Information Systems 38 (2013) 887–907

Contents lists available at SciVerse ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

Cross-lingual entity matching and infobox alignment in Wikipedia

Daniel Rinser, Dustin Lange*, Felix Naumann

Hasso Plattner Institute, Potsdam, Germany

ARTICLE INFO

Available online 23 October 2012

Keywords:
Entity resolution
Schema matching
Linked data
Data quality on the web

ABSTRACT

Wikipedia has grown to a huge, multi-lingual source of encyclopedic knowledge. Apart from textual content, a large and ever-increasing number of articles feature so-called infoboxes, which provide factual information about the articles' subjects. As the different language versions evolve independently, they provide different information on the same topics. Correspondences between infobox attributes in different language editions can be leveraged for several use cases, such as automatic detection and resolution of inconsistencies in infobox data across language versions, or the automatic augmentation of infoboxes in one language with data from other language versions.

We present an instance-based schema matching technique that exploits information overlap in infoboxes across different language editions. As a prerequisite we present a graph-based approach to identify articles in different languages representing the same real-world entity using (and correcting) the interlanguage links in Wikipedia. To account for the untyped nature of infobox schemas, we present a robust similarity measure that can reliably quantify the similarity of strings with mixed types of data. The qualitative evaluation on the basis of manually labeled attribute correspondences between infoboxes in four of the largest Wikipedia editions demonstrates the effectiveness of the proposed approach.

1. Entity and attribute matching across Wikipedia languages

Wikipedia is a well-known public encyclopedia. While most of the information contained in Wikipedia is in textual form, the so-called *infoboxes* provide semi-structured, factual information. They are displayed as tables in many Wikipedia articles and state basic facts about the subject. There are different *templates* for infoboxes, each targeting a specific category of articles and providing fields for properties that are relevant for the respective subject type. For example, in the English Wikipedia, there is a class of infoboxes about companies, one to describe the fundamental facts about countries (such as their capital and population), one for musical artists, etc. However, each of the currently 281 language versions[1] defines and maintains its own set of infobox classes with their own set of properties, as well as providing sometimes different values for corresponding attributes.

Fig. 1 shows extracts of the English and German infoboxes for the city of Berlin. The arrows indicate matches between properties. It is already apparent that matching purely based on property names is futile: the terms Population density and Bevölkerungsdichte or Governing parties and Reg. Parteien have no textual similarity. However, their property values are more revealing: ⟨3,857.6/km^2⟩ and ⟨3.875 Einw. je km^2⟩ or ⟨SPD/Die Linke⟩ and ⟨SPD und Die Linke⟩ have a high textual similarity, respectively.

Our overall goal is to automatically find a mapping between attributes of infobox templates across different

* Corresponding author. Tel.: +49 3315509282.
E-mail addresses: daniel.rinser@alumni.hpi.uni-potsdam.de (D. Rinser), dustin.lange@hpi.uni-potsdam.de (D. Lange), naumann@hpi.uni-potsdam.de (F. Naumann).

[1] As of March 2011.

0306-4379/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.is.2012.10.003

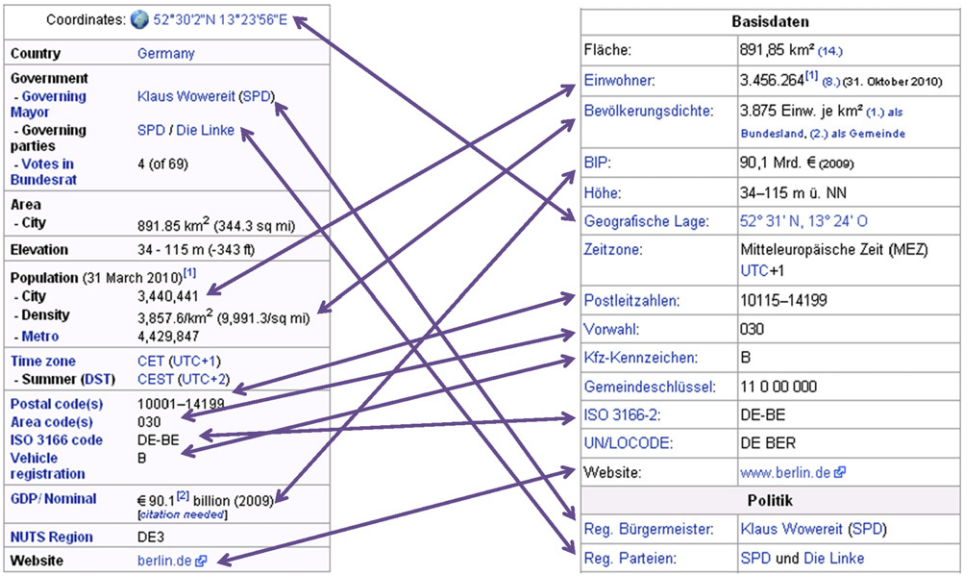

Fig. 1. A mapping between the English and German infoboxes for Berlin.

language versions. Such a mapping can be valuable for several different use cases: First, it can be used to increase the information quality and quantity in Wikipedia infoboxes, or at least help the Wikipedia communities to do so. Inconsistencies among the data provided by different editions for corresponding attributes could be detected automatically. For example, the infobox in the English article about Germany claims that the population is 81,799,600, while the German article specifies a value of 81,768,000 for the same country. Detecting such conflicts can help the Wikipedia communities to increase consistency and information quality across language versions. Further, the detected inconsistencies could be resolved automatically by fusing the data in infoboxes, as proposed by [1]. Finally, the coverage of information in infoboxes could be increased significantly by completing missing attribute values in one Wikipedia edition with data found in other editions.

An infobox template does not describe a strict schema, so that we need to collect the infobox template attributes from the template instances. For the purpose of this paper, an infobox template is determined by the set of attributes that are mentioned in any article that reference the template.

The task of matching attributes of corresponding infoboxes across language versions is a specific application of *schema matching*. Automatic schema matching is a highly researched topic and numerous different approaches have been developed for this task as surveyed in [2,3].

Among these, *schema-level matchers* exploit attribute labels, schema constraints, and structural similarities of the schemas. However, in the setting of Wikipedia infoboxes these techniques are not useful, because infobox definitions only describe a rather loose list of supported properties, as opposed to a strict relational or XML schema. Attribute names in infoboxes are not always sound, often cryptic or abbreviated, and the exact semantics of the attributes are not always clear from their names alone. Moreover, due to our multi-lingual scenario, attributes are labeled in different natural languages. This latter problem might be tackled by employing bilingual dictionaries, if the previously mentioned issues were solved. Due to the flat nature of infoboxes and their lack of constraints or types, other constraint-based matching approaches must fail.

On the other hand, there are *instance-based matching* approaches, which leverage instance data of multiple data sources. Here, the basic assumption is that similarity of the instances of the attributes reflects the similarity of the attributes. To assess this similarity, instance-based approaches usually analyze the attributes of each schema individually, collecting information about value patterns and ranges, amongst others, such as in [4]. A different, duplicate-based approach exploits information overlap across data sources [5]. The idea there is to find two representations of same real-world objects (duplicates) and then suggest mappings between attributes that have the same or similar values. This approach has one important requirement: the data sources need to share a sufficient amount of common instances (or tuples, in a relational setting), i.e., instances describing the same real-world entity. Furthermore, the duplicates either have to be known in advance or have to be discovered despite a lack of knowledge of corresponding attributes.

The approach presented in this paper is based on such duplicate-based matching. Our approach consists of three steps: entity matching, template matching, and attribute matching. The process is visualized in Fig. 2. (1) *Entity matching*: First, we find articles in different language versions that describe the same real-world entity. In particular, we make use of the cross-language links that are present for most Wikipedia articles and provide links between same entities across different language versions. We present a graph-based approach to resolve conflicts in

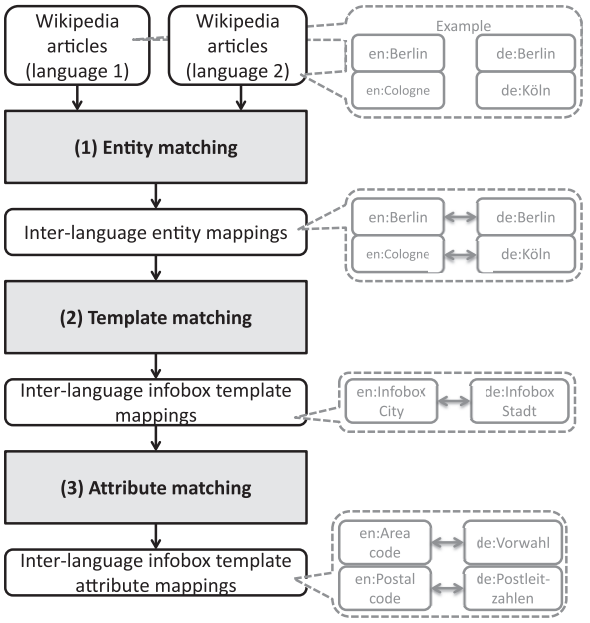

Fig. 2. Overview of our approach.

the linking information. (2) *Template matching*: We determine a cross-lingual mapping between infobox templates by analyzing template co-occurrences in the language versions. (3) *Attribute matching*: The infobox attribute values of the corresponding articles are compared to identify matching attributes across the language versions, assuming that the values of corresponding attributes are highly similar for the majority of article pairs.

As a first step we analyze the quality of Wikipedia's interlanguage links in Section 2. We show how to use those links to create clusters of semantically equivalent entities with only one entity from each language in Section 3. This entity matching approach is evaluated in Section 4. In Section 5, we show how a cross-lingual mapping between infobox templates can be established. The infobox attribute matching approach is described in Section 6 and in turn evaluated in Section 7. Related work in the areas of ILLs, concept identification, and infobox attribute matching is discussed in Section 8. Finally, Section 9 draws conclusions and discusses future work.

2. Interlanguage links

Our basic assumption is that there is a considerable amount of information overlap across the different Wikipedia language editions. Our infobox matching approach presented later requires mappings between articles in different language editions describing the same real-world entity in order to compare their attribute values. In this section we establish such a mapping, i.e., we identify groups of articles in different Wikipedia language editions that describe the same real-world entity. This problem is significantly easier than the general entity matching problem, due to the existence of so-called *interlanguage links* (ILLs) [6]. These links are community-maintained and provide a unidirectional mapping between pairs of articles in different language editions. However, several characteristics of these ILLs, such as their unidirectional nature and the existence of conflicts, entail challenges to unambiguously identify articles describing the same entity.

2.1. Interlanguage links in Wikipedia

Wikipedia defines ILLs as links between "nearly equivalent or exactly equivalent" pages in different languages [6]. They are mostly manually created by the authors of Wikipedia articles and are displayed in the sidebar accompanying every article. The main purpose of these links is to support the navigation between different language versions of articles for human readers; for example, a local topic might be covered in more detail and accuracy in the corresponding local Wikipedia edition than in other language editions. Any article in any language edition can have a list of such links, but can link to at most one article of every other language edition. Each Wikipedia edition maintains ILLs autonomously, and thus back-links are not automatically inferred (though it is generally encouraged by the Wikipedia guidelines to add such back-links manually).

ILLs might be incorrect or unsuitable to identify matching entities for several reasons:

- Vague definition of *equivalence*: Each Wikipedia community has implemented its own set of best practices and guidelines regulating structure, content, and organization of articles. Thus, in practice, the required *equivalence* of articles is often softened to *similarity*, simply because there is no "nearly equivalent or exactly equivalent" article in the other language.
- Different article granularities: in one edition, several related topics or entities may be covered by a corresponding number of different articles, while in other editions this set of topics might be covered by only a single article. Should all finer-grained articles link to the same general article in the other language? To which of the finer-grained articles should the general article link? For instance, Fig. 5 shows the German de: Alt, which had a choice of being linked to the more fine-grained concepts en: Alto or en: Contralto (and is linked to the latter).
- Homonyms are the source of erroneous ILLs, because authors often make linking decisions without regarding the actual article content, and are thus misled by syntactically similar article titles.
- Cluster size and consistency: We expect each ILL-connected subset of entities to form a cluster no larger than the number of languages considered. As we show in the next section, the transitive closure of ILLs in fact yields many clusters of much larger size.

Ultimately, many of the problems of ILLs for the task of automated entity matching are rooted in the *purpose* of those links, which is to aid navigation for human users rather than to provide an unambiguous mapping for data mining tasks. In fact, the currently implemented system for ILLs is not without controversy even within the

Wikipedia community [7] and there are several proposals for improvements [8–11]. Nevertheless, we show that ILLs indeed help in the task of entity matching using intelligent filtering for entity clusters.

