Predictive Models of Form Filling

Alnur Ali
Microsoft Corporation
Redmond, WA 98052
alnurali@microsoft.com

Christopher Meek
Microsoft Research
Redmond, WA 98052
meek@microsoft.com

ABSTRACT
In this paper we investigate predictive models of form filling. A predictive model of form filling aims to reduce the amount of time a user spends filling out a form by predicting the values of fields on the form and using these predictions to make suggestions to the form filler. Existing predictive models ignore both the values of other fields on the form and the values previously entered by users other than the current form filler when predicting the value of a target field. We introduce a novel model of predictive form filling named the Collaborative and Contextually Frequently Used model that utilizes these sources of information. We demonstrate that our model outperforms the existing, standard models of predictive form filling using a real-world collection of forms. We also demonstrate that using the values used by other form fillers can significantly improve the performance of the existing models.

Author Keywords
Form filling, CCFU, MFU, MRU, Bayesian network, Hidden Web.

ACM Classification Keywords
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *information filtering, retrieval models*. H.5.3 [**Information Interfaces and Presentation**]: Group and Organization Interfaces – *collaborative computing, evaluation/methodology*. I.2.1 [**Artificial Intelligence**]: Applications and Expert Systems – *Office automation*.

INTRODUCTION
People frequently have to fill out forms. For example, it is estimated that 70 million professionals, roughly equivalent to 59% of all professionals in the United States, fill out forms regularly [11]. Furthermore, people frequently have to fill out the *same* form, over and over. For example, a salesperson may have to fill out a purchase order every time his client makes an order. This represents an opportunity to use predictive algorithms to improve productivity.

There are a variety of approaches by which predictive algorithms can be used to aid a user when he is filling out a form. The most common approach is to generate a list of likely values for the field the user is attempting to fill out, called a *suggestion list*. The suggestion list for a given field is presented to the user when he navigates to that field. A variety of user interfaces can be used to display a suggestion list to the user; currently, the most popular is to use a drop-down list, for example in Figure 1. The user can then navigate through the items in a suggestion list by scrolling or typing, although other input modalities, such as speaking or touching, could be considered.

There has been little published research on developing predictive models of form filling. Current approaches do not consider the values of other fields on the form when predicting the value of the field the user is currently filling out. This means that these approaches cannot model the naturally occurring dependencies between fields, such as that between a field intended to capture the name of the form filler and a field intended to capture the filler's address. In addition, current approaches, while typically considering previously entered values from the current user, do not consider values for fields previously entered by other form fillers when making predictions.

We introduce a novel model of predictive form filling named the *Collaborative and Contextually Frequently Used (CCFU)* model to utilize these two information sources. We demonstrate that the CCFU improves over the standard models of form filling with a real-world dataset. We also show how incorporating the values used by other form fillers can improve the performance of existing models.

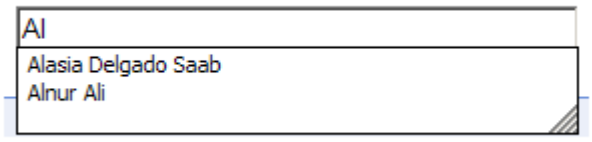

Figure 1. A suggestion list with prefix filtering.

This paper is organized as follows. In section 2, we describe current approaches to predictive form filling and some related work. In section 3, we formally define the standard and CCFU predictive models of form filling and describe how they can be used and learned from data. In section 4,

we present an experimental evaluation of the standard and CCFU predictive models, in several evaluation scenarios, and using several evaluation metrics. Finally, in section 5, we provide a discussion of our contributions and future work.

BACKGROUND

The goal of a form is to present a user interface which can be used to rapidly and accurately gather data from many users. As a result, a form commonly contains several standard user interface elements, such as text boxes, drop-down lists, and date pickers, all of which are generically referred to as fields. The user is expected to enter information into these fields. In this paper we restrict our attention to fields in which the user is expected to enter textual data, although the methods apply generally.

There are two common scenarios in which users fill out forms. First, the *singleton* scenario, is where the user is modifying the value of a single field. Second, the *cumulative* scenario, is where the user is filling out every field on a form. The cumulative scenario assumes that the fields on a form are typically filled out in a particular order.

The goal of a predictive model of form filling is to reduce the amount of time a user spends filling out a form in these scenarios, by offering the user predictions for values of each target field on a form via suggestion lists. Three predictive models are most commonly used in practice.

