TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol. 12, No. 11, November 2014, pp. 7935 ~ 7945
DOI: 10.11591/telkomnika.v12i11.6481

Winner-Takes-All based Multi-Strategy Learning for Information Extraction

Dwi Hendratmo Widyantoro*, Kurnia Muludi, Kuspriyanto
School of Electrical Engineering & Informatics, Institute of Technology Bandung
*Corresponding author, e-mail: dwi@stei.itb.ac.id

Abstract

The proliferation of information on the Internet has enabled one find any information he/she is looking for. Nevertheless, almost all of these informations are designed for human readers and are not machine readable. Information extraction is a task that addresses the above problem by extracting a piece of information from unstructured formats. This paper proposes a winner-takes-all based multi-strategy learning for information extraction. Unlike the majority of multi-strategy approaches that commonly combine the prediction of all base learnings involved, our approach takes a different strategy by employing only the best, single predictor for a specific information task. The best predictor (among other predictors) is identified during training phase using k-fold cross validation, which is then retrained on the full training set. Empirical evaluation on two benchmarks data sets demonstrates the effectiveness of our strategy. Out of 26 information extraction cases, our strategy outperforms other information extraction algorithms and strategies in 16 cases. The winner-takes-all strategy in general eliminates the difficult situation in multi-strategy learning when the majority of base learners cannot make correct prediction, resulting in incorrect prediction on its output. In such a case, the best predictor with correct prediction in our strategy will take over for the overal prediction.

Keywords*: winner takes all, multi-strategy learning, information extraction*

1. Introduction

The rapid growth of Internet causes textual information become abundance. Until now, Information Retrieval technology is not enough to fulfill a specific information need because this technology only provides information in the level of document collection. Although current state-of-the-art of search engine allows one find relevant documents quickly, a significant effort of cognitive task is still required to scan the documents content in order to extract the information needed. A tool or method that is able to carry out such a task is of great importance to address the information load problem.

Information Extraction is the process of automatically obtaining pre-specified specific information (e.g., events, entity or its relationship) from unstructured data such text documents and web pages [1]. Information Extraction is very useful for many applications such as business intelligence, automatic annotation on web pages, text mining, and knowledge management. The basic task of many information extraction researches has been focused on named entity recognition (NER) and relation extraction in texts. Named entity is a sequence of words that represents a real world entities such as names of person, organization, location etc [2]. Named entity recognition is a task that attempts to identify these entities and then to categorize them into entity class. Meanwhile, relation extraction is a task to detect and identify the semantic relation between named entities, for example, the CEO of a company or CEO(person, company). The recent development of information extraction research addresses the issue of open information extraction, which attempts to discover important relation and entities (without limiting the type of relation or entity).

This paper focuses on the extraction method for NER. Two main approaches that have been well developed for the extraction of named entity include (1) rule-based [3-5] and (2) statistical-based [6, 7] extraction methods. The rule-based approach employs extraction rules, which can be manually crafted or inductively (automatically) generated from training examples through learning process. The majority of NER has been addressed using the rule-based approach [3-5]. The statistical-based methods consider NER as sequence labeling problem in

Received July 14, 2014; Revised September 10, 2014; Accepted September 28, 2014

which each token in the text is labeled either entity or non-entity. Belonging to this approach includes the ones employing Hidden Markov Model, Maximum Entropy Markov Model and Conditional Random Field [6, 7].

Information Extraction can also be viewed as classification problem where text is divided into tokens and classified into related classes [8, 9]. Generally, classification methods require a lot of training examples in order for the methods to be able to generate accurate extraction rules. Each classification method (classifier) typically biases toward a set of assumptions that the classifier uses to make predictions. As a result, no single classifier is superior over diverse information extraction tasks (domains). Researchers address this problem by employing multi classifiers (often termed as multi strategy or hybrid methods), expecting that the weaknesses of a single classifier can be compensated by the strength of other classifiers. The performance of multi classifier methods were reported better over that of single classifier [10-12].

Most of multi classifiers (multi strategy) approaches for information extraction problem make predictions based on the combination of predictions of different classifiers. The differences among them are mainly in the specific method to weigh the significance (confidence) level of each classifier during the calculation of final prediction. Involving multiple classifiers during the classification process can avoid making incorrect prediction when the task is relatively easy for the majority of classifiers. However, it also could prevent the system to produce correct prediction particularly when the task is easy to learn (i.e, can be correctly predicted) only for a few, the minority of classifiers. In the latter case, the final prediction will be incorrect because the incorrect prediction from the majority of classifiers overshadows the correct prediction of the minority one.

