Error Link Detection and Correction in Wikipedia

Chengyu Wang, Rong Zhang, Xiaofeng He; Aoying Zhou
School of Computer Science and Software Engineering
East China Normal University, Shanghai, China
chywang2013@gmail.com, \{rzhang,xfhe,ayzhou\}@sei.ecnu.edu.cn

ABSTRACT
The hyperlink structure of Wikipedia forms a rich semantic network connecting entities and concepts, enabling it as a valuable source for knowledge harvesting. Wikipedia, as crowd-sourced data, faces various data quality issues which significantly impacts knowledge systems depending on it as the information source. One such issue occurs when an anchor text in a Wikipage links to a wrong Wikipage, causing the error link problem. While much of previous work has focused on leveraging Wikipedia for entity linking, little has been done to detect error links.

In this paper, we address the error link problem, and propose algorithms to detect and correct error links. We introduce an efficient method to generate candidate error links based on iterative ranking in an Anchor Text Semantic Network. This greatly reduces the problem space. A more accurate pairwise learning model was used to detect error links from the reduced candidate error link set, while suggesting correct links in the same time. This approach is effective when data sparsity is a challenging issue. The experiments on both English and Chinese Wikipedia illustrate the effectiveness of our approach. We also provide a preliminary analysis on possible causes of error links in English and Chinese Wikipedia.

Keywords
error link; Wikipedia; LinkRank; pairwise learning

1. INTRODUCTION
Wikipedia serves as a valuable data source for knowledge sharing to accomplish various tasks, such as knowledge base population [12, 25], taxonomy construction [13, 20], entity linking [7, 24], etc. Although the collaboratively generated data in Wikipedia contains the “wisdom of the crowds” and is updated on a daily basis, a number of data quality issues exist in Wikipedia, which negatively impact the credibility of Wikipedia, particularly when it is used as the data source to build other knowledge systems. Some of the well-known data quality issues include the lack of cross-lingual links between articles of different language versions [29], missing links between Wikipages [26], vandalism behaviors which intentionally destroy the contents of Wikipages [27], and the controversy issues where contributors have different viewpoints on a certain subject [3]. Research efforts have been focused on these fields to improve the data quality of Wikipedia.

In this paper, we pay attention to a different problem, the error link issue. Error link occurs when an anchor text in one Wikipage points to another Wikipage whose description of the entity is not what anchor text actually means. Error link phenomenon is mostly due to the multiple senses (polysemy) or ambiguity of the anchor text in which a link is created between anchor text and the entity with different meaning. Take a case from English Wikipedia as an example. Wikipage Facebook
gives a brief introduction to Facebook. Anchor text “Java” in this page links to Wikipage Java 2 (an island in Indonesia). But based on the context, we are pretty sure that the contributor of this Wikipage actually refers to the JAVA programming language when mentioning “Java” in sentence “The backend is written in Java ”. With high confidence, we treat the link from Facebook to Java 2 as an error link. This error link can be corrected easily by pointing the anchor text to Java (programming language) 3.

High linking quality can be achieved by frequent checking and validation by contributors of Wikipedia, but manual checking is prohibitively costly because the number of entities and links increases rapidly as new Wikipages are continuously added. For instance, approximately 30,000 new articles are created per month in Wikipedia [3]. Furthermore, as Weaver et al. estimates, the average error rate of Wikipedia statements is 2.8% [30]. Therefore, without automatic checking, linking errors are almost inevitable.

The task of identifying error links and correcting them is important for several reasons: i) it helps to maintain high quality and credibility of Wikipedia contents; ii) it refines the semantic relations between entities in Wikipedia and potentially improves the performance of tasks such as semantic computing; and iii) applications (e.g., knowledge bases such as YAGO [25]) which take Wikipedia as input will benefit from higher quality data source.

To solve the error link problem, we need to study the semantic relations between anchor texts and entities in Wikipedia, which is similar to Entity Linking (EL) [7, 22, 24]. However, existing EL techniques are undesirable to solve the error link task due to the serious data sparsity issue and non-existent ground truth assumption. There are only a small number of error links, while we have to take the entire Wikipedia content and hyperlink structure as input when trying to identify the errors. This data sparsity issue makes it computationally expensive to directly predict whether each anchor text

1https://en.wikipedia.org/wiki/Facebook
2https://en.wikipedia.org/wiki/Java
3https://en.wikipedia.org/wiki/Java_(programming_language)
in Wikipedia is correctly linked to the target Wikipage. Besides, Wikipedia is normally treated as the “ground truth” in EL research. For instance, the prior link probability of a text mention given an entity is computed based on the Wikipedia link structure [22, 23, 24]. However, if we wish to detect and correct error links by linking anchor texts to Wikopages directly, the accuracy may be negatively affected because we need to compute linking quality metrics based on the erroneous link structure.

In this paper, we take a two-stage approach to solve the error link problem. Because error links are mostly caused by the ambiguity of anchor texts, in the first stage, we identify ambiguous anchor texts and extract Anchor Text Semantic Networks (ATSN for short) which capture the hyperlink structure of entities related to those ambiguous anchor texts. A LinkRank algorithm is proposed to generate candidate error links by calculating the “goodness” of the links. In this stage, our goal is to deal with the data sparsity issue by reducing the problem space to a small set of suspicious links. In the second stage, we propose a pairwise supervised learning model on the reduced data set to single out error links with higher precision, while making link correction suggestions in the same time. In summary, we make the following contributions.

- We formalize the error link problem. Based on ATSN, we propose a LinkRank algorithm to generate candidate error links from entire Wikipedia link set. This reduces the problem space considerably.
- We train a pairwise supervised learning model to perform error link detection and correction with higher accuracy on the reduced data set. Graph-based features and context-based features are engineered for the model.
- Extensive experiments are conducted on both English and Chinese Wikipedia to illustrate the effectiveness of the proposed approach. We also perform a preliminary analysis on error links we identified, and present the possible causes.

The rest of this paper is organized as follows. Section 2 summarizes the related work. We define the error link problem formally in Section 3, and introduce our solution briefly. Details of our two-stage approach for addressing the error link problem are described in Section 4 and Section 5. Experimental results are presented in Section 6. We give a preliminary analysis on possible causes of error links in Section 7, and conclude our paper and discuss the future work in Section 8.

2. RELATED WORK AND DISCUSSION

The error link problem is inspired by EL and other similar tasks, which analyze semantic relations between entities and text mentions. In this section, we overview the related work, and provide a discussion on the relation between EL and error link problem.

