RDLL at CrossLink Anchor Extraction Considering Ambiguity in CLLD

Fuminori Kimura
Kinugasa Research Organization
Ritsumeikan University
Japan
fkimura@is.ritsumei.ac.jp

Kensuke Horita
Graduate School of Information and Engineering
Ritsumeikan University
Japan
is0038ep@ed.ritsumei.ac.jp

Yuuki Konishi
OMRON SOFTWARE Co.,Ltd
Japan

Hisato Harada
Graduate School of Information and Engineering
Ritsumeikan University
Japan
is0034hf@ed.ritsumei.ac.jp

Akira Maeda
College of Information and Engineering
Ritsumeikan University
Japan
amaeda@is.ritsumei.ac.jp

ABSTRACT

In this paper, we describe our work in NTCIR-10 on the task of cross-lingual link discovery (CLLD). Our proposed method is focused mainly on two aspects in order to accomplish this task: how to find important anchors from an original article in order to crosslink and how to find the correct links to articles in the target language for the original articles. The system first uses online data collected from Japanese Wikipedia articles in order to build a basic crosslink database. These data will be applied in order to identify the anchors and find out the relevant corresponding English articles.

We carried out this task in three steps. First, we parsed the Japanese articles and extracted the candidate anchors. Second, we ranked anchors on the basis of the weights of their importance. Third, we determined the correct English articles for each anchor.

We marked LMAP 0.151 with manual assessment.

Keywords

Top consecutive nouns cohesion, machine translation, Japanese to English

1. INTRODUCTION

Cross-lingual link discovery (CLLD) is a research topic in which potential links between documents among different languages are discovered automatically.

Wikipedia is mentioned as a language resource that can be used for CLLD. Wikipedia is a multilingual online encyclopedia that contains a large number of articles. It has a wide range of hyperlinks between documents of the same language. However, in different languages, such links are rare. Therefore, it is a difficult problem when users want to obtain information and knowledge from different language resources.

Regarding CLLD in the NTCIR, there is a task to find anchors and the corresponding target articles for either Japanese, Chinese, or Korean. In addition, there is work on discovering target articles and extracting the corresponding anchors from any language to English, Japanese, Chinese, and Korean. In this paper, we tackle the Japanese to English subtask.

2. PROPOSED METHOD

The proposed method consists of the following two steps.

- Anchor extraction
- Related English article extraction

For anchor extraction, first, the proposed method extracts candidate anchor texts from an original Japanese document and ranks them. Second, the proposed method selects higher ranked candidate anchor texts as anchors.

For related English article extraction, the proposed method detects English documents related to the extracted anchors. First, the method translates the anchors into English and detects English articles that contain the translated anchor in the beginning of the title. Second, the proposed method ranks detected English articles on the basis of cosine similarity association with the original article that was translated into English. Higher ranked English articles are treated as ones related to the anchor.

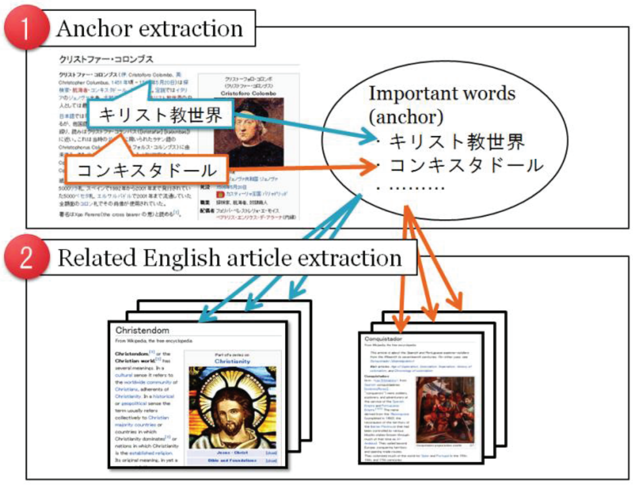


Figure 1. Anchor extraction

2.1 Anchor extraction

The related English article extraction method consists of the following three steps.