In this paper we present our work on addressing the problem encountered by multi classifiers approach when the task is difficult for most classifiers as described above. Rather than involving all classifiers during the classification process, our approach only employs a single classifier for making prediction, selected among other classifiers as the best one for the task during the training phase. This idea is based on the principle that it is better to let the best expert or specialist to perform the task in his/her expertise. To our knowledge, there has been little prior work (if any) that discussed and evaluated this approach, particularly in information extraction task. The main paper contribution is to introduce the use of single best predictor in multi classifier approach (or winner-takes-all strategy) for information extraction. Our experiment on two domains with a total of 26 information tasks demonstrates the effectiveness our approach.

The rest of the paper is organized as follows. Section 2 discusses prior relevant work and shows how their works differ from our proposed method. In Section 3, we describe information extraction as classification problem, which will be used as the main skeleton of extraction process. The detail description of our multi-strategy learning based on winner-takes-all princiople is presented in Section 4. Evaluation of the proposed approach on two domains and the discussion of experiment results is provided in Section 5, followed by concluding remark in Section 6.

2. Related Work on Multi-Srategy Learning and Information Extraction

Multi-strategy learning is an approach that attempts to learn problems with various levels of complexity by employing multiple types of biases and computational paradigms in a learning process. It typically works by running multiple classifiers and makes final decision based on the combination of these classifiers outputs. Assumming the diversity among classifiers in their learning paradigms, how data and noise are interpreted and how the outputs are carefully combined, the performance of multi-classifier is expected to be better than a single classifier.

Multi-classifiers can be categorized into four topological constructions based on the classifiers decision-making steps [13]. The first is conditional topology, a strategy that selects a primary classifier to perform the task of classification and the next classifier is selected only if the earlier fails to identify the new data. The second is hierarchical (serial or selection-based) topology, which employ classifiers in sucsessions, reducing the number of possible classes as it moves from one classifier to the next. The third topology is hybrid, a topology that adopts the strategy for selecting the best classfier based on the given data. Lastly is multiple (parallel or

fusion-based) topology where it first runs all classifiers and all the results are combined. This topology is the most common implementation in a multi-classifiers systems. Prior works using parallel (fusion-based) topology vary greatly in the selection of combinatorial functions, which can be fixed (liner, non-linear or statistical) functions or dynamically trained functions.

The multi-strategy learning with multiple (parallel) topology has been shown useful by Freitag for information extraction problem [11]. In this work, the classifiers are treated as black boxes with only their reliability, which are functions of classifiers' confidences, are considered to model the combiners. The confidence is calculated from the validation set during the training phase. Regression is then applied to map confidence to the probability of correctness. Predictions of a new instance is performed by combining the calculated probabilities of various classifiers to make the best choice. Their combination approach improves over individual classifiers. Similar success on the parallel topology was also reported by Neumann (), which employs two classifiers: (1) Maximum-Entropy Modeling based classifier and (2) tree-based classifier based on Data-Oriented Parsing. Neumann's approach used voting mechanism applied by iterative tag-insertion algorithm.

Similar method in using validation set for estimating the extractor's performance was also employed by Hiyakumoto, Lita & Nyberg [10]. During the training phase, the best extractor for a given extraction task is identified during the training phase. Instead of using fusion-based topology, their extraction process is based on conditional topology. In particular, the best extractor for the task (i.e., identified during the training phase) is applied first. If it fails (i.e., its confidence level below a threshold) the second best extractor will be selected, and so on.

The work of Feilmayr, Vojinovic and Pröll [14] represents a hybrid information extraction approach based on serial topology. The first part is knowledge-based information extraction system that extract enriched features set using manually coded rules. The results of the first part are then fed to the second part, which is inductive-based learner.

Other major information extraction systems that had been developed include LP2 [5], SNOW-IE [15], RAPIER [4], SRV [16] and ELIE [8]. LP2learns symbolic rules for identifying *start tag* and *end tag* class of slot separately [5]. It employs covering algorithm, starting from specific rules and attempts to generalize in order to cover as much positive examples as possible. It performs two-step processes. During the first step, it learns tagging rules using simple bottom-up generalization. Start and end tags are considered positive examples while the rest are negative examples. The second step is to choose the best generalization.