EL focuses on linking a text mention in natural language input or semi-structured input to a named entity in a knowledge base. Given a text mention \( m \) and a collection of entities \( E_m \), the EL system selects the entity from \( E_m \) that \( m \) most probably refers to. A recent survey on EL can be found in [22]. Various paradigms have been utilized to solve the EL problem, including machine learning-based models [1, 4, 19, 24], graph-based ranking [8, 9], probabilistic-based approaches [6], etc.

Classification models solve the EL problem by predicting whether a text mention refers to a certain entity. Piltz and Paab [19] employ the SVM classifier based on thematic features of text mention’s and entity’s context. Aktolga et al. [1] utilize the logistic regression model for EL to perform classification. However, the number of negative instances are far more than positive instances in the EL task. To handle this imbalance issue, many EL systems adopt the Learning to Rank framework to select the most probable entity. LINDEN [24] utilizes the max-margin technique to learn feature weights of a score function in order to give a rank to all the candidate entities for each text mention. It performs EL by selecting the entity with the highest rank if the score value is higher than a learned threshold. Dredze et al. [4] model the EL task as the optimization problem: given a feature function \( f \), the correct entity \( y \) should receive a higher score \( f(y) \) plus a margin than other entities.

Graph-based ranking methods can improve the effectiveness of the EL system by selecting the entity with highest ranking score. Han et al. [8] propose a referent graph to model the global topological interdependence to make different entity linking decisions in one document. Hoffart et al. [9] develop a similar model which represents the EL model as mention-entity graph. Besides, probabilistic models are employed to perform EL as well. In [6], Han and Sun introduce an entity-mention model based on a probabilistic, generative approach. In the model, the distribution of entities in document, possible text mentions given an entity and possible context of an entity are encoded so that the model can make the linking decision based on these three evidences.

Besides adding links between Wikopages, several works focus on adding links in general documents that link to Wikopages, which is known as Wikification. Wikification can be treated as a generalized EL task which aims to link all the text mentions in a document to the Wikipedia knowledge base. Mihalcea and Csomai [14] develop a system called Wikiify! to label links in the document to Wikipedia articles using keyword extraction and word sense disambiguation techniques. Milne and Witten [15] apply a supervised learning technique to link texts to Wikipedia. Granitzer et al. [5] introduce a content-based strategy which aims to link Wikopages.

Due to the popularity of Wikipedia, other EL-like tasks have been addressed to study the relations between text mentions and entities in Wikipedia. For example, Wikipedia link discovery aims to add links to newly added Wikopages to existing Wikopages. Suner-Can and Birturk [26] combine different approaches by considering the link information, Wikipedia category system and contextual linkiness. In [16], Noraset al. introduce a system 3W to identify mentions in Wikipedia and then add links to their referent entities.

While previous work of missing link discovery enriches the link structure in Wikipedia, the error link problem addressed in this paper tries to correct the errors in the link structure. Wang et al. [28] discuss several data quality issues in Chinese Wikipedia, including the error link problem. Paulheim and Bizer [18] identify incorrect statements in RDF linked datasets, and evaluate the algorithm on DBPedia and NELL. DSNotify [21] is a system to maintain the links between dynamic linked datasets from a resource-centric perspective. Pateman and Johnson [17] propose to highligth the Wikipedia link errors and find possible alternatives by analyzing the “semantic contribution” of Wikipedia links. However, none of the prior work proposes a general framework to detect and correct error links accurately and automatically in Wikipedia. The error link problem has close relations but clear distinctions compared to existing EL (or similar) tasks, discussed as follows.

Both EL and error link problem study the relations between text mentions and entities, which can be addressed considering various signals in the contextual data, including the semantic relatedness between entities, the contextual information of text mentions and entities, external knowledge bases, etc. In this sense, the problem of error link can be regarded as a special case of entity linking. However, error link problem distinguishes itself from existing EL-related tasks in the following aspects: (i) The central task of EL is
3. ERROR LINK PROBLEM IN WIKIPEDIA

In this section, we begin to formalize the error link problem in Wikipedia. Then we give a brief introduction to the two-stage approach which solves this problem.

3.1 Problem Statement

Wikipedia consists of millions of entities and links, which can be treated as a large knowledge repository. Except irrelevant pages such as administrative pages, template pages, etc., each remaining Wikipedia page describes a unique entity [25]. The title of a Wikipedia page is regarded as the name for the entity. In many Wikipedia pages, there are several anchor texts with links to other pages. If the anchor text \( m \) in Wikipedia page \( e_i \) links to Wikipedia page \( e_j \), then it means there is a link \( l_{i,j} \) between \( e_i \) and \( e_j \), and \( m \) is a text mention for entity \( e_j \). An error link occurs if an anchor text \( m \) in Wikipedia page \( e_i \) links to Wikipedia page \( e_j \), while \( e_j \) is not the correct entity to be linked to. Because entities and Wikipedia pages have one-to-one correspondence, for simplicity, we do not distinguish between Wikipedia pages and entities. For example, \( e_i \) can refer to an entity or the Wikipedia page describing this entity.

The goal of error link detection and correction is to discover error links, and try to suggest the correct links. Given the entire Wikipedia dataset \( W \) as input, the proposed approach in this paper automatically generates triples \( l_{i,j}, l_{i,j}' \) such that link \( l_{i,j} \) is an existing error link in \( W \) and \( l_{i,j}' \) is the corresponding correct link. It denotes an anchor text in Wikipedia page \( e_i \) links to Wikipedia page \( e_j \) erroneously, while it should link to Wikipedia page \( e_j' \) instead.

However, we need to point out that Wikipedia is still an incomplete knowledge repository, which covers only a small portion of entities in real world. For error link correction, this work only considers the situation where the correct target entity exists in Wikipedia. Solving error link problem where correct entities are not found in Wikipedia is left to future work.

3.2 Problem Analysis and Solution Overview

The major challenge of error link detection and correction is that there are only a few error links in Wikipedia, leading to the data sparsity problem. Moreover, to the best of our knowledge, there is no prior work or benchmarks available as “ground truth”. Labeling a large number of error links manually is difficult, therefore it is infeasible to apply classification methods directly on the Wikipedia link set due to the data sparsity issue. To alleviate the problem, we split the process into two sub-tasks: i) obtain a candidate error link set that has higher density of error links; and ii) perform supervised error link prediction and correction jointly on the candidate error link set. Accordingly, the solution proposed in this paper adopts a two-stage process to handle the two sub-tasks mentioned above:

- **Candidate Error Link Generation**

  Because the phenomenon of error links is rooted in the ambiguity of anchor text, we mine Wikipedia to construct a dictionary \( M = \{(m, E_m)\} \) where \( m \) is an ambiguous anchor text and \( E_m \) is the set of all possible referent entities for \( m \).