1. Extract candidate anchor.
2. Calculate the importance of the anchor.
3. Rank the anchor.

Figure 2 shows the overall step flow of the proposed method.

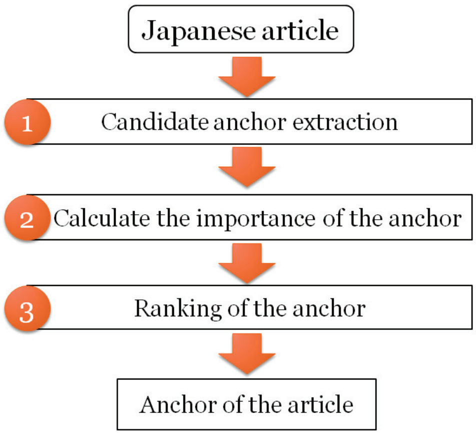

Japanese article
1
Candidate anchor extraction
2
Calculate the importance of the anchor
3
Ranking of the anchor
Anchor of the article

Figure 2. Overall step flow of proposed method

2.1.1 *Candidate anchor extraction*

We conduct a morphological analysis by using the morphological analyzer MeCab[1] after presteps such as HTML tag and newline elimination. We treat nouns as candidate anchor words because about 80% titles of Japanese Wikipedia articles consist of only nouns, (titles with only nouns: 1,074,764/total number of Japanese Wikipedia article titles: 1,342,099). Some of them consist of some nouns and particles such as "アムステルダムの防塞線" and "日本の離島架橋."

・Top consecutive noun cohesion

We continuously connect nouns from the top to several of them and treat them as one compound word. We call this connecting "top consecutive noun cohesion (TCNC)." When consecutive nouns appear, TCNC adopts all possible binding patterns. In other words, TCNC obtains several compound words that are the same in number as the number of consecutive nouns. When three consecutive nouns appear, TCNC obtains three compound words: the top noun of the consecutive nouns, the top and second noun, and all nouns (Figure 3). All of these obtained compound words are treated as candidate anchors. If we use the word N-gram method, much noise is derived. TCNC, however, can reduce noise.

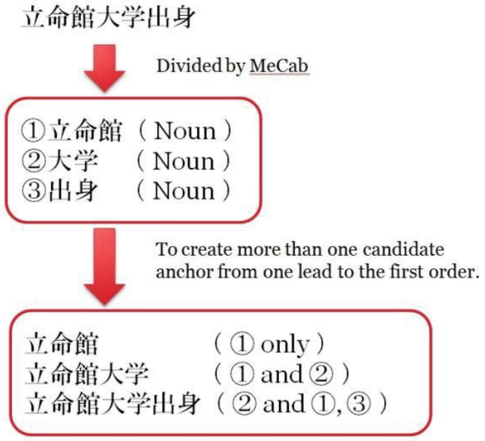


Figure 3. Example of top consecutive noun cohesion (TCNC)

2.1.2 *Importance of anchor calculation*

We must select anchors from candidate anchors extracted by TCNC because there are times when more than 250 candidate anchors are obtained. Therefore, we must rank the candidate anchors. We calculate the weights of candidate anchors with the following four methods.

2.1.2.1 *TF-IDF*

TF-IDF[1] is calculated by multiplying the term frequency and inverse document frequency. TF-IDF for anchor *t* in article *d* is calculated with the following formula.

$$tf\text{-}idf_{t,d} = tf_{t,d} \times idf_t$$

, where $tf_{t,d}$ is the term frequency in article *d* and idf_t is the inverse document frequency for anchor *t*. The value of TF-IDF is higher when anchors occur frequently in article *d* and occur in few articles. It was also used in some studies in NTCIR-9[2].

