Automatically Filling Form-Based Web Interfaces with Free Text Inputs

Guilherme A. Toda† Eli Cortez† Filipe Mesquita†
Altigran S. da Silva†* Edleno Moura† Marden Neubert‡

†Federal University of Amazonas
{gat,eccv,fsm,alti,edleno}@dcc.ufam.edu.br

‡UoI Inc.
marden@uolinc.com

ABSTRACT

On the web of today the most prevalent solution for users to interact with data-intensive applications is the use of form-based interfaces composed by several data input fields, such as text boxes, radio buttons, pull-down lists, check boxes, etc. Although these interfaces are popular and effective, in many cases, free text interfaces are preferred over form-based ones. In this paper we discuss the proposal and the implementation of a novel IR-based method for using data rich free text to interact with form-based interfaces. Our solution takes a free text as input, extracts implicitly data values from it and fills appropriate fields using them. For this task, we rely on values of previous submissions for each field, which are freely obtained from the usage of form-based interfaces.

Categories and Subject Descriptors

J.0 [**Computer Applications**]: General

General Terms

Algorithms, Management

Keywords

Data Extraction, Form Filling, Web Application

1. INTRODUCTION

The Web is abundant in data-intensive applications such as online stores, digital libraries, and data sharing services (e.g. Craiglists, Googlebase), which store and maintain high volumes of data in the so called *Web databases*. One of the challenges regarding the development of this type of application is building intuitive interfaces for allowing users to interact with their underlying structured databases. The most prevalent solution in this direction is designing form-based interfaces which contain *data input fields*, such as text boxes, radio buttons, pull-down lists, check boxes and other input mechanisms.

This situation is very common in popular auction sites such as *eBay* and *amazon.com* which extensively use form-based interfaces for allowing users to register offers. In fact, there may be distinct form-based interfaces with specific fields depending on the product being offered. For instance, in the experiments presented in this paper, we consider distinct interfaces corresponding to the categories "vehicles", "mobile phones" from the Brazilian popular auction site *TodaOferta.com*. Interestingly, as some of these sites (e.g., *eBay* and *TodaOferta*) also allow entering offers using generic free text descriptions, users often avoid using form-based interfaces. However, in many cases the lacking of structured information obtained through these interfaces may prevent the proper use of services based on searching, mining, recommendation and integration over offers.

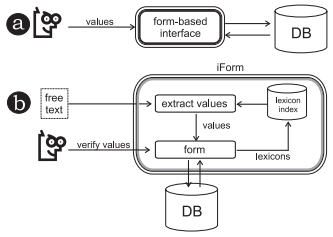

Figure 1: Overview of a pure form-based interface (a) and iForm (b).

In this paper we propose an aproach that consists in deploying a system that receives a free text input (e.g., an offer or an ad), and recognizes implicit data values occurring in it that can be used to appropriately fill out the fields in a form based interface. Unlike approaches for information extraction from free text [2, 1], our approach does not require a training phase over manually labeled texts, since it simply relies on values previously entered in form-based interfaces.

2. THE IFORM APPROACH

In this section, we present the iForm approach in contrast to form-based interfaces as illustrated in Figure 1. Before using iForm for boosting a form-based interface, we consider that users have been using the interface as shown in Figure 1a, by typing values in fields to interact with the database DB. The values manually entered in this process allow us to build and maintain *lexicons*, which stores terms used for filling each field. Lexicons are the crux of iForm approach. Users dealing with iForm, as illustrated in Figure 1b, provide a free text document or portions as input and iForm extract values for filling the form by matching

*Contact author.

WWW 2009, April 20–24, 2009, Madrid, Spain.
ACM 978-1-60558-487-4/09/04.

the contents of the input text with the lexicon of each field. For simplicity, we refer to the user free text document or portions as *input text* from now on. Users may want to verify the form filled by our method to make corrections and then proceed with the request submission. After that, iForm automatically updates the lexicon index for improving the extraction quality as users deal with iForm.

The iForm approach consists in extracting values from the input text using lexicons and filling a target form-based interface using them. In order to identify suitable field values in the input text we first break it into *segments* and then estimate the *affinity* of each segment to a field. Intuitively, the affinity score would be 1 if the segment is found as a value in the field domain (i.e., the set of all possible values of the field) and 0, otherwise. Since this solution is intractable in many cases, we try to estimate the affinity by estimating how frequent terms composing the segment are in the values of a field. For this, we rely on a set of *representative* terms for each field, such that most values in the field domain presents at least one term from this set. In practice, we consider that the lexicon index contains enough representative terms for identifying values.

