Semi-supervised Sequence Labeling for Named Entity Extraction based on Tri-Training: Case Study on Chinese Person Name Extraction

Chien-Lung Chou[1]
National Central University,
Taoyuan, Taiwan
formatc.chou@gmail.com

Chia-Hui Chang
National Central University,
Taoyuan, Taiwan
chia@csie.ncu.edu.tw

Shin-Yi Wu
Industrial Technology
Research Institute, Taiwan
sywu@itri.org.tw

Abstract

Named entity extraction is a fundamental task for many knowledge engineering applications. Existing studies rely on annotated training data, which is quite expensive when used to obtain large data sets, limiting the effectiveness of recognition. In this research, we propose an automatic labeling procedure to prepare training data from structured resources which contain known named entities. While this automatically labeled training data may contain noise, a self-testing procedure may be used as a follow-up to remove low-confidence annotation and increase the extraction performance with less training data. In addition to the preparation of labeled training data, we also employed semi-supervised learning to utilize large unlabeled training data. By modifying tri-training for sequence labeling and deriving the proper initialization, we can further improve entity extraction. In the task of Chinese personal name extraction with 364,685 sentences (8,672 news articles) and 54,449 (11,856 distinct) person names, an F-measure of 90.4% can be achieved.

1 Introduction

Detecting named entities in documents is one of the most important tasks for message understanding. For example, the #Microposts 2014 Workshop hosted an "Entity Extraction and Linking Challenge", which aimed to automatically extract entities from English microposts and link them to the corresponding English DBpedia v3.9 resources (if a linkage existed). Like many other types of research, this task relies on annotated training examples that require large amounts of manual labeling, leading to a limited number of training examples (e.g. 2.3K tweets). While human-labelled training examples (L) have high quality, their cost is very high. Thus the major concern in this paper is how to prepare training data for entity extraction learning on the Web.

In practice, sometimes there are existing structured databases of known entities that are valuable to improve extraction accuracy. For examples, personal names, school names, and company names can be obtained from a Who's Who website, and accessible government data for registered schools and businesses, respectively. Meanwhile, there are many unlabeled training examples that can be used for many information extraction tasks. If we can automatically label known entities in the unlabeled training examples, we can obtain large labeled training set. While such training data may contain errors, self-testing can be applied to filter unreliable labeling with less confidence.

On the other hand, the use of unlabeled training examples (U) has also been proved to be a promising technique for classification. For example, co-training (Blum and Mitchell, 1998) and tri-training (Zhou et al. 2005) are two successful techniques that use examples with high-confidence as predicted by the other classifier or examples with consensus answers from the other two classifiers in order to prepare new labeled training data for learning. By estimating the error rate of each learned classifier, we can calculate the maximum number of new consensus answers for learning to ensure the error rates are reduced.

In this paper, we explore the possibility of extending semi-supervised learning to sequence labeling via tri-training so that unlabeled training examples can also be used in the learning phase. The challenge here is to obtain a common label sequence as a consensus answer from multiple models. As enumerating

[1] This research was partially supported by ITRI, Taiwan under grant B2-101052.

Proceedings of Third Workshop on Semantic Web and Information Extraction, pages 33–40,
Dublin, Ireland, 24 August, 2014.

all possible label sequences will be too time-consuming, we employ a confidence level to control the co-labeling answer such that a label sequence with the largest probability is selected. Comparing with a common label sequence from multiple models, the most probable label sequence has larger chance to obtain a consensus answer for training and testing.

In addition to the extension of tri-training algorithm to sequence labeling, another key issue with tri-training is the assumption of the initial error rate (0.5), leading to a limited number of co-labeling examples for training and early termination for large set training. Therefore, a new estimation method is devised for the estimation of initial error rate to alleviate the problem and improve the overall performance.

To validate the proposed method, we conduct experiments on Chinese personal name extraction using 7,000 known Chinese celebrity names (abbreviated as CCN). We collect news articles containing these personal names from Google's search engine (using these names as keywords) and automatically label these articles containing CCN and known reporters' names. In a test set of 8,672 news articles (364,685 sentences) containing 54,449 personal names (11,856 distinct names), the basic model built on CRF (conditional random field) has a performance of 76.8% F-measure when using 500 celebrity names for preparing training data, and is improved to 86.4% F-measure when 7,000 celebrity names are used. With self-testing, the performance is improved to 88.9%. Finally, tri-training can further improve the performance through unlabeled data to 90.4%.

