Untangling the Cross-Lingual Link Structure of Wikipedia

Gerard de Melo
Max Planck Institute for Informatics
Saarbrücken, Germany
demelo@mpi-inf.mpg.de

Gerhard Weikum
Max Planck Institute for Informatics
Saarbrücken, Germany
weikum@mpi-inf.mpg.de

Abstract

Wikipedia articles in different languages are connected by interwiki links that are increasingly being recognized as a valuable source of cross-lingual information. Unfortunately, large numbers of links are imprecise or simply wrong. In this paper, techniques to detect such problems are identified. We formalize their removal as an optimization task based on graph repair operations. We then present an algorithm with provable properties that uses linear programming and a region growing technique to tackle this challenge. This allows us to transform Wikipedia into a much more consistent multilingual register of the world's entities and concepts.

1 Introduction

Motivation. The open community-maintained encyclopedia Wikipedia has not only turned the Internet into a more useful and linguistically diverse source of information, but is also increasingly being used in computational applications as a large-scale source of linguistic and encyclopedic knowledge. To allow cross-lingual navigation, Wikipedia offers cross-lingual *interwiki* links that for instance connect the Indonesian article about Albert Einstein to the corresponding articles in over 100 other languages. Such links are extraordinarily valuable for cross-lingual applications.

In the ideal case, a set of articles connected directly or indirectly via such links would all describe the same entity or concept. Due to conceptual drift, different granularities, as well as mistakes made by editors, we frequently find concepts as different as *economics* and *manager* in the same connected component. Filtering out inaccurate links enables us to exploit Wikipedia's multilinguality in a much safer manner and allows us to create a multilingual register of named entities.

Contribution. Our research contributions are: 1) We identify criteria to detect inaccurate connections in Wikipedia's cross-lingual link structure. 2) We formalize the task of removing such links as an optimization problem. 3) We introduce an algorithm that attempts to repair the cross-lingual graph in a minimally invasive way. This algorithm has an approximation guarantee with respect to optimal solutions. 4) We show how this algorithm can be used to combine all editions of Wikipedia into a single large-scale multilingual register of named entities and concepts.

2 Detecting Inaccurate Links

In this paper, we model the union of cross-lingual links provided by all editions of Wikipedia as an undirected graph $G = (V, E)$ with edge weights $w(e)$ for $e \in E$. In our experiments, we simply honour each individual link equally by defining $w(e) = 2$ if there are reciprocal links between the two pages, 1 if there is a single link, and 0 otherwise. However, our framework is flexible enough to deal with more advanced weighting schemes, e.g. one could easily plug in cross-lingual measures of semantic relatedness between article texts.

It turns out that an astonishing number of connected components in this graph harbour inaccurate links between articles. For instance, the Esperanto article '*Germana Imperiestro*' is about German emporers and another Esperanto article '*Germana Imperiestra Regno*' is about the German Empire, but, as of June 2010, both are linked to the English and German articles about the German Empire. Over time, some inaccurate links may be fixed, but in this and in large numbers of other cases, the imprecise connection has persisted for many years. In order to detect such cases, we need to have some way of specifying that two articles are likely to be distinct.

Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 844–853,
Uppsala, Sweden, 11-16 July 2010.

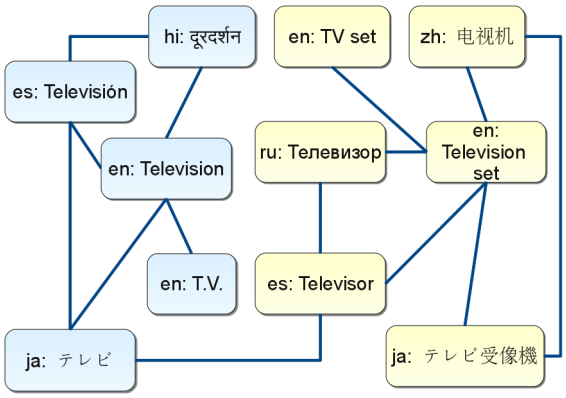

Figure 1: Connected component with inaccurate links (simplified)

2.1 Distinctness Assertions

Figure 1 shows a connected component that conflates the concept of television as a medium with the concept of TV sets as devices. Among other things, we would like to state that '*Television*' and '*T.V.*' are distinct from '*Television set*' and '*TV set*'. In general, we may have several sets of entities $D_{i,1}, \ldots, D_{i,l_i}$, for which we assume that any two entities u,v from different sets are pairwise distinct with some degree of confidence or weight. In our example, $D_{i,1} = \{$'*Television*','*T.V.*'$\}$ would be one set, and $D_{i,2} = \{$'*Television set*','*TV set*'$\}$ would be another set, which means that we are assuming '*Television*', for example, to be distinct from both '*Television set*' and '*TV set*'.

Definition 1. *(Distinctness Assertions) Given a set of nodes V, a* distinctness assertion *is a collection $D_i = (D_{i,1}, \ldots, D_{i,l_i})$ of pairwise disjoint (i.e. $D_{i,j} \cap D_{i,k} = \emptyset$ for $j \neq k$) subsets $D_{i,j} \subset V$ that expresses that any two nodes $u \in D_{i,j}$, $v \in D_{i,k}$ from different subsets ($j \neq k$) are asserted to be distinct from each other with some weight $w(D_i) \in \mathbb{R}$.*

We found that many components with inaccurate links can be identified automatically with the following distinctness assertions.

Criterion 1. *(Distinctness between articles from the same Wikipedia edition)* For each language-specific edition of Wikipedia, a separate assertion $(D_{i,1}, D_{i,2}, \ldots)$ can be made, where each $D_{i,j}$ contains an individual *article* together with its respective redirection pages. Two articles from the same Wikipedia very likely describe distinct concepts unless they are redirects of each other. For example, '*Georgia (country)*' is distinct from '*Georgia (U.S. State)*'. Additionally, there are also redirects that are clearly marked by a category or template as involving topic drift, e.g. redirects from songs to albums or artists, from products to companies, etc. We keep such redirects in a $D_{i,j}$ distinct from the one of their redirect targets.