Once we identify set P_k of potential values S_{ab} for each field f_k, we aim at finding a *mapping* between values in P_k and fields in the form-based interface with maximum aggregate affinity, such that (1) only a single segment is assigned to each field and (2) the segments select are non-overlapping, i.e., there are no segments S_{ab} and S_{cd} for $a < c$ in the mapping such that $b \geq c$. The last step in our approach consists in using the final mapping to fill out the fields of the form-based interface.

3. EXPERIMENTS

We have conducted *two* sets of experiments. In the first one, we used jobs postings for comparing iForm with CRF [2], a state-of-art data extraction method.In the second experiment we tested our method with real multi-typed web forms for submissions of *Car ads/offers* and *Mobile Phones offers*.

Comparison with CRF

This experiment compares iForm and CRF [2] for the task of extracting segments from text inputs and filling out a form. We took from the RISE Jobs collection a subset of 100 job postings whose segments to be extracted were manually labeled. These job postings form an adequate training set for CRF, since this method requires examples of values to be extracted to appear within the context they occur. This same set was used to create the lexicons for iForm.

Next, we tested both approaches using a distinct set of 50 documents, whose extraction outcome was available from RISE, allowing us to automatically verify the results.

According to the results presented in the Table 1, iForm had significant superior F-measure levels in 7 fields. The lower quality obtained by CRF is explained by the fact that segments to be extracted from typical free text inputs, such as jobs postings, may not appear in a regular context, which is an important requirement for CRF.

Experiments with a Real Form-Based Interface

To evaluate the performance of our approach within typical scenarios, we test iForm with a form-based interfaces from a Brazilian popular auction site (`http://www.todaoferta.com`). We performed experiments with *vehicle* form, that contains 4 text boxes and 28 check boxes – a total of 32 fields, and *mobile phones* form that consists of 2 text boxes and 35 check boxes – a total of 37 fields.

Field	iForm	CRF	T-Test	Wilcoxon
State	**0.98**	0.79	1.00%	1.00%
City	**0.93**	0.54	1.00%	1.00%
Language	**0.85**	0.66	1.00%	1.00%
Required Degree	**0.82**	0.00	3.00%	1.00%
Platform	**0.66**	0.5	1.00%	1.00%
Title	**0.57**	0.44	1.00%	1.00%
Company	**0.16**	0.09	1.00%	1.00%
Salary	0.14	**0.23**	2.00%	3.00%
Average	**0.56**	0.39	1.00%	1.00%
Submission-level	**0.61**	0.42	1.00%	1.00%

Table 1: Field-level f-measure for each field and submission-level f-measure for the comparative experiment.

To create the lexicons, we used real offers submitted during October 2008 to each interface. Tests were performed using other 50 offers for each interface, distinct from the offers used to create the lexicons. The results are presented in Table 2.

A detailed inspection on the offers entered by users in this interface, revealed that, in many cases, users simply cut and past mobile phone specifications from the manufacturer's web sites, leading to a high uniformity in the free text advertisements submitted to the text boxes.

Type of Field	# Fields	Precision	Recall	F-Measure
Text Box	*4*	85.00%	86.00%	0.85
Check Box	*28*	77.00%	77.00%	0.77
Average		81.00%	81.50%	0.81

(a) Vehicles

Type of Field	# Fields	Precision	Recall	F-Measure
Text Box	*2*	91.00%	87.00%	0.89
Check Box	*35*	99.00%	99.00%	0.99
Average		95.00%	93.00%	0.94

(b) Mobile Phones

Table 2: Field-level results for offers from *TodaOferta*.

4. CONCLUSION

In this paper we presented a framework called iForm for automatically using data values implicitly available in free text documents as input for a pre-existing form-based interface. We proposed a novel IR-based method for identifying implicit values in the free text documents and filling the form-based interface using them. We tested our approach with representative instances of the problem and found that it achieved better results than CRF, a state-of-art data extraction model. Our experiments also demonstrate that our approach is able to properly deal with different types of input fields, such as text boxes, pull-down lists and check boxes.

5. REFERENCES

[1] M. E. Califf and R. J. Mooney. Relational learning of pattern-match rules for information extraction. pages 328–334. American Association for Artificial Intelligence, 1999.

[2] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. pages 282–289. Morgan Kaufmann Publishers Inc., 2001.