2.1.2.2 *Okapi BM25*

Okapi BM25 takes average document length into account, which is not taken into account in TF-IDF. Okapi BM25 was used in studies by Kim[3] and Tang[4] in NTCIR-9. The formula is as follows.

$$BM25(t,d) = \ln\left(\frac{N - df(t) + 0.5}{df(t) + 0.5} \cdot \frac{(k+1) \cdot tf_{t,d}}{k\left((1-b) + b\frac{|d|}{avegdl} \right)} \right)$$

where *k* and *b* are parameters (we set *k* = 2.0 and *b* = 0.75), $tf_{t,\,d}$ is the number of occurrences of anchor *t* in article *d,* $|D|$ is the document length of article *d*, and *avegdl* is the average document length of all articles in the Japanese Wikipedia.

2.1.2.3 *Dice coefficient*

The dice coefficient considers how often two words co-occur. In this paper, the value of the dice coefficient is higher the more the candidate anchor *q* in the original article *j* and title of the original article T_i appear in the original article. The purpose of using the dice coefficient is to give a higher rank to terms that have a relationship with the title of the original article. The dice coefficient *S (Ti, q)* is calculated with the following formula.

[1] http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html

$$S(T_{i,}q) = \frac{2\sum_{j=1}^{M} w_{ij}w_{qj}}{\sum_{j=1}^{M} w_{ij} + \sum_{j=1}^{M} w_{qj}}$$

$$W_{ij} = \begin{cases} 1: & \text{title } T_i \text{ appears in Japanese title of article } j. \\ 0: & \text{do not appear.} \end{cases}$$

$$W_{qj} = \begin{cases} 1: & \text{anchor } q \text{ appears in Japanese title of article } j. \\ 0: & \text{do not appear.} \end{cases}$$

Where M is the number of all Japanese Wikipedia articles and q is the candidate anchor in the original article. T_i is title of original article i.

2.1.2.4 TF-Dice coefficient

The dice coefficient does not consider the frequency of the anchor in the article, so it is difficult to obtain the relationship between the anchor and title of an original article. Therefore, we propose a weight calculation method that multiplies the term frequency and dice coefficient in order to find candidate anchors that deeply relate to the title and frequently appear in the original article.

The TF-Dice coefficient is calculated with the following formula. In this coefficient, the term frequency tf is normalized by the average frequency of all candidate anchors in the article.

$$TF * Dice(d_i, q) = tf_{d_i,q} \times S(T_i, q)$$

where d_i is an original article, q is a candidate anchor in the original article, and $S\ (T_i,\ q)$ is the dice coefficient values of the candidate anchor and the title of the original article.

2.2 Related English article extraction

The related English article extraction method consists of the following three steps.

1. Translate anchors with multiple methods.

2. Detect related English articles for translated candidate anchors.

3. Rank these by using the cosine similarity[5] between a detected English article and the translated original one.

Figure 4 shows the step flow of the proposed method.

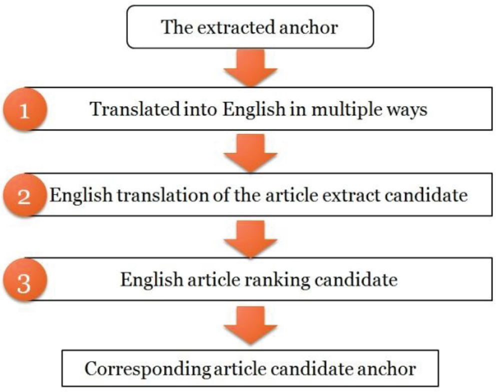

The extracted anchor
1
Translated into English in multiple ways
2
English translation of the article extract candidate
3
English article ranking candidate
Corresponding article candidate anchor

Figure 4. Steps flow of the proposed method.

2.2.1 Anchor translation with multiple methods

For translation, we use three methods: Microsoft Translator as machine translation, the Japanese-English dictionary EDICT, and the second language link DB, which is a database that manually stores Wikipedia links between the same articles written in different languages. In this paper, we compare the case of using all three translation methods with the case of using two methods without the second language link.

Translating with only one method may cause mistranslation or no translation to be found. Therefore, we use these three translation methods in order to prevent these problems.