2.2. Analysis of the Wikipedia interlanguage links

In this section we examine the structure of ILLs from two angles: First we classify and quantify different linkage situations, and second we regard the overall topology of ILLs. Our analysis is based on the official `langlinks.sql.gz` MySQL dumps of the English (*en*), German (*de*), French (*fr*), Italian (*it*), Dutch (*nl*), and Spanish (*es*) Wikipedia language editions [12]. The raw data have been slightly preprocessed in order to overcome some inconveniences:

- Page redirects have been resolved, that is, ILLs pointing to a redirect page were dereferenced.
- ILLs originating from or targeting a page that is not in the main article namespace [13] of the respective Wikipedia edition were discarded.
- Only links among the six language editions mentioned above have been considered.

2.2.1. Linkage situations

Let A and B be the sets of all articles in two different Wikipedia language editions. We define $\mathcal{L}_{AB}$ to be the set of ILLs from articles in A to articles in B. Since neither symmetry nor transitivity are technically enforced, we identify three possible situations, visualized in Fig. 3:

Bi-directional links: All links $\langle a,b\rangle \in \mathcal{L}_{AB}$ for which a backlink $\langle b,a\rangle \in \mathcal{L}_{BA}$ exists:

$$\mathcal{B} = \{\langle a,b\rangle \in \mathcal{L}_{AB} \,|\, \langle b,a\rangle \in \mathcal{L}_{BA}\}$$

Uni-directional links: All links $\langle a,b\rangle \in \mathcal{L}_{AB}$ for which there is no link $\langle b,a'\rangle \in \mathcal{L}_{BA}$. That is, the target article of the link does not have a link to *any* article in the source language:

$$\mathcal{U}_{AB} = \{\langle a,b\rangle \in \mathcal{L}_{AB} \,|\, \neg\exists a' \in A : \langle b,a'\rangle \in \mathcal{L}_{BA}\}$$

Conflicts: All links $\langle a,b\rangle \in \mathcal{L}_{AB}$ for which the backlink $\langle b,a'\rangle \in \mathcal{L}_{BA}$ targets a different article $a' \in A$:

$$\mathcal{C}_{AB} = \{\langle a,b\rangle \in \mathcal{L}_{AB} \,|\, \exists a' \in A : a' \neq a \wedge \langle b,a'\rangle \in \mathcal{L}_{BA}\}$$

Table 1 quantifies the degree to which each of the described link constellations occurs among all 15 pairs of the six Wikipedia language editions considered.

There are some interesting observations to be made in this data: the percentage of bi-directional links is surprisingly high throughout all considered language pairs, ranging from 97.26% (en ↔ de) to 98.26% (fr ↔ nl). Of the 9,010,048 ILLs among all six languages, there are 4,405,202 bi-directional link pairs (97.78% of all links). This suggests an overall high quality of interlanguage links, since bi-directional links are an indication for consensus between two language editions. However, in absolute numbers there is a considerable amount of conflicts, namely 115,605 conflicts between all 15 language pairs (the sum of all values in rows $\mathcal{C}_{AB}$ and $\mathcal{C}_{BA}$ in Table 1). In the next section we propose an entity matching method to resolve these conflicts by choosing the most likely correct links.

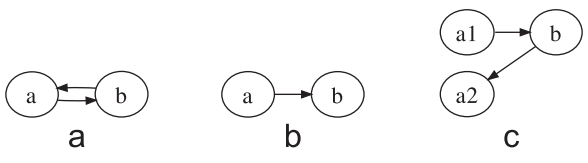

Fig. 3. Different constellations of ILLs between language pairs. (a) Bi-directional. (b) Uni-directional. (c) Conflict.

3. A graph-based approach for entity matching

Given the six language editions mentioned above, our goal is to create clusters of entities in which there is at most one entity from each language. To this end, we first analyze the existing link topology and then suggest filter conditions to remove links that conflict with our goal and retain clusters of same entities.

3.1. Analysis of the ILL topology

The interlanguage links in Wikipedia form a large directed graph, with articles from different language versions as vertices and ILLs between them as directed edges. One interesting aspect to investigate are the *connectivity* properties of this graph, or, more specifically, the amount, sizes, and properties of the *connected components* in the graph. Apart from the mere sizes of the components, a key property to analyze is whether a component contains more than one article in any given language. We call such components *incoherent* (cf. [14]) and the property itself will be referred to as a *conflict* in a component.

Obviously, all components with a size larger than the number of languages considered (in this case 6) must be incoherent, but conflicts can also occur in smaller components with a size of at least 3 (because the source and target Wikipedia editions of an ILL must differ). For the six languages included in this analysis, the graph consists of 3,402,643 vertices (articles) and 9,010,048 edges – we consider only articles that are either a source or a target of at least one ILL.

Since the graph is a *directed* (probably cyclic) graph, there are two commonly used definitions of connectivity for graphs or subgraphs:

Weak connectivity: A directed graph is *weakly connected*, iff for every pair $\langle v,w\rangle$ of vertices there is a path from v to w on the underlying undirected graph.

Strong connectivity: A directed graph is *strongly connected*, iff for every pair $\langle v,w\rangle$ of vertices there is a directed path from v to w and a directed path from w to v.

Due to the high degree of bi-directional links in the interlanguage link graph, we introduce a third definition of connectivity:

Bi-directional connectivity: A directed graph is bi-directionally connected, iff the *undirected* graph of bi-directional edges is connected. In other words, the graph is bi-directionally connected, if after removing all non-bi-directional edges and replacing the bi-directional edges with undirected edges, the resulting undirected graph is

Table 1
Number of occurrences of each link constellation between all 15 language pairs. The basis for the percentage of bi-directional links in the fourth row is the average number of interlanguage links $\mathcal{L}_{AB}$, $\mathcal{L}_{BA}$ among the two languages: $2|\mathcal{B}|/(|\mathcal{L}_{AB}|+|\mathcal{L}_{BA}|)$.

	en↔de	**en↔fr**	**en↔it**	**en↔nl**	**en↔es**
Total links $\mathcal{L}_{AB}$	516,019	532,694	409,671	379,393	338,144
Total links $\mathcal{L}_{BA}$	515,789	535,574	413,615	378,907	341,361
Bi-directional links $\mathcal{B}$	501,781	521,942	402,662	372,117	330,978
Bi-directional links $\mathcal{B}$ (%)	97.26	97.72	97.82	98.15	97.42
Uni-directional links $\mathcal{U}_{AB}$	3444	2374	1457	2027	1691
Uni-directional links $\mathcal{U}_{BA}$	5886	5986	6051	3238	5232
Conflicts $\mathcal{C}_{AB}$	10,794	8378	5552	5249	5475
Conflicts $\mathcal{C}_{BA}$	8122	7646	4902	3552	5151
	de↔fr	**de↔it**	**de↔nl**	**de↔es**	**fr↔it**
Total links $\mathcal{L}_{AB}$	309,398	238,048	238,354	189,102	275,885
Total links $\mathcal{L}_{BA}$	311,025	240,485	238,292	191,286	277,462
Bi-directional links $\mathcal{B}$	303,515	233,907	233,702	185,220	271,056
Bi-directional links $\mathcal{B}$ (%)	97.84	97.76	98.06	97.38	97.97
Uni-directional links $\mathcal{U}_{AB}$	2180	1628	1975	1536	2198
Uni-directional links $\mathcal{U}_{BA}$	3096	3762	2160	3117	3666
Conflicts $\mathcal{C}_{AB}$	3703	2513	2677	2346	2631
Conflicts $\mathcal{C}_{BA}$	4414	2816	2430	2949	2740
	fr↔nl	**fr↔es**	**it↔nl**	**it↔es**	**nl↔es**
Total links $\mathcal{L}_{AB}$	249,591	230,133	221,094	200,567	168,298
Total links $\mathcal{L}_{BA}$	248,784	231,305	219,522	200,510	169,740
Bi-directional links B	244,863	225,475	216,458	196,033	165,493
Bi-directional links B (%)	98.26	97.73	98.25	97.75	97.91
Uni-directional links $\mathcal{U}_{AB}$	2118	1927	2744	2461	1396
Uni-directional links $\mathcal{U}_{BA}$	1861	2924	1320	2281	2303
Conflicts $\mathcal{C}_{AB}$	2610	2731	1892	2073	1409
Conflicts $\mathcal{C}_{BA}$	2060	2906	1744	2196	1944

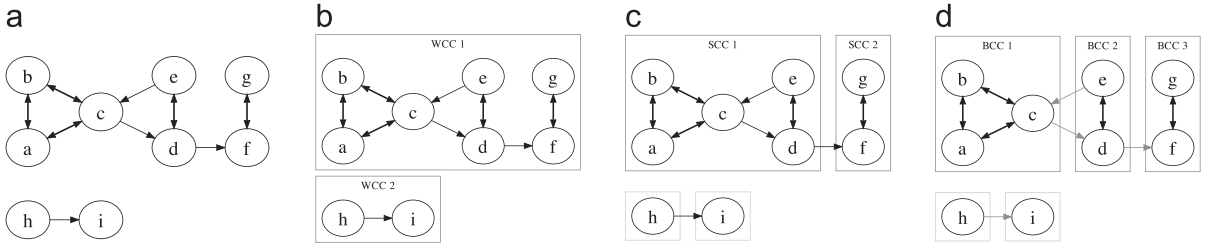

Fig. 4. Example digraph with connected components highlighted based on different definitions of connectivity. (a) Original example digraph *G*. (b) Weakly connected components in *G*. (c) Strongly connected components in *G*. (d) Bi-directionally connected components in *G*.

connected. Note that a bi-directionally connected graph is also strongly connected.

Based on the different connectivity definitions, we discuss three approaches for decomposing a directed graph into subgraphs. *Weakly connected components* (*WCCs*) are the maximal weakly connected subgraphs of a digraph; *strongly connected components* (*SCCs*) are the maximal strongly connected subgraphs; and *bi-directionally connected components* (*BCCs*) are the maximal bi-directionally connected subgraphs. Fig. 4 illustrates this: the original graph (Fig. 4(a)) is decomposed into WCCs (Fig. 4(b)), SCCs (Fig. 4(c)), and BCCs (Fig. 4(d)). To find all three types of components we employ standard $O(|\mathcal{L}_{AB}|+|\mathcal{L}_{BA}|)$ algorithms.

The example demonstrates that the graph has fewer, but larger weakly connected components than strongly connected components (by definition, every strongly connected component is also weakly connected). Furthermore, since every bi-directionally connected component is also strongly connected (but not vice versa), the third approach partitions the graph even more, resulting in more and smaller components.

Table 2 shows statistics for the three graph decomposition approaches. As expected, there are fewer weakly connected components than strongly connected components, and fewer strongly connected components than bi-directionally connected components. The more interesting numbers, however, are the sizes of the largest components and the number of incoherent components. The largest weakly connected component contains no less than 108 articles,[2] which describe clearly distinct entities, ranging from en: Joint stock company, over en: German Student Corps to en: Uncle and en: Sister. This size, and the fact that in total there are 40,590 WCCs

[2] More specifically, it contains 26 English, 26 German, 21 French, 13 Italian, 13 Dutch, and nine Spanish articles.

Table 2
Numbers of connected components in the ILL graph.

	WCCs	SCCs	BCCs
Number of components	1,062,641	1,067,753	1,068,192
Total pages in components	3,402,643	3,319,822	3,319,055
Average component size	3.202	3.109	3.107
Largest component	108	17	15
Number of incoherent components	40,590	3,469	2,980

Note: Only components with size >1 are considered—that is also why the numbers of pages in the components differ across the approaches.

containing multiple articles for at least one of the languages (though most of them are not nearly as large), shows that there is a large amount of inaccurate (or at least imprecise) links that collectively establish paths between clearly unrelated entities.

While the situation improves dramatically when requiring strong or even bi-directional connectivity between articles, many incoherent components remain. Nevertheless, a closer look at the larger incoherent SCCs and BCCs reveals that in almost all cases, strongly or bi-directionally connected articles center around roughly the same topic, as shown in the incoherent SCC of Fig. 5. The example illustrates different article granularities as a common source of conflicts: the German article covers both *alto* and *contralto*, while the other languages cover them in separate articles.

