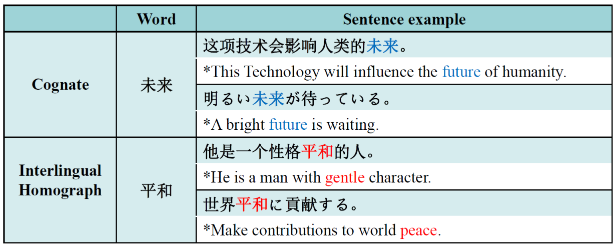
Automatic Interlingual Homograph Recognition with Context Features

Yi Han Ryohei Sasano Koichi Takeda
Graduate School of Informatics, Nagoya University
han.yi.u2@s.mail.nagoya-u.ac.jp {sasano,takedasu}@i.nagoya-u.ac.jp

Abstract

In this paper, we propose a method that incorporates cross-lingual word embedding similarity and degree of co-occurrence in parallel sentences to automate the process of recognizing interlingual homographs. We conduct experiments with multiple word embedding models and different co-occurrence metrics in both Chinese-Japanese and English-Dutch language pairs. Experimental results demonstrate that our proposed method is able to produce accurate and consistent predictions across languages.



	Word	Sentence example
Cognate	未来	这项技术会影响人类的未来。
		*This Technology will influence the future of humanity.
		明るい未来が待っている。
		*A bright future is waiting.
Interlingual Homograph	平和	他是一个性格平和的人。
		*He is a man with gentle character.
		世界平和に貢献する。
		*Make contributions to world peace.

Figure 1 An Example of *cognate* and *interlingual homograph* in Chinese-Japaneses. * denotes the English translation of sentence examples.

1 Introduction

When learning a foreign language, we often come across words in different languages sharing identical orthographic forms. This is commonly seen in languages with similar writing systems. Such form-identical words with similar semantic meaning across languages are called *cognates*, while those with different semantic meanings are called *interlingual homographs* [1, 2]. For instance, the Dutch word “angle” means “sting”, as opposed to its form-identical word in English. It is not unique for phonographic writing systems. In languages sharing logographic writing systems [3] such as Chinese and Japanese, we can also see interlingual homograph examples like the word “平和”, which means “gentle” in Chinese whereas “peace” or “harmony” in Japanese. For second language learners, interlingual homographs can cause learning difficulties as second language acquisition often comprises relating a foreign language to ones’ native language [4, 5]. Likewise, interlingual homographs can also pose challenges to natural language processing (NLP) tasks. However, unlike the in-depth investigation of monolingual homographs in tasks like word sense disambiguation and machine translation [6], less attention has been paid to the interlingual homographs. Dominant approaches in psychology and language education introduce *interlingual homograph recognition* to alleviate the semantic ambiguity. Despite the merits, massive manual annotation works by bilinguals are quite costly [2].

We contribute to this question by automating the process of recognizing interlingual homographs, which can be executed efficiently allowing the absence of linguistic knowledge. Specifically, we manage to involve context features by adopting word embedding similarity and degree of co-occurrence to perform recognition. We conduct experiment on two pairs of languages that are etymologically distant from each other, namely, Chinese-Japanese and English-Dutch to exploit the feasibility of our proposed method. Experimental results on both pairs prove the effectiveness of the proposed method.

2 Methodology

In this work, we tackle the interlingual homograph recognition across languages. As we cannot find clues from their appearance, we need to make predictions based on their context information. Motivated by this, we formulate our criterion with two important components: **word embedding similarity** and **degree of co-occurrence in parallel sentences**.

2.1 Word Embedding Similarity

The distribution hypothesis suggests that the more semantically similar two words are, the more they occur in similar linguistic contexts [7]. An intuitive way to decide whether a pair of words are cognates or interlingual homographs, is to exploit the word embedding similarity. Generally, there are two types of word embedding, namely the static word embedding, such as Glove[8] and fastText[9], and the dynamic/contextual embedding, such as ELMo[10] and BERT[11].