The first model is the *most frequently used (MFU)* model. The MFU populates the suggestion lists for each field by considering the values previously entered by the *active* user, the current form filler, and sorting them by frequency. The suggestions are sorted from most frequent to least frequent. This simple model is able to bias predictions based on *history*, the previously entered values for a form, but not *context*, the values of fields other than the current target field. Additionally, this predictive model is a *personal* model, as it generates predictions by only examining the history of the active user. This is in opposition to a *collaborative* model, which generates predictions by considering the history of the active and *non-active* users, which are users other than the current form filler.

The second model is the *most recently used (MRU)*. The MRU is populates the suggestion list in the same way as the MFU, but sorts the suggestions by *recency*. That is, each suggestion is ordered by the time elapsed since the suggestion was last used.

The final model is the *deterministic* model. The deterministic model requires an initial training phase in which the user provides values for common fields such as first name, last name, and zip code. During form filling, if the model detects any fields for which the user has provided values, the deterministic method enters the value for the field on the form. This model has a number of shortcomings. First, it requires upfront effort from the user to train the model. Second, unlike the MFU and MRU models, this model is static. For example, a user's address may not always be the same, and furthermore, on different forms, the user may want to enter different addresses. Third, it requires additional effort from the user to retrain the model in examples like these. Due to its limitations, we do not use the deterministic model in our experiments. Most commercial e-mail and Web browsing software products include an implementation of MFU or MRU.

There has also been some research on methods to improve the MFU and MRU models. These methods generally function by removing some members of a suggestion list according to some criteria. Thus, we call them *suggestion list filtering* methods, because they filter out items from a suggestion list. The *prefix filtering* method works by removing items from a suggestion list whose prefix does not match the letters entered by the user so far (Figure 1). The *wildcarding* method works by allowing users to enter wildcards in place of their actual, intended text [3]. Items in a suggestion list that do not match the wildcard resolution are removed. This method is especially advantageous in mobile phone scenarios because users can type less, as well as cases when the user cannot exactly remember or spell their full input. In this paper, all of the predictive form filling models we compare employ prefix filtering.

PREDICTIVE MODELS OF FORM FILLING

In this section, we develop a framework for building predictive models of form filling. We first introduce notation for what a form is, then build on this notation and develop the CCFU model, and then develop the MFU and MRU models using this notation. We also discuss how to use and learn these models from data.

Forms and Fields

A *form* $\boldsymbol{F}$ is a set of fields F_i each assigned to a value f_i. That is, $\boldsymbol{F} = \{F_1 = f_1, F_2 = f_2, \ldots, F_n = f_n\}$. When $f_i = \emptyset$, then a field is considered *blank*, meaning it has not been filled out.

In this paper, we assume that the set $\boldsymbol{F}$ is given to us. That is, we do not attempt to infer the fields in a form as in [6]. Additionally, we assume that we are given the set of possible values $V_i = \{v_i^1, \ldots, v_i^m\}$ for each field F_i.

Furthermore, we order the elements F_i of $\boldsymbol{F}$ in the way that corresponds to the order in which they are typically filled out. For example, $\boldsymbol{F} = \{F_1, F_2, \ldots, F_n\}$ implies that F_i is typically filled out before F_j if $i < j$. Thus, $\boldsymbol{F} \backslash F_i$ represents a set of fields that can be used as context by a model when predicting the value of field F_i.

Predictive Models of Form Filling

As discussed earlier, our models will build a suggestion list that contains ranked predictions for the user's current target field. More formally, a predictive model of form filling is required to be able to answer queries about the probability of values for a field given the values of other fields, history, and some initial text entered by the user in the target field.

For this paper we focus on predictive models that can provide a conditional probability for each field

$$P(F_i = f_i | context(F_i), history)$$

where $context(F_i)$ is a set of fields deemed by a model to be contextually relevant in predicting the value of field F_i and $history$ is a slight abuse of notation that serves to indicate whether a model is trained with personal or collaborative user history. We will return to the idea of training a personal and collaborative model later in this section.

The conditional above can be written in terms of a joint probability distribution

$$\frac{P(F_i = f_i, context(F_i) | history)}{P(context(F_i) | history)}$$

Thus, it is simplest for our models to model the full joint over all fields, and then marginalize over certain fields in order to extract the relevant conditional. However, modeling the full joint is difficult due to the amount of training data and storage space required. Thus, models may differ in the kinds of independence assumptions they make in order to make modeling the full joint tractable. These assumptions, in turn, affect the accuracy of a predictive model. We return to the nature of these differences later in this section when we derive each model of predictive form filling.

