# Toward Advice Mining: Conditional Random Fields for Extracting Advice-revealing Text Units

[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 are platforms for personal communications on sharing information with others. Such information is often expressed in the form of advice. In this paper, we address the problem of *advice-revealing text unit* (**ATU**) extraction from online forums due to its usefulness in travel domain. We represent advice as a two-tuple comprising an advice-revealing sentence and its context sentences. To extract the advice-revealing sentences, we propose to define the task as a sequence labeling problem, using three different types of features: syntactic, contextual, and semantic features. To extract the context sentences, we propose to use a *2 Dimensional CRF* (2D-CRF) model, which gives the best performance compared to traditional machine learning models. Finally, we present a solution to the integrated problem of extracting both advice-revealing sentences and their respective context sentences at the same time using our proposed models, i.e., *Multiple Linear CRF* (ML-CRF) and *2 Dimensional CRF Plus* (2D-CRF+). The experimental results show that ML-CRF performs better than any other models studied in this paper for extracting advice-revealing sentences and context sentences.

## 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, Conditional Random Field

## 1. INTRODUCTION

Web forums contain a huge number of posts generated by millions of internet users and grow every day. They enable the users to easily discuss with others various topics as well as share their personal experiences and thoughts on designated topics. 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 cameras they purchased, interesting books they read, or the latest film they watched. Web forums often contain explicit key lessons and know-how's gleaned from people's past experiences which are really worthy to be well selected and presented to other people and/or intelligent agents to help providing context-sensitive assistance. Such key lessons are often expressed in the form of *advice*.

In the travel domain, for example, travelers usually seek pieces of advice from travel Web forums before they visit some tourist places [2, 6]. They provide a 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 a mobile device [23]. In fact, we chose the travel domain for our research with some potential applications in mind, such as advice retrieval system for travelers, context-aware advice generation, and tourism marketers' assessment tools.

As part of an effort to mine experiential knowledge, we address a new problem referred to as advice mining in which advice is extracted and aggregated from Web forum, and subsequently stored with well-organized indices. As the first step toward a full-fledged advice mining, we tackle the problem of extracting an *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 sentence:** a sentence that explains or clarifies an advice-revealing sentence in more detail with contextual information. It usually describes when, where, or in what situation people would find the advice-revealing sentence relevant and potentially helpful.

While a finer-level analysis of an advice sentence would help mining pieces of advice, this capability can serve the purpose of retrieving advice-containing articles with the advice anchored in a particular context for human consumption as well as for further processing to result in a formal representation of advice broken down into smaller elements for inferencing. Below are examples of advice-revealing sentences

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
*CIKM'13,* Oct. 27–Nov. 1, 2013, San Francisco, CA, USA.
Copyright 2013 ACM 978-1-4503-2263-8/13/10 ...$15.00.
http://dx.doi.org/10.1145/2505515.2505520.

as well as their corresponding context sentences extracted from well-known Web forums.

1. **advice-revealing sentence:** "*But you are advised to exchange only a small amount at the airport as you won't get a good rate there*"; **context sentence:** "*I am planning a 2 week trip to Bangalore and am not sure whether it is better to change my Euros here or to wait until I get to India*".

2. **advice-revealing sentence:** "*Instead of standing on a long line with your purchases, just take your receipt to a VAT Tax machine, which will scan the bar code on it and validate it*"; **context sentence:** "*There are several things all travelers to Paris (and Europe generally) should know*".

Constructing an **ATU** constitutes solving two problems: (1) *advice-revealing sentence extraction* and (2) *context sentence extraction* in the text region. For the advice-revealing sentence extraction problem, we propose to define it as a sequence labeling problem using three different types of features: syntactic, contextual, and semantic features. There are two reasons why we treat it as a sequence labeling problem rather than a binary classification. First, we found that advice-revealing sentences tend to appear contiguously in a forum thread. Second, we saw a potential to apply the *Skip-Chain CRF* model that considers dependency between repeated mentions, where our *sentence generalization* method is employed to find redundant sentences and construct the skip-edges.

Knowing only advice-revealing sentences without their contexts is not enough for the advice mining task. For example, a forum thread contains an advice-revealing sentence "*instead of standing on a long line with your purchases, just take your receipt to a VAT Tax machine, which will scan the bar code on it and validate it*". Storing only this sentence in advice repository would create problems when it is asked for because the system would not know when or in what situation the advice could be relevant and useful. If the system also stores the corresponding context sentence like "*there are several things all travelers to Paris (and Europe generally) should know*" (which is also located in the same thread), the advice-revealing sentence becomes clearer and can be given only to the people who are or will be in *Paris*. Therefore, extracting context sentences is essential to making advice-revealing sentences useful. While the context sentence extraction problem can be solved with a regular binary classifier using traditional machine learning models, we propose to use the 2D-CRF model since it can capture dependency between any contiguous position in the context sentence candidate sequences. Our experimental results shows that the 2D-CRF gives the best performance compared to traditional machine learning models for extracting context sentences.

Finally, We present a novel solution that extracts advice-revealing sentences and their respective context sentences at the same time in a single pass by using two new CRF-based models referred to as *Multiple Linear CRF* (ML-CRF) and *2 Dimensional CRF Plus* (2D-CRF+). ML-CRF and 2D-CRF+ can model two types of dependency: (1) dependency between contiguous positions in two different candidate sequences (*advice-revealing sentence candidate sequence* and *context sentence candidate sequence*), (2) dependency between these two types of candidate sequence. In brief, our main contribution is to define and cope with a new problem referred to as *advice-revealing text unit (**ATU**) detection from Web forums*, which was never addressed in the past.

