Lifelong Learning Maxent for Suggestion Classification

Thi-Lan Ngo[1,3,4], Tu Vu[2,3], Hideaki Takeda[3], Son Bao Pham[4], Xuan Hieu Phan[4]

[1] University of Information and Communication Technology, Thainguyen University, Vietnam

[2] Hanoi University of Science and Technology, Vietnam

[3] National Institute of Informatics, Tokyo, Japan

[4] University of Engineering and Technology, Vietnam National University, Vietnam

ntlan@ictu.edu.vn, vutu201130@gmail.com, takeda@nii.ac.jp,
sonpb@vnu.edu.vn, hieupx@vnu.edu.vn

Abstract. Suggestion classification for opinion data is defined as identifying a given utterance by suggestion or non-suggestion class. In this paper, we introduce a method called LLMaxent which is the solution for the cross-domain suggestion classification. LLMaxent is a lifelong machine learning approach using maximum entropy (Maxent). In the course of lifelong learning, the drawn knowledge from the past tasks is retained and supported for the future learning. From that, we build a classifier by using labeled data in existed domains for suggestion classification in a new domain. The experimental results show that the proposed novel model can improve the performance of cross-domain suggestion classification. This is one of the preliminary research in lifelong machine learning using Maxent. Its effect is not only for suggestion classification but also for cross-domain text classification in general.

Keywords. Suggestion mining, cross-domain suggestion classification, lifelong learning, maximum entropy.

1 Introduction

Suggestion mining in opinion texts is an emerging and potential topic which has attracted researcher's attention in the explosion of information technology times. It is defined as a sentence classification task, i.e., classify a given sentence into a suggestion or non-suggestion class [18, 2, 21]. The suggestion is referred as two ways, one is giving recommendations or tips for the fellow customers with the variety of choice [12, 11, 22]; the other is considering product/service improvement for the brand owners [2, 23]. Currently, suggestion classification are trained as statistical classifiers with a variety of features in the restricted domain. In fact, these user-generated contents on text are expanded into many different domains that made it difficult to label training data manually. In addition, supervised classification is presented as a typical domain-specific learner, but the performance have a strong decrease in cross-domain or being transferred into different domains. Building these systems require a large amount of annotated data in each domain, human labor-intensive and time-consuming. Thus, a reasonable way is using labeled data in the existing domains for suggestion classification in a new domain. To address this issue, we introduce a new method, called LLMaxent, to cross-domain suggestion classification.

In this paper, we aim to build a system which can adapt to other domains. The challenge is how to utilize labeled suggestion datasets in past domains (source domains) into the new domain (target domain). This raises an interesting task, cross-domain suggestion classification in particular and supervised classification in general. The real world always changes so everything also changes constantly. As a result, the labeling needs to be updated continuously if we use an isolate learning model. The isolate learning model runs a machine

Computación y Sistemas, Vol. 22, No. 4, 2018, pp. 1385–1393
ISSN 1405-5546
doi: 10.13053/CyS-22-4-3107

learning algorithm through a data to generate a model and then apply the model to real-life tasks. This model will not consider the learned knowledge in the past or other related information as supporter for future learning. Herein, we tackle suggestion classification transferred from past domains into future domain by using lifelong machine learning, or lifelong learning (LL). Because the learning paradigm of LL imitates to human-learned that "retaining the learned knowledge from the past and use the knowledge to help future learning" [27, 26, 8, 15].

We develop a LL model based on maximum entropy classification to suggest mining cross-domain, called LLMaxent. LLMaxent model will be tested on Suggestion datasets in English. Our contributions are in two aspects:

— A novel lifelong learning approach to suggestion classification, LLMaxent, is proposed.

— We come out a method that uses maximum weighted entropy and frequency of words in the past domains; then embedding the knowledge gained to improve learning domain based on suggestion words. Lastly, a better classifier is built and experimented on English data.

The paper is structured as follows. Section 2 describes the relevant studies in suggestion classification in two pieces single domain, cross-domain and LL with cross-domain classification. Section 3 briefly describes basic concepts in LL and Maxent. Section 4 is hypothesis research. Section 5 states about the solution for the problem is proposed LLMaxent model. In Section 6, we show experiments and evaluation approach to the tasks of single and cross-domain suggestion classification. In this section, we also evaluate the performance of different base classifiers. Finally, Section 7 draws the conclusions and suggestions for the future research.

