2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Recognizing Cross-lingual Textual Entailment with Co-training using Similarity and Difference Views

Jiang Zhao, Man Lan*
Department of Computer Science and Technology
East China Normal University, Shanghai, P.R.China
Email: 51121201042@ecnu.cn
mlan@cs.ecnu.edu.cn

Zheng-Yu Niu
Baidu Inc.
Beijing, P.R.China
Email: niuzhengyu@baidu.com

Donghong Ji
School of Computer Science
Wuhan University, Hubei, P.R.China
Email: dhji@whu.edu.cn

***Abstract*—Cross-lingual textual entailment is a relatively new problem that detects the entailment relationship between two text fragments written in different languages. Previous work adopted machine learning algorithms and similarity measures as features to address this task. In order to overcome the high cost of human annotation and further improve the recognition performance, we present a novel co-training approach to solve this problem. We first use an off-the-shelf machine translation tool to eliminate the language gap between two texts. Then we measure the similarities and differences between two texts and regard them as *sufficient* and *redundant* views. We use those two views to conduct the co-training procedure to perform classification. Besides, a new effective Kullback-Leibler (KL) based criterion is proposed to select the results from all possible iterations. Experiments on cross-lingual datasets provided by SemEval 2013 show that our method significantly outperforms the baseline systems and previous work.**

I. Introduction

Traditional textual entailment (TE) [1] aims to detect the semantic entailment relationship (e.g., *forward*, *backward*) between two topic-related text fragments (termed Text (T) and Hypothesis (H) respectively). We specify that T entails H if the opinion or the statement of H holds in every circumstance in which T is true and thus the entailment relationship between T and H is *forward*, that is T→ H. Here is an example from the monolingual TE corpus [2], where T entails H because

Example 1
T: The earliest known Egyptian mummy, nicknamed "Ginger" for its hair colour, dates back to approximately 3300 BC.
H: The oldest known Egyptian mummy dates back to roughly 3300 BC and is nicknamed "Ginger".
(id=130, entailment=*forward*)

the statement of H can be inferred from the meaning of T. However, in Example 1, H cannot entail T because H does not explain the reason for the nickname. In this case, the pairs of sentences in TE are described in the same language while cross-lingual textual entailment (CLTE) [3] extends this task by involving cross-linguality where T and H are written in different languages. Targeting to recognize the semantic equivalence and difference between two text fragments written in different languages, CLTE can be integrated into a number of cross-lingual natural language processing (NLP) applications, for example, plagiarism detection [4] and information retrieval [5], where semantic inference across languages is needed. Besides, automatical content synchronization for the documents in different language versions to enrich each other is the ideal scenario for the exploitation of CLTE components [3], which will be of great benefit to manage textual information in multilingual settings, such as Wikipedia.

As a relatively new task, CLTE attracts many researchers' attention since Mehdad et al [3] proposed this task in 2010. A great deal of approaches have been proposed to solve the CLTE problem and most of these methods involve two steps: first they used the machine translators to convert the sentences into same language and then extracted a variety of discriminating features to feed to classifiers built with labeled training data (e.g., SVM [6], decision tree [7]) to make prediction. It is noteworthy that given the translated or cross-lingual sentence pairs, most discriminating features used in previous work only includes the similarity scores of the two sentence strings, such as WordNet-based similarities used in [8], cosine similarity, Dice coefficient and other nine similarity measures used in [9], which are calculated with the help of external lexical resources (e.g., WordNet [10], VerbOcean [11]). These similarity measures recently are successfully applied in CLTE task and the evaluation of semantic textual similarity in the *SEM 2013 Shared Task [12]. Meanwhile, in our preliminary work, we observed that many entailment relationships between two sentences can be determined by only tiny parts of the sentences. In Example 1, we can see that T and H share the same meaning words except the underlined terms in T, which suggests T entails H and H cannot entail T. In another example, i.e., Example 2 (for ease of exposition, we select the monolingual TE pair as our example, however this phenomenon is also often present in cross-lingual setting), we can see that, each sentence has a

Example 2
T: A study conducted by Harvard University researchers has concluded that children who attend an Independence Day celebration are more likely to become Republicans as adults.
H: Children who attend an Independence Day celebration (specifically celebrations without rain) are more likely to become Republicans when adults, according to a study carried out by a group of researchers.
(id=5, entailment=*no_entailment*)

small part different from the other, which indicates that there is no entailment relationship between them, i.e., *no_entailment*. Based on this observation, a novel sentence difference feature type was proposed to characterize this phenomenon in our previous work [6] and achieved the best result in this task. The similarity and difference measurements can be regarded as two *sufficient* and *redundant* views of data.

Additionally, like many NLP applications, the labeled data

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

for CLTE is fairly expensive to obtain because it costs a lot of human effort. Therefore, up to now the CLTE corpus is quite scarce and the amount of labeled instances is still small. The systems trained on insufficient data always lead to poor performance, for example, the best results reported in SemEval 2013 [2] in terms of accuracy are only 43.4%-45.8% on four language pairs.

