2011 International Conference on Electrical Engineering and Informatics
17-19 July 2011, Bandung, Indonesia

Multi-Inductive Learning Approach for Information Extraction

Kurnia Muludi[#1], Dwi H Widyantoro[*2], Kuspriyanto[*3], Oerip S. Santoso[*4],

[#] *Computer Center, The University of Lampung*
Jl. Sumantri Brojonegoro 1, Bandar Lampung 35145, Indonesia
[1] kurnia@unila.ac.id

[*] *School of Electrical Engineering and Informatics – Bandung Institute of Technology*
Jl. Ganeca 10, Bandung, Indonesia
[2] dwi@if.itb.ac.id
[3] kuspriyanto@yahoo.com
[4] pireo46@yahoo.com

Abstract— **The vast amount of information in the Internet is not easy to find and use. Information Extraction technology is one of alternatives that can solve this problem. Conventional Natural Language Processing approach is hampered by its portability, scalability and adaptability. Introduction of Machine Learning into Information Extraction is one of solutions. Inductive Learning only needs annotated training examples. The problem is there is no performance consistency of algorithms on various information domains. Automatic and smart classifier selection from various machine learning algorithms is one of the best way to handle this problem. The goal of this paper is to propose a method for Information Extraction System based on Inductive Learning and Meta Learning that have good performance. In this paper Multi-Inductive Learning is developed to answer that question. Multi-Inductive Learning is consist of several Inductive Learning algorithms that have significant difference in their mechanism. This is to ensure there is bias variance in this method. Through k-fold cross validation on training document, Multi-Inductive Learning algorithm can choose the best classifier for each slot on a certain domain. These best classifiers then employ to do full extraction on testing document. The conducted experiment shows that Multi-Inductive Learning has better performance than that of single Inductive Learning algorithm-based Information Extraction systems. On Reuters Corporate Acquisition, Multi-Inductive Learning gives a score of 46.3 % and has the best performance among other state of the art information systems. Out of nine slots that should be extracted, six of them give the best performance. Multi-Inductive Learning also gives better performance on Job Posting dataset. Average performance of it gives 82.1 % and is the best among other state of the art of Information Extraction. Out of 17 slots that should be tested, nine of them are extracted with the best performance.**

Keywords— ***Information Extraction*, *inductive learning*, *meta learning, multi inductive learning.***

I. INTRODUCTION

The very fast internet growth causes textual information become abundance. Until now Information Retrieval technology is not enough to fulfill the specific information need because this technology only provides information in the level of document collection. Tools and smart methods development that can access document content are crucial issues on Knowledge Management.

Information Extraction is the process to get information about *pre-specified events*, entity or relationships in the text like newswire and web pages. Many information extraction researches focus on entity recognition which is a basic task. In general, Information Extraction task can be regarded as information entity recognition task in the text. Information Extraction is very useful for many applications such as business intelligence, automatic annotation on web pages, text mining, and knowledge management.

Information Extraction can be approached as classification problem where text is divided into tokens and classified into related classes. Generally, classification methods need a lot of training examples in order the method to be able to generate extraction rules. The problem is there is no single classifier performs constantly among domains.

In this paper we will discuss how multi classifier approach can perform better than single classifier one on Information Extraction.

II. RELATED WORK

A. State of The Art of Information Extraction

LP2 [3] learning by using symbolic rules for identifying *start tag* and *end tag* class of slot. LP2 identifies start tag and end tag separately. Besides using token features and orthographic, it uses linguistic information such as morphology and POS, and user-defined dictionary or gazetteer. This learning algorithm is covering algorithm which start from specific rules and tries to generalize in order to cover as much as positive examples. This process is strengthened by correcting error that show up. This process is done in two steps. First, simple bottom-up generalization is done for learning tagging rules. Second, learning correction rules for diminishing the error made by tagging rules.

In the first step, learning tagging rules set, each rule is used for identifying either start or end tag of information fragment. LP2 approach is token classification where *start* and *end* fragment are positive example where the rest are negative examples. For each positive example is treated in the following steps. First, create initial rule; second, generalize rule; third, take k-best generalization of rules and throws the rest.

