FormYak: Converting forms to conversations

Scott Carter, Laurent Denoue, Matthew Cooper, Jennifer Marlow*
FX Palo Alto Laboratory, Inc.
Palo Alto, CA
carter,denoue,cooper,marlow@fxpal.com

ABSTRACT

Historically, people have interacted with companies and institutions through telephone-based dialogue systems and paper-based forms. Now, these interactions are rapidly moving to web- and phone-based chat systems. While converting traditional telephone dialogues to chat is relatively straightforward, converting forms to conversational interfaces can be challenging. In this work, we introduce methods and interfaces to enable the conversion of PDF and web-based documents that solicit user input into chat-based dialogues. Document data is first extracted to associate fields and their textual descriptions using meta-data and lightweight visual analysis. The field labels, their spatial layout, and associated text are further analyzed to group related fields into natural conversational units. These correspond to questions presented to users in chat interfaces to solicit information needed to complete the original documents and downstream processes they support. This user supplied data can be inserted into the source documents and/or in downstream databases. User studies of our tool show that it streamlines form-to-chat conversion and produces conversational dialogues of at least the same quality as a purely manual approach.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability;

KEYWORDS

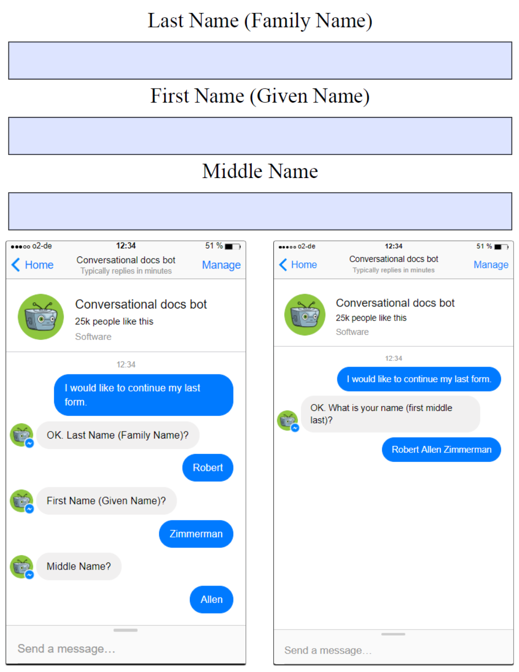
Form, conversational agent, interface

ACM Reference Format:
Scott Carter, Laurent Denoue, Matthew Cooper, Jennifer Marlow. 2018. FormYak: Converting forms to conversations. In *DocEng '18: ACM Symposium on Document Engineering 2018, August 28–31, 2018, Halifax, NS, Canada.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3209280.3229108

1 INTRODUCTION

Enterprise chat is an increasingly important channel for businesses to interact with consumers. Estimates suggest that chatbots, "will be responsible for cost savings of over $8 billion per annum by 2022" [4] and represent a billion dollar business by 2025 [3]. However, most businesses still require end users to complete paper- and web-based forms. This work focuses on adapting these processes for acquiring customer information to conversation-based interaction.

The core problem facing companies shifting from document-centered to chat-centered interaction is that documents are inherently more structured than chat. Information entered into forms can flow more-or-less directly into backend databases and processes. Chat operates via natural language which appeals to customers for its convenience. However, chat requires different styles of organizing content as well as more flexible recognition technologies. In this work, we focus on techniques to *translate* paper- and web-based documents into chat interfaces, as well as techniques to understand unstructured user responses in such a way that they can be reflowed appropriately into pre-existing backend databases and processes.


Last Name (Family Name)
First Name (Given Name)
Middle Name

I would like to continue my last form.
OK. Last Name (Family Name)?
Robert
First Name (Given Name)?
Zimmerman
Middle Name?
Allen

I would like to continue my last form.
OK. What is your name (first middle last)?
Robert Allen Zimmerman