Figure 5 shows an example of the proposed translation method.

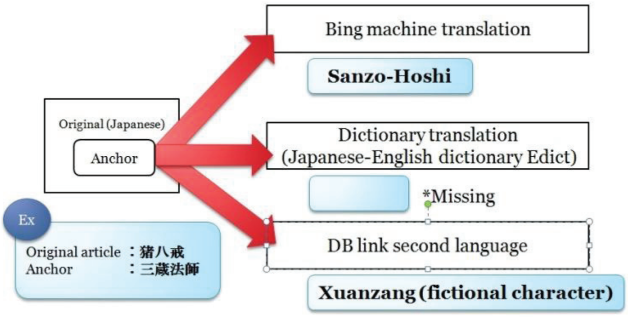


Figure 5. Example of proposed translation method

2.2.2 Related English article detection

In these steps, we detect related English articles for each obtained translation of an anchor. We conduct prefix matching for each obtained translation of an anchor in order to detect related English articles. If we conduct an exact match, much misdetection are caused by ambiguity of representation. Prefix matching can defuse this problem. Figure 6 shows an example of prefix match detection.

However, this approach may extract unnecessary articles. Moreover, we also tried partial matching. Partial matching detected more noisy articles than did prefix matching, so we did not adopt partial matching. To reduce the number of these detected noisy articles, we use the ranking of detected English articles mentioned in the following section.

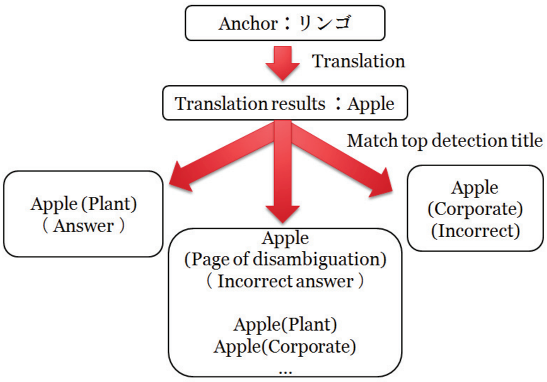
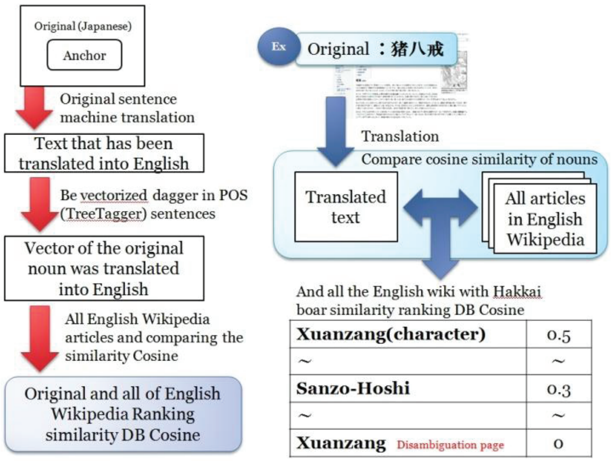


Figure 6. Examples of prefix match detection (anchor: apple).

2.2.3 Detected English article ranking

The detected English articles for the candidate anchors also contain unnecessary articles as mentioned above. Therefore, we rank these articles by using all of the text in the original article of the anchor in order to remove unnecessary ones. Figure 7 shows the procedure of ranking with the proposed method. First, the method translates the original article by using Microsoft Translator. Second, it extracts translated nouns by using TreeTagger as part-of-speech tagging. It also extracts all of the nouns in the detected English articles in the same way. These extracted nouns are used for creating term vectors for each article. Third, the method calculates the cosine similarity between the English article and the original article by using each term vector. Fourth, it creates a database to store the results of this calculation.

