Discriminative Models for Automatic Acquisition of Translation Equivalences

Chun-Xiang Zhang, Sheng Li, and Tie-Jun Zhao

Abstract: Translation equivalence is very important for bilingual lexicography, machine translation system and cross-lingual information retrieval. Extraction of equivalences from bilingual sentence pairs belongs to data mining problem. In this paper, discriminative learning methods are employed to filter translation equivalences. Discriminative features including translation literality, phrase alignment probability, and phrase length ratio are used to evaluate equivalences. 1000 equivalences randomly selected are filtered and then evaluated. Experimental results indicate that its precision is 87.8% and recall is 89.8% for support vector machine.

Keywords: Data mining, discriminative features, discriminative learning, translation equivalence.

1. INTRODUCTION

Translation equivalences are very useful in a variety of applications such as bilingual lexicography [1], machine translation system [2] and cross-lingual information retrieval [3]. This is a data-mining problem. Many methods have been proposed for acquisition of translation equivalences. For example, Zhang builds mutual information matrix for a bilingual sentence pair, where the value for cell of matrix denotes the point-wise mutual information between word pair. Box-shaped region whose mutual information is similar is looked upon as equivalent phrase pair [4]. Kaji parses source language sentence and target language sentence. Phrase alignment is implemented on parsing trees of source and target sentences according to word alignment results [2]. But this method is restricted by accuracy of target language parser and grammar incompatibility of source-target languages, which leads its performance unsatisfying. Wong employes translation corresponding tree [5] to specify the correspondence between parsing tree of source sentence and target sentence, from which equivalences can be acquired. This partly solves the problems. Kenji uses translation literality to evaluate literality of bilingual sentence pairs and cleans the corpus in order to improve the quality of extracted equivalences [6]. But he has not considered filtering translation equivalences.

In this paper, translation equivalences are extracted from translation corresponding trees of bilingual sentence pairs. Discriminative features including translation literality, phrase alignment probability, and phrase length ratio are used to evaluate equivalences. Pearson R correlation coefficient is applied to evaluate the performance of discriminative features. At the same time, discriminative learning methods are employed to filter extracted equivalences. 1000 equivalences randomly selected are filtered and then evaluated in open test. Experimental results indicate that its precision is 87.8% and recall is 89.8% for support vector machine.

2. EXTRACTION OF TRANSLATION EQUIVALENCES

We extract translation equivalences from translation corresponding tree of Chinese-English bilingual sentence pairs. For a bilingual sentence pair (C,E), the process of equivalence extraction is shown as follows:

1. Tag and parse C and tag E. We assume that T is the parsing tree of Chinese sentence C.
2. Align words between C and E by word alignment tool. Extract corresponding words (called word links) from word alignment results.
3. For each subtree m in parsing tree T.

According to extracted word links, get string s from E which is the translation of subtree m. m→s is a translation equivalence.

For example, in the case of following bilingual sentence pair, the process of extracting equivalences is shown in Fig. 1.

Manuscript received April 3, 2005; revised June 4, 2006; accepted December 7, 2006. Recommended by Editorial Board member Hoon Kang under the direction of Editor Jin Young Choi. This work was supported by High Technology Research and Development Program of China (2002AA 117010-09), and National Grand Natural Science Foundation of China (Grant No. 60435020).

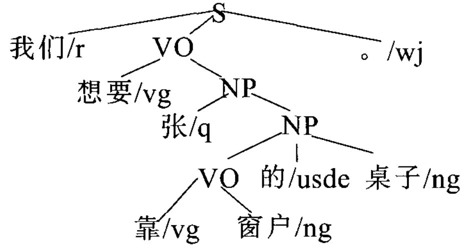
Chun-Xiang Zhang, Sheng Li, and Tie-Jun Zhao are with the School of Computer Science and Technology, Harbin Institute of Technology, 150001, China (e-mails: cxzhang@mtlab.hit.edu.cn, lisheng@hope.hit.edu.cn, tjzhao@mtlab.hit.edu.cn).

Chinese-English bilingual sentence pair:
Chinese: 我们想要张靠窗户的桌子。
English: We want to have a table near the window.