In order to address the above problems, in this paper we propose a novel co-training method to improve the performance of cross-lingual textual entailment recognition. The strategy of co-training is to use two views of data to train two classifiers and add the predicted instances by one classifier to expand the training set of another classifier and then re-train the two classifiers on the expanded training sets. Specifically, we first use a state-of-the-art machine translator (MT) to solve the cross-linguality problem. Then, we extract two types of features, i.e., similarity features and difference features, from each sentence pair and these two feature types serve as two sufficient and redundant views for the subsequent co-training algorithm. After that, the co-training algorithm iteratively trains two classifiers on two views of data and outputs the results after a certain iteration. Instead of picking out the results after a fixed number of iteration, we come up with a new criterion based on the distribution agreement between the training and test data to select the results from all possible iterations.

The specific contributions of this paper are the following.

- Different from previous work which solely relies on similarity features to conduct supervised classification process, we apply co-training algorithm to CLTE using two different types of feature (i.e., similarity and difference) to overcome the shortage of labeled data. Experiments on benchmark corpus show that our approach significantly outperforms the baseline and previous work.
- We propose a new KL-based criterion to select the results from all possible iterations in co-training and it is more effective than traditional stop criterions according to the empirical results.

The rest of this paper is organized as follows. Section 2 reviews related work on cross-lingual textual entailment and co-training algorithm. Section 3 presents our proposed approach for recognizing cross-lingual textual entailment. Section 4 reports the details about experiments and results. Section 5 discusses the results and the influences of parameters. Finally, we conclude this paper in Section 6 along with future work.

II. Related Work

A. Cross-lingual Textual Entailment

This task faces two main challenges. The first is the cross-linguality that two sentences in a pair are written in different languages. The strategies in monolingual TE cannot be directly used and different languages have different lingustic properties (such as, phonological, morphological, etc.) and structures [10]. The second is the language ambiguity which is an inherent and pervasive phenomenon existing in natural language, that is, the same meaning can be conveyed by a plenty of different texts and a certain span of texts can express different meanings in different contexts as well. Therefore, it causes a many-to-many mapping relationship between language expressions and meanings [13]. CLTE has to deal with the ambiguities in each of the two different languages and the ambiguities resulting from cross-lingual situation as well.

To address the above cross-linguality challenge, two different solutions have been proposed since Mehdad et al [3] first presented this task. The first method is to fully translate sentences in one language into another language using off-the-shelf MTs [6], [7], [14] and [15]. This translation solution brings this problem back to monolingual TE and thus they can directly adopt a lot of existing resources and algorithms proposed in the TE task. For instance, [15] exploited the EDITS tool developed for TE to solve CLTE after translation procedure. The second method directly adopts cross-lingual strategies by aligning the unit (e.g., a word or a phrase) of sentences in one language with sentences in another language with the aid of external resources. For example, [16] and [17] exploited the IBM alignment models trained using corresponding parallel data to establish mappings between words in two languages and extracted the features from the aligned sentences, such as the numbers of aligned words. In this work, we used the first strategy because that translation approaches can achieve quite good results proved in [3] and the second strategy needs extra large parallel corpora which are hard to obtain in some languages.

Traditional TE approaches can be roughly divided into two groups: logic inference based methods ([18], [19]) and machine learning based methods ([2], [9] and [20]). The logic inference based approach is to map the natural language expressions to the first-order logical meaning representations and then use an automated reasoning tool like Vampire or a theorem prover to check whether there is an entailment relationship or not along two directions. The machine learning based approaches usually extract different features including: various similarity measures at different levels such as, the word co-occurence similarity [10] at the surface text level, the cardinality of strings [14] or combinations of several string similarity measures [9] at the shallow sematic level, the dependency grammar tree similarity [20] at the syntax level and more complex similarity measures [8] at deep sematic level; the measures originating from machine translation evaluation (e.g., BLEU) [14], [21] where one sentence is treated as the translation of the other and other aspect features [22].

Besides, in order to tackle the challenge of language ambiguity, many lexical/semantic resources (e.g., WordNet, VerbOcean, Wikipedia) were used during the recognition procedure [11], [8]. Specifically, when estimating the similarities of two sentences, we can utilize WordNet to obtain the semantic information including *i*) semantic equivalence between two terms (i.e., synonymy relations), *ii*) lexical relations preserving entailment between words (i.e., hyponymy relations) and *iii*) similarity scores between words (e.g., Lin similarity). Apart from building mapping relationships between words using parallel corpora as mentioned above, [11] performed Latent Semantic Analysis (LSA) over Wikipedia to measure the relatedness between two words in the dataset.

In addition to the similarity features used in previous work, we also propose difference features based on our preliminary work [6] and feed this two views of features (i.e., similarity

and difference features) to the co-training algorithm to address CLTE. Under the framework of co-training, two views of features can be exploited more effectively and make less generalization errors on the unlabeled data rather than just simply mixing the two features together.