SNOW-IE is Information Extraction System that is based on relational learning algorithm [15]. This system identifiyies text fragment completely without separating *start* tag and *end* tag. This algorithm consist of two steps. First, all posible text fragments are filtered to separate non relavant negative instances. Two criteria are used for filtering: (a) if no general features exists on positive examples, and (b) the confidence value of the fragment is less then a given treshold. Every fragment candidate is represented by using pre-defined features. Features are extracted from three parts; the fragment itself, preceeding fragment, and following fragment part. On the second step, correct fragments are collected from the rest of fragments. The first step results in high recall, while the second one results in high precision.

RAPIER uses *Inductive Logic Programming* to discover extraction rules [4]. Rapier does not separating start tag and end tag, but learn to identify complete relevant string. Bottom-up search is performed through the most specific rule for each example and repeatedly trying to generalize to cover more positive examples. Rapier learns rules of *pre-filler*, *post-filler* and *filler*. Pre-filler tries to match text before target slot and post-filler tries to macth text after target slot. Every pattern is a sequence element that can be matched. RAPIER then proceeds to generalize these rules by selecting pairs of rules and generalizing them by obtaining the least general generalization of each pair of rules. To consider all possible pre- and postfiller patterns would be prohibitive so RAPIER starts generating pre- and post-fillers from the filler outwards. The rules are ordered by Information Gain and weighted by the size of the rule, with small rules being preferred. If a rule has no bad prediction on training examples, it is added to the final rule, replacing any less general rules with worse performances.

SRV employs simple features combination and relational features [16]. Different rule sets are learned for classifying each text fragment as an instance or non-instance of a single attribute value. SRV learns top-down, greedily adding predicates of some predefined types. Rules are validated and their accuracy are estimated using three-fold cross validation; and the three resulting rule sets are then merged. The accuracy estimations are available for each

prediction. An advantage of relational learners is being able to acquire powerful relational rules that cover a larger and more flexible context than most other rule-learning and statistical approaches. The downside is that the large space of possible rules can lead to high training times and there is no guarantee of finding optimal rules (local maxima problem).

The ELIE system adopts Support Vector Machines (SVMs) for *Begin*/*End* tagging [8]. Highly improved results are achieved by augmenting this setup with a second level (L2) of *begin*/*end* classifiers. The L2 end classifier focuses on finding suitable *end* tags for matching left-over *begin* tags from the first-level (L1) begin classifier, and the L2 begin classifier matches left-over end tags. While the L1 classifiers are trained on a very high number of tokens, almost all of which are negative instances (O), the L2 classifiers only consider the near context of left-over L1 begin/end tags which allows a more focused classification. Hence the L1 classifiers must be tuned to favor precision over recall to avoid producing lots of false positives (spurious extractions) from all over text, but the L2 classifiers can be tuned to favor recall over precision since they only classify a very small subset of all the tokens. In this way, by adding the second level the recall of the overall system can be increased without overly hurting the precision.

3. Token Classification-based Information Extraction

Information Extraction (IE) in this paper is cast as token classification problem. In contrast to text classification that attempts to categorize the entire text, the token classification predicts whether a token is a member of a pre-specified information need (extraction slot). Casting IE as token classification problem has an advantage in that it has the flexibility for selecting the variety of classification methods, which have already been well studied.

In classification-based IE, texts are split into tokens. Each token is encoded with features and is labeled with its class (either belonging to an extraction slot or not) for the learning process. Any classification algorithm can then be applied to create the classification model of each extraction slot. Given a new text, the extraction process is performed by identifying the classes of tokens and filling in extraction slots from a sequence of tokens with the same classes.

In this paper, we adopt the classification-based IE developed by Finn [8] and Siefkess [9]. For a given extraction tasks, the system employs two classifiers: (1) one identifies the start token of fragments and (2) another identifies the end token of fragments. The system extracts fragment s from the start token to the closest following end token. The two classfiers, during the training phase, create two models of begin and end tokens independently.

