Automatic Extraction of Advice-revealing Sentences for Advice Mining from Online Forums

[1]Alfan Farizki Wicaksono, [2]Sung-Hyon Myaeng
[1, 2]Department of Computer Science
[2]Division of Web Science and Technology
Korea Advanced Institute of Science and Technology (KAIST)
Daejeon, Republic of Korea
{alfan.farizki, myaeng}@kaist.ac.kr

ABSTRACT

Web forums often contain explicit key learnings gleaned from people's experiences since they are platforms for personal communications on sharing information with others. One of the key learnings contained in Web forums is often expressed in the form of *advice*. As part of human experience mining from Web resources, we aim to provide a methodology to extract advice-revealing sentences from Web forums due to its usefulness, especially in travel domain. Instead of viewing the problem as a simple classification, we define it as a sequence labeling problem using various features. We identify three different types of features (i.e., syntactic features, context features, and *sentence informativeness*) and propose a new way of using Hidden Markov Model (HMM) for labeling sequential sentences, which in our experiment gave the best performance for our task. Moreover, the sentence informativeness score serves as an important feature for this task. It is worth noting that this work is the first attempt to extract advice-revealing sentences from Web forums.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing

General Terms

Algorithms, Performance, Experimentation

Keywords

Advice Mining, Sequence Labeling, Extension of HMM

1. INTRODUCTION

Web has become a place where ordinary Internet users log their daily stories and experiences in the form of Weblogs, comments in boards, or articles in Web forums. In particular, *Web forums* enable users to focus on specific topics of their choice, discuss issues, and share their personal experiences and thoughts with others easily. For example, Web forums such as *Tripadvisor*, *Fodors*, and *Amazon* make it easy for people to share their experiences with others about the places they visited, the hotel services they received, new products they purchased, interesting books they read, or even the latest film they watched. As a result, Web forums contain a huge amount of human knowledge, often based on personal experiences.

As a manifestation of personal experiences, Web forums often contain explicit key lessons and know-how gleaned from people's past experiences which are really worthy to be well presented to other people and used by intelligent agents in providing context-sensitive assistance. Such key lessons are often expressed in the form of *advice*. In travel domain, for example, travelers usually seek advice from travel Web forums before they visit some tourism places [2, 7]. This advice gives them perspective on where they should travel, what they should do, and what they should be aware of. When advice is represented in an appropriate form and indexed with situational and contextual variables, it can serve as useful knowledge for decisions to be made on the go using mobile devices [22]. In fact, we chose the travel domain with some potential applications in mind, such as advice retrieval for travelers, context-aware advice generation, and tourism marketers' assessment tools.

In this paper, we address a new problem referred to as *advice mining*, in which advice is extracted and aggregated from Web forums, and subsequently stored in a well-organized knowledge repository. We call such knowledge as *advice-revealing text unit* (ATU) comprising the following two elements:

1. **Advice-revealing sentence:** A sentence that contains a suggestion for or guide to an action to be taken in a particular context.

2. **Context:** An element that explains or clarifies an advice-revealing sentence in more detail with contextual information. This element is divided into three separated sub-elements: Place, Time, and Condition. The sub-element "Condition" is used to describe other context besides place and time. An advice-revealing sentence can be anchored with one or more sub-elements.

This representation (see also Figure 1) allows a user to retrieve advice by posing general query such as "*need advice for traveling to **China***", or more specific query such as "*need*

K-CAP '13, June 23-26, 2013, Banff, Canada.
Copyright 2013 ACM 978-1-45-03-2102-0/13/06 ...$15.00.

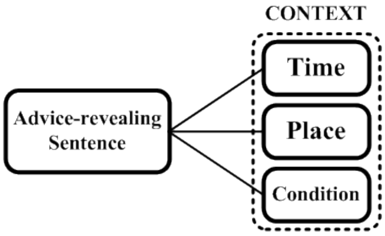

Figure 1: Advice-revealing Text Unit (ATU)

advice for traveling to ***China*** *in* ***Summer***". The first query may result in the system output containing pieces of advice clustered around time or conditions because there is no specific information about the time or condition.

As part of an effort to capture ATU from Web forums, we tackle the problem of identifying sentences that contain explicit advice, which is the first step toward building a rich representation of advice. We note that this capability can serve the purpose of retrieving advice-containing articles with a hot spot, i.e., key advice, for human consumption as well. Below are the examples of advice-revealing sentences extracted from well-known Web forums.

1. *"We just got back from Rio and just wanted to mention to first timers, like us, to make sure you allow enough time for your transfer back to the airport."* [**Fodors**]

2. *"If a quick visit to a credit union doesn't do the trick, the best you can do is to find out if Chase has an arrangement with a foreign bank so at least you save the out of network fees."* [**Fodors**]

3. *"Take a head as your head acts as a chimney venting the warm air and make sure to cover your ears."* [**TripAdvisor**]

This problem was addressed recently by Kozawa et al. [11], and Wicaksono and Myaeng [23] for Japanese data and English Weblogs, respectively. They both defined the problem as a binary classification task (i.e., advice and non-advice labels) using a traditional machine learning model (i.e., SVM). In our work, we also view the problem as a sentence-level classification, but focus on identifying advice-specific features and devising a different method that fully utilizes them. For this goal, we use travel Web forums where useful advice is more concentrated than general Weblogs.

We made an observation on our data and found that advice-revealing sentences tend to appear contiguously in Web forums, which means that there is strong dependency between contiguous sentences in a concentrated region of the text. Based on this observation, we employed a sequence labeling machine learning approach (which is different from previous work) that can naturally model such dependency. To help the machine learning algorithm in classifying the target sentence, we defined three types of feature: syntactic features (e.g., cue patterns based on *class sequential rules* [13], typed dependencies, presence of imperative mood expression, etc.), context features (i.e., features that can leverage dependency information between neighboring positions), and semantic feature (i.e., sentence informativeness). Finally, we propose a new sequence labeling method based on HMM (we call it as F-HMM) that can operate at a more general level and does not always have to rely on presence of words. We show that F-HMM performs better than 2 well-known sequential labeling algorithms (i.e., CRF and SVM$^{\text{hmm}}$) for extracting advice-revealing sentences.

In summary, this paper makes three contributions. First, we propose a task of extracting advice-revealing sentences from Web forums, which was never addressed before, as well as identify features relevant to the task. Second, we define our task as a sequence labeling problem, which is not only necessary for a better modeling purpose but also critical for enhanced performance compared to the traditional machine learning framework. Third, we also propose a new way of using HMM (i.e., F-HMM), which is useful for labeling sequential sentences.

2. RELATED WORK

As far as we know, there are only two previous studies that addressed the problem of extracting advice-revealing sentences. Kozawa et al. [11] proposed methods to extract prior-advice from the Web in order to provide users prior-information before they do a particular activity. Following that work, Wicaksono and Myaeng [23] specifically addressed the problem of extracting advice-revealing sentences from Weblogs. Both studies defined the problem as a binary classification task using SVM.

Our work addresses two different technical issues compared to the aforementioned previous work, which make our task challenging. First, simply applying features proposed by the previous work is not sufficient for our task since Web forums have unique characteristics compared to Weblogs and other online platforms. For example, people's conversations are held in the form of posted messages wrapped by a container so-called "thread". We show that leveraging forum-specific features (e.g., forum-specific cue words, presence of sentence in the first post, etc.) gives significant improvement to our task, which means that it is critical to utilize features of various types specific to the text types. Second, we found that advice-revealing sentences tend to appear contiguously in Web forums, which means that there is strong dependency between contiguous sentences in a concentrated region of the text. Traditional machine learning models used by the previous work obviously cannot deal with this kind of dependency naturally, making it necessary to develop a new model.

Several past studies have also addressed the way to extract other useful knowledge that can be found in Web forums and Weblogs. Park et al. [16] tried to harvest human's experience from Weblogs for experience retrieval and experiential knowledge distillation. They identified linguistically-oriented features for machine learning algorithms. Glance et al. [10] tried to leverage Web forums to develop a marketing and business intelligent application because Web forums usually contain opinions and commentaries about consumer products. Ding et al. [9] and Yang et al. [24] devised methods to detect contexts and answers of the questions in Web forum threads. Their goal is to provide a thread summary as well as enrich the knowledge base of community-based question and answering (CQA) services such as *Live QnA* and *Yahoo! Answers.*

3. PROPOSED METHOD

Before we describe our approach, we formally define our task (as mentioned in Definition 1) to give a better understanding.

Definition 1. **(TASK DEFINITION)**. Given a thread with $|S|$ sentences $\{s_1, s_2, s_3, ..., s_{|S|}\}$, the task of *advice-revealing sentence extraction* aims to determine a prediction function H, which maps a sentence s_i into one of two predefined labels (i.e., advice and non-advice). Formally, we determine a prediction function H so that $Y_i = H(s_i)$, where $Y_i \in \{Advice, NonAdvice\}$.

As mentioned in the previous section, we employ a machine learning approach to tackle the problem, which means that we need to devise different types of features that can characterize advice and non-advice revealing sentences. There are three feature types: syntactic features (e.g., cue patterns, typed dependencies, presence of imperative mood expression, etc.), context features (i.e., features that can leverage dependency information between neighboring positions), and semantic feature (i.e., sentence informativeness). The first three sub-sections of this section (3.1, 3.2, and 3.3) explain our proposed features, while the last sub-section describes our models that exploit the proposed features to solve the problem.

3.1 Discovering Cue Patterns

A class sequential rule (CSR) mining algorithm is a useful data mining technique that can find all labeled sequential patterns with a user-specified minimum support [13]. Intuitively, this algorithm can be naturally applied to find all sequential patterns that frequently appear in the advice as well as non-advice revealing sentences. The discovered patterns that can characterize the advice-revealing sentences are used as our features.

Liu [13] introduced some important definitions regarding CSR. Let $I = \{i_1, i_2, i_3, ..., i_r\}$ be a set of items. An *itemset* X is a non-empty set of items $X \subseteq I$. Furthermore, a *sequence* is defined as an ordered list of itemsets. We denote a sequence s by $< a_1 a_2 ... a_m >$, where a_i is an itemset. a_i is denoted by $\{x_1, x_2, ..., x_k\}$, where $x_j \in I$ is an Item. Items in an itemset are assumed to be in lexicographic order. An item can occur only once in an itemset of a sequence, but it can occur multiple times in different itemsets. For instance, suppose we have sequence $< \{2,3\}\{3,6\}\{4,6\}\{1,3,7\} >$, item 3 occurs three times in three different itemsets (i.e., $\{2,3\}, \{3,6\}, \{1,3,7\}$), but it occurs only once in each aforementioned itemsets. A sequence $s_1 =< a_1 a_2 ... a_m >$ is a subsequence of another sequence $s_2 =< b_1 b_2 ... b_n >$, if there exist integers $1 \leqslant j_1 < j_2 < ... < j_{m-1} < j_m \leqslant n$ such that $a_1 \subseteq b_{j_1}, a_2 \subseteq b_{j_2}, ..., a_m \subseteq b_{j_m}$.

For example, let $I = \{1, 2, 3, 4, 5\}$. The sequence $< \{2,3\} \{5\} >$ is contained in $< \{2\}\{3\}\{2,3,4\}\{4,5\}\{6\} >$ because $\{2,3\} \subseteq \{2,3,4\}$ and $\{5\} \subseteq \{4,5\}$. However, $\{2,3\}$ is not contained in $< \{2\}\{3\} >$ and vice versa. Input for CSR mining algorithm is a sequence database D containing a set of pairs, i.e., $D = \{(s_1, l_1), (s_2, l_2), ..., (s_n, l_n)\}$, where s_i is a sequence and $l_i \in L$ is its respective class. A class sequential rule (CSR), R, is of the form $X \longmapsto l$. An instance (s_i, l_i) in D is said to *cover* R if X is a subsequence of s_i. Moreover, an instance (s_i, l_i) in D is said to *satisfy* R if X is a subsequence of s_i and $l_i = l$. The *support* of R, denoted by $support(R)$, is the fraction of pairs in D that covers R. The *confidence* of R, denoted by $conf(R)$, is the proportion of sequence in D that covers R also satisfies R. In other words, $conf(R)$ represents the probability of R being true.

To construct a sequence database in our case, we process each sentence in our dataset as well as its corresponding label to generate rules in the form of $X \longmapsto l$, where $l \in \{Advice, NonAdvice\}$. To create a sequence X, first, we tokenize the corresponding sentence into a list of words. Second, we only keep pronouns, modal words (e.g., "would", "can", etc.), and cue phrases/words (e.g., "make sure", "i suggest", "recommend", etc.), skipping all others. Third, we use a part-of-speech tag, instead of a word, in every position before and after modal words. For example, the sentence "i would like to recommend" is transformed into "i would VB recommend", where "VB" is a part-of-speech tag. Cue words are usually good indicators for advice-revealing sentences while part-of-speech tags reduce the sparseness of words. For example, suppose we have a sequence database as described in Table 1.

ID	Sequence	Class
r_1	$< you, can, VB >$	*Advice*
r_2	$< you, can >$	*Advice*
r_3	$< you, can >$	*NonAdvice*
r_4	$< make, sure, you >$	*Advice*
r_5	$< i, will, RB >$	*NonAdvice*

Table 1: Example of Sequence Database

Using the minimum support of 50% and minimum confidence of 60%, one of the discovered CSRs is $< you, can > \longmapsto Advice$ with support of 60% and confidence of 66.67%. In brief, given a sequence database D, a minimum support value, and a minimum confidence value, a CSR mining algorithm discovers all CSRs in D. Details of the mining algorithm is explained by Liu [13].

After constructing our sequence database, we run the CSR mining algorithm described by Liu [13] to discover sequential patterns (CSRs). In our experiment, we empirically set minimum confidence at 85% and minimum support at 2 occurrences (0.03%) in the sequence database. Each discovered CSR serves as a binary feature for our models. That is, there are several binary feature functions $\{f_i(s)\}_{i=1}^{m}$ corresponding to their respective CSRs, where m is the number of discovered CSRs. If a sentence s contains a particular CSR, then $f_i(s) = 1$; otherwise $f_i(s) = 0$.

3.2 Sentence Informativeness

Each word carries a different amount of information contributing to the informativeness of a sentence. Bearing in mind that an advice-revealing sentence must be informative to the users, we can utilize the term informativeness theory to define a feature for our task. One of the famous term informativeness measures is *inverse document frequency* (IDF), which was introduced by Sparck-Jones [20]. The rationale behind IDF is that the importance of a term is inversely correlated to the number of documents containing the term in the collection.

Church and Gale [5] introduced the notion of *burstiness*, which is defined as follows: $burst(w) = \frac{tf(w)}{df(w)}$, where $tf(w)$ denotes term frequency in the collection and $df(w)$ is the number of documents containing term w. High burstiness ($tf(w) \gg df(w)$) of a particular word typically shows that the word bears rich content or information since multiple occurences of the word are most likely to "burst" within a small number of documents. They also introduced another term informativeness measure referred to as *Residual IDF*, which is defined as follows: $ridf(w) = idf(w) - \widehat{idf}(w)$, where $\widehat{idf}(w)$ is expected IDF that follows Poisson distribution [6]. They argued that informative terms tend to have high deviation between actual IDF and expected IDF in the collection.

To use a term informativeness measure as one of our features, we introduce the notion of *sentence informativeness measure*, which is simply a summation of informativeness scores of all the *nouns* contained in a sentence. Sentence informativeness value is then used as a single real-valued feature for our models. The rationale behind using nouns is that they are usually content words expressing the topic of a sentence. Alternatively, informativeness of a sentence S is defined as follows.

$$SI(S) = \sum_{i=1}^{N} TI(nw_i), \qquad (1)$$

where $SI(S)$ is an informativeness score of sentence S, N is the number of nouns contained in S, and $TI(nw_i)$ is a term informativeness score of i^{th} noun computed by IDF, burstiness, or Residual IDF. In our case, we treat a forum thread as a "collection" and a sentence in the thread as a "document". In order to incorporate information specific to forum data, we penalize the informativeness score by following the rules below sequentially.

1. *If the sentence is located in the first post, the current sentence informativeness score is penalized by* a%. We found that sentences located in the first post usually do not contain useful advice for readers. First post usually contains questions needed to be answered.

2. *If the sentence is a question sentence, the current sentence informativeness score is penalized by* b%. A question is simply characterized by the appearance of a question mark.

3. *If the sentence is a non-complete sentence, the current sentence informativeness score is penalized by* c%. Incomplete sentences in a Web forum usually express greetings (e.g., "hi all", "hello all"), gratitude (e.g., "thank you for your advice"), hope (e.g., "hope you enjoy your trip"), or even spam. An incomplete sentence can be detected using a dependency parser[1] and an imperative mood detector like the one proposed by Wicaksono and Myaeng [23]. If a non-imperative sentence does not contain a *nominal subject*, denoted by the "nsubj" dependency relation, it means that it is incomplete; otherwise it is considered complete.

where a, b, and c are empirically determined. Based on our experiment, the best performance is achieved when we set a, b, and c more than 70(%).

3.3 Features for Our Model

This sub-section summarizes our proposed features used for our models. The proposed features are categorized into three as described in Table 2. Syntactic features leverage linguistic information of the target sentence to be classified. To determine whether or not a sentence contains an imperative mood expression, we use the heuristic method proposed by Wicaksono and Myaeng [23]. CSRs are discovered using the method previously mentioned in section 3.1. Typed dependencies within a sentence are determined using the Stanford dependency parser [14]. It provides a simple description of the grammatical relationships in a sentence. In our case, we only pay attention to *conjunct*, *clausal subject*, and *nominal subject* relations, which are denoted by "conj", "csubj", and "nsubj", respectively. Forum-specific cue phrases are mostly indicators of non-advice sentences because they are usually expressions of greetings, gratitude, or hope. Typed dependency based features and forum-specific features are essentially binary features. If a sentence contains a particular feature, its corresponding feature function value is set to 1; otherwise it is set to 0.

Syntactic Features	
1.	Whether or not a target sentence contains an imperative mood expression
2.	Discovered class sequential rules (CSRs)
3.	List of a target sentence's typed dependencies
4.	Presence of forum-specific cue phrases such as "thank you", "enjoy your trips", etc. (characterizing non-advice)
Context Features	
1.	Jaccard similarity between a target sentence and its N preceding sentences
2.	Jaccard similarity between a target sentence and its M succeeding sentences
3.	Whether a target sentence and its N preceding sentences are in the same post
4.	Whether a target sentence and its N succeeding sentences are in the same post
Semantic Features	
1.	Sentence informativeness score

Table 2: Features for Our Model

Context features provide information "stored" between neighboring sentences in a forum thread. For example, Jaccard similarity is computed to capture dependency between a target sentence and fixed numbers of its preceding and succeeding sentences (we set $N = 2$). Each similarity feature is a single real-valued feature. Finally, a sentence informativeness score described in section 3.2 is used as a semantic feature since advice-revealing sentences must contain useful information for the users.

3.4 Labeling Sequential Sentences

Based on our further observation, we found that advice-revealing sentences tend to appear contiguously in the Forum data. As in Table 3, we can see that sentence Y_t tends to have the same label with its previous sentence Y_{t-1}. A Chi-square statistical test value of 1,390 ($p - value < 0.001$) indicates general strong dependency between contiguous sentences in a thread, although the likelihood varies with their location in a thread. Therefore, a good model for the problem should be able to capture this dependency well. Unfortunately, traditional machine learning models such as SVM and Maximum Entropy cannot capture this kind of dependency naturally. Even though information about the surrounding tokens can be used as a feature, yet these approaches still have a limitation because the classifier uses fixed size neighbors to classify.

[1] We use Stanford Dependency Parser [14]

	$Y_t = A$	$Y_t \neq A$
$Y_{t-1} = A$	**1683**	503
$Y_{t-1} \neq A$	637	**2076**

Table 3: Dependency Between Contiguous Sentences ($\chi^2 = 1,390, p - value < 0.001$)

Instead of treating our task in Definition 1 as binary classification problem, we see it as a sequence labeling problem to consider the sentence-level dependency. Formally, the output of the problem is a sequence of labels $\mathbf{Y} = (y_1, y_2, ..., y_n)$ which corresponds to an observable sequence $\mathbf{X} = (x_1, x_2, ..., x_n)$. Moreover, each label y_i of a particular token is dependent on the labels of other tokens in the sequence, particularly y_{i-1} and y_{i+1}. Suppose each label y_i can take a value from Σ, then the problem can be seen as a multiclass classification problem with $|\Sigma|^n$ different classes.

In recent years, many researchers in natural language processing area have employed Conditional Random Fields (CRFs) [12] and SVM$^{\text{hmm}}$ [1], which are known to be the state-of-the-art algorithms for solving sequence labeling problems. To solve the problem, CRFs and SVM$^{\text{hmm}}$ form a vector of feature functions $\Phi(\mathbf{X}, \mathbf{Y}) = [\Phi_1 \Phi_2 \Phi_3 ... \Phi_{|\Sigma|} \Phi_{trans}]$, where Φ_i is the feature vector associated with the i^{th} token and Φ_{trans} is the feature vector storing surrounding information. The following scoring function is then used to determine the predicted sequence of labels $\mathbf{Y}^*$.

$$score(\mathbf{X}, \mathbf{Y}; \mathbf{w}) = \mathbf{w}^T \Phi(\mathbf{X}, \mathbf{Y}), \tag{2}$$

where $\mathbf{w}$ is weight vector associated with feature vectors. Finally, given the observable sequence $\mathbf{X}$, the predicted sequence of labels $\mathbf{Y}^*$ is computed as $\max_{\mathbf{Y}} score(\mathbf{X}, \mathbf{Y}; \mathbf{w})$. The weight vector $\mathbf{w}$ is typically trained to optimize the following equation given training examples $\mathbf{T} = \{(\mathbf{Y}^1, \mathbf{X}^1), ..., (\mathbf{Y}^n, \mathbf{X}^n)\}$.

$$\min_{\mathbf{w}} \frac{\lambda}{2} ||\mathbf{w}||^2 + \sum_{(\mathbf{X},\mathbf{Y}) \in \mathbf{T}} L(\mathbf{X}, \mathbf{Y}; \mathbf{w}), \tag{3}$$

where λ is a regularization constant. CRFs use the negative log-likelihood loss function as follows.

$$L(\mathbf{X}, \mathbf{Y}; \mathbf{w}) = -\log \left[\exp \left(\frac{score(\mathbf{X}, \mathbf{Y}; \mathbf{w})}{\mathbf{Z}} \right) \right], \tag{4}$$

where $\mathbf{Z}$ is a normalization factor over all possible sequence of labels. SVM$^{\text{hmm}}$ uses the margin-based loss function as follows.

$$\begin{aligned} L(\mathbf{X}, \mathbf{Y}; \mathbf{w}) = \max_{\overline{\mathbf{Y}}} score(\mathbf{X}, \overline{\mathbf{Y}}; \mathbf{w}) + \Delta(\overline{\mathbf{Y}}, \mathbf{Y}) \\ -score(\mathbf{X}, \mathbf{Y}; \mathbf{w}), \end{aligned} \tag{5}$$

where $\Delta(\overline{\mathbf{Y}}, \mathbf{Y})$ is loss function calculated as the number of tag differences between $\mathbf{Y}$ and $\overline{\mathbf{Y}}$. Particularly, we ran an experiment using the special case of CRFs so-called *Linear CRFs* that make first-order Markov independence assumption.

Hidden Markov Model (HMM) is another statistical tool for modeling sequential data. Although CRFs and SVM$^{\text{hmm}}$ have been proven to be superior to HMM in most cases, HMM is still more efficient in terms of training time in many cases [15, 19, 21]. In contrast to CRFs and SVM$^{\text{hmm}}$ (which are discriminative models), HMM is a generative model that assumes an observable sequence $\mathbf{X} = (x_1, x_2, ..., x_n)$ is generated by the model that gives some information about the sequence of hidden states (labels) $\mathbf{Y} = (y_1, y_2, ..., y_n)$. The goal of HMM is to maximize the joint probability of paired hidden and observable sequence $(\mathbf{Y}, \mathbf{X})$ to find the predicted sequence of labels $\mathbf{Y}^*$. Given Markov assumption, the last statement is formulated as follows.

$$\mathbf{Y}^* = \max_{y_1, y_2, \cdots, y_n} \prod_{i=1}^{n} P(x_i|y_i).P(y_i|y_{i-1}), \tag{6}$$

where y_0 denotes initial marker, $P(x_i|y_i)$ is *emission probability* of an observation x_i being produced from the hidden state y_i and $P(y_i|y_{i-1})$ is *transition probability* of from state y_{i-1} to state y_i. In this model, observations are typically task-appropriate atomic entities, such as words, characters, or nucleotides, where the number of distinct entities is finite. But, in our case, an observable token represents a sentence. As a result, if we just follow the typical use of HMM, the number of distinct tokens in training data can be very large and lead to serious data sparseness problem.

To cope with the data sparseness problem, recent studies have proposed to use a *word-based language model* to estimate the emission probabilities $P(x_i|y_i)$ [3, 18, 17]. It is formulated as follows.

$$P(x_i|y_i) \approx P(\mathbf{W}; L_{c=y_i}) = P(w_1 w_2 ... w_n; L_{c=y_i}), \tag{7}$$

where $\mathbf{W} = w_1 w_2 ... w_n$ is an n-word sentence corresponding to an observable entity x_i and $L_{c=y_i}$ is a word-based n-gram language model trained on data corresponding to class label y_i. An emission probability for a unigram language model is estimated as $P(\mathbf{W}; L_{c=y_i}) = \prod_{i=1}^{n} P(w_i; L_{c=y_i})$ and bigram language model is estimated as $P(\mathbf{W}; L_{c=y_i}) = \prod_{i=1}^{n} P(w_i|w_{i-1}; L_{c=y_i})$. Unfortunately, estimating emission probabilities using a word-based language model is still prone to the data sparseness problem since low-frequency words may occur frequently in the data. Another method that can generalize better to unseen word sequence (i.e., sentence) is then needed.

3.4.1 Feature-based HMM (F-HMM)

We propose a new way of using HMM for labeling sequential sentences. The main idea is that a sentence is represented as a composition of its *top-N features*. By using top-N features, the model can operate at a more general level and does not always have to rely on presence of words since features can be defined at the level that does not suffer from data sparseness. Top-N features can be obtained using any available feature selection method that runs over all features such as those described in section 3.3.

Our approach extends the traditional way of using HMM with directly observable emissions by allowing for arbitrary abstractions of the observations and external features associated with them. Therefore, we call this approach as feature-based HMM. Suppose S and O are non-empty sets, where $S = \{s_1, s_2, ..., s_m\}$ is a list of sentences on our dataset and $O = \{o_1, o_2, ..., o_l\}$ is a list of unique labels, then there exists a *mapping* $\phi : S \to O$ from S into O which assigns to each member of S a unique member in O. The process of determining the mapping ϕ is described as follows.

1. Suppose base feature set $F_N = \{f_1, f_2, ..., f_n\}$ is set of top-N features obtained using any feature selection method that runs over all pre-defined features.

2. The set of features of a sentence s_j, F_{s_j}, is extracted using method described previously. A sentence s_j is now represented as a *k-tuple* $(e_1e_2...e_k)$, where $F_N \bigcap F_{s_j} = \{e_1, e_2, ..., e_k\}$.

3. Each distinct k-tuple in the dataset is then re-labeled with a distinct symbolic identifier $o_1, o_2, ..., o_l$ at the end. Finally, an observable sequence is represented as $\mathbf{X} = (x_1, x_2, ..., x_n)$, where $x_i \in \{o_1, o_2, ..., o_l\}$.

In our case, $\mathbf{X} = (x_1, x_2, ..., x_n)$ is an observation sequence and $\mathbf{Y} = (y_1, y_2, ..., y_n)$ is a set of hidden states, where $x_i \in \{o_1, o_2, ..., o_l\}$ and $y_i \in \{Advice, NonAdvice\}$. For example, suppose we have Top-N features $F_N = \{F_1, F_3, F_6\}$ and a sequence of sentence $S = \{s1, s2, s3, s4\}$. First, suppose we extract the features for each sentence s_i and obtain $F_{s_1} = \{F_1, F_2\}$, $F_{s_2} = \{F_1, F_3, F_7\}$, $F_{s_3} = \{F_1, F_3, F_8, F_9\}$, $F_{s_4} = \{F_2, F_6\}$. Second, we remove all features except F_1, F_3, and F_6 from each F_{s_j} as a *k-tuple*. We then get $F_{s_1} = (F_1)$, $F_{s_2} = (F_1F_3)$, $F_{s_3} = (F_1F_3)$, $F_{s_4} = (F_2)$. Third, we re-label each distinct *k-tuple* using simple identifier o_1, o_2, o_3. Finally, we obtain the mapping function ϕ represented as the ordered pairs $\{(s_1, o_1), (s_2, o_2), (s_3, o_2), (s_4, o_3)\}$.

To estimate an emission probability $P(x_i|y_i)$, we use Maximum Likelihood Estimation (MLE) as described in the following equation since we have annotated data.

$$P(x_i|y_i) = \frac{Count(x_i, y_i) + \alpha}{Count(y_i) + |\sum|.\alpha} \tag{8}$$

$Count(x_i, y_i)$ is the number of times that observation x_i was labeled as y_i in our training data and α is smoothing parameter. Similarly, we also use MLE to estimate a transition probability $P(y_i|y_{i-1})$ as follows.

$$P(y_i|y_{i-1}) = \frac{Count(y_i, y_{i-1})}{Count(y_{i-1})} \tag{9}$$

where $Count(y_i, y_{i-1})$ is the number of times that y_{i-1} is followed by y_i in our training data.

4. EXPERIMENTS

4.1 Data and Evaluation

To collect our data, we crawled several Web forum threads from two well-known travel forums (*InsightVacations*[2] and *Fodors*[3]). We then selected 150 threads randomly from each Web forum so that our dataset consists of a total of 300 threads containing around 5199 sentences (2336 advice sentences and 2863 non-advice sentences). In order to build a ground truth, we asked two human annotators to label sentences as to whether they reveal advice in each thread. The *kappa* statistic for inter-annotator agreement in identifying advice is *0.76*, which means that our definition of advice is quite clear and there is sufficient consensus regarding what advice is. The intersection of the two annotators' judgments was used as the ground truth so that the experimental results are based on the clear cases with minimal ambiguity. We used precision, recall, and F1-score to measure the performance of the proposed method and the baselines. Due to the rather limited size of the dataset, we used 5-fold cross validation.

4.2 Baseline

We implemented two previous works as our baselines, i.e., method proposed by Wicaksono and Myaeng (*baseline #1*) [23] and Kozawa et al. (*baseline #2*) [11] because they also addressed the problem of advice-revealing sentence extraction. While the baseline #1 was applied directly to our dataset, the second one had to be modified because it was developed for Japanese data. Technique to adopt their method was described by Wicaksono and Myaeng [23]. Both previous works defined the problem as binary classification using SVM model. But, they devised different feature sets as mentioned below.

Features used by baseline #1 [23]: (1) set of clue expressions defined through investigating data. (2) proper-noun and modal verb contained in a sentence. (3) set of clue verbs found in the sentence, proper-noun attached to the clue verbs as nominal subject, and POS tags of those clue verbs. (4) presence of imperative mood expression. (5) presence of opinionated copula.

Features used by baseline #2 [11]: (1) first modal verb found in the sentence is any of the following: "could", "might", "must", "shall", "should", "will", and "would". (2) Set of clue expressions defined through observation. (3) Frequency of opinionated words. (4) Through feature 1, 2, and 3 for the previous and next two sentences of the target sentence.

4.3 Scenarios and Results

The first experiment was to validate the features we proposed for the task to determine whether they would increase the level of the baseline machine learning models as well as the proposed one. We evaluated the features using two well-known traditional discriminative machine learning models (i.e., SVM and Maximum Entropy) and made a comparison against two baselines: *baseline #1* [23] and *baseline #2* [11]. Moreover, we also tried three different term informativeness measures. As shown in Table 4, the method of using our features significantly outperforms the two baselines. The improvement is attributed to the use of forum-specific features, which the two baselines do not use. We also notice that "burstiness" gives the best performance compared to the other informativeness measures.

	Model	Prec.	Rec.	F1
Baseline	#1	68.8%	33.8%	45.4%
	#2	60.9%	21.4%	31.6%
IDF	MaxEnt	75.8%	62.7%	68.6%
	SVM	71.7%	68.6%	70.1%
Burstiness	MaxEnt	77.6%	65.0%	70.7%
	SVM	71.4%	71.1%	**71.2%**
Residual IDF	MaxEnt	76.0%	62.3%	68.5%
	SVM	68.8%	69.8%	69.3%

Table 4: (Extraction Performance) Validation of the proposed features and the comparison among the informativeness measures using traditional machine learning models

We computed *Fisher score* values [4] for the features to see top-5 most discriminative features. As shown in Table 5, our "sentence informativeness" acts as the most important feature for the task. In fact, top-10 positions are mostly

[2] http://forums.insightvacations.com

[3] http://www.fodors.com

occupied by syntactic features such as discovered CSRs, imperative mood expression, and forum-specific cue phrases.

Feature	Fisher score
Sentence informativeness	0.236
CSR $< \{you\}, \{VB\} \rightarrow advice >$	0.045
Jaccard similarity between i^{th} sentence (target) and $(i-2)^{th}$ sentence	0.037
Forum-specific cue phrases	0.032
Imperative mood expression	0.031

Table 5: Top-5 Most Discriminative Features

The second experiment was to evaluate sequence labeling models for extracting advice-revealing sentences from Web forums. We only show the results of using the features that gave the best performance in the first experiment, i.e., syntactic, context, and semantic (using burstiness measure) features. First, we ran an experiment using generative models, i.e., HMM. Besides a word-based language model, we also tried a class-based language model that leverages part-of-speech information associated with each word. The class-based language model can group low-frequency words together into equivalence classes (part-of-speech) by estimating emission probabilities with $P(x_i|y_i) \approx P(\mathbf{T}; L_{c=y_i})$, where $\mathbf{T} = t_1t_2...t_n$ are classes (POS) of words $w_1w_2...w_n$, respectively. To implement F-HMM, we used *Fisher score* as our feature selection tool since it is independent of the classifiers. Here, we only show the result of using Top-17 features (**N=17**) since they gave the best performance based on our preliminary experiment. In fact, we tried several values for N from $N = 13$ to $N = 40$, and found that the performance is not much different. Finally, we ran an experiment using discriminative models (Linear CRFs and SVM$^{\text{hmm}}$). The experimental results are summarized in Table 6.

In general, the sequence labeling models performed better than traditional machine learning models for our task since the former can leverage information between contiguous sentences. F-HMM significantly outperformed typical HMM whose emission probabilities are estimated using a language model. Surprisingly, Linear CRFs and SVM$^{\text{hmm}}$ are no better than F-HMM in terms of F1, even when the same top-17 features were applied. The overall improvement was mainly due to the increase in recall. The discriminative models are still strong in precision. Nonetheless, we find the current result very promising because HMM has been shown to be inferior to CRFs and SVM$^{\text{hmm}}$. It calls for further research with different datasets and feature sets. Another point to make is that Linear CRFs and SVM$^{\text{hmm}}$ require large amounts of computations. Considering that applications like our advice extraction system must run for a Web-scale data, this result shows the benefit of using F-HMM since HMM is known to be more efficient in terms of training time compared to Linear CRFs and SVM$^{\text{hmm}}$.

5. CONCLUSIONS AND FUTURE WORKS

For relatively a new task of extracting advice-revealing sentences from Web forums, we identified features that leverage linguistic, contextual, and semantic information of a sentence in order to take advantage of the features of various types. We also proposed F-HMM where the data sparseness problem can be modulated using Top-N features. This is also in line with the fact that sequence labeling models such as CRFs, SVM$^{\text{hmm}}$, and HMM have been shown to be better than traditional machine learning models for solving our problem.

Model	Prec.	Rec.	F1
Hidden Markov Model			
Word-based unigram	61.9%	73.3%	67.1%
Word-based bigram	66.1%	74.1%	69.8%
Class-based unigram	52.3%	80.4%	63.4%
Class-based bigram	56.6%	88.1%	69.0%
F-HMM (N=17)	70.4%	81.5%	**75.6%**
Linear CRFs / SVM$^{\text{hmm}}$			
Linear CRFs	74.5%	74.6%	74.6%
SVM$^{\text{hmm}}$	77.3%	66.2%	71.3%
Linear CRFs (N=17)	74.8%	70.5%	72.6%
SVM$^{\text{hmm}}$ (N=17)	76.4%	66.3%	71.0%

Table 6: (Extraction Performance) Comparison against HMM and other state-of-the-art sequence labeling models

Once those advice-revealing sentences have been successfully extracted, there are three potential applications as future implications or benefits of this task besides providing an easiness for people in accessing travel advice:

1. **Advice retrieval system:** A special-purpose search engine to provide pieces of advice for a specific situation or problem.

2. **Context aware application:** The rapid development of mobile devices such as smart phones and tablet PCs triggers many researches in developing such information recommendation system considering users' contexts. After the system detects the users' contexts, e.g., place and time, the system could provide advice for the users in accordance with current place and time. For instance, user A is going to have lunch at restaurant B. Subsequently, the recommender system may present automatically pieces of advice regarding what kind of menu user A should eat, what kind of menu user A should avoid, or maybe what kind of action user A should avoid. The returned advice is retrieved from extracted advice-revealing sentences depository. The advice was written by other people who have experiences in having lunch at restaurant B before.

3. **Assessment tool:** Extracted advice provides a big boost to the tourist destination marketers. After people visited a particular tourist destination, they usually write down pieces of advice in accordance with their experience on a travel weblog. This advice is manifestation of their experiences being communicated about the strengths and weakness of the destination. Understanding individual travel experiences from extracted advice is clearly a cost-effective method for destination marketers to assess their service quality and improve travelers' overall experiences.

In the future, there are several issues we will address. First, we will extract context sentences of extracted advice-revealing sentences to follow up our work. Second, we will work on how to implement advice mining system on a web-scale data. To implement our idea on the web-scale data, there are some important problems: (1) how to develop depository for a web-scale data, (2) how to increase the size of the training data, (3) how to extract advice from a web-scale data efficiently, (4) how to implement on other data besides online forums. Point 1 can be easily solved since there are many available frameworks for a large-scale depository nowadays. Regarding point 2, we can consider to leverage a bootstrapping approach by increasing our training data using a small set of available annotated data. Regarding point 3, we have shown that our F-HMM can be very useful in this case. Moreover, using the HMM framework can also cover the problem addressed in point 2 because the HMM parameters can be estimated using an Expectation-Maximization (EM) algorithm, which is known as a platform for semi-supervised learning given a small annotated data and large scale incomplete data [8]. To estimate the parameters on large scale incomplete data, the EM algorithm performs several iterations to update the assignment of parameter values. In our case, the first assignment (iteration 0) can be estimated using our small annotated data instead of random assignment. Regarding point 4, we need to devise different features since our current work focuses on only online forums, but we may keep several independent features such as discovered CSRs and imperative mood expression.

6. ACKNOWLEDGMENT

The first author is a grantee of a Korean Government Scholarship from the Ministry of Education, Science and Technology (MEST), Republic of Korea. This research was also partially supported by WCU (World Class University) program under the National Research Foundation of Korea and funded by the MEST of Korea (Project No: R31-30007).

7. REFERENCES

[1] Y. Altun, I. Tsochantaridis, and T. Hofmann. Hidden markov support vector machines. In *The Proceedings of ICML*, 2003.

[2] Travel Industry Association. *Executive Summaries - Travelers Use of the Internet, 2004 Edition*. Washington, DC: TIA, 2005.

[3] R. Barzilay and L. Lee. Catching the drift: Probabilistic content models with applications to generation and summarization. In *The Proceedings of HLT-NAACL*, pages 113–120, 2004.

[4] Y. W. Chen and C. J. Lin. Combining svms with various feature selection strategies. *Feature Extraction*, 207:315–324, 2006.

[5] K. W. Church and W. A. Gale. Poisson mixtures. *Natural Language Engineering*, 1(2):163–190, 1995a.

[6] K. W. Church and W. A. Gale. Inverse document frequency (idf): A measure of deviations from poisson. In *The Proceedings of Third the Workshop on Very Large Corpora*, pages 121–130, 1995b.

[7] Compete, Inc. *Embracing Consumer Buzz Creates Measurement Challenges for Marketers.* December 2006.

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[9] S. Ding, G. Cong, C. Y. Lin, and X. Zhu. Using conditional random fields to extract contexts and answers of questions from online forums. In *The Proceedings of 46th Annual Meeting of the ACL*, pages 710–718, June 2008.

[10] N. Glance, M. Hurst, K. Nigam, M. Siegler, R. Stockton, and T. Tomokiyo. Deriving marketing intelligence from online discussions. In *The Proceedings of the KDD*, pages 419–428, 2005.

[11] S. Kozawa, M. Okamoto, S. Nagano, K. Cho, and S. Matsubara. Advice extraction from web for providing prior information concerning outdoor activities. In *The 4th International Symposium on Intelligent Interactive Multimedia Systems and Services*, pages 251–260, 2011.

[12] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *The Proceedings of ICML*, 2001.

[13] B. Liu. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[14] M.-C. D. Marneffe, B. MacCartney, and C. D. Manning. Generating typed dependency parses from phrase structure parses. *LREC*, 2006.

[15] N. Nguyen and Y. Guo. Comparisons of sequence labeling algorithms and extensions. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pages 681–688, New York, NY, USA, 2007. ACM.

[16] K. C. Park, Y. J. Jeong, and S. H. Myaeng. Detecting experiences from weblogs. In *The Proceedings of the 48th Annual Meeting of the ACL*, pages 1464–1472, 2010.

[17] Z. Qu and Y. Liu. Finding problem solving threads in online forums. In *The Proceedings of the 5th International Joint Conference on Natural Language Processing*, pages 1413–1417, 2011.

[18] Z. Qu and Y. Liu. Sentence dependency tagging in online question answering forums. In *The Proceedings of the 50th Annual Meeting of the ACL*, pages 554–562, 2012.

[19] N. A. Smith, D. L. Vail, and J. D. Lafferty. Computationally efficient m-estimation of log-linear structure models. In *ACL*, 2007.

[20] K. Sparck-Jones. Index term weighting. *Information Storage and Retrieval*, 9:619–633, 1973.

[21] T. L. M. van Kasteren, G. Englebienne, and B. J. A. Kröse. Activity recognition using semi-markov models on real world smart home datasets. *J. Ambient Intell. Smart Environ.*, 2(3):311–325, Aug. 2010.

[22] M. Van-Setten, S. Pokraev, and J. Koolwaaij. Context-aware recommendations in the mobile tourist application compass. In *The Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-based Systems*, pages 235–244, 2004.

[23] Alfan F. Wicaksono and S. H. Myaeng. Mining advices from weblogs. In *The Proceedings of 21st ACM CIKM*, pages 2347–2350, 2012.

[24] W. Y. Yang, Y. Cao, and C. Y. Lin. A structural support vector method for extracting contexts and answers of questions from online forums. In *The Proceedings of the 2009 Conference on EMNLP*, volume 2, pages 514–523, 2009.