Criterion 2. *(Distinctness between categories from the same Wikipedia edition)* For each language-specific edition of Wikipedia, a separate assertion $(D_{i,1}, D_{i,2}, \ldots)$ is made, where each $D_{i,j}$ contains a *category* page together with any redirects. For instance, '*Category:Writers*' is distinct from '*Category:Writing*'.

Criterion 3. *(Distinctness for links with anchor identifiers)* The English '*Division by zero*', for instance, links to the German '*Null#Division*'. The latter is only a part of a larger article about the number zero in general, so we can make a distinctness assertion to separate '*Division by zero*' from '*Null*'. In general, for each interwiki link or redirection with an anchor identifier, we add an assertion $(D_{i,1}, D_{i,2})$ where $D_{i,1}$,$D_{i,2}$ represent the respective articles without anchor identifiers.

These three types of distinctness assertions are instantiated for all articles and categories of all Wikipedia editions. The assertion weights are tunable; the simplest choice is using a uniform weight for all assertions (note that these weights are different from the edge weights in the graph). We will revisit this issue in our experiments.

2.2 Enforcing Consistency

Given a graph G representing cross-lingual links between Wikipedia pages, as well as distinctness assertions $D_1, \ldots, D_n$ with weights $w(D_i)$, we may find that nodes that are asserted to be distinct are in the same connected component. We can then try to apply repair operations to reconcile the graph's link structure with the distinctness asssertions and obtain global consistency. There are two ways to modify the input, and for each we can also consider the corresponding weights as a sort of *cost* that quantifies how much we are changing the original input:

a) **Edge cutting:** We may remove an edge $e \in E$ from the graph, paying cost $w(e)$.
b) **Distinctness assertion relaxation:** We may remove a node $v \in V$ from a distinctness assertion D_i, paying cost $w(D_i)$.

Removing edges allows us to split connected components into multiple smaller components, thereby ensuring that two nodes asserted to be distinct are no longer connected directly or indirectly. In Figure 1, for instance, we could delete the edge from the Spanish '*TV set*' article to the Japanese '*television*' article. In constrast, removing nodes from distinctness assertions means that we decide to give up our claim of them being distinct, instead allowing them to share a connected component.

Our reliance on costs is based on the assumption that the link structure or topology of the graph provides the best indication of which cross-lingual links to remove. In Figure 1, we have distinctness assertions between nodes in two densely connected clusters that are tied together only by a single spurious link. In such cases, edge removals can easily yield separate connected components. When, however, the two nodes are strongly connected via many different paths with high weights, we may instead opt for removing one of the two nodes from the distinctness assertion.

The aim will be to balance the costs for removing edges from the graph with the costs for removing nodes from distinctness assertions to produce a consistent solution with a minimal total repair cost. We accommodate our knowledge about distinctness while staying as close as possible to what Wikipedia provides as input.

This can be formalized as the **Weighted Distinctness-Based Graph Separation (WDGS)** problem. Let G be an undirected graph with a set of vertices V and a set of edges E weighted by $w : E \to \mathbb{R}$. If we use a set $C \subseteq V$ to specify which edges we want to cut from the original graph, and sets U_i to specify which nodes we want to remove from distinctness assertions, we can begin by defining WDGS solutions as follows.

Definition 2. *(**WDGS** Solution). Given a graph $G = (V, E)$ and n distinctness assertions $D_1, \ldots, D_n$, a tuple $(C, U_1, \ldots, U_n)$ is a valid* WDGS *solution if and only if $\forall i, j, k \neq j, u \in D_{i,j} \setminus U_i$, $v \in D_{i,k} \setminus U_i$: $\mathrm{P}(u, v, E \setminus C) = \emptyset$, i.e. the set of paths from u to v in the graph $(V, E \setminus C)$ is empty.*

Definition 3. *(**WDGS** Cost). Let $w : E \to \mathbb{R}$ be a weight function for edges $e \in E$, and $w(D_i)$ $(i = 1 \ldots n)$ be weights for the distinctness assertions. The (total) cost of a* WDGS *solution $S = (C, U_1, \ldots, U_n)$ is then defined as*

$$c(S) = c(C, U_1, \ldots, U_n) = \left[\sum_{e \in C} w(e)\right] + \left[\sum_{i=1}^{n} |U_i|\, w(D_i)\right]$$

Definition 4. *(**WDGS**). A* WDGS *problem instance P consists of a graph $G = (V, E)$ with edge weights $w(e)$ and n distinctness assertions D_1, $\ldots$, D_n with weights $w(D_i)$. The objective consists in finding a solution $(C, U_1, \ldots, U_n)$ with minimal cost $c(C, U_1, \ldots, U_n)$.*

It turns out that finding optimal solutions efficiently is a hard problem (proofs in Appendix A).

Theorem 1. WDGS *is NP-hard and APX-hard. If the Unique Games Conjecture (Khot, 2002) holds, then it is NP-hard to approximate* WDGS *within any constant factor $\alpha > 0$.*

3 Approximation Algorithm

Due to the hardness of WDGS, we devise a polynomial-time approximation algorithm with an approximation factor of $4 \ln(nq + 1)$ where n is the number of distinctness assertions and $q = \max_{i,j} |D_{i,j}|$. This means that for all problem instances P, we can guarantee

$$\frac{c(S(P))}{c(S^*(P))} \leq 4 \ln(nq + 1),$$

where $S(P)$ is the solution determined by our algorithm, and $S^*(P)$ is an optimal solution. Note that this approximation guarantee is independent of how long each D_i is, and that it merely represents an upper bound on the worst case scenario. In practice, the results tend to be much closer to the optimum, as will be shown in Section 4.