Regardless of these differences, we can use the following straightforward algorithm for building a suggestion list given a predictive model:

1. For each field $F_i \in \boldsymbol{F}$, order the list of possible values V_i ordered by the conditional probability of the value $P(F_i = v_i^j | context(F_i), history)$.

2. Employ a suggestion list filtering method for pruning the ranked list in response to user input.

Next we consider the MFU, MRU, and CCFU as specific instances of predictive models of form filling.

The MFU and MRU Predictive Models

In both the MFU and MRU model there is an independence assumption between the various fields of a form. This means that we can represent the joint by

$$P(F_1 = f_1, \ldots, F_n = f_n | history) = \prod_{i=1}^{n} P(F_i = f_i | history)$$

However, the two methods differ in the way in which they utilize history. For the MFU one simply needs to keep track of the number of times that each possible value is used: a simple maximum likelihood estimate of the probability of the field's value. In the case of the MRU, there are a variety of alternative methods by which one can update the probabilities. One simple method is to use an unnormalized distribution in which the value of the most recent value of the field is set to a maximal value and the previous values are appropriately decremented.

The Collaborative Contextually Frequently Used (CCFU) Predictive Model

The CCFU uses a Bayesian network to model the joint [9]. The advantage of using a Bayesian network is that there are standard algorithms for efficiently answering the conditional probability queries required for form filling. There are a variety of alternative models one can use to represent the joint including undirected graphical models and dependency networks [4].

The CCFU does not make the restrictive independence assumptions of the MFU and MRU models, but rather uses data to determine what independence assumptions are warranted. To this end, we use a greedy structure learning algorithm over a set of previously filled out forms to learn the dependencies between fields in a form [2]. In order to realistically model user behavior in a cumulative scenario, we assume that any field F_j that is after a field F_i (i.e., $i < j$) in the field ordering for a form cannot directly affect the value of F_i. As a result, we first specify to the structure learning algorithm that any edges between F_i and $F_j, i < j$ are forbidden. The structure learning algorithm then proceeds by adding edges between pairs of permitted fields in an initially unconnected graph. If a pair of connected fields are found to be probabilistically dependent, the edge is kept; otherwise, it is deleted. A fragment of the learned network is shown in Figure 2.

A variety of algorithms exist for performing inference on Bayesian networks. It is entirely likely that a form filler may leave some fields on a form blank. Thus, we choose the junction tree inference algorithm in order to achieve exact and tractable results that deal with missing data well [5]. In the case of missing data, we consider the value of the field as unobserved rather than null, so that other fields containing values are not blocked by these likely to be uninformative fields.

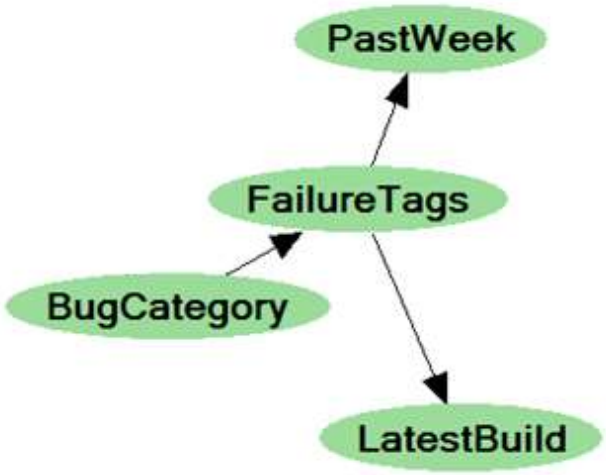

Figure 2. A fragment of CCFU's learned Bayesian network.

Training Predictive Models of Form Filling

As mentioned previously, we delineate between personal and collaborative predictive models of form filling. Personal models rely on data from the active user's history; collaborative models rely on data from the active and non-active users' histories. One advantage of a collaborative model is that one can learn more about possible values for each field and have more data to learn about cross-field dependencies.

The CCFU is a collaborative model; it is trained from collaborative user data. However, the CCFU additionally biases predictions towards the history of the active user. Although the CCFU considers collaborative user data when presenting suggestions to the user, it presents suggestions previously entered by the active user first, sorted by their posterior probability, and suggestions previously entered by non-active users second, sorted by their posterior probability. For example, see Figure 3.