The next step is to choose the best generalization. K-best generalizations have (a) better accuracy, (b) cover more positive examples, (c) cover other part of input, and (d) have error rate less than a given treshold. Rules that are not included in this step then added into best rules pool. Instances that already covered by this pool are then removed from positive examples. Once an instance have been covered by the rule, this instance will never be included in the rule induction process. Initial rule set tends to have high precision but low recall. In this phase, recall is improve through learning using contextual rules.

SNoW-IE [12] is Information Extraction System that based on relational learning algorithm. This system identifiyies text fragment completely without separating start tag and end tag. SNoW-IE have token, orthographic, POS and semantic features. This algorithm consist of two steps. First, all posible text fragments are filtered. This is for the purpose of separating non relavant negative instance. Two criterias are used, (a) if there is no general features on positive examples, and (b) the confidence value of the fragment is less then the given treshold. The first step results in high recall, while the second one results in high precision. SNoW-IE is based on *relational learning* in form of *Inductive Logic Programming* (ILP). Every fragment candidate is represented by using pre-defined features. Features are extracted from three parts; the fragment itself, preceeding part of the fragment, and after fragment part. On the second step, correct fragments are collected from the rest of fragments.

Rapier [2] uses *Inductive Logic Programming* to discover extraction rules. Rapier does not separating start tag and end tag, but learn to identify complete relevant string. Bottom-up search is done through the most specific for each example and repeatedly trying to generalize to cover more positive examples. Rapier uses token, POS and semantic features. Rapier uses different representation from other systems. It uses template filling, so it does not use tagging in the text. Each template is filled by slot that asscociated to relevan text. This approach does not accomodate slot apearance in the text and it does not tolerate ambigue text. As an example on job advertisement corpus can have template 'platforms: windows'. This approach prevents the word of 'windows' in the text for other context other than 'platforms'. Rapier's algorithm tries to fill the template and it searchs from specific to general.

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 sequence element that can be matched. Rapier then proceeds to generalize these rules by selecting pairs of rules and generalizing them by getting 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. It maintains a list of the *k* best rules and repeatedly adds generalizations of the pre- and post-filler seed rules, working outward from the filler. The rules are ordered by Information Gain and weighted by the size of the rule, with small rules being preferred. When a rule gives no bad predictions on the training examples it is added to the final rule-base replacing any less general rules that it performs worst.

SRV [6] uses simple features combination (such as world length, kind of character, POS) and relational features (mapping a token to another token, e.g. next-token, subject-verb). Feature values can be sets, e.g. all synonyms and hypernyms (super ordinate concepts) listed by WordNet are combined in a set for each token. Different rule sets are learned for classifying each text fragment as an instance or non-instance of a single attribute value; there is no component for template unification or other post processing. SRV learns top-down, greedily adding predicates of some predefined types: the number of tokens in the fragment (length), whether a condition is matched by one or several (some) or by all (every) tokens in the fragment; *position* specifies the position of a token in a some predicate, *relpos* constrains the ordering and distance between two tokens. Rules are validated and their accuracy estimated by three-fold cross validation; the three resulting rule sets are merged. The accuracy estimations are available for each prediction. An advantage of relational learners is their 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 [5] uses Support Vector Machines (SVMs) for Begin/End tagging. Highly improved results are reached 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.

B. Meta-Learning

Meta-learning learn how learning system can improve its efficiency through experience. The purpose is how to make learning process can be flexible to domain or task that is handled [16]. All learning systems work through adaptation to the specific environment that have implication to partial ordering or bias to the set of all posible hypotheses explaining concept [9].

Meta-learning is different from base-learning in the scope of its adaptation level: Meta-learning studies how to choose bias dinamically contrast to base learner where bias is a priori or user parameterized [16]. For example on inductive learning scenario (e.g decision tree, SVM, etc) over some data produces a hypothesis that depends on the fixed bias embbeded in the learner. Learning takes place at the base-level and the quality of hypothesis normally improves with an increasing number of examples. Nonetheless, successive applications of the learner over the same data always produces the same hypothesis, independently of performance; no knowledge is extracted across domains or tasks [11]. Meta-learning in this case, aims to discover ways to dynamically search for the best learning strategy as the number of tasks increases [13]. A computer program qualifies as a learning machine if its performance improve with experience [10]. According to [16] experience is knowledge gained from the analysis of several tasks. Meta-learning is focused on the need of learner to adapt continually on several level abstractions. Learning in this case is not on the base level but also across task (meta) level. Several areas of study related to meta-learning are building meta-learner of base-learners [17], selecting inductive bias dynamically [4] building meta-rules matching task properties with algorithm performance [1], inductive transfer [11] and learning to learn [13].