B. Semi-supervised Learning and Co-training

Semi-supervised learning is a class of machine learning methods, which makes use of large amount of unlabeled data in the classifier training process. Many semi-supervised learning methods have been proposed including: EM with generative mixture models [23], self-training [24], co-training [25], transductive support vector machines [26]. Among them, co-training [25] serves as a prominent achievement in this area and exploits two subparts of features to build two classifiers separately. Then these two classifiers can teach each other in the subsequent training procedure by iteratively adding the predicted unlabeled examples by one classifier to the training set of another classifier. As a result, the two subparts of features (i.e., two views) must satisfy two requirements: (1) sufficiency (in a certain degree we can trust the labels predicted by the two classifiers), and (2) conditionally independence given the class (so that the most confident predictions from one classifier are independently and identically distributed with the samples from another classifier). Although the requirements of sufficient and redundant views may not always be met in most real-world cases, the co-training paradigm has been successfully applied in many applications such as cross-lingual sentiment classification [27] and domain adaption [28].

III. Co-training for Cross-lingual Textual Entailment

A. Motivation

The idea of using co-training algorithm for CLTE is motivated by the following considerations. First, co-training as a semi-supervised learning method can effectively exploit a large number of unlabeled data. To date, the best known performance of CLTE is low (i.e., 43.4%-45.8% in accuracy on different language datasets in SemEval 2013 [2]) due to the insufficiency of labeled data, while co-training can make full use of unlabeled data. Second, on one hand, as shown in Section 1, almost all previous work used similarity features which have been proved to be useful to address CLTE. On the other hand, in one of our preliminary work [6] we proposed novel difference features, which are found to be effective for CLTE as well. The similarity and difference features can be intuitively regarded as two views of data and thus can be easily integrated with the co-training algorithm in nature.

Generally, the co-training algorithm exploits the diversity between two classifiers to obtain useful information from newly-labeled data to teach each other in the training process. This diversity can be achieved by *i*) applying the same learning algorithm on two sufficient and redundant views; *ii*) using different learning algorithms on a single view (i.e., the whole feature set); *iii*) employing the same learning algorithm but different parameters on the same feature set. In this paper, we use the same learning algorithm (i.e. SVM) with two different views, i.e. similarity and difference, to perform the co-training procedure.

B. The Proposed Co-training Framework

Let $L = \{(x_1^s, x_1^d, y_1), ..., (x_{|L|}^s, x_{|L|}^d, y_{|L|})\}$ denote the labeled example set, where x_i^s, x_i^d are the feature representations of the i-th instance based on similarity and difference measures, y_i is the label of the i-th instance (i.e., the entailment relationship) and $|L|$ is the number of labeled instances. Similarly, let $U = \{(x_{|L|+1}^s, x_{|L|+1}^d, \tilde{y}_{|L|+1}), ..., (x_{|L|+|U|}^s, x_{|L|+|U|}^d, \tilde{y}_{|L|+|U|})\}$ denote the unlabeled example set, where the instances are also represented in two views, $\tilde{y}$ means the predicted label and $|U|$ is the number of unlabeled examples. The detailed description of the proposed co-training algorithm for CLTE is illustrated in Algorithm 1.

Algorithm 1 The Co-training algorithm for CLTE

```
Given:
    - labeled example set L, unlabeled example set U
    - the number of iteration T
    - the number of most confident predicted examples k
    - a classification algorithm C and its parameters p
PROCESS:
 1: L1 ← (x_i^s, y_i) ∈ L, L2 ← (x_i^d, y_i) ∈ L
 2: h1 ← C(L1, p), h2 ← C(L2, p)
 3: repeat
 4:    ỹ_k ← sign(h1(x_k^s)), where x_k^s ∈ U
 5:    π1 ← {(x_j^d, ỹ_j)| j is the index of k most confident
       predictions by h1}
 6:    L2 ← L2 ∪ π1; U ← U ⊖ π1
 7:    ỹ_k ← sign(h2(x_k^d)), where x_k^d ∈ U
 8:    π2 ← {(x_j^s, ỹ_j)| j is the index of k most confident
       predictions by h2}
 9:    L1 ← L1 ∪ π2; U ← U ⊖ π2
10:    h1 ← C(L1, p), h2 ← C(L2, p)
11:    if U is empty then exit
12: until T iterations
OUTPUT:
13: for each x_u ∈ U do
14:    ỹ_u ← sign(h1(x_u^s) + h2(x_u^d))
15: end for
```

Initially, line 1-2 build two classifiers separately on similarity and difference view using the same learning method. Then in each iteration illustrated in line 3-12, each classifier performs the prediction on the current unlabeled example set U and after that the top k examples with highest confidence predicted from one classifier are selected and added to the training set of another and consequently these added examples are removed from U (corresponding to line 4-6 and 7-9). In line 10, the two classifiers are re-trained using their expanded training sets. After T iterations, the final predictions are performed by selecting the class labels with the highest confidence which consist of the outputs of the two classifiers. In the following subsection, we propose a novel KL-based criterion to select the final results from all possible iterations rather than choose the results after a fixed iteration number T.

