Electronic Preprint for Journal of Information Processing Vol.26

Regular Paper

# Estimating Reference Scopes of Wikipedia Article Inner-links

Renzhi Wang[1,a)] Mizuho Iwaihara[1]

Received: December 10, 2017, Accepted: April 6, 2018

**Abstract:** Wikipedia is the largest online encyclopedia, utilized as machine-knowledgeable and semantic resources. Links within Wikipedia indicate that two linked articles or parts of them are related each other about their topics. Existing link detection methods focus on linking to article titles, because most of links in Wikipedia point to article titles. But there is a number of links in Wikipedia pointing to corresponding specific segments, such as paragraphs, because the whole article is too general and it is hard for readers to obtain the intention of the link. We propose a method to automatically predict whether a link target is a specific segment or the whole article, and evaluate which segment is most relevant. We propose a combination method of Latent Dirichlet Allocation (LDA) and Maximum Likelihood Estimation (MLE) to represent every segment as a vector, and then we obtain similarity of each segment pair. Finally, we utilize variance, standard deviation and other statistical features to produce prediction results. We also apply word embeddings to embed all the segments into a semantic space and calculate cosine similarities between segment pairs. Then we utilize Random Forest to train a classifier to predict link scopes. Evaluations on Wikipedia articles show an ensemble of the proposed features achieved the best results.

**Keywords:** Wikipedia, link suggestion, LDA, word embedding, PMI

## 1. Introduction

Wikipedia articles are edited by various volunteers from all over the world, with different thoughts and styles. Wikipedia is structured via a number of links between different articles, which imply that the two linked articles are closely related. In this paper, we call by a *segment* a logical text unit that can be a link target, such as a section and a subsection. Majority of links within Wikipedia are pointing to article titles, and only small fractions point to segment titles. However, when readers browse topic via links, sometimes they are only interested in certain segments while the link itself is pointing to article titles, thus the readers could get lost in such long articles. To avoid such situations, administrators and editors often modify link target text from an article title to a specific segment title. **Figure 1** shows an example that an editor modified the link target from the article title to the specific segment. In article "Super Mario 64," there is a link, first pointed to the whole article GameCube, and then the editor corrected the link to its segment "Controller".

Current link detection methods are focusing on links to article titles [1], [2], [7], [12], [13]. They often first generate a candidate set for a link origin by analyzing existing link connections between articles, then rank all articles in the candidate set, and select the most relevant article as the link target. But in our case, it is hard to find a candidate set by using existing link connections, due to shortage of links that point to segment titles. Besides, the length of segment texts are usually short, so it is necessary to design appropriate feature representation of segment texts that can capture latent semantic relationships. Then we can perform accurate similarity comparisons between segment pairs, which is crucial in the candidate set generating process.

In this paper, we discuss the following link suggestion problem: Given a link source that is a position in a Wikipedia article, we find the most likely link target, which is either a whole article, or a segment in an article. In general, segments of an article are forming a nested structure, such as sections, subsections, and paragraphs. Determining the level of the target segment can be an interesting research issue, but in this paper, for simplicity we decompose one article into non-overlapping segments corresponding to sections. Earlier versions of this paper [19], [20] discussed applying Latent Dirichlet Allocation (LDA) [3], [4] method for topic detection, where segments are represented as vectors of word probabilities. In this paper, we discuss im-

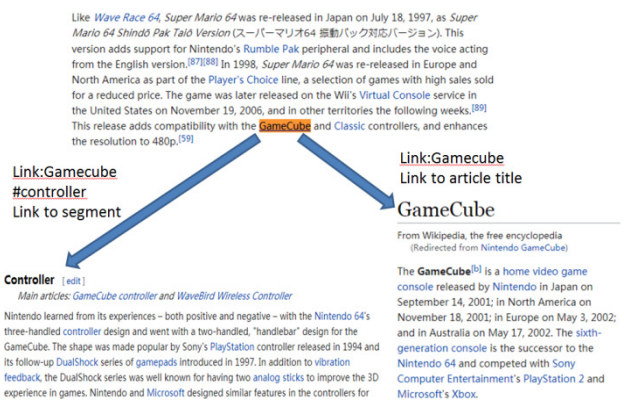



**Fig. 1** Wikipedia links.

[1] Waseda University, Kitakyushu, Fukuoka 808–0135, Japan
[a)] ouninnyuki.ips@asagi.waseda.jp

proving accuracies of the LDA-based methods, by combining LDA with Maximum Likelihood Estimation (MLE) in a nonlinear way, which enables us to compute topical similarities on the segment level. We call this method *smoothing LDA*, or *smLDA* for short. Furthermore, considering about effectiveness of word co-occurrence, we utilize features based on Normalized Point-wise Mutual Information (NPMI) [5], to measure likelihoods of words co-occurring in different segments. We also consider semantic similarities between segment pairs, utilizing word embeddings. Word2vec [16], [17] is an open source project released by Google which achieves state of the art performances in various natural language tasks. In our research, we utilize the word2vec model to embed words to vectors.

It is hard to predict whether a link should point to a segment, by only using similarities between segment pairs. We compute similarities between one target segment and all the segments in another article, to obtain similarity distributions in one article. We define statistical features based on these similarity distributions. Then we train a classifier to determine whether the link should point to a specific segment rather than the whole article. When we confirm that the link should point to a segment, we compare the similarities between segment pairs to find the most related segment. To solve the imbalanced data problem, we utilize logistic regression as a filter before final prediction. Our evaluation results show that our method is effective.