2 Related Work

Entity extraction is the task of recognizing named entities from unstructured text documents, which is one of the information tasks to test how well a machine can understand the messages written in natural language and automate mundane tasks normally performed by human. The development of machine learning research from classification to sequence labeling such as the HMM (Hidden Markov Model) (Bikel et al., 1997) and the CRF (Conditional Random Field) (McCallum and Wei, 2003) has been widely discussed in recent years. While supervised learning shows an impressive improvement over unsupervised learning, it requires large training data to be labeled with answers. Therefore, semi-supervised approaches are proposed.

Semi-supervised learning refers to techniques that also make use of unlabeled data for training. Many approaches have been previously proposed for semi-supervised learning, including: generative models, self-learning, co-training, graph-based methods (Zhou et al. 2005) and information-theoretic regularization (Zheng et al. 2009). In contrast, although a number of semi-supervised classifications have been proposed, semi-supervised learning for sequence segmentation has received considerably less attention and is designed according to a different philosophy.

Co-training and tri-training have been mainly discussed for classification tasks with relatively few labeled training examples. For example, the original co-training paper by Blum and Mitchell (1998) described experiments to classify web pages into two classes using only 12 labeled web pages as examples. This co-training algorithm requires two views of the training data and learns a separate classifier for each view using labeled examples. Nigam and Ghani (2000) demonstrated that co-training performed better when the independent feature set assumption is valid. For comparison, they conducted their experiments on the same (WebKB course) data set used by Blum and Mitchell.

Goldman and Zhou (2000) relaxed the redundant and independent assumption and presented an algorithm that uses two different supervised learning algorithms to learn a separate classifier from the provided labeled data. Empirical results demonstrated that two standard classifiers can be used to successfully label data for each other with 95% confidence interval.

Tri-training (Zhou, et al. 2005) was an improvement of co-training, which used three classifiers and a voting mechanism to solve the confidence issue of co-labeled answers by two classifiers. In each round of tri-training, the classifiers h_j and h_k choose some examples in U to label for h_i ($i,j,k \in \{1,2,3\}$, $i \neq j \neq k$). Let L_i^t denote the set of examples that are labeled for h_i in the t-th round. Then the training set for h_i in the t-th round are $L \cup L_i^t$. Note that the unlabeled examples labeled in the t-th round, i.e. L_i^t, won't be put into the original labeled example set, i.e. L. Instead, in the $(t+1)$-th round all the examples in L_i^t will be regarded as unlabeled and put into U again.

While Tri-training has been used in many classification tasks, the application in sequence labeling tasks is limited. Chen et al. (2006) proposed an agreement measure that computed the unit consistency

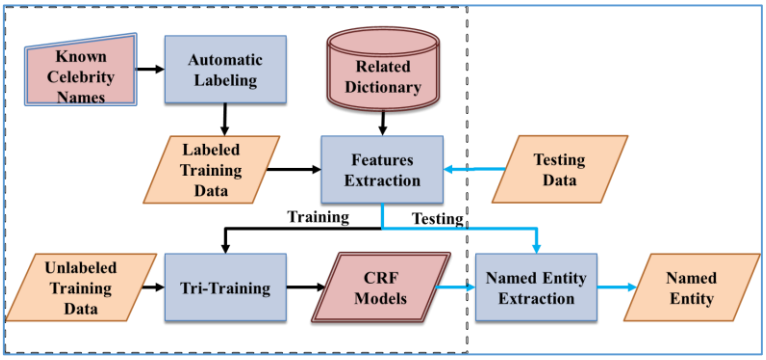

Figure 1 Semi-Supervised Named Entity Extraction Based on Automatic Labeling and Tri-training

between two label sequences from two models. Then based on the agreement measure, the idea is to choose a sentence, which is correctly labeled by h_j and h_k but is not parsed correctly by the target classifier h_i, to be a new training sample. A control parameter is used to determine the percentage (30%) of examples selected for the next round. The process iterates until no more unlabeled examples are available. Thus, Chen et al.'s method does not ensure the PAC learning theory.