C. A Novel Proposed KL-based Criterion

The co-training algorithm iteratively trains two classifiers and outputs the results after a certain number of iteration. Thus, there is a question about how to set an appropriate

iteration number T or a stop criterion that can achieve optimum performance. On one hand, if T is too small, then it almost degenerates to supervised learning setting and the benefit of unlabeled data cannot be fully exploited. On the other hand, if T is too big, the classification mistake made in previous iteration may be reinforced in the following iteration, and thus it leads to worse performance. Previous work ([25], [27] and [28]) set the number of iteration by hand or set a confident threshold and then stopped the iteration when there is no confident prediction any more. Instead of picking out the results after a manually fixed iteration, in this work we propose a novel criterion to automatically pick out the results from the possible iterations. Our assumption is that the training and test instances are randomly drawn from the same distribution. So the distribution estimated from training instances should agree with that from test samples with respect to the predicted labels. Therefore, we propose to use Kullback-Leibler (KL) divergence [29] to measure the degree of *agreement* between training and test data. KL divergence is a commonly used measure to assess the difference between two probability distributions P and Q, which is defined to be

$$D_{KL}(P||Q) = \int_{-\infty}^{+\infty} \ln(\frac{p(x)}{q(x)})p(x)dx \quad (1)$$

We use the predicted labels to calculate the KL divergence between training and test data as follows: suppose we have m classes to predict indexed by $l \in \{1, .., m\}$, let X^l_{trn} denote the training instances with label l, X^l_{tst} denote the test instances with the predicted label l and $P(X)$ denote the distribution of data X, then the KL divergence of training and test data is calculated as

$$KL(trn, tst) = \frac{1}{m}\sum_l D_{KL}(P(X^l_{trn})||P(X^l_{tst})) \quad (2)$$

We assume that the data follows Gaussian distribution in the experiments. We then have two KL divergences for the two classifiers in each iteration and use the averaged value as the KL score for each iteration. Finally, we select the results with the smallest KL divergence as the final results of co-training.

D. Feature Representation

We used the following six types of features in our experiments, which is based on our previous work [6].

Basic feature (BC): This feature consists of length measures on different sets including $|A|, |B|, |A - B|, |B - A|, |A \bigcup B|, |A \bigcap B|, |A|/|B|, |B|/|A|$, where A and B stand for two translated texts and the length of a set is defined as the number of non-repeated elements in this set.

Surface Text Similarity features (STS): Nine measurement functions are applied to a pair of texts including: Jaccard coefficient, Dice coefficient, overlap coefficient, weighted overlap coefficient, cosine similarity, Manhattan distance, Euclidean distance, edit distance, Jaro-Winker distance. The vector representations of texts use a TFIDF-based weighting scheme and the edit and Jaro-Winker distance operate at the word level.

Sematic Similarity features (SS): This feature is to model the semantic representations of sentence pairs. First the semantic representations of these sentences are constructed by weighted textual matrix factorization (WTMF) [30] and then they are used to calculate the following seven similarity measures: word-to-word based similarity, word-to-sentence based similarity, sentence-to-sentence based similarity, cosine similarity, weighted overlap coefficient, Manhattan and Euclidean distance.

Sentence Difference features (SD): This feature records the number of unmatched words in each sentence. Besides, we also count the number of unmatched words in each sentence that belongs to a small set of POS tags (i.e., NN, JJ, RB, VB and CD tags).

Grammatical Relationship features (GR): Every sentence is POS tagged and dependency parsed and then we apply the STS measures to the newly generated sentence consisting of only POS tags. Similarly, we also apply the BC measures to the sentence consisting of only grammatical relations.

Bias features(BS): This feature examines the differences between two sentences in some special aspects, i.e., polarity (affirmative or negative statement of a sentence) and named entities. If the polarities of two sentences are the same, we set the feature to 1, otherwise -1. Similarly, if all named entities in one sentence are found in the other sentence, the feature is set to 1, otherwise -1.

E. Two Views of Features

As stated above, the co-training algorithm trains two models on two different views and in this paper we state that sentence similarity and difference features are two different views of a pair of sentence. So for each type of feature we split the above features into similarity and difference view as shown in Table I. Here "the rest" in similarity view column means the features in the corresponding feature type minus the features belong to difference view. We exploit this two views to perform the co-training algorithm in the following experiments. In order to make a reasonable comparison, we also perform the classification by simply mixing them together.

TABLE I. THE PARTITION OF FEATURES IN EACH FEATURE TYPE.

Feature Type	Similarity View	Difference View
BC	the rest	$\|A - B\|, \|B - A\|$
STS	weighted overlap, Jaccard, Dice, Overlap, cosine similarity	Manhattan, Euclidean, Edit, Jar-Winker distance
SS	the rest	Manhattan, Euclidean
SD	-	all
GR	the rest	the combination of difference features of BC and STS
BS	-	all

IV. EXPERIMENTS AND RESULTS

A. Datasets

To evaluate our proposed co-training approach, we use the data set provided by SemEval 2013 [2] task 8 since to the best of our knowledge it is the biggest corpus for CLTE task so far. It consists of a collection of 1500 sentence pairs (1000 for training and 500 for test) in each language pair with four entailment relationships: *forward, backward, bidirectional, non-entailment* and each entailment relationship has equal number of examples. Four different language pairs are provided: German-English (DEU-ENG), French-English

(FRA-ENG), Italian-English (ITA-ENG) and Spanish-English (SPA-ENG).

