Lifelong Learning Maxent for Suggestion Classification

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

University of Information and Communication Technology, Thainguyen University [1]
Hanoi University of Science and Technology, Vietnam [2]
National Institute of Informatics, Tokyo, Japan [3]
University of Engineering and Technology, Vietnam National University[4]
ntlan@ictu.edu.vn, vutu201130@gmail.com, takeda@nii.ac.jp
sonpb@vnu.edu.vn, hieupx@vnu.edu.vn

Abstract. Suggestion analysis of opinion data is classifying a given utterance into one of two classes: suggestion and non-suggestion. In this paper, we introduce a new method, called LLMaxent, to cross-domain suggestion classification. LLMaxent is an approach to lifelong machine learning using maximum entropy (Maxent) method. Based on the main idea of lifelong learning, that is retaining the knowledge learned from past tasks and using it to help future learning, we build a classifier can use labelled 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 the first study of lifelong machine learning using Maxent and our method is not only useful 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 from opinion texts is a potential new research topic emerging and has attracted many researcher's attention lately. Suggestion mining from opinion texts is defined as a sentence classification task, i.e., classify a given sentence is a suggestion and non-suggestion [1–3]. The suggestion is referred to advise, recommendations and tips to the fellow customers on a variety of points of interest [4–6] and wishes to improvements product/service to brand owners [2, 7]. Most existing studies trained statistical classifiers experimenting with a variety of features and in a specific domain. In fact, these users generated opinion texts can span many different domains that it is difficult to manually label training data for all of them. Addition, supervised classification systems generally are typically domain-specific, and the performance decreases strongly in cross-domain or transfer between different domains. Building these systems orders a large amount of annotated data for every domain, which needs much human labor-intensive and time-consuming. Thus, a reasonable way is using labelled data in existed domains for suggestion classification in a new domain. To address the issues, 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 labelled suggestion datasets in past domains (source domains) into another 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 labelling needs to be done continuously if we use isolate learning model which runs a machine learning algorithm on a given dataset to generate a model and then applied the model to real-life tasks. These models do not consider the knowledge learned in the past or other related information to use for helping future learning. Herein, we tackle suggestion classification transferred from some past domains into future domain by using lifelong machine learning, or lifelong learning (LL). Because the learning paradigm of LL imitates to human learn that "retaining the learned knowledge from the past and use the knowledge to help future learning" [8–11].

We develop a new LL model based on maximum entropy classification to suggestion mining cross-domain, called LLMaxent. LLMaxent model is tested on Suggestion datasets in English and Vietnamese.
Our contributions are in two folds:

- A novel lifelong learning approach to suggestion classification, LLMaxent, is proposed.
- We come out a method that uses past weights of maximum entropy and frequency of words in the past domains to embed the knowledge gained in the past and to improve learning domain dependent suggestion words to build a better classifier and do well with English data.

The paper is structured as follows. Section 2 describes the related work on suggestion classification in single and cross-domain and LL with cross-domain classification. Section 3 is brief and basic concepts in LL and Maxent. Section 4 is statement research problem. Section 5 explain proposed LLMaxent model. In Section 6, we show experiments and evaluate our approach to the tasks of single and cross-domain suggestion classification. In that section, we evaluate also the performance of our different base classifiers. Finally, Section 7 draw the conclusions and work in the future.

2 Related Studies

Our work mainly related to suggestion mining and Lifelong learning for cross-domain. In suggestion mining area, the experiments on suggestion classification in single-domain were performed by [2, 4–7] using rule and machine learning approach. Negi et al [1] concluded suggestion classification using both machine learning approach and deep learning approach in single domain and cross-validation on a number of datasets from different domains. However, they performed experiments transfer learning to one domain. Moreover, the experiments just showed the performance of the classifier is significantly reduced when it is trained in a domain and evaluated on the other domain. They have not yet given any solution for improving the efficiency of classifiers in cross-domain classification. Unlike the previous studies, our goal is building suggestion classification model which can adapt learning through many different domains.

In lifelong learning and multi-task learning area, existing lifelong learning approaches focused on exploiting invariance [8] and other types of knowledge [12, 13, 10, 14, 11] across multiple tasks. Multi-task learning optimizes the learning of multiple related tasks at the same time [17, 15, 16]. However, these methods are not for suggestion mining. Also, LL based maximum entropy is quite different from all these existing techniques [13, 12, 10, 9].

3 Background

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

3.1 Lifelong Learning

Although many machine learning studies are related to LL, e.g., lifelong learning [8, 12, 9], a unified definition for LL just is given in 2015 [10] and more fully discussion in [14] as 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 [18, 19], multitask learning [17], never-ending learning [20] and domain adaptation [19], but LL is still chosen for our goal that is building a suggestion classification system which can adapt a large number of different domains and always ready for new domains in the future, because of the reasons as following:

- Whilst Multitask Learning must co-learn all tasks simultaneously, i.e., the learner optimizes the learning across all tasks by using some shared knowledge, LL can generate some prior knowledge from the past tasks to help new task learning without any information from this new task. 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, $t_1, t_2, , t_{n-1}$, to new task, t_n. However, almost the entire literature on transfer learning perform with one source domain (i.e., n=2). Moreover, 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 results of the past learning.
- 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 When it does not need any 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. [21]. Then, it has been used in many NLP tasks such as machine translation, tagging, parsing [22–24]. A Maximum Entropy model can combine various forms of contextual information in a principled way without any distributional assumptions on the observed data. It can train with 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 [21]:

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

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 to the task of suggestion classification from texts (discussions, tweets, reviews, comments, status) and state problem of cross-domain suggestion classification in 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 talking about suggestions and proposals towards a target (usually a brand owner, company, producer or a person), and put forward some ideas or plans for someone 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 knowledge learned 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 has known and the future domains, $D_{i+1}, ..., D_n$ have unknown. The built classifier need 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 of 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 umber of occurrences of w in the past domains and cue sets.
4. KBL: This learner is explained in Sub-section **??**.

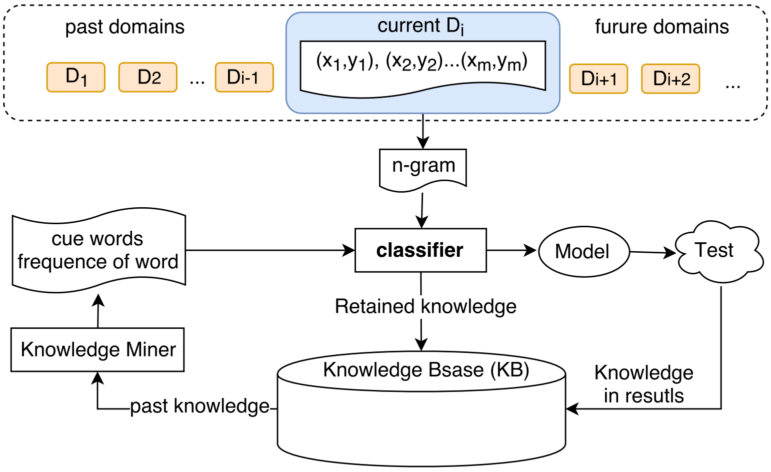

Fig. 1. The LLMaxent system architecture.

5.1 Knowledge-Based Learner

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

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

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 word. 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 past domain $D^{i''}$.

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

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

in which,
$N(x_i, c_{i'}, V^{i''})$ is the times that (x_i, y_i) occurs 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 returns a value in $[0, 2]$. So p^{t*} probability distribution in Equation (2) is uniquely consist [27]. Because it uniquely maximizes the entropy over distributions that satisfy *constraint equation* of maximum entropy model [21], 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)[27], Improved Iterative Scaling (IIS) [23], or L-BFGS[25].

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 words with high prediction probability are high meaning in the classification process and they will be updated in the cue words set. We choose α word w, which weight λ corresponding to c_i ($\lambda(w, c)$) is highest, make cue words for the c_i class. α is called threshold value of cue words update at current domain t. If a word w occurs in more 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.

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

```
procedure GetCueWords($\mathbf{D^t} = (\mathbf{X^t}, \mathbf{Y^t})$, $\mathbf{V^t} = \{w|w \in x_i \ and \ (x_i, y_i) \in D^t\}$
$\lambda^* = \{\lambda(\mathbf{w_k}, \mathbf{c_i})|\mathbf{w_k} \in \mathbf{V^t}, \mathbf{c_i} \in \mathbf{Y}\}$, $\mathbf{Cues_c^{t-1}}$, $\alpha$)
    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 $\lambda_(w, c) = max^l_{(i=1)}(\lambda(w, y_i))$ 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 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: $\mathbf{Nw}[\mathbf{len}(\mathbf{Y^t})] \leftarrow 0$
    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 $| \ Nw[y_i] - Nw[y_j] \ | < \beta$ 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 c class: $\mathbf{Cues_c^t}$
end procedure
```

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

6 Experimental Studies

Model estimation involves setting the weight values. We use quasi-Newton methods like L–BFGS since recent studies have shown it is fast and efficient. As mentioned earlier, 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 was performed to classify a sentence/tweet is a suggestion or non-suggestion. 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 different types of data and sources.

6.2 Results

We compare our proposed LLMaxent model with Maxent which is implemented according to Nigam[23]. 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.

The results of LLMaxent model 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 few real-life applications may contain some suggestion words that are absent in the training data of current, while these suggestion words have appeared in some past domains. So the past domain knowledge can contribute to the target domain classification. However, to see the advances of the general Lifelong Learning system, it needs a large number of training domains in the past. In some case, it not good due to knowledge in the new domain is too far away 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.

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

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 [7]
Electronics	273/3782	Type of data: review Domain: electronics Type of suggestion: explicit	Negi & Buitelaar [26]
Hotel	407/7534	Type of data: review Domain: hotel Type of suggestion: explicit	Negi & Buitelaar [26]
Forum	1517/5229	Type of data: post in forum Domain: Feedly mobile app & Windows App Type of suggestion: explicit	Negi [1]
Microsoft	238/3000	Type of data: tweets Domain: Microsoft phones Type of suggestion: explicit	Dong et al [5] Negi [1]
Hastag	966/3628	Type of data: tweets Domain: open domain Type of suggestion: explicit	Negi [1]

7 Conclusions

In this study, we have presented an new approach for cross-domain suggestion classification in opinion text data as comments, reviews, posts. We proposed a novel method for lifelong machine learning based on maximum entropy. We investigated a cue-based approach and combined with a frequency of words in past domains to cross-domain suggestion classification. Our method was evaluated cross-domain suggestion data and obtained the promising results. However, lifelong learning needs a larger number of tasks or domains. Hence in the future, we will add new domains for suggestion classification and conduct experiment on other text classification problems.

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

Train	Test	Maxent			LLMaxent		
		Pre.	Rec.	f_1-score	Pre.	Rec.	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

Acknowledgments

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

References

1. Negi, S.: Suggestion Mining from Opinionated Text. In: ACL 2016, pp.119 (2016).
2. Brun, C. and Hagege, C.: Suggestion Mining: Detecting Suggestions for Improvement in Users' Comments. In: Research in Computing Science, 70, pp.199-209 (2013).
3. Pozzi, F.A., Fersini, E., Messina, E. and Liu, B.: Sentiment Analysis in Social Networks. Morgan Kaufmann, San Francisco (2016).
4. Goldberg, A.B., Fillmore, N., Andrzejewski, D., Xu, Z., Gibson, B. and Zhu, X.: May all your wishes come true: A study of wishes and how to recognize them. In: Human Language Technologies, pp. 263-271 (2009).
5. Dong, L., Wei, F., Duan, Y., Liu, X., Zhou, M. and Xu, K.: The Automated Acquisition of Suggestions from Tweets. In: AAAI (2013).
6. Ramanand, J., Bhavsar, K. and Pedanekar, N.: Wishful thinking: finding suggestions and'buy'wishes from product reviews. In: NAACL HLT 2010 Workshop, ACL, pp. 54-61 (2010).
7. Wicaksono, A.F. and Myaeng, S.H.: Automatic extraction of advice-revealing sentences foradvice mining from online forums. In: the international conference on Knowledge capture, pp. 97-104 (2013).
8. Thrun, S.: Lifelong learning algorithms. In: Learning to learn, 8, pp.181-209 (1998).
9. Silver, D.L., Yang, Q. and Li, L.: Lifelong Machine Learning Systems: Beyond Learning Algorithms. In: AAAI Spring Symposium: Lifelong Machine Learning, Vol. 13, p. 05 (2013).
10. Chen, Z., Ma, N. and Liu, B.: Lifelong Learning for Sentiment Classification. In: ACL, pp. 750-756 (2015).
11. Liu, B.:Lifelong machine learning: a paradigm for continuous learning. In: Frontiers of Computer Science, 11(3), pp.359-361 (2017).
12. Chen, Z. and Liu, B.: Topic modeling using topics from many domains, lifelong learning and big data. In: International Conference on Machine Learning, pp. 703-711 (2014).
13. Ruvolo, P. and Eaton, E.: ELLA: An efficient lifelong learning algorithm. In: International Conference on Machine Learning, pp. 507-515 (2013).
14. Chen, Z. and Liu, B.: Lifelong Machine Learning. In: Artificial Intelligence and Machine Learning, 10(3), pp.1-145 (2016).
15. Chen, J., Zhou, J. and Ye, J.: Integrating low-rank and group-sparse structures for robust multi-task learning. In: SIGKDD, pp. 42-50 (2011).
16. Zhang, Y., Owechko, Y. and Zhang, J.: Learning-based driver workload estimation. In: Computational intelligence in automotive applications, pp. 1-17 (2008).
17. Caruana, R.: Multitask learning. In: Learning to learn, pp. 95-133 (1998).
18. Zhang, Y., Owechko, Y. and Zhang, J.:Learning-based driver workload estimation. In: Computational intelligence in automotive applications,pp.1-17, (2008).
19. Pan, S.J. and Yang, Q.: A survey on transfer learning. In: Transactions on knowledge and data engineering, 22(10), pp.1345-1359 (2010).

20. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka Jr, E.R. and Mitchell, T.M.: Toward an Architecture for Never-Ending Language Learning. In: AAAI, Vol.5, p.3 (2010).
21. Berger, A.L., Pietra, V.J.D. and Pietra, S.A.D.: A maximum entropy approach to natural language processing. In: Computational linguistics, 22(1), pp.39-71 (1996).
22. Ratnaparkhi, A.: A simple introduction to maximum entropy models for natural language processing. In: IRCS Technical Reports Series, p.81 (1997).
23. Nigam, K., Lafferty, J. and McCallum, A.: Using maximum entropy for text classification. In: IJCAI workshop on machine learning for information filtering, Vol. 1, pp. 61-67 (1999).
24. Klein, D., Smarr, J., Nguyen, H. and Manning, C.D.: Named entity recognition with character-level models. In: HLT-NAACL, Vol. 4, pp. 180-183 (2003).
25. Liu, Dong C., and Jorge Nocedal.:On the limited memory BFGS method for large scale optimization. In: Mathematical programming 45, no. 1, pp. 503-528, (1989).
26. Negi, S. and Buitelaar, P.: Towards the Extraction of Customer-to-Customer Suggestions from Reviews. In: Empirical Methods in Natural Language Processing, pp. 2159-2167 (2015).
27. Darroch, John N., and Douglas Ratcliff.: Generalized iterative scaling for log-linear models. In: The annals of mathematical statistics (1972): 1470-1480.