## 2. RELATED WORK

Works on extracting useful knowledge have been done in the past. Park et al. [17], and Inui et al. [9] tried to harvest human's experience from Weblogs for experience retrieval and experiential knowledge distillation. Moreover, we have also seen the flourishing of research in the area of opinion mining in the recent years [16]. Computational approaches to opinion mining currently eschew a general theory of emotions and focus on extracting the affective content of a text from the detection of expressions of sentiment [1]. Asher et al. [1] mentioned that it is important for NLP systems to go beyond positive and negative sentiment expressions and identify a wide range of opinion expressions. Moreover, they categorize opinion expressions using a typology of four top-level categories: Sentiment expressions, which is the current focus of opinion mining, Reporting expressions, Judgment expressions, and Advice expressions, which urge the reader to adopt a certain course of action or opinion [1]. They hope that opinion mining researchers will go beyond these four categories in the future. Our work contributes in extracting advice expressions together with their contexts from texts, especially online forums, which was never addressed before.

While there is no previous work that addressed the same problem like ours, two previous studies addressed only 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. Wicaksono and Myaeng [24] addressed the problem of extracting advice-revealing sentences from English Weblogs. Our work has two unique technical challenges compared to the previous work. First, since Web forums have unique characteristics compared to Weblogs and other online platforms, we face the problem of handling categorically different features for an optimal performance, which may require a new computational model. For example, we found that advice-revealing sentences tend to appear contiguously in forum threads, which means that devising a model that is capable of capturing this information is very crucial. Second, we face a new problem of extracting context sentences that correspond to advice-revealing sentences.

## 3. OUR DATA

To construct a collection for the experiments, we crawled Web forum threads from two well-known travel forums (*InsightVacations*[1] and *Fodors*[2]). We then selected 150 threads randomly from each Web forum to result in a dataset of 300 threads containing 5199 sentences. Two annotators were asked to label advice-revealing sentences and their corresponding context sentences. The *kappa* statistic for inter-annotator agreement is 0.76 in identifying advice-revealing sentences, and 0.68 in identifying context sentences. We used the intersection of the two annotators' judgments for the advice-revealing sentence extraction task and the union of the two annotators' judgments for the context sentence extraction task. Table 1 shows our data in detail. On aver-

[1]http://forums.insightvacations.com
[2]http://www.fodors.com

age, each thread has 7 advice-revealing sentences and each advice-revealing sentence has 3 context sentences.

| Type | # | Total # |
|---|---|---|
| Advice-revealing sentence | 2,336 | 5,199 |
| Non Advice-revealing sentence | 2,863 | |
| pair of {advice, context} | 7,360 | 71,783 |
| pair of {advice, non-context} | 64,423 | |

**Table 1: Our Dataset**

# 4. ADVICE-REVEALING SENTENCE EXTRACTION

This section focuses on how to automatically extract advice-revealing sentences from online forums. To give a better understanding of this task, we formally define it as follows: 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 $Y_i \in \{Advice, NonAdvice\}$.

We propose a machine learning approach to automatically extract advice-revealing sentences from Web forums, which means that devising good features that can characterize advice as well as non-advice revealing sentences is a very important process.

## 4.1 Features for Our Model

The features we defined for our machine learning model are categorized into three as described in Table 2.

| | |
|---|---|
| **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 M succeeding sentences are in the same post |
| **Semantic Features** | |
| 1. | Sentence informativeness score |

**Table 2: Features for Our Model**

**Syntactic Feature**. 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 [24].

Class sequential rules (CSRs) are discovered using CSR mining algorithm, which is known as a data mining technique that can find all labeled sequential patterns with a user-specified minimum support [13]. 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 $f_i(s) = 0$. Intuitively, the discovered patterns can act as good cue patterns since they appear frequently in either advice-revealing sentences or non-advice-revealing sentences. 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.

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.

**Context Feature.** 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. Each similarity feature is a single real-valued feature.

**Semantic Feature.** Each word actually carries a different amount of information contributing to the informativeness degree of a sentence. Bearing in mind that an advice-revealing sentence must be informative to the users, we can obviously leverage term informativeness theory to define one of our features for this classification task. There are several well-known term informativeness measures such as *inverse document frequency* (IDF) [20], *burstiness* [4], and *residual IDF* [5]. In our experiment, we used these three measures as basis for our semantic feature.

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.

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

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 computing $TI(w)$, we make a distinction between local and global informativeness. For $TI_{local}(w)$ representing *local informativeness score*, we treat each sentence in a forum thread as a single document and a forum thread as one collection of documents. For $TI_{global}(w)$ representing *global informativeness score*, however, the sentences contained in the whole forum threads are now treated as one collection of documents. Now, $TI(w)$ is computed by combining both local and global informativeness scores as in the following equation.

$$TI(w) = (1-\alpha).TI_{global}(w) + \alpha.TI_{local}(w), \qquad (2)$$

where $\alpha$ is set empirically[3]. The rationale behind using this combination is that a term sometimes seems to be very important with respect to a particular forum thread (local information), but not necessarily from the global view. In order to incorporate forum-specific information, we penalize the informativeness score if the sentence has at least one of the following conditions: *located in the first post*, *a question sentence*, or *incomplete*.

## 4.2 Using Linear CRF

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. Therefore, instead of treating the task as binary classification using traditional machine learning models (as in [11, 24]), we see it as a sequence labeling problem to consider the sentence-level dependency.

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

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

Linear Conditional Random Field (Linear CRF) [12] is known to be a state-of-the-art algorithm for solving sequence labeling problems. In recent years, many researchers in the natural language processing area have successfully employed Linear CRF to solve their problems such as syntactic parsing and named entity recognition [22, 15]. In brief, given an observable sequence $\mathbf{X} = (x_1, x_2, ..., x_n)$, where $n$ is the number of sentences in a post, the goal is to find the sequence of hidden labels $\mathbf{Y} = (y_1, y_2, ..., y_n)$ using a conditional distribution function as follows.

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})}\left(\prod_{i=1}^{n-1}\Phi(y_i, y_{i+1}, \mathbf{x}, i)\right), \qquad (3)$$

[3]We set $\alpha$ to 0.6 in our experiment

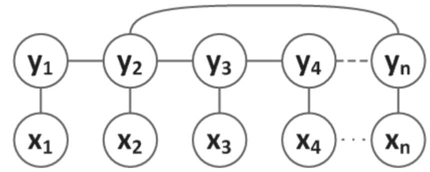


**Figure 1: Skip-Chain CRF**

where $\{\Phi\}$ are the potentials over several feature functions as well as their respective weights, and $Z(\mathbf{x})$ is a normalization factor.

