compared using their values taking into account their types (i.e., date, integer, object, etc.). For attributes, the type is calculated; for properties, the type is given by the ontology. We show that this approach is robust with respect to rapid changes in Wikipedia, differently from approaches that first map infoboxes among Wikipedia editions. The evaluation has been performed on the Italian mappings. We compared our results with the current mappings on a random sample re-annotated by the authors. We report results comparable to the ones obtained by a human annotator in term of precision (around 87%), but our approach leads to a significant improvement in recall (around 80%) and speed.

Finally, we mapped 45,978 Wikipedia infobox attributes to DBpedia properties in 14 different languages for which mappings were not yet available; the resource is made available in an open format.[6]

2 Problem Formalization

We consider the problem of automatically mapping attributes of Wikipedia infoboxes into properties of the DBpedia ontology. The problem can be classified as schema/ontology matching in which we are interested in equivalence relations between attributes and properties.

An infobox is a set of *attribute/value* pairs that represent a summary of the most salient characteristics Wikipedia articles have in common. For example, the

[6] `http://www.airpedia.org/`

infobox `Officeholder` in the English Wikipedia contains generic attributes, such as `name`, `birth_date`, and `birth_place`, and specific ones, such as `term_start`, `party`, and `office`. Notice that each Wikipedia edition is maintained by different communities and has different guidelines that can have a strong impact on the mapping results. For example, in the Italian edition, `Carica_pubblica` (`Officeholder`) does not contain generic attributes that are usually contained in the infobox `Bio`. In addition, there are no constraints on types, therefore in some editions of Wikipedia there can be a single attribute `born` containing both place and date of birth, while other languages decide to split this information into different attributes.

A DBpedia property is a relation that describes a particular characteristic of an object. It has a *domain* and a *range*. The domain is the set of objects where such property can be applied. For instance, `birthDate` is a property of `Person`, therefore `Person` is its domain. Around 20% of the DBpedia properties use the class `owl:Thing` as domain. The range is the set of possible values of the property. It can be a scalar (date, integer, etc.) or an object (`Person`, `Place`, etc.). For example, the range of `birthDate` is date and the range of `spouse` is `Person`.

Manual mappings are performed as follows. First, human annotators assign an infobox to a class in the DBpedia ontology. Then, they map the attributes of the infobox to the properties of the ontology class (or to its ancestors). An example of mapping is shown in Figure 1.

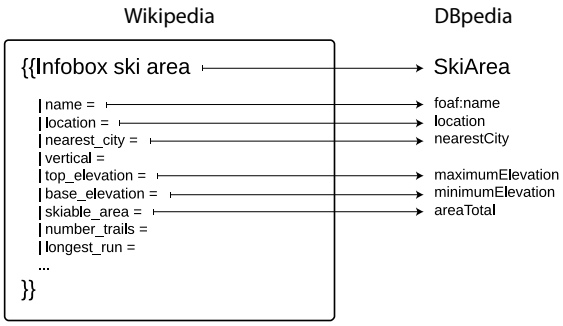

Fig. 1. Example of DBpedia mapping

The rest of the section is devoted to analyze the difficulties to adapt existing systems that perform infobox matching and completion (e.g., [13,4,1]) to solve this task. We could use existing approaches to map infoboxes between different Wikipedia editions and, then, use the existing DBpedia mappings to extend the mappings to languages not yet covered. An example is shown in Figure 2, where the template `Persondata` in English has been mapped to `Bio` in Italian, and similarly `Officeholder` to `Carica_pubblica`. Suppose that Italian mappings do not exist yet, they can be derived using the existing English DBpedia mappings. However, approaching the problem in this manner leads to a series of problems.

– Alignment of Wikipedia templates in different languages is often not possible, because there are no shared rules among the different Wikipedia communities on the management of infoboxes. In the example of Figure 2, `Carica_pubblica` only refers to politician, while `Officeholder` is more general.
– Properties may be mapped to different infoboxes in different languages. For example, the Italian DBpedia uses attributes of the `Bio` template to map generic biographical information, because specialized templates, such as `Carica_pubblica`, in the Italian Wikipedia do not contain generic information. This is not true in the English edition and in many other languages.
– Due to the previous point, some infoboxes are not mapped to any DBpedia class. This is the case of the `Persondata` template in English: since its information is repeated in the more specialized templates (for example, date of birth, name, occupation), the DBpedia annotators ignored it. A system that should align `Bio` and `Persondata`, and then transfer the mappings from English to Italian, would not map `Bio` to any DBpedia class since there is no mapping available for `Persondata`; therefore, all the generic biographical information would be lost.