Because a token itself is not enough for learning the token classification, an instance is described as a token feature vector. Therefore, each instance is labeled as either *start tag* or not for learning the model of start token, and it is labeled as *end tag* or not for modelling the end token. The feature vector of a token in this paper consists of *window size*, *Part of Speech* (POS), *Ortography*, *Tokenkind*, *Lemma* and *Lookup*. The window size determines the number of tokens before and after the current token that will be considered. Each token is also tagged with its part of speech (e.g., Noun, Verb, etc). The ortography features include whether the token is capitalized, upper-case or lower-case. *Tokenkind* describes whether the token is a word, numeric, symbol or punctuation. *Lemma* is the basic form of token as a result of morphological analysis. Lookup provides a value associated with the token, based on user-defined dictionary. It contains the list of city names, country names, first names, and last names. For example, a "bandung" token could be assigned a value "city_name".

Figure 1 describes the learning and extraction process of classification-based information extraction. The learning process requires two sets of training data for learning *start_tag* model and *end_tag* models, respectively. The training data are sets of token feature vector fv_t and their label. The LEARN_MODEL in the figure implements any inductive learning algorithm for classification task (i.e., Naïve Bayes, Decision Tree, Nearest Neighbour, etc.). The extraction process consists of two steps. The first step is to classify all tokens as either a *start_tag*, an *end_tag* or none of them. In the second step, for each identified *star_tag* token, it will extract a sequence of tokens from the *start_tag* token to the closest following token that is an *end_tag*.

Learn_Extraction_Model (L, *TD*)

Input: Lis an inductive learning algorithm.
TD= Training_data$_{start\_tag}$&*Training_data*$_{end\_tag}$
/* *Training_data*$_{start\_tag}$←{$\langle fv_t, label\rangle$}where $label = \begin{cases} +, a\ start\ tag \\ - \end{cases}$
/* *Training_data*$_{end\_tag}$←{$\langle fv_t, label\rangle$}where $label = \begin{cases} +, a\ end\ tag \\ - \end{cases}$

start_tag_model←LEARN_MODEL (L, *Training_data*$_{start\_tag}$)
end_tag_model←LEARN_MODEL (L, *Training_data*$_{end\_tag}$)

Output: (*start_tag_model*, *end_tag_model*)

Extract (L, *Extraction_Model, D*)

Input: Lis an inductive learning algorithm.
Extraction_Model = (*start_tag_model*, *end_tag_model*)
D = {(fv_t)}is a document

/* ***First stage***: classification of each token whether or not belong to *start_tag*, *end_tag*
for each*token t* **in***document D* **do**:
start_tag_label$_t$←CLASSIFY(L, *start_tag_model*, fv_t)
end_tag_label$_t$←CLASSIFY(L, *end_tag_model*, fv_t)
end for

/* ***Second stage***: extracting phrase between (including) tokens of start and end tags
extraction_result = {}
for each *start_tag_label*$_t$ == '+' **do**
i ←Position (*t*) // *t* is i-*th* token in D
j ←**min**Position (*t*) where *end_tag_label*$_t$ = '+' **and** Position (*t*) ≥ i
extraction = sequence of token from i-*th* to j-*th* position.
Result ← *Result* ∪*extraction*
end for

Output: *Result*

Figure 1. Learning and Extraction Process of Classification-based Information Extraction

4. Winner-Takes-All Multi-Strategy Learning for Information Extraction

In addition to employing multiple learning algorithms with diverse learning paradigms (biases) during the training phase, most multi-strategy approaches also use multiple models during the prediction phase by combining the models' outputs in one way or another. This approach will generally work well when the majority of base-learners are able to correctly learn the underlying concepts. However, this is not the case when the problem is difficult for most of the base-learners such that they mostly incorrecly predict the outputs, causing a final prediction that will be likely incorrect.

Our strategy to address the above observations is to use a single, best learner during the prediction process. If a learner is identified as the best one for a particular task (among other learners) during the training phase, then it is better to use that best learner for predicting that particular task (i.e, the winner takes all). Therefore, other learners that do not perform well on the task will not interfere the prediction of the best one. As an extreme example, consider a task of extracting "date" field. A learner that learns the regular expression of various date

formats will likely become the best learner, compared to other learners such as Bayesian and Nearest Neighbor learners. With majority voting, the correct prediction of reguler expression learner could be easily defeated by the other two learners that might ocassionally make incorrect prediction. This will not be the case if the best (reguler expression) learner is consistently employed for extracting "date" field.