Note also that the number of Wikipedia editions taken into account in this analysis (six) is rather small. While this obviously does not affect the results of the analysis of links between language pairs, it has a massive impact on the *accumulation* of conflicts: analyzing the Wikipedia data from August 2008, Bolikowski has shown that the largest weakly connected component in the *complete* ILL graph (covering about 250 languages) consists of 72,284 articles [14].

The amount of conflicts and the surprisingly complex structure of the ILL-graph show that the identification of conflict-free sets of articles describing the same real-world entity is not as trivial as it might have appeared at first glance. However, the findings of this analysis help to design an appropriate approach, presented next.

3.2. Whittling down large clusters

We describe a multi-step, graph-based approach to identify articles describing the same real-world entity that leverages ILLs. We use the term *cluster* to describe a set of articles. There are two requirements for our entity matching algorithm, which both aim to prohibit ambiguity in the resulting clusters:

1. The clusters have to be *disjoint*: an article may only appear in one cluster.
2. The clusters have to be *coherent*: a cluster may contain at most one article from each language.

Obviously, there is a straightforward way to build such clusters that obeys the requirements: decompose the inter-language link graph into connected components (according to one of the above described definitions of connectivity)

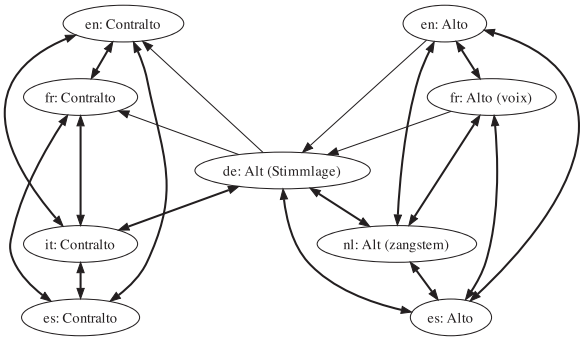

Fig. 5. Real-world example of an incoherent strongly connected component.

and discard all incoherent components, as proposed by Adar et al. [15]. However, throwing away those components means a large loss of potential concepts, especially because components containing a conflict tend to be much larger than coherent components. It would be desirable to find an approach that decomposes incoherent components into smaller, coherent components.

The basic idea of our approach is to decompose the ILL graph into strongly connected components (SCCs), and to resolve conflicts by partitioning incoherent SCCs further using two even stricter connectivity measures. Next, each step of the approach and the underlying motivation is explained in detail; an overview of the steps is shown in Fig. 8, which also includes the exact number of components after each step.

Step 1: *Decompose ILL graph into SCCs*. In general, all connectivity definitions described earlier (weak, strong, and bi-directional) form a reasonable basis to decompose the ILL graph into subgraphs (components). However, weak connectivity is the loosest form, and can lead to quite large components spanning several completely unrelated topics. While in some cases it might be advantageous to incorporate sparsely linked articles into a component (e.g., in the case of new articles that are not yet fully integrated into the network of ILLs), the fact that an article (or a group of articles) is attached to other articles by only a single, uni-directional link is often an indication of this link being incorrect or at least imprecise.

On the other hand, bi-directional connectivity is a rather strong requirement for clusters. Fig. 6 shows two real-world examples illustrating that requiring bi-directional connectivity is sometimes too strict. Especially in the first example, there is a very strong linkage between all articles in the

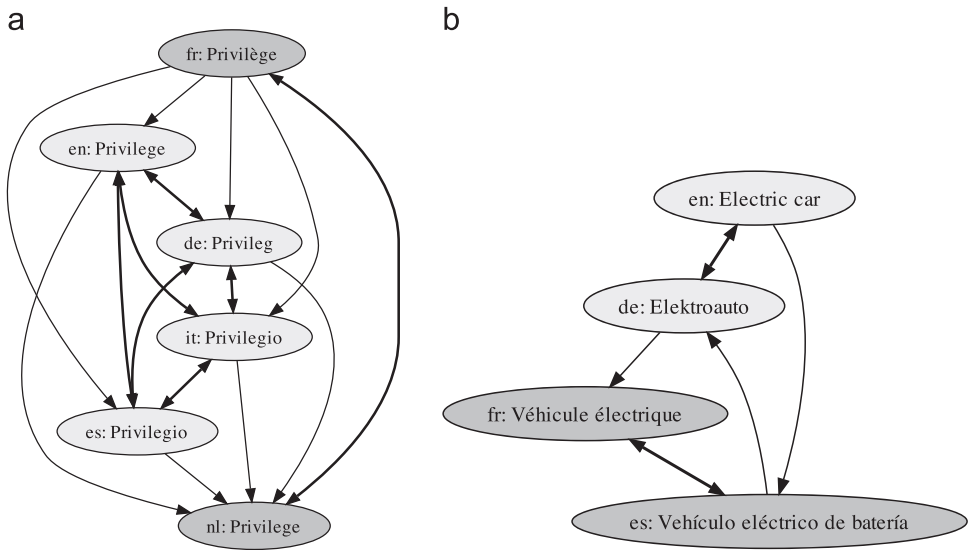

Fig. 6. Two real-world examples illustrating that bi-directional connectivity is sometimes too strict.

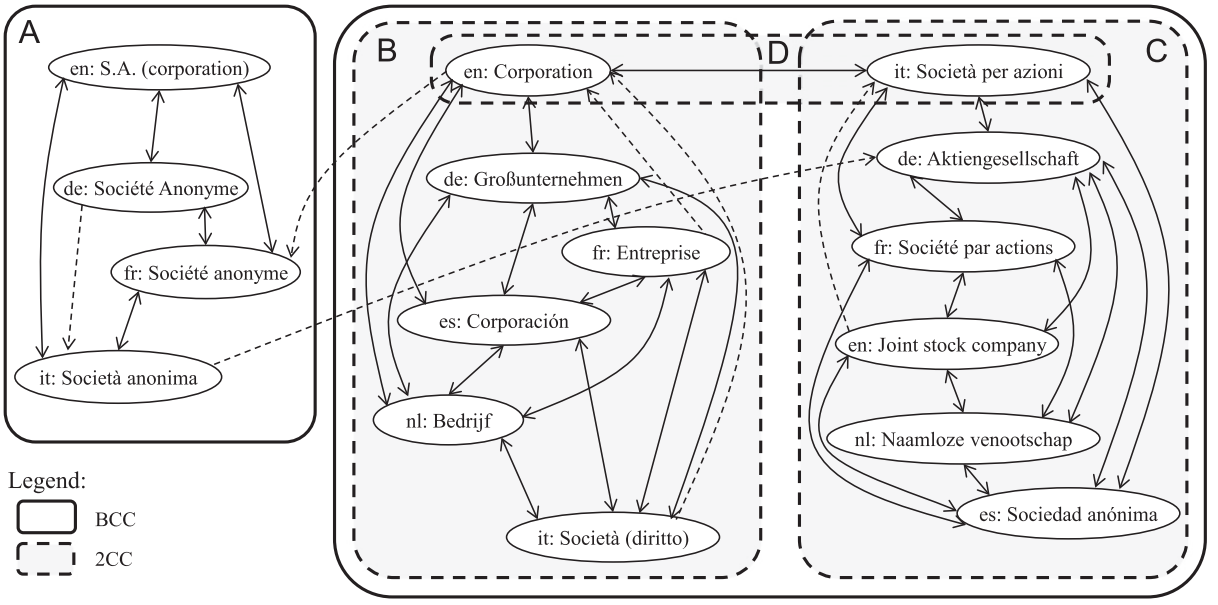

Fig. 7. The "Corporation" example demonstrates the step-wise decomposition of conflicted graphs.

component. Nevertheless, in both examples, the components (which are strongly connected) would each be split into two separate bi-directionally connected components.

Decomposing the graph into strongly connected components, in contrast, is a good compromise with regard to strictness. All coherent SCCs are added to the final set of clusters, while the rest (3469 incoherent components) is processed further in the following steps.

Step 2: *Decompose incoherent SCCs into bi-directionally connected components*. Fig. 7 shows an example of one of the largest incoherent SCCs (subsequently referred to as the "Corporation" example). Decomposing this component into bi-directionally connected components leaves us with two such components. The first component on the left is coherent and is thus added to the set of final clusters. The second component, however, is still quite large and contains two articles in each language. Therefore, it is processed further in Step 3.

Overall, from the 3,469 incoherent components we generated 4241 BCCs, of which 2980 are again incoherent and are processed in the next step.

Step 3: *Decompose incoherent BCCs into bi-connected components* (2CC). To further decompose incoherent BCCs, an even stricter connectivity constraint is applied: *Bi-connectivity* (also known as 2-*connectivity*) is defined for *undirected* graphs and requires that each pair of vertices

$\langle v,w \rangle$ is connected through at least *two vertex-independent paths*. In other words, a graph is bi-connected, iff it has no cut vertices—vertices whose removal would disconnect the graph. For the *directed* graph of ILLs, we define bi-connectivity on the underlying undirected graph that is formed by retaining only the bi-directional edges of the original graph as undirected edges.

Analogous to the definition of other connected components, a *bi-connected component* (2CC) is a maximal bi-connected subgraph of the original graph. One characteristic of bi-connected components is that they are edge-disjoint, but not necessarily vertex-disjoint. More precisely, all cut vertices of the graph are contained in at least two 2CCs. This is a problem with respect to the first requirement (disjoint clusters) and entails that we cannot simply adopt all 2CCs as clusters. We tackle this problem by finding a (preferably maximal) subset of 2CCs that are mutually vertex-disjoint. This is accomplished by selecting the combination of 2CCs with the maximal number of total vertices from the set of all possible 2CC combinations (that is, all combinations of 2CCs that are vertex-disjoint).

For the "Corporation" example, Fig. 7 shows the BCC *A* and the three 2CCs *B*, *C*, and *D*—note that en: Corporation and it: Società per azioni are cut vertices and as such each belongs to two components. There are four possible combinations of 2CCs that do not involve shared vertices: {*B*}, {*C*}, {*D*}, and {*B*,*C*}. Of these combinations, the latter incorporates considerably more pages than the other three, hence this subset of 2CCs is selected and the contained components (*B* and *C*) are added to the final set of clusters. The resulting, now coherent components are thus *A*, *B*, and *C*.

A schematic summary of the complete algorithm, with focus on the flow of components, is presented in Fig. 8. The numbers in parentheses denote the aggregated number of components of each type. These numbers show that the 3469 incoherent SCCs after Step 1, rather than being discarded, are finally decomposed into 5664 coherent components—894 in Step 2 and another 4770 in Step 3.

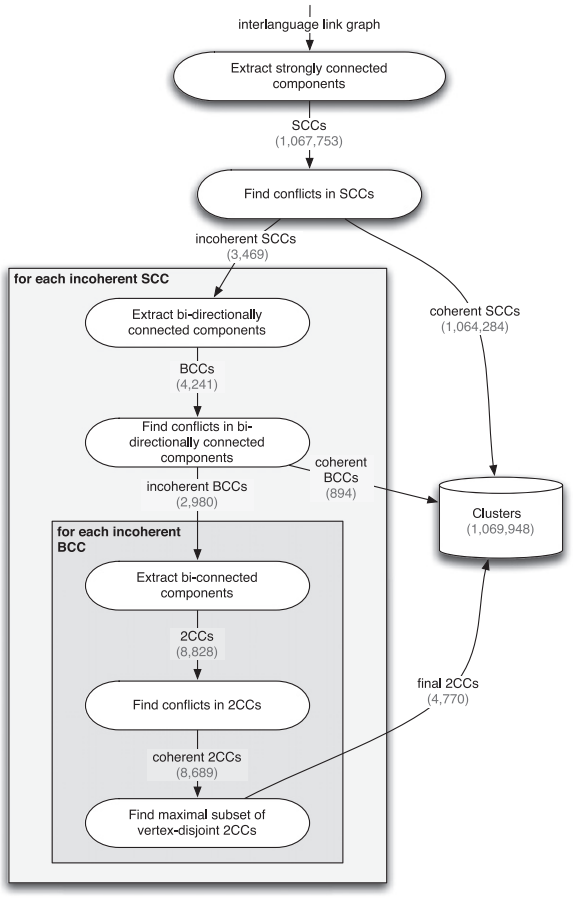

Fig. 8. Schematic summary of the graph-based entity matching algorithm. The numbers in parentheses denote the aggregated number of components of each type.

4. Evaluation of entity matching

A *representative* qualitative evaluation of the clusters produced by our approach would require the manual validation of a significant portion of the clusters, which is very labor-intense, because the article contents have to be read and evaluated (a simple comparison of the article title is not sufficient). Thus, we perform a structural and theoretical evaluation of the results.