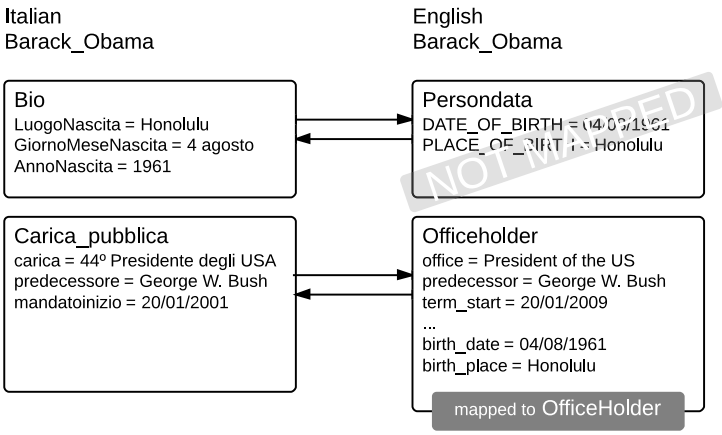

Fig. 2. An example of infobox alignment

3 Workflow of the System

In this work, we propose an automatic system for generating DBpedia mappings. Formally, given an infobox I and an attribute A_I contained in I, our system maps the pair $\langle I, A_I \rangle$ to a relation R in the DBpedia ontology.

Our approach exploits the redundancy of Wikipedia across editions in different languages, assuming that, if values of a particular infobox attribute are similar to values of a particular DBpedia property, then we can map the attribute to the property.

This approach requires existing versions of DBpedia to train the system, in particular we exploit the English, German, French, Spanish, and Portuguese editions. Given a target language l, the system extracts the mappings between DBpedia properties and infobox atttributes in such language. Note that the target language l can also be included in the set of languages chosen as training data; however, in our experiments we do not use this approach since we are interested in building mappings for those chapters of Wikipedia for which the corresponding DBpedia does not exist yet. Our system consists of three main modules: pre-processing, mapping extraction, and post-processing. Figure 3 depicts the workflow of the system.

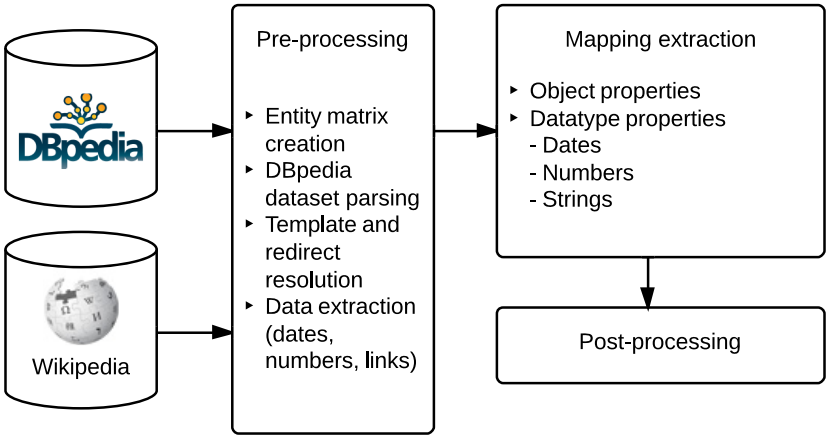

Fig. 3. Workflow of the system

4 Pre-processing

This section describes how we collect and normalize the data needed for the mapping between DBpedia and Wikipedia.

4.1 Entity Matrix Creation

The proposed approach makes considerable use of the redundancy of information among different versions of Wikipedia. In particular, we focus on the semi-structured information contained in the infoboxes. For example, the English Wikipedia page of Barack Obama contains an infobox with his birth date, birth place, etc. The same information is often included in the infoboxes of the corresponding pages in other Wikipedia editions. Therefore, the first step consists in building a matrix that aggregates the entities

(rows) in the different languages of Wikipedia (columns). The alignment is trivial as Wikipedia provides cross-language links between pairs of articles describing the same concept in different editions.

The accuracy of cross-language links has been investigated in the Semantic Web community [13,7], and conflicts have been found in less than 1% of the articles. In our implementation, when a conflict is found, the corresponding page is discarded.