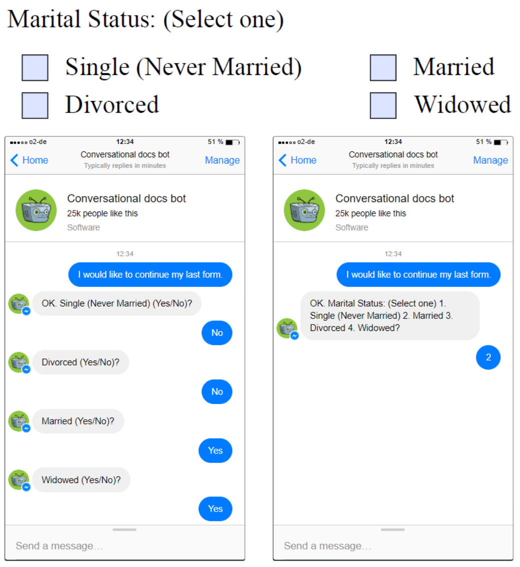
Figure 1: Converting a section of a structured document (top) directly from the output of document analysis tools results in a complicated conversational interface (bottom left). Our tool simplifies the experience (bottom right).

*Dr. Marlow is now at Google.

2 MOTIVATION

Many organizations have developed methods for automatic document analysis; it is worth exploring why these tools are not sufficient to solve the problem of converting structured documents to conversational interfaces. Translating structured document content to a conversational format is rarely straightforward. If the output of third-party document analysis tools is forwarded directly to end users the result is a chat bot UI that is at best unusable and at worst encourages end users to submit inaccurate information.

Consider the example in Figure 1, top. Third-party document analysis tools attempt to automatically associate all of the labels with corresponding fields. First, in many cases, these tools fail, producing incorrect (or empty) labels. However, even in the best case, if all field/label associations are correct, and those labels are used directly for a conversational UI, the resulting chat interaction is unnecessarily complicated for the end user (see Figure 1, bottom left). On the other hand, grouping and ordering field/label associations (as we describe later in this document) allows the UI to generate a single question for the end user (see Figure 1, bottom right). With our user interface, guide text is generated automatically from the ordered labels within the group, along with embedded separator characters (e.g., a comma or space) so that the system can easily parse the end user's response.


Marital Status: (Select one)

☐ Single (Never Married)	☐ Married
☐ Divorced	☐ Widowed

I would like to continue my last form.

OK. Single (Never Married) (Yes/No)?

No

Divorced (Yes/No)?

No

Married (Yes/No)?

Yes

Widowed (Yes/No)?

Yes

I would like to continue my last form.

OK. Marital Status: (Select one) 1. Single (Never Married) 2. Married 3. Divorced 4. Widowed?

2

Figure 2: Document content (top) converted directly from document analysis tools results in a conversational interface that encourages incorrect information from the end user (bottom left). Our tool presents a much simpler UI, and one in which the end user is nudged toward valid responses (bottom right).

In other cases, simply sending field/label associations directly to a chat user interface can encourage incorrect responses. Consider Figure 2, top, which includes four checkboxes. For PDFs and web pages, checkboxes are encoded as separate form elements, and therefore a naive automatic approach would generate a question for each checkbox (see Figure 2, bottom left). In this case, the open label, "Marital Status (Select one)" is lost since it is not associated with any field. Furthermore, since the checkboxes are not grouped, the user must answer each individually, which can lead to errors (e.g., responding "Yes" to two fields). However, with our system, since that open label is grouped with the other field/label associations, the resulting conversational UI restricts the user to a single response, as intended (see Figure 2, bottom right).

We describe interfaces here by which an administrative or *admin* user can verify or revise automatic field groupings and the ordering of information that is presented to end users. In our semi-automated admin UI, users can make changes to system generated summaries of field/label groupings. This can also be used to indicate that some fields are optional. For example, the admin user can indicate that "Apt. Number" is optional. If only a single entry is optional in a list, this is unambiguous (the system can assume that the option entry is not present if the user's response is a list of n-1 elements). However, if more than one entry is optional the system will not be able to confidently parse the resulting user response. Therefore the admin UI will ask the admin user to split groups with multiple optional responses.

3 RELATED WORK

Various chatbot platforms including Lex from AWS[1] and the Microsoft Bot Framework[2] can support conversational interaction with users to complete forms. However these tools require manual configuration in terms of defining desired entities for users to supply, and specifying their required formats.