As we describe in Section 8, most prior approaches rely solely on the extraction of weakly connected components to identify clusters. Only Adar et al. acknowledge incoherent components (by completely discarding them) [15], while others do not tackle this problem at all [16,17]. Either alternative bears its problems: discarding all incoherent components means a large loss of potential clusters, while retaining them unchanged leads to ambiguities in the clusters and huge groups of supposedly related articles. Kinzler, in contrast, presents the most innovative and promising approach so far [18] (see Section 8 for details). Therefore, we compare the clusters produced by our algorithm with those of a re-implementation of Kinzler's algorithm fed with the same ILL data. Table 3 shows basic statistics of the two approaches.

Table 3
Statistics comparing our approach and Kinzler's algorithm.

	Our approach	Kinzler
# **concepts**	1,069,948	1,069,576
# **pages in concepts**	3,316,247	3,317,752
Average concept size	3.099	3.102
# **complete cliques**	1,055,859	1,054,099

The numbers reveal that both approaches are very similar with respect to the structural properties of their resulting clusters. Both approaches overlap in 1,066,249 entities, leaving 3327 entities exclusively identified by our approach and 3716 exclusively by Kinzler's. Kinzler's algorithm incorporates slightly more articles in slightly fewer clusters, hence producing marginally larger clusters.

As Kinzler himself has observed, the result of his iterative approach depends strongly on the order in which clusters are merged. Lacking obvious choices for the order, he does not suggest any specific order, and we likewise do

not impose one. This can often lead to non-optimal results, especially if there are several article candidates for the same language. In this scenario, if the "wrong" alternative is picked first, it "poisons" the cluster and prevents the better alternative of the same language from being included into the cluster. Our approach, in contrast, works top-down taking a more holistic view on the graph, and is thus not dependent on some merge order.

To demonstrate this difference, we have applied both entity matching algorithms to the "Corporation" example from Fig. 7. Both algorithms resolve the incoherent linkage situation by producing several smaller clusters. However, the two approaches group articles differently: in contrast to our approach (with resulting clusters *A*, *B*, and *C* shown in Fig. 7), Kinzler incorporates the article it: Società per azioni instead of it: Società (diritto) into cluster *B*, although the latter is much more interlinked with the other articles in this cluster. This has an impact on cluster *C*, too, because the article it: Società per azioni (which would fit very well into this cluster) is already assigned to cluster *B*. Moreover, it: Società (diritto) is not included in any cluster, because cluster *B* already contains an Italian article and the articles in cluster *C* do not directly link to this article. The reason for this result is the merge order: the algorithm has merged en: Corporation and it: Società per azioni first, thus preventing a segmentation that would be superior with regard to the *overall* interlinkage of the articles. Our graph-based approach does not suffer from this issue, as it correctly recognizes the clusters *A*, *B*, and *C*.

In summary, the high degree of consensus between the two algorithms, despite their contrary approaches (top-down versus bottom-up), suggests an overall good quality of both. The main difference is their handling of conflicts: Kinzler's algorithm tries to maximize the cluster sizes and sometimes groups sparsely interlinked articles together due to an unfavorable merge order. On the other hand, our algorithm tries to find the most strongly interlinked sub-components, sometimes at the expense of cluster sizes. Both approaches run in under 1 min.

Note that both approaches rely on the ILL structure in Wikipedia, and thus their quality is directly dependent on the quality of the interlanguage links. Though individual errors in the link structure are often reliably recognized and hence do not have a strong impact on the clusters, both approaches require an overall good ILL graph: related articles need to be interlinked stronger than unrelated articles. Enriching ILLs in Wikipedia by analyzing the actual content of multilingual articles has already been approached by other researchers [19].

5. Infobox template matching

After having successfully identified articles describing the same real-world entity, we can now proceed with the actual infobox attribute matching approach. The goal of this approach is to compute a mapping between corresponding attributes for pairs of corresponding infobox templates in different languages. First we take a closer look at infoboxes and their attributes, as well as at the challenges that infobox data poses. In this section, we show how a cross-lingual mapping between infobox *templates* can be established, before the actual infobox *attribute* matching approach is presented in detail in Section 6.

5.1. A closer look at infoboxes

Since infobox data differs in many important aspects from data stored in structured data sources, such as relational or XML databases, it is imperative to be aware of the consequences of these differences for extensional schema matching. In this section, we clarify the main terminology concerning infoboxes used in the sequel of this chapter. We also analyze infoboxes, their attributes, and their usage in articles from a statistical point of view.

Fig. 9 shows the upper part of the rendered en: Infobox company in the en: BMW article in the English Wikipedia and the corresponding portion of the infobox template invocation in the article's wikitext. As we can see in the wikitext, infobox instances state the *name of the invoked template* (*Infobox company*), and contain a *set of attributes*, each consisting of a name (for example *foundation*) and an associated value (1916). The attribute values are not typed and can in fact contain arbitrary portions of wikitext, such as wiki links (`[[Automotive industry]]`), image references (`[[Image:BMW Logo.svg|160px]]`), invocations of other templates (`FWB|BMW`), text, numbers, dates, or any combination of these types. The example also demonstrates that the attribute names do not necessarily correspond with the display labels in the rendered infobox. In the majority of infobox templates, many attributes are optional.

An *infobox template* defines the structure and layout for a specific type of infoboxes. Hence, it establishes a class of infoboxes of the same type. Infobox templates are usually used by several different articles, while each article defines different values for the template's attributes. For example, there is an en: Infobox company template in the English Wikipedia, which is included in several thousand articles about individual companies, each of them providing appropriate values for the attributes. Such an inclusion of an infobox template in an article, with specific values for a template's attributes, will be denoted as an *infobox instance*. Thus, the example shown in Fig. 9 is an instance of the en: Infobox company template. Similarly, infobox *attributes* are the formal attributes that an infobox template defines (industry or founder in the example). Infobox instances assign *values* to attributes, thus forming *attribute instances* (for example, foundation = 1916 is an attribute instance).

```
{{Infobox company
|company_name    = Bayerische Motoren Werke AG
|company_logo    = [[Image:BMW Logo.svg|160px]]
|company_type    = [[Aktiengesellschaft]]
                  ({{FWB|BMW}})
|industry        = [[Automotive industry]]
|foundation      = 1916
|founder         = [[Franz Josef Popp]]
}}
```

Fig. 9. Example wikitext of an infobox template.

Table 4
Some statistics about infobox templates and their usage in the four language editions.

	en	de	fr	nl
Articles (no redirect pages)	3,431,632	1,100,058	1,077,530	661,561
Articles with infobox	1,021,951	205,439	253,514	206,042
Articles with infobox (%)	29.78	18.68	23.53	31.14
Infobox templates	2247	675	806	959
Articles per infobox template (average)	455	304	315	215
Articles per infobox template (median)	14	31	23	15

Naming conventions allow the identification of infobox templates in a wiki page's source code. We ignore the rare cases in which known infoboxes do not follow these conventions. In our own experience, this is the case for less than 1% of the infobox templates.

The purpose of Wikipedia infobox templates is *not* to provide a structured, machine-readable representation of the data, but is rather purely based on reuse and layouting considerations. This has implications on the design of infobox templates and thus on their suitability for extensional schema matching.

A major challenge is that infobox templates define only a rather *loose schema*: the set of allowed attributes is only implicitly given by the template *implementation*; in many cases this implementation allows alternative names for attributes. Moreover, attributes are untyped and unconstrained. In fact, attribute values can contain any type of wikitext, and very often they contain a mixture of different types of data, such as text, numbers, dates, wiki links, and even style markup (for example: `operating_income= 289 million <small> (2009) </small>`). In our specific setting of cross-lingual matching of infoboxes, another challenge arises: the data sources whose schemas are to be matched are composed of different (natural) *languages* and use different *formatting* conventions (numbers, dates, units).

Earlier, we have seen that different article granularity is an issue for entity matching. A similar problem arises in the context of infobox attribute matching: the attributes in different language editions often expose a very *different granularity*. For instance, in the context of countries, the English Wikipedia distinguishes between population_estimate, population_estimate_year and population_estimate_rank, while the German Wikipedia encodes all this information in a single attribute EINWOHNER. There may also be *hidden dependencies* between different attributes, for example between the population of a country and the year from which this number originates. Hence, both combinations, 2,014,345 (2006) and 2,143,271 (2009) for the same subject may be absolutely correct and not constitute a conflict.

Another problem arises when *lists of multiple values* for one property are split across multiple separate attributes, for example ruling_party1, ruling_party2, ruling_party3. This is obviously a problem for schema matching, because the *order* of the values is not necessarily consistent across languages.

Table 4 shows some general statistics about the number of articles and infoboxes in the four Wikipedia editions, as well as about the number of distinct infobox templates and their inclusion in articles (lower part).

The striking divergence between the average and the median number of articles per infobox template indicate a long tail of infobox templates. In fact, the number of articles per infobox template is inversely proportional to its statistical rank, thus the distribution follows a power law. Many infobox templates are used only once, caused by infobox templates that are too specific, abandoned, or new.

Another interesting aspect is the number and distribution of used attributes across infoboxes. Table 5 presents the total number of attribute instances in the four language editions, as well as the average and maximum numbers of attributes per infobox instance.

5.2. Template matching approach

Due to the autonomy of Wikipedia editions, each has its completely independent set of (infobox) templates. Thus, before we can match infobox attributes across language versions, we need a mapping between corresponding infobox templates for all language pairs, i.e., pairs of infobox templates that describe similar properties about the entity. For instance, it is not useful to match the attributes of en: Infobox mountain and de: Infobox Album.

Similar to articles, the granularity of infobox templates is not consistent across language editions; there often is no clear one-to-one mapping between those templates. For example, the English Wikipedia uses the en: Infobox settlement for practically all cities in the world. Other Wikipedias, however, often have customized infobox templates for cities in different countries (for example, de: Infobox Ort in Schweden, de: Infobox Ort in Japan, etc. in the German Wikipedia and nl: Infobox plaats in Australië, nl: Infobox gemeente Colombia, etc. in the Dutch Wikipedia).

From a technical point of view, template definitions are ordinary wiki pages that reside in a separate template namespace.[3] Being ordinary pages, they can themselves contain interlanguage links. However, the degree of interlinkage of templates via ILLs is lower and varies strongly across languages. A much simpler and more reliable approach to find corresponding templates is to count co-occurrences of infobox templates leveraging the entity

[3] http://en.wikipedia.org/wiki/Wikipedia:Template_namespace.

matching of the previous sections. More precisely, we define two infobox templates T_a in language version W_a and T_b in language version W_b to co-occur, if:

- T_a is used in an article A_a in language version W_a, and
- T_b is used in an article A_b in language version W_b, and
- A_a and A_b refer to the same entity as determined by entity matching.

For instance, the English article en: Allianz uses en: Infobox company and the German article de: Allianz SE uses de: Infobox Unternehmen, and both articles describe the same entity, hence both infobox templates co-occur in this case. We count such co-occurrences for each pair of infobox templates. Table 6 shows the top 20 co-occurring infobox templates between the English and German language editions along with the respective co-occurrence count.

Table 5
Basic statistics about infobox attribute-value instances in the four language editions.

	en	de	fr	nl
Sum	26,106,574	4,460,167	5,109,543	4,019,835
Avg	26	22	20	20
Max	246	119	220	98

Table 6
Top 20 infobox co-occurrences between English and German Wikipedia.

English infobox	German infobox	Count
German location	Gemeinde in Deutschland	11,948
French commune	Gemeinde in Frankreich	6238
Film	Film	6041
Settlement	Ort in den Vereinigten Staaten	4551
Italian comune	Gemeinde in Italien	4153
Musical artist	Band	3483
Football biography	Fußballspieler	3128
Ice hockey player	Eishockeyspieler	2478
Settlement	Ort in Polen	2035
Settlement	Ort in Tschechien	1923
Album	Musikalbum	1897
Ort in Österreich	Gemeinde in Österreich	1568
Mountain	Berg	1550
Planet	Asteroid	1415
Football biography 2	Fußballspieler	1339
Aircraft Begin	Flugzeug	1227
Airport	Flughafen	1146
U.S. County	County (Vereinigte Staaten)	1073
Automobile	PKW-Modell	915
VG	Computer- und Videospiel	912

Just like regular Wikipedia pages, templates can also be redirected. For instance, there is an en: Infobox company template, and there also is an en: Infobox Company template (note the capitalization), which redirects to en: Infobox company. Prior to building the infobox template mapping, template redirects are resolved by recursively translating all redirecting templates to their redirect target.