In the rest of the paper, $P_{l_1}, P_{l_2}, \ldots$ denote the Wikipedia pages in languages $l_1, l_1, \ldots$, and P denotes the entity described by the corresponding row in the entity matrix. Figure 4 shows a portion of the matrix.

en	de	it	es	...
Xolile Yawa	Xolile Yawa	*null*	*null*	...
The Locket	*null*	Il segreto del medaglione	*null*	...
Barack Obama	Barack Obama	Barack Obama	Barack Obama	...
null	*null*	Giorgio Dendi	*null*	...
Secoya People	*null*	Secoya	Aido pai	...
...	...	...	...	...

Fig. 4. A portion of the entity matrix

4.2 DBpedia Dataset Parsing

DBpedia releases its ontology description in OWL format. The source file contains the description of the classes and properties, with all their characteristics. In our case, we search for the type (range) of each property. Depending on this feature, we can split them into two categories:

- *Datatype properties*, when the relation connects instances of classes to literals of XML (scalar values). For example `birthDate` connects a `Person` to a date.
- *Object properties*, when the relation connects instances of two classes (not necessarily different). For example, `birthPlace` connects a `Person` to a `Place` and `spouse` connects a `Person` to a `Person`.

Performing the mapping task, we use different strategies depending on the range of the category.

4.3 Template and Redirect Resolution

In Wikipedia, templates are particular pages created to be included into other pages. Infoboxes are a particular subset of templates that are usually rendered as a table in the upper-right corner of the corresponding Wikipedia article. Although this particular subset of templates is useful for information extraction from Wikipedia, only around 10% of templates belong to this category: the majority of them is used to give graphic

coherence to the same types of elements in different articles. For example, countries are often shown in Wikipedia infoboxes as the flag of the country followed by the name. These templates are often used as values for the infobox attributes. Since different languages have different strategies in using templates, the alignment between values containing templates is not trivial. During the alignment phase, these discrepancies may lead to errors. To address this problem, we pre-process the attribute values using the *Bliki engine*,[7] a parser that converts templates to their expanded text. After this operation, templates such as `{{EGY}}` are rendered as the Egypt flag followed by the name of the country linked to its page.

4.4 Data Extraction

In our approach, the main difficulty consists in the comparison between data obtained from DBpedia and attribute values stored in Wikipedia infoboxes. This is due to the fact that DBpedia is strongly typed, while Wikipedia does not have an explicit type system. Attribute values often contain a mixture of dates, numbers, and text, represented, formatted, and approximated in different ways depending on the Wikipedia edition and on the users who edit articles. These types of data can be formatted in different ways in different languages. For example, in English, we can express a date using different patterns, such as, "June 4th, 1983", "04/06/1983", or even "06/04/1983." Furthermore, numeric values can be approximated using variable precision depending on a particular edition of Wikipedia. For instance, the total area of Egypt is 1,002,450 in the English Wikipedia and 1.001.449 in the Italian one, where both the value and the format are different.

To tackle these problems, we defined a function e that, using a set of heuristics for numbers and dates, extracts – for each attribute value – four different sets of elements: numbers, dates, links and text tokens.

attribute	value
name	Diego Maradona
image	Maradona at 2012 GCC Champions League final.JPG
image_size	250
birth_place	[[Lanús]], [[Buenos Aires province|Buenos Aires]], [[Argentina]]
birth_date	{{Birth date and age|1960|10|30|df=yes}}
height	{{height|m=1.65}}
youthyears1	1968–1969
youthyears2	1970–1974
youthyears3	1975–1976
...	...

Fig. 5. `Infobox_football_biography` attributes for Diego Maradona

[7] `https://code.google.com/p/gwtwiki/`

In Figure 5 an example of `Infobox_football_biography` is presented. In the birth_place value, the value "[[Lanús]], [[Buenos Aires province|Buenos Aires]], [[Argentina]]" of the attribute *birth_place* is converted into the bag of links {`Lanús`, `Buenos_Aires_province`, `Argentina`} and the set of tokens {Lanús, ",", Buenos, Aires, ",", Argentina}, leaving the remaining sets (dates and numbers) empty. In the birth_date value, the template "Birth date and age" is parsed using the Bliki engine (see Section 4.3), resulting in "30 October 1960 (age 52)"; then, the string is converted into the set of dates {1960-10-30}, the set of numbers {30, 1960, 52}, and the set of tokens {30, October, 1960, (, age, 52,)}, leaving the links set empty.