- **Link Classification and Correction**

  In order to solve the error link detection and correction simultaneously, we take a pairwise learning model \( f \). Given \( <l_{i,j}, l_{i,j}'> \) as input, \( f \) predicts whether or not \( l_{i,j} \) is an error link and \( l_{i,j}' \) is a correct link jointly. Link correction can be done according to the prediction results.

For each \((m, E_m) \in M\), we construct an ATSN \( G_m\) to represent the hyperlink structure of entities in \( E_m\). We propose the LinkRank algorithm to calculate the “goodness” of links in \( G_m\) which is used to filter the candidate error links. The candidate error link set \( CL \) is a collection of \(<l_{i,j}, l_{i,j}'>\) pairs where \( l_{i,j} \) is a candidate error link and \( l_{i,j}' \) is the link which is most probably correct.

- **Link Classification and Correction**

  In order to solve the error link detection and correction simultaneously, we take a pairwise learning model \( f \). Given \(<l_{i,j}, l_{i,j}'>\) as input, \( f \) predicts whether or not \( l_{i,j} \) is an error link and \( l_{i,j}' \) is a correct link jointly. Link correction can be done according to the prediction results.

4. CANDIDATE ERROR LINK GENERATION

In this section, we describe how to generate candidate error link set from Wikipedia in detail.

4.1 Dictionary Construction

Wikipedia provides abundant features for the relations between anchor texts and entities. To construct the dictionary \( M \) consisting of ambiguous anchor texts and entities, we utilize data sources such as redirect pages, disambiguation pages and hyperlinks in Wikipedia to extract all the possible referent entities \( E_m \) for anchor text \( m \). The detailed construction method can be found in [24].

After the the dictionary \( M \) is created, it can be used to filter links with ambiguous anchor texts. This greatly reduces the search space of error links. An example of the dictionary is shown in Table 2. For example, if a link with anchor text “New York” points to New York City, then it is likely that this is an error link, because anchor text “New York” can refer to the magazine New York as well.

The dictionary has been heavily used in EL tasks [22, 23, 24]. Compared to existing approaches, the dictionary in the error link task has two significant differences: i) instead of extracting all mention-entity relations, we only focus on ambiguous anchor texts; ii) in EL, the count information for each entity is recorded as prior knowledge [24]. In these approaches, Wikipedia is treated as the “ground truth”. While in this paper, we do not make this assumption, but will check whether links in Wikipedia are correct or not.

4.2 Anchor Text Semantic Network

Let \( G_W = (V_W, L_W) \) denote the Wikipedia link-graph where \( V_W \) is the entity set in Wikipedia and \( L_W \) is the link set among entities. Based on the dictionary \( M \) constructed, for each anchor text \( m \), we construct an ATSN based on \( G_W \).
Table 2: An example of dictionary $M$

<table>
<thead>
<tr>
<th>Anchor Text $m$</th>
<th>Possible Referent Entity Collection $E_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java</td>
<td>Java (programming language)</td>
</tr>
<tr>
<td>New York</td>
<td>New York City (magazine) New York (film)</td>
</tr>
</tbody>
</table>

Formally, an ATSN w.r.t. anchor text $m$ is a weighted directed graph $G_m = (V_m, L_m, W_m)$ where $V_m$ is the node set consisting of entities, $L_m$ is the set of links (directed edges), and $W_m$ is the set of weights for the links in $L_m$.

For an entity $v_i$, its inlink and outlink nodes are defined as follows: $\text{InLinkNodes}_i = \{e_j | l_j,i \in L_m\}$ and $\text{OutLinkNodes}_i = \{e_j | l_i,j \in L_m\}$. Let $\text{Neighbor}(v_i)$ denote the union of inlink and outlink nodes of $v_i$ (i.e., $v_i$'s neighbors). $V_m$ consists of two types of nodes: all entities in $E_m$ and all neighbors of these entities, defined as: $V_m = \bigcup_{v_i \in E_m} \text{Neighbor}(v_i) \cup E_m$. A link $l_{i,j} \in L_m$ exists iff $v_i \in V_m$, $v_j \in V_m$, and $l_{i,j} \in L_m$. Therefore, there are two types of links in $L_m$: inlinks and outlinks of entities in $E_m$, and links that connect neighbors of these entities.

Fig. 1 shows part of the ATSN w.r.t. anchor text “Java”. Entities of the ATSN include i) entities that can be referred as “Java” (e.g., Java, Java (programming language), see Table 2), and ii) the neighbors of them (e.g., Facebook, PHP, etc.). It includes links among these two types of nodes, such as Facebook $\rightarrow$ Java, Java (programming language) $\rightarrow$ PHP, etc.

The ATSN represents the link structure of entities related to the anchor text, therefore the characteristics of error links. It can help us identify candidate error links. The weights $W_m$ can be calculated by LinkRank algorithm, described in Section 4.3.

![Figure 1: Part of the structure of the ATSN w.r.t. anchor text “Java”](image)

### 4.3 LinkRank Algorithm

In this section, we propose the LinkRank algorithm that calculates the “goodness” of links in $G_m$. The result is used as the weights $W_m$ for $G_m$.

In previous work, measurements such as Wikipedia Link-based Measure (WLM) [31] reveal the semantic closeness between two entities, but they only consider the local property of the graph (such as inlinks). However, the link structure in Wikipedia is relatively sparse. For instance, in Chinese Wikipedia, approximately 15% of the entity pairs $e_i, e_j$ have no common inlinks for all $l_{i,j} \in L_W$, which makes it impossible for WLM to measure the semantic closeness. The LinkRank we introduce ranks all links by exploiting the global structure of an ATSN. Similar to PageRank [2] and HITS [11], LinkRank employs an iterative ranking process, but the ranking subjects and the algorithm itself are different from PageRank and HITS. For error link problem, it is important to rank links instead of nodes (entities). The result reveals the “goodness” of a link in an ATSN, instead of a Wikipage.

#### 4.3.1 Iterative Ranking Process

The detailed iterative procedure of LinkRank is shown below, and summarized in Alg. 1.