The rest of this paper is organized as follows. Section 2 shows related work. Section 3 introduces similarity measures suitable for segment pairs. In Section 4 we derive statistical features on similarity distribution, to be used for predicting link targets. In Section 5 we describe our datasets in detail, explain our experimental process and we present evaluation results in various situations. In Section 5, we address a conclusion and future work.

## 2. Related Work

Automatically discovering missing links in Wikipedia has been discussed in the literature. Adafre et al. [1] propose a method which can rank pages using co-citation and page title information. They use LTRank to identify similar pages and select top similar articles as the prediction results. This method utilizes organizational structures of articles. It first creates a full title representation of Wikipedia page $d$ by collecting all the titles of the pages that cite $d$. Then the title representation is submitted to a search engine as a query in order to retrieve pages that are similar to $d$. However, LTRank is not suitable for detecting links at the segment level, because there are not enough segment-level links, and Wikipedia segment titles are often short, like "early life," which are insufficient to evaluate semantic relatedness.

Zhang proposed a method utilizing TF-IDF and the vector space model to detect document-to-document links and anchor-to-BEP links [22]. Best Entry Point (BEP) is similar to our task. The best entry point here is a specific article belonging to a general article. The difference here is that Zhang's work is still focusing on the whole article, while our task is to detect the best segment in an article. Zhang's research regards the source article as the query and selects top-k similar articles as the result. Then deep-first iteration is repeated to find the final result. This method uses TF-IDF and the vector-space model to compute similarities. However, since TF-IDF relies on explicit term matching, its results are heavily affected by corpus. Since each segment of one article is limited in length, not enough terms can be extracted, so the TF-IDF method performs worse than other methods that also utilize latent semantic relationships. In this paper, we discuss an integrated method of term probabilities, LDA, point-wise mutual information, and word embeddings.

Milne et al. [13] proposed a machine learning-based link detector to detect links between Wikipedia articles, but the work is also the article level. In their method, they did not simply evaluate textual similarity between two articles, but for each article pair, they evaluate five features: link probability, relatedness, disambiguation confidence, generality, location and spread. Then they train a classifier, for predicting whether there should be a link between an article pair. Its result was much better than other similarity-based methods.

## 3. Similarity Measures on Segment Pairs

### 3.1 Overview of the Proposed Method

**Figure 2** shows our frame work of link-scope prediction. We first extend the corpus by augmenting with related articles, then divide the target articles into segments. For similarity values, we utilize three measures: 1) smoothing LDA, which couples LDA with obvious word probabilities, 2) word2vec, and 3) segment-level PMI. For smoothing LDA and word2vec, pivoted cosine similarities of segment pairs are used to avoid giving excessively higher scores to short segments. On the other hand, PMI in this paper directly gives a similarity value on each pair of segments. In total, three types of similarity values are computed on each segment pair. Then statistical features based on the similarity distribution on segment pairs are computed (Section 4). Finally, a random forest classifier is used to determine whether the link should point article title or one of the segment titles.

### 3.2 Extending Corpus

As the world's largest online encyclopedia, Wikipedia is organized as a large, complex network, where articles are connected

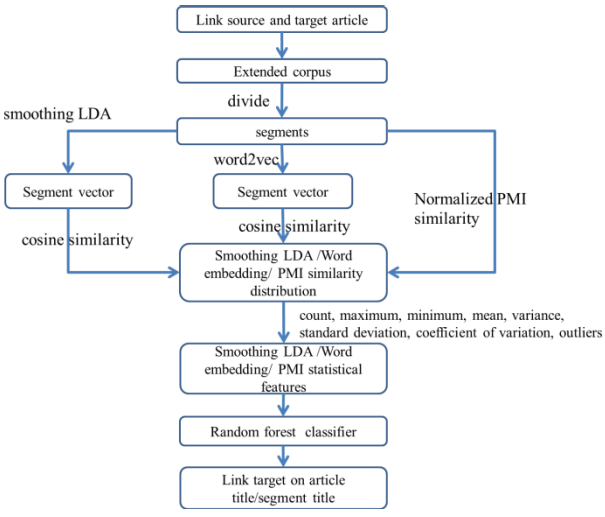



**Fig. 2** Overview of predicting link scopes.

by interlinks. Given a target, we assume that interlinks from a target article to another article complement the contents of the target article, by incorporating the contents of the linked articles. In order to construct a corpus that covers enough terms related to the topics of the target article, we collect articles having links from the target article. Given a target article $A$, we regard the union of all the articles that $A$ has a link to and $A$ itself as the corpus. Certain links may point to a specific segment, but we include its whole article into the corpus. Wikipedia featured articles are well-written, less erroneous and stable, so they are suitable for our experiments. Our evaluation dataset consists of randomly sampled featured articles. Since our objective is to suggest a segment-level link, we decompose the articles in the corpus into segments based on their logical structures, such as paragraphs. The following steps are operated on the segments in the corpus.

### 3.3 Smoothing LDA

To determine whether a link should point to one article or its segment title, our approach is to compare semantic similarities between the target segment and the whole article, and each of its segments. Here, segments are represented as vectors of similarity values, and cosine similarities between the vectors of segment pairs are computed. For similarity measures, we adopt models based on explicit word co-occurrence, latent topics (LDA), and word relatedness (PMI and word embeddings). Then we discuss a number of techniques to integrate these similarity measures.

We argue that linked articles bring additional information to the target article and affect the topics of the target article. The well-known LDA model [3], [4] can extract latent topics from the corpus. In the LDA model, documents are regarded as topic distribution and topic is regarded as a word distribution. A document $d$ is sampled from a topic distribution $\theta$, and a topic $z$ is represented over words by word distribution $\phi$.