B. Preprocess

We performed the following text preprocessing. First, we employed the Google Translation service to translate the sentences in German, French, Italian and Spanish into English, thus two sentences in a pair were in the same language. After that, all sentences were tokenized and lemmatized and all stop words were removed followed by equivalent replacement procedure. The equivalent replacement procedure consists of three kinds of replacements as follows:

Abbreviative replacement. For each word whose length is 2 or 3, we examined if it is an acronym of some expressions in the other sentence. This is because many phrases or organizations in one sentence can be abbreviated to a set of capitalized letters in another sentence, e.g. "New Jersey" is usually wrote as "NJ" for short.

Semantic replacement. In this step, we replaced a lemma in one sentence with another lemma in the other sentence if they are *i*) in the same synonymy set; or *ii*) gloss-related, which means a lemma appears in the gloss of the other. WordNet 2.1[1] was used for looking up the synonymy and gloss of a lemma. No word sense disambiguation was performed and all synsets for a particular lemma were considered.

Context replacement. Two lemmas are considered to be replaceable if they are in the same context. The context of a lemma is defined as the non-stopword lemmas surround it. In the experiments, we set the number of surrounding non-stopword lemmas as 3.

C. Performance Evaluation and Learning Method

To make the experimental results comparable and reasonable, we used the official metric to evaluate the system performance, i.e., *accuracy*. The learning method in co-training was SVM_Multiclass[2] with linear kernel and the trade-off parameter C=1000. The parameters in WTMF were: the dimension of sematic space was 100, the weight of missing words was 100 and the regularization factor was 0.01. In fact, the 1000 training examples consists of two parts: 500 training examples and 500 test examples in SemEval 2012. So we used the 500 test examples in SemEval 2012 as development set to tune all the parameters.

D. Baseline and Our Systems

In the experiments, we designed eight systems listed in Table II. The first five supervised learning systems with different feature sets act as the baseline systems and last three systems are to examine the performance of different configurations of the proposed co-training algorithm. We applied our proposed co-training based approach to conduct CLTE using all features and the best feature subset as shown in System *all-fea+co-training* and *sub-fea+co-training* respectively. As a comparison, System *all-fea* and *sub-fea* directly exploited SVM_Multiclass to perform CLTE. To verify the sufficiency and redundance of the two constructed views, we only used the similarity and difference features with SVM_Multiclass in System *simView* and *diffView* respectively. In addition, to explore the influence of the number of training examples, System *sub-fea** and *sub-fea*+co-training* only used the 500 training examples of SemEval 2012 as the training set instead of using 1000 examples (500 training + 500 test in SemEval 2012). In our preliminary experiments, the best feature subset for SPA-ENG, ITA-ENG, FRA-ENG and DEU-ENG are ALL-*BS*, ALL-*STS*, *SD+BC+STS* and ALL-*GR*, respectively.

TABLE II. THE CONFIGURATION OF EIGHT SYSTEMS IN OUR EXPERIMENTS, WHERE THE FIRST FIVE SYSTEMS ARE IN SUPERVISED LEARNING SETTING WITH DIFFERENT FEATURE SETS AND THE LAST THREE SYSTEMS ARE IN CO-TRAINING SETTING WITH DIFFERENT FEATURE SETS. SYSTEMS WITH ASTERISK ONLY USE 500 TRAINING EXAMPLES IN SEMEVAL 2012 AT FIRST.

System	Brief description
simView	similarity features and SVM_Multiclass
diffView	difference features and SVM_Multiclass
all-fea	all feature sets and SVM_Multiclass
sub-fea	best feature sets and SVM_Multiclass
*sub-fea**	best feature sets and SVM_Multiclass
all-fea+co-training	co-training with all feature sets
sub-fea+co-training	co-training with best feature sets
sub-fea+co-training*	co-training with best feature sets

E. Results

Table III summarizes the experimental results with respect to accuracy of five baseline systems described in Table II on four language pairs and the best results of CLTE officially published by SemEval 2013, where the best performance of baseline systems on each language pair is shown in bold font. From the table, we can find that the results of our baseline systems are comparable with those of SemEval 2013 (i.e., on DEU-ENG and FRA-ENG, our baseline systems achieve the same performance as the best known results but a little bit worse on the other two datasets). This indicates that our constructed baseline systems are reasonable and reliable.

TABLE III. THE PERFORMANCE (IN ACCURACY) OF 5 BASELINE SYSTEMS AND THE BEST RESULTS OFFICIALLY REPORTED BY SEMEVAL 2013 ON FOUR LANGUAGE PAIRS.