On Building meta learner from base learner, a set of q base learners are applied to a training set $T_{train}: \{(\tilde{X}i, c_i)\}_{i=1}^{m}$ to produce q hypotheses, $\{h_j\}_{j=1}^{q}$, also called level-0 generalizers.Meta-learning takes place when training set T_{train} is redefined into a new set T'_{train}. The redefinition replaces each vector $\widetilde{\mathrm{X}}_i$ with the class predicted by each of the q hypothesis on $\widetilde{\mathrm{X}}_i$:

$$T'_{train} = \{(\widetilde{\mathrm{X}}'_i, c_i)\} = \{((h_1(\widetilde{\mathrm{X}}_i), h_2(\widetilde{\mathrm{X}}_i), \dots, h_q(\widetilde{\mathrm{X}}_i)), , c_i)\}$$

The new training set T'_{train} serves as input to a set of meta-learners, which produce a new set of hypotheses.

Dynamic selection of bias enables a learning algorithm to shift region of expertise along the tasks. The goal is to change hypothesis space to have better coverage of the task under analysis. During dynamic bias selection, meta-learning is a required component and is often acting as a guideline in the search over the bias space. [4] develop a framework for the study of dynamic bias as a search in three different tiers. In the first tier, searching over a hypotesis space $\mathcal{H}_{\mathcal{L}}$ where a learning algorithm L looks for the best hypothesis approximating the target concept (most learning algorithms assume this space fixed). For dynamic bias selection to take place, a learning algorithm L must search in a second tier, where the strength and size of $\mathcal{H}_{\mathcal{L}}$ can be modified separately. Modification of the meta-spaces defined in the second tier is done in the third tier. The problem can arise here is where to stop building more tiers (i.e. more met-meta-spaces).

One important property of meta-learning is to provide guidelines of how to relate a learning algorithm with those domains in which the algorithm performs well. The general approach is through defining a set of domain characteristics or meta-features that relevant to the performance of a learning algorithm; those meta-features enable us to build a meta-domain T_{meta} relating domain characteristics with algorithm performance (once a sufficient number of domain has been analyzed). A set of rules finally can be induced using meta-learner over T_{meta} to discover the conditions under wich a learning algoritm outperforms others.

Learning is not an isolated task that starts from zero every time a new problem domain appears. With experience accumulation, a learning mechanism is expected to perform increasingly better. For learning to improve through time, meta-knowledge must be transferred across domains or tasks. The process is known as inductive tansfer [11]. [14] propose a learning algorithm where domains are clustered when mutually related. A new domain is assigned to the most related cluster; inductive transfer takes place when generalization exploits information about the selected cluster. Further [15] propose a learning algorithm where domains are clustered when mutually related. A new domain is assigned to the most related cluster; inductive transfer takes place when generalization exploits information about the selected cluster.

[14] propose general framework to differenciate between learning at base-level and meta-level. In the base-level simply tries to find the correct hypothesis h on a fixed hypothesis space $\{H\}$.

[8] propose Learning Classifier System which is a parallel, message-passing, rule-based system. Each message or rule is a condition-action pair; if a message matches the condition part, the rule is candidate to activate and execute the action part. The system assumes an input interface or set of detectors that translates signal from an external environment into messages.

C. Meta-Learning and Information Extraction

Meta-learning implementation in Information Extraction is done by [7] This system scheme is depict in Figure 1. In this system, learners are considered as black boxes and only its reliability as a function of modeled confidence is considered. Linear regression and calculated probabilities are used to order all predictions. For each prediction made, a datapoint (x,y) is created, where x is the prediction confidence and y is 1 if the prediction is correct else 0. The result is a line equation that map from learner confidence to probability of success. Prediction with the highest estimate is chosen as the top prediction. MIL is different form [7] since there is no combiner in it.