In the algorithm, the initial weights $w^{(0)}$ of all links are assigned uniformly, i.e., $w^{(0)} = 1$ (Line 3). To update the weights, we use a weight propagation approach according to the link structure of $G_m$. Let $\text{InLinks}_i$ and $\text{OutLinks}_i$ denote the collections of inlinks and outlinks of $v_i$ in $G_m$, respectively, defined as: $\text{InLinks}_i = \{l_{j,i} \in L_m | l_{j,i} \neq i\}$ and $\text{OutLinks}_i = \{l_{i,j} \in L_m | l_{i,j} \neq i\}$. In each iteration $n$, every link $l_{i,j}$ passes its weight uniformly to its outlinks (Line 11). Thus, the transition weight of $l_{i,j}$ in $n^{th}$ iteration is:

$$w_{i,j}^{(n)} = \frac{1}{|\text{OutLinks}_i|} \cdot w_{i,j}^{(n-1)}$$

Denote the links that have zero out-degree in $G_m$ as $\mathcal{T}_m$ (Line 5). These weights can not be passed to other parts of the graph. To deal with this issue, the weights of these links are distributed equally to all the links. Hence, in each iteration, every link $l_{i,j}$ receives the transition weights from $\text{InLinks}_i$ and weights from $\mathcal{T}_m$ (Line 14). Thus, the weight update rule for $l_{i,j}$ is expressed as:

$$w_{i,j}^{(n)} = \sum_{l_{k,i} \in \text{InLinks}_i} w_{k,i}^{(n)} + \frac{1}{|L_m|} \sum_{l_{p,q} \in \mathcal{T}_m} w_{p,q}^{(n-1)}$$

The iterative process converges if the difference of $w^{(n)}$ and $w^{(n-1)}$ is smaller than a small threshold $\epsilon$ (Line 16). We take the weight $w_{i,j}^{(n)}$ for $l_{i,j}$ as rank value, denoted as $w_{i,j}$ (Line 22).

\begin{algorithm}
\caption{LinkRank Algorithm}
\begin{algorithmic}
\State Input: Unweighted ATSN $G_m = (V_m, L_m)$, threshold $\epsilon$.
\State Output: ATSN $G_m = (V_m, L_m, W_m)$.
\State $\mathcal{T}_m = \emptyset$.
\ForEach{$l_{i,j} \in L_m$}
\State $w_{i,j}^{(0)} = 1$;
\EndFor
\If{$|\text{OutLinks}_i| = 0$}
\State $\mathcal{T}_m = \mathcal{T}_m \cup \{l_{i,j}\}$;
\EndIf
\Repeat
\State $n = n + 1$;
\ForEach{$l_{i,j} \in L_m \setminus \mathcal{T}_m$}
\State $w_{i,j}^{(n)} = \frac{1}{|\text{OutLinks}_i|} \cdot w_{i,j}^{(n-1)}$;
\EndFor
\ForEach{$l_{i,j} \in L_m$}
\State $w_{i,j}^{(n)} = \sum_{l_{k,i} \in \text{InLinks}_i} w_{k,i}^{(n)} + \frac{1}{|L_m|} \sum_{l_{p,q} \in \mathcal{T}_m} w_{p,q}^{(n-1)}$;
\EndFor
\Until{$|w^{(n)} - w^{(n+1)}| < \epsilon$}
\State break;
\EndRepeat
\State return $G_m = (V_m, L_m, W_m)$;
\end{algorithmic}
\end{algorithm}

#### 4.3.2 Matrix Interpretation

The iterative ranking process can be represented as matrix computation. We assign each link an integer index from 1 to $|L_m|$ and
use this index to represent the link. Let \( \mathbf{w}^{(i)} \) be an \( |L_m| \times 1 \) weight vector for all links in \( G_m \). \( \mathbf{M} \) is the \( |L_m| \times |L_m| \) weight transition matrix. \( M_{i,j} \) is the proportion of weight that is passed from the \( i^{th} \) link \( l_{p,q} \) to the \( j^{th} \) link \( l_{r,s} \). If \( q = r \), \( M_{i,j} = \frac{1}{|OutLink_j|} \); otherwise, \( M_{i,j} = 0 \). To deal with the links with zero out-degree, we add an additional term to \( \mathbf{M} \) to represent the weights that are distributed among all links. So the weight transition matrix becomes:

\[
\mathbf{S} = \mathbf{M} + \mathbf{a}^T \left( \frac{1}{|L_m|} \right)
\]

where \( \mathbf{a} \) is an \( |L_m| \times 1 \) vector. If the \( i^{th} \) link \( l_{p,q} \) in \( G_m \) has zero out-degree, then \( a_i = 1 \); otherwise, \( a_i = 0 \). The weight vector \( \mathbf{w} \) can be computed recursively as: \( \mathbf{w}^{(i+1)} = \mathbf{S} \cdot \mathbf{w}^{(i)} \). Similar to PageRank [2], the equation has a closed-form solution, i.e., the principal eigenvector of transition matrix \( \mathbf{S} \).

From the random walk perspective, a random surfer stands at any link \( l_{p,q} \) at a time. Each time the algorithm selects a link \( l_{p,r} \) from \( OutLink_q \) with probability \( \frac{1}{|OutLink_q|} \) to make a transfer. If there are no outlinks, the algorithm randomly picks a link, each with probability \( \frac{1}{|OutLink_j|} \). Therefore, we can also view the LinkRank algorithm as the process of calculating the distribution of location of random surfers on the links.

4.4 Candidate Error Link Detection

In this section, we propose a method to generate candidate error link set \( CL \) based on LinkRank.

4.4.1 Measuring Semantic Closeness

In Wikipedia, links between Wikipages have correlation with the semantic closeness between entities. If Wikipage \( e_i \) should link to Wikipage \( e_j \) instead of \( e_i \), then entity \( e_j \) should be semantically closer to entity \( e_i \) than \( e_j \). However, the difficulty is that we cannot directly use the “goodness” of link \( l_{i,j} \) (such as \( w_{i,j} \) calculated by LinkRank) to measure the semantic closeness between \( e_i \) and \( e_j \). Assume \( l_{i,j} \) is an existing error link and \( l_{i,j}' \) is the correct one, but not present in Wikipedia. In this case, \( w_{i,j}' \) is not much meaningful, and \( w_{i,j} \) does not even exist. Other measurements have their own limitation for a sparse link-graph.

In this paper, we take an indirect approach. If entity \( e_i \) is semantically closer to entity \( e_j \) than \( e_j \), then Wikipage \( e_i \) is more likely to connect to the neighbors of Wikipage \( e_j \). Consider the case in Fig. 1. In the graph, Wikipage Facebook links to Wikipage Java instead of Java (programming language). However, Wikipage Facebook links to a lot of Java’s neighbors (i.e., Microsoft, Apache Hive and PHP) while it does not link to Java’s neighbors. This signals that the link from Wikipage Facebook to Wikipage Java is likely to be an error link.