To compute the similarity of word embeddings, we have to assure that they are in the same vector space. As the words in our setting are from two different languages, we need to introduce an operation: cross-lingual mapping. Cross-lingual mapping aligns independently trained monolingual word embeddings into a single shared space. Existing approaches usually use a bilingual dictionary as supervision signals. Formally, let L_1 and L_2 represent a pair of languages, and let u and v represent words from L_1 and L_2. Given a bilingual dictionary $\mathrm{Z} = \{(u_n, v_n)\}_{n=1}^{N}$, we obtain representations of each word: $\mathbf{u}_1, \ldots, \mathbf{u}_N, \mathbf{v}_1, \ldots, \mathbf{v}_N$, where $\mathbf{u}_n, \mathbf{v}_n \in \mathbb{R}^d$. Mikolov [12] learns the optimal projection matrix W by minimizing:

$$W^* = \underset{W \in \mathbb{R}^{d\times d}}{\arg\min} ||W\mathbf{A} - \mathbf{B}||_F, \tag{1}$$

where $\mathbf{A}$ and $\mathbf{B}$ are two matrix containing all embeddings of words in $\mathbf{Z}$, namely $\mathbf{A} = [\mathbf{u}_1, \ldots, \mathbf{u}_N]$, $\mathbf{B} = [\mathbf{v}_1, \ldots, \mathbf{v}_N]$, $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{d\times N}$. Xing [13] restrict W to be orthogonal, turning Equation 1 into the Procrustes problem [14, 15] by:

$$W^* = UV^T,\ U\Sigma V^T = \mathrm{SVD}(\mathbf{B}\mathbf{A}^T), \tag{2}$$

where $\mathrm{SVD}(\cdot)$ is the singular value decomposition.

We generally follow Xing's method to get a projection matrix, except that we obtain W with parallel sentences instead of bilingual dictionary. Let $\mathrm{D} = \{(x_n, y_n)\}_{n=1}^{N}$ represent a parallel corpus of L_1 and L_2. For each sentence $x_n = w_1^1, \ldots, w_I^1$, $y_n = w_1^2, \ldots, w_{I'}^2$ we obtain sentence embedding by averaging the word embeddings:

$$\mathbf{x}_n = \frac{1}{I}\sum_{i=1}^{I} \mathbf{w}_i^1, \quad \mathbf{y}_n = \frac{1}{I'}\sum_{i=1}^{I'} \mathbf{w}_i^2. \tag{3}$$

Thus in our setting, $\mathbf{A} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$, $\mathbf{B} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]$. We then perform Equation 2 to get W.

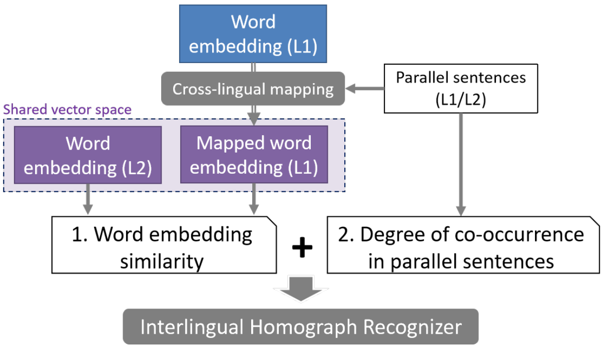

Word embedding (L1)
Cross-lingual mapping
Parallel sentences (L1/L2)
Shared vector space
Word embedding (L2)
Mapped word embedding (L1)
1. Word embedding similarity
+
2. Degree of co-occurrence in parallel sentences
Interlingual Homograph Recognizer

Figure 2 Overview of our proposed method

For a pair of form-identical words (z^1, z^2), $z^1 \in L_1, z^2 \in L_2$, we first obtain word embeddings in corresponding languages $(\mathbf{z}^1, \mathbf{z}^2)$, then compute the cosine similarity by:

$$s = \cos(W\mathbf{z}^1, \mathbf{z}^2). \tag{4}$$

With contextual word embedding models [16], we obtain the embedding of a single word z in an alternative way: select a set of sentences containing z; compute embeddings of z in every sentence; get the average of all embeddings. For Chinese and Japanese, we take an average embedding of all characters comprising word z.