3 System Architecture

Due to the high cost of labeling, most benchmarks for NER are limited to several thousand sentences. For example, the English dataset for the CoNLL 2003 shared task (Tjong et al., 2003) consists of 14,987 training sentences for four entity categories, PER, LOC, ORG, and MISC. But it is unclear whether sufficient data is provided for training or the learning algorithms have reached their capacity. Therefore, two intuitive ways are considered in this paper: one is automatic labeling of unlabeled data for preparing a large amount of annotated training examples, and the other is semi-supervised learning for making use of both labeled and unlabeled data during learning.

For the former, *automatic labeling* is sometimes possible, especially for named entities which can be obtained from Web resources like DBpedia. For example, suppose we want to train a named entity extractor for the Reuters Corpus, we can use the known entities from CoNLL 2003 shared task as queries to obtain documents that contain queries from the Reuters Corpus and label the articles automatically. While such automatic annotation may involve wrong labeling, we can apply *self-testing* to filter low-confidence labels. Overall, the benefit of the large amount of labeled training examples is greater than the noise it may cause.

In this paper, we propose a hybrid model composed of the following modules: automatic labeling, feature engineering, and tri-training based algorithm for training and testing. The framework is illustrated in Figure 1.

3.1 Tri-training for Classification

Let L denote the labeled example set with size $|L|$ and U denote the unlabeled example set with size $|U|$. In each round, t, tri-training uses two models, h_j and h_k, to label the answer of each instance x from unlabeled training data U. If h_j and h_k give the same answer, then we could use x and the common answer pair as newly training example, i.e. $L_i^t = \{(x, y): x \in U, y = h_j^t(x) = h_k^t(x)\})$ for model h_i $(i, j, k \in \{1,2,3\},\ i \neq j \neq k)$. To ensure that the error rate is reduced through iterations, when training h_i, Eq. (1) must be satisfied,

$$e_i^t |L_i^t| < e_i^{t-1} |L_i^{t-1}| \tag{1}$$

where e_i^t denotes the error rate of model h_i in L_i^t, which is estimated by h_j and h_k in the t-th round using the labeled data L by dividing the number of labeled examples on which both h_j and h_k make an incorrect estimation by the number of labeled examples for which the estimation made by h_j is the same as that made by h_k, as shown in Eq. (2).[2,]

[2] Assuming that the unlabeled examples hold the same distribution as that held by the labeled ones.

$$e_i^t = \frac{|\{(x,y)\in L,\, h_j^t(x)=h_k^t(x)\neq y\}|}{|\{(x,y)\in L,\, h_j^t(x)=h_k^t(x)\}|} \tag{2}$$

If $\left|L_i^t\right|$ is too large, such that Eq. (1) is violated, it would be necessary to sample maximum u examples from L_i^t such that Eq. (1) can be satisfied.

$$u = \left\lceil \frac{e_i^{t-1}|L_i^{t-1}|}{e_i^t} - 1 \right\rceil \tag{3}$$

$$S_i^t = \begin{cases} Subsample(L_i^t, u) & violated\ Eq.(1) \\ L_i^t & otherwise \end{cases} \tag{4}$$

For the last step in each round, the union of the labeled training examples L and S_i^t, i.e. $L \cup S_i^t$, is used as training data to update classifier h_i for this iteration.

3.2 Modification for the Initialization

According to Eq. (1), the product of error rate and new training examples define an upper bound for the next iteration. Meanwhile, $\left|L_i^{t-1}\right|$ should satisfy Eq. (5) such that $\left|L_i^t\right|$ after subsampling, i.e., u, is still bigger than $\left|L_i^{t-1}\right|$.

$$|L_i^{t-1}| > \frac{e_i^t}{e_i^{t-1} - e_i^t} \tag{5}$$

In order to estimate the size of $\left|L_i^1\right|$, i.e., the number of new training examples for the first round, we need to estimate e_i^0, e_i^1, and $\left|L_i^0\right|$ first. Zhou et al. assumed a 0.5 error rate for e_i^0, computed e_i^1 by h_j and h_k, and estimated the lower bound for $\left|L_i^0\right|$ by Eq. (6), thus:

$$\left|L_i^0\right| = \left\lfloor \frac{e_i^1}{e_i^0 - e_i^1} + 1 \right\rfloor = \left\lfloor \frac{e_i^1}{0.5 - e_i^1} + 1 \right\rfloor \tag{6}$$

The problem with this initialization is that, for a larger dataset $|L|$, such an initialization will have no effect on retraining and will lead to an early stop for tri-training. For example, consider the case when the error rate e_i^1 is less than 0.4, then the value of $\left|L_i^0\right|$ will be no more than 5, leading to a small upper bound for $e_i^1\left|L_i^1\right|$ according to Eq. (1). That is to say, we can only sample a small subset $\left|S_i^1\right|$ from L_i^1 for training h_i based on Eq. (4). On the other hand, if e_i^1 is close to 0.5 such that the value of $\left|L_i^0\right|$ is greater than the original dataset $|L|$, it may completely alter the behavior of h_i.