We define the Semantic Closeness (SC) \( SC(e_i \rightarrow e_j) \) as the sum of weights of all links from \( e_i \) to \( e_j \)’s neighbors, denoted as:

\[
SC(e_i \rightarrow e_j) = \sum_{e_{j'} \in Neighbor(e_j) \cap l_{i,j} \in L_m} w_{i,j'}
\]

The meaning of semantic closeness \( SC(e_i \rightarrow e_j) \) implies that: i) large rank value of links indicates there are “good” links from \( e_i \) to \( e_j \)’s neighbors; ii) larger number of links from \( e_i \) to \( e_j \)’s neighbors is a sign of close connection between \( e_i \) and \( e_j \).

4.4.2 Algorithm for Candidate Error Link Detection

We now introduce our algorithm in detail. The procedure is illustrated in Alg. 2.

For an ambiguous anchor text \( m \), let \( CL_m \) denote the candidate error link set w.r.t. \( m \). Consider a link \( l_{i,j} \) in ATSN \( G_m \). If \( e_i \notin E_m \) and \( e_j \in E_m \), then we calculate the semantic closeness between \( e_i \) and all the entities in \( E_m \) (Line 5). Let \( e_j' \) denote the entity in \( E_m \) that \( e_i \) is semantically closest to (Line 7). To check whether \( l_{i,j} \) is an error link, we compare the semantic closeness between \( e_i, e_j \) and \( e_i, e_j' \) pairs. Here, we employ a heuristic rule: if the following inequality holds

\[
\frac{SC(e_i \rightarrow e_j) - SC(e_i \rightarrow e_j')}{SC(e_i \rightarrow e_j')} > \tau
\]

where \( \tau \) is a predefined threshold (\( \tau \in (0, 1) \)), then \( l_{i,j} \) is regarded as an candidate error link, and the most probably correct link is \( l_{i,j}' \). The link pair \( l_{i,j}, l_{i,j}' \) is added to \( CL_m \) (Line 8). For example, given the anchor text “Java”, we search for all possible referent entities from the dictionary and retrieve all the links that point to an entity with the surface name “Java” from the ATSN. For each link (e.g. Facebook \( \rightarrow \) Java), we decide whether it is an error link and enlarge the candidate error link set.

The method avoids direct processing on the entire big link-graph of Wikipedia. We only need to process each ATSN \( G_m \) related to each ambiguous anchor text \( m \) in dictionary \( M \). The final candidate error link set \( CL \) is the union of all \( CL_m \).

Algorithm 2 Candidate Error Link Detection Algorithm

**Input:** ATSN \( G_m = (V_m, L_m, W_m) \), entity set \( E_m \), threshold \( \tau \).

**Output:** Candidate error link set \( CL \) w.r.t. \( m \).

1: \( CL_m = \emptyset \);
2: for each \( l_{i,j} \in L_m \) do
3: if \( e_j \in E_m \) and \( e_j \notin E_m \) then
4: for each \( e_k \in E_m \) do
5: \( SC(e_i \rightarrow e_k) = \sum_{e_{j'} \in Neighbor(e_j) \cap l_{i,j} \in L_m} w_{i,j'} \);
6: end for
7: \( e_j' = \arg\max_{e_j \in E_m} SC(e_i \rightarrow e_j) \);
8: if \( \frac{SC(e_i \rightarrow e_j') - SC(e_i \rightarrow e_j)}{SC(e_i \rightarrow e_j')} > \tau \) then
9: \( CL_m = CL_m \cup \{l_{i,j}, l_{i,j}'\} \);
10: end if
11: end if
12: end for
13: return \( CL_m \);

5. LINK CLASSIFICATION AND CORRECTION

The candidate error link set contains higher density of error links and corresponding possibly correct links. In this section, we propose a supervised pairwise learning technique to predict error links with high precision. We also provide link correction suggestions based on prediction results in the same time.

5.1 Feature Definition

Several signals are useful for identifying error links, including graph-based and context-based features.

5.1.1 Graph-Based Features

The graph-based features can be directly derived from the hyperlink structure of Wikipedia. We do not take the semantic closeness of entities as a feature. This is because we have utilized semantic closeness to identify candidate links. If it is false positive, it will only enforce the error in the training process. Instead, we define the following features from the graph.

Inlink Similarity Feature. The inlink similarity between \( e_i \) and \( e_j \) can be measured as the Jaccard similarity between \( InLinkNode_i \) and \( InLinkNode_j \) in Wikipedia. This is a natural way to measure the graph-based feature. This is a natural way to measure the graph-based feature.
and InLinkNodej. In our statistical analysis, about 15% of entity pairs <ei, ej> have no common inlinks for all links li,j in Chinese Wikipedia, which results in the zero value in Jaccard similarity. However, we have noticed that the number of inlinks of these entities are different. To emphasize the difference, we propose the smoothed Jaccard similarity as the feature, defined as follows:

$$ILS(i, j) = \frac{|InLinkNodei \cap InLinkNodej| + 1}{|InLinkNodei \cup InLinkNodej| + 1}$$

**Outlink Similarity Feature.** Similarly, we define the outlink similarity as follows:

$$OLS(i, j) = \frac{|OutLinkNodei \cap OutLinkNodej| + 1}{|OutLinkNodei \cup OutLinkNodej| + 1}$$

We observe that, if ei should not link to ej, ei’s neighbors have a low probability to connect to ej. For example, in Fig. 1, the neighbors of Facebook (e.g. Microsoft, Apache Hive, PHP) link to Java (programming language), rather than Java. Based on the intuition, we define inlink/outlink relatedness features. In Fig. 2, we present the inlink and outlink relatedness distributions of all Wikipedia links and error links in Chinese Wikipedia.