Our algorithm first solves a linear program (LP) relaxation of the original problem, which gives us hints as to which edges should most likely be cut and which nodes should most likely be removed from distinctness assertions. Note that this is a continuous LP, not an integer linear program (ILP); the latter would not be tractable due to the large number of variables and constraints of the problem. After solving the linear program, a new – extended – graph is constructed and the optimal LP solution is used to define a distance metric on it. The final solution is obtained by smartly selecting regions in this extended graph as the individual output components, employing a region

growing technique in the spirit of the seminal work by Leighton and Rao (1999). Edges that cross the boundaries of these regions are cut.

Definition 5. *Given a* WDGS *instance, we define a linear program of the following form:*

minimize

$$\sum_{e \in E} d_e w(e) + \sum_{i=1}^{n} \sum_{j=1}^{l_i} \sum_{v \in D_{i,j}} u_{i,v} w(D_i)$$

subject to

$$p_{i,j,v} = u_{i,v} \qquad \forall i,\, j{<}l_i,\, v \in D_{i,j} \qquad (1)$$

$$p_{i,j,v} + u_{i,v} \geq 1 \qquad \forall i,\, j{<}l_i,\, v \in \bigcup_{k>j} D_{i,k} \qquad (2)$$

$$p_{i,j,v} \leq p_{i,j,u} + d_e \qquad \forall i,\, j{<}l_i,\, e{=}(u,v) \in E \qquad (3)$$

$$d_e \geq 0 \qquad \forall e \in E \qquad (4)$$

$$u_{i,v} \geq 0 \qquad \forall i,\, v \in \bigcup_{j=1}^{l_i} D_{i,j} \qquad (5)$$

$$p_{i,j,v} \geq 0 \qquad \forall i,\, j{<}l_i,\, v{\in}V \qquad (6)$$

The LP uses decision variables d_e and $u_{i,v}$, and auxiliary variables $p_{i,j,v}$ that we refer to as *potential variables*. The d_e variables indicate whether (in the continuous LP: to what degree) an edge e should be deleted, and the $u_{i,v}$ variables indicate whether (to what degree) v should be removed from a distinctness assertion D_i. The LP objective function corresponds to Definition 3, aiming to minimize the total costs. A potential variable $p_{i,j,v}$ reflects a sort of potential difference between an assertion $D_{i,j}$ and a node v. If $p_{i,j,v} = 0$, then v is still connected to nodes in $D_{i,j}$. Constraints (1) and (2) enforce potential differences between $D_{i,j}$ and all nodes in $D_{i,k}$ with $k > j$. For instance, for distinctness between '*New York City*' and '*New York*' (the state), they might require '*New York*' to have a potential of 1, while '*New York City*' has a potential of 0. The potential variables are tied to the deletion variables d_e for edges in Constraint (3) as well as to the $u_{i,v}$ in Constraints (1) and (2). This means that the potential difference $p_{i,j,v} + u_{i,v} \geq 1$ can only be obtained if edges are deleted on every path between '*New York City*' and '*New York*', or if at least one of these two nodes is removed from the distinctness assertion (by setting the corresponding $u_{i,v}$ to non-zero values). Constraints (4), (5), (6) ensure non-negativity.

Having solved the linear program, the next major step is to convert the optimal LP solution into the final – discrete – solution. We cannot rely on standard rounding methods to turn the optimal fractional values of the d_e and $u_{i,v}$ variables into a valid solution. Often, all solution variables have small values and rounding will merely produce an empty $(C, U_1, \ldots, U_n) = (\emptyset, \emptyset, \ldots, \emptyset)$. Instead, a more sophisticated technique is necessary. The optimal solution of the LP can be used to define an extended graph G' with a distance metric d between nodes. The algorithm then operates on this graph, in each iteration selecting regions that become output components and removing them from the graph. A simple example is shown in Figure 2. The extended graph contains additional nodes and edges representing distinctness assertions. Cutting one of these additional edges corresponds to removing a node from a distinctness assertion.

Definition 6. *Given $G = (V, E)$ and distinctness assertions $D_1, \ldots, D_n$ with weights $w(D_i)$, we define an undirected graph $G' = (V', E')$ where $V' = V \cup \{\mathrm{v}_{i,v} \mid i = 1 \ldots n, w(D_i) > 0, v \in \bigcup_j D_{i,j}\}$, $E' = \{e \in E \mid w(e) > 0\} \cup \{(v, \mathrm{v}_{i,v}) \mid v \in D_{i,j}, w(D_i) > 0\}$. We accordingly extend the definition of $w(e)$ to additionally cover the new edges by defining $w(e) = w(D_i)$ for $e = (v, \mathrm{v}_{i,v})$. We also extend it for sets S of edges by defining $w(S) = \sum_{e \in S} w(e)$. Finally, we define a node distance metric*

$$d(u,v) = \begin{cases} 0 & u = v \\ d_e & (u,v) \in E \\ u_{i,v} & u = \mathrm{v}_{i,v} \\ u_{i,u} & v = \mathrm{v}_{i,u} \\ \min\limits_{p \in \mathrm{P}(u,v,E')} \sum\limits_{(u',v') \in p} d(u',v') & \textit{otherwise,} \end{cases}$$

where $\mathrm{P}(u, v, E')$ denotes the set of acyclic paths between two nodes in E'. We further fix

$$\hat{c}_\mathrm{f} = \sum_{(u,v) \in E'} d(u,v)\, w(e)$$

as the weight of the fractional solution of the LP ($\hat{c}_\mathrm{f}$ is a constant based on the original E', irrespective of later modifications to the graph).

Definition 7. *Around a given node v in G', we consider regions $R(v, r) \subseteq V$ with radius r. The cut $C(v, r)$ of a given region is defined as the set of edges in G' with one endpoint within the region and one outside the region:*

$$R(v,r) = \{v' \in V' \mid d(v,v') \leq r\}$$

$$C(v,r) = \{e \in E' \mid |e \cap R(v,r)| = 1\}$$

For sets of nodes $S \subseteq V$, we define $R(S, r) = \bigcup_{v \in S} R(v, r)$ and $C(S, r) = \bigcup_{v \in S} C(v, r)$.