To avoid this difficulty, we propose a new estimation for the product $e_i^0\left|L_i^0\right|$. Let $L^C(h_j, h_k)$ denote the set of labeled examples (from L) on which the classification made by h_j is the same as that made by h_k in the initial round, and $L^W(h_j, h_k)$ denote the set of examples from $L^C(h_j, h_k)$ on which both h_j and h_k make incorrect classification, as shown in Eq. (7) and (8). In addition, we define $L_i^W(h_j, h_k)$ to be the set of examples from $L^C(h_j, h_k)$ on which h_i makes incorrect classification in the initial round, as shown in Eq. (9). The relationship among $L^C(h_j, h_k)$, $L^W(h_j, h_k)$, and $L_i^W(h_j, h_k)$ is illustrated in Figure 2.

$$L^C(h_j, h_k) = \{(x,y) \in L : h_j(x) = h_k(x)\} \tag{7}$$

$$L^W(h_j, h_k) = \{(x,y) \in L^C(h_j, h_k) : h_j(x) \neq y\} \tag{8}$$

$$L_i^W(h_j, h_k) = \{(x,y) \in L^C(h_j, h_k) : h_i(x) \neq y\} \tag{9}$$

By replacing $e_i^0\left|L_i^0\right|$ with $L_i^W(h_j, h_k)$ and estimation of e_i^1 by $\left|L^W(h_j, h_k)\right| / \left|L^C(h_j, h_k)\right|$, we can estimate an upper bound for $\left|L_i^0\right|$ via Eq. (3). That is to say, we can compute an upper bound for $\left|L_i^0\right|$ and replace Eq. (3) by Eq. (10) to estimate the maximum data size of $\left|L_i^1\right|$, in the first round.

$$\left|L_i^0\right| = \left\lceil \frac{e_i^0|L_i^0|}{e_i^1} - 1 \right\rceil = \left\lceil \frac{L_i^W(h_j,h_k) * L^C(h_j,h_k)}{L^W(h_j,h_k)} - 1 \right\rceil \tag{10}$$

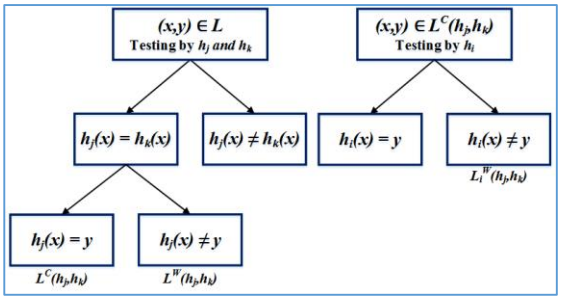

Figure 2 The relationship among Eq. (7), (8), and (9).

3.3 Modification for Co-Labeling

The tri-training algorithm was originally designed for traditional classification. For sequence labeling, we need to define what should be the common labels for the input example x when two models (training time) or three models (testing time) are involved. In Chen et al.'s work, they only consider the most probable label sequence from each model; the selection method chooses examples (for h_i) with the highest-agreement labeled sentences by h_j and h_k, and the lowest-agreement labeled sentences by h_i and h_j; finally, the newly training samples were labeled by h_j (ignoring the label result by h_k).

As the probability for two sequence labelers to output the same label sequence is low (a total of $5^{|l|}$ (BIEOS Tagging) possible label sequences with length l), we propose a different method to resolve this issue. Assume that each model can output the m best label sequences with highest probability (m=5). Let $P_i(y|x)$ denote the probability that an instance x has label y estimated by h_i. We select the label with the largest probability sum by the co-labeling models. In other words, we could use h_j and h_k to estimate possible labels, then choose the label y with the maximum probability sum, $P_j(y|x) + P_k(y|x)$, to re-train h_i. Thus, the set of examples, L_i^t, prepared for h_i in the t-th round is defined as follows:

$$L_i^t = \left\{(x,y): x \in U, \max_y (P_j(y|x) + P_k(y|x)) \geqq \theta * 2\right\} \tag{11}$$

where θ (default 0.5) is a threshold that controls the quality of the training examples provided to h_i.