**Inlink Relatedness Feature.** Inlink relatedness is defined as the fraction of number of entities in InLinkNodei that link to ej and the size of InLinkNodei in total, shown as follows:

$$ILR(i, j) = \frac{|\{ek \in InLinkNodei | lk, j \in Lm\}|}{|InLinkNodei|}$$

**Outlink Relatedness Feature.** Similarly, the observation can be applied to outlinks, too. The outlink relatedness w.r.t. ei and ej is:

$$OLR(i, j) = \frac{|\{ek \in OutLinkNodei | lk, j \in Lm\}|}{|OutLinkNodei|}$$

The inlink/outlink relatedness features can reveal the characteristics of error links. Fig. 2 shows that error links tend to have lower values of inlink and outlink relatedness. Therefore, the graph-based feature vector for li,j is:

$$\vec{v}_{G}(l_{i,j}) = <ILS(i, j), OLS(i, j), ILR(i, j), OLR(i, j)>$$

**Frequent Contextual Similarity Feature.** Using all the n-grams in S1 ∪ S2 to generate word vectors will lead to high-dimensionality due to the large number of n-grams. We only take the top-k most frequent n-grams in S1 ∪ S2 to form word vector representation. Denote FS1 and FS2 as the multisets of frequent n-grams in Wikipage ei and ej, respectively. The frequent contextual similarity feature is defined as:

$$FCS(i, j) = \frac{FS1^T \cdot FS2}{||FS1||_2 \cdot ||FS2||_2}$$

Thus, context-based feature vector for li,j is represented as:

$$\vec{v}_{C}(l_{i,j}) = <CS(i, j), FCS(i, j)>$$

**5.2 Pairwise Learning**

One approach to detect error links is to treat the problem as a binary classification on each link li,j. This requires an absolute measure of “goodness”. In this way, we need to build a model to classify a link first, then design an algorithm to make link corrections.

In the pairwise ranking approach, the model input is a link pair <li,j,li,j'> ∈ CL. Candidate error link li,j can be evaluated together with other links (i.e., a probably correct link li,j') for the same anchor text, which avoids being mapped to a global scale of “goodness” [10]. A set of graph-based and context-based features w.r.t. to a link li,j are engineered, and transformed as follows to fit the pairwise model. Let \(\vec{v}(li,j) = <\vec{v}_{G}(li,j), \vec{v}_{C}(li,j)>\) denote the feature vector for the link li,j. Given a candidate error link li,j and a probably correct link li,j', besides \(\vec{v}(li,j)\) and \(\vec{v}(li,j')\), we generate another feature vector based on the subtraction of the previous two feature vectors, defined as:

$$\vec{v}_{S}(li,j, li,j') = \vec{v}(li,j) - \vec{v}(li,j')$$

As a result, the feature vector for a data instance <li,j, li,j'> for pairwise learning can be represented as:

$$\vec{v}_{PL}(li,j, li,j') = <\vec{v}(li,j), \vec{v}(li,j'), \vec{v}_{S}(li,j, li,j')>$$

After the learning process, we can perform link correction by extracting all the positive data instances. We take all the correct links li,j' as the corrections for error links li,j. The pairwise learning approach introduced above is a general framework in that any classification algorithm can be employed to train the model f. In this paper, we employ Support Vector Machine as the classifier due to its strong discrimination power and wide application.

**6. EXPERIMENTS**

In this section, we conduct comprehensive experiments on English and Chinese Wikipedia datasets to evaluate the performance of our approach. We first illustrate the effectiveness of the candidate error link generation process, then evaluate the pairwise learning model in various aspects. Comparison between our approaches with baseline approaches is also conducted.

**6.1 Datasets**

In the experiments, we use two datasets: English and Chinese Wikipedia dumps\(^1\). We prepossess the datasets by first removing all irrelevant pages such as administrative and template pages because they do not provide information about entities. For remaining Wikipages, we take the titles as names of entities and extract all the hyperlinks between these Wikipages. The detailed statistics are shown in Table 3.

\(^1\)Download website: http://download.wikipedia.com/

\(\text{English version: 20140903 Chinese version: 20140912} \)
6.2 Candidate Error Link Generation

In this section, we present the results on the candidate error link generation process, as well as the comparison results between our method with three baselines.

6.2.1 Baselines

To the best of our knowledge, there is no prior work addressing the candidate error link generation problem. To show the effectiveness of our approach, we set up the following baselines to generate candidate error link set.

- **Simple**: It utilizes Wikipedia disambiguation information to generate candidate error links.\(^3\)
- **AnchorText**: Error link problem is mostly due to the ambiguity of anchor texts. Wikipedia links with ambiguous anchor texts are treated as candidate error links.
- **Unweighted**: Candidate error link pair \(<l_{i,j}, l_{i,j}'>\) is generated by measuring whether or not \(e_i\) is more closely to \(e_{j'}\) than \(e_j\). The major difference between this approach and ours is that we do not use the LinkRank weighting technique.

6.2.2 Experiments and Results

In the experiments, we use different methods to generate candidate link sets, and estimate the percentage of error links. Higher percentage means the method is effective to generate candidate error links. However, it is infeasible to obtain the “ground truth” (i.e., all the error links in Wikipedia) to calculate the percentage. For each experiment, we randomly sample 500 links from candidate error link set, and ask human annotators to check whether they are error links based on the content of Wikipedia. We perform the same experiments using baselines Simple, AnchorText and Unweighted, and our method (denoted as LinkRank) under different values of threshold \(\tau\). The results are shown in Table 4.

From the experimental results, results from Simple and AnchorText are not comparative with the result of our method. If we use the simple method or consider links with ambiguous anchor texts only, it is difficult to generate candidate error links with high density, which shows the serious data sparsity problem in our task.

Unweighted and LinkRank can greatly increase the density of the candidate error links by considering the link structure of an ATSN. LinkRank outperforms Unweighted in all the settings of \(\tau\) for both English and Chinese datasets. It shows the effectiveness of our link weighting technique. Moreover, when \(\tau\) becomes larger, the density of error links increases simultaneously, from 5.6% to 11.6% for English and from 4.0% to 8.4% for Chinese.

Another finding is that the effectiveness of our algorithm is related to the different language versions of Wikipedia. The density of error links for English Wikipedia is higher than that for Chinese Wikipedia in every group of experiments using LinkRank. In Table 3, we can see the average link per article for English Wikipedia is 26, and 12 for Chinese, which means the hyperlink structure of English Wikipedia has higher quality. Because our method is mostly based on the analysis of the hyperlink structure from ATSN, the denser hyperlink structure makes the characteristics of error links in English Wikipedia easier to detect.

6.3 Link Classification and Correction

In this section, we evaluate the performance of the pairwise learning model we proposed in this paper, and compare it with baselines.