## 4.3 Using Skip-Chain CRF

When the same entity is mentioned more than once in a sequence, in many cases all entity mentions have the same label. We can take advantage of this fact by favoring labelings that treat repeated entities identically, and by propagating pieces of evidence from all entity mentions so that the extraction decision can be made based on global information (all mentions separated by long-range positions). However, Linear CRF cannot take advantage of this dependency since it is based on the markov assumption among labels (i.e., dependency between nearby nodes only). The goal of Skip-Chain CRF is to relax this assumption by modeling dependency between distant nodes as well as nearby nodes [21]. Figure 1 shows the graphical model of Skip-Chain CRF. The distribution over hidden labels is defined as follows.

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})}\left(\prod_{i=1}^{n-1}\Phi(y_i, y_{i+1}, \mathbf{x}, i)\right). \left(\prod_{(u,v)\in\tau}\Theta(y_u, y_v, \mathbf{x})\right), \qquad (4)$$

where $\tau$ is a set of skip-edges, $\{\Phi\}$ are the potentials over the linear-chain edges, $\{\Theta\}$ are the potentials over the skip-edges, and $Z(\mathbf{x})$ is the normalization factor.

In our case, skip-edges cannot be constructed and found easily since few entities representing sentences are identical. We need to operate at a more general level to ensure that a sufficient number of skip-edges exist for a meaningful modeling. To do that, we propose to use *sentence generalization* with top-N features in constructing skip-edges between sequential sentences in our data. Suppose $S$ and $O$ are non-empty sets, where $S = \{s_1, s_2, ..., s_m\}$ is a list of sentences in our dataset and $O = \{o_1, o_2, ..., o_l\}$ is a list of unique labels. Then, there exists a *mapping* $\phi : S \rightarrow O$ from $S$ into $O$ which assigns to each member of $S$ a unique member in $O$. The process of determining the mapping $\phi$ is described as follows.

1. We obtain top-N feature set $F_N = \{f_1, f_2, ..., f_n\}$ using any feature selection method that runs over all predefined features.

2. We extract $F_{s_j}$, i.e., the set of features of a sentence $s_j$. 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, skip-edges are constructed between any pair of two positions $(y_u, y_v)$, where $\phi(s_u) = \phi(s_v)$.

# 5. CONTEXT SENTENCE EXTRACTION

In this section, we aim to extract context sentences for each extracted advice-revealing sentence. Here, we assume that all real advice-revealing sentences have been extracted using the method described in the previous section.

In our definition, context sentences are located in the same thread where their corresponding advice-revealing sentences are extracted. An advice revealing sentence may correspond to more than one context sentences, and vice versa. Moreover, an advice revealing sentence may act as a context sentence for other advice-revealing sentence. The context sentence extraction task can be formally defined as follows: given a set of sentences $S = \{s_1, s_2, ..., s_n\}$ in a thread, and a set of advice-revealing sentences $A = \{a_1, a_2, ..., a_m\}$ found in the thread, where $a_i \in S$ and $m \leq n$, the task is to find whether a pair $\{(a_i, s_j)|a_i \in A, s_j \in S\}$ is an **ATU** consisting of an advice-revealing sentence and its context sentence.

We propose two different methods for extracting context sentences. Both of the methods are based on supervised machine learning in which the models are trained using the predefined features.

## 5.1 Extracting Context Sentences Individually

For a forum thread of M advice-revealing sentences, this method performs M runs of a context extraction algorithm for each advice-revealing sentence. Given an advice-revealing sentence $a_i$ in a thread, a traditional machine learning model (e.g., SVM and Maximum Entropy) is employed to detect whether or not each sentence $s_j \in S$ in the same thread is a context sentence of $a_i$. The features for extracting context sentences are listed below.

**Feature-1.** Jaccard similarity between an advice-revealing sentence and its corresponding context sentence candidate. The rationale behind this feature is that an advice-revealing sentence and its context sentence would share similar words.

**Feature-2.** Semantic similarity between an advice-revealing sentence and its corresponding context sentence candidate. This similarity measure is used to bridge the lexical gap between an advice-revealing sentence and its context sentence candidates. We use a sentence relatedness measure provided in Open Roget's project [10].

**Feature-3.** Whether a context sentence candidate is located before its corresponding advice-revealing sentence in a forum thread. Based on our observation, context sentences tend to appear before the corresponding advice-revealing sentence.

**Feature-4.** Whether a context sentence candidate has not been identified as an advice-revealing sentence.

**Feature-5.** Whether a context sentence candidate is incomplete. Incomplete sentences usually do not contain meaningful information since they usually represent greetings (e.g., "*good morning all*") or hopes (e.g., "*enjoy your trip*"). An incomplete sentence can be detected using a dependency parser[4] and an imperative mood detector like the one proposed by Wicaksono and Myaeng [24]. 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.

[4]We use Stanford Dependency Parser [14]

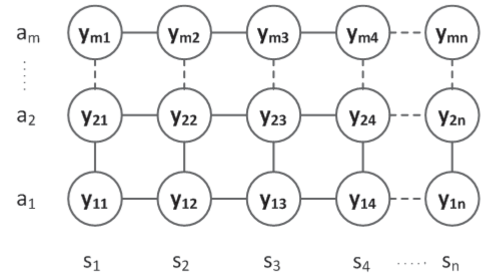


**Figure 2: 2 Dimensional Conditional Random Field (2D-CRF)**

**Feature-6.** Whether a context sentence candidate contains a proper noun. A proper noun usually represents a named entity such as *country-name*, *city-name*, *date*, *organization*, etc. This is obviously a good indicator for a context sentence.