We can obtain the probability of a term in a document by the following formula:

$$P_{LDA}(w \mid d, \hat{\theta}, \hat{\phi}) = \sum_{z=1}^{K} P(w \mid z, \hat{\phi}) P(z \mid \hat{\theta}, d) \tag{1}$$

Here, $\hat{\theta}$ and $\hat{\phi}$ are the posterior estimates of $\theta$ and $\phi$, respectively, and $K$ is the given number of latent topics. LDA does not perform very well on long tail words, so there are a number of variants over the basic LDA model. One of the variants is to combine LDA and Maximum Likelihood Estimation. The authors of Refs. [8], [9], [18], [20], [21] utilize linear combination of word-level probabilities from LDA, document-level probability and collection-level probability to smooth the results. The document-level probabilities and collection-level probabilities are obvious parts which can be observed by simple term occurrences. The LDA model can estimate the word probabilities from latent topics which can be regarded as latent part. However, the existing methods [8], [9], [18] adopted a linear combination of the obvious part and latent part. But in the assumption of LDA, the word probability of the document is based on the corpus while the document-level probability is based on the current document, so linear combinations may not be the best choice, because it is ad hoc to current documents. To optimize the combination, we propose the following nonlinear combination of the obvious part and latent part:

$$\begin{aligned} p(w \mid D) \;=\; & \frac{1}{e^{\alpha N_d}+1}\left[\frac{1}{e^{-\beta N_d}+1} P_{ML}(w \mid D) \right. \\ & \left. + \frac{1}{e^{\beta N_d}+1} P_{ML}(w \mid Coll)\right] \\ & + \frac{1}{e^{-\alpha N_d}+1} P_{LDA}(w \mid D) \end{aligned} \tag{2}$$

Here, $N_d$ is the number of terms appearing in the segment. The first part $P_{ML}(w \mid D)$ of the formula is the probability of the word $w$ appearing in the document $D$ by term frequencies. The weight proportion of the obvious part and latent part will affect the word probability. This document-level probability is combined with the collection-level probability $P_{ML}(w \mid Coll)$ by the smoothing parameter $\beta$. We adjust the value of $\beta$ to optimize the obvious part. Smoothing parameter $\alpha$ is to adjust the ratio of the obvious part and latent part. By these two formulae, we can obtain the probabilities of all the words in the corpus. In Eq. (2), the word probabilities are combined by sigmoid functions on parameters $\alpha$ and $\beta$. Sigmoid functions continuously range between 0 and 1, and their change rate around the two sides (0 and 1) is small, which fits for long tail distribution well, and we can also estimate the change rate easily by changing the parameters $\alpha$ and $\beta$. These properties make sigmoid functions suitable to smooth the nonlinear combination. Perplexity has been used to evaluate performance of generative language models. The perplexity is smaller when the fitness between the model and data is better. We evaluated perplexity over the corpus to determine the parameter combinations where the perplexity is minimal. In our experiment, we set the topic number k of the LDA model as 100, and we determined the optimum smoothing parameters $\alpha$ as 0.3 and $\beta$ as 0.8 from our experimental dataset.

We could use all the words in the corpus as elements of the vector. But to reduce the dimensions of the vectors, we only select words which appear in more than three segments.

### 3.4 Word Embeddings

Another effective method for representing a document as a vector is word embeddings [16], [17]. Its input is a large text corpus and output is a vector on reals for each unique word. The tool word2vec embeds all the words into a low dimension space, where each word is represented as one point in this vector space, and the relatedness between two words is measured as the cosine similarity between the two vectors of the words. Document2vec [10] represents documents as vectors, but for new documents it needs to retrain the model. Since our test document set is disjoint from the training document set, docment2vec does not fit with our algorithm. Thus we select word2vec as our word embedding model. We define segment vectors as the sum of all the word vectors in the segment. We compute the cosine similarity between segment pairs as their semantic similarity.

### 3.5 Pointwise Mutual Information

Pointwise mutual information (PMI) is a measure of association developed in information theory and statistics. In NLP tasks, PMI has been often used for finding collocations and association

between words. PMI measures how likely two words are to co-occur. Our objective here is to compute the similarity between two segments, rather than words. Mihalcea et al. [11] propose a method to compute the similarity between two sentences based on the normalized PMI of all word pairs in these two sentences. The scoring function is as follows:

$$sim(S_1, S_2) = \frac{1}{2}\left(\frac{\sum_{w\in\{S_1\}}(maxSim(w, S_2) * idf(w))}{\sum_{w\in\{S_1\}} idf(w)} + \frac{\sum_{w\in\{S_2\}}(maxSim(w, S_1) * idf(w))}{\sum_{w\in\{S_2\}} idf(w)}\right) \quad (3)$$

Here, $maxSim(w, S_1)$ is the maximum lexical similarity between the word $w$ in segment $S_1$ and all the words in segment $S_2$, calculated by normalized PMI. $maxSim(w, S_2)$ is defined in a similar manner. $idf(w)$ is the inverse document frequency of the word $w$ calculated from the corpus. This similarity score ranges between 0 and 1, with a score of 1 indicating identical segments, and a score of 0 indicating no semantic overlap between the two segments. In the PMI method, we do not represent segments as vectors, but directly calculate similarities of segment pairs.

Considering the text length of each segment is diverse, the length may affect the similarity value. Thus we smooth all above similarity by pivoted document length normalization [15].