2.2 Degree of Co-occurrence in Parallel Sentences

Degree of co-occurrence reveals how often two words occur in similar linguistic contexts. We develop this intuition further and assume that a pair of interlingual homographs are less likely to appear in parallel sentences. We introduce two methods to measure degree of co-occurrence: pointwise mutual information (PMI) and Jaccard similarity coefficient. Given a parallel corpus $\mathrm{D} = \{(x_n, y_n)\}_{n=1}^{N}$, the PMI of a pair of form-identical words (z^1, z^2) is:

$$\mathrm{PMI}(z^1, z^2) = \log \frac{P_{\mathrm{D}}(z^1, z^2)}{P_{\mathrm{D}}(z^1)P_{\mathrm{D}}(z^2)}, \tag{5}$$

where $P_{\mathrm{D}}(z^1, z^2)$ represents the probability of $z^1 \in \{x_n\}$ meanwhile $z^2 \in \{y_n\}$. $P_{\mathrm{D}}(z^1)$ denotes the probability of $z^1 \in \{x_n\}$ and $P_{\mathrm{D}}(z^2)$ denotes the probability of $z^2 \in \{y_n\}$

Jaccard similarity coefficient is:

$$\mathrm{Jaccard}(z^1, z^2) = \frac{C(z^1, z^2)}{C(z^1) + C(z^2) - C(z^1, z^2)}, \tag{6}$$

where $C(z^1, z^2)$ denotes counts of $z^1 \in \{x_n\}$ meanwhile $z^2 \in \{y_n\}$. $C(z^1)$ represents counts of $z^1 \in \{x_n\}$ and $C(z^2)$ represents counts of $z^2 \in \{y_n\}$.

Table 1 Statistics of cognates and homograph datasets

Language Pair	Cognates	Homographs
Chinese-Japanese	173	173
English-Dutch	52	52

2.3 Proposed Method

Figure 2 illustrates the framework of our proposed method. Given a pair of form-identical words, we first obtain word embeddings with embedding models and align them to a shared space with a linear mapping estimated with a parallel corpus. Then we get a similarity score by computing the cosine similarity of embeddings across languages. We also extract degree of co-occurrence from parallel sentences. Finally, we compute the z-score of the above two scores and fuse them by addition calculation in pairs. We make decisions whether a pair of words are interlingual homographs or cognates by the fusion scores.

3 Experiment

3.1 Dataset

We conduct experiments on two languages pairs: Chinese-Japanese and English-Dutch. Each language pair involves two datasets, i.e., cognates and interlingual homographs. For English-Dutch language pair, we directly take advantage of an existing database containing English-Dutch cognates and interlingual homographs [17]. For Chinese-Japanese, we refer to a Chinese-Japanese homograph dictionary [18] to derive interlingual homographs. Note that, as our work focuses on interlingual homographs with explicit disparity, we exclude the form-identical words with partially overlapped meanings. We then refer to Chinese-Japanese dictionary [19] to extract identical cognates. Table 1 lists the numbers of cognate pairs and homograph pairs for each of the Chinese-Japanese and English-Dutch datasets. We use Wikipedia dataset[1)] for contextual word embedding extraction. We extract 1 million sentence pairs respectively from Chinese-Japanese and English-Dutch WikiMatrix [20] as parallel sentences.

3.2 Word Embedding Models

We employ fastText [9], BERT, and multilingual BERT (mBERT) [11], representing static word embedding model, monolingual contextual embedding model, and multilingual contextual embedding model, respectively.