System	SPA-ENG	ITA-ENG	FRA-ENG	DEU-ENG
simView	0.398	0.402	0.436	0.438
diffView	0.414	0.412	**0.458**	0.416
all-fea	**0.428**	**0.426**	0.438	0.422
sub-fea	0.404	0.420	0.450	0.436
*sub-fea**	0.422	0.416	0.436	**0.452**
Best results in SemEval 2013	0.434	0.454	0.458	0.452

Table IV summarizes the experimental results of our proposed co-training based systems described in Table II on four language pairs. In the co-training setting, we set the growth size (i.e., k most confident predicted examples) at each iteration for FRA-ENG, DEU-ENG, ITA-ENG as 10 and 5 for SPA-ENG. From this table, we can find that on four language pairs, our co-training method obtains better performance than the best results reported in SemEval 2013.

To examine the impact of the difference in data on the performance variation, we employed the paired t-test to verify whether there is significant difference between different systems. Table V summarizes the paired-sample t-test results

[1] http://wordnet.princeton.edu/

[2] http://www.cs.cornell.edu/people/tj/svm_light/svm_multiclass.html

TABLE IV. THE PERFORMANCE (IN ACCURACY) OF OUR PROPOSED CO-TRAINING SYSTEMS AND THE BEST RESULTS OFFICIALLY REPORTED BY SEMEVAL 2013 ON FOUR LANGUAGE PAIRS.

System	SPA-ENG	ITA-ENG	FRA-ENG	DEU-ENG
all-fea+co-training	0.433	0.437	0.479	**0.477**
sub-fea+co-training	**0.443**	0.433	**0.491**	0.467
sub-fea+co-training*	0.439	**0.477**	0.481	0.455
Best results in SemEval 2013	0.434	0.454	0.458	0.452

on four language pairs between the best baseline systems (i.e., *all-fea*, *all-fea*, *diffView* and *sub-fea** on SPA-ENG, ITA-ENG, FRA-ENG and DEU-ENG respectively) and three co-training systems (i.e., *all-fea/sub-fea/sub-fea*+co-training*). The systems with insignificant performance differences are grouped into one set and " $>$ " and " $>>$ " denote better than at significance level 0.05 and 0.005 respectively.

V. DISCUSSION AND ANALYSIS

A. Discussion

From Table III and Table V, we have several findings as follows.

Firstly, the performance of two views, i.e., difference and similarity features, are comparable to each other. Moreover, they are also comparable to the other three supervised baseline systems. This indicates that the proposed two views are sufficient and redundant and thus they are suitable for the co-training algorithm. Specifically, on FRA-ENG, the system using only difference features achieves the best results among the other four systems with all or similarity features which indicates that the difference features are quite effective in themselves.

Secondly, our proposed co-training approach outperforms all corresponding supervised learning baseline systems over all language pairs with respect to different feature sets and different training instances. Specifically, on SPA-ENG, ITA-ENG, FRA-ENG and DEU-ENG, our best co-training systems achieve 1.5%, 5.1%, 4.1%, 2.5% accuracy improvements over the best baseline systems. To explore the possible reason why co-training performs better than supervised learning, we calculated the KL divergences between the training and test data on four language pairs and the KL divergences for FRA-ENG, DEU-ENG, ITA-ENG and SPA-ENG are 58.16, 63.03, 73.60 and 58.34, respectively. It is clear that the KL divergences between training and test data are quite big and thus the models learned on training data only may not perform good on test samples. However, our co-training method can make use of the test data in the training phase to overcome this difference between data. This may explain that our proposed co-training with two sufficient and redundant views and test data significantly improves the performance of CLTE, which is also confirmed by the following paired-sample t-tests listed in Table V.

Thirdly, our proposed co-training method outperforms the best results reported in SemEval 2013 over all language pairs. Specifically, our method obtains an improvement of 0.9%, 2.3%, 3.3%, 2.5% in terms of accuracy on SPA-ENG, ITA-ENG, FRA-ENG and DEU-ENG respectively. Notice that the results of our method reported in Table IV may be not the best results it can achieve since our result selection criterion based on KL-divergence in co-training does not always pick out the best results from all possible iterations.

Lastly, under the co-training framework, the systems with 1000 initial training instances achieve the best results on three language datasets except ITA-ENG and surprisingly the system *sub-fea*+co-training* with only 500 initial training instances achieves the best results on ITA-ENG dataset. This may be due to the big difference between 500 test samples of SemEval 2012 and 500 test samples of SemEval 2013 on ITA-ENG dataset (the KL divergences for FRA-ENG, DEU-ENG, ITA-ENG and SPA-ENG are 15.17, 40.35, 69.27 and 60.27, respectively).

In a nutshell, our proposed co-training method with two views is efficient and promising.

B. Influences of Parameters

In this subsection, we discuss the influences of two critical parameters in the co-training algorithm on CLTE, i.e., the growth size k and iteration number T.