## 4. Predicting Link Scopes

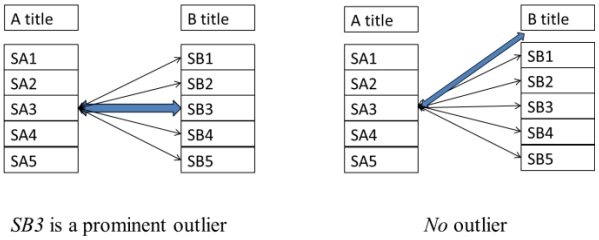
### 4.1 Strategies

A Wikipedia interlink may point to an article title or a segment title. Wikipedia's guideline [*1] specifies on links to sections that "If an existing article has a section specifically about the topic, you can redirect or link directly to it, by following the article name with a number sign (#) and the name of the section." This gives us a reason to find from the target article a segment which is remarkably sharing common topics with the link source, for automatic generation of segment-level links.

Let us re-consider the example of the link from segment "Re-releases and remarks" in Wikipedia article "Super Mario 64" in Fig. 1. The link on source "GameCube" is linked to target segment "Controller" of article "GameCube." On date June 6th, 2016 there are 12 segments in "GameCube." In the sentence containing the link source "GameCube," word "controllers" is also seen, which suggests that the context of referring "GameCube" is about topic "controller." In fact, the target segment "Controller" in article "GameCube" has more occurrences of "controller" than any other segments, so we can find the target segment by cosine similarities on obvious word vectors. On the other hand, if topic relatedness between segments is implicit, we need to rely on similarity measures that can capture corpus-level word relatedness, such as PMI and word embeddings. In addition to such topical similarity, it is desirable that link targets are more informative than the link source, in terms of comprehensiveness and/or generalities, but these quality measures deserve future studies.

As Wikipedia's guideline dictates, a certain Wiki should point to a segment if the segment is specifically about the topic. However, if none of the segments are particularly related to the link source, or most of the segments contain more or less related words, then the link should point to the article title. So we need to compare similarity values of target segments, to determine whether the link should point to one of the segment titles or the article title.

[*1] https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Linking#Section_links




*SB3* is a prominent outlier *No* outlier

**Fig. 3** Outlier of similarity distribution.

Suppose that we determine whether a link from a segment $SA_i$ in article A points to the specific title of a segment $SB_j$ in article B or the article title of B (**Fig. 3**). If segment $SB_j$ is prominently more relevant to $SA_i$ than the other segments in B, then $SB_j$ is specifically more similar with $SA_i$ than the whole article B, so $SB_j$ should be chosen as the link target (Fig. 3 left). But if all the segments in article B are not strongly similar with segment $SA$, we need to determine either we do not create a link to article B or create a link to the title of B (Fig. 3 right). To realize this approach, we adopt the following assumption.

Assumption: If segment $SA_i$ should have a link to segment $SB_j$, then $SB_j$ should be the most related segment with $SA_i$ in article $B$, and the other segments in $B$ are just slightly related with $SA_i$. In other words, if we rank the segments $SB_j$ in article $B$ by the similarities with $SA_i$, then the similarity between $SA_i$ and $SB_j$ should be a prominent outlier. If segment $SA_i$ should be linked to the article title of $B$, then all the segments in $B$ should be slightly related with $SA_i$, but there is no obvious outlier.

### 4.2 Statistical Features on Similarity Distribution

Based on the above assumption, we construct our feature set as follows. Given articles A and B and a segment $SA_i$ in A, we define the *similarity distribution* of article B with respect to segment SA as the set of similarity values between $SA_i$ and each segment in B. Now we define the following features which characterize distribution of similarity values in the segments of article B:

- Range of similarity distribution.

We note that similarity values are quite diverse between segments. Therefore, we characterize the segments if one article by descriptive statistics of similarity distributions as follows: The *number of the segments* in the article, and *maximum, minimum* and *mean of the similarity values* of the segments in the article. For example, suppose that article B has three segments $SB_1$, $SB_2$, and $SB_3$ and the similarity values between $SA$ and $SB_1$, $SB_2$, $SB_3$ are 0.5, 0.2, and 0.1, respectively. Then the number of the segments is three, the maximum, minimum and mean of the similarity values are 0.5, 0.1, and 0.3, respectively.

- Dispersion of similarity distribution.

Based on our assumption, if a link should point to a segment there will be at least one segment in the link target article which is highly similar to the link source. In an ideal situation, if the link points to an article title, not a specific segment, then the sim-

ilarities of all the segments in the article toward the link source segment are close between each other, so the dispersion of these similarity values is small. Thus dispersion of similarities within one article is an important clue for determining whether the link should be on the article level or segment level. Therefore we introduce the following statistical features: *variance*, *standard deviation*, and *coefficient of variation*.

- Outliers.

According to our assumption, if a link should point to a segment title, then it is more likely that there exists an outlier segment, having an outstandingly larger similarity than other segments. Z-score is often used to detect outliers, which is defined as $(x_i - \bar{X})/S$ where $x_i$ is one sample, $\bar{X}$ and $S$ are the mean and standard deviation of the samples, respectively. Samples having z-scores outside of range $[-2, 2]$ are commonly considered as outliers. In our case, similarity values between segments are quite diverse, so that in this paper, we consider a sample whose z-score is outside of range $[-3, 3]$ is an outlier. We introduce *the number of outliers* as one of our features.

In summary, we introduce the following nine statistical features: number of the segments, max similarity, min similarity, mean similarity, variance, standard deviation, coefficient of variation, small outlier count and large outlier count.