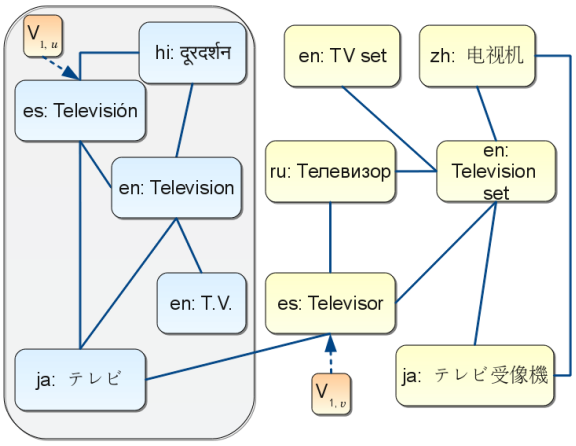

Figure 2: Extended graph with two added nodes $v_{1,u}$, $v_{1,v}$ representing distinctness between '*Televisión*' and '*Televisor*', and a region around $v_{1,u}$ that would cut the link from the Japanese '*Television*' to '*Televisor*'

Definition 8. *Given* $q = \max_{i,j} |D_{i,j}|$*, we approximate the optimal cost of regions as:*

$$\hat{c}(v,r) = \sum_{\substack{e=(u,u')\in E':\\ e\subseteq R(v,r)}} d(u,u')\, w(e) \quad (1)$$
$$+ \sum_{\substack{e\in C(v,r)\\ v'\in e\cap R(v,r)}} (r - d(v,v'))\, w(e)$$

$$\hat{c}(S,r) = \frac{1}{nq}\hat{c}_{\mathrm{f}} + \sum_{v\in S} \hat{c}(v,r) \quad (2)$$

The first summand accounts for the edges entirely within the region, and the second one accounts for the edges in $C(v,r)$ to the extent that they are within the radius. The definition of $\hat{c}(S,r)$ contains an additional slack component that is required for the approximation guarantee proof.

Based on these definitions, Algorithm 3.1 uses the LP solution to construct the extended graph. It then repeatedly, as long as there is an unsatisfied assertion D_i, chooses a set S of nodes containing one node from each relevant $D_{i,j}$. Around the nodes in S it simultaneously grows $|S|$ regions with the same radius, a technique previously suggested by Avidor and Langberg (2007). These regions are essentially output components that determine the solution. Repeatedly choosing the radius that minimizes $\frac{w(C(S,r))}{\hat{c}(S,r)}$ allows us to obtain the approximation guarantee, because the distances in this extended graph are based on the solution of the LP. The properties of this algorithm are given by the following two theorems (proofs in Appendix A).

Theorem 2. *The algorithm yields a* valid WDGS *solution* $(C, U_1, \ldots, U_n)$.

Theorem 3. *The algorithm yields a solution* $(C, U_1, \ldots, U_n)$ *with an approximation factor of* $4\ln(nq+1)$ *with respect to the cost of the optimal* WDGS *solution* $(C^*, U_1^*, \ldots, U_n^*)$*, where* n *is the number of distinctness assertions and* $q = \max_{i,j} |D_{i,j}|$*. This solution can be obtained in polynomial time.*

4 Results

4.1 Wikipedia

We downloaded February 2010 XML dumps of all available editions of Wikipedia, in total 272 editions that amount to 86.5 GB uncompressed. From these dumps we produced two datasets. Dataset A captures cross-lingual interwiki links between pages, in total 77.07 million undirected edges (146.76 million original links). Dataset B additionally includes 2.2 million redirect-based edges. Wikipedia deals with interwiki links to redirects transparently, however there are many redirects with titles that do not co-refer, e.g. redirects from members of a band to the band, or from aspects of a topic to the topic in general. We only included redirects in the following cases:

- the titles of redirect and redirect target match after Unicode NFKD normalization, diacritics removal, case conversion, and removal of punctuation characters
- the redirect uses certain templates or categories that indicate co-reference with the target (alternative names, abbreviations, etc.)

We treated them like reciprocal interwiki links by assigning them a weight of 2.

4.2 Application of Algorithm

The choice of distinctness assertion weights depends on how lenient we wish to be towards conceptual drift, allowing us to opt for more fine- or more coarse-grained distinctions. In our experiments, we decided to prefer fine-grained conceptual distinctions, and settled on a weight of 100.

We analysed over 20 million connected components in each dataset, checking for distinctness assertions. For the roughly 110,000 connected components with relevant distinctness assertions,