5 Mapping Extraction

In this section, we describe the matching algorithm used to determine whether an attribute A_I contained in the infobox I in Wikipedia can be mapped to a given property R in DBpedia. To find the mappings, we have to calculate the pairwise similarities between the elements in the set of all the possible attributes A_I and the elements in the set of all the possible properties R. The candidates are represented as pairs (A_I, R), the pairs with the highest similarity $S(A_I, R)$ are considered correct mappings. The similarity is an average result calculated using instance-based similarities between the values of property R in different DBpedia editions and the values of the attribute A_I in different Wikipedia pages in the target language. This process can lead to large number of comparisons to determine if a pair (A_I, R) can be mapped. The rest of the section provides a detailed and formal description of the algorithm.

Given a relation R in DBpedia in languages $L = \{l_1, l_2, \ldots, l_n\}$ and a target language l, the algorithm works as follows.

1. We build the following set, discarding entities that are not involved in the relation:

$$\Pi_R = \{P_{l_i} : P_{l_i} \text{ has its corresponding } P_l \text{ and exists at least an instance of } R \text{ in DBpedia in language } l_i.\}$$

2. For each pair (A_I, R), we compute S_l:

$$S_l(A_I, R) = \frac{\sum_{P_{l_i} \in \Pi_R} \sigma_l(e(A_I, P_l), v(R, P_{l_i}))}{|\Pi_R|}$$

   where the function σ_l is defined in Section 6 and the division by $|\Pi_R|$ is used to calculate the average similarity between attributes and properties based on their values in different languages.
3. All pairs A_I, R for which $S_l(A_I, R) < \lambda$ are discarded. Varying λ, we can change the trade-off between precision and recall.
4. For each infobox I, for which at least a pair (A_I, R) exists, we select A_I^* such that the pair (A_I^*, R) maximizes the function S.
5. Finally, we obtain the set M_R of the selected pairs (A_I, R).

6 Inner Similarity Function

The inner similarity $\sigma_l(e(A_I, P_l), v(R, P_{l_i})) \rightarrow [0, 1]$ is computed between the value of A_I in language l, extracted and normalized by the function e defined in Section 4.4, and the values of R in the DBpedia editions in languages $l_1, l_2, \ldots, l_n$, extracted by the function v. In sections 6.1 and 6.2, the function σ_l is formally defined depending on the two categories used to classify the property R (see Section 4.2). We use V_W and V_D to indicate the values returned by the functions e and v, respectively.

6.1 Similarity between Object Properties

When the range of the property R is an object, the value V_D corresponds to a Wikipedia page. Using the entity matrix E, we look for the equivalent page V_D^l in the target language l. Then, we search V_D^l in the links set of V_W, and we set $\sigma_l(V_D, V_W) = 1/k$ if we find it – k is the cardinality of the links subset of V_W. By dividing by k, we downgrade the similarity in case of partial matching. If the links set of V_W does not contain V_D^l, or if V_D does not have a corresponding article in the target language (and therefore V_D^l does not exist), we compare the string representations of V_D and V_W (see Section 6.2).

6.2 Similarity between Datatype Properties

When the range of the property R is not an object, we handle 9 types of data: calendar related (*date*, *gYearMonth*, *gYear*), numeric (*double*, *float*, *nonNegativeInteger*, *positiveInteger*, *integer*), and *string*. We discard the `boolean` type, as it affects only 4 properties out of 1,775, and it is never used in languages different from English.

Calendar Related Data. Given the value V_D of type *date* and the set V_W, we compute $\sigma_l(V_D, V_W)$ by searching the day, the month and the year of V_D in the set V_W. In particular, the month is given only if it appears as text, or if it is included in the numbers set of V_W together with the day and the year. Similarly, we look at the day only if it appears with the month. We look at the date parts separately, because some Wikipedia editions split them into different infobox attributes. We assign a value of $1/3$ to each part of the date V_D that appears in V_W.

$$\sigma_l(V_D, V_W) = \begin{cases} 1 & \text{if day-month-year are present in } V_W \\ 2/3 & \text{if day-month are present in } V_W \\ 2/3 & \text{if month-year are present in } V_W \\ 1/3 & \text{if year is present in } V_W \end{cases}$$

Similarly, for *gYearMonth* we set $\sigma_l(V_D, V_W) = 1$ if both month and year appear in the dates set of V_W, and $\sigma_l(V_D, V_W) = 0.5$ if V_W contains only one of them. Finally, for *gYear* we set $\sigma_l(V_D, V_W) = 1$ if the year is included in the numbers set of V_W.