### 4.3 Prediction

We introduced statistical features that characterize similarity distribution on segment pairs, where similarities are measured by smoothing LDA, which focuses on topic similarity, while word2vec and PMI models compute word-level relatedness between segment pairs. To combine all the features, we train a classifier based on these features to predict whether a target link points to an article title or a specific segment. The features on similarity distributions do not have obvious linear relationships, so we utilize the nonlinear classifier random forest [6] as our classifier. There are a number of parameters in the random forest model. We determine all the parameters based on our real dataset. The most important two parameters are n_estimetors, which is the number of subtrees, and max_depth, which is the maximum depth of subtrees. The model becomes stable when n_estimetors is sufficiently large. The optimal value of max_depth depends on data distribution. In our experiment we set n_estimetors as 50 and max_depth as 8.

In Wikipedia, the number of links pointing to segments and the number of links pointing to article titles are heavily imbalanced. In training a binary classifier on such an imbalanced dataset, oversampling and under-sampling are often used. In this paper, we adopt the following filtering strategy: First we apply logistic regression to estimate the probability of a link being segment-level before we train the random forest model. In the step we filter the training data by removing samples with low probabilities by logistic regression. We believe the filter step can remove negative samples from the dataset that be used to train random forest model and improve the accuracy.

When a target link is predicted as pointing to a segment, we need to determine which segment should be its target. In this step, instead of simply selecting the most similar segment as the target, the selected segment must be an outlier in the similarity distribution.

## 5. Experimental Evaluation

### 5.1 Dataset

As for the reference data of our evaluation, we face a problem such that a considerable portion of Wikipedia articles are not properly given article-level or segment-level links, since the Wikipedia guideline is often not observed. To evaluate on properly linked articles, we utilize featured articles of Wikipedia. Wikipedia nominates a number of high quality articles that reach the standard of the featured article criteria as featured articles, which are supposed to be well-written, comprehensive, well-researched, neutral and stable. These articles are usually edited by experienced authors and checked by Wikipedia's administrators. Thus we adopt links from featured articles as our golden standard, assuming that their links are appropriately given, pointing to segments when there are specifically relevant segments in the target articles. We randomly selected 1,000 featured articles as our dataset. **Table 1** shows the detail of the dataset.

There are totally 1,689 links pointing to segments and 153,918 links pointing to article titles. We use these links as our reference dataset. The ratio between the segment links and article links is nearly $1 : 100$. This ratio is rarely observed in most of articles, because the average number of links in one article is 41, so most of articles just have article-level links. We divide the dataset into small subsets, to control the ratio between positive and negative samples, where positives are segment-level links.

### 5.2 Preparement

In the step of deriving vectors from segments in the word2vec model, we train the model by the whole English Wikipedia, where the vector dimension is set to 300. We use the sum of all the vectors of the words in a segment to obtain its segment vector.

From the dataset, we observe that the ratio between positive and negative samples is imbalanced. We need a careful setup for training, because the classifier tries to adjust the parameters to put all samples into the correct classes. If negative samples are overwhelming majority, the positive samples could be regarded as invalid values by the classifier. There are two approaches to tackle this problem. The first one is randomly sampling negatives to balance positives and negatives, and then train using this sampled set. But its disadvantages are obvious. This sampling process will lose a large number of effective data. The lost negative samples will cause learning errors, degrading precision.

Another solution is resampling positives until positives and negatives are balanced. But this oversampling can cause over fitting easily, and it does not help the classifier to learn positive samples, because resampling does not increase new positive sam-

**Table 1** Dataset infomation.

| Dataset | Count |
|---|---|
| Articles | 1000 |
| Segments | 7478 |
| Links pointing to segments | 1689 |
| Links pointing to article titles | 153918 |

ples. Instead it just balances the dataset by repeating addition of limited positive samples.

In the experiments, we perform sampling negatives to balance the dataset. We randomly sample negatives several times, and train the classifier. We set the parameters as 50 trees for the forest and the tree depth as 5. For the filter in the stacks, we adopt logistic regression which penalizes L2 norm to avoid over fitting. We set as 0.5 as the minimum probability of filtering by logistic regression.

For the prediction results, we calculate average precision, recall, and F1-score.

### 5.3 Feature Importance

Before we evaluate on the dataset, first we have to measure whether each proposed feature is effective to distinguish the positive class. We conduct correlation analysis to test significance of each feature against the reference data. Mann-Whitney is a nonparametric test of the null hypothesis that two samples come from the same population against an alternative hypothesis, such that a particular population tends to have larger values than the other. The test results are shown in **Table 2**.

From the test we can find max similarity, mean similarity, variance, standard deviation, small similarity outlier are related to the reference classification. But in our assumption, the large similarity outlier should be related. For further analysis, we performed the Two-Sample Kolmogorov-Smirnov test. This is a nonparametric test for equality of continuous, one-dimensional probability distributions that can be used to compare a sample with a reference probability distribution (one-sample K–S test), or to compare two samples (two-sample K–S test). The results are shown in **Table 3**.

We can see in Table 3 that the small similarity outlier is not quite effective, but from these two tests, we can see variance and standard deviation are strongly effective. These significance results indicate that our features are effective to distinguish segment links via these features. Therefore, the test results support our assumption on similarity distribution.