Given a set of base learners **L** = {L_1, L_2, ..., L_n},and a set of information extraction tasks **S** = {s_1, s_2, ..., s_m}where s_i is an information slot to be extracted, our strategy can be outlined as follows:

a) Find the best learner L_i in **L** for each extraction task *s* in **S**.
b) Re-train the best learner for each extraction task using full training data.
c) For extraction task *s*, apply only the best learner (as identified in Step 1 and re-trained in Step 2) for that task.

FIND_BEST_LEARNER (**L**, *s*, *TD*S)

Input: A set of base learners **L** = {L_1, L_2, ..., L_n},
s is a specified information task ,
TD_S is training data for information task s.

Max_accuracy← 0;
Best_Learner← null;
for each L_i **in** **Ldo**:
    accuracy←PERFORMANCE (L_i, TD_S)
    if (*accuracy*> Max_*accuracy*)
        Max_accuracy←*accuracy*
        Best_Learner←L_i
end for

Output: *Best_Learner*

Figure 2. Finding the Best Learning Algorithm for a Specific Extraction Task

Figure 2 describes the algorithm for finding the best learner for a specific extraction task. Given a specified information extraction task (e.g., name of city), training data for the task, and a set of inductive learning algorithms, it measures the performance of each inductive learning algorithm on the same training data and select the one with the best accuracy (the winner). The function PERFORMANCE employs *k*-fold cross validation in order to measure the accuracy of learning algorithm. *k*-fold cross validation is an experimental method that divides training data into *k* partitions, and run *k* experiments where in each i^{th} experiment, the i^{th} partition is used as the validation set while the rest of partitions for training set. The inductive learning algorithm is trained using training set and its accuracy is measured using the validation set. The final accuracy is averaged over the *k* trials.

The detail of our extraction strategy is depicted by Figure 3. During the training phase, it first identifies the best inductive learning algorithm for each extraction task (i.e., stored in variable *Best_Learner*). The extraction model of each extraction task is then re-learned using the full training data using the best inductive learning algorithm for the task. Finally, for a given information task, it will use only the best extraction model for the extraction process.

/* Training Phase

Input: Learners **L** = *{L1, L 2, ..., L n}*,
Information Extraction Tasks *S = {s1, s2, ..., sm}*,
Training data sets*TD= {TDs1, TDs2, ...,TDsm}.*

/* Find the best learner for each task
for each*s* **in***S***do**:
Best_Learner [*s*] ←**FIND_BEST_LEARNER** (*L,s, TDs)*
end for

/* Re-learn the extraction model for each task
for each*s* **in***S***do**:
Extraction_Model [*s*] = **LEARN_EXTRACTION_MODEL** (*Best_Learner* [*s*],*TDs)*
end for

Output: (*Best_Learner*, *Extraction_Model*)

/* Extraction Phase

Input: Information Extraction Tasks *S = {s1, s2, ..., sm}*
a document*D* = $\{(fv_t)\}$

for each*s***in***S***do**:
Extraction_Result [s] = **EXTRACT** (*Best_Learner* [s], *Extraction_Model*[s], D)
end for

Output: *Extraction_Result*

Figure 3. Multi-Inductive Learning Strategy for Information Extraction Task

5. Evaluation

This section describes our experiments to evaluate the performance of our proposed strategy in comparison with other well know information extraction methods as well as methods based on multi-strategy learning. We employ *F-measure* (van Rijsbergen, 1975) as the performance measure and we treat the precision and recall in the F-measure equally as follows:

$$F\_measure = \frac{2 . precision . recall}{precision + recall} \quad (1)$$

Precision is a portion of correct extraction over all extracted fragments, while recall measures a portion of correct extraction over all relevan fragments. High precision generally causes low recall, and vice versa. The *F-measure* provides the balance between precision and recall.

5.1. Data Set

For the evaluation we consider two benchmark data sets commonly used for information extraction task. The first data set is *Reuters Corporate Acquisition* (Freitag, 1998b). It consists of 600 articles retrieved from *Reuter Newswires* covering news about corporate acquisitions. The second data set is Job Posting, which consists of 300 newsgroup message about job vacany in Austin, Texas. Table 1 & 2 provide the summary of the data sets used in the experiments.