During testing, the label y for an instance x is determined by three models h_1, h_2 and h_3. We choose the output with the largest probability sum from *3* models with a confidence $\theta * 3$ or $\theta * 2$. If the label with the largest probability sum from *3* models is not greater than $\theta * 3$, then we choose the one with the largest probability from single model with a maximum probability. That is to say, if the label with the largest probability sum from three models is not greater than $\theta * 3$, then we choose the one with the largest probability sum from two models with a confidence of $\theta * 2$. The last selection criterion is the label with the maximum probability estimated by the three models as shown in Eq. (12).

$$y = \max_y \begin{Bmatrix} \max_y (P_1(y|x) + P_2(y|x) + P_3(y|x)) \geq \theta * 3 \\ \max_y \left(P_i(y|x) + P_j(y|x)\right) \geq \theta * 2, i,j \in \{1,2,3\}, i \neq j \\ \max_y (P_1(y|x), P_2(y|x), P_3(y|x)) \end{Bmatrix} \tag{12}$$

4 Experiments

We apply our proposed approach on Chinese personal name extraction. We use known celebrity names to query search engines for news articles from four websites (including Liberty Times, Apple Daily, China Times, and United Daily News) and collect the top 10 search results for sentences that contain the query keyword and uses these query keyword as extraction target via automatic labeling. Given different numbers of personal names, we prepare six datasets by automatically labeling as mentioned in the beginning of Section 3 and consider them as labeled training examples. We also crawl these four news websites from 2013/01/01 to 2013/03/31 and obtain 20,974 articles for unlabeled and testing data. To increase the possibility of containing person names, we select sentences that include some common

Table 1 Labeled dataset (L) and unlabeled dataset (U) for Chinese person name extraction

	L						U
	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Dataset 5	Dataset 6	--
#Names	500	1,000	2,000	3,000	5,000	7,053	--
Sentences	5,548	10,928	21,267	30,653	50,738	67,104	240,994
Words	106,535	208,383	400,111	567,794	913,516	1,188,822	4,251,861

surname followed by some common first name to obtain 240,994 as unlabeled data (U) (Table 1). For testing, we manually labeled 8,672 news articles, yielding a total of 364,685 sentences with 54,449 person names (11,856 distinct person names).

For the tagging scheme, we used BIEOS to mark the named entities to be extracted. Fourteen features were used in the experiment including, common surnames, first names, job titles, numeric tokens, alphabet tokens, punctuation symbol, and common characters in front or behind personal names. The predefined dictionaries contain 486 job titles, 224 surnames, 38,261 first names, and 107 symbols as well as 223 common words in front of and behind person name. We use CRF++ (Kudo 2004) for the following experiment. With a template involving unigram macros and the previous three tokens and behind, a total of 195 features are produced. We define precision, recall and F-measure based on the number personal names as follows:

$$Precision = \frac{Correctly\ identified\ names}{Identified\ names} \quad (13)$$

$$Recall = \frac{Correctly\ identified\ names}{Real\ names} \quad (14)$$

$$F - Measure = 2PR/(P+R) \quad (15)$$

4.1 Performance of Automatic Labeling & Self-Testing

As mentioned above, using the query keyword itself to label the collected news articles (called uni-labeling) only labels a small part of known person names. Therefore, we also use all celebrity names and six report name patterns such as "UDN [reporter name]/Taipei" (聯合報[記者名]/台北報導), to label all collected articles (called Full-labelling). While this automatic labelling procedure does not ensure perfect training data, it provides acceptable labelled training for semi-supervised learning. As shown in Figure 3, the automatic labelling procedure can greatly improve the performance on the testing data.

Based on this basic model, we apply self-testing to filter examples with low confidence and retrain a new model with the set of high confidence examples. The idea is to use the trained CRF model to test the training data themselves and output the conditional probability for the most possible label sequence. By removing examples with low confidence we can retrain a new model with the set of high confidence examples. As indicated by black-dashed line (with + symbol) in Figure 4, the F-measures increases as the data size increases. The performance of self-testing is improved for all datasets with confidence levels from 0.5 to 0.9. An F-measure of 0.815 (Dataset 1) to 0.889 (Dataset 6) can be obtained, depending on the number of celebrity names we have. The best performance is achieved at confidence level 0.8 for all data sets except for dataset 3 which has the best performance when T = 0.9.