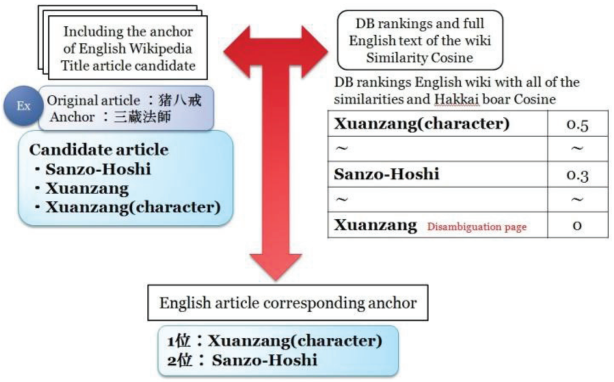

Original (Japanese)
Anchor
Original sentence machine translation
Text that has been translated into English
Be vectorized dagger in POS (TreeTagger) sentences
Vector of the original noun was translated into English
All English Wikipedia articles and comparing the similarity Cosine
Original and all of English Wikipedia Ranking similarity DB Cosine
Ex
Original：猪八戒
Translation
Compare cosine similarity of nouns
Translated text
All articles in English Wikipedia
And all the English wiki with Hakkai boar similarity ranking DB Cosine

Xuanzang(character)	0.5
~	~
Sanzo-Hoshi	0.3
~	~
Xuanzang Disambiguation page	0

Figure 7. Processing flow for creating a ranking of the proposed method

Then, we are able to rank detected English articles. The criteria of this ranking are in comparison with the English version of Wikipedia articles. Then, an English article detected by an anchor with lower importance to the original article is calculated with low similarity. Therefore, this method does not adapt to threshold cosine similarity in order to remove useless articles. However, the proposed method remove articles that have a similarity value of 0 from related English articles because there is no relationship in such articles.

Figure 8 shows an example of extracting a corresponding English article by using the proposed method. In this case, three candidate English articles were found. One of the three candidates, "Xuanzang," was removed from the candidates because its cosine similarity was 0.


Including the anchor of English Wikipedia Title article candidate
Ex
Original article ：猪八戒
Anchor ：三蔵法師
Candidate article
・Sanzo-Hoshi
・Xuanzang
・Xuanzang(character)
DB rankings and full English text of the wiki Similarity Cosine
DB rankings English wiki with all of the similarities and Hakkai boar Cosine
English article corresponding anchor
1位：Xuanzang(character)
2位：Sanzo-Hoshi

Xuanzang(character)	0.5
~	~
Sanzo-Hoshi	0.3
~	~
Xuanzang Disambiguation page	0

Figure 8. Example of extracting corresponding English articles for anchor

3. EXPERIMENTS

In this experiment, we used 25 articles that were prepared for the NTCIR-10 Crosslink task. We tackled the Japanese to English CLLD subtask. The Wikipedia corpus used in this experiment was also distributed by NTCIR. We evaluated our submitted runs by using LMAP R-Prec with two answer sets, "Wikipedia ground truth" and "Manual Assessment." We submitted five runs. All of them used the same anchor extraction method mentioned in section 2. We used four ranking methods: TF-IDF (tfidf), Okapi BM25 (okapiBM25), Dice coefficient (dice), and TF-Dice coefficient (tfdice). In addition, our fifth run used the TF-Dice coefficient ranking method and the second language link. We submitted five results. One of them was obtained by using the second language link.

The other results were not obtained with the second language link. We compared them to examine the influence of using the second language link.

Table 1. F2F evaluation with manual assessment results: LMAP, R-Prec

Japanese-to-English		
Run-ID	**LMAP**	**R-Prec**
RDLL_A2F_J2E_05_tfdiceLL	0.137	0.160
RDLL_A2F_J2E_03_dice	0.060	0.083
RDLL_A2F_J2E_04_tfdice	0.060	0.091
RDLL_A2F_J2E_02_okapiBM25	0.052	0.080
RDLL_A2F_J2E_01_tfidf	0.046	0.089

Table 2. A2F evaluation with manual assessment results: LMAP, R-Prec

Japanese-to-English		
Run-ID	**LMAP**	**R-Prec**
RDLL_A2F_J2E_05_tfdiceLL	0.151	0.105
RDLL_A2F_J2E_03_dice	0.029	0.031
RDLL_A2F_J2E_04_tfdice	0.029	0.028
RDLL_A2F_J2E_02_okapiBM25	0.027	0.030
RDLL_A2F_J2E_01_tfidf	0.023	0.024

4. CONSIDERATION

The results of the experiment discussed in the previous section showed that our proposed method showed low precision. There were three causes.