As with the distribution of infobox templates within individual Wikipedia editions, the noise ratio increases significantly towards the lower end of the co-occurrence distributions. This is mainly due to different semantic interpretations of entities (for example, the English en: PostScript article uses en: Infobox programming language, while the German de: PostScript article includes de: Infobox Dateiformat (file format)). Noise is also caused by incorrect entity mappings, articles with multiple infoboxes, and simple errors or typos in the infobox name. To filter out those noisy infobox template pairs, we apply two thresholds:

1. An absolute threshold of at least five co-occurrences.
2. A relative threshold of at least 20% co-occurrences:

$$\frac{cofreq(T_a,T_b)}{\min\{freq_a(T_a),freq_b(T_b)\}} \geq 0.2 \quad (1)$$

where $cofreq(T_a,T_b)$ is the co-occurrence frequency of templates T_a and T_b, and $freq_a(T_a)$ and $freq_b(T_b)$ are the individual frequencies of those templates that co-occur with at least one other template.

Note that even after these filtering steps, there is an $m:n$ relationship between corresponding templates. E.g., a template en: Infobox company might correspond both to de: Infobox Organisation and de: Infobox Firma, while the latter might also correspond to en: Infobox corporation. This is intended; after all, we are interested in all attribute mappings between all templates that are somehow related.

Table 7 shows some basic statistics about the co-occurrences of infobox templates between the six language pairs, after resolving template redirects and applying the thresholds described above. Apart from quantifying the degree of infobox co-occurrence between the different language pairs, the numbers allow for another interesting observation: similar to the distribution of infobox templates within individual Wikipedia editions, also for infobox co-occurrences there is a large divergence between the average and the median number of article pairs per infobox template pair (and this is after filtering out the

Table 7
Infobox co-occurrence statistics after resolving template redirects and applying thresholds.

	en↔de	en↔fr	en↔nl	de↔fr	de↔nl	fr↔nl
Infobox template pairs	456	627	603	334	353	419
Article pairs	91,728	122,164	104,864	54,251	50,048	74,425
Article pairs (average)	201	195	174	162	142	178
Article pairs (median)	23	19	18	20	21	20

most infrequent pairs via the thresholds). This imbalance poses a challenge for the attribute matching algorithm, because it entails that for the majority of infobox template pair only few article pairs can serve as instances.

6. Infobox attribute matching

As described above, the main idea of the attribute matching approach is to analyze and compare the values of attributes across infobox template pairs in different language editions. The attribute matching is done separately for each pair of matched infobox templates, and the goal is to produce a 1 : 1 attribute mapping between corresponding attributes for each of these template pairs. Of course, an individual attribute might be mapped to different attributes in different templates.

6.1. Process overview

The input to our attribute matching system is a set of co-occurring infobox *instance* pairs for a given template pair. Consider the pair of corresponding infobox templates en: Infobox country and de: Infobox Staat. To perform attribute matching we are interested in all article pairs $\langle A_{en}, A_{de} \rangle$, where A_{en} uses en: Infobox country, A_{de} uses de: Infobox Staat, and both articles describe the same entity. One example of such an article pair is ⟨en: Argentina, de: Argentinien⟩. Given that both articles describe the same real-world entity, we can assume that the values of corresponding attributes are often same or similar. Even if this is not the case for this particular article pair, we assume that it is the case for a sufficient number of other pairs.

Fig. 10 shows an overview of our matching approach. After selecting the pairs of corresponding articles as described above, we preprocess all infobox attribute values in all article pairs to remove noise, such as formatting instructions, comments, whitespace, links, or footnotes.

Next, we compute pair-wise similarities for all attribute combinations for each of the infobox instance pairs. Fig. 11 shows a fictitious example of two infobox instances with

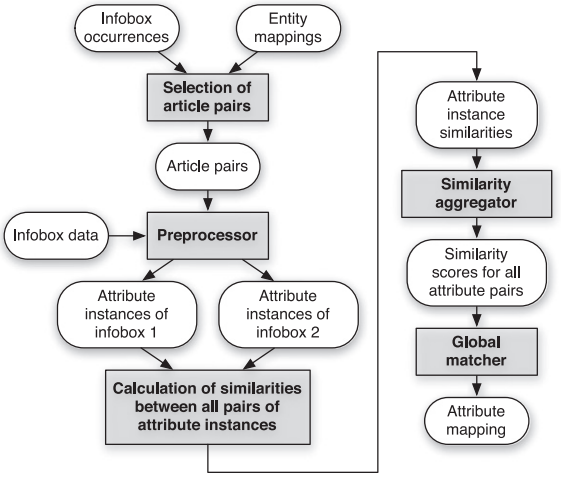

Fig. 10. Overview of the matching process.

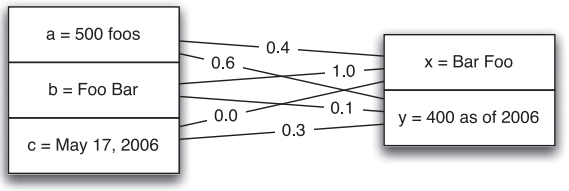

Fig. 11. Attribute instance similarities.

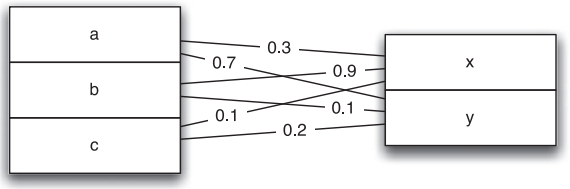

Fig. 12. Attribute-level similarities.

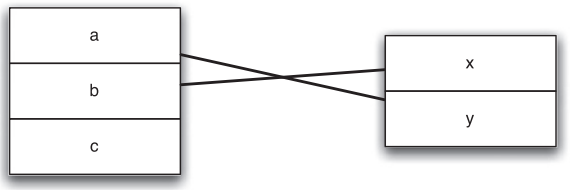

Fig. 13. Final mapping of attributes.

similarity scores for all pairs of attribute instances. These pair-wise instance similarities are computed for all article pairs in which the infobox templates under consideration co-occur. The similarity measure used to compare two attribute instances is described in Section 6.2.

Having determined the pair-wise similarities of attribute instances, we aggregate the obtained similarity scores for each attribute pair of corresponding templates. The result is a similarity score for each pair of attributes of the two infobox templates, as outlined in Fig. 12. The details of this aggregation as well as further aspects that influence the attribute-level similarity scores are explained in Section 6.3.

In a last step, the attribute-level similarity scores are used to derive correspondences between the attributes, that is, a one-to-one mapping of corresponding attributes in the two infobox templates is generated (Fig. 13). An in-depth explanation of this *global matching* is presented in Section 6.4.

6.2. Similarity of attribute values

A core task of the matcher is to determine how similar two attribute instances are. Since we do not want to rely on mere equality of attribute values, a suitable measure that quantifies the degree of similarity has to be found. First we identify the challenge caused by the untyped nature of infobox attributes, and proposes a solution that involves the separation of (implicit) data types contained in attribute values. These different data types are then examined individually to determine the similarity

between attribute values. Next, we identify wiki links and external links in attribute values as a further valuable feature. Finally, all individual similarity features are combined into one overall similarity score for a pair of attribute instances.

6.2.1. Separating data types

Wikipedia infobox attribute values do not have an explicit data type, and often contain a mixture of numbers, dates, and text. This heterogeneity makes an effective determination of similarity between these values considerably more difficult. There are many established string similarity measures, such as the Levenshtein Distance [20] or the Jaro similarity measure [21] (for a comprehensive overview and evaluation of string similarity measures, see [22]). While these measures produce good results to approximately match strings, they are not suited for comparing other types of data, especially numeric values (19 vs. 20) and dates ("30.12.1979" vs. "01.01.1980"). Moreover, both numbers and dates are often formatted differently depending on the local context. Thus, the similarity of "123,456" appearing in the English Wikipedia and the same string in the German Wikipedia should be quite low (due to the different representations of thousand separators and decimal points). On the other hand, "January 25, 1980" is semantically equivalent to "25.01.1980", which should be reflected by a maximum similarity score.

To tackle these problems, one could try to detect the predominant data type for each attribute value, and adapt the computation of similarity depending on this data type. However, since infobox attribute values frequently feature a combination of data types, it is often not feasible to find one single data type that fits the attribute value well. Instead, we decompose the attribute value strings into their number, date, and string portions, and apply different similarity measures for each type of data. Afterwards, the individual similarity scores for each data type are averaged, weighted with the character fraction that the instances of the respective data type occupy in the original string.

To demonstrate this method, Listing 1 shows an example of an attribute value with different intermingled data types. First, we detect all substrings that might represent a date and try to parse them with a very robust heuristic-driven parser, which does not rely on fixed date formats.[4] The successfully parsed dates are stored and the original substring representing the date is removed from the attribute value. The same is then done for numbers. Here, of course, we consider the locale of the respective Wikipedia language for parsing the number strings. Moreover, common exponential suffixes, such as "millions" or "bn" (and the corresponding variations in other languages), are reliably detected and resolved (that is, "3.4 bn" is interpreted as 3.4×10^9). The remaining part of the attribute value is the text portion. Table 8 shows the resulting data for our example attribute value and the fraction of characters for that type as weight.

[4] Our implementation uses the parser of the `DateTime` class in the POJava library, see http://pojava.org/.

Table 8
Separation of different data types (dates, numbers, text) for the example attribute value shown in Listing 1.

Data type	Data	Weight
Dates	`Sat Jun 14 00:00:00 CET 1777` `Mon Jul 04 00:00:00 CET 1960`	25/76
Numbers	`13.0` `50.0`	4/76
Text	`(original-star version)` `(current-stars version)`	47/76

Listing 1. Example attribute value (taken from the Adoption attribute of the English **en: Flag of the United States** article) with intermingled data types.

```
June 14, 1777 (original 13-star version) July 4,
    1960 (current 50-stars version)
```

6.2.2. A data type-aware similarity measure

Having separated the different data types, we can now calculate the similarity between pairs of attribute instances by determining individual scores and aggregating them to an overall score.

Similarity of the string portions: Much research has been conducted on determining approximate string similarities. The most commonly used measures can be classified into three categories: character-based, token-based, and hybrid approaches [22]. Many infobox attribute values are rather short strings, but there is also a good portion of them that is longer and contains continuous text rather than single words. For this reason, token-based methods outperform character-based similarity measures in our scenario. However, the most important characteristic of our use case is the cross-lingual nature: the strings we need to compare are composed of different natural languages. Thus, using words as tokens is not use useful and we use n-gram tokens, as they are much less sensitive to language differences (at least with the Indo-European family languages considered here).

As a result of these considerations, we employ Jaccard similarity with character-level n-grams ($n=2..4$) to compare the string portion of attribute values with token sets T_1 and T_2:

$$sim_{\text{string}}(s_1,s_2) = \frac{|T_1 \cap T_2|}{|T_1 \cup T_2|} \tag{2}$$

Similarity of the numeric portions: A simple and in our experiments effective approach is to calculate the ratio between the absolute values of the numbers. To guarantee that the resulting value is in the range [0,1], we divide the smaller number by the larger number.

However, since numbers often describe hard facts where there is no room for deviations, we want to "reward" two numbers for being *exactly equal*. For example, it is unlikely that an attribute describing the number of doors of a car with a value of `3` relates to an attribute in the other language with a value of `4`, although the ratio between those numbers is quite high (0.75). Thus, we introduce a 50% penalty for numbers that are not exactly

equal. Hence, the similarity function for two numbers n_1 and n_2 is defined as

$$sim_{\text{num}}(n_1,n_2)=\begin{cases}1, & \text{if } n_1=n_2\\ \frac{1}{2}\cdot\frac{\min\{|n_1|,|n_2|\}}{\max\{|n_1|,|n_2|\}}, & \text{otherwise}\end{cases} \quad (3)$$

Since we extract not only one number per attribute, but rather all contained numbers, we need to compare two *sets* of numbers, N_1 and N_2. To use sim_{num} we first have to select adequate pairs of numbers $\langle n_1,n_2\rangle \in N_1 \times N_{2/\text{mml:mn}>}$ from the two values. We do this by finding a one-to-one mapping between the numbers in both sets, such that the total similarity between the pairs is maximal (greedily approximated due to the computational complexity of this *assignment problem*). Our algorithm is based on the token assignment approach of the TagLink similarity measure ([23], referred to as *Algorithm*1): First, we compute the pairwise similarity sim_{num} for all pairs of numbers $\langle n_1,n_2\rangle$, and sort the number pairs by their similarity in descending order. Then we select the pair $\langle b_1,b_2\rangle$ with the highest similarity, and remove all other pairs $\{\langle n_1,n_2\rangle | n_1=b_1 \vee n_2=b_2\}$ from the list of pairs. This is repeated, until there are no pairs left in the list. The resulting set of matching pairs is M_n. This method is also referred to as the *greedy algorithm* for maximum weight matching [24].