Word alignment result:

我们/1 想要/2 张/3 靠/4 窗户/5 的/6 桌子/7 。/8
We/1 want to/2 have/3 a/4 table/5 near/6 the/7 window/8 ./9
(1:1); (2:2); (4:6); (5:8); (7:5); (8:9);

Parsing tree of Chinese sentence:



Extracted translation equivalences:

VO[靠/vg 窗户/ng] → near the window
NP[VO[靠/vg 窗户/ng]的/usde 桌子/ng]→table near the window
NP[张/q NP[VO[靠/vg 窗户/ng]的/usde 桌子/ng]]→table near the window
VO[想要/vg NP[张/q NP[VO[靠/vg 窗户/ng]的/usde 桌子/ng]]]→want to have a table near the window

Fig. 1. Extract translation equivalences from bilingual sentence pairs.

3. EVALUATION OF TRANSLATION EQUIVALENCES

Equivalences extracted from above include lots of noise because the whole extraction process is restricted by accuracy of word alignment tool and Chinese parser. They should be filtered. Left part of extracted equivalence is a phrase with parsing information and the right part is only a phrase string. But when we filter them, phrase strings are only considered. For example, on determining whether 'VO[靠/vg 窗户/ng]→near the window' is a correct equivalence, we only consider '靠窗户→near the window'. Here, discriminative features including translation literality, phrase alignment probability, and phrase length ratio are utilized to evaluate equivalences.

3.1. Translation literality

A bilingual sentence pair that has many word correspondences is more literal. Translation literality is a widely used measure for weighting literality of bilingual sentence pairs [6]. It can also be used for computing the confidence that phrase in source language can be translated from and to phrase in target language. Translation literality of equivalence Ph_c→Ph_e is usually computed as (1) describes.

$$L(Ph_c, Ph_e) = \frac{Link(Ph_c, Ph_e)}{Num(Ph_c) + Num(Ph_e)}. \tag{1}$$

Here, Ph_c denotes Chinese phrase of equivalence. Ph_e denotes its English phrase. Link(Ph_c,Ph_e) denotes the number of word links between Ph_c and Ph_e. $Num(X)$ is the number of words in phrase X.

3.2. Phrase alignment probability

Brown uses $P(F|E)$ to compute the alignment probability of target language string E given source language string F [7]. The alignment probability $P(F|E)$ is shown in (2).

$$P(F \mid E) = \frac{1}{(l+1)^m} \prod_{j=1}^{m} \sum_{i=0}^{l} t(f_j \mid e_i). \tag{2}$$

In our approach, IBM Model-1 is applied to compute word-to-word translation probability $t(f|e)$ that a word f in the source language is translated given word e in the target language.

Given training data consisting of bilingual sentence pairs: $\{(f^{(s)},e^{(s)}), s=1,2,\cdots,S\}$, we use (3) to train word-to-word translation probability $t(f|e)$.

$$t(f \mid e) = \lambda_e^{-1} \sum_{s=1}^{S} c(f \mid e; f^{(s)}, e^{(s)}), \tag{3}$$

$$c(f \mid e, f^{(s)}, e^{(s)}) = \frac{t(f \mid e)}{\sum_{k=1}^{l} t(f \mid e_k)} \sum_{j=1}^{m} \delta(f, f_j) \sum_{i=1}^{l} \delta(e, e_i). \tag{4}$$

Here λ_e^{-1} is a normalization factor. $c(f \mid e, f^{(s)}, e^{(s)})$ denotes the expected number of times that word e connects to word f. We use $P(Ph_c|Ph_e)$ to calculate the alignment probability between Chinese phrase Ph_c and English phrase Ph_e in equivalence. We employ bilingual corpus including 300000 Chinese-English bilingual sentence pairs from general domain, which is developed by MOE-MS Key Laboratory of Natural Language Processing and Speech, to train word-to-word translation probability $t(f|e)$.

3.3. Phrase length ratio

The sentence length ratio is a very good indication of the alignment of a bilingual sentence pair [8]. For a given equivalence, we use phrase length ratio (described in formula (5)) to compute the confidence that phrase in source language can be translated from and to phrase in target language. For the language pair of Chinese and English, phrase length can be defined in several different ways. In general, a Chinese sentence does not have word boundary information. So one way to define Chinese phrase length is to count the number of bytes of the phrase. Another way is to first segment Chinese sentence into words and count how many words are in Chinese phrase. For English phrase, we can count its length in bytes and in words.

$$\begin{aligned} P(A \mid Ph_c, Ph_e) &= P(|Ph_c| <-> |Ph_e| \,|\, Ph_c, Ph_e) \\ &\cong P(|Ph_c| <-> |Ph_e| \,|\, |Ph_c|, |Ph_e|) \\ &\cong P(|Ph_c| - |Ph_e|) \\ &\cong P(D(|Ph_c|, |Ph_e|)). \end{aligned} \qquad (5)$$

Here, $|Ph_c|$, $|Ph_e|$ denote the length of Chinese phrase Ph_c and the length of English phrase Ph_e respectively.