Table 1. Summary of *Corporate Acquitition* Data Set

No	IE Tasks	#Event	Example
1	acqabr	1450	*Norcros*
2	acqloc	213	*Southern California*
3	acquired	683	*Norcross Plc*
4	dlramt	283	*542.2 MLN STG, almost a billion dlrs, not disclo*
5	purchabr	1263	*Sahlen, Sahlen and Associates*
6	purchaser	624	*Sahlen and Associated Inc*
7	seller	267	*CSR Ltd*
8	sellerabr	431	*CSR*
9	status	461	*proposed, approved, agreed in principle*

Table 2. Summary of *Job Posting* Data Set

No	IE Tasks	#Event	Exmples
1	application	605	*DB2, Oracle, DB2 server, sybase*
2	area	980	*Failure analysis, multimedia, TCP/IP, internet*
3	city	639	*Austin, Battle Creek, San Antonio*
4	company	291	*Alliance, CPS, Charter Professional Services Inc*
5	country	363	*US, USA, England, UK*
6	desired_degree	21	*PhD, BS, BSCS, Masters, MSCS*
7	desired_years_experience	45	*5, 4, 10*
8	id	299	*NEWTNews.872347949.11738.consults.ws-n*
9	language	867	*RPG, COBOL, CICS, Java, c, c++, SQL*
10	platform	705	*AS400, Windows 95, windows, portable system, PC*
11	post_date	288	*30 Aug 1997, 11 Sep 1997*
12	recruiter	325	*Resource Spectrum*
13	req_degree	80	*BS, B.S., Bachelor, Bachelor's, BSCS*
14	req_years_experience	173	*2, 2+, Two, 5, 4*
15	salary	143	*$50k to $70k, to $60k*
16	state	462	*TX, Texas, Miami, Georgia, MI*
17	title	466	*ALC Application Programmer, Visual Basic Developers*

5.2. Setup of Experiments

As described in earlier section, our multi-strategy approach requires several base inductive learning algoritms from which to select the best learning algorithm for a specified information task. In this experiment we incorporate Perceptron Algorithm with Uneven Margin (PA) [18], Support Vector Machine (SVM), Averaged One-Dependence Estimators (AODE) [19] and *k*-Nearest Neighbour (*k*-NN) [20] as the base learners, which provides diverse learning biases.

For performance comparison with other existing multi-strategy learnings, we run experiments using Voting and Bagging algorithms. Voting algorithm has been commonly used as the baseline for combining classifiers by averaging the probability estimates over the combined classifiers [21]. The implementation of Voting algorithm is based on Kittler et. Al [22] and Kuncheva [23]. We use the same base learners for Voting algorithm as in our multi-strategy approach. The Bagging algorithm is implemented as in Breiman [24]. It poses a meta-learning algorithm where a number of classification iteration is performed only by a single classifier in order to reduce variance. Its prediction is based on the average of probability estimates. AODE is employed as the classifier of Bagging algorithm. Additionally, we also compare the

performance of WTA-based multi-strategy approach with well known information extraction algorithms, including RAPIER [4], SRV [16], ELIE [8], LP2 [5], SNOW-IE [15].

5.3. Results

Table 3 shows the performance comparison on*Reuture's Corporate Acquition*data set. Out of nine information tasks, our WTA-based multi-strategy approach outperforms other algorithms in seven cases (*acqabr,acqloc,acquired, dlramt, purchabr, purchaser*, and *status*). The best results of the other cases (*seller* and *sellerabr*) are provided by the SRV algorithm. The average performance of our strategy (0.46) is also significantly higher than the others. The dominant base learners employed in our multi-strategy approach are SVM and PA inductive learning algorithms. *Seller* and *sellerabr* are among the most difficult information extraction tasks since all algorithms and strategies suffer from low *F-measure* on these tasks.