## 5.2 Extracting Context Sentences Together

The method described in the previous sub-section is a straightforward application of traditional machine learning based on the features we identified. It is based on the assumption that there is no dependency in between context sentence candidate sequences that correspond to their respective advice-revealing sentences. This assumption does not necessarily hold as we have shown the nature of forum threads that tend to have dependency between contiguous positions. To capture such dependencies, we use 2 Dimensional Conditional Random Field (2D-CRF). Using 2D-CRF enables us to extract context sentences for all advice-revealing sentences together. The 2D-CRF has been used for a few natural language processing tasks. Ding et al. [8] employed 2D-CRF to detect contexts and answers of questions in forum threads. Qu and Liu [18] also used 2D-CRF to label the dependency relation between sentences in forum threads.

The graphical model of 2D-CRF is shown in Figure 2 (only the hidden states). There are $m$ rows and $n$ columns in the graphical model. For the **ATU** extraction problem, the $i^{th}$ row and $j^{th}$ column correspond to an advice-revealing sentence and a sentence in a forum thread, respectively. One row represents one pass of extracting context sentences for a particular advice-revealing sentence using the traditional machine learning model.

2D-CRF described in Figure 2 consists of $m$ chains, where $y_{i,j}$ is the variable in chain $i$ at time $j$. The clique indices for this 2D-CRF are of the form $\{(i,j),(i,j+1)\}$ for each of the *within-chain edges* and $\{(i,j),(i+1,j)\}$ for each of the *between-chain edges*. The distribution over hidden labels is defined as follows.

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})}\left(\prod_{j=1}^{n-1}\prod_{i=1}^{m}\Phi(y_{i,j}, y_{i,j+1}, \mathbf{x}, j)\right)\cdot \left(\prod_{j=1}^{n}\prod_{i=1}^{m-1}\Psi(y_{i,j}, y_{i+1,j}, \mathbf{x}, j)\right), \tag{5}$$

where $\{\Phi\}$ are the potentials over the within-chain edges, $\{\Psi\}$ are the potentials over the between-chain edges, and $Z(\mathbf{x})$ is the normalization factor. The potentials factorize according to the features $\{f\}$ and weights $\{\lambda\}$ as in the following formulation.

$$\Phi(.) = exp\left(\sum_k \lambda_k^a f_k^a(y_{i,j}, y_{i,j+1}, \mathbf{x}, j)\right), \quad (6)$$

$$\Psi(.) = exp\left(\sum_g \lambda_g^b f_g^b(y_{i,j}, y_{i+1,j}, \mathbf{x}, j)\right), \quad (7)$$

where $\{f_k^a\}$ are features for within-chain edges that exploit dependency between contiguous positions in a thread and $\{f_g^b\}$ are features for between-chain edges that exploit dependency between advice-revealing sentences. For $\{f_k^a\}$ and $\{f_g^b\}$, we use the same feature set as for context sentence extraction using traditional machine learning models.

# 6. EXTRACTING ADVICE-CONTEXT SENTENCES IN ONE PASS

In the previous section, we tackle the advice-revealing sentence extraction and the context sentence extraction tasks separately. In Section 5, we specifically mention the methods for tackling context sentence extraction task by assuming that advice-revealing sentences have been perfectly extracted (with 100% accuracy) previously. Considering that the advice-revealing sentence extraction task is still far from perfect, it is desirable to find a new robust method that can handle both of the tasks together and reflect the reality. We propose a solution, in which we extract advice-revealing sentences and their corresponding context sentences at the same time (in one pass). To do that, we propose to use two new CRF-based models referred to as Multiple Linear CRF (ML-CRF) and 2D-CRF Plus (2D-CRF+). ML-CRF and 2D-CRF+ can model two types of dependency: (1) dependency between contiguous positions in two different candidate sequences, i.e., *advice-revealing sentence candidate sequence* and *context sentence candidate sequence*, (2) dependency between these two types of candidate sequence. The first dependency can be modeled using previously mentioned CRF-based methods such as Linear CRF, Skip-Chain CRF, and 2D-CRF when we extract advice-revealing sentences and context sentences separately. However, the second dependency can only be modeled when we extract both advice-revealing sentences and context sentences at the same time.

We define our new task as follows: given a set of sentences $\mathbf{S} = \{s_1, s_2, ..., s_n\}$ in a thread, the task is to discover two sets: (1) a set of advice-revealing sentences $\mathbf{A} = \{a_1, a_2, ..., a_m\}$, and (2) a set of context sentence lists $\mathbf{C} = \{\mathbf{c_1}, \mathbf{c_2}, ..., \mathbf{c_m}\}$, where $a_i \in \mathbf{S}$, $m \leqslant n$, $\mathbf{c_i} = \{c_{i,1}, c_{i,2}, ..., c_{i,k}\}$ is a set of context sentences for $a_i$ (i.e., a pair $\{(a_i, c_{i,j}) | a_i \in \mathbf{A}, c_{i,j} \in \mathbf{c_i}\}$ is an **ATU**), $c_{i,j} \in \mathbf{S}$, and $k \leqslant n$. This task definition is similar to those in other previous studies such as questions-answers detection from Online forums and Email conversations [7, 19] and contexts-questions detection from Online forums [8]. It means that our ML-CRF and 2D-CRF+ also be applied for these tasks.

Previously, the definition of context sentence only applies to the sentences that explain or clarify an advice-revealing sentence in more detail, which means that we only focus on extracting context sentences for those advice-revealing sentences that have been extracted before. We cannot apply such definition on our new task since the advice-revealing sentences and context sentences are unknown when the extraction algorithm is executed at the first time. To solve this problem, we need to define "context sentence" in a more abstract level. Based on this abstraction, all sentences in a Web forum thread (advice and non-advice revealing sentences) can have their respective context sentences. Another problem was then raised since our dataset only contains annotated context sentences for advice-revealing sentences. We did not hire annotators to label context sentences for non-advice-revealing sentences. Therefore, we had to complete our dataset before applying ML-CRF and 2D-CRF+ for tackling the new task.