The length difference $D(|Ph_c|,|Ph_e|)$ of Chinese phrase Ph_c and English phrase Ph_e is assumed to be a normal distribution [8]. It is computed according to (6).

$$D(|Ph_c|, |Ph_e|) = \frac{|Ph_e| - c|Ph_c|}{\sqrt{(|Ph_c|+1)\sigma^2}} \sim N(0,1). \qquad (6)$$

Here c is a constant indicating the mean length ratio which is the expected number of unites in English phrase Ph_e per unite in Chinese phrase Ph_c. σ^2 is the variance of c. For training data set $\{Ph^i_c \rightarrow Ph^i_e | i=1,2,\ldots,n\}$, c is computed according to (7). σ^2 is computed as (8) describes.

$$c = \sum_{i=1}^{n} Num_{unite}(Ph_e^i) \Big/ \sum_{i=1}^{n} Num_{unite}(Ph_c^i), \qquad (7)$$

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}((Num_{unite}(Ph_e^i) / Num_{unite}(Ph_c^i)) - c)^2. (8)$$

$Num_{unite}(X)$ is the number of unites in phrase X. unit may be word or character. For equivalence ‘靠窗户→near the window’, Num_{word}(靠 窗户)=2, Num_{word}(near the window)=3, $Num_{character}$(靠 窗户) =3, $Num_{character}$(near the window)=13.

Phrase length ratio is defined in (9):

$$L = P(D(|Ph_c|, |Ph_e|)). \qquad (9)$$

We employ training data to estimate c and σ^2. There are four methods to compute c and σ^2. So, we obtain 4 kinds of discriminative features according to (6). They are described in (10).

$$\begin{aligned} &L_1 : c = Num_{word}(Ph_e) / Num_{word}(Ph_c), \\ &L_2 : c = Num_{word}(Ph_e) / Num_{character}(Ph_c), \\ &L_3 : c = Num_{character}(Ph_e) / Num_{word}(Ph_c), \\ &L_4 : c = Num_{character}(Ph_e) / Num_{character}(Ph_c). \end{aligned} \qquad (10)$$

We use translation literality $L(Ph_c,Ph_e)$, phrase alignment probability $P(Ph_c|Ph_e)$, and phrase length ratio L_1,L_2,L_3,L_4 to score for equivalences respectively. If evaluation score of equivalence is larger, the confidence that it is correct translation equivalence is higher. N-Best strategy is employed to select translation equivalences with high confidence according to automatic evaluation scores.

4. DISCRIMINATIVE LEARNING

In order to improve filtering performance further, a linear combination model of multiple features is used to evaluate extracted equivalences, which will lead more noise being filtered. Discriminative features including translation literality $L(Ph_c,Ph_e)$, phrase alignment probability $P(Ph_c,Ph_e)$, and phrase length ratio L_1,L_2,L_3,L_4 are utilized. For a given translation equivalence $Ph_c \rightarrow Ph_e$, its evaluation score $y(Ph_c \rightarrow Ph_e)$ is calculated (11) describes. If $y(Ph_c \rightarrow Ph_e)$ is larger, the confidence of $Ph_c \rightarrow Ph_e$ being a correct equivalence is higher.

$$\begin{aligned} y(Ph_c \rightarrow Ph_e) = K_1 * L_1 + K_2 * L_2 + K_3 * L_3 + K_4 * L_4 \\ + K_5 * L(Ph_c, Ph_e) + K_6 * P(Ph_c \mid Ph_e). \end{aligned} \qquad (11)$$

So, the classification function can be defined as $h(Ph_c \rightarrow Ph_e) = WX + \theta$, where $W=(K_1,K_2,K_3,K_4,K_5,K_6)$ and $X=(L_1,L_2,L_3,L_4,L(Ph_c,Ph_e), P(Ph_c|Ph_e))$. In order to make the performance of classifier $h(Ph_c \rightarrow Ph_e)$ optimal, we employ training data to train parameters W and θ. We could ask human to annotate equivalences. If Ph_e can interpret Ph_c semantically, $Ph_c \rightarrow Ph_e$ is annotated as a positive instance. Otherwise it is viewed as a negative instance. Based on annotated equivalences, we could employ supervised learning method to train the classifier $h(Ph_c \rightarrow Ph_e)$. Here, we adopt discriminative learning methods: support vector machine and perceptron classifier to solve the problem.