**Table 2** Mann-Whitney test on features.

| Feature | Significance | decision |
|---|---|---|
| Element count | 0.454 | retain |
| Max similarity | 0.018 | reject |
| Min similarity | 0.423 | retain |
| Variance | 0.023 | reject |
| Mean similarity | 0.002 | reject |
| Coefficient of variation | 0.611 | retain |
| Standard deviation | 0.023 | reject |
| Small similarity outlier | 0.010 | reject |
| Large similarity outlier | 0.079 | reject |

**Table 3** Kolmogorov-Smirnov test on features.

| Feature | Significance | Decision |
|---|---|---|
| Element count | 0.227 | retain |
| Max similarity | 0.000 | reject |
| Min similarity | 0.636 | retain |
| Variance | 0.000 | reject |
| Mean similarity | 0.001 | reject |
| Coefficient of variation | 0.714 | retain |
| Standard deviation | 0.000 | reject |
| Small similarity outlier | 0.821 | retain |
| Large similarity outlier | 0.612 | retain |

### 5.4 Baseline

Previous researches [1], [13], [22] are all focusing on linkability to article titles, not targeting to segments. Since there is no preceding work, we choose our baseline based on a simple idea that if more than half of non-stop words in the link source occur in segment title or segment content, then the link should point to the segment title. Otherwise, if more than half of non-stop words in link source do not occur segment title or segment content, then the link should point to the article title. If the words occur in multiple segments, then the segment contain the words most is chosen as the target of the link.

### 5.5 Classification Result

To explore the effect of the W2V dimension on the results, we tested on three different dimensions (50, 100, 300). All these three dimensions are trained by the whole English Wikipedia. The results are shown in **Table 4**, where a gentle improvement of precision is observed as the dimension increases.

Our experiments so far utilized W2V trained on 300 dimensions. Our first step is to determine whether a link should point to an article title or segment. We controlled the ratio of positives and negatives from 1 : 1 to 1 : 100. All the samples are randomly selected from the dataset. In each dataset, we use a random half data for training and the remaining half data for testing. **Tables 5** to **8** show the results of using random forest as the classifier. In Tables 5 to 8, the notation such as SimDist(smLDA + W2V) means the input of classifier is the similarity distribution where similarities are given by smoothing LDA and W2V.

The second step is to determine which segment is most relevant to the link source. We compute the cosine similarities between the source segment and all the target segments, to select the tar-

**Table 4** Effect of varying W2V dimensions, Pos : Neg = 1 : 10.

| Pos:Neg=1:10 | Precision | Recall | F1 |
|---|---|---|---|
| W2V(50) | 14.3% | 76.5% | 24.1% |
| W2V(100) | 14.8% | 71.7% | 24.5% |
| W2V(300) | 15.5% | 76.5% | 25.7% |

**Table 5** Pos : Neg = 1 : 1.

| Pos:Neg=1:1 | Precision | Recall | F1 |
|---|---|---|---|
| SimDist(smLDA) | 61.0% | 82.4% | 70.2% |
| SimDist(W2V) | 67.0% | 76.5% | 71.4% |
| SimDist(PMI) | 62.0% | 76.3% | 68.4% |
| SimDist(smLDA + W2V) | 51.7% | 88.2% | 65.2% |
| SimDist(smLDA + PMI) | 53.5% | 88.3% | 66.6% |
| SimDist(W2V + PMI) | 53.6% | 88.2% | 66.7% |
| SimDist(smLDA +W2V +PMI) | 72.0% | 76.8% | 74.3% |
| Random result | 50% | 50% | 50% |
| Baseline | 61% | 57% | 59% |

**Table 6** Pos : Neg = 1 : 10.

| Pos:Neg=1:10 | Precision | Recall | F1 |
|---|---|---|---|
| SimDist(smLDA) | 13.3% | 76.4% | 22.6% |
| SimDist(W2V) | 13.6% | 82.4% | 23.0% |
| SimDist(PMI) | 12.0% | 74.1% | 19.9% |
| SimDist(smLDA + W2V) | 13.4% | 76.5% | 22.8% |
| SimDist(LDA + PMI) | 12.1% | 76.5% | 21.0% |
| SimDist(W2V + PMI) | 12.1% | 78.4% | 20.9% |
| SimDist(smLDA+W2V +PMI) | 14.4% | 74.1% | 24.1% |
| Random result | 10% | 50% | 16.6% |
| Baseline | 12.8% | 46.2% | 20.0% |

**Table 7** Pos : Neg = 1 : 50.

| Pos:Neg=1:50 | Precision | Recall | F1 |
|---|---|---|---|
| SimDist(smLDA) | 2.3% | 48.9% | 4.5% |
| SimDist (W2V) | 2.6% | 57.8% | 5.0% |
| SimDist (PMI) | 2.2% | 66.6% | 4.3% |
| SimDist (smLDA + W2V ) | 2.5% | 75.6% | 4.9% |
| SimDist (smLDA + PMI) | 2.3% | 67.6% | 4.4% |
| SimDist (W2V + PMI) | 2.3% | 73.3% | 4.4% |
| SimDist(smLDA+W2V +PMI) | 2.6% | 82.8% | 5.1% |
| Random result | 2% | 50% | 3.8% |
| Baseline | 2.1% | 45% | 4.0% |

**Table 8** Pos : Neg = 1 : 100.

| Pos:Neg=1:100 | Precision | Recall | F1 |
|---|---|---|---|
| SimDist(smLDA) | 1.2% | 53.2% | 2.3% |
| SimDist(W2V) | 0.9% | 65.2% | 1.8% |
| SimDist(PMI) | 1.0% | 61.0% | 1.9% |
| SimDist(smLDA + W2V) | 1.0% | 53.9% | 2.0% |
| SimDist(smLDA +PMI) | 0.9% | 56.0% | 1.7% |
| SimDist(W2V + PMI) | 1.0% | 48.0% | 1.9% |
| SimDist(smLDA+W2V +PMI) | 1.2% | 56.1% | 2.4% |
| Random result | 1% | 50% | 1.9% |
| Baseline | 1.1% | 41% | 2.1% |

**Table 9** Predicting the most related segment.

| Average segment count=8 | Accuracy |
|---|---|
| SimDist(smLDA) | 21.3% |
| SimDist(W2V) | 32.0% |
| SimDist(PMI) | 40.0% |
| Random result | 12.5% |

get segment most similar to the source. After that, we calculate whether the most similar segment is an outlier in the entire similarity distribution. If the outlier condition is satisfied, we select the segment as our prediction result, otherwise we obtain no result and it will be counted as negative (unsuccessful). The result is shown in **Table 9**.