Table 3. Performance Comparison on *Reuters Corporate Acquisition* Data Set

	RAPIER [4]	SRV[16]	ELIE[8]	Mullti-Strategy Approach		**WTA-based Multi-Strategy Approach**	
				Voting	Bagging		
acqabr	0.26	0.38	0.40	0.31	0.24	**0.57**	(SVM)
acqloc	0.24	0.22	0.34	0.05	0.25	**0.47**	(SVM)
acquired	0.29	0.39	0.43	0.06	0.22	**0.51**	(SVM)
dlramt	0.39	0.62	0.59	0.09	0.34	**0.65**	(PA)
purchabr	0.24	0.48	0.29	0.28	0.39	**0.49**	(SVM)
purchaser	0.28	0.45	0.46	0.22	0.33	**0.52**	(PA)
seller	0.15	**0.23**	0.16	0.00	0.23	0.22	(SVM)
sellerabr	0.09	**0.25**	0.13	0.00	0.24	0.21	(SVM)
status	0.41	0.47	0.50	0.12	0.25	**0.53**	(PA)
Average	0.27	0.41	0.39	0.13	0.28	**0.46**	

Table 4 depicts the performance comparison on*Job Posting* data set. Although not as dominant as in the *Corporate Acquisition* data set, our WTA-based multi-strategy algorithm achieves highest performance on nine cases (*city, company, desired_degree, platform, recruiter, req_degree, salary, state,* and *title*) out of 17 cases. The highest performance on other cases are achieved by RAPIER (2 cases), LP2 (5 cases) and SNOW-IE (2 case). The best performance on past_date is shared by RAPIER, LP2, SNOW-IE and our strategy since these methods produce the same *F-measure* (0.99). Similar to *Corporate Acquisition* data set, the average performance of our multi-strategy learning (0.81) in *Job Posting* data set is significantly higher than the other methods.

Table 4. Performance Comparison on *Job Posting* Data Set

	RAPIER[4]	LP2[5]	SNOW-IE[15]	Mullti-Strategy Approach		**WTA-based Multi-Strategy Approach**	
				Voting	Baging		
application	0.69	**0.78**	0.61	0.17	0.20	0.74	(PA)
Area	0.42	**0.67**	0.52	0.06	0.08	0.57	(PA)
City	0.90	0.93	0.89	0.74	0.50	**0.95**	**(SVM)**
Company	0.70	0.72	0.75	0.62	0.34	**0.82**	**(PA)**
Country	0.93	0.81	**0.95**	0.64	0.57	0.59	(PA)
desired_degree	0.72	0.65	0.61	0.14	0.06	**0.74**	**(PA)**
desired_years _experience	**0.87**	0.60	0.79	0.71	0.79	0.86	(SVM)
Id	0.97	**1.00**	0.99	0.73	0.57	0.99	(SVM)
Language	0.81	**0.91**	0.82	0.37	0.39	0.88	(PA)
Platform	0.72	0.80	0.74	0.33	0.30	**0.82**	**(PA)**
post_date	**0.99**	**0.99**	**0.99**	0.97	0.97	**0.99**	(SVM)
Recruiter	0.68	0.81	0.85	0.57	0.59	**0.87**	**(PA)**
req_degree	0.81	0.85	0.83	0.48	0.37	**0.86**	**(PA)**
req_years _experience	0.67	0.69	**0.84**	0.63	0.62	0.81	(SVM)
Salary	0.67	0.63	0.73	0.49	0.25	**0.84**	**(PA)**
State	0.90	0.85	**0.92**	0.61	0.40	**0.92**	**(SVM)**
Title	0.40	0.44	0.53	0.27	0.16	**0.69**	**(SVM)**
Average	0.75	0.77	0.79	0.50	0.42	**82,1**	

6. Conclusion

This paper has described our winner-takes-all based multi-strategy learning for information extraction, which is cast as classification problem. Unlike multi-strategy learning that generally combines the predictions of various learning algorithms, our strategy simply employs the best learning algorithm for a specific information task, selected among other learning algorithms with varying biases. Our approach has an advantage in a difficult situation when only a single or minority of learning algorithms can make correct predictions. In such a situation, the majority of learning algorithms with incorrect predictions will outweigh the final prediction, which is not the case in our winner-takes-all based multi-strategy learning. Nevertheless, our approach also has a shortcoming in that it requires more computational effort in selecting the best learning algorithms since it has to evaluate the performance of all base learners during the training phase. Empirical evaluation on two benchmark data sets (*Reuters Corporate Acquisition* and *Job Posting*) reveals that our strategy is quite effective. In both data sets, our approach outperforms the other approaches in term of F-measure.