Algorithm 3.1 WDGS Approximation Algorithm

procedure SELECT($V, E, V', E', w, D_1, \ldots, D_n, l_1, \ldots, l_n$)
  solve linear program given by Definition 5 ▷ determine optimal fractional solution
  construct $G' = (V', E')$ ▷ extended graph (Definition 6)
  $C \leftarrow \{e \in E \mid w(e) = 0\}$ ▷ cut zero-weighted edges
  $U_i \leftarrow \bigcup_{j=1}^{l_i - 1} D_{i,j} \quad \forall i : w(D_i) = 0$ ▷ remove zero-weighted D_i
  while $\exists i, j, k > j, u \in D_{i,j}, v \in D_{i,k} : \mathrm{P}(\mathrm{v}_{i,u}, \mathrm{v}_{i,v}, E') \neq \emptyset$ **do** ▷ find unsatisfied assertion
    $S \leftarrow \emptyset$ ▷ set of nodes around which regions will be grown
    for all j in $1 \ldots l_i - 1$ **do** ▷ arbitrarily choose node from each $D_{i,j}$
      if $\exists v \in D_{i,j} : \mathrm{v}_{i,v} \in V'$ **then** $S \leftarrow S \cup \mathrm{v}_{i,v}$
    $D \leftarrow \{d(u,v) \leq \frac{1}{2} \mid u \in S, v \in V'\} \cup \{\frac{1}{2}\}$ ▷ set of distances
    choose ϵ such that $\forall d, d' \in D : 0 < \epsilon \ll |d - d'|$ ▷ infinitesimally small
    $r \leftarrow \operatorname{argmin}_{r = d - \epsilon:\ d \in D \setminus \{0\}} \frac{w(C(S,r))}{\hat{c}(S,r)}$ ▷ choose optimal radius (ties broken arbitrarily)
    $V' \leftarrow V' \setminus R(S,r)$ ▷ remove regions from G'
    $E' \leftarrow \{e \in E' \mid e \subseteq V'\}$
    $C \leftarrow C \cup (C(S,r) \cap E)$ ▷ update global solution
    for all i' in $1 \ldots n$ **do**
      $U_{i'} \leftarrow U_{i'} \cup \{v \mid (\mathrm{v}_{i',v}, v) \in C(S,r)\}$
      for all j in $1 \ldots l_{i'}$ **do** $D_{i',j} \leftarrow D_{i',j} \cap V'$ ▷ prune distinctness assertions
  return $(C, U_1, \ldots, U_n)$

we applied our algorithm, relying on the commercial CPLEX tool to solve the linear programs. In most cases, the LP solving took less than a second, however the LP sizes grow exponentially with the number of nodes and hence the time complexity increases similarly. In about 300 cases per dataset, CPLEX took too long and was automatically killed or the linear program was a priori deemed too large to complete in a short amount of time. For these cases, we adopted an alternative strategy described later on.

Table 1 provides the experimental results for the two datasets. Dataset B is more connected and thus has fewer connected components with more pairs of nodes asserted to be distinct by distinctness assertions. The LP given by Definition 5 provides fractional solutions that constitute lower bounds on the optimal solution (cf. also Lemma 5 in Appendix A), so the optimal solution cannot have a cost lower than the fractional LP solution. Table 1 shows that in practice, our algorithm achieves near-optimal results.

Table 1: Algorithm Results

	Dataset A	Dataset B
Connected components	23,356,027	21,161,631
– with distinctness assertions	112,857	113,714
– algorithm applied successfully	112,580	113,387
Distinctness assertions	380,694	379,724
Node pairs considered distinct	916,554	1,047,299
Lower bound on optimal cost	1,255,111	1,245,004
Cost of our solution	1,306,747	1,294,196
Factor	1.04	1.04
Edges to be deleted (undirected)	1,209,798	1,199,181
Nodes to be merged	603	573

4.3 Linguistic Adequacy

The near-optimal results of our algorithm apply with respect to our problem formalization, which aims at repairing the graph in a minimally invasive way. It may happen, however, that the graph's topology is misleading, and that in a specific case deleting many cross-lingual links to separate two entities is more appropriate than looking for a conservative way to separate them. This led us

to study the linguistic adequacy. Two annotators evaluated 200 randomly selected separated pairs from Dataset A consisting of an English and a German article, with an inter-annotator agreement (Cohen κ) of 0.656. Examples are given in Table 2. We obtained a precision of $87.97\% \pm 0.04\%$ (Wilson score interval) against the consensus annotation. Many of the errors are the result of articles having many inaccurate outgoing links, in which case they may be assigned to the wrong component. In other cases, we noted duplicate articles in Wikipedia.

Occasionally, we also observed differences in scope, where one article would actually describe two related concepts in a single page. Our algorithm will then either make a somewhat arbitrary assignment to the component of either the first or second concept, or the broader generalization of the two concepts becomes a separate, more general connected component.

4.4 Large Problem Instances

When problem instances become too large, the linear programs can become too unwieldy for linear optimization software to cope with on current hardware. In such cases, the graphs tend to be very sparsely connected, consisting of many smaller, more densely connected subgraphs. We thus investigated graph partitioning heuristics to decompose larger graphs into smaller parts that can more easily be handled with our algorithm. The METIS algorithms (Karypis and Kumar, 1998) can decompose graphs with hundreds of thousands of nodes almost instantly, but favour equally sized clusters over lower cut costs. We obtained partitionings with costs orders of magnitude lower using the heuristic by Dhillon et al. (2007).

4.5 Database of Named Entities

The partitioning heuristics allowed us to process all entries in the complete set of Wikipedia dumps and produce a clean output set of connected components where each Wikipedia article or category belongs to a connected component consisting of pages about the same entity or concept. We can regard these connected components as equivalence classes. This means that we obtain a large-scale multilingual database of named entities and their translations. We are also able to more safely transfer information cross-lingually between editions. For example, when an article a has a category c in the French Wikipedia, we can suggest the corresponding Indonesian category for the corresponding Indonesian article.

Moreover, we believe that this database will help extend resources like DBPedia and YAGO that to date have exclusively used the English Wikipedia as their repository of entities and classes. With YAGO's category heuristics, even entirely non-English connected components can be assigned a class in WordNet as long as at least one of the relevant categories has an English page. So, the French Wikipedia article on the Dutch schooner '*JR Tolkien*', despite the lack of a corresponding English article, can be assigned to the WordNet synset for '*ship*'. Using YAGO's plural heuristic to distinguish classes (Einstein *is a* physicist) from topic descriptors (Einstein *belongs to the topic* physics), we determined that over 4.8 million connected components can be linked to WordNet, greatly surpassing the 3.2 million articles covered by the English Wikipedia alone.