Numeric Data. While for calendar related data we expect to find the exact value, often properties involving numbers can have slightly different values in different languages (see Section 4.4 for an example). If $V_D = 0$, we check if the numbers subset of V_W contains 0. If true, then $\sigma_l(V_D, V_W) = 1$, otherwise $\sigma_l(V_D, V_W) = 0$. If $V_D \neq 0$, we search for values in V_W near to V_D, setting a tolerance $\nu > 0$. For each n in the numbers set of V_W, we calculate $\varepsilon = |V_D - n| \,/\, |V_D|$. If $\varepsilon < \nu$, then we set $\sigma_l(V_D, V_W) = 1$ and exit the loop. If the end of the loop is reached, we set $\sigma_l(V_D, V_W) = 0$.

Strings. String kernels are used to compare strings. To compute the similarity, this family of kernel functions takes into account two strings and looks for contiguous and non-contiguous subsequences of a given length they have in common. Non contiguous occurrences are penalized according to the number of gaps they contain. Formally, let Σ be an alphabet of $|\Sigma|$ symbols, and $s = s_1 s_2 \dots s_{|s|}$ a finite sequence over Σ (i.e., $s_i \in \Sigma, 1 \leqslant i \leqslant |s|$). Let $\mathbf{i} = [i_1, i_2, \dots, i_n]$, with $1 \leqslant i_1 < i_2 < \dots < i_n \leqslant |s|$, be a subset of the indices in s, we will denote as $s[\mathbf{i}] \in \Sigma^n$ the subsequence $s_{i_1} s_{i_2} \dots s_{i_n}$. Note that $s[\mathbf{i}]$ does not necessarily form a contiguous n-gram of s. The length spanned by $s[\mathbf{i}]$ in s is $l(\mathbf{i}) = i_n - i_1 + 1$. The gap-weighted subsequences kernel (or string kernel) of length n is defined as

$$K_n(s,t) = \langle \phi^n(s), \phi^n(t) \rangle = \sum_{u \in \Sigma^n} \phi^n_u(s) \phi^n_u(t), \tag{1}$$

where

$$\phi^n_u(s) = \sum_{\mathbf{i}: u = s[\mathbf{i}]} \mu^{l(\mathbf{i})}, u \in \Sigma^n \tag{2}$$

and $\mu \in]0, 1]$ is the decay factor used to penalize non-contiguous subsequences.[8] An explicit computation of Equation 1 is unfeasible even for small values of n. To evaluate more efficiently K_n, we use the recursive formulation based on a dynamic programming implementation [8,14,5].

In our implementation, subsequences are n-grams (strings are tokenized), where $n = \min\{|V_D|, |V^*_W|\}$ and V^*_W is the tokenized set of V_W where some n-grams have been replaced with their translation when cross-language links exist. The similarity function is defined as the first strictly positive value returned by the following loop:

$$\sigma_l(V_D, V_W) = \frac{K_i(V_D, V^*_W)}{n - i + 1} \qquad \text{for each } i = n, n-1, \dots, 1.$$

7 Post-processing

Some infoboxes contain attributes with multiple values. For example, the musical genre of a particular album can be "rock" and "pop", or a book can have more than one author. In these cases, Wikipedia provides more than one attribute describing the same relation, and adds an incremental index after the name of the attribute (sometimes

[8] Notice that by choosing $\mu = 1$ sparse subsequences are not penalized. The algorithm does not take into account sparse subsequences with $\mu \to 0$.

also adding an underscore between the attribute name and the index). For example, the `Infobox_settlement` template contain the attribute `twinX` used for twin cities, where X can vary from 1 to 9. In our system, if M_R contains a mapping $A_I \rightarrow R$, we also add the set of mappings $A'_I \rightarrow R$ where the name of attribute A' differs from A only for an added or replaced digit. This filter is applied on the set M of mappings built in the mapping phase (Section 5) and is only used to increase recall.