References

[1] Finn A, Kushmeric N. *Information Extraction by Convergent Boundary Classification*. Proceedings of the AAAI Workshop on Adaptive Text Extraction and Mining. 2004.
[2] Aggarwal CC, Zhai C. Mining Text Data. Springer. 2012.
[3] Sodderland S. Learning information extraction rules for semi-structured and free text. *Machine Learning*. 1999; 34 (1-3): 233-272.
[4] Califf ME, Mooney RJ. *Relational Learning of Pattern-Match Rules for Information Extraction*. Proceedings of the Sixteenth National Conference on Artificial Intelligence and Eleventh Conference on Innovative Applications of Artificial Intelligence. Orlando, FL. 1999: 328-334.
[5] Ciravegna F. *(LP)2, an adaptive algorithm for information extraction from Web-related texts*. Proceedings of IJCAI-2001 Workshop on Adaptive Text Extraction and Mining. Seattle, USA. 2001.
[6] McCallum A, Freitag D, Pereira FC. *Maximum Entropy Markow Models for Information Extraction and Segmentation*. Proceedings of the 17th International Conference on Machine Learning. 2000: 591-598.
[7] Sarawagi S, Cohen WW. *Semi-Markov Conditional Random Fields for Information Extraction*. Advances in Neural Information Processing Systems. 2004: 1185-1192.
[8] Finn A. A Multi-Level Boundary Classification Approach to Information Extraction. Ph.D. Thesis. Dublin: University College Dublin; 2006.
[9] Siefkess C. An Incrementally Trainable Statistical Approach to Information Extraction based on Token Classification and Rich Content Models. Ph.D. Thesis. Berlin, Freie Universitat Berlin; 2007.
[10] Hiyakumoto L, Lita LV, Nyberg E. *Multi-Strategy Information Extraction for Question Answering*. Proceedings of FLAIRS Conference. 2005: 678-683.
[11] Freitag D. *Multistrategy Learning for Information Extraction*.Proceedings of International Conference of Machine Learning. 1998: 161-169.
[12] Sigletos G, Paliouras G, Spyropoulos CD, Stamatopoulos T. *Meta-learning beyond Classification: A Framework for Information Extraction from the Web*. International Workshop & Tutorial on Adaptive Text Extraction and Mining held in conjunction with the 14th European Conference on Machine Learning. 2003: 58.
[13] Lam L. Classifier Combinations: Implementations and Theoretical Issues. *Multiple Classifier Systems*. Springer Berlin Heidelberg. 2000: 77-86.
[14] Feilmayr C, Vojinovic K, Proll B. *Designing a Multi-dimensional Space for Hybrid Information Extraction.* Database and Expert Systems Applications (DEXA), 23rd International Workshop on. IEEE. 2012: 121-125.
[15] Roth D, Yih WT. *Relational Learning via Propositional Algorithms: An Information Extraction Case Study*. Proceedings of International Joint Conference on Artificial Intelligence. 2001; 17(1): 1257-1263.
[16] Freitag D. *Toward General-Purpose Learning for Information Extraction*. Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics. San Francisco, CA. 1998; 1: 404-408.
[17] Van Rijsbergen CJ. Information Retrieval. Second Edition. Butterworth-Heinemann. 1979.
[18] Li Y, Zaragoza H, Herbrich R, Shawe-Taylor J, Kandola JS. *Perceptron Algorithm with Uneven Margin*. Proceedings of International Conference on Machine Learning. 2002; 2: 379-386.
[19] Webb GI, Boughton JR, Wang Z. Not So Naïve Bayes: Aggregating One-Dependence Estimator. *Machine Learning*. 2005; 58(1): 5-24.
[20] Aha DW, Kibler D, Albert MK. Instance-based Learning Algorithms. *Machine Learning*. 1991; 6(1): 37-66.

[21] Witten IH, Frank E, Hall MA. Data Mining. Practical Machine Learning Tools and Techniques. Third Edition. Morgan Kauffman. 2011.
[22] Kittler J, Hatef M, Duin RP, Matas J. On Combining Classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1998; 20(3): 226-239.
[23] Kuncheva LI, Whitaker CJ, Shipp CA, Duin RP. Limits on the Majority Vote Accuracy in Classifier Fusion. *Pattern Analysis & Applications*. 2003; 6(1): 22-31.
[24] Breiman L. Bagging Predictors. *Machine Learning*. 1996; 24(2): 123-140.