We identify the two classes with the symbol $y \in \{-1,+1\}$, which indicates negative instance or positive instance. A training set of instances $S=\{X_i, y_i | i=1,2,\cdots,n\}$ is given. SVM chooses the optimal hyperplane $W^*X+\theta^*=0$ which can separate two different classes with the maximal distance. The optimum separating hyperplane is found based on the following (12).

$$\text{Minimize:} \quad \min\{\frac{1}{2}\|W\|^2\}. \qquad (12)$$

subject to the constraint:

$$y_i(WX_i + \theta) >= 1, \qquad i = 1,2,\ldots,n.$$

They are converted into the following quadratic programming optimization problem to get a^* [9].

$$\text{Minimize:} \quad \min Q(a) = \frac{1}{2}\sum_{i,j=1}^{n} a_i a_j y_i y_j X_i X_j - \sum_{i=1}^{n} a_i. (13)$$

subject to the constraint:

$$\sum_{i=1}^{n} a_i y_i = 0, \quad a_i >= 0, \quad i = 1,2,...,n.$$

We obtain W^* and θ^* according to (14).

$$W^* = \sum_{i=1}^{n} a_i^* y_i X_i, \qquad \theta^* = y_i - W^* X_i. \tag{14}$$

If the problem is not linearly separable, kernel transformations $K(X,Y)$ over vector spaces can be used to convert them from its feature space into high-dimensionality space. Under this situation, the discrimination function is shown in (15):

$$h(Ph_c -> Ph_e) = \sum_{i=1}^{n} a_i^* y_i K(X, X_i) + \theta^*. \tag{15}$$

Actually $h(Ph_c \rightarrow Ph_e) = WX + \theta$ can be viewed as a perceptron classifier [10].

In this paper, SVM and perceptron classifier are employed to solve the problem respectively.

5. EXPERIMENT

Method described in Section 2 is used. 286790 translation equivalences are obtained from 100000 Chinese-English bilingual sentence pairs. Performances of word alignment tool and Chinese parser are shown in Table 1.

We randomly select 6041 equivalences from these 286790 translation equivalences. Two human annotators are asked to manually annotate these 6041 equivalences together. We divide these 6041 equivalences into two parts. One is training data and the other is test data. They are described in Table 2.

c and σ^2 are parameters in phrase length ratio feature of L_1, L_2, L_3, and L_4. We employ training data to estimate parameter c according to (7). Parameter σ^2 is estimated by (8).

We use discriminative features mentioned in Section 3 to score for every equivalence in test data. Then pearson R correlation coefficient is applied to evaluate the magnitude of the association between automatic evaluation and manually-annotated results. Its computation is described as (16). The evaluation results are shown in Table 3.

Table 1. Word alignment tool and Chinese parser.

	Precision	Recall
Word alignment tool	86%	89%
Chinese parser	78%	79%

Table 2. Training data and test data.

	Positive	Negative	Error rate
Training data	3697	1338	26.57%
Test data	743	263	26.14%

Table 3. Pearson R correlation coefficient between automatic evaluation and manually-annotated results.

	Pearson R
$L(Ph_c,Ph_e)$	0.39362
$P(Ph_c \mid Ph_e)$	0.20239
L_1	0.35263
L_2	0.29717
L_3	0.43276
L_4	0.38780

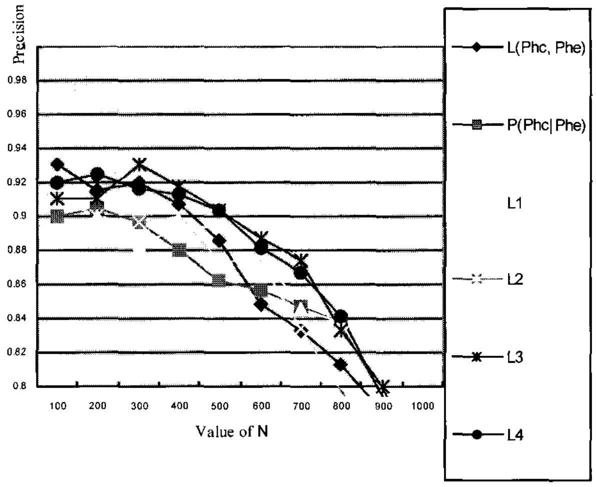


Fig. 2. Filtering performance of discriminative features under different N-Best strategy.

$$r = \frac{n(\sum XY) - (\sum X)(\sum Y)}{\sqrt{[n\sum X^2 - (\sum X)^2]}\sqrt{[n\sum Y^2 - (\sum Y)^2]}}. \tag{16}$$

From Table 3, we find that L_3 does better than other features on evaluating translation equivalences.