First, many noise anchors appeared when we extracted anchors.

Top consecutive noun cohesion created many candidate anchors because TCNC created many consecutive nouns from one compound word. We explain an example in the case of "調理師養成施設."

The word "調理師養成施設" was divided into four nouns, "調理," "師," "養成," and "施設," by a morphological analyzer. Using top consecutive noun cohesion, we got four candidate anchors: "調理," "調理師," "調理師養成," and "調理師養成施設."

However, the correct anchor was only "調理師," so we also acquired three noise anchors.

Second is that proper nouns were not successfully translated.

Dictionaries and machine translation obtain wrong translations for proper nouns if they do not have suitable translations. We searched unrelated English articles because we obtained the wrong translation. Therefore, we could not get the appropriate number of English articles.

Third is that we extracted too many anchors, much more than necessary. We always extracted 250 anchors, and we also found the maximum number of target articles for each anchor. However, the average number of correct answers of the anchors for one article was 25. We extracted a lot of inappropriate anchors, so the precision was low.

We submitted one run that used a second language link. Comparing the results between the run that used the second language link and the runs without it, the run that used it showed good accuracy. The second language link is a manually made link, so appropriate links can be obtained by using it. However, CLLD's purpose is to find a link automatically. Considering this, we feel that we should not use the second language link in this task.

5. CONCLUSION

We tackled the Japanese to English CLLD subtask in NTCIR-10.

First, we extracted nouns by using a morphological analyzer. Second, we connected continuously ordered nouns and selected candidate anchors for them. Third, we ranked them by using four weighting methods: Dice coefficient, TF-IDF, Okapi BM25, and TF-Dice coefficient.

To find target articles for anchors, we translated anchors by using machine translation and a dictionary. Then, we discovered English articles that contained the translated anchors at the beginning of their title. We ranked these English articles on the basis of cosine similarity between the translated articles and the original ones.

Our result achieved LMAP 0.151 with manual assessment.

6. FUTURE WORK

To improve the performance of CLLD, there are two points to consider.

First is that we must select fewer candidate anchors as the anchor sets. It is necessary to set a threshold of rank when we select the candidate anchors in order to reduce noise.

Second is that articles must be focused on fewer targets for each candidate anchor by categorizing the articles. We think that we will be able to get good results when we use category information in order to find correlation between the genre of an anchor and the target article.

7. REFERENCES

[1] Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze., *Introduction to Information Retrieval,* Cambridge University Press New York, NY, USA ©2008

[2] Pham Huy Anh, Takashi Yukawa. " Using Concept base and Wikipedia for Cross-Lingual Link Discovery" Proceedings of NTCIR-9 Workshop Meeting, December 6-9, 2011, Tokyo, Japan, p.464-468

[3] Jungi Kim, Iryna Gurevych. "UKP at CrossLink: Anchor Text Translation for Cross-lingual Link Discovery". Proceedings of NTCIR-9 Workshop Meeting, December 6-9, 2011, Tokyo, Japan, p.487-494

[4] Ling-Xiang Tang, Daniel Cavanagh, Andrew Trotman, Shlomo Geva, Yue Xu and Laurianne Sitbon. " Automated Cross-lingual Link Discovery in Wikipedia" Proceedings of NTCIR-9 Workshop Meeting, December 6-9, 2011, Tokyo, Japan, p.512-519

[5] Salton, G., Wong, A. and Yang, C.S.:" A vector space model for automatic indexing", *Communications of the ACM,* Volume 18 Issue 11 ,pp. 613-620