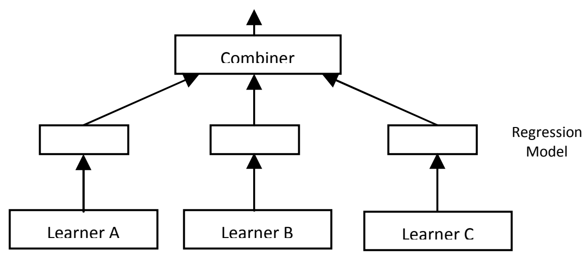

Fig 1. Multi-strategy learning scheme for Information Extraction by [7]

III. METHOD

MIL concept is inspired by the idea how to use document training to look for best classifier for each slot in certain domain. The best classifier for each slot is chosen to extract information in testing documents. Process is started by evaluating each classifier through k-fold cross validation on training documents D_l. The result of this process is a map connecting each slot to classifier performance rank. The classifier with best performance for each slot is choosen to extract information form testing document D_t. Given *Extraction Scenario S* where $\{Slot_1, Slot_2, \ldots, Slot_m\} \epsilon S$, *Base learner* $\{l_1, l_2, \ldots, l_n\} \epsilon L$, *Performance Index* $PI_{slot,Learner}$ = *F(slot, Learner, D_l)* is performance each learner of *L* for each slot= on *Training Document D_l*, (where Dataset $D = D_l + D_t$, D_t = *Testing Document*). Base learner consist of several learners that have significant different in their learning mechanism. To characterize each learner, Performance Index of each learner on each slot is measured. This is done by doing 10-fold cross validation on *Training Document D_l*. MIL then ascociating base-learner with each slot. In this situation, meta-learning is area of expertise search for each learner. The next step is to choose the best learner that will be used to extract information from *Testing Document D_t*. Multi-Inductive Learning algorithm is shown in Figure 2.

```
/* Multi-Inductive Learning Algorithm

Input : Base Learner L = {L1, L2, ..., Ln}, Extraction
        Schenario S where {Slot1,Slot2,...,Slotm} ϵ S,, Training
        Documents Dl, Testing Documents Dt
        where D = Dl + Dt

/* generate Meta-info by k-fold validation test for every
   learner & slot on Training Document

Performance Pslot,Learner = P(learner,slot,Dl)
                            where (slot ∈ S) ∧ (learner ∈ L)
                           /*k-fold cross validation on Dl

/* select best learner for each slot
for each slot in S do:

        Mslot = arg max (Pslot,learner)
              learner ∈ L

     /*retrain each learner on each slot on full Learning
        Document
     Extraction Rule Rslot = train (slot, Mslot, Dl)

 end for
```

Fig 2. Multi-Inductive Learning Algorithm

Figure 3 shows extraction process algorithm on Testing Document.

```
 /* extraction slot filler on Testing Document
Result ← {}
for each slot in E do:
        for each document in Dt do :
                Extract slot filler (slot, document, Rslot)
                /*using best learner to extract doc

                Result ← Result + {document,(slot1, filler
                                  of slot1), ..., (slotm,
                                  filler of slotm)}
        end for
end for
return Result
```

Fig 3. Extraction process algorithm

IV. Result and Discussion

Experiment is conduct using two dataset (dataset Reuters Corporate and dataset Job Posting). Base classifier are PAUM (IND1), SVM (IND2), AODE (IND3), and KNN (IND4). These base classifiers are chosen as they are varied in their approaches. This is to guarantee a variation of bias in MIL. Performance measure in this experiment is F-Measure. As comparison several results of other methods that are using the same datasets are displayed.

A. *MIL performance on Dataset Reuters Corporate*

Area of expertise test on this dataset is shown in Table 1. It is shown for example, on *acqabr* slot IND2 learner perform better than the rest. On the contrary, IND1 learner perform better than the rest on slot *dlramnt*.

TABLE I
Area of Expertise Test (10-fold cross validation test) of Learners on Dataset Reuters Corporate Acquitition

Slot	$P(_{slot,learner})$			
	IND1	IND2	IND3	IND4
acqabr	45,8	**51,9**	18,9	23,5
acqloc	40,1	**44,0**	16,2	2,9
acquired	46,9	**48,9**	26,2	0,0
dlramt	**63,4**	60,1	28,0	6,3
purchabr	42,4	**45,0**	35,0	21,8
purchaser	**48,9**	48,6	37,6	0,2
seller	18,9	**21,4**	21,3	0,2
sellerabr	16,0	18,0	**19,8**	8,3
status	**52,4**	52,2	21,2	10,4

According to this analysis, the best learner that is chosen by MIL to extract information on testing document for slot *acqabr, acqloc, acquired, purchabr* and *seller* is IND2, while for extracting slot *dlramt, purchaser,* and *status* is IND1 and for extracting slot *sellerabr* is IND3.