We sort translation equivalences in test data according to automatic evaluation score and employ N-Best strategy to label equivalences. The front N equivalences whose scores are highest are labeled as positive instances and others are labeled as negative ones. We set N=100,200,300,...,900, and label equivalences according to evaluation score under different N-Best strategy. Then automatically-labeled results are evaluated according to manually-annotated results. We use precision as measure to evaluate filtering performance. The evaluation results are shown in Fig. 2.

From Fig. 2, we can see that L_3 does better than other features on filtering performance.

We employ training data to train SVM and perceptron classifier respectively. We use optimized SVM and perceptron classifier to classify test data. Then classifying results are evaluated according to results given by human annotators. Precision, recall

Table 4. Filtering performance of translation equivalences.

	Precision	Recall	F1	Error rate
No-Filtering	73.9%	—	—	26.1%
Perceptron	82.0%	90.7%	86.1%	18.0%
SVM	87.8%	89.8%	88.8%	12.2%

and F1 are used as measures to evaluate filtering performance of classifiers. Equivalences labeled as positive instances are selected and negative ones are deleted. We use error rate to evaluate performance of filtered test data. The results see Table 4. From Table 4, we can find that its precision is 87.8% and recall is 89.8% for SVM. Error rate of test data decreases from 26.14% to 12.2% after filtering. Filtering performance exceeds no-filtering performance.

6. CONCLUSIONS

In this paper, we extract translation equivalences from translation corresponding trees of bilingual sentence pairs. Discriminative features including translation literality, phrase alignment probability, and phrase length ratio are used to evaluate equivalences. At the same time, discriminative learning methods are applied to filter extracted equivalences. Experimental results indicate that its precision is 87.8% for SVM, which exceeds that of translation equivalences without filtering. Automatically-Filtering can decrease human labeling cost.

REFERENCES

[1] W. A. Gale and K. W. Church, "Identifying word correspondences in parallel texts," *Proc. of the 4th DARPA Workshop on Speech and Natural Language*, pp. 152-157, 1991.

[2] H. Kaji, Y. Kida, and Y. Morimoto, "Learning translation templates from bilingual texts," *Proc. of the 14th International Conference on Computational Linguistics*, pp. 672-678, 1992.

[3] D. W. Oard and B. J. Dorr, *A Survey of Multilingual Text Retrieval*, Technical Report, University of Maryland, 1996.

[4] Y. Zhang, S. Vogel, and A. Waibel, "Integrated phrase segmentation and alignment model for statistical machine translation," *Proc. of International Conference on Natural Language Processing and Knowledge Engineering*, 2003.

[5] F. Wong, D. C. Hu, Y. H. Mao, and M. C. Dong, "A flexible example annotation schema: Translation corresponding tree representation," *Proc. of the 20th International Conference on Computational Linguistics*, pp. 1079-1085, 2004.

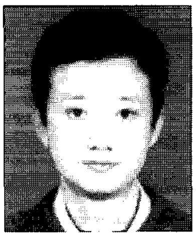
[6] K. Imamura and E. Sumita, "Bilingual corpus cleaning focusing on translation literality," *Proc. of the 7th International Conference on Spoken Language Processing*, pp. 1713-1716, 2002.

[7] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer, "The mathematics of statistical machine translation: Parameter estimation," *Computational Linguistics*, vol. 19, no. 2, pp. 263-311, 1993.

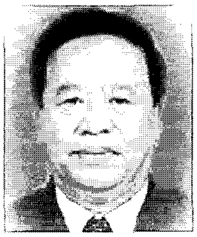
[8] K. W. Church, "Char_align: A program for aligning parallel texts at the character level," *Proc. of Meeting of the Association for Computational Linguistics*, pp. 1-8, 1993.

[9] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273-297, 1995.

[10] Y. Li, H. Zaragoza, R. Herbrich, J. Shawe-Taylor, and J. Kandola, "The perceptron algorithm with uneven margins," *Proc. of the 9th International Conference on Machine Learning*, pp. 379-386, 2002.



Chun-Xiang Zhang is a doctor candidate in Harbin Institute of Technology, China. His research interests are natural language process-ing and machine learning.



Sheng Li is a Professor and Ph.D. supervisor in Harbin Institute of Technology, China. His research interests are natural language process-ing and machine learning.

Tie-Jun Zhao is a Professor and Ph.D. supervisor in Harbin Institute of Technology, China. His research interests are natural language process-ing and machine learning.