To complete our dataset, we employed semi-supervised approach, which means that annotating context sentences for non-advice-revealing sentences was done automatically using a machine learning algorithm. Moreover, the automatically labeled data is used only for the training, which is in fact more desirable than requiring them to be done manually, and has nothing to do with the evaluation result. The machine learning model was then trained using all available context sentences (for advice-revealing sentences) that had been manually tagged by our annotators. We modified the features described in Section 5.1 for our machine learning model. We omitted feature-4 since advice-revealing sentences were unknown. We generalized Feature-1, Feature-2, and Feature-3 so that they applies for all sentences in a forum thread. Moreover, we employed a Linear CRF as the machine learning model for exploiting the modified features since it gave the best performance compared to traditional machine learning models (i.e., SVM and maximum entropy) (this will be shown in Section 7.2).

## 6.1 Using Multiple Linear CRF

Multiple Linear Conditional Random Field (ML-CRF) is an undirected model that encode a conditional probability distribution between a state sequence $\mathbf{y^{(1)}} = \{y_j^{(1)}\}$ and multiple state sequences $\{\mathbf{y_i^{(2)}}\}$, given the an observation sequence $\mathbf{x} = \{x_j\}$, in which a state $y_j^{(1)}$ is associated with a state sequence $\mathbf{y_j^2} = \{y_{j,k}^{(2)}\}$. For the **ATU** extraction problem, $y_j^{(1)} \in \{Advice, NonAdvice\}$ and $y_{j,k}^{(2)}$ corresponds to a candidate of a context sentence for the sentence $x_j$, where $y_{j,k}^{(2)} \in \{Context, NonContext\}$. As shown in Figure 3, there are $n$ rows and $n$ columns in the graphical model (only the hidden states), where $n$ is the number of sentences in a forum thread. One row (in the unshaded area) represents one pass of extracting context sentences for the corresponding candidate of an advice-revealing sentence; meanwhile, the shaded sequence, which transverses through the diagonal area (see $\{y_{i,i}^{(2)}\}_{i=1}^n$), represents one pass of extracting advice-revealing sentences.

Specifically, we use $\mathbf{ML\text{-}CRF_m}$ to denote an ML-CRF in which dependency edges are constructed between a state $y_i^{(1)}$ and several states $\{y_{j,i}^{(2)}\}_{j \in Q}$, where $Q = \{x | x \in \mathbb{N}_{>0} \wedge |i - x| \leqslant m\}$. We call $m$ as the *value of sliding window edge*. This edge construction may reveals the dependency between a state in the advice-revealing sentence candidate sequence ($\mathbf{y^{(1)}}$) and a corresponding state in a context sentence sequence as well as its neighboring context sentence sequences. Figure 3 shows an example of ML-CRF$_1$, in which, for example, edges are constructed for the following three pairs: $\{y_2^{(1)}, y_{1,2}^{(2)}\}$, $\{y_2^{(1)}, y_{2,2}^{(2)}\}$, $\{y_2^{(1)}, y_{3,2}^{(2)}\}$. Finally, the distribution over hidden states is defined as follows:

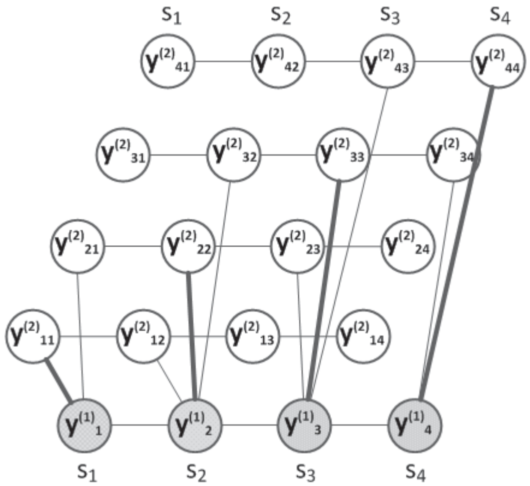

Figure 3: Multiple Linear CRF ($\mathbf{ML\text{-}CRF_1}$)

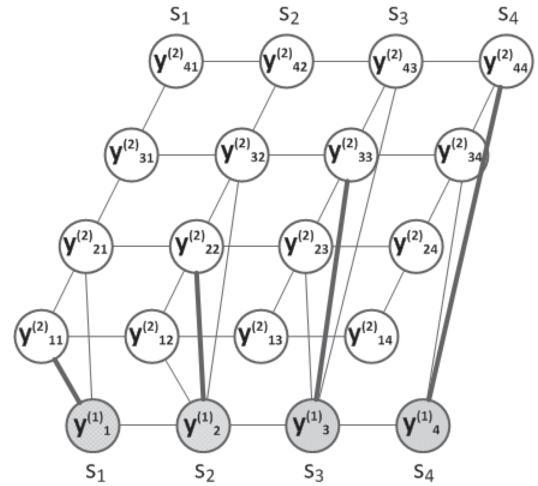

Figure 4: 2 Dimensional CRF+ ($\mathbf{2D\text{-}CRF_1+}$)

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})}\left(\prod_{j=1}^{n-1}\prod_{i=1}^{n}\Phi(y_{i,j}^{(2)}, y_{i,j+1}^{(2)}, \mathbf{x}, j)\right) \cdot \left(\prod_{j=1}^{n}\prod_{i\in Q}\xi(y_j^{(1)}, y_{i,j}^{(2)}, \mathbf{x}, j)\right) \cdot \left(\prod_{j=1}^{n-1}\delta(y_j^{(1)}, y_{j+1}^{(1)}, \mathbf{x}, j)\right), \quad Q = \{k|k \in \mathbb{N}_{>0} \wedge |j-k| \leqslant m\}, \tag{8}$$

where $n$ is the number of sentences in a forum thread, $m$ is the value of sliding window edge, $\{\Phi\}$ are the potentials over the edges in the context sentence candidate sequences, $\{\xi\}$ are the potentials over the edges in between two types of sequence (this depends on the value of $m$), $\{\delta\}$ are the potentials over the edges in the advice-revealing sentence candidate sequence, and $Z(x)$ is the normalization factor. The potentials factorize according to the features $\{f\}$ and weights $\{\lambda\}$ as in the following formulation.