Table 2 Pretrained BERT and mBERT Models used in our experiment

Language	Model
Chinese	bert-base-chinese
Japanese	bert-base-japanese-char
English	bert-base-cased
Dutch	bert-base-dutch-cased
Multilingual	multi_cased_L12_H_768_A_12

For fastText, Facebook has published pretrained 300-dimensional word embeddings[2)] for 157 languages from which we extract embeddings for our target languages. For BERT and mBERT, we use 12-layers transformer encoder pretrained by huggingface with masked language modeling[3)]. The contextual word embeddings produced by these models are all 768-dimensional.

It's worth noting that because in Chinese BERT and mBERT, tokens are processed in the form of characters, so we also choose to use Japanese BERT with character-based tokenization instead of commonly used word-base model for coordination and fair comparison. The models used in this work are summarized in Table 2:

3.3 Experimental Settings

As described in Section 2, we explore the proposed method in three groups of experiments, including the word embedding similarity (EmbSim), degree of co-occurrence (CoR), and the fusion of these two, represented as follows.

- **EmbSim**: fastText, BERT, mBERT
- **CoR**: PMI, Jaccard
- **Fusion**: EmbSim+Jaccard

Particularly, we extract contextual embedding of words in our dataset, described in Section 3.1 by the following procedures. (1) For each word, we search the Wikipedia dataset by the word and select 300 sentences. (2) Derive embedding vectors of this word by putting each selected sentence into a pre-trained contextual embedding language model. (3) Take an average of derived vectors as the integrated representation, i.e., contextual embedding of this word.

Note that, to testify our method in a most general scenario, we conform to the original settings of all above

[1)] https://dumps.wikimedia.org

[2)] https://github.com/facebookresearch/fastText

[3)] https://huggingface.co

Table 3 Interlingual homograph recognition performance in terms of F1 score and Accuracy.

Group	System	Chinese-Japanese		English-Dutch	
		F1	Acc.	F1	Acc.
EmbSim	fastText	0.861	0.867	0.860	0.865
	BERT	0.759	0.817	0.757	0.798
	mBERT	0.468	0.488	0.793	0.760
CoR	PMI	0.486	0.509	0.603	0.596
	Jaccard	0.800	0.817	0.783	0.798
Fusion	fastText+Jaccard	**0.928**	**0.934**	**0.869**	**0.875**
	BERT+Jaccard	0.847	0.845	0.772	0.779
	mBERT+Jaccard	0.817	0.800	0.830	0.826

Table 4 A misleading example with contradictory between co-occurrence statistics and PMI scores.

Word	Chinese	Japanese	Co-occurrence	PMI
委員	6433	6851	4278	4.58
一味	25	105	1	5.94

mentioned pre-trained language models, without parameter tuning.

3.4 Experimental Results

Table 3 shows the results of all experiments. We report F1 score and accuracy for the assessment of the interlingual recognition capability of our method.

EmbSim fastText demonstrates superior performance compared with the other two contextual word embedding models. We attribute contextual word embedding models' inferior performance to the absence of fine-tuning process and the challenge brought by their dynamic property. If we compare monolingual BERT and mBERT, the results vary by languages. Specifically, English-Dutch pair benefits more from mBERT, while Chinese-Japanese pair benefits much more from monolingual BERT.

CoR Jaccard much outperforms PMI in both language pairs. We blame PMI's poor performance on the unbalanced numbers of words appearing in WikiMatrix data. Table 4 shows an example to demonstrate this problem, where "委員" is a cognate, which means "committee member" in both Chinese and Japanese, and "一味" is an interlingual homograph, which means "blindly" in Chinese while "conspirators" in Japanese. From the statistics, we can easily draw a conclusion that "一味" is more likely to be an interlingual homograph than "委員", however, the PMI score shows the opposite result.

Fusion We choose Jaccard to corporate each method in the EmbSim group. As illustrated, all three methods gain improvements with Jaccard, among which, the fastText+Jaccard won the best place among all set-ups. This shows that semantic information contained in word embeddings sometimes is not enough, it is advisable to supplement it with extra knowledge.