5 Related Work

A number of projects have used Wikipedia as a database of named entities (Ponzetto and Strube, 2007; Silberer et al., 2008). The most well-known are probably DBpedia (Auer et al., 2007), which serves as a hub in the Linked Data Web, Freebase[1], which combines human input and automatic extractors, and YAGO (Suchanek et al., 2007), which adds an ontological structure on top of Wikipedia's entities. Wikipedia has been used cross-lingually for cross-lingual IR (Nguyen et al., 2009), question answering (Ferrández et al., 2007) as well as for learning transliterations (Pasternack and Roth, 2009), among other things.

Mihalcea and Csomai (2007) have studied predicting new links within a single edition of Wikipedia. Sorg and Cimiano (2008) considered the problem of suggesting new cross-lingual links, which could be used as additional inputs in our problem. Adar et al. (2009) and Bouma et al. (2009) show how cross-lingual links can be used to propagate information from one Wikipedia's infoboxes to another edition.

Our aggregation consistency algorithm uses theoretical ideas put forward by researchers studying graph cuts (Leighton and Rao, 1999; Garg et al., 1996; Avidor and Langberg, 2007). Our problem setting is related to that of correlation clustering (Bansal et al., 2004), where a graph consist-

[1] `http://www.freebase.com/`

Table 2: Examples of separated concepts

English concept	German concept (translated)	Explanation
Coffee percolator	French Press	different types of brewing devices
Baqa-Jatt	Baqa al-Gharbiyye	Baqa-Jatt is a city resulting from a merger of Baqa al-Gharbiyye and Jatt
Leucothoe (plant)	Leucothea (Orchamos)	the second refers to a figure of Greek mythology
Old Belarusian language	Ruthenian language	the second is often considered slightly broader

ing of positively and negatively labelled similarity edges is clustered such that similar items are grouped together, however our approach is much more generic than conventional correlation clustering. Charikar et al. (2005) studied a variation of correlation clustering that is similar to WDGS, but since a negative edge would have to be added between each relevant pair of entities in a distinctness assertion, the approximation guarantee would only be $O(\log(n\ |V|^2))$. Minimally invasive repair operations on graphs have also been studied for graph similarity computation (Zeng et al., 2009), where two graphs are provided as input.

6 Conclusions and Future Work

We have presented an algorithmic framework for the problem of co-reference that produces consistent partitions by intelligently removing edges or allowing nodes to remain connected. This algorithm has successfully been applied to Wikipedia's cross-lingual graph, where we identified and eliminated surprisingly large numbers of inaccurate connections, leading to a large-scale multilingual register of names.

In future work, we would like to investigate how our algorithm behaves in extended settings, e.g. we can use heuristics to connect isolated, unconnected articles to likely candidates in other Wikipedias using weighted edges. This can be extended to include mappings from multiple languages to WordNet synsets, with the hope that the weights and link structure will then allow the algorithm to make the final disambiguation decision. Additional scenarios include dealing with co-reference on the Linked Data Web or mappings between thesauri. As such resources are increasingly being linked to Wikipedia and DBpedia, we believe that our techniques will prove useful in making mappings more consistent.

A Proofs

Proof (Theorem 1). We shall reduce the minimum multicut problem to WDGS. The hardness claims then follow from Chawla et al. (2005). Given a graph $G = (V, E)$ with a positive cost $c(e)$ for each $e \in E$, and a set $D = \{(s_i, t_i) \mid i = 1 \dots k\}$ of k demand pairs, our goal is to find a multicut M with respect to D with minimum total cost $\sum_{e \in M} c(e)$. We convert each demand pair (s_i, t_i) into a distinctness assertion $D_i = (\{s_i\}, \{t_i\})$ with weight $w(D_i) = 1 + \sum_{e \in E} c(e)$. An optimal WDGS solution $(C, U_1, \dots, U_k)$ with cost c then implies a multicut C with the same weight, because each $w(D_i) > \sum_{e \in E} c(e)$, so all demand pairs will be satisfied. C is a minimal multicut because any multicut C' with lower cost would imply a valid WDGS solution $(C', \emptyset, \dots, \emptyset)$ with a cost lower than the optimal one, which is a contradiction. □

Lemma 4. *The linear program given by Definition 5 enforces that for any i,j,k ≠ j,u ∈ $D_{i,j}$, $v \in D_{i,k}$, and any path $v_0, \dots, v_t$ with $v_0 = u$, $v_t = v$ we obtain $u_{i,u} + \sum_{l=0}^{t-1} d_{(v_l, v_{l+1})} + u_{i,v} \geq 1$. The integer linear program obtained by augmenting Definition 5 with integer constraints $d_e, u_{i,v}, p_{i,j,v} \in \{0, 1\}$ (for all applicable e, i, j, v) produces optimal solutions $(C, U_1, \dots, U_k)$ for* WDGS *problems, obtained as $C = (\{e \in E \mid d_e = 1\}$, $U_i = \{v \mid u_{i,v} = 1\}$.*

Proof. Without loss of generality, let us assume that $j < k$. The LP constraints give us $p_{i,j,v_t} \leq p_{i,j,v_{t-1}} + d_{(v_{t-1}, v_t)}, \dots, p_{i,j,v_1} \leq p_{i,j,v_0} + d_{(v_0, v_1)}$, as well as $p_{i,j,v_0} = u_{i,u}$ and $p_{i,j,v_t} + u_{i,v} \geq 1$. Hence $1 \leq p_{i,j,v_t} + u_{i,v} \leq u_{i,u} + \sum_{l=0}^{t-1} d_{(v_l, v_{l+1})} + u_{i,v}$.

With added integrality constraints, we obtain either $u \in U_i$, $v \in U_i$, or at least one edge along any path from u to v is cut, i.e. $\mathrm{P}(u, v, E \setminus C) = \emptyset$.

This proves that any ILP solution enduces a valid WDGS solution (Definition 2).