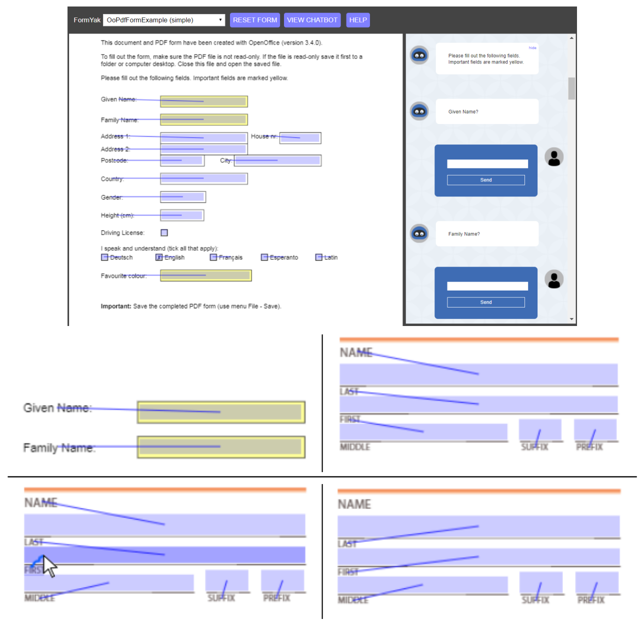
Other work involves injecting traditional UI components into a chat-based interface. Wu et al. [10] discuss a system that injects form widgets into the chat client, allowing users to submit responses to forms inside chat tools. The work does not address translating the form or its fragments into textual dialogue. Relatedly, Denoue et al. [8] describe a system that allows users to inject into chat messages multimedia content extracted from shared documents.

Our work aims to substantially accelerate the translation of structured documents to chat-based dialogues. We combine analysis and interface elements to enable rapid, flexible construction of a chat dialogue based on a source document's content and structure. Because other known conversion methods do not leverage a semi-automated interface they tend to be limited and require an admin user with programming background to manually transfer document information to the conversational interface. For example, fobi.io [2] is a chatbot conversion tool that only works with Google Forms, which restricts the view and controller (e.g., the interface) as well as the model (e.g., the database). Other frameworks work only with web forms and require modifying the underlying code itself [1].

4 SYSTEM

The core of this work is a semi-automated administrator tool, FormYak (Figure 3, top), that facilitates the creation of usable conversational dialogues [3]. Structured documents, or forms, are for the

[1]https://aws.amazon.com/lex/
[2]https://dev.botframework.com/
[3]https://youtu.be/yX_sCaBEKUA


This document and PDF form have been created with OpenOffice (version 3.4.0).

To fill out the form, make sure the PDF file is not read-only. If the file is read-only save it first to a folder or computer desktop. Close this file and open the saved file.

Please fill out the following fields. Important fields are marked yellow.

I speak and understand (tick all that apply):

Important: Save the completed PDF form (use menu File - Save).

Please fill out the following fields. Important fields are marked yellow.

Given Name?

Family Name?

Figure 3: The FormYak web UI (top) attempts to associate all fields and labels. In some cases (middle left), it succeeds. However, in other cases (middle right) labels are assigned to incorrect fields. Users can draw a line connecting fields and labels to correct these errors (bottom).

most part collections of fields and their associated labels. A naive approach to converting forms to conversations associates labels with fields, sends the label to the user as a message with a "?" appended, and injects the user's response back into the field. This approach produces chat dialogues that are both unnatural and error-prone as discussed earlier. Errors introduced by automatic analysis of form documents can further degrade the user experience.

Therefore, a tool converting forms to conversations must handle:

(1) Associating labels and fields as well as grouping associated labels and label/field pairs;
(2) Ordering associated pairs, groups, and other labels;
(3) Converting (sets of) labels into questions.

To address these issues, FormYak associates, groups, and orders labels and fields. First, for web forms, associating labels and fields can be trivial: fields and labels are bound together in HTML markup through name and form fields. For example, the HTML "<input ... name=myName>" is linked to "<label ... for=myName>". For form-enabled paper (PDF) forms, often similar markup is available in the PDF content itself. If not, the system can take advantage of off-the-shelf tools [6, 7]. For web forms, a server-side rendering engine based on Chrome Headless images and extracts a JSON representation of the form fields (text and bounding boxes). For PDF forms, we use PDFBOX (Apache Project) to image and also extract form fields if the PDF is form-enabled. For web forms, the algorithm uses LABEL fields associated with INPUT fields. For PDF forms, the system compared the location of text with close-by form fields and also matches form field names with any nearby open text to give users an initial grouping. The image of the forms without the text is used to group fields, e.g. found inside a box using heuristic image processing methods.