6.3.1 Experimental Settings

To apply the link classification and correction models on candidate error links, we sample the candidate set to generate the dataset for training and validation. To address the imbalanced classification issue, we over-sample positive instances by three times for training. The sizes of two datasets (i.e., English and Chinese Wikipedia) are four thousand and two thousand, respectively. Each instance is a tuple \(<l_{i,j}, l_{i,j}'>\), denoting that \(l_{i,j}\) is an candidate error link and that \(l_{i,j}'>\) is a probably correct link. For each instance, we ask human annotators to check the corresponding links in Wikipedia dataset and label them as positive or negative. We use the WEKA\(^6\) toolkit for classification models. For content analysis and the extraction of context-based features, we build up dictionaries containing stop words and meaningless symbols in English and Chinese, respectively. We use the open source Ansj\(^7\) toolkit to perform Chinese NLP analysis such as Chinese word segmentation before generating n-grams.

6.3.2 Overall Performance Evaluation

We first evaluate the methods for link classification and correction. We use 10-fold cross validation on the dataset. Precision, Recall and F-Measure are employed as the evaluation metrics. We introduce methods for evaluating the performance of link classification and correction as follows:

- **PL-Full**: It is the pairwise learning approach for link classification and correction using all the features (Section 5).

<table>
<thead>
<tr>
<th>Method</th>
<th># Error links in sample set</th>
<th>Density of error links</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple</td>
<td>0</td>
<td>0% (approx.)</td>
</tr>
<tr>
<td>AnchorText</td>
<td>0</td>
<td>0% (approx.)</td>
</tr>
<tr>
<td>Unweighted</td>
<td>21</td>
<td>4.2%</td>
</tr>
<tr>
<td>LinkRank ((\tau = 0.2))</td>
<td>28</td>
<td>5.6%</td>
</tr>
<tr>
<td>LinkRank ((\tau = 0.4))</td>
<td>34</td>
<td>6.8%</td>
</tr>
<tr>
<td>LinkRank ((\tau = 0.6))</td>
<td>43</td>
<td>8.6%</td>
</tr>
<tr>
<td>LinkRank ((\tau = 0.8))</td>
<td>58</td>
<td>11.6%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th># Error links in sample set</th>
<th>Density of error links</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple</td>
<td>0</td>
<td>0% (approx.)</td>
</tr>
<tr>
<td>AnchorText</td>
<td>1</td>
<td>0.2%</td>
</tr>
<tr>
<td>Unweighted</td>
<td>17</td>
<td>3.4%</td>
</tr>
<tr>
<td>LinkRank ((\tau = 0.2))</td>
<td>20</td>
<td>4.9%</td>
</tr>
<tr>
<td>LinkRank ((\tau = 0.4))</td>
<td>26</td>
<td>5.2%</td>
</tr>
<tr>
<td>LinkRank ((\tau = 0.6))</td>
<td>32</td>
<td>7.6%</td>
</tr>
<tr>
<td>LinkRank ((\tau = 0.8))</td>
<td>42</td>
<td>8.4%</td>
</tr>
</tbody>
</table>

---

\(^3\)Here is a simple example w.r.t. Java. We extract the disambiguation page (e.g., Java (disambiguation)), find entities related to “Java” in that page (e.g., Java, Java (programming language), etc.), and treat all the links that point to these entities as candidate error links (e.g., Facebook \(\rightarrow\) Java). In this way, we process all the disambiguation pages from Wikipedia and then generate the candidate error link set.

\(^6\)http://www.cs.waikato.ac.nz/ml/weka/

\(^7\)http://nlphina.github.io/ansj_seg/
- PL-G: It is an variant of the approach PL-Full. We only use graph-based features in the implementation.
- PL-C: It is an variant of the approach PL-Full. We only use context-based features in the implementation.

We set up experiments for all the methods mentioned above. The results for English and Chinese Wikipedia datasets are illustrated in Fig. 3. PL-G achieves higher performance than PL-C. It can be seen that the engineered graph-based features have stronger discriminative power than context-based features. The possible reason is that in Wikipedia, the contents of some Wikipages are relatively incomplete. The vector space based n-gram similarity method is not sufficient for distinguishing error/correct links. Combining all the features together, in PL-Full, with proper parameter tuning process, the polynomial kernel SVM with degree \( p = 4 \) and tolerance parameter \( C = 100 \) achieves the highest F-measure 80.3\% for English Wikipedia, and 76.2\% for Chinese Wikipedia with degree \( p = 3 \) and tolerance parameter \( C = 100 \). The results show the engineered features along with the pairwise learning approach achieve higher accuracy than baselines.

![Figure 3: Results for link classification and correction.](image)

### 6.3.3 Comparison with Other Methods

In this section, we make a comprehensive comparison between different approaches for the error link problem. We divide these methods into three categories and compare them with our pairwise learning method. The experimental results are shown in Table 5.

- **VSM Based Methods.** VSM based methods are simple approaches which adopt the Vector Space Model to represent the contents of Wikipages. For an instance \(< l_{i,j}, l_{i,j}' >\), if the content of Wikipage \( e_i \) is more similar to \( e_i \) than that of \( e_j \), the link \( l_{i,j} \) is regarded as an error link. We denote VSM and IntroVSim as the approaches which compare the contents in the whole article and in the introduction part (regarded as the entity summary) to correct error links. The low performance shows the simple method of context similarity comparison can not solve the error link task effectively due to the high-dimensional, sparse representation of the contents.

- **EL Based Methods.** The error link problem can be regarded as a special case of EL, which is discussed in Section 2. We apply EL techniques to correct error links, and experimentally prove that they are not directly capable of correcting error links.

We obtain the implementation of the EL system LINDEN [24] from the authors\(^8\) and re-implement the Wikify! [14] system to correct error links. For an error link \( l_{i,j} \), it regards the content of Wikipage \( e_i \) as the context, and links the anchor text \( m \) in \( e_i \) to an entity \( e_j \) in Wikipedia. If \( e_j \neq e_j \) it predicts \( l_{i,j} \) as an error link. We say the EL system successfully correct an error link \( l_{i,j} \) if it outputs \( e_j = e_j \) where \( l_{i,j}' \) is the correction for the link \( l_{i,j} \).

From the experiments, we can see that LINDEN has a low accuracy to correct error links in English and Chinese Wikipedia. To link text mentions to entities correctly, a lot of measurements need to be computed based on the link structure, e.g. the prior probability of an entity given a text mention, the semantic associativity between entities, etc [23, 24]. Error links affect the performance of these measurements negatively when predicting the correct link by EL. Another finding is that cases of linking errors tend to happen between “tailed” entities where few inlinks/outlinks are added in these Wikipages. This also causes the missing link problem [26], further making the semantic relatedness between entities unavoidably inaccurate. On the contrary, our link correction method is not based on the ranking of candidate entities. It directly predicts the relative “goodness” between two links, considering both link structure and content similarity. Thus it is less sensitive to the missing and error link issue.