Given the mapping of numbers, the most intuitive way to compute the overall similarity of the two sets of numbers, is to simply calculate the arithmetic mean of the similarities of matched numbers. This, however, does not account for unmatched numbers in either of the sets. If, for example, $N_1=\{3\}$ and $N_2=\{1,2,3,4,5\}$, we would find one perfect match $\langle 3,3\rangle$. Concluding that the similarity of N_1 and N_2 is 1.0, however, would be unnatural because there are four unmatched numbers in N_2. Thus, we define the similarity between two sets of numbers as

$$sim_{nums}(N_1,N_2)=\frac{\sum_{\langle n_1,n_2\rangle\in M_n} sim_{num}(n_1,n_2)}{\max\{|N_1|,|N_2|\}} \quad (4)$$

Similarity of the date portions: We define the similarity of two dates as their absolute difference (in days) relative to the maximum range of dates in the two attributes:

$$sim_{date}(d_1,d_2)=1-\frac{|d_1-d_2|}{maxDate-minDate} \quad (5)$$

where *minDate* is the earliest date found in any attribute instance of either of the two attributes that are being compared, and *maxDate* is the latest such date. The difference between them, again, is expressed in days.

To compare two *sets* of dates D_1 and D_2, the same approach as for numbers is used. We first create a one-to-one mapping M_d between dates in the two sets and calculate the final similarity as

$$sim_{dates}(D_1,D_2)=\frac{\sum_{\langle d_1,d_2\rangle\in M_d} sim_{date}(d_1,d_2)}{\max\{|D_1|,|D_2|\}} \quad (6)$$

6.2.3. Wiki links and external links for similarity

In many cases an attribute value contains a link to another Wikipedia page: Barack Obama was born in `[[Honolulu]]`, which denotes a link to http://en.wikipedia.org/wiki/Honolulu. The basic idea is to consider the set of articles targeted by a wiki link in both attribute instances. With the help of the entity mapping we can easily determine which of these articles describe the same entity, and thus, how much overlap there is between the two sets of targeted articles. To relate the number of overlapping wiki links to the total number of wiki links contained in each of the two attribute instances, we use the Dice coefficient [25], which is in [0,1]:

$$sim_{wikilinks}(W_1,W_2)=\frac{2\cdot|W_1\cap W_2|}{|W_1|+|W_2|} \quad (7)$$

where $W_1 \cap W_2$ is the set of language-independent entities that are targeted by wiki links in both attribute instances.

The same is done for external links, although for them no entity matching is necessary. We simply relate the number of overlapping (i.e., same) external links with the number of external links in both attribute instances, again, using the Dice coefficient:

$$sim_{externallinks}(E_1,E_2)=\frac{2\cdot|E_1\cap E_2|}{|E_1|+|E_2|} \quad (8)$$

6.2.4. Combining the individual similarities

The individual similarities need to be combined into an overall similarity score between a pair of attribute instances. As described before, we have not only separated the different data types of the attribute values, but have also determined the shares of the respective data types in the original attribute value string. We use these fractions to weight the individual data type specific similarities. Let f_{s_1} and f_{s_2} be the fractions of characters assigned to the string portions of the values of two attribute instances, and len_1, len_2 the string lengths of the original attribute values. Then, the average string fraction f_s, weighted with respect to the length of the attribute values, is

$$f_s=\frac{len_1\cdot f_{s_1}+len_2\cdot f_{s_2}}{len_1+len_2}$$

The weighted average number and date fractions f_n and f_d are calculated analogously:

$$f_n=\frac{len_1\cdot f_{n_1}+len_2\cdot f_{n_2}}{len_1+len_2}, \quad f_d=\frac{len_1\cdot f_{d_1}+len_2\cdot f_{d_2}}{len_1+len_2}$$

Given the average fractions of each data type, we can now compute the similarity sim_{val} of the values of two attribute instances a_1 and a_2 by combining the similarities of the string, number, and date portions, and weighting them by their fractions as well as by additional weights w_s, w_n, and w_d. For the sake of clarity, let sim_s be the similarity sim_{string} of the string portions s_1 and s_2 of the attributes, sim_n the similarity of their number portions, and sim_d the similarity of the date portions:

$$sim_{val}(a_1,a_2)=\frac{w_s\cdot f_s\cdot sim_s+w_n\cdot f_n\cdot sim_n+w_d\cdot f_d\cdot sim_d}{w_s\cdot f_s+w_n\cdot f_n+w_d\cdot f_d}$$

The concrete weights used in our experiments are $w_s=0.11$, $w_n=0.44$, and $w_d=0.44$. These weights have been largely determined empirically, but there is a good

reason for the low weight of the string portion: the *information density* is much higher for numbers and dates than for text: for the two strings "ca. 4" and "ca. 9" we calculate $f_s=0.75$ and $f_n=0.25$. Without further weights, the perfect similarity of the string portion (ca.) would outweigh the rather poor number similarity, leading to a quite high overall score of 0.86. Weighting the number and date portions higher than the string portion adjusts this imbalance.

Next, the two link overlap ratios are considered, leading to the final similarity score $sim_{instance}$ between two attribute instances a_1 and a_2. Again for clarity, let sim_w and sim_e be the similarities between the sets of wiki links and external links, respectively.

$$sim_{instance}(a_1,a_2)=\frac{w_v\cdot sim_{val}(a_1,a_2)+w_w\cdot sim_w+w_e\cdot sim_e}{w_v+w_w+w_e}$$

Here, the concrete weights in our experiments are $w_v=0.3$, $w_w=0.6$, and $w_e=0.1$. Wiki link overlap is weighted very high, because two attribute instances containing wiki links that target the same entity are a quite reliable indicator for the equivalence of these two attributes.

The wiki link overlap is only considered if both attribute instances contain wiki links, otherwise w_w is set to 0. The reason for requiring *both* attribute instances to contain the links is simple: while in one language edition, parts of the attribute value may be linked to the corresponding article, this may not be the case in other Wikipedia editions. The same applies to external links: if *both* attribute instances do not contain external links we set $w_e=0$.

6.3. Similarity of attribute pairs

Having designed a similarity measure for pairs of individual attribute instances, we now derive the similarity between pairs of attributes. As described in Section 6.1, all co-occurring attribute instances for each matched template pair are compared with each other. More precisely, the attribute instance similarity $sim_{instance}$ is calculated for all attribute combinations for each pair of articles that describe the same entity and use the respective pair of infobox templates. This way, we obtain a list of instance-level similarities for each pair of attributes. The total similarity between two attributes can now be defined as the arithmetic mean of all instance-level similarities:

$$sim_{attr}(a_1,a_2)=\frac{\sum_{\langle a_{i_1},a_{i_2}\rangle\in A_{a_1,a_2}} sim_{instance}(a_1,a_2)}{|A_{a_1,a_2}|} \quad (9)$$

where A_{a_1,a_2} is the set of co-occurring attribute instance pairs for the two attributes a_1 and a_2.

Due to alternative attribute names (or typos in attribute names), there may be multiple attributes in one infobox template that correspond to an attribute in the other infobox. For example, the template en: Infobox software uses programming language and programming_language interchangeably, which are both equally good match candidates for the German Programmiersprache attribute. Since for now we allow only one-to-one matches, we have to choose one of these attributes. In such a case, it is reasonable to pick the alternative that is used more frequently. However, the approach so far only considers the average similarity of attribute pairs, but disregards attribute frequencies. Therefore, we present two additional methods to incorporate attribute frequencies into the matching:

1. To filter out very infrequent attributes, a threshold for attribute frequencies is applied. Let $freq(a_n)$ be the number of occurrences of attribute a_n in the set of compared infobox instance pairs I. The threshold condition is then:

   $$\frac{freq(a_1)}{|I|}>\frac{1}{30}\wedge\frac{freq(a_2)}{|I|}>\frac{1}{30}$$

   The similarity score of attribute pairs that do not satisfy this condition is set to 0, hence excluding them from the further matching process. These attribute pairs often originate from typos in one of the attribute names, thus we eliminate some false positives.
2. We incorporate the number of co-occurrences $cofreq(a_1,a_2)$ of two attributes a_1 and a_2 into the attribute similarity score forming the *justed attribute similarity* $sim_{adj}(a_1,a_2)$:

   $$sim_{adj}(a_1,a_2)=sim_{attr}(a_1,a_2)\cdot log_{10}(cofreq(a_1,a_2))$$

   In this way, additionally to regarding the raw similarity score, we consider two attributes to be a better match if they co-occur more frequently. In the case of programming language versus programming_language, for example, the former is used much more frequently together with the German Programmiersprache attribute than the latter (139 versus 2 co-occurrences). Incorporating the number of co-occurrences into the score ensures that, despite the slightly better raw similarity score between programming_language and Programmiersprache, the other alternative is chosen. Scaling the co-occurrences logarithmically proved to be a good compromise, as it penalizes very low frequencies, but on the other hand does not value high frequencies exaggeratedly.

Due to the additional factor, the final similarity score sim_{adj} does not yield results in the range [0,1] anymore. This, however, is no problem, because the scores of all attribute pairs are scaled alike.

6.4. Global matching

Given the similarity scores for each pair of co-occurring attributes, we now have to derive an overall schema mapping. Here, we confine ourselves to finding one-to-one mappings between attributes, although this does not always precisely represent reality. Determining one-to-many or even *m:n* mappings together with appropriate concatenation operators is left for future work (see Section 9).

Though the degree of information overlap in corresponding infobox templates is mostly high, the mapping of attributes between two concrete infoboxes is usually not complete. This means that often many attributes in one infobox do not have a corresponding partner in the second infobox, and vice versa. It is thus imperative for the global matching algorithm not to enforce a complete mapping among attributes.

Table 9
Evaluation of the computed attribute mapping. $|R|$ is the number of relevant attribute pairs, $|M|$ the number of matched attribute pairs and thus $|R \cap M|$ is the number of true positives.

	en↔de	en↔fr	en↔nl	de↔fr	de↔nl	fr↔nl	Total
$\|R\|$	292	284	269	214	177	181	**1417**
$\|M\|$	299	298	268	217	180	179	**1441**
$\|R \cap M\|$	275	275	255	197	165	168	**1335**
Precision (%)	91.97	92.28	95.15	90.78	91.67	93.85	**92.64**
Recall (%)	94.17	96.83	94.80	92.06	93.22	92.82	**94.21**
F_1 **measure** (%)	93.06	94.50	94.97	91.42	92.44	93.33	**93.42**

Commonly used bi-partite matching algorithms like algorithms to achieve a *stable marriage* [26] or the Hungarian algorithm [27] always enforce a complete mapping. To avoid false positives we choose another straightforward and quite strict approach to global matching, requiring both attributes to *mutually prefer each other*: First, for each attribute in both infobox templates, the match with the best score is determined. Then, for each attribute a of the first infobox template, the attribute's best match b is adopted to the final mapping only if a is also b's best match.

Quantifying confidence: It is desirable for an automatic schema matching tool to express its confidence in a found match. In this way, the obtained mapping can be filtered further, with the aim to increase precision (usually at the expense of recall). We calculate two such measures for each reported match:

Absolute confidence is an absolute measure of confidence and is defined as the similarity score sim_{attr} of the attribute match:

$$c_{abs}(a,b) = sim_{attr}(a,b)$$

Relative confidence is the confidence that the match is better than other possible matches for the participating attributes. It is calculated as the proportion of the match's similarity score to the average score of both attributes' second best matches *sbm*:

$$c_{rel}(a,b) = 1 - \frac{sim_{attr}(a,sbm_a) + sim_{attr}(sbm_b,b)}{2 \cdot sim_{attr}(a,b)}$$

As we show in the evaluation, both confidence measures together are a quite good discriminator to filter out false positives and thus to increase precision.