4 Conclusion and Future Work

In this work, we integrate word embedding similarity into degree of co-occurrence in parallel sentences to automatically recognize interlingual homographs in different languages. We perform it on two language pairs, i.e., Chinese-Japanese and English-Dutch, and the experimental results exhibit the effectiveness of our method. fastText shows better performance than contextualized embeddings and by the supplement of Jaccard information, the performance can be further improved in both language pairs.

A gap can be observed between the performance of Chinese-Japanese with mBERT and the other performances. There is a possible reason that too many Chinese and Japanese identical tokens are shared when pretraining multilingual BERT and the oversharing would bring a negative effect to the multilingual pretrained language model. We will test this assumption and try to mitigate this problem in future work.

References

[1] Ton Dijkstra, Jonathan Grainger, and Walter JB Van Heuven. Recognition of cognates and interlingual homographs: The neglected role of phonology. **Journal of Memory and Language**, Vol. 41, pp. 496–518, 1999.

[2] Kristin Lemhöfer and Ton Dijkstra. Recognizing cognates and interlingual homographs: Effects of code similarity in language-specific and generalized lexical decision. **Memory & Cognition**, Vol. 32, pp. 533–550, 2004.

[3] Richard Sproat and Alexander Gutkin. The taxonomy of writing systems: How to measure how logographic a system is. **Computational Linguistics**, pp. 477–528, 2021.

[4] Kexin Xiong and Katsuo Tamaoka. Investigation on the correspondence between the part-speech characteristics of

japanese-chinese homomorphic two-character kanji words (In Japanese). 2014.

[5] Robert W. Long and Yui Hatcho. The first language’s impact on l2: Investigating intralingual and interlingual errors. **English Language Teaching**, 2018.

[6] Frederick Liu, Han Lu, and Graham Neubig. Handling homographs in neural machine translation. In **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)**, pp. 1336–1345, 2018.

[7] Zellig S. Harris. Distributional structure. 1981.

[8] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global vectors for word representation. In **Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 1532–1543, 2014.

[9] Piotr Bojanowski, Édouard Grave, Armand Joulin, and Tomáš Mikolov. Enriching word vectors with subword information. **Transactions of the Association for Computational Linguistics (TACL)**, Vol. 5, pp. 135–146, 2017.

[10] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In **Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (NAACL-HLT)**, pp. 2227–2237, 2018.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)**, pp. 4171–4186, 2019.

[12] Tomás Mikolov, Quoc V. Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. **arXiv preprint arXiv:1309.4168**, 2013.

[13] Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. Normalized word embedding and orthogonal transform for bilingual word translation. In **The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)**, pp. 1006–1011, 2015.

[14] Zirui Wang, Jiateng Xie, Ruochen Xu, Yiming Yang, Graham Neubig, and Jaime G. Carbonell. Cross-lingual alignment vs joint training: A comparative study and A simple unified framework. In **8th International Conference on Learning Representations (ICLR)**, 2020.

[15] Guillaume Lample, Alexis Conneau, Marc’Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. In **6th International Conference on Learning Representations (ICLR)**, 2018.

[16] Hanan Aldarmaki and Mona Diab. Context-aware cross-lingual mapping. In **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)**, pp. 3906–3911, 2019.

[17] Eva D Poort and Jennifer M. Rodd. A database of dutch–english cognates, interlingual homographs and translation equivalents. **Journal of Cognition**, 2019.

[18] Changfu Xu Yongquan Wang, Shinjiro Koizumi. **Chinese Japanese Interlingual Homograph Dictionary**. Commercial Press (In Chinese), 2009.

[19] Ltd. Obunsha Co. **Dual solution to learn Japanese and Chinese dictionaries: Standard Mandarin Dictionary**. Foreign language education research publisher (In Chinese), 2005.

[20] Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia. In **Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)**, pp. 1351–1361, 2021.