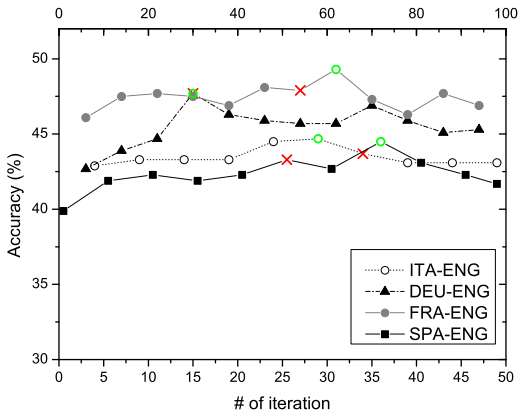

Fig. 1. The accuracy curves over different iteration number Ts. The green circles stand for the best results among all possible iterations and the red crosses represent the results selected by our KL-based criterion.

1) Growth Size k: To explore the influence of growth size k in co-training, we conducted a series of experiments on different k values (e.g., 5, 10, ... , 50) over four language pairs using all features. We achieved the best results when k=5 on SPA-ENG dataset and k=10 on DEU-ENG, FRA-ENG and ITA-ENG datasets. The accuracy curves over k on different language pairs have many local maxima instead of a global maximum, because the iteration numbers of the results selected by our criterion on different language pairs are different. Besides, for the purpose of balance, we also added equal number of most confident predictions for each class to the training set in each iteration. However, it performs worse than that of just adding the k most confident predictions regardless of classes. To further explore the reason for this, we observe and find that the classifiers are prone to misclassify examples with *forward*, *backward* and *bidirectional* labels and consequently in the next iteration, the classifiers will

TABLE V. STATISTICAL SIGNIFICANCE TESTS RESULTS.

Language pair	Paired-sample t-test results
SPA-ENG	(*sub-fea+co-training*, *sub-fea*+co-training*) > *all-fea+co-training* > *all-fea*
ITA-ENG	*sub-fea*+co-training* >> *all-fea+co-training* >> (*sub-fea+co-training*, *all-fea*)
FRA-ENG	*sub-fea+co-training* >> (*sub-fea*+co-training*, *all-fea+co-training*, *diffView*)
DEU-ENG	(*all-fea+co-training*, *sub-fea+co-training*, *sub-fea*+co-training*) >> *sub-fea**

input more misclassified training examples and lead to worse performance using this strategy. Therefore, in this work, we set k=5 for SPA-ENG and k=10 for the other three languages regardless of classes.

2) Iteration Number T: We conducted experiments to examine the influence of iteration number T using all features. Figure 1 shows the accuracy curves over different iteration numbers on different language pairs. The green circles in the figure stand for the best results among all possible iterations and the red crosses represent the final results selected by our KL-based criterion. As discussed above, we fixed the growth size $k = 10$ on DEU-ENG, FRA-ENG, ITA-ENG datasets and $k = 5$ on SPA-ENG and thus the maximum iteration number for DEU-ENG, FRA-ENG, ITA-ENG is 50 and 100 for SPA-ENG (since there are 500 test samples in total on each language pair). We plotted the four curves in a single figure and used the upper x-axis for SPA-ENG for aesthetic. Generally, we find:

- the performance of four curves increases as the iteration number increases and achieved a global optimum performance at a certain iteration number. And then the performance decreases as the iteration number increases. This is because that more noisy examples are selected as the iteration number increases and added into the next iteration;
- our proposed KL-based criterion is able to dynamically choose the optimum number of iteration for different datasets and it achieves a better performance than using a fixed number of iteration (i.e., the traditional criterion). Specifically, on DEU-ENG dataset our KL-based criterion achieves the best result and approximately approaches the best result on the other three datasets. This indicates that our proposed KL-based stop criterion is effective for co-training algorithm.

VI. CONCLUSIONS

In this paper, we propose two views (i.e., similarity and difference) of features for CLTE task and based on these two views we propose a co-training framework to improve the performance of CLTE. Besides, a new KL-based criterion is also proposed to select the final results based on the agreement between the distributions of training and test data. Results on the datasets provided by SemEval 2013 show that our method significantly outperforms all previous best systems. Specifically, for SPA-ENG, ITA-ENG, FRA-ENG, DEU-ENG, our best results achieve 1.5%, 5.1%, 3.3%, 2.5%, accuracy improvements over the best baseline systems and 0.9%, 2.3%, 3.3%, 2.5% improvements over the best results published in SemEval 2013 competition. This indicates that our method is feasible and promising. However, although co-training can improve the performance, the overall performance is still quite low (an average accuracy of 47.2%), so our future work includes: discover more discriminating features to better characterize the ambiguity and variety of language and exploit other learning strategies.

ACKNOWLEDGEMENTS

The authors would like to thank the reviewers for their helpful suggestions and comments, which improve the final version of this paper. This research is supported by grants from National Natural Science Foundation of China (No.61173062, No.60903093) and Shanghai Knowledge Service Platform Project (No. ZF1213).

REFERENCES

[1] I. Dagan and O. Glickman, "Probabilistic textual entailment: Generic applied modeling of language variability," in *Proceedings of the PASCAL Workshop on Learning Methods for Text Understanding and Mining*, 2004.

[2] M. Negri, A. Marchetti, Y. Mehdad, L. Bentivogli, and D. Giampiccolo, "Semeval-2013 Task 8: Cross-lingual Textual Entailment for Content Synchronization," in *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013)*, 2013.