7. Evaluation of attribute matching

We now proceed to evaluate the attribute matching method described above. Due to poor infobox naming conventions in the Italian Wikipedia (it is not possible to automatically distinguish infoboxes from any other template) and buggy XML dumps of the Spanish Wikipedia at the time of experimenting,[5] we restrict our experimental evaluation to the four Wikipedia language editions English, German, French, and Dutch. Before evaluating the most important aspect of our matching system, its effectiveness, we briefly present runtime results.

[5] The bug has been reported at https://bugzilla.wikimedia.org/show_bug.cgi?id=18694.

7.1. Efficiency

Table 10 shows the runtime of the matcher to find attribute correspondences between all infobox template pairs for the six language pairs. The times denote *total CPU time* on the test system powered by two quad-core Intel Xeon X5355 processors and 16 GB main memory. Due to the multi-threaded implementation, the actual runtime is much lower on a multi-core system. As demonstrated by the numbers, the runtime scales roughly linearly with the number of infobox instance pairs (cf. Table 7 in Section 5.2). In summary, the computational performance is sufficiently good and scales well enough to process larger amounts of infobox template or language pairs.

7.2. Evaluation of matching effectiveness

To evaluate the overall performance of our matching approach, we have manually (and carefully) created the *expected* attribute correspondences for several infobox template pairs across all six language pairs. In total, these hand-crafted evaluation mappings cover 96 infobox template pairs (this is 3.44% of all co-occurring infobox template pairs in the considered language pairs) with a total of 1417 expected attribute pairs. The infobox template pairs have been manually selected to cover a diverse range of templates (e.g., templates with few or many instances) and matching problems (e.g., templates where attributes in the different languages use different units).

Table 9 shows the evaluation results of our approach, broken down by language pair. The upper half of the table presents the absolute numbers of relevant/expected attribute pairs (as defined by the hand-crafted mappings), actually matched attribute pairs and true positives (correctly matched pairs). The lower half shows the three metrics commonly used to evaluate results of information retrieval and schema matching approaches:

$$Precision = \frac{|R \cap M|}{|M|}, \quad Recall = \frac{|R \cap M|}{|R|}$$

$$F_1 measure = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

Of the total of 1417 expected attribute pairs, we correctly identify 1335 pairs, resulting in 82 false negatives and a recall of 94.21%. The total number of matched attribute pairs is 1441, thus there are 106 false positives

Table 10
Runtime of the matcher in total CPU time.

en↔de	222 m 02 s
en↔fr	203 m 39 s
en↔nl	156 m 21 s
de↔fr	93 m 31 s
de↔nl	72 m 59 s
fr↔nl	116 m 16 s

leading to an overall precision of 92.64%. Combining these two metrics, we obtain a satisfyingly high F_1 measure of 93.42%. Breaking down these numbers to the different language pairs, we still see consistently high precision and recall. Thus, we conclude that the different combinations of natural languages do not have a high impact on the matcher's performance (though this probably is not the case when matching infoboxes originating from very dissimilar languages, especially if these languages do not share the same script).

7.3. Applying confidence thresholds

In Section 6.4 we introduced two types of confidence measures: absolute and relative confidence. Assuming that both of them correlate roughly with the quality of a match, it should be possible to control the precision/recall ratio by introducing a threshold on either of the confidence measures or a reasonable combination of both. To test this assumption, Fig. 14 shows the impact of different thresholds θ on

- absolute confidence: $c_{\text{abs}} \geq \theta$ (Fig. 14(a)),
- relative confidence: $c_{rel} \geq \theta$ (Fig. 14(b)),
- harmonic mean of absolute and relative confidence: $(2 \cdot c_{abs} \cdot c_{rel})/(c_{abs}+c_{abs}) \geq \theta$ (Fig. 14(c)).

The plots show that both confidence measures taken separately can be used to increase matching precision (at the expense of recall). However, combining both confidence measures by taking their harmonic mean is a much better filtering criterion than either of them alone: We can increase precision from originally 92.6% to 96.9% while retaining a recall of 74.4%. With either of the confidence measures alone, this level of precision is only reached for much lower values of recall: 61.8% (c_{abs}) and 52.5% (c_{rel}). This is actually not too surprising: Only few false positives have both a high absolute *and* a high relative confidence.

Since we are now able to influence precision and recall with the help of the confidence threshold, we can plot a standard precision–recall diagram by sampling precision and recall at different confidence thresholds (see Fig. 15). Note that by merely adjusting the threshold, we can only filter out false positives; recall is bound by the recall obtained without applying any threshold (94.21%).

8. Related work

Research related to this work stems from several areas. First, the properties of interlanguage links (ILs) have been analyzed from different points of view. Second, ILLs have been exploited for entity matching in several other works. Finally, the problem of Wikipedia infobox attribute matching has been explored in different ways by at least two independent research groups, thus highlighting the potential and interest in finding correspondences between infobox attributes.

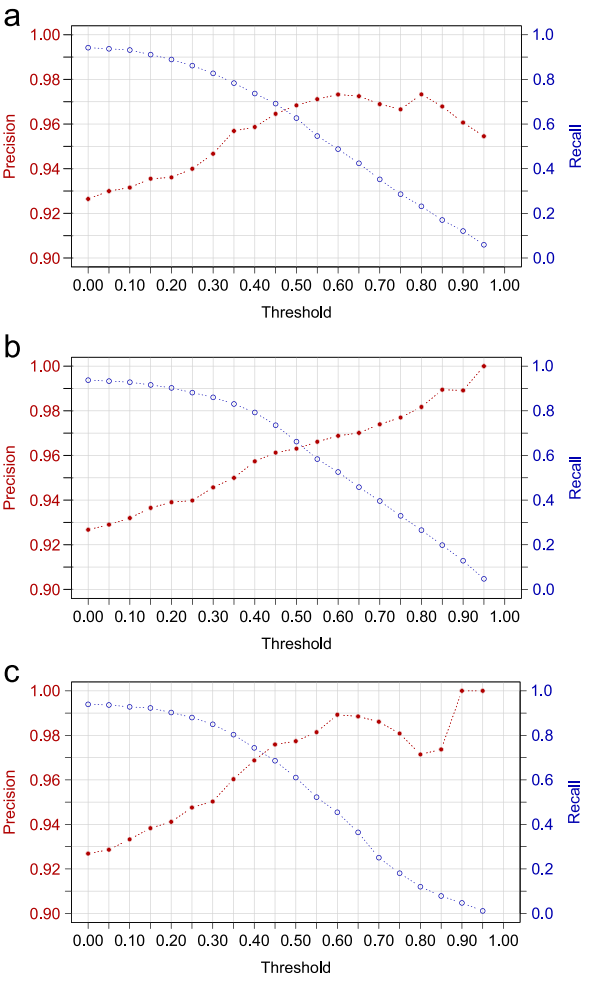

Fig. 14. Impact of different thresholds (sampled at 0.05 intervals) on precision and recall. Note the different scales for precision and recall. (a) Absolute confidence. (b) Relative confidence. (c) Harmonic mean of absolute and relative confidence.

8.1. Analysis of interlanguage links

Bolikowski provides a very detailed analysis of the topology of the complete Wikipedia ILL graph spanning all 250+ languages [14]. In this graph, vertices represent articles and directed edges represent the interlanguage links connecting them. With respect to (weakly) connected components in the graph, Bolikowski distinguishes *coherent* and *incoherent* components. A component is denoted as coherent, iff no two articles in this component belong to the same Wikipedia language edition. The main focus of the analysis is on the distributions of component sizes, node degrees, and clustering coefficients, each separately

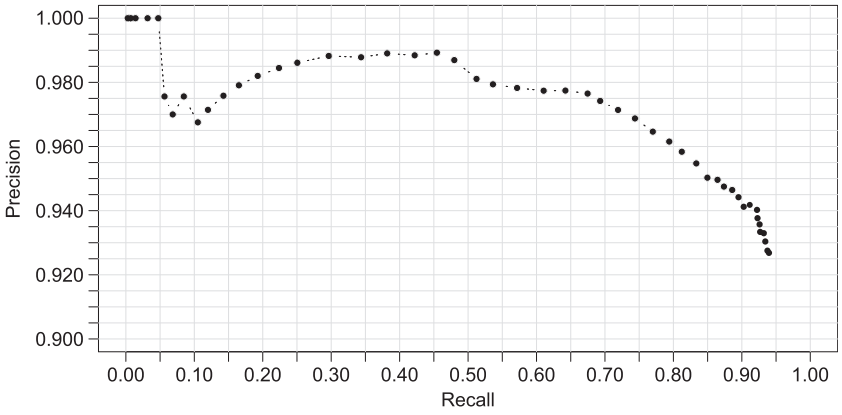

Fig. 15. Precision–recall diagram for the complete set of evaluation mappings (all language pairs) obtained by sampling the combined confidence threshold at 0.02 intervals. Note that the range of the precision axis starts at 0.9.

determined for coherent and incoherent components. The results of this analysis largely coincide with our observations, despite the lower number of languages considered. Bolikowski concludes that instead of consisting of only isolated cliques (which would indicate perfectly coherent and completely intra-linked components), the interlanguage link topology can be "informally described as a scale-free network of loosely connected near-cliques".

Further analysis of ILLs on the example of the Chinese, Japanese, Korean, and English Wikipedia editions has been conducted by Arai et al. [28]. Rather than examining the topology of the graph, the authors focus on specific link patterns, such as mutual, inconsistent, and unidirectional links, between pairs of languages. They discover that between 88.7% to 93.8% of all ILLs between the four considered languages are bidirectional.

Finally, Hammwöhner analyzes the quality of different aspects of Wikipedia, including that of ILLs [29]. Without concrete measurements the author lists several observed phenomena that may lead to inconsistencies among ILLs. These phenomena are mainly related to problems with disambiguation pages, links to wrong homonyms, and different granularities of articles in different languages.

8.2. Entity matching in Wikipedia

The task of entity matching in multi-lingual Wikipedia is a prerequisite for several research areas, especially for cross-lingual information retrieval and the building of multi-lingual lexical resources. Hence, this problem has been tackled (more or less profoundly) by different researchers.

Adar et al. [15] have approached the problem from a graph-theoretical point of view by identifying weakly connected components in the graph of ILLs between English, French, Spanish, and German Wikipedia editions. To eliminate conflicts, they simply discarded all components containing more than one article in any given language. A specific evaluation of their entity matching approach has not been conducted.

Hassan and Mihalcea [16] complete the ILLs between English, Spanish, Arabic, and Romanian Wikipedia editions by computing the transitive and symmetric[6] closures, which amounts to clustering weakly connected articles.

Hecht and Gergle [17] also propose a graph-based view on ILLs. To complete missing links between 25 language editions, they perform a breadth-first search on the graph, ignoring edge direction, and thus, again, find weakly connected components. While the algorithm's positive characteristic as "missing link finder" is highlighted, incoherency in the identified components and discovery of erroneous ILLs are not discussed. A rather narrow, and thus not very representative, human evaluation was performed: based on the analysis of 75 articles per language, the percentage of missing links varies between 2% and 8% (depending on the language pair), while the evaluation of merely 25 bilingual article pairs did not reveal any incorrect links. In another context the same authors have described a different two-pass approach, however only very briefly and without evaluation [30].

Bolikowski has not only analyzed ILLs from a graph-theoretical point of view, but also briefly proposes a method to approximately identify articles representing the same topic/entity [14]. The proposed approach, however, depends on a reference language edition. Nodes not connected to any node of the reference language are not considered by this approach. Thus, this method is not well suited for the task of entity matching across many languages.

A bottom-up approach to identify clusters in Wikipedia is presented by Kinzler [18]: Initially, for every single article in any language, a cluster is created. Afterwards, clusters are iteratively merged until no more merges are possible. The key is the merge condition: two clusters are merged, if (1) there is an ILL from any article in the first cluster to any article in the second cluster, and vice versa; and (2) the language sets of articles in the clusters are disjoint (to avoid the formation of incoherent components). A manual qualitative evaluation of 250 (out of 2.8 million) clusters spanning five languages (English,

[6] In their work symmetry is incorrectly referred to as "reflectivity".

German, French, Dutch, and Norwegian) resulted in no erroneously merged articles and six missing articles. Further details on this algorithm and a comprehensive comparison to our approach is given in Section 4.

The MENTA (Multilingual Entity Taxonomy) project integrates entities from different language editions of Wikipedia and WordNet [31]. Similar to our approach, the authors define a graph on the cross-lingual links and then remove conflicting links. In their work, they define the optimization problem of removing as few links as possible to achieve a coherent graph. They apply an algorithm based on a linear program that runs with a logarithmic approximation guarantee [32].