$$\Phi(.) = exp\left(\sum_m \lambda_m^{(2)}.f_m^{(2)}(y_{i,j}^{(2)}, y_{i,j+1}^{(2)}, \mathbf{x}, j)\right) \cdot exp\left(\sum_g \lambda_g^{(1)}.f_g^{(1)}(y_j^{(1)}, y_{j+1}^{(1)}, \mathbf{x}, j)\right), \tag{9}$$

$$\xi(.) = \delta(.) = exp\left(\sum_m \lambda_m^{(2)}.f_m^{(2)}(y_{i,j}^{(2)}, y_{i,j+1}^{(2)}, \mathbf{x}, j)\right), \tag{10}$$

where $\{f_g^{(1)}\}$ is a set of features for advice-revealing sentence extraction (see Table 2), $\{f_m^{(2)}\}$ is a set of features for context sentence extraction (see Section 5.1), and $\{\lambda_g^{(1)}\}$ and $\{\lambda_m^{(2)}\}$ are the respective weights for each of two feature sets. In this integrated solution, we modified features for context extraction described in Section 5.1 since advice revealing sentences are unknown when the extraction algorithm is executed at the first time. We omitted feature-4 and generalized Feature-1, Feature-2, and Feature-3 so that they applies for all sentences in a forum thread. As shown in Equation 9, ML-CRF also allows a state $y_{i,j}^{(2)}$ in the context sentence candidate sequence to utilize features of a state $y_j^{(1)}$ in the advice-revealing sentence candidate sequence.

## 6.2 Using 2D-CRF+

A 2 Dimensional Conditional Random Field Plus (2D-CRF+) is similar to the ML-CRF model. The 2D-CRF+ also encodes a conditional probability distribution between a state sequence $\mathbf{y^{(1)}} = \{y_j^{(1)}\}$ and multiple state sequences $\{\mathbf{y_i^{(2)}}\}$, given the an observation sequence $\mathbf{x} = \{x_j\}$, in which a state $y_j^{(1)}$ is associated with a state sequence $\mathbf{y_j^2} = \{y_{j,k}^{(2)}\}$. Moreover, the application of the 2D-CRF+ for the **ATU** extraction problem is completely the same as the application of the ML-CRF for the same problem. The difference is that the 2D-CRF+ models dependency between two contiguous context sentence candidate sequences by constructing between-chain edges, just like the 2D-CRF described in Section 5.2.

Specifically, we use $\mathbf{2D\text{-}CRF_m+}$ to denote a 2D-CRF+ in which dependency edges are constructed between a state $y_i^{(1)}$ and several states $\{y_{j,i}^{(2)}\}$, where $Q = \{x|x \in \mathbb{N}_{>0} \wedge |i-x| \leqslant m\}$ and $m$ is the value of sliding window edge. Figure 4 shows an example of 2D-CRF$_1$+ (only the hidden states). We need to modify Equation 8 by adding a new term $\prod_{j=1}^{n}\prod_{i=1}^{n-1}\Psi(y_{i,j}^{(2)}, y_{i+1,j}^{(2)}, \mathbf{x}, j)$ to define the distribution over hidden states, in which $\{\Psi\}$ are potentials over the between-chain edges in the context sentence candidate sequences. The factorization of the potentials $\{\Psi\}$ is completely the same as $\{\Phi\}$ (i.e., $\Psi(.) = \Phi(.)$), which means that we just transfer all features from within-chain edges into between-chain edges in the context sentence candidate sequences.

# 7. EXPERIMENTS

Several experiments were carried out to see the performances of our models. 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.

### 7.1 Results for Advice-revealing Sentence Extraction

For baselines, we implemented two baselines: *baseline #1* [24] and *baseline #2* [11]. Both baselines defined the problem as binary classification using SVM model. Baseline #2 used various Japanese-specific linguistic features and baseline #1 introduced several features for English data, such as the presence of imperative mood expression and opinionated copula. While the baseline #1 was applied directly to our dataset, the second one had to be modified because it was developed for Japanese data. The performances of the two baselines are shown in Table 4.

For the case of using local information alone, we ran an experiment using two state-of-the-art traditional machine learning models (i.e., SVM and Maximum Entropy) and a Linear CRF model. We used all the proposed features but only local information for sentence informativeness measure. Moreover, we also tried three different term informativeness measures in this experiment. The results are very promising as in Table 4. First, our proposed method significantly outperforms the two baselines, perhaps because we leveraged forum-specific information as features. Second, the Linear CRF model outperforms the two traditional machine learning models across all the different informativeness measures because it can leverage dependency between contiguous sentences. The result also signifies that it is a good idea to treat the problem as sequence labeling problem rather than a binary classification problem.

| Model | Prec. | Rec. | F1 |
|---|---|---|---|
| Baseline #1 | 68.8% | 33.8% | 45.4% |
| Baseline #2 | 60.9% | 21.4% | 31.6% |
| **IDF (local)** | | | |
| MaxEnt | 75.8% | 62.7% | 68.6% |
| SVM | 71.7% | 68.6% | 70.1% |
| Linear CRF | 72.4% | 71.2% | **71.8%** |
| **Burstiness (local)** | | | |
| MaxEnt | 77.6% | 65.0% | 70.7% |
| SVM | 71.4% | 71.1% | 71.2% |
| Linear CRF | 74.5% | 74.6% | **74.6%** |
| **Residual IDF (local)** | | | |
| MaxEnt | 76.0% | 62.3% | 68.5% |
| SVM | 68.8% | 69.8% | 69.3% |
| Linear CRF | 72.5% | 71.3% | **71.9%** |

**Table 4: Baseline performances and comparison between traditional ML model and Linear CRF using all proposed features (only local information for semantic feature) for advice-revealing sentence extraction**

We also ran an experiment using the same setting as before, except that we used both local and global information for semantic feature. Since using both outperforms the case of using local information alone, this performance becomes a baseline when we examine the benefit of using Skip-Chain CRF.