### 6.4 Discussion

The error link problem is a seemingly trivial problem due to the fact that there is abundant research on text mentions, entities and their semantic links in the task of EL, word sense disambiguation, link analysis, etc. However, in the previous experiments, we argue that it is difficult to provide a solution for detecting and correcting error links based on existing approaches. We have explored the simple method of detecting error links based on Wikipedia dis-

\(^8\)Note that the YAGO-related features in LINDEN are not added for the Chinese Wikipedia error link set, because there is no Chinese version of YAGO or its equivalence available [28].
Ambiguity between Concepts and Named Entities (ACNE). Some word phrases can refer to common concepts or named entities according to the context. Wikipedia "Tactical role-playing" introduces a type of video games, which links to Wikipage Steam (water in the gas phase). It should link to Wikipage Steam (software), a software platform.

We present some of the error links we have found in English and Chinese Wikipedia in Table 7 and Table 8. For each error link, if it has not been explained above, we give an explanation how the error is occurred and the correct link predicted by our approach.

7.2 Distributions of Categories of Error Links

Based on the preliminary analysis, the distribution of categories of error links is shown in Table 6. For both English and Chinese Wikipedia, error links in the category MSNE account for the majority of all the error links, with the percentage of 75.8% and 83.6%, respectively. The rest of the error links are in the categories ACNE and MSC. The probable causes for the skewed distribution are discussed as follows: i) Wikipedia contains abundant entities but few concepts [25]. Most links tend to point to named entities rather than concepts. As a consequence, most error links are related to named entities. ii) Different senses of named entities can be very similar. In the previous example, both entities related to Bob Gibson are person names (musician and baseball pitcher respectively). In contrast, the senses of the latter two categories are very different, making it difficult for contributors to make the wrong decision when adding links.

8. Conclusion and Future Work

In this paper, we propose to detect and correct error links in Wikipedia effectively. More specifically, the task can be divided into two steps: candidate error link generation, and error link classification and correction. We propose a LinkRank algorithm to detect candidate error links based on ATSN. We employ a pairwise learning technique to determine which are error links and make correction suggestions simultaneously. The experimental results on English and Chinese Wikipedia demonstrate that the proposed approach achieves accurate results. We further present a preliminary analysis based error links in English and Chinese Wikipedia. There are two pieces of future work. Our work only focuses on error links where correct entities exist in Wikipedia. A more challenging problem would be detecting error links where there are no correct links, and finding correct entities from the Web. Besides, although our approach is mainly Wikipedia-centric, it has reasonably wide application for error link detection for Web-scale networks.
Table 7: Cases of error links in English Wikipedia

<table>
<thead>
<tr>
<th>Category</th>
<th>Source Wikipage</th>
<th>Target Wikipage</th>
<th>Correct Wikipage</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSNE</td>
<td>Augustus of Prima Porta</td>
<td>Mars</td>
<td>Mars (mythology)</td>
</tr>
<tr>
<td></td>
<td>Josh White</td>
<td>Bob Gibson</td>
<td>Bob Gibson (musician)</td>
</tr>
<tr>
<td>MSC</td>
<td>Cheltenham Town F.C.</td>
<td>Administration (law)</td>
<td>Administration (British football)</td>
</tr>
<tr>
<td>ACNE</td>
<td>Tactical role-playing game</td>
<td>Steam</td>
<td>Steam (software)</td>
</tr>
<tr>
<td></td>
<td>Ireland in the Eurosong Contest 2011</td>
<td>Lipstick</td>
<td>Lipstick (Jedward song)</td>
</tr>
</tbody>
</table>

1 Augustus of Prima Porta is a marble statue of Augustus Caesar, which has the bas-relief of the Roman god of Mars. The anchor text “Mars” points to the Wikipage describing the planet Mars.

2 Wikipage Lipstick (Jedward song) describes a song by Irish pop duo Jedward. But Wikipage Ireland in the Eurosong Contest 2011 links to Wikipage Lipstick (a cosmetic product) when the contributor refers to the song.

Table 8: Cases of error links in Chinese Wikipedia

<table>
<thead>
<tr>
<th>Category</th>
<th>Source Wikipage</th>
<th>Target Wikipage</th>
<th>Correct Wikipage</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSNE</td>
<td>Theodore Beza* (泰奥多尔·贝扎)</td>
<td>Baden (巴登)</td>
<td>Baden (Switzerland) (巴登 (瑞士))</td>
</tr>
<tr>
<td></td>
<td>Light Rail 705 &amp; 706* (香港轻铁705、706线)</td>
<td>Ginza Station (银座站)</td>
<td>Ginza Stop (Hong Kong) (银座站 (香港))</td>
</tr>
<tr>
<td>MSC</td>
<td>Unit sphere* (单位球面)</td>
<td>Boundary (边界)</td>
<td>Boundary (topology) (边界 (拓扑学))</td>
</tr>
<tr>
<td>ACNE</td>
<td>Donnie Yen# (甄子丹)</td>
<td>Hero (英雄)</td>
<td>Hero (film) (英雄 (电影))</td>
</tr>
<tr>
<td></td>
<td>Zhou Yang (actress) (周扬 (演员))</td>
<td>Tea house (茶馆)</td>
<td>Tea House (TV series) (茶馆 (电视剧))</td>
</tr>
</tbody>
</table>

1 Theodore Beza was a Swiss reformer and scholar, whose hometown was Baden in Switzerland. The link to Beza’s hometown points to a location in Germany whose name is also Baden.

2 Ginza Station is a subway station in Tokyo, Japan. It has the same Chinese name with a light rail stop in Hong Kong. Light Rail 705 & 706 actually goes past the stop Ginza in Hong Kong.

3 Wikipage Boundary describes the dividing line or location between two areas, such as two countries. The word is also a Mathematical term in topology, introduced in Wikipage Boundary (topology). In Wikipage Unit sphere, the contributor uses the word as an anchor text in a topological sense, but carelessly links to the wrong page.

4 Donnie Yen is a Hong Kong actor who starred in the film Hero in 2002. Wikipage Hero describes the general concept.

5 Zhou Yang is a Chinese actress who starred in the TV series Tea House. Wikipage Tea house describes the place where people drink tea.

We will also extend the error link detection technique to heterogeneous information networks (e.g. DBLP network), knowledge graphs (e.g. DBPedia) and other types of data in the future.

Acknowledgements. This work is supported by the National Key Research and Development Program of China under Grant No. 2016YFB1000904.

9. REFERENCES