Clearly, the integer program's objective function minimizes $c(C, U_1, \ldots, U_n)$ (Definition 3) if $C = (\{e \in E \mid d_e = 1\}$, $U_i = \{v \mid u_{i,v} = 1\}$. To see that the solutions are optimal, it thus suffices to observe that any optimal WDGS solution $(C^*, U_1^*, \ldots, U_n^*)$ yields a feasible ILP solution $d_e = I_{C^*}(e)$, $u_{i,v} = I_{U_i^*}(v)$. □

Proof (Theorem 2). $r_i < \frac{1}{2}$ holds for any radius r_i chosen by the algorithm, so for any region $R(v_0, r)$ grown around a node v_0, and any two nodes u, v within that region, the triangle inequality gives us $d(u, v) \leq d(u, v_0) + d(v_0, v) < \frac{1}{2} + \frac{1}{2} = 1$ (maximal distance condition). At the same time, by Lemma 4 and Definition 6 for any $u \in D_{i,j}$, $v \in D_{i,k}$ $(j \neq k)$, we obtain $d(\mathrm{v}_{i,u}, \mathrm{v}_{i,v}) = d(\mathrm{v}_{i,u}, u) + d(u, v) + d(v, \mathrm{v}_{i,v}) \geq 1$. With the maximal distance condition above, this means that $\mathrm{v}_{i,u}$ and $\mathrm{v}_{i,v}$ cannot be in the same region. Hence u, v cannot be in the same region, unless the edge from $\mathrm{v}_{i,u}$ to u is cut (in which case u will be placed in U_i) or the edge from v to $\mathrm{v}_{i,v}$ is cut (in which case v will be placed in U_i). Since each region is separated from other regions via C, we obtain that $\forall i, j, k \neq j, u, v$: $u \in D_{i,j} \setminus U_i$, $v \in D_{i,k} \setminus U_i$ implies $\mathrm{P}(u, v, E \setminus C) = \emptyset$, so a valid solution is obtained. □

Lemma 5 (essentially due to Garg et al. (1996))**.** *For any i where $\exists j, k > j, u \in D_{i,j}, v \in D_{i,k}$: $\mathrm{P}(\mathrm{v}_{i,u}, \mathrm{v}_{i,v}, E') \neq \emptyset$ and $w(D_i) > 0$, there exists an r such that $w(C(S, r)) \leq 2\ln(nq+1)\,\hat{c}(S, r)$, $0 \leq r < \frac{1}{2}$ for any set S consisting of $\mathrm{v}_{i,v}$ nodes.*

Proof. Define $w(S, r) = \sum_{v \in S} w(C(v, r))$. We will prove that there exists an appropriate r with $w(C(S, r)) \leq w(S, r) \leq 2\ln(nq+1)\,\hat{c}(S, r)$. Assume, for reductio ad absurdum, that $\forall r \in [0, \frac{1}{2})$: $w(S, r) > 2\ln(nq+1)\hat{c}(S, r)$. As we expand the radius r, we note that $\hat{c}(S, r)\frac{d}{dr} = w(S, r)$ whereever $\hat{c}$ is differentiable with respect to r. There are only a finite number of points $r_1, \ldots, r_{l-1}$ in $(0, \frac{1}{2})$ where this is not the case (namely, when $\exists u \in S,\ v \in V' : d(u, v) = r_i$). Also note that $\hat{c}$ increases monotonically for increasing values of r, and that it is universally greater than zero (since there is a path between $\mathrm{v}_{i,u}$, $\mathrm{v}_{i,v}$). Set $r_0 = 0$, $r_l = \frac{1}{2}$ and choose ϵ such that $0 < \epsilon \ll \min\{r_{j+1} - r_j \mid j < l\}$. Our assumption then implies:

$$\sum_{j=1}^{l} \int_{r_{j-1}+\epsilon}^{r_j-\epsilon} \frac{w(S,r)}{\hat{c}(S,r)}\, dr > \left[\sum_{j=1}^{l} r_j - r_{j-1} - 2\epsilon\right] 2\ln(nq+1)$$

$$\sum_{j=1}^{l} \ln \hat{c}(S, r_j - \epsilon) - \ln \hat{c}(S, r_{j-1} + \epsilon) > \left(\tfrac{1}{2} - 2l\epsilon\right) 2\ln(nq+1)$$

$$\ln \hat{c}(S, \tfrac{1}{2} - \epsilon) - \ln \hat{c}(S, 0) > (1 - 4l\epsilon)\ln(nq+1)$$

$$\frac{\hat{c}(S, \frac{1}{2}-\epsilon)}{\hat{c}(S,0)} > (nq+1)^{1-4l\epsilon}$$

$$\hat{c}(S, \tfrac{1}{2} - \epsilon) > (nq+1)^{1-4l\epsilon}\hat{c}(S, 0)$$

For small ϵ, the right term can get arbitrarily close to $(nq+1)\hat{c}(S, 0) \geq \hat{c}_{\mathrm{f}} + \hat{c}(S, 0)$, which is strictly larger than $\hat{c}(S, \frac{1}{2} - \epsilon)$ no matter how small ϵ becomes, so the initial assumption is false. □

Proof (Theorem 3). Let S_i, r_i denote the set S and radius r chosen in particular iterations, and c_i the corresponding costs incurred: $c_i = w(C(S_i, r) \cap E) + |U_i| w(D_i) = w(C(D_i, r))$. Note that any r_i chosen by the algorithm will in fact fulfil the criterion described by Lemma 5, because r_i is chosen to minimize the ratio between the two terms, and the minimizing $r \in [0, \frac{1}{2})$ must be among the r considered by the algorithm ($w(C(D_i, r))$ only changes at one of those points, so the minimum is reached by approaching the points from the left). Hence, we obtain $c_i \leq 2\ln(n+1)\hat{c}(S_i, r_i)$. For our global solution, note that there is no overlap between the regions chosen within an iteration, since regions have a radius strictly smaller than $\frac{1}{2}$, while $\mathrm{v}_{i,u}$, $\mathrm{v}_{i,v}$ for $u \in D_{i,j}$, $v \in D_{i,k}$, $j \neq k$ have a distance of at least 1. Nor is there any overlap between regions from different iterations, because in each iteration the selected regions are removed from G'. Globally, we therefore obtain $c(C, U_1, \ldots, U_n) = \sum_i c_i < 2\ln(nq+1) \sum_i \hat{c}(S_i, r_i) \leq 2\ln(nq+1)2\hat{c}_{\mathrm{f}}$ (observe that $i \leq nq$). Since $\hat{c}_{\mathrm{f}}$ is the objective score for the fractional LP relaxation solution of the WDGS ILP (Lemma 4), we obtain $\hat{c}_{\mathrm{f}} \leq c(C^*, U_1^*, \ldots, U_n^*)$, and thus $c(C, U_1, \ldots, U_n) < 4\ln(n+1)c(C^*, U_1^*, \ldots, U_n^*)$.