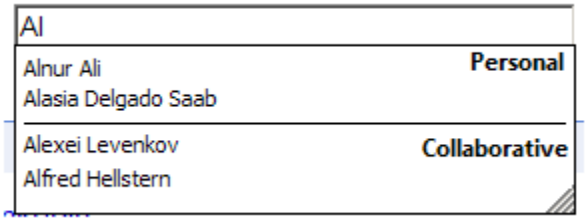

Figure 3. An example suggestion list, as prepared by the CCFU.

On the other hand, the MFU and MRU are personal models; they build suggestion lists from active user history only. In order to better understand the value of the context and collaboration we also consider collaborative versions of the MFU and MRU models. In the *collaborative MFU (CMFU)* model, we populate the suggestion list for a target field by sorting the values that the active and all non-active users have previously used for the field based on frequency. In the *collaborative MRU (CMRU)* model, we populate the suggestion list the same way as for the CMFU, but sort based on recency.

Additionally, to understand the value to a collaborative model of biasing predictions towards the active user's history, we also experiment with a variant of the CCFU, that we call the *Collaboratively and Contextually Frequently Used Combined model (CCFUC)*, that does not discriminate between personal and collaborative suggestions. The CCFUC does not segment suggestions according to their user history source; instead it simply ranks them according to their posterior probability.

EVALUATION

We acquired a snapshot of 1356 forms submitted to a bug-tracking database. Each form contained 31 fields that were a mix of short, long, categorical, numerical, and free textual fields. The dataset was divided into a training set of 963 forms and a test set of 393 forms. Evaluations consisted of comparing the CCFU with the MFU, CMFU, MRU, CMRU, CCFUC, and the cost of unaided, manual typing in order to fill out a form. In all these cases we measured two numbers. First, we measured the number of items visited when a user scrolls through the suggestion list to find their desired input. Second, we measured the number of keystrokes when a user types in a suggestion list with prefix filtering to find their desired input. We then averaged these two metrics over all trial. Due to the mix of fields we used, the variance of these numbers was high. Lastly, we computed statistical significance between models using a one-sided, paired Student's t test.

Evaluation Metrics

There are two natural ways for a user to fill out a field given a suggestion list. The user may scroll through the list of suggestions in the suggestion list, looking for their target value v_i^*. A user may also type his input into a field. In the case where suggestion list filtering is enabled on a field, the suggestion list filtering can help the user find their v_i^*. It is likely that in practice, users use a combination of scrolling and typing. However, there has been little research on characterizing exactly what combination of scrolling and typing users employ. This is furthermore challenging to do, because it is likely that this combination would change as a user adapts to the model being empirically studied. Additionally, some fields are more suited to scrolling, while others are more suited to typing – for example, categorical and free text fields, respectively. In the absence of a rigorous characterization of user behavior, we introduce two cost measures that represent the two extremes of scrolling and typing when filling out a field.

We define *the scrolling cost* to be the position of v_i^* on the suggestions list if $v_i^* \in S_i$, or some constant k plus the length of v_i^* if $v_i^* \notin S_i$. Intuitively, this is because the user must first search the entire suggestion list for v_i^* and then type it out in full, without prefix filtering, because the model could not provide it for them. We use k as the length of the longest scrolling list between all models, for a given field. It is unfair to penalize models for having shorter or longer suggestions lists than others because the maximum allowed length of a suggestion list is arbitrary. For example, we could impose a maximum of ten entries per suggestion list for all models, rather than allowing them to vary on their own, so we pick a common constant.

We define *the typing cost* to be the number of characters the user has to type with prefix filtering until v_i^* is the one and only remaining suggestion in a model's suggestion list if $v_i^* \in S_i$, or the length of v_i^* if $v_i^* \notin S_i$. Intuitively, this is because the user has to type out their v_i^* in full, without prefix filtering, since the model could not provide it for them.

In addition to comparing alternative predictive form filling methods we evaluate the impact of restricting the size of the

suggestion list. The *suggestion list size* is the cardinality of the set of suggestions for a field $|S_i|$. Intuitively, this is the number of suggestions that a model presents to a user. The suggestion list size is said to be *bounded* at k if a model chooses to present to the user a proper subset of cardinality k of S_i prior to any suggestion list filtering. The suggestion list size is said to be *unbounded* if a model presents to the user the whole set S_i prior to any suggestion list filtering.