Grouping and ordering is more complicated. Because these tasks are useful almost exclusively for translating content from forms to chat interfaces, this translation task is an emergent need (past work has investigated content ordering in text documents but is not applicable to forms [9]). We propose new methods for grouping and ordering that combine offline analysis with a user interface that takes advantage of familiar actions.

The UI must group associated pairs of fields and labels as well as form text that is not necessarily associated with any field, which we call open labels. The UI orders individual associated pairs, open labels, as well as groups, and provides mechanisms to manually revise automatic associations. At the same time, corresponding changes to the chat dialogue are rendered to verify the correctness of manual refinements. The web interface displays the web or PDF form and uses the JSON representation to overlay boxes. A CANVAS allows users to draw over the form in order to create or modify groupings. A chat UI shows the form translated to chat dialogue. The tool generates Microsoft's FormFlow-compatible dialog scripts (using NodeJS) that can be deployed to several popular chat platforms such as Slack, Messenger, Skype and plain text-based SMS.

We use two methods to allow admin users to associate, group, and order fields: a crossing-based interface [5] as well as a click-based interface[4]. The crossing interface uses two simple gestures for grouping (regions) and ordering (lines). Drawing a line associates all label/field pairs and open labels with boundaries that cross that region or line (see Figure 3). Grouping label/field pairs and open labels together signals to the backend chat generation system that all of those elements should be combined into a single question to the chat user. If checkboxes are grouped together this can trigger other changes in the chat dialog, such as enumeration.

Furthermore, drawing a straight line down and/or to the right orders label/field pairs, open labels, and groups. If the mark starts on an item that is already ordered then the ordering enumerates up from that position. Otherwise the ordering enumerates up from the current maximum ordered item. Label/field associations themselves have no notion of ordering, so in that case lines only create associations. Users can also delete any field/label associations, groupings, or orderings at any time with a keystroke.

Finally, end user input must be directed to the correct field in the original paper- or web-document. For basic field/label associations, this is straightforward. When an admin clicks on a question in the chat preview the form viewer automatically navigates to the field, label, or group associated with that question. When the question refers to a group, the admin can edit the ordering of the fields within the group to match the ordering that the end user will supply.

The admin can also use a separate panel to enforce other syntax rules on the end user's response (e.g., that the response should be a choice from a list, open text, open numeric, day/time, or, for experts, a regular expression). To remap user responses to the original source document, ordering and grouping needs to be inverted faithfully. In cases such as Figure 1, in which three name fields are grouped in one chat question, strings can be processed in the requested order.

[4] *admin* users are distinct from the *end* users whose information is solicited in chat to complete the original document

Fields may have certain response requirements so that they can be remapped to a common format for downstream entry in a database. These constraints can take the form of different patterns, e.g. any text, any number, or a specific format such as MM/DD/YY or DD/MM/YYYY, etc. Admin users can supply such response constraints as well as example entries in the chat panel of the interface. The system can optionally infer constraints from metadata gleaned form the form itself or learned from past entries. In cases where there is neither document information nor any common sense default for a field nor any response constraints set by the admin user, the system can assume unformatted input by default.

5 EVALUATION

We conducted a preliminary user evaluation with six people (researchers and interns at a research lab) to assess the user experience of the system and its various features. The study took 30-40 minutes to complete, and participants interacted with FormYak via a web browser on their own desktop/monitor setups in an office setting. First, they read a brief description of FormYak and watched an instructional video showing the main features and actions they could perform using the tool. They were encouraged to stop the video after each feature was demonstrated and given the opportunity to re-create that action using a test form in a separate window.

After familiarizing themselves with the tool's main features, participants were then asked to engage with two additional sample forms. In this case, they were shown FormYak's automated output and asked to edit the chat UI output for two different forms. Instructions were: "*