### 5.6 Discussions

The results in Tables 5–8 show that the similarity distribution method combining all of the theree similarity scores SimDist(LDA + W2V + PMI) is stably peforming best in predicting links pointing to segments, surpassing the baseline and the one and two-similarity feature sets. The correlation analysis results of Tables 2 and 3 show that our statistical features on similarity distributions are effecitive to determine whether segment-level links occur.

**Table 10** Removing under sampling, Pos : Neg = 1 : 10.

| Pos:Neg=1:10 SimDist(smLDA+W2V +PMI) | Precision | Recall | F1 |
|---|---|---|---|
| undersampling | 14.4% | 74.1% | 24.1% |
| without undersampling | 10.2% | 46.5% | 16.7% |

**Table 11** Corpora for training word vectors, Pos : Neg = 1 : 10.

| Pos:Neg=1:10 | Precision | Recall | F1 |
|---|---|---|---|
| SimDist(smLDA) | 13.0% | 70.5% | 22.0% |
| SimDist(W2V) - Google news | 13.0% | 68.2% | 21.8% |
| SimDist(W2V) -Wikipedia | 13.6% | 82.4% | 23.0% |
| Baseline | 12.8% | 46.2% | 20.0% |

**Table 12** With and without filtering, Pos : Neg = 1 : 10.

| Pos:Neg=1:10 SimDist(smLDA+W2V +PMI) | Precision | Recall | F1 |
|---|---|---|---|
| With filtering process | 16.6% | 64.3% | 26.4% |
| Without filtering process | 14.4% | 74.1% | 24.1% |

**Table 13** Effect of filtering thresholds, Pos : Neg = 1 : 10.

| Pos:Neg=1:10 SimDist(smLDA+W2V +PMI) | Precision | Recall | F1 |
|---|---|---|---|
| Without filtering process | 14.4% | 74.1% | 24.1% |
| Filter threshold <0.6 | 24.8% | 12.0% | 16.1% |
| Filter threshold <0.5 | 24.0% | 46.0% | 31.5% |
| Filter threshold <0.4 | 16.1% | 50.2% | 24.4% |
| Filter threshold <0.3 | 18.0% | 52.8% | 26.8% |

Our method works well when the positive and negative samples are balanced to 1 : 1. However, even though we already use the sampling method to balance the dataset, the influence of data imbalance is still strong. We find that when the ratio of postives and negatives is more than 1 : 10, precision falls down significantly. We still need to improve to deal with such imbalanced datasets.

Since our dataset is imbalanced, we can observe that the sampling process is quite important in our method. **Table 10** shows that if we remove the undersampling process, the F1 score decreases significantly.

**Table 11** shows the results of the W2V similarities when vectors are trained by Google news. The result of SimDist(W2V) (Google news) is close to smoothing LDA, and falling behind of SimDist(W2V) (Wikipedia). This can be due to the fact that our testing articles are also from Wikipedia. But in selecting the most related segment process, our combination method SimDist(smLDA + W2V + PMI) further improves over single-similarity SimDist(W2V).

We also compare the cases of with and without filtering in the model. **Tables 12** and **13** are the result of using SimDist(LDA +

**Table 14** Paired T-test.

| | Mean of the differences | p-value | Decision on null hypothesis |
|---|---|---|---|
| Proposed method and random result | 0.3096447 | 1.988e-10 | reject |
| Proposed method and baseline method | 0.106599 | 0.01925 | reject |
| baseline method and random result | 0.2030457 | 1.708e-05 | reject |

W2V + PMI). Table 12 shows that filtering contributes well in our task, improving precision by 2.1 percent, while recall is decreased by 10.3 percent. Overall, filtering improves F1 score by 2.9 percent.

Table 13 shows the impact of changing filtering thresholds, where the changes have significant differences to the results. If the filtering is strict (low thresholds), false positives increase, harming the recall. If the filtering is loose, false negatives increase, degrading the precision. In our experiment, the threshold of 0.5 gives the best result.

We applied the paired T-test to judge whether our proposed method is significantly better than the random method and the baseline method. The paired T-test checks whether the average difference in their performance over the data sets is significantly different from zero. We randomly select 200 samples with 100 positive samples and 100 negative samples as the T-test dataset. Then we use our method (our method was trained on the other samples, which also contain 100 positive samples and 100 negative samples), the baseline method and random method to predict the result of the T-test dataset. **Table 14** shows the test resutls, which reveals that our proposing method is significantly better than the baseline and random methods with p-value $< 0.05$.

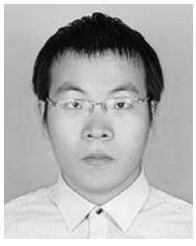
## 6. Conclusion and Future Work

In this paper, we proposed a text mining algorithm for determining whether links in Wikipedia articles should point to relevant segments, or article titles. We believe that our research is the first work for link detection on the segment level. Our approach is combining the LDA model with MLE by a nonlinear combination, word embeddings, and PMI. We also introduced statistical features on segment similarity distributions, to determine whether the most relevant segment is outstandingly similar to be a link target. Our rigorous evaluations show that our suit of the proposing methods are achieving the best performace. Our method performs well when the dataset is balanced. In future work, we plan to design features to furhter improve accuracies when the dataset is imbalanced. We also try to utilize category hierarchies when selecting the most related segment. In this paper, link targets are either the article title or segment titles, where segments are non-overlapping. We shall extend our appoarch to selecting segments which are nested into logical units, such as sections, subsections and paragraphs, where topical overlapping needs to be considered.