Comparing Predictive Model Performance in the Cumulative Scenario

We begin by comparing our CCFU with the MFU and MRU in the cumulative form filling scenario. The results of the experiment for both scrolling and typing cost are presented in Table 1. The CCFU scrolling cost is significantly lower than MFU and MRU's scrolling cost at $p < 0.0005$. The CCFU's typing cost is also significantly lower than the MFU, MRU, and manual typing cost, at $p < 0.009$. This demonstrates the value of the combination of collaboration and context in the predictive form filling task.

One of the ways in which collaboration positively impacts CCFU's scrolling cost is by reducing the number of times that the target value does not occur in the suggestion list. This is due to the fact that the active user has access to values he and other users previously entered for a field, which is advantageous because an active user may not always enter values from his own personal history.

However, for certain fields, it is undesirable to utilize collaboration. For example, suggesting values entered by other users for fields intended to capture the user's billing address or password would constitute a privacy issue.

Additionally, the value of collaboration is related to the temporal nature of the form. For instance, a new employee who joins a team would likely benefit a great deal from a collaborative model, as his personal data store for all fields would be non-existent. We leave understanding what kinds of fields benefit most from collaboration to future work.

Context also positively impacts CCFU's scrolling cost by sorting the suggestion list based on the evidence from other fields seen in the form being filled out.

All methods improve significantly over manual typing cost. This demonstrates the value of applying prefix filtering to predictive form filling tasks. The application of prefix filtering is a powerful method to quickly filter down even large suggestion lists, such as those learned from active and non-active user history.

Note that typing cost is an especially good indicator of model performance in cases when field values can be lengthy. For example, it was common in our dataset to see values for fields such as "FeatureCrew = Picture Control - Embedded Image."

	Average scrolling cost	Average typing cost
CCFU	167	84
MFU	351	88
MRU	352	88
Manual typing cost		149

Table 1. Comparing Predictive Model Performance in the Cumulative Scenario.

Comparing Predictive Model Performance in the Singleton Scenario

We next compare predictive model performance in the singleton form filling scenario. The results of the experiment are presented for a representative field in Table 2. The CCFU's scrolling and typing costs are significantly lower than the MFU and MRU's scrolling and typing costs at $p < 0.0005$. The CCFU's typing costs is also significantly lower than the manual typing cost at $p < 0.0005$.

	Average scrolling cost	Average typing cost
CCFU	22	3
MFU	45	4
MRU	45	4
Manual typing cost		7

Table 2. Comparing Predictive Model Performance in the Singleton Scenario on Field "AssignedTo."

Improving Standard Predictive Model Performance by Incorporating Collaboration

As demonstrated in the previous sections, the CCFU model is superior to both the standard MRU and MFU. In order to better understand the value of collaboration and context in improving a predictive form filling model we compared performance between the CCFU and the CMRU and CMFU models. The results of this experiment are given in Table 3.

Both CMFU and CMRU show a statistically significant improvement as compared to the MFU and MRU. This demonstrates the value of incorporating collaboration to these models. In fact, the CMFU and CMRU typing costs are now equal to the CCFU typing cost. However, the CCFU scrolling cost is still significantly lower than the CMFU and CMRU scrolling costs at $p < 0.0005$, illustrating the value of context and biasing predictions towards active user data in a predictive model.

	Average scrolling cost	Average typing cost
CCFU	167	84
CCFUC	193	84
CMFU	197	83
CMRU	194	83
Manual typing cost		149

Table 3. Improving Standard Predictive Model Performance by Incorporating Collaboration.

Exploring the Value of Personal History in a Collaborative Model

To demonstrate the value of biasing predictions towards active user data in a collaborative model, we also evaluated the performance of the CCFUC model, which does not discriminate between personal and collaborative suggestions. The results are again shown in Table 3.

Although the CCFU typing cost are about the same as the CCFUC typing cost, the CCFU scrolling cost is significantly lower. This demonstrates that biasing predictions towards active user data can be useful in a predictive model of form filling. Personal suggestions are frequently used by the active user. However, in cases where there are no useful personal suggestions, collaborative suggestions help the active user avoid typing out the full input for a field. In cases where the active user is uncertain what to input for a field, for example the name of a hotel to stay at during an upcoming business trip, collaborative suggestions can help the active user browse for legitimate, commonly-used inputs.