8.3. Infobox attribute matching

The task of infobox attribute matching falls into the broader area of schema matching, and there is a multitude of further relevant work. A general survey of common schema matching approaches is presented by Rahm and Bernstein [2]. They describe and compare different methods, and establish a taxonomy that is often used to classify schema matching approaches. However, many of the traditionally used techniques leverage the information of (mostly relational) schema definitions, such as column data types and constraints, and are thus not applicable for the task at hand. Thus, our discussion of related work focusses on approaches for infobox attribute matching. Yet, our approach is inspired by a (traditional) instance-based schema matching solution, namely duplicate-based schema matching [5]. While other instance-based matching approaches usually analyze higher-level characteristics of attribute instances within individual schemas, duplicate-based schema matching relies on a priori identified instances representing same real-world objects. To accommodate the multilingual nature of the infobox alignment task, as well as motivated by the unique characteristics of infobox attributes, that matching process has been largely adapted in several important aspects.

To predict the matching probability of pairs of infobox attribute instances across different language versions, Adar et al. employ self-supervised machine learning with a logistic regression classifier using a broad range of features, such as equality and n-gram/word overlap of attribute keys and values, wiki link overlap, correlation of numerical attributes, and translation-based features [15]. To generate a labeled training set, they first find attribute pairs that have exactly equal values for many article pairs. Then they adopt *all* attribute instance pairs for these attributes (not only the equal ones) as positive training data. Negative training data is derived from the positive matches, assuming that each attribute in one infobox template only corresponds to one attribute in the other infobox template. After training, the classifier is then used to predict if two attribute instances match (which is modeled as a binary decision). The likelihood that two attributes are a match is then simply defined as the ratio between the number of matched and total instance pairs. Their approach was evaluated on different levels: the *classifier* reaches a precision of 90.7% (tested using 10-fold cross-validation), which is not to be confused with the precision of the final attribute mapping. In order to quantify the precision of the attribute mapping, they manually verified 200 attribute pairs, of which 86% were classified as correct matches. However, since they only verified *found* matches, the recall of the approach remains unknown.

Our approach shares some ideas with the method presented by Adar et al. However, our approach puts a much stronger emphasis on the similarity measure for attribute instances, especially with respect to the handling of different data types. We do not use machine learning techniques, but rather rely on static weighting of different similarity features. Altogether, these decisions seem to pay off, as demonstrated by the high precision of 92.6% of our approach, at an even higher recall of 94.2% (see Section 7).

Also, Bouma et al. perform a matching of infobox attributes based on instance-data [33]. They first normalize all infobox attribute values, especially with respect to number, date formats, and some units. Then they tested these normalized attribute values for exact equality across different Wikipedia language versions. The authors themselves state that their results are not directly comparable to those in [15] but concede that their recall and precision is lower (and thus also lower than our results). In particular, their limited set of normalizations and reliance on strict value equality have a negative impact on recall. Further, formatting irregularities and other effects negatively affect precision.

9. Conclusions

To find an effective approach for the identification of groups of Wikipedia articles describing the same real-world entities, we have analyzed the interlanguage linkage situation in Wikipedia. While we found that the majority of the interlanguage links (ILLs) are of good quality, the analysis revealed that there are many conflicting links. In extreme cases, the accumulation of such erroneous or imprecise links result in large groups of articles that are connected by ILLs, but describe entirely unrelated topics. We have shown how the ILLs in Wikipedia can be leveraged to identify disjoint sets of articles representing the same entity. The presented top-down approach operates on the graph of ILLs and decomposes this graph into coherent components by successively applying stricter connectivity measures.

Using this entity mapping, we have demonstrated that an automatic instance-based matching of Wikipedia infobox attributes is feasible and quite successful, despite the challenging characteristics of infobox data. Our fundamental assumption on which this approach is based on, held true: attribute pairs that often contain highly similar values in different language editions are likely to correspond. The good evaluation results demonstrate that putting a strong focus on a robust similarity measure that is capable of quantifying the similarity of strings with different types of data, such as numbers and dates, is a successful strategy.

Future work: There still remains, however, room for improvement. Currently, our approach is limited to one-to-one mappings between attributes. During the work with the infobox data, we have found at least two scenarios that cannot be represented with a one-to-one mapping: First, there exist alternative names for attributes that are used interchangeably (see Section 6.3). With the current

implementation, we have to choose between such alternative attributes. It would, however, be preferable to find and report these alternatives. The second scenario are one-to-many matches, resulting from different attribute specificity. Further, depending on the further application of the attribute mapping, it is not always sufficient to merely relate a set of attributes to an attribute in the other template. We would rather like to determine the correct order and concatenation operators for the attributes on the "many" side, too [34]. Moreover, further processing and normalization of the data, such as detection and conversion of units, could yield better results. It would also be interesting to evaluate whether the incorporation of numeric correlation features (e.g., with the help of the Pearson correlation coefficient) can be used to reliably detect corresponding numeric values, even if they are represented in different units.

With respect to entity identification, one interesting approach could be to feed the found infobox attribute mapping back to the entity matching stage. It could then be used to resolve conflicts in components by assessing the similarity of infobox data between multiple competing articles of one language version and the other articles in the component (of course, only if all involved articles contain infoboxes).

Applications: Several possible applications that can take advantage of aligned infobox schemas have already been suggested in Section 1. One very interesting such application is that of *conflict detection*. Given an attribute mapping between a pair of infobox templates, one can analyze all corresponding attribute instance pairs and try to detect inconsistencies between the attribute values. While it is trivial to detect whether attribute values differ from each other (based on strict string equality), the main challenge here is to decide whether differing values really constitute a semantic conflict (as opposed to formatting, unit, and language differences). Having *detected* potential conflicts, these inconsistencies can either be corrected automatically, or tools can be designed to support human Wikipedia authors with the correction of the data. Automatic conflict resolution is most probably only feasible for a limited set of obvious inconsistencies, because it involves not only deciding which of the conflicting values is correct, but also needs to transform the value to the expected format in the other language editions (which is not clearly defined due to the loose schema). This transformation includes adaption of number and date formats, but also translation of textual elements in the attribute values. Providing tool-support for authors, in contrast, is a much more straightforward approach. Such a tool could present potential inconsistencies to Wikipedia authors, along with proposed resolutions, but leave the final decision and execution to a human author. Though not fully automatic, the benefit of such a tool would still be high, because without tool support inconsistencies in attribute values are only rarely recognized.

References

[1] E. Tacchini, A. Schultz, C. Bizer, Experiments with Wikipedia cross-language data fusion, in: Proceedings of the Workshop on Scripting and Development for the Semantic Web (SFSW), Crete, Greece, 2009.

[2] E. Rahm, P.A. Bernstein, A survey of approaches to automatic schema matching, VLDB Journal 10 (4) (2001) 334–350.

[3] J. Euzenat, P. Shvaiko, Ontology Matching, Springer Verlag, Berlin, Heidelberg, New York, 2007.

[4] H.H. Do, E. Rahm, COMA—a system for flexible combination of schema matching approaches, in: Proceedings of the International Conference on Very Large Databases (VLDB), 2002, pp. 610–621.

[5] A. Bilke, F. Naumann, Schema matching using duplicates, in: Proceedings of the International Conference on Data Engineering (ICDE), Washington, DC, 2005, pp. 69–80.

[6] Wikipedia, Help: Interlanguage links, 2010. URL 〈http://en.wikipedia.org/wiki/Help:Interlanguage_links〉.

[7] Serious problems with interlanguage links—`wikien-l` mailing list, 2008. URL 〈http://www.mail-archive.com/wikien-l@lists.wikimedia.org/msg00371.html〉.

[8] D. Kinzler, Ideas for a smarter inter-language link system, 2008. URL 〈http://brightbyte.de/page/Ideas_for_a_smarter_inter-language_link_system〉.

[9] Wikimedia Meta wiki, Fine interwiki, 2010. URL 〈http://meta.wikimedia.org/wiki/Fine_interwiki〉.

[10] Wikimedia Meta wiki, Interlanguage use case, 2010. URL 〈http://meta.wikimedia.org/wiki/Interlanguage_use_case〉.

[11] Wikimedia Meta wiki, A newer look at the interlanguage link, 2010. URL 〈http://meta.wikimedia.org/wiki/A_newer_look_at_the_interlanguage_link〉.

[12] Wikimedia, Database backup dumps, 2010. URL 〈http://dumps.wikimedia.org/backup-index.html〉.

[13] Wikipedia, Namespace, 2010. URL 〈http://en.wikipedia.org/wiki/Wikipedia:Namespace〉.

[14] Ł. Bolikowski, Scale-free topology of the interlanguage links in Wikipedia, arXiv preprint. URL 〈http://arxiv.org/abs/0904.0564v2〉.

[15] E. Adar, M. Skinner, D.S. Weld, Information arbitrage across multilingual Wikipedia, in: Proceedings of the International Conference on Web Search and Data Mining (WSDM), 2009, pp. 94–103.

[16] S. Hassan, R. Mihalcea, Cross-lingual semantic relatedness using encyclopedic knowledge, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Morristown, NJ, USA, 2009, pp. 1192–1201.

[17] B. Hecht, D. Gergle, The Tower of Babel meets Web 2.0: User-generated content and its applications in a multilingual context, in: Proceedings of the International Conference on Human Factors in Computing Systems (CHI), New York, NY, USA, 2010, pp. 291–300.

[18] D. Kinzler, Building language-independent concepts from Wikipedia, Presented at WikiSym 2008—the International Symposium on Wikis, 2008. URL 〈http://brightbyte.de/repos/papers/2008/LangLinks-paper.pdf〉.

[19] P. Sorg, P. Cimiano, Enriching the crosslingual link structure of Wikipedia—a classification-based approach, in: Proceedings of the AAAI Workshop on Wikipedia and Artificial Intelligence, 2008.

[20] V.I. Levenshtein, Binary codes capable of correcting deletions, insertions, and reversals, Soviet Physics Doklady 10 (8) (1966) 707–710.

[21] M.A. Jaro, Advances in record linking methodology as applied to the 1985 census of Tampa Florida, Journal of the American Statistical Society 64 (1989) 1183–1210.

[22] W.W. Cohen, P. Ravikumar, S.E. Fienberg, A comparison of string distance metrics for name-matching tasks, in: Proceedings of IJCAI Workshop on Information Integration, 2003, pp. 73–78.

[23] A. Salhi, H. Camacho, A string metric based on a one-to-one greedy matching algorithm, Research in Computing Science: Advances in Computer Science and Engineering 19 (2006) 171–182.

[24] D.E. Drake, S. Hougardy, A simple approximation algorithm for the weighted matching problem, Information Processing Letters 85 (2003) 211–213.

[25] L.R. Dice, Measures of the amount of ecologic association between species, Ecology 26 (3) (1945) 297–302.

[26] D. Gale, L.S. Shapley, College admissions and the stability of marriage, The American Mathematical Monthly 69 (1) (1962) 9–15.

[27] H.W. Kuhn, The Hungarian Method for the assignment problem, Naval Research Logistics Quarterly 2 (1–2) (1955) 83–97.

[28] Y. Arai, T. Fukuhara, H. Masuda, H. Nakagawa, Analyzing interlanguage links of Wikipedias, in: Proceedings of the Wikimania Conference, 2008.

[29] R. Hammwöhner, Interlingual aspects of Wikipedia's quality, in: Proceedings of the International Conference on Information Quality (IQ), Boston, MA, 2007, pp. 39–49.

[30] B. Hecht, D. Gergle, Measuring self-focus bias in community-maintained knowledge repositories, in: Proceedings of the International

Conference on Communities and Technologies (C&T), University Park, PA, USA, 2009, pp. 11–20.
[31] G. de Melo, G. Weikum, MENTA: inducing multilingual taxonomies from wikipedia, in: Proceedings of the International Conference on Information and Knowledge Management (CIKM), 2010, pp. 1099–1108.
[32] G. de Melo, G. Weikum, Untangling the cross-lingual link structure of Wikipedia, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), 2010, pp. 844–853.
[33] G. Bouma, S. Duarte, Z. Islam, Cross-lingual alignment and completion of Wikipedia templates, in: Proceedings of the International Workshop on Cross Lingual Information Access: Addressing the Information Need of Multilingual Societies, 2009, pp. 21–29.
[34] R. Dhamankar, Y. Lee, A. Doan, A. Halevy, P. Domingos, iMAP: Discovering complex semantic matches between database schemas, in: Proceedings of the ACM International Conference on Management of Data (SIGMOD), 2004, pp. 383–394.