4.2 Performance of Tri-Training

Next, we evaluate the effect of using unlabeled training data based on tri-training. In our initial attempt to apply original tri-training, we obtained no improvement for all datasets. As shown in Figure 5, the final data size used for training and the performance is similar to those values obtained for the self-testing results (with confidence level 0.8). This is because we have a very small estimation of $\left|L_i^0\right|$ by Eq. (6) when a 0.5 initial error rate for e_i^0 ($i \in \{1,2,3\}$) is assumed. Therefore, it does not make any improvement on retraining.

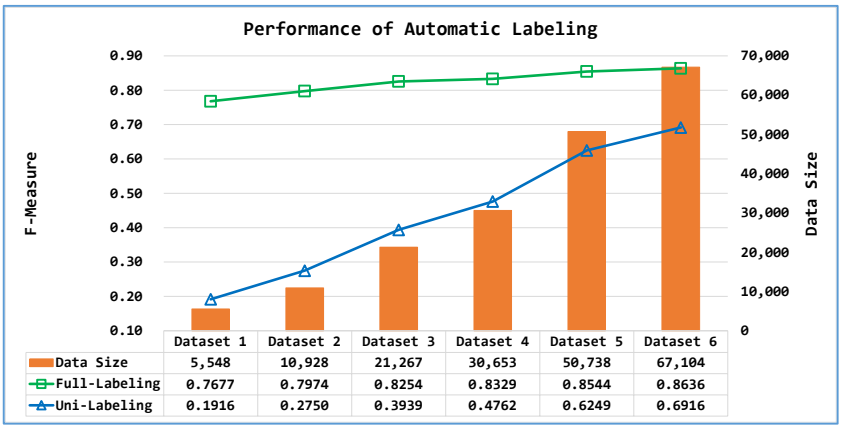

Figure 3 Performance Comparison of automatic labeling

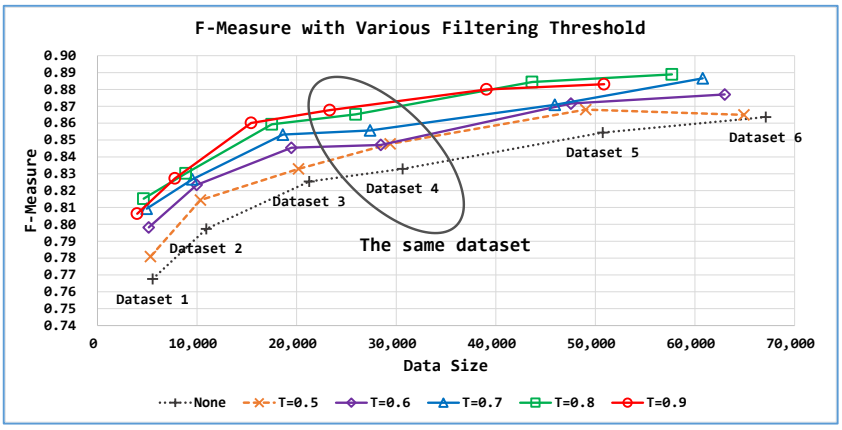

Figure 4 Performance Comparison of self-testing

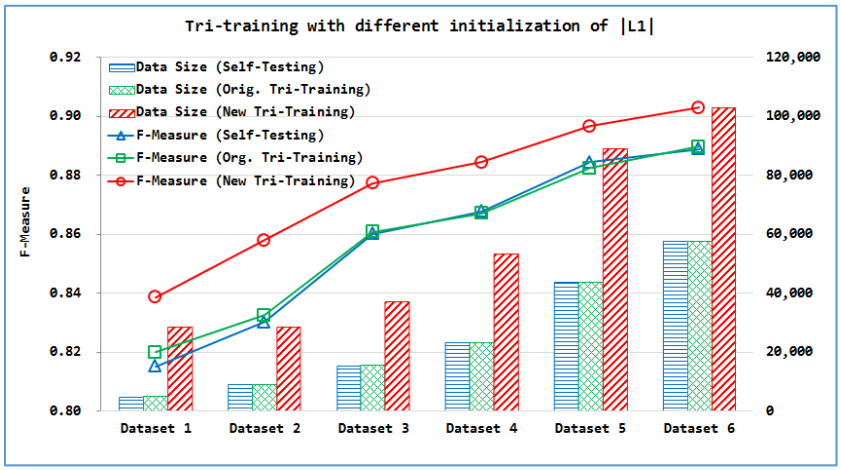

Figure 5 Performance of Tri-training with different initialization for $|L^1|$

However, with the new initialization by Eq. (10), the number of examples that can be sampled from unlabeled dataset $|L_i^1|$ is greatly increased. For dataset 1, the unlabeled data selected is five times the original data size (an increase from 4,637 to 25,234), leading to an improvement of 2.4% in F-measure (from 0.815 to 0.839). For dataset 2, the final data size is twice the original data size (from 8,881 to 26,173) with an F-measure improvement of 2.7% (from 0.830 to 0.857). For dataset 6, since $|L_i^1|$ is too large to be loaded for training with L, we only use 75% for experiment. The improvement in F-measure is 1.5%. Overall, an improvement of 1.2% ~ 2.7% can be obtained with this tri-training algorithm.

5 Conclusion

Named entity extraction has been approached with supervised approaches that require large labeled training examples to achieve good performance. This research makes use of automatic labeling based on known entity names to create a large corpus of labeled training data. While such data may contain noise, the benefit with large labeled training data still is more significant than noise it inherits. In practice, we might have a large amount of unlabeled data. Therefore, we applied tri-training to make use of such unlabeled data and to modify the co-labeling mechanism for sequence labeling to improve the performance. Instead of assuming a constant error rate for the initial error of each classifier, we proposed a new way to estimate the number of examples selected from unlabeled data. As shown in the experiments, such a semi-supervised approach can further improve the F-measure to 0.904 for dataset 6 with 7,000 celebrity names.

Reference