Exploring the Effects of Suggestion List Size on Model Performance

Finally, we perform an experiment in order to determine the effect of suggestion list size on predictive model performance. We repeated the experiment as in Table 3, except with a suggestion list bound at 5. The results of the experiment are shown in Table 4. We chose to compare CCFU against CMFU, CMRU, and CCFUC in order to minimize variability, since model typing costs equalize among collaborative models.

We observed that each predictive model's scrolling and typing costs increased over the trial in Table 3. This suggests that each predictive model's performance declined when clamping its suggestion list size, because the suggestion list is curtailed even though it may have contained the user's desired input.

Thus, limiting the size of a suggestion list should be avoided, as it can adversely affect model performance. However, in some situations, it may be necessary to apply some limit to the size of a suggestion list; for example, on a mobile phone, model storage space and network traffic should be kept to a minimum.

Separately, the suggestion list user interface itself should be restricted to a certain size, so as to not inundate users with suggestions. We leave exploring the effects on user scrolling and typing behavior of changing the size of the suggestion list user interface to future work.

On the other hand, we see that both the CCFU and CCFUC scrolling and typing costs remained lower than those of the MFU and MRU. The commonality between the CCFU and CCFUC is the reliance on context; this again serves to indicate the role context can play in improving a predictive model performance.

	Average scrolling cost	Average typing cost
CCFU	371	92
CCFUC	469	100
CMFU	498	102
CMRU	513	105
Manual typing cost		149

Table 4. Exploring the Effects of Suggestion List Size on Model Performance.

DISCUSSION AND FUTURE WORK

In this paper we investigated predictive models of form filling. We introduced a new predictive model of form filling, the CCFU, that outperforms the standard models by using contextual and collaborative information sources. The CCFU uses a Bayesian network to compute how likely a suggestion is for a target field, and learns the probabilities from active and non-active user data. Our evaluation demonstrated significant improvements in efficiency when using our CCFU model. We also demonstrated that existing predictive models of form filling could be extended by incorporating collaborative information sources also yielding significant improvements in efficiency.

The efficiency gains demonstrate the potential value of the CCFU. Our analysis of efficiency gains assumed idealized models of user interaction. In future work we would like to evaluate the CCFU models in a user study to evaluate the efficiency gains realized by real rather than idealized users. In addition to a direct evaluation of predictive models of form filling, such a user study would be useful for building models of how users interact with form filing user interfaces. Such a model would likely enable further development and evaluation of predictive models of form filling without the need for additional user studies. It is useful to note that in addition to gains in efficiency, it is likely that employing more powerful predictive model would improve the consistency of data entry. Developing metrics for the evaluation of consistency would be valuable.

In addition, we believe that the predictive performance of the CCFU could be improved by understanding which kinds of fields are more suited to personal or collaborative predictions. For example, as noted earlier, fields intended to capture the active user's billing address are more apt to personal prediction. In such cases, values previously used by the active user could be ranked higher than those by non-active users. On the other hand, some fields are clearly more suited to collaborative prediction, in which case the collaborative suggestions might be ranked higher than the personal ones. Exploring different kinds of collaborative training methods might also boost model performance for these kinds of fields. For instance, when predicting the value of a field intended to capture the name of the active user's manager, it might be helpful to suggest previously used values by members of the active user's social network.

There are a number of assumptions made in building a CCFU model that could be relaxed. For instance, the current approach to constructing the Bayesian network used by the CCFU assumes that an ordering of the fields on a forms is given. This assumption simplifies the learning process but necessary. In addition to removing this assumption in learning a Bayesian network, it would be useful to evaluate other types of graphical models for this prediction task such as dependency networks. In addition, the CCFU assumes that the fields on a form are known a priori; that is, they are not inferred from the visual layout of the form. Automatically inferring these fields could extend the utility of the CCFU: the CCFU could predict the values of the fields on a second form, given what a model learns about the fields on the first form.

Using the CCFU to Mine the Hidden Web

We have shown that the CCFU can be used to reduce the time a user spends filling out a form. However, the CCFU could also be applied to the problem of mining the *Hidden Web*. The Hidden Web refers to content stored on Web servers that can only be indexed by Web crawlers by first filling out a form.