[3] Y. Mehdad, M. Negri, and M. Federico, "Towards cross-lingual textual entailment," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Los Angeles, California, June 2010, pp. 321–324. [Online]. Available: http://www.aclweb.org/anthology/N10-1045

[4] Z. Ceska, M. Toman, and K. Jezek, "Multilingual plagiarism detection," in *Proceedings of the 13th International Conference on Artificial Intelligence: Methodology, Systems, and Applications*, ser. AIMSA '08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 83–92.

[5] P. Sorg and P. Cimiano, "Cross-lingual information retrieval with explicit semantic analysis," in *Working Notes for the CLEF 2008 Workshop*, 2008. [Online]. Available: http://www.clef-campaign.org/2008/working_notes/sorg_paperCLEF2008.pdf

[6] J. Zhao, M. Lan, and Z.-Y. Niu, "Ecnucs: Recognizing cross-lingual textual entailment using multiple text similarity and text difference measures," in *Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. Atlanta, Georgia, USA: Association for Computational Linguistics, June 2013, pp. 118–123.

[7] D. Vilarino, D. Pinto, S. León, Y. Alemán, and H. Gómez-Adorno, "Buap: N-gram based feature evaluation for the cross-lingual textual entailment task," in *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013)*, 2013.

[8] J. J. Castillo, "A wordnet-based semantic approach to textual entailment and cross-lingual textual entailment," *International Journal of Machine Learning and Cybernetics*, vol. 2, no. 3, pp. 177–189, 2011.

[9] P. Malakasiotis and I. Androutsopoulos, "Learning textual entailment using svms and string similarity measures," in *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, 2007, pp. 42–47.

[10] T.-L. Ha, "Lexical-syntactic approaches for english-dutch cross-lingual textual entailment," Master's thesis, University of Groningen, 2011.

[11] Y. Mehdad, M. Negri, and M. Federico, "Using bilingual parallel corpora for cross-lingual textual entailment," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011, pp. 1336–1345.

[12] E. Agirre, D. Cer, M. Diab, A. Gonzalez-Agirre, and W. Guo, "*sem 2013 shared task: Semantic textual similarity," in *Second Joint Conference on Lexical and Computational Semantics (*SEM)*. Association for Computational Linguistics, June 2013, pp. 32–43.

[13] O. Glickman, "Applied textual entailment," Ph.D. dissertation, Bar Ilan University, 2006.

[14] S. Jimenez, C. Becerra, and A. Gelbukh, "Softcardinality: Learning to identify directional cross-lingual entailment from cardinalities and smt," in *Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. Association for Computational Linguistics, June 2013, pp. 34–38.

[15] F. Meng, H. Xiong, and Q. Liu, "Ict: a translation based method for cross-lingual textual entailment," in *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*. Association for Computational Linguistics, 2012, pp. 715–720.

[16] M. Turchi and M. Negri, "Altn: Word alignment features for cross-lingual textual entailment," in *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013)*, 2013.

[17] K. Wäschle and S. Fendrich, "Hdu: cross-lingual textual entailment with smt features," in *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*. Association for Computational Linguistics, 2012, pp. 467–471.

[18] J. Bos and K. Markert, "Recognising textual entailment with logical inference," in *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 2005, pp. 628–635.

[19] M. Tatu and D. Moldovan, "A semantic approach to recognizing textual entailment," in *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 2005, pp. 371–378.

[20] P. Malakasiotis, "Paraphrase recognition using machine learning to combine similarity measures," in *Proceedings of the ACL-IJCNLP 2009 Student Research Workshop*, 2009, pp. 27–35.

[21] D. Perez and E. Alfonseca, "Application of the bleu algorithm for recognising textual entailments," in *Proceedings of the First Challenge Workshop Recognising Textual Entailment*. Citeseer, 2005, pp. 9–12.

[22] A. Iftene and A. Balahur-Dobrescu, "Hypothesis transformation and semantic variability rules used in recognizing textual entailment," in *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*. Association for Computational Linguistics, 2007, pp. 125–130.

[23] A. Fujino, N. Ueda, and K. Saito, "A hybrid generative/discriminative approach to semi-supervised classifier design," in *Proceedings of the National Conference on Atificial Intelligence*, vol. 20, no. 2. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005, p. 764.

[24] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 1995, pp. 189–196.

[25] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, ser. COLT' 98. New York, NY, USA: ACM, 1998, pp. 92–100.

[26] T. Joachims, "Transductive inference for text classification using support vector machines," in *ICML*, vol. 99, 1999, pp. 200–209.

[27] X. Wan, "Co-training for cross-lingual sentiment classification," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*. Association for Computational Linguistics, 2009, pp. 235–243.

[28] M. Chen, K. Q. Weinberger, and J. Blitzer, "Co-training for domain adaptation," in *Advances in Neural Information Processing Systems*, 2011, pp. 2456–2464.

[29] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[30] W. Guo and M. Diab, "Modeling sentences in the latent space," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, 2012, pp. 864–872.