Rie Kubota Ando and Tong Zhang. 2005. A high-performance semi-supervised learning method for text chunking. In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL '05). pp.1-9.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. COLT' 98 Proceedings of the eleventh annual conference on Computational learning theory, pp. 92-100.

Wenliang Chen, Yujie Zhang and Hitoshi Isahara. Chinese Chunking with Tri-training Learning, The 21st International Conference on the Computer Processing of Oriental Languages (ICCPOL2006), LNCS, Vol. 4285, Springer, pp. 466-473, Singapore, Dec. 2006.

Sally Goldman and Yan Zhou. 2000. Enhancing supervised learning with unlabeled data. ICML'00 Proceedings of the 17th International Conference on Machine Learning, pp. 327-334.

Feng Jiao, Shaojun Wang, Chi-Hoon Lee, Russell Greiner, and Dale Schuurmans. 2006. Semi-supervised conditional random fields for improved sequence segmentation and labeling. In Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics (ACL-44), pp. 209-216.

Taku Kudo. CRF++: Yet Another CRF toolkit. http://crfpp.googlecode.com/svn/trunk/doc/index.html

Wei Li, and Andrew McCallum. 2005. Semi-supervised sequence modeling with syntactic topic models. In Proceedings of the National Conference on Artificial Intelligence - Volume 2 (AAAI '05), pp. 813-818.

Gideon S. Mann and Andrew McCallum. 2010. Generalized expectation criteria for semi-supervised learning with weakly labeled data. Journal of machine learning research, Volume 11, pp.955-984.

Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4 (CONLL '03), pp. 188-191.

Kamal Nigam and Rayid Ghani. 2000. Analyzing the effectiveness and applicability of co-training. CIKM '00 Proceedings of the ninth international conference on Information and knowledge management, pp. 86-93.

Cícero Nogueira dos Santos, Ruy Luiz Milidiú. 2012. Named entity recognition. Entropy Guided Transformation Learning: Algorithms and Applications, Springer, Briefs in Computer Science, pp. 51-58.

Erik F. Tjong, Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. In Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4 (CONLL '03), pp. 142-147.

Lei Zheng, Shaojun Wang, Yan Liu, and Chi-Hoon Lee. 2009. Information theoretic regularization for semi-supervised boosting. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '09), pp. 1017-1026.

Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. 2005. Learning from labeled and unlabeled data on a directed graph. In Proceedings of the 22nd international conference on Machine learning (ICML '05), pp. 1036-1043.

Zhi-Hua Zhou and Ming Li. 2005. Tri-Training: Exploiting Unlabeled Data Using Three Classifiers. IEEE Transactions on Knowledge and Data Engineering archive, Volume 17 Issue 11, pp. 1529-1541.