We ran an experiment using our improved model (Skip-Chain CRF) to see the effect in comparison with the Linear CRF. To construct the skip-edges (using our sentence generalization method), we employed the *Fisher scoring* method as the feature selection tool since it is known to be independent of the classifier being used [3]. As shown in Table 5, combining both local and global information for sentence informativeness score generally improved the performance across all the models from the previous setting. We can also see that Skip-Chain CRF performs better than Linear CRF because it can take advantage of long-range dependency between similar entities.

| Model | Prec. | Rec. | F1 |
|---|---|---|---|
| **IDF (local+global)** | | | |
| MaxEnt | 76.6% | 64.5% | 70.1% |
| SVM | 70.0% | 71.4% | 70.7% |
| Linear CRF | 74.2% | 73.5% | 73.8% |
| Skip-Chain CRF | 73.2% | 77.8% | **74.9%** |
| **Burstiness (local+global)** | | | |
| MaxEnt | 77.6% | 66.1% | 71.4% |
| SVM | 70.7% | 72.6% | 71.6% |
| Linear CRF | 74.8% | 73.6% | 74.2% |
| Skip-Chain CRF | 73.4% | 78.6% | **75.1%** |
| **Residual IDF (local+global)** | | | |
| MaxEnt | 77.2% | 63.6% | 69.8% |
| SVM | 45.0% | 56.3% | 50.0% |
| Linear CRF | 74.7% | 70.8% | 72.7% |
| Skip-Chain CRF | 72.4% | 76.9% | **73.9%** |

**Table 5: Comparison between traditional ML model, Linear CRF, and Skip-Chain CRF using all proposed features (local and global information for semantic feature) for advice-revealing sentence extraction**

### 7.2 Results for Context Sentence Extraction

In this experiment, we used traditional machine learning models to extract context sentences individually (for each advice-revealing sentence) leveraging all the features mentioned in Section 5.1. As shown in Table 6, SVM and Maximum Entropy models have similar performance for the context sentence extraction task, achieving around 53% in terms of F1-score.

Furthermore, we used Linear CRF for extracting context sentences individually. As shown in Table 6, the result shows that Linear CRF significantly outperforms the traditional machine learning models. This is not surprising due to the nature of a Web forum thread that has dependency between contiguous sentences.

We finally ran an experiment with 2D-CRF model for the context sentence extraction task. In this case, the context sentences of all the detected advice-revealing sentences

| Model | Prec. | Rec. | F1 |
|---|---|---|---|
| **Traditional Machine Learning Model** | | | |
| SVM | 52.5% | 54.2% | 53.3% |
| MaxEnt | 51.5% | 56.0% | **53.6%** |
| **CRF-based Model** | | | |
| Linear CRF | 66.3% | 62.9% | 64.5% |
| 2D-CRF | 57.2% | 76.8% | **65.5%** |

**Table 6: SVM, Maximum Entropy, Linear CRF, and 2D-CRF model for context sentence extraction**

are extracted together at once. We used the same features as mentioned in Section 5.1 for within-chain edges and between-chain edges. As shown in Table 6, 2D-CRF significantly outperforms the traditional machine learning models since 2D-CRF can model dependency for both between-chain positions and within-chain positions. Moreover, 2D-CRF is still better than Linear CRF in terms of F1-score.

### 7.3 Results of Extracting Advice and Context in One Pass

In this sub-section, we show our experimental results for the case of using our proposed CRF-based models (i.e., ML-$\mathrm{CRF_m}$ and 2D-$\mathrm{CRF_m}$+) for extracting advice-revealing sentences and their respective context sentences at the same time. Basically, we ran experiments using the value of sliding window edge $m = \{0, 1, 2, 3, 4, n\}$, where $n$ is the number of sentences in a forum thread. Moreover, we used syntactic, context, and semantic (burstiness & local+global) features as well as all the features mentioned in Section 5.1, except Feature-4, since advice revealing sentences are unknown when the extraction algorithm is executed at the first time. Table 7 shows that results for advice-revealing sentence extraction. Even though it is not significant, ML-$\mathrm{CRF_m}$ ($m = \{0, 1, 2\}$) shows an improvement compared to the use of Skip-Chain CRF. Meanwhile, 2D-$\mathrm{CRF_m}$+ worsens the performance significantly. The results shown in Table 7 suggest the fact that there exists dependency between advice-revealing sentence candidate sequence and context sentence candidate sequences. But, when we create too many dependency edges either *in between two types of sequence* or *among context sentence candidate sequences* (e.g, 2D-CRF+), it may harm the performance. This may be a byproduct of non-convergence during optimization since there are many additional parameters and the dataset is fairly small.

| Model | Prec. | Rec. | F1 |
|---|---|---|---|
| **Multiple Linear CRF** | | | |
| ML-$\mathrm{CRF_0}$ | 75.3% | 76.1% | **75.7%** |
| ML-$\mathrm{CRF_1}$ | 77.0% | 74.5% | **75.7%** |
| ML-$\mathrm{CRF_2}$ | 76.8% | 74.6% | **75.7%** |
| ML-$\mathrm{CRF_3}$ | 76.7% | 73.5% | 75.1% |
| ML-$\mathrm{CRF_n}$ | 75.8% | 61.0% | 67.6% |
| **2 Dimensional CRF+** | | | |
| 2D-$\mathrm{CRF_0}$+ | 81.8% | 44.9% | 58.0% |
| 2D-$\mathrm{CRF_1}$+ | 83.6% | 46.0% | **59.3%** |
| 2D-$\mathrm{CRF_2}$+ | 82.8% | 36.2% | 50.4% |
| 2D-$\mathrm{CRF_3}$+ | 85.4% | 42.5% | 56.7% |
| 2D-$\mathrm{CRF_n}$+ | 86.2% | 30.4% | 44.9% |

**Table 7: The Performance of ML-CRF and 2D-CRF+ for extracting advice-revealing sentences**

We then evaluated the performance of ML-CRF and 2D-CRF+ for context sentence extraction. In this experiment, we evaluated the performance of context sentence extraction only for the corresponding correctly extracted advice-revealing sentences, because we obtained the results from two different tasks at the same time. For example, in case of ML-$\mathrm{CRF_2}$, we only evaluated extracted context sentences only for all advice-revealing sentences indicated by Precision = 76.8%, Recall = 74.6%, and F1-score = 75.7% (true positive). Therefore, we cannot directly compare the results obtained in this experiment with the results described in Section 7.2 since experiments mentioned in Section 7.2 assumed that advice-revealing sentences have been perfectly extracted (with 100% accuracy) previously.