**Acknowledgments** The authors are grateful to the anonymous reviews who made a number of helpful and constructive comments. This work was in part supported by JSPS KAKENHI Grant Number JP16K00423.

### References

[1] Adafre, S.F. and de Rijke, M.: Discovering missing links in Wikipedia, *Proc. 3rd International Workshop on Link Discovery*, pp.90–97 (2005).
[2] Besnik, F., Katja, M. and Avishek, A.: Automated News Suggestions for Populating Wikipedia Entity Pages, *CIKM '15*, *Proc. 24th ACM International on Conference on Information and Knowledge Management*, pp.323–332 (2015).
[3] Blei, D.M. and Moreno, P.J.: Topic segmentation with an aspect hidden Markov model, *Proc. SIGIR* (2001).
[4] Blei, D.M., Ng, A.Y. and Jordan, M.J.: Latent Dirichlet allocation, *Journal of Machine Learning Research*, Vol.3, pp.993–1022 (2003).
[5] Bouma, G.: Normalized (pointwise) mutual information in collocation extraction, *Proc. GSCL*, pp.31–40 (2009).
[6] Breiman, L.: Random forests, *Machine Learning*, Vol.45, No.1, pp.5–32 (2001).
[7] Knoth, P., Novotny, J. and Zdrahal, Z.: Automatic generation of inter-passage links based on semantic similarity, *Proc. 23rd International Conference on Computational Linguistics*, pp.590–598 (2010).
[8] Lavrenko, V. and Croft, W.B.: Relevance-based language models, *SIGIR 2001*, pp.120–127 (2001).
[9] Liu, X. and Croft, W.B.: Cluster-based retrieval using language models, *Proc. 27th International ACM SIGIR Conf. Research and Development Information Retrieval*, pp.186–193 (2004).
[10] Le, Q.V. and Mikolov, T.: Distributed Representations of Sentences and Documents, *ICML*, Vol.14, pp.1188–1196 (2014).
[11] Mihalcea, R., Corley, C. and Strapparava, C: Corpus-based and knowledge-based measures of text semantic similarity, *AAAI*, Vol.6, pp.775–780 (2006).
[12] Milne, D., Witten, I.H.: An Effective, Low-Cost Measure of Semantic Relatedness Obtained from Wikipedia Links, *Proc. AAAI Workshop on Wikipedia and Artificial Intelligence: An Evolving Synergy*, Chicago, pp.25–30 (2008).
[13] Milne, D., Witten, I.H.: Learning to link with Wikipedia, *CIKM '08*, *Proc. 17th ACM Conference on Information and Knowledge Management*, pp.509–518 (2008).
[14] Rosenthal, G. and Rosenthal, J.A.: *Statistics and data interpretation for social work*, Springer Publishing Company (2011)
[15] Singhal, A., Buckley, C., Mitra, M. and Salton, G.: Pivoted document length normalization, *Proc. ACM SIGIR*, pp.21–29 (1996).
[16] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. and Dean, J.: Distributed Representations of Words and Phrases and their Compositionality, *NIPS '13*, pp.3111–3119 (2013).
[17] Mikolov, T., Chen, K., Corrado, G. and Dean, J.: Efficient Estimation of Word Representations in Vector Space, *ICLR '13*, *Proc. Workshop at International Conference on Learning Representations* (2013).
[18] Xing, W. and Croft, W.B.: LDA-Based Document Models for Ad-hoc Retrieval, *Proc. 29th ACM SIGIR Conf.*, pp.178–185 (2006).
[19] Wang, R. and Iwaihara, M.: Suggesting Specific Segments as Link Targets in Wikipedia, *Proc. Int. Conf. Asian Digital Libraries* (*ICADL2014*), LNCS, Vol.10075, pp.394–405, Springer (2016).
[20] Wang, R., Wu, J. and Iwaihara, M.: Finding co-occurring topics in Wikipedia article segments, *Proc. Int. Conf. Asian Digital Libraries* (*ICADL2014*), pp.252–259, Springer International Publishing (2014).
[21] Zhai, C. and Lafferty, J.: A study of smoothing methods for language models applied to ad hoc information retrieval, *Proc. 24th ACM SIGIR 2001*, pp.334–342 (2001).
[22] Zhang, J. and Kamps, J.: A content-based link detection approach using the vector space model, *International Workshop of the Initiative for the Evaluation of XML Retrieval*, pp.395–400 (2009).



**Renzhi Wang** received his B.S. degree in Computer Science and Technology from Sichuan University in 2013. He received his M.Eng. degree from Waseda University in 2014. He is now a Ph.D. candidate in Waseda University.

**Mizuho Iwaihara** received his B.Eng., M.Eng. and D.Eng. degrees all from Kyushu University, in 1988, 1990, and 1993 respectively. He was a research associate and then an associate professor in Kyushu University, from 1993 to 2001. From 2001 to 2009, he was an associate professor at Department of Social Informatics, Kyoto University. Since 2009, he is a professor at Graduate School of IPS, Waseda University. He is a member of IEICE, IPSJ, ACM and IEEE CS.

(Editor in Charge: *Yuu Suzuki*)