Some research has been done on developing models of predictive form filling for the purpose of mining the Hidden Web. The *Hidden Web Exposer (HiWE)*, proposed by [10], attempts to fill out a form in order to index content from a Web server, by selecting the most reliable inputs from all possible suggestions for each field. HiWE first learns suggestions for each field by either having a human trainer input them into the model or crawling the Web for possible suggestions itself. HiWE then weights the reliability of each suggestion based on the training data source: suggestions added by humans are weighted higher than ones discovered by HiWE itself. At index time, the inputs chosen from the list of suggestions to fill out a form are selected based on one of three different selection criteria. First, the permutation of inputs that possesses the largest minimum weight for any one input is chosen. Second, the permutation of inputs that possesses the largest mean suggestion weight is chosen. Third, the permutation of inputs that possesses the largest product of suggestion weights is chosen. The last criterion is equivalent to treating the weights as probabilities of success, and taking the joint over all weights, but assuming the weights are independent. By extension, this model also considers fields bearing these weights to be independent, indicating that this model also neglects context when filling out forms.

[8] extend the work of [10], but also take on a slightly different goal. They aim to extract as much *diverse* content as possible. They accomplish this by filling out a form with a randomly chosen permutation of suggestions, measuring the results returned by the Web server for novelty, and repeating this process until no more novel content is being returned, the user has told the crawler to terminate, or a maximum iteration threshold has been reached. Although this model is effective at indexing fresh content, it is not a good model of realistic user behavior.

Separately, [7] employ a purely deterministic approach in filling out forms for the Hidden Web. The model simply fills out a field by using any previously entered value for that field.

Hidden Web approaches attempt to extract reliable, frequently accessed, and potentially diverse content from a Web server gated by a form. Scrolling cost represents a way to tie the CCFU to the problem of mining the Hidden Web. The CCFU may be used to generate predictions for fields that have a low scrolling cost, implying a high posterior probability, in order to generate the most frequently accessed content. Conversely, the CCFU can be used to generate predictions for fields that have a high scrolling cost, implying a low posterior probability, in order to generate the least frequently accessed, or most diverse, content. The only requirements are that a field ordering be chosen for each form encountered on the Web, and that the CCFU have access to a training set of filled out forms before being used for inference.

An ordering could be algorithmically determined by a method like the following. First, partition the set of fields $\boldsymbol{F}$ in the Bayesian network into three subsets. The first subset includes fields that are only predictive of other fields; these fields should go first in the ordering. The second subset includes fields that are only predicted by other fields; these fields should go last in the ordering. The third subset includes fields that are both predicted by and predict other nodes; these fields should go in the middle of the ordering. Second, obtain the user's desired field ordering. Third, combine these two field orderings.

ACKNOWLEDGMENTS

We thank Max Chickering for helpful discussions.

REFERENCES

1. Breese, J. S., Heckerman, D., and Kadie, C. Empirical analysis of predictive algorithms for collaborative filtering. Proceedings of the Fourteenth Annual

Conference on Uncertainty in Artificial Intelligence, 1998, Morgan Kaufmann Publishers, 43-52.

2. Chickering, D. M. Optimal structure identification with greedy search. Journal of Machine Learning Research, Volume 3 (2002), 507-554.
3. Church, K. and Thiesson, B. The wild thing! Proceedings of the Association for Computational Linguistics, 2005, Association for Computational Linguistics, 93-96.
4. Heckerman, D. et al. Dependency networks for inference, collaborative filtering, and data visualization. Journal of Machine Learning Research, Volume 1 (2000), 49-75.
5. Jensen, F. V. Bayesian Networks and Decision Graphs. Springer, 2002.
6. Kaljuvee, O., Buyukkokten, O., Garcia-Molina, H., and Paepcke, A. Efficient Web form entry on PDAs. Proceedings of the 10th International World Wide Web Conference, 2001, Association for Computing Machinery, 663-672.
7. Konopnicki, D. and Shmueli, O. Information gathering in the World-Wide Web: the W3QL query language and the W3QS system. Association for Computing Machinery Transactions on Database Systems, Volume 23, Number 4 (1998), 369-410.
8. Liddle, S. W., Embley, D. W., Scott, D. T., and Yau, S. H. Extracting data behind Web forms. Proceedings of the Joint Workshop on Conceptual Modeling Approaches, 2002, Springer, 402-413.
9. Pearl, J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, 1998.
10. Raghavan, S. and Garcia-Molina, H. Crawling the hidden web. Proceedings of the 27th International Conference on Very Large Data Bases, 2001, Morgan Kaufmann Publishers, 129-138.
11. Viola, P. and Narasimhan, M. Learning to extract information from semi-structured text using a discriminative context free grammar. Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2005, Association for Computing Machinery, 330-337.