2 Related Studies

Our work mainly mention suggestion mining and Lifelong learning for cross-domain. In suggestion mining area, the experiments on single-domain were performed by [2, 12, 11, 22, 23] using rule and machine learning approach. In the same aspect, Negi et al [18] conducted suggestion classification on both machine learning approach and deep learning approach in single domain and cross-validation from different domains. However, the cross-validation from different domains is only proceeded by transferring learning to one domain. Also, the experiments showed the performance of the classifier is significantly reduced while being trained in one domain and evaluated on the other domain. Moreover, they have not yet given any solution for improving the efficiency of classifiers in cross-domain classification. The striking idea of our research to previous studies is building suggestion classification model which can adapt learning to different domains.

In the lifelong and multi-task learning area, existing lifelong learning approaches focused on exploiting invariance [27] and other types of knowledge [6, 25, 8, 7, 15] across multiple tasks. Multi-task learning optimizes the learning of multiple related tasks at the same time [4, 5, 28]. However, these methods are not for suggestion mining. Also, LL based maximum entropy is quite different from all these existing techniques [25, 6, 8, 26].

3 Background

This section provides a brief introduction to lifelong machine learning and Maximum Entropy modelling. The reasons that we use them in cross-domain suggestion classification and many other domains are also explained.

3.1 Lifelong Learning

Although many machine learning studies are related to LL, e.g., lifelong learning [27, 6, 26], a unified definition for LL is just given in 2015 [8] and fully discussed [7] following:

Definition (Lifelong Learning):
"A learner has performed learning on a sequence of tasks, from 1 to $N-1$. When faced with the N^{th} task, it uses the knowledge gained in the past $N-1$ tasks to help learning for the N^{th} task."

According to the above definition, an LL system needs the four general components: (1) Past Information Store (PIS) to stores the information resulted from the past learning; (2) Knowledge Base (KB) to stores the knowledge mined or consolidated from PIS; (3) Knowledge Miner (KM) to mines knowledge from PIS. The knowledge, which is mined, is stored to KB; (4) Knowledge-Based Learner (KBL) is able to leverage the knowledge and/or some information in PIS for the new task from the knowledge in KB.

There are the techniques related learning in cross-domain such as transfer learning [13, 20], multitask learning [4], never-ending learning [3] and domain adaptation [20]. However, LL is still the chosen one for our goal because it build a suggestion classification system which can adapt a large number of different domains and always ready for new domains in the future, as following reasons:

— Whilst Multitask Learning must co-learn all tasks simultaneously, i.e., the learner optimizes the learning across all tasks by using shared knowledge, LL can generate some prior knowledge from the past tasks to help new task learning without necessity of information on new task. Besides, LL does not jointly optimize the learning of the other tasks.

— Like as Transfer Learning (or Domain Adaptation), the goal of LL is to learn well for t_n by transferring some shared knowledge from past tasks, $t_1, t_2, \ldots, t_{n-1}$, to new task, t_n. However, the literature on transfer learning majorly perform one source domain (i.e., n=2). And, the goal of Transfer Learning is to learn well only for the target task (new task). The optimize of source tasks (past tasks) learning is irrelevant. It does not use the results of the past learning or knowledge mined from the past learning results.

— The learner of LL has performed learning on a sequence of tasks with or without seeing the future task data so far. The future task learning simply uses the knowledge without information of future task data, learning simply uses the knowledge in the past. This makes LL different from both Transfer Learning and Multitask Learning.

— LL is suitable for big data and many tasks (i.e., $n-1$ should be large).

3.2 Maximum Entropy Model

The first introduction of maximum entropy model (Maxent) to Natural Language Processing (NLP) area was presented by Berger et al. [1]. Then, it has been used in many NLP tasks such as machine translation, tagging, parsing [24, 19, 14]. A Maximum Entropy model can combine various forms of contextual information into a principled way without any distributional assumptions on the observed data. It can train millions of features and data points. It can scale extremely well and decode or predict very fast. Because of these advantages, we used Maxent as the foundation for building a lifelong learning suggestion classifier.

The goal of Maxent is estimating a p probability distribution with maximum entropy (or "uncertainty") subject to the constraints (or "evidence"). p has the parametric form [1]:

$$p^*(y|x) = \frac{exp\left(\sum_i \lambda_i f_i\left(x, y\right)\right)}{\sum_{y'} exp\left(\sum_i \lambda_i f_i\left(x, y'\right)\right)}, \qquad (1)$$

in which, x is input object (observed object); y is the classified label; f_i is a feature function; λ_i is a weight of feature i.

4 Problem Statement

In this section, we introduce the task of suggestion classification into texts (discussions, tweets, reviews, comments, status) and state problem of cross-domain suggestion classification to many domains using LL approach. The first problem is suggestion mining. It aims to classify a sentence or a tweet into suggestion (positive class) or non-suggestion (negative class).

A sentence/tweet is seen as a suggestion if the sentence/tweet is about suggestions and proposals towards a target (usually a brand owner, company, producer or a person. This puts forward some ideas or plans to think about. Suggestion can be advice, tips, hints, experiments, instructions.

The suggestion classification problem can be stated as Definition 1.

Definition 1: Suggestion Classification problem Let set D of domains, D = $\{D_1, D_2, ...D_n\}$, each $D_i \in D$ is a dataset $D_i = \{(x_1, y_1), (x_2, y_2), ...(x_m, y_m)\}$ in which, x_i is a sentence or tweet, y_i is label corresponding with x_i, y_i = suggestion, none suggestion. Suggestion classification in a D_i domain is seen as seeking a predictor f (also called a classifier) that maps an input vector x to the corresponding class label y.

The second problem is cross-domain suggestion classification. Our aim is building a classifier can retain and accumulate the learned knowledge in the past and use it seamlessly for future learning. Like the learning human process and capability, over time it can learn more and more and store more and more knowledgeable, and learn more and more effective. Based on the prior research and background of LL, in the scope of our work, we stated the lifelong learning problem for suggestion classification on many domains as Definition 2.

Definition 2: Lifelong Learning problem for suggestion classification
Let set of domains D (as in Definition 1), we need to build a classifier which is satisfied Definition 1 and has performed learning on a sequence domains, $D_1, D_2, ..., D_{i-1}$. When classify on D_i domain, it uses the knowledge gained in the past $i-1$ domains to help classifying for the current domain, D_i, and other domains in the future, $D_{i+1}, D_{i+2}, ..., D_n$.

Herein, we consider that the current domain, D_i is known and the future domains, $D_{i+1}, ..., D_n$ are unknown. The built classifier need to satisfy three key characteristics of LL: continuous learning, knowledge accumulation and maintenance in the KB, and the ability to use the past knowledge to help future learning. The solution for above problems is described in Section 5.

5 Proposed Method: LLMaxent Model

A general architecture of LL system is shown in Figure 1.

To build an LL system, we need to determine four components: Past Information Store (PIS), Knowledge Base (KB), Knowledge Miner (KM), and Knowledge-Based Learner (KBL). This means we need to determine the information should be retained from the past domain learning, the forms of knowledge will be used to help future learning, and the way which the system obtain the knowledge.

1. PIS: After past domain learning t, we have information original data ($D^t{}_{train}$), the results of prediction of model (D^t_{pri}) and predict probability of a token w in the dictionary of D_{train} ($w \in V^t_{train}$) belong to class c_j ($\lambda^t_i(w_k, c_j)$), in which V^t_{train} is dictionary of domain D^t. We do not store original data (D^t_{train}), we only store total of the frequency of token w in sentence x_i in D^t_{train} ($N^t(w, c_j, D^t_{train})(w \in V_{train},\ c_j \in Y)$ and $\lambda^t_i(w_k, c_j)$).
2. KB: number of occurrences of w in the past domains
   $N^{KB}(w, c_j) = \sum N^t(w, c_j, D^t train)$ and the sets of cue words to identification class c_j. For example, the cue words of suggestion include "should", "recommend", "advice" and so on. The way of mining cue sets is presented in 5.2.
3. KM: It mined number of occurrences of w in the past domains and cue sets.
4. KBL: This learner is explained in Sub-section 5.1.

5.1 Knowledge-Based Learner

From Equation 1, we see probability distribution p^t of learning domain D^t that we need to seek, has following parametric form as Equation 2:

$$p^{t*}(y|x) = \frac{exp\left(\sum_{w \in V^t} \lambda_(w, c) f_{(w,c)}(x, y)\right)}{\sum_{c' \in Y^t} exp\left(\sum_{w \in V^t, c' \in Y^t} \lambda_(w, c') f_(w, c')(x, y)\right)}, \quad (2)$$

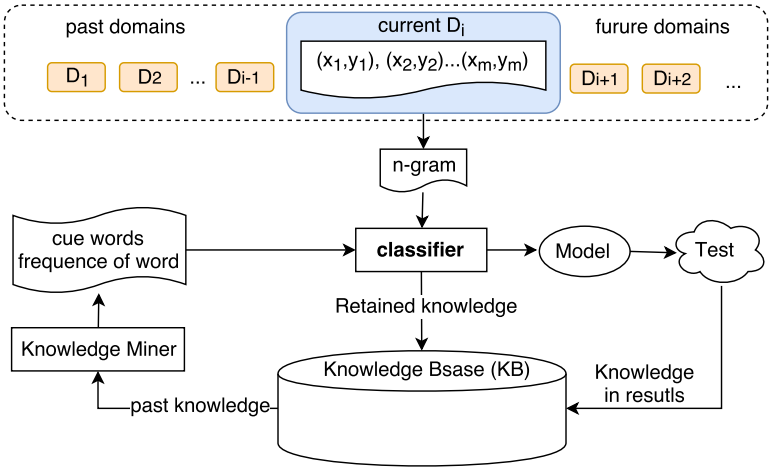

Fig. 1. The LLMaxent system architecture.

in which:
$D^t_{train} = (X^t, Y^t)$ is training data of domain D;
$X^t = \{x_i\}$ includes the sentences or tweets and Y^t is set of the labels;
$V^t = \{w | w \in x_i\}$;
$(x, y) \in D^t_{train}$.

In order to train the MaxEnt models and use knowledge base, we used two kinds of feature templates from the training data and KB: n-gram and cue words. For n-gram feature, we use uni-gram and bi-gram and a token is a n-gram. A $(x_i \; contains \; token \; w_k)$ is a context predicate of the model. The form of feature function as Equation 3.

$$f_{j(w_k,c_{i'})}(x_i, y_i) = \begin{cases} \frac{N(w_k,x_i)}{N(x_i)} + \frac{\sum_{i''=1}^{t-1} N(w_k,c_{i'},V^{i''})}{\sum_{i''=1}^{t-1} N(V^{i''})} & if(y = c_{i'}) \; and \; (\; w_k \in x_i), \\ 0 & otherwise, \end{cases} \tag{3}$$

in which:
$w_k \in V^t$, $c_{i'} \in Y$, $(x - i, y_i) \in D^t$;
$N(w_k, x_i)$ is number of times that w_k token occurs in sentence x_i;
$N(x_i)$ is number of token in the x_i. In other words, $N(x_i)$ is the length of x_i;
$\sum_{i''=1}^{t-1} N(w_k, V^{i''}) = N^{KB}(w_k, c_{i'})$ is total of the times that w_k occurs in the sentences has label is $c_{i'}$ in the past domain and $N^{KB}(w_k, c_{i'})$ is called the knowledge base frequency of token w_k;
$N(V^{i''})$ is number of the tokens in the past domain $D^{i''}$.

To use the cue words as features of model, we call x_i which contains a cue word as a context predicate of the model. Calling $Cues_{c_{i'}}$ is a set of the cue words to identify the $c_{i'}$ class, we have the form of feature functions as following:

$$f_{j(w_k,c_{i'})}(x_i, y_i) = \begin{cases} \frac{1}{N(x_i)} + \frac{N(x_i,c_{i'},V^{i''})}{\sum_{i''=1}^{t-1} V^{i''}} & if \; (y = c_{i'}) \& (w_k \in x_i) \& (w_k \in Cues_{c_{i'}}), \\ 0 & otherwise, \end{cases} \tag{4}$$

in which:
$N(x_i, c_{i'}, V^{i''})$ is the times that (x_i, y_i) occur in the past domains where (x_i, y_i) satisfies $x_i \ni w_k$ with w_k is the cue word ($w_k \in Cues_{c_{i'}}$) and $y_i = c_{i'}$.
$V^{i''}$ is the dictionary of domain $D^{i''}$.

We can easily see that f function returning a value in $[0, 2]$. So p^{t*} probability distribution in Equation (2) is uniquely including [9]. Because it uniquely maximizes the entropy over distributions that satisfy *constraint equation* of maximum entropy model [1], and uniquely maximizes the likelihood over distributions of the form (1). The model parameters for the distribution p are obtained via Generalized Iterative Scaling (GIS) [10], Improved Iterative Scaling (IIS) [19], or L-BFGS [16].

5.2 Building Cues Set

We automatically extract cue words from all of past domains and use them directly to classify unseen sentences in the future domains. The automatically discovered cue words for c class in the past domains are stored in the corresponding $Cues_c$ set. The main idea of cue words extraction is that the higher prediction probability words are, the higher their meaning are. From that, in the classification process they will be updated in the cue words set. To set cue words for the c_i class, we choose α word w, which has weighted score λ corresponding to c_i ($\lambda(w, c)$) is highest. As in, α is called threshold value of cue words which update at the current domain t. If a word w occurs more than one class, we consider the word w for the highest probability class. The following algorithm will explain in detail for our cue words extraction automatically.

In the lifelong machine learning process over many domains, the sets of cue words are considered again. Inappropriate cue words will be excluded from the cue words sets. To search these inappropriate cue words, we count the times that cue words occur in each class c_i in the current domain. From the cue words sets, we exclude the cue words, whose frequency in each different classes is less than threshold value β, by using Algorithm 2. As in, β is called the threshold value of excluding cue words. After testing the model for the new domain D^{t+1}, we obtain the predicted model results D'^{t+1}. By using Algorithm 2 for D'^{t+1}, we continue to exclude the inappropriate cue words.

Algorithm 1 Get relevant cue words of a class c at learning domain t

procedure GETCUEWORDS($\mathbf{D^t} = (\mathbf{X^t}, \mathbf{Y^t})$, $\mathbf{V^t} = \{w | w \in x_i \; and \; (x_i, y_i) \in D^t\}$ $\lambda^* = \{\lambda(\mathbf{w_k}, \mathbf{c_i}) | \mathbf{w_k} \in \mathbf{V^t}, \mathbf{c_i} \in \mathbf{Y}\}$, $\mathbf{Cues_c^{t-1}}$, α)
  create empty set of cues words: $\mathbf{Cues_c^t} = \mathbf{Cues_c^{t-1}}$
  create empty set of cues words: $\mathbf{Temp_c^t} = \emptyset$
  for each $w \in \mathbf{V^t}$ **do**
    if $\lambda_(w, c) = max_{(i=1)}^{l}(\lambda(w, y_i))$ **then**
      $\mathbf{Temp_c^t} \leftarrow \mathbf{Temp_c^t} \cup \{\mathbf{w}\}$
    end if
  end for
  sort $\mathbf{temp_c^t}$ in descending order
  for each $i \in [1, len(tepm_c^t)]$ **do**
    if $(i < \alpha)$ **then**
      $\mathbf{Cues_c^t} \leftarrow \mathbf{Cues_c^t} \cup \{\mathbf{w_i}\}$
    end if
  end for
  return the set of cue words corresponding to the $\mathbf{c}$ class: $\mathbf{Cues_c^t}$
end procedure

Algorithm 2 Exclude unreasonable cue words

procedure EXCLUDECUEWORDS($\mathbf{D^t} = (\mathbf{X^t}, \mathbf{Y^t})$ is training domain, $\mathbf{Cues_c^{t-1}}$)
  create a list: $\mathbf{Nw[len(Y^t)]} \leftarrow 0$
  for each $y_i \in \mathbf{Y}$ **do**
    for each $w \in Cues_c^t$ and $x \in X^t$ **do**
      if $w \in x$ **then**
        $Nw[y_i] = Nw[y_i] + 1$
      end if
    end for
  end for
  for each $y_i, y_j \in \mathbf{Y}$ **do**
    if $\mid Nw[y_i] - Nw[y_j] \mid < \beta$ **then**
      $\mathbf{Cues_c^t} \leftarrow \mathbf{Cues_c^t} \backslash \{\mathbf{w}\}$
    end if
  end for
  return the set of cue words corresponding to the $\mathbf{c}$ class: $\mathbf{Cues_c^t}$
end procedure

Table 1. Names of the 6 datasets and the proportion of suggestion in each dataset

Name	Proportion (Suggestion/total)	Characteristic	Public
Advice	2192/5199	Type of data: post in forum Domain: travel Type of suggestion: explicit, implicit	Wicaksono & Myaeng [23]
Electronics	273/3782	Type of data: review Domain: electronics Type of suggestion: explicit	Negi & Buitelaar [17]
Hotel	407/7534	Type of data: review Domain: hotel Type of suggestion: explicit	Negi & Buitelaar [17]
Forum	1517/5229	Type of data: post in forum Domain: Feedly mobile app & Windows App Type of suggestion: explicit	Negi [18]
Microsoft	238/3000	Type of data: tweets Domain: Microsoft phones Type of suggestion: explicit	Dong et al [11] Negi [18]
Hastag	966/3628	Type of data: tweets Domain: open domain Type of suggestion: explicit	Negi [18]

5.3 Results

We compare our proposed LLMaxent model with Maxent which is implemented according to Nigam[19]. We use 5 domains for training and the remaining domain for testing. For example, in Table2, "– advice" mean 5 domain which different to "advice" domain is used for training, "advice" domain is not used for training, it is only used for testing.

6 Experimental Studies

Model estimation involves setting the weight values. We train maxent with L-BGFS because it was shown it is fast and efficient [19]. As mentioned early, we used uni-gram and bi-gram features of the model. We use $\alpha = 50$ and $\beta = 1$ in our experiments. We use $precision$, $recall$ and $F_1 - score$ is measure the score.

6.1 Datasets

In this paper, the experiment is classifying a sentence/tweet into a suggestion or non-suggestion class. Labeled suggestion data is available[1]. We revise again and report experiment data in Table 1. We can observe that these data sets have not only different topics but also different types of data and sources.

The results of LLMaxent model is higher than Maxent which is implemented according to Nigam's model. Because the training data in the current domain may not be fully representative of the test data due to the sample selection bias. The data in real-life applications may contain some suggested words which are absent in the training data of current.

Meanwhile, these suggested words have appeared in some past domains, the past domain knowledge can contribute to the target domain classification. However, to see the advances of the

[1] http://server1.nlp.insight-centre.org/sapnadatasets/

Table 2. Macro, micro average F1-score of the suggestion class of Maxent model and LLMaxent model

Train	Test	Maxent			LLMaxent		
		Precision	Recall	f_1-score	Precision	Recall	f_1-score
- advice	advice	35.66	2.98	5.39	33.3	33.1	29.8
- electronic	electronic	29.31	5.64	8.66	21.34	30.52	22.85
- forum	forum	36.42	3.89	7.02	31.19	77.67	44.38
- hashtag	hashtag	35.64	3.37	6.09	24.32	35.14	26.68
- hotel	hotel	34.26	3.97	6.79	24.32	35.14	26.68
- microsoft	microsoft	8.11	1.26	2.18	10.01	65.55	17.36

general Lifelong Learning system, it needs a large number of training domains in the past. In some cases, it can be not good due to the knowledge in new domain having a sharp difference from the learned domains.

Nevertheless, in the big data opportunity, a Lifelong Learning system can be promoted by its continuous learning when abundant information and extensive sharing of concepts across tasks/domains from opinion data generated by the user in the Web.

7 Conclusions

In this study, we have presented a new approach for cross-domain suggestion classification in opinion text data as comments, reviews, posts. We proposed a novel method which approached lifelong machine learning based on maximum entropy. We investigated a cue-based approach and combined it into its frequency in past domains. The evaluation are proceeded and obtained the promising results.

However, lifelong learning needs a larger number of tasks or domains. Hence in the future, the new domains for suggestion classification should be advisory and experiments on other text classification problems should be conducted.

Acknowledgements

This work was supported by the project QG.16.34 from Vietnam National University, Hanoi (VNU).

References

1. **Berger, A. L., Pietra, V. J. D., & Pietra, S. A. D. (1996).** A maximum entropy approach to natural language processing. *Computational linguistics*, Vol. 22, No. 1, pp. 39–71.
2. **Brun, C. & Hagege, C. (2013).** Suggestion mining: Detecting suggestions for improvement in users' comments. *Research in Computing Science*, Vol. 70, No. 79.7179, pp. 5379–62.
3. **Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka Jr, E. R., & Mitchell, T. M. (2010).** Toward an architecture for never-ending language learning. *AAAI*, volume 5, Atlanta, pp. 3.
4. **Caruana, R. (1997).** Multitask learning. *Machine learning*, Vol. 28, No. 1, pp. 41–75.
5. **Chen, J., Zhou, J., & Ye, J. (2011).** Integrating low-rank and group-sparse structures for robust multi-task learning. *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 42–50.
6. **Chen, Z. & Liu, B. (2014).** Topic modeling using topics from many domains, lifelong learning and big data. *International Conference on Machine Learning*, pp. 703–711.
7. **Chen, Z. & Liu, B. (2016).** Lifelong machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, Vol. 10, No. 3, pp. 1–145.
8. **Chen, Z., Ma, N., & Liu, B. (2018).** Lifelong learning for sentiment classification. *arXiv preprint arXiv:1801.02808*.
9. **Darroch, J. N. & Ratcliff, D. (1972).** Generalized iterative scaling for log-linear models. *The annals of mathematical statistics*, pp. 1470–1480.
10. **Darroch, J. N. & Ratcliff, D. (1972).** Generalized iterative scaling for log-linear models. *The annals of mathematical statistics*, pp. 1470–1480.

11. **Dong, L., Wei, F., Duan, Y., Liu, X., Zhou, M., & Xu, K. (2013).** The automated acquisition of suggestions from tweets. *AAAI*.
12. **Goldberg, A. B., Fillmore, N., Andrzejewski, D., Xu, Z., Gibson, B., & Zhu, X. (2009).** May all your wishes come true: A study of wishes and how to recognize them. *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, pp. 263–271.
13. **Jiang, J. (2008).** A literature survey on domain adaptation of statistical classifiers. *URL: http://sifaka. cs. uiuc. edu/jiang4/domainadaptation/survey*, Vol. 3, pp. 1–12.
14. **Klein, D., Smarr, J., Nguyen, H., & Manning, C. D. (2003).** Named entity recognition with character-level models. *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, Association for Computational Linguistics, pp. 180–183.
15. **Liu, B. (2017).** Lifelong machine learning: a paradigm for continuous learning. *Frontiers of Computer Science*, Vol. 11, No. 3, pp. 359–361.
16. **Liu, D. C. & Nocedal, J. (1989).** On the limited memory bfgs method for large scale optimization. *Mathematical programming*, Vol. 45, No. 1-3, pp. 503–528.
17. **Negi, S. & Buitelaar, P. (2015).** Towards the extraction of customer-to-customer suggestions from reviews. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 2159–2167.
18. **Negi, S. & Buitelaar, P. (2017).** Suggestion mining from opinionated text. *Sentiment Analysis in Social Networks*, pp. 129–139.
19. **Nigam, K., Lafferty, J., & McCallum, A. (1999).** Using maximum entropy for text classification. *IJCAI-99 workshop on machine learning for information filtering*, volume 1, pp. 61–67.
20. **Pan, S. J., Yang, Q., et al. (2010).** A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, Vol. 22, No. 10, pp. 1345–1359.
21. **Pozzi, F. A., Fersini, E., Messina, E., & Liu, B. (2016).** *Sentiment analysis in social networks*. Morgan Kaufmann.
22. **Ramanand, J., Bhavsar, K., & Pedanekar, N. (2010).** Wishful thinking: finding suggestions and'buy'wishes from product reviews. *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*, Association for Computational Linguistics, pp. 54–61.
23. **Ramanand, J., Bhavsar, K., & Pedanekar, N. (2010).** Wishful thinking: finding suggestions and'buy'wishes from product reviews. *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*, Association for Computational Linguistics, pp. 54–61.
24. **Ratnaparkhi, A. (1997).** A simple introduction to maximum entropy models for natural language processing. *IRCS Technical Reports Series*, pp. 81.
25. **Ruvolo, P. & Eaton, E. (2013).** Ella: An efficient lifelong learning algorithm. *International Conference on Machine Learning*, pp. 507–515.
26. **Silver, D. L., Yang, Q., & Li, L. (2013).** Lifelong machine learning systems: Beyond learning algorithms. *AAAI Spring Symposium: Lifelong Machine Learning*, volume 13, pp. 05.
27. **Thrun, S. (1998).** Lifelong learning algorithms. In *Learning to learn*. Springer, pp. 181–209.
28. **Zhang, Y., Owechko, Y., & Zhang, J. (2008).** Learning-based driver workload estimation. In *Computational intelligence in automotive applications*. Springer, pp. 1–17.

Article received on 12/12/2017; accepted on 15/02/2018.
Corresponding author is Thi-Lan Ngo.