Table 2 shows performance of MIL on testing document for dataset *Corporate Acquition*. It shows MIL performance is better than other methods on slot *acqabr, acqloc,acquired, dlramt, purchabr, purchaser*, and status. Average performance of MIL is 46.3% which is higher than Rapier (27.8%), SRV (41.2 %) and ELIE (39.4%). This result is supported by the chosen best learner from IND1 and IND2. IND1 performs best on slot: *dlramt, purchabr, purchaser* and *status.* While IND2 is best on slot *acqabr*, *acqloc*, and *acquired.*

On *slot*: *seller* and *sellerabr,* MIL performance is a little bit lower than SRV but better than RAPIER and ELIE. Generally all methods do not get good result in these slots.

TABLE 2
Multi-Inductive Learning (MIL) Performance on Dataset Reuters Corporate Acquitition

Method	Rapier	SRV	ELIE/L2 (SMO-SVM)	MIL
Slot	Ref [2]	Ref [6]	Ref [5]	
acqabr	26.0	38.1	39.7	**57,0**
acqloc	24.2	22.3	34.4	**46,8**
acquired	28.8	38.5	43.5	**50,6**
dlramt	39.3	61.8	59.0	**65,0**

purchabr	24.0	48.5	28.7	**48,7**
purchaser	27.7	45.1	46.2	**52,0**
seller	15.3	**23.4**	15.6	22,4
sellerabr	8.6	**25.1**	13.4	21,0
status	41.3	47.0	49.7	**53,4**
Average	27.8	41.2	39.4	**46,3**

B. *MIL performance on Dataset Job Posting*

Area of expertise test on this dataset is shown in Table 3. It is shown that IND1 learner is expert on slot *application, area, company, country, desired_degree, language, paltform, recruiter, req_degree,* and *salary*. While IND2 learner is expert on slot *city, desired_years_experience, id, post_date, req_years_experience, state* and *title.*

TABLE 3
AREA OF EXPERTISE TEST (10-FOLD CROSS VALIDATION TEST) OF LEARNERS ON DATASET JOB POSTING

Slot	$P(_{slot,learner})$			
	IND1	IND2	IND3	IND4
application	**66,7**	57,4	19,1	19.7
area	**48,6**	42,9	7,8	18.5
city	71,1	**74,0**	49,8	50.0
company	**72,5**	66,9	30,7	39.3
country	**56,4**	46,4	51,9	21.4
desired_degree	**46,4**	45,6	7,6	5.8
desired_years_experience	72,3	**80,7**	75,9	59.5
id	96,3	**96,8**	52,0	96.6
language	**84,4**	75,9	35,9	39.4
platform	**74,9**	67,5	23,2	22.3
post_date	97,5	**97,8**	96,9	97.5
recruiter	**81,8**	80,4	52,6	51.6
req_degree	**78,5**	70,1	19,0	19.4
req_years_experience	70,7	**74,0**	56,9	69.8
salary	**80,0**	78,8	25,6	55.3
state	60,7	**61,7**	38,3	42.3
title	54,2	**56,7**	13,6	30.9

Table 4 shows performance of MIL on testing document for dataset *Job Posting.* It shows MIL performance is better than other methods on slot *city, company, desired_degree, platform, recruiter, req_degree, salary, state,* and *title*. This performance is contributed by IND1 which is best on slot *application, area, company, country, desired_degree, language, platform, recruiter, req_degree,* and *salary.* While the best learner for slot *city, desired_years_experience, id, post_date, req_years_ experience, state* and *title* is IND2. If we compare MIL to other state of the art methods in Information Extraction, the average performance of MIL is 82.1% which is better than RAPIER (75.1 %) , LP2 (77.2%), and SNOW (78.7%).