To obtain a solution in polynomial time, note that the LP size is polynomial with respect to nq and may be solved using a polynomial algorithm (Karmarkar, 1984). The subsequent steps run in $O(nq)$ iterations, each growing up to $|V|$ regions using $O(|V|^2)$ uniform cost searches. □

References

Eytan Adar, Michael Skinner, and Daniel S. Weld. 2009. Information arbitrage across multi-lingual Wikipedia. In Ricardo A. Baeza-Yates, Paolo Boldi, Berthier A. Ribeiro-Neto, and Berkant Barla Cambazoglu, editors, *Proceedings of the 2nd International Conference on Web Search and Web Data Mining, WSDM 2009*, pages 94–103. ACM.

Sören Auer, Chris Bizer, Jens Lehmann, Georgi Kobilarov, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: a nucleus for a web of open data. In Aberer et al., editor, *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11–15, 2007*, Lecture Notes in Computer Science 4825. Springer.

Adi Avidor and Michael Langberg. 2007. The multi-multiway cut problem. *Theoretical Computer Science*, 377(1-3):35–42.

Nikhil Bansal, Avrim Blum, and Shuchi Chawla. 2004. Correlation clustering. *Machine Learning*, 56(1-3):89–113.

Gosse Bouma, Sergio Duarte, and Zahurul Islam. 2009. Cross-lingual alignment and completion of Wikipedia templates. In *CLIAWS3 '09: Proceedings of the Third International Workshop on Cross Lingual Information Access*, pages 21–29, Morristown, NJ, USA. Association for Computational Linguistics.

Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. 2005. Clustering with qualitative information. *Journal of Computer and System Sciences*, 71(3):360–383.

Shuchi Chawla, Robert Krauthgamer, Ravi Kumar, Yuval Rabani, and D. Sivakumar. 2005. On the hardness of approximating multicut and sparsest-cut. In *In Proceedings of the 20th Annual IEEE Conference on Computational Complexity*, pages 144–153.

Inderjit S. Dhillon, Yuqiang Guan, and Brian Kulis. 2007. Weighted graph cuts without eigenvectors. a multilevel approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(11):1944–1957.

Sergio Ferrández, Antonio Toral, Óscar Ferrández, Antonio Ferrández, and Rafael Muñoz. 2007. Applying Wikipedia's multilingual knowledge to cross-lingual question answering. In *NLDB*, pages 352–363.

Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. 1996. Approximate max-flow min-(multi)cut theorems and their applications. *SIAM Journal on Computing (SICOMP)*, 25:698–707.

Narendra Karmarkar. 1984. A new polynomial-time algorithm for linear programming. In *STOC '84: Proceedings of the 16th Annual ACM Symposium on Theory of Computing*, pages 302–311, New York, NY, USA. ACM.

George Karypis and Vipin Kumar. 1998. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392.

Subhash Khot. 2002. On the power of unique 2-prover 1-round games. In *STOC '02: Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 767–775, New York, NY, USA. ACM.

Tom Leighton and Satish Rao. 1999. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM*, 46(6):787–832.

Rada Mihalcea and Andras Csomai. 2007. Wikify!: Linking documents to encyclopedic knowledge. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management (CIKM 2007)*, pages 233–242, New York, NY, USA. ACM.

D. Nguyen, A. Overwijk, C. Hauff, R.B. Trieschnigg, D. Hiemstra, and F.M.G. Jong de. 2009. WikiTranslate: query translation for cross-lingual information retrieval using only Wikipedia. In Carol Peters, Thomas Deselaers, Nicola Ferro, and Julio Gonzalo, editors, *Evaluating Systems for Multilingual and Multimodal Information Access*, Lecture Notes in Computer Science 5706, pages 58–65.

Jeff Pasternack and Dan Roth. 2009. Learning better transliterations. In *CIKM '09: Proceeding of the 18th ACM Conference on Information and Knowledge Management*, pages 177–186, New York, NY, USA. ACM.

Simone Paolo Ponzetto and Michael Strube. 2007. Deriving a large scale taxonomy from Wikipedia. In *AAAI 2007: Proceedings of the 22nd Conference on Artificial Intelligence*, pages 1440–1445. AAAI Press.

Carina Silberer, Wolodja Wentland, Johannes Knopp, and Matthias Hartung. 2008. Building a multilingual lexical resource for named entity disambiguation, translation and transliteration. In European, editor, *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco.

Philipp Sorg and Philipp Cimiano. 2008. Enriching the crosslingual link structure of Wikipedia - a classification-based approach. In *Proceedings of the AAAI 2008 Workshop on Wikipedia and Artifical Intelligence*.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A Core of Semantic Knowledge. In *Proceedings of the 16th International World Wide Web conference, WWW*, New York, NY, USA. ACM Press.

Zhiping Zeng, Anthony K. H. Tung, Jianyong Wang, Jianhua Feng, and Lizhu Zhou. 2009. Comparing stars: On approximating graph edit distance. *Proceedings of the VLDB Endowment*, 2(1):25–36.