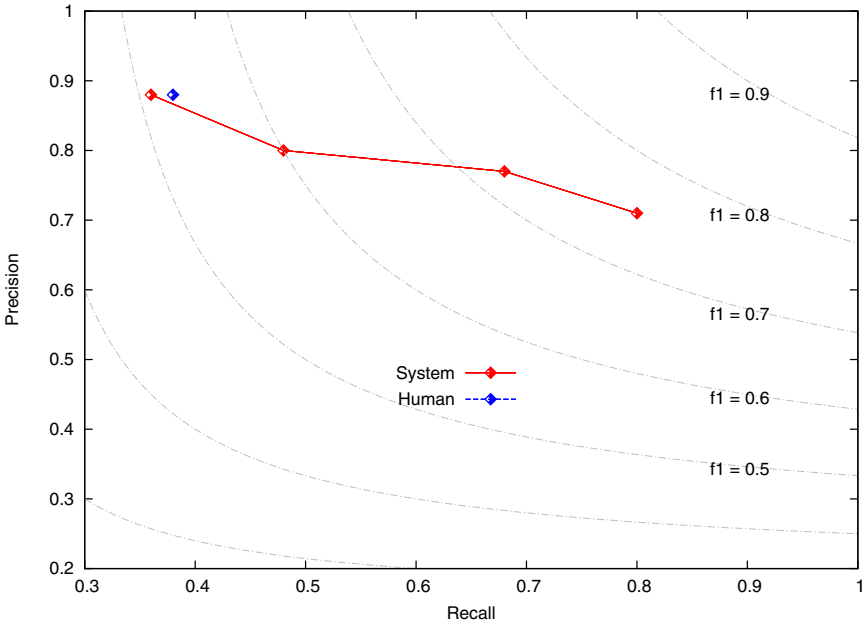

Fig. 6. Precision/recall curve of our system compared with the DBpedia original manual mapping in Italian. From left to right, λ value is 0.9, 0.7, 0.5, and 0.3.

8 Evaluation

Experiments have been carried on Italian, using existing DBpedia editions in five languages (English, Spanish, Portuguese, German, and French) as training data. To perform the evaluation, three annotators created a gold standard by manually annotating 15 infoboxes (for a total of 100 different attributes), randomly extracted from the first 100 most frequent infoboxes in the Italian Wikipedia. The inter annotator agreement is 91%, with respect to Fleiss' kappa measure [6]. The gold standard is available online on the Airpedia website.[9] As baseline, we use the manually mapped Italian infoboxes that can be downloaded from the DBpedia official website. [10] Specifically, we used the

[9] `http://www.airpedia.org/download/dbpedia-property-mappings-in-14-languages/`

[10] `http://mappings.dbpedia.org/`

version available on April 5th, 2013, made available by the Italian DBpedia project,[11] consisting of around 50 infoboxes and 469 attributes (in 18 infoboxes) mapped by one annotator during the spring 2012.

Figure 6 shows the precision/recall curve. Different precision/recall points are obtained by varying the parameter λ described in Section 5. The grey dashed lines join points with the same F_1. The results show that the coverage of the baseline (Human) is around 38% with a precision of around 88%. Our system is able to achieve comparable results in term of precision (87%), but it leads to a significant improvement in recall maintaining acceptable precision. Specifically, we can see that, by exploiting existing mappings, we can cover up to 70% of the attributes with a precision around 80%. Even though the procedure is not generally error-prone, we believe that it can be used as a starting point for releasing new DBpedia editions or extending existing ones. In the next section, we describe the current release of the resource.

9 The Resource

Overall, our system mapped 45,978 Wikipedia infobox attributes to DBpedia properties in 14 different languages for which mappings do not yet exist.[12] For each language, we only consider templates that appear more than 10 times in the corresponding Wikipedia and release the mappings paired with the value of the function f, described in the Section 5. The system has been trained on the DBpedia datasets in 6 languages (English, Italian, French, German, Spanish and Portuguese).

Table 1 shows the number of mappings extracted for each language ($\lambda = 0.3$). Notice that, even if the precision is not 100% and the process still needs human supervision, our approach can drastically reduce the time required, estimated in around 5 minutes per mapping per language if performed from scratch.[13]

Table 1. Mappings extracted and available as a resource

Language	Mappings	Language	Mappings
Belarusian	1,895	Norwegian	4,226
Danish	3,303	Romanian	4,563
Estonian	1,297	Slovak	2,407
Finnish	3,766	Albanian	1,144
Icelandic	646	Serbian	4,343
Lithuanian	3,733	Swedish	5,073
Latvian	2,085	Ukranian	5,760

10 Related Work

The main reference for our work is the DBpedia project [2]. Started in 2007, it aims at building a large-scale knowledge base semi-automatically extracted from Wikipedia.

[11] `http://it.dbpedia.org/`

[12] The complete resource is available at `http://www.airpedia.org/`

[13] This is an average time evaluated during the mapping of the Italian DBpedia.

Wikipedia infobox attribute names do not use the same vocabulary, and this results in multiple properties having the same meaning but different names and vice versa. In order to do the *mapping-based* extraction, DBpedia organizes the infobox templates into a hierarchy, thus creating the DBpedia ontology with infobox templates as classes. They manually construct a set of property and object extraction rules based on the infobox class. Nowadays, the ontology covers 359 classes which form a subsumption hierarchy and are described by 1,775 different properties. The English version is populated by around 1.7M Wikipedia pages, although the English Wikipedia contains almost 4M pages.

Yago [16], similarly to DBpedia, extracts structured information and facts from Wikipedia using rules on page categories. Conversely, FreeBase [3] and WikiData [18] are collaborative knowledge bases composed mainly by their community members.

The problem faced in this paper falls into the broader area of schema matching. A general survey on this topic is presented by Rahm and Bernstein [12]. Their work compares and describes different techniques, establishing also a taxonomy that is used to classify schema matching approaches. Similarly, Shvaiko and Euzenat [15] present a new classification of schema-based matching techniques. It also overviews some of the recent schema/ontology matching systems, pointing which part of the solution space they cover.

Bouma et al. [4] propose a method for automatically completing Wikipedia templates. Cross-language links are used to add and complete templates and infoboxes in Dutch with information derived from the English Wikipedia. First, the authors show that alignment between English and Dutch Wikipedia is accurate, and that the result can be used to expand the number of template attribute-value pairs in Dutch Wikipedia by 50%. Second, they show that matching template tuples can be found automatically, and that an accurate set of matching template/attribute pairs can be derived using intersective bidirectional alignment. In addition, the alignment provides valuable information for normalization of template and attribute names and can be used to detect potential mistakes. The method extends the number of tuples by 50% (27% for existing Dutch pages).

Adar et al. [1] present Ziggurat, an automatic system for aligning Wikipedia infoboxes, creating new infoboxes as necessary, filling in missing information, and detecting inconsistencies between parallel articles. Ziggurat uses self-supervised learning to allow the content in one language to benefit from parallel content in others. Experiments demonstrate the method's feasibility, even in the absence of dictionaries.

Nguyen et al. [9] propose WikiMatch, an approach for the infobox alignment task that uses different sources of similarity. The evaluation is provided on a subset of Wikipedia infoboxes in English, Portuguese and Vietnamese.

More recently, Rinser et al. [13] propose a three-stage general approach to infobox alignment between different versions of Wikipedia in different languages. First, it aligns entities using inter-language links; then, it uses an instance-based approach to match infoboxes in different languages; finally, it aligns infobox attributes, again using an instance-based approach.

11 Conclusion and Future Work

In this paper, we have studied the problem of automatically mapping the attributes of Wikipedia infoboxes to properties of the DBpedia ontology. To solve this problem, we have devised an instance-based approach that uses existing DBpedia editions as training data. We evaluated the system on Italian data, using 100 manually annotated infobox attributes, demonstrating that our results are comparable with the current mappings in term of precision (87% versus 88% for the human annotation), but they lead to a significant improvement in term of recall (70%) and speed (a single mapping may need up to 5 minutes by a human), maintaining an acceptable precision (80%). The system has been used to map 45,978 infobox attributes in 14 different languages for which mappings were not yet available; the resource is made available in an open format.

There remains room for further improvements. For example, the similarity function can be refined with a smarter normalization and a better recognition of typed entities (like temporal expressions, units, and common abbreviations).

We will also evaluate to what extent (precision/recall) DBpedia class mappings can be generated from the property mappings automatically found using our system.

Finally, we will adapt the proposed approach to detect errors in the DBpedia mappings (during our tests we encountered a relevant number of wrong mappings in DBpedia), or to maintain the mappings up-to-date whenever the corresponding Wikipedia templates are updated by the Wikipedia editors.

References

1. Adar, E., Skinner, M., Weld, D.S.: Information arbitrage across multi-lingual Wikipedia. In: Proceedings of the Second ACM International Conference on Web Search and Data Mining, WSDM 2009, pp. 94–103. ACM, New York (2009), http://doi.acm.org/10.1145/1498759.1498813
2. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia - a crystallization point for the web of data. Web Semant. 7(3), 154–165 (2009), http://dx.doi.org/10.1016/j.websem.2009.07.002
3. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, pp. 1247–1250. ACM, New York (2008), http://doi.acm.org/10.1145/1376616.1376746
4. Bouma, G., Duarte, S., Islam, Z.: Cross-lingual alignment and completion of Wikipedia templates. In: Proceedings of the Third International Workshop on Cross Lingual Information Access: Addressing the Information Need of Multilingual Societies, CLIAWS3 2009, pp. 21–29. Association for Computational Linguistics, Stroudsburg (2009), http://dl.acm.org/citation.cfm?id=1572433.1572437
5. Cancedda, N., Gaussier, E., Goutte, C., Renders, J.M.: Word sequence kernels. J. Mach. Learn. Res. 3, 1059–1082 (2003), http://dl.acm.org/citation.cfm?id=944919.944963
6. Fleiss, J.L.: Measuring Nominal Scale Agreement Among Many Raters. Psychological Bulletin 76(5), 378–382 (1971), http://dx.doi.org/10.1037/h0031619

7. Kontokostas, D., Bratsas, C., Auer, S., Hellmann, S., Antoniou, I., Metakides, G.: Internationalization of Linked Data: The case of the Greek DBpedia edition. Web Semantics: Science, Services and Agents on the World Wide Web 15, 51–61 (2012), http://www.sciencedirect.com/science/article/pii/S1570826812000030
8. Lodhi, H., Shawe-Taylor, J., Cristianini, N.: Text classification using string kernels. Journal of Machine Learning Research 2, 563–569 (2002)
9. Nguyen, T., Moreira, V., Nguyen, H., Nguyen, H., Freire, J.: Multilingual schema matching for Wikipedia infoboxes. Proc. VLDB Endow. 5(2), 133–144 (2011), http://dl.acm.org/citation.cfm?id=2078324.2078329
10. Palmero Aprosio, A., Giuliano, C., Lavelli, A.: Automatic expansion of DBpedia exploiting Wikipedia cross-language information. In: Cimiano, P., Corcho, O., Presutti, V., Hollink, L., Rudolph, S. (eds.) ESWC 2013. LNCS, vol. 7882, pp. 397–411. Springer, Heidelberg (2013)
11. Palmero Aprosio, A., Giuliano, C., Lavelli, A.: Automatic Mapping of Wikipedia Templates for Fast Deployment of Localised DBpedia Datasets. In: Proceedings of the 13th International Conference on Knowledge Management and Knowledge Technologies (2013)
12. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. The VLDB Journal 10(4), 334–350 (2001), http://dx.doi.org/10.1007/s007780100057
13. Rinser, D., Lange, D., Naumann, F.: Cross-lingual entity matching and infobox alignment in Wikipedia. Information Systems 38(6), 887–907 (2013), http://www.sciencedirect.com/science/article/pii/S0306437912001299
14. Saunders, C., Tschach, H., Taylor, J.S.: Syllables and other String Kernel Extensions. In: Proc. 19th International Conference on Machine Learning (ICML 2002), pp. 530–537 (2002)
15. Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. Journal on Data Semantics 4, 146–171 (2005)
16. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Proceedings of the 16th International Conference on World Wide Web, WWW 2007, pp. 697–706. ACM, New York (2007), http://doi.acm.org/10.1145/1242572.1242667
17. Sultana, A., Hasan, Q.M., Biswas, A.K., Das, S., Rahman, H., Ding, C., Li, C.: Infobox suggestion for Wikipedia entities. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM 2012, pp. 2307–2310. ACM, New York (2012), http://doi.acm.org/10.1145/2396761.2398627
18. Vrandečić, D.: Wikidata: a new platform for collaborative data collection. In: Proceedings of the 21st International Conference Companion on World Wide Web, WWW 2012 Companion, pp. 1063–1064. ACM, New York (2012), http://doi.acm.org/10.1145/2187980.2188242