TABLE 4
MULTI-INDUCTIVE LEARNING (MIL) PERFORMANCE ON DATASET JOB POSTING

Method	Rapier	LP2	SNOW	MIL
Slot	[2]	[3]	[12]	
application	69,3	**78,4**	60,9	73,9
area	42,4	**66,9**	51,6	57,3
city	90,4	93,0	89,0	**95,5**
company	70,0	71,9	75,4	**82,0**
country	93,2	81,0	95,5	58,8
desired_degree	72,2	65,1	60,9	**74,5**
desired_years _experience	**87,5**	60,4	79,0	86,0
id	97,5	**100,0**	99,7	99,0
language	80,6	**91,0**	82,5	88,2
platform	72,5	80,5	74,1	**81,9**
post_date	**99,5**	**99,5**	99,2	99,0
recruiter	68,4	80,6	85,3	**87,2**
req_degree	81,5	84,7	83,5	**85,8**
req_years _experience	67,1	68,8	**83,9**	81,0
salary	67,4	62,8	72,9	**84,1**
state	90,2	84,7	91,7	**92,5**
title	40,5	43,9	52,7	**69,0**
average	75,1	77,2	78,7	**82,1**

V. CONCLUSIONS

Through classification approach, Information extraction can be solved through inductive learning. Nevertheless single classifier approach is not always consistent in performance across domains and slots. Multi-inductive learning is proposed to cope with this problem. By carefully choosing base classifiers, meta-learner in Multi-Inductive Learning can perform better than single classifier approach and other state of the art in Information Extraction.

REFERENCES

[1] Bensusan, H., Cristophe, G.C., and Kennedy, C.J. 2000. A high order Approach to Meta Learning. Eleventh Europian Conference on Machine Learning, Barcelona, Spain.

[2] Califf, M.E. dan Mooney, R.J. 1999. Relational Learning of Pattern-Match Rules for Information Extraction. In Proceedings of the Sixteenth National Conference on Artificial Intelligence and Eleventh Conference on Innovative Applications of Artificial Intelligence (Orlando, FL, 1999), pp. 328-334.

[3] Ciravegna, F. 2001. (LP)2, an adaptive algorithm for information extraction from Web-related texts. In IJCAI-2001 Workshop on Adaptive Text Extraction and Mining. Seattle, USA, 2001.

[4] DesJardins, M. and Gordon, D.F. 1995a. Special issue on bias evaluations and selection. Machine Learning 20 (1/2)R. E. Sorace, V. S. Reinhardt, and S. A. Vaughn, "High-speed digital-to-RF converter," U.S. Patent 5 668 842, Sept. 16, 1997.

[5] Finn, A. 2006. A Multi-Level Boundary Classification Approach to Information Extraction. Ph.D. Thesis, University College Dublin.

[6] Freitag, D. 1998b. Toward general-purpose learning for information extraction. In Christian Boitet and Pete Whitelock, eds., Proc. 36th Annual Meeting of the Association for Computational Linguistics, pp. 404–408. San Francisco, CA, 1998.

[7] Freitag, D. 1998a. Multistrategy for Information Extraction. Proceeding of the Fifteenth International Conference on Machine Learning. 161-169. "PDCA12-70 data sheet," Opto Speed SA, Mezzovico, Switzerland.

[8] Holland, J, Lashon, B., Marco, C., David, G., Stephanie, F., Rick, R., Robert, S., Luca, L.P., Wolfgang, S., and Stewart, W. 2000. What is a learning classifier system? Lecture Notes in AI. Springer Verlag.

[9] Mitchell, T. 1980. The need for biases in generalizations. Technical Report CBM-TR-117. Computer Science Department, Rutgers University , New Brunswick, NJ 08904.

[10] Mitchell, T.M. 1997. Machine Learning. The McGraw-Hill Company,Inc. 414 p.

[11] Pratt, L dan Thrun, S. 1997. Second Special Issue on Inductive Transfer. Machine Learning. 28.

[12] Roth, D and Yih, W.T. 2001. Relational learning via propositional algorithms: An information extraction case study. In International Joint Conference on Artificial Intelligence.

[13] Thrun, S. 1998. Lifelong Learning Algorithms. Learning to Learn. Kluwer Academic Publishers, MA.

[14] Thrun, S. and Pratt, L. 1998. Learning to Learn. Introduction and Overview. Kluwer Academic Publishers, MA.

[15] Thrun, S. and Sullivan, J. 1998. Clustering Learning Task and the Selective Cross task transfer of Knowledge. Learning to Learn. Kluwer Academic Publishers, MA.

[16] Vilalta, R. dan Drissi, Y. 2002. A Perspective View and Survey of Meta-Learniing. Artificial Intelligence Review, 18(2), 77-95.

[17] Wolpert, D. 1992. Stacked Generalization. Neural Network, 5:241-259.