To compare the performance of ML-CRF and 2D-CRF+ with the previous models for context sentence extraction, we can run once again either SVM, Maximum Entropy, or Linear CRF for only those advice-revealing sentences that have been correctly extracted (true positive). We chose the Linear CRF since it gave the best performance compared to the traditional machine learning as described in Table 7. Unfortunately, we cannot run the 2D-CRF in this case since it requires an ideal situation, i.e., all real advice-revealing sentences in a target thread must be known before. Moreover, when we used the Linear CRF to extract context sentences.

Table 8 shows the performances of the ML-$\mathrm{CRF_m}$ and the 2D-$\mathrm{CRF_m}$+ for extracting context sentences. We only show top-2 model settings from both ML-CRF and 2D-CRF+ that give the best performance as mentioned in Table 7. In Table 8, we use a term *sector* to denote an area between two horizontal lines. A sector describes the performance of a model (either ML-CRF or 2D-CRF+) and the performance of the Linear CRF that ran over the same set of advice-revealing sentences.

As shown in Table 8, the ML-CRF and 2D-CRF+ outperformed the Linear CRF significantly for context sentence extraction. Specifically, ML-CRF performed better than 2D-CRF+. Once again, the results suggest the fact that capturing dependency between an advice-revealing sentence candidate sequence and context sentence candidate sequences is important. But, constructing too many dependency edges, especially among context sentence candidate sequences, will harm the performance for context sentence extraction.

| Model | Prec. | Rec. | F1 |
|---|---|---|---|
| ML-$\mathrm{CRF_1}$ | 80.0% | 67.6% | **73.3%** |
| Linear CRF | 81.6% | 28.6% | 42.3% |
| ML-$\mathrm{CRF_2}$ | 79.8% | 70.1% | **74.6%** |
| Linear CRF | 82.0% | 28.5% | 42.3% |
| 2D-$\mathrm{CRF_0}$+ | 79.7% | 37.3% | **50.8%** |
| Linear CRF | 83.3% | 29.0% | 43.1% |
| 2D-$\mathrm{CRF_1}$+ | 81.3% | 37.3% | **51.2%** |
| Linear CRF | 81.6% | 28.8% | 42.6% |

**Table 8: The Performance of ML-CRF and 2D-CRF+ for extracting context sentences. The results are also compared with the Linear CRF model**

## 8. CONCLUSIONS AND FUTURE WORKS

We presented a methodology to extract advice-revealing sentences as well as their context sentences from Web forums. Our experiments show that the Skip-Chain CRF performs better than the Linear CRF and traditional machine learning models for advice-revealing sentences since it can model dependency between nearby sentences as well as two similar sentences separated by long range of positions. In case of context sentence extraction, the 2D-CRF performs better than traditional machine learning model because it can model dependency for both between-chain positions and within-chain positions. Finally, we show that our Multiple Linear CRF (ML-CRF) performs better than any other models studied in this paper for extracting advice-revealing sentences and context sentences, which means that the ML-CRF is currently the best model for implementing **ATU** extraction system in the real situation. Given the promising results, we plan to explore along two lines. First, we still need to improve the performance by devising other features and models. At the same time, we also need to evaluate the efficiency side. Second, we plan to work on actually implementing advice mining system by designing a refined model of expressing advice.

## 9. ACKNOWLEDGMENT

The first author is a grantee of a Korean Government Scholarship from the Ministry of Education, Science and Technology (MEST), Republic of Korea.

## 10. REFERENCES


[1] N. Asher, F. Benamara, and Y. Mathieu. Appraisal of opinion expressions in discourse. *Lingvistic Investigationes*, 31.2:279–292, 2009.

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

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

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

[5] 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.

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

[7] G. Cong, L. Wang, C.-Y. Lin, Y.-I. Song, and Y. Sun. Finding question-answer pairs from online forums. In *Proceedings of the 31st annual international ACM SIGIR conference*, SIGIR '08, pages 467–474, New York, NY, USA, 2008. ACM.

[8] 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 *Proceedings of the 46th Annual Meeting of the ACL*, pages 710–718, June 2008.

[9] K. Inui, S. Abe, K. Hara, H. Morita, C. Sao, M. Eguchi, A. Sumida, K. Murakami, and S. Matsuyoshi. Experience mining: Building a large-scale database of personal experiences and opinions from web documents. In *Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT '08. IEEE/WIC/ACM International Conference on*, volume 1, pages 314 –321, dec. 2008.

[10] A. Kennedy and S. Szpakowicz. A supervised method of feature weighting for measuring semantic relatedness. In *Proceedings of the 24th Canadian conference on Advances in artificial intelligence*, Canadian AI'11, pages 222–233, Berlin, Heidelberg, 2011. Springer-Verlag.

[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. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth ICML*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[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] A. McCallum and W. Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 188–191, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[16] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135, Jan. 2008.

[17] K. C. Park, Y. Jeong, and S. H. Myaeng. Detecting experiences from weblogs. In *Proceedings of the 48th Annual Meeting of the ACL*, pages 1464–1472, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

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

[19] L. Shrestha and K. McKeown. Detection of question-answer pairs in email conversations. In *Proceedings of the 20th international conference on COLING*, COLING '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

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

[21] C. Sutton and A. Mccallum. Mccallum: Collective segmentation and labeling of distant entities in information extraction. In *ICML Workshop on Statistical Relational Learning and Its Connections*, 2004.

[22] Y. Tsuruoka, J. Tsujii, and S. Ananiadou. Fast full parsing by linear-chain conditional random fields. In *Proceedings of the 12th Conference of the EACL*, EACL '09, pages 790–798, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[23] 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.

[24] A. F. Wicaksono and S.-H. Myaeng. Mining advices from weblogs. In *Proceedings of the 21st ACM CIKM*, CIKM '12, pages 2347–2350, New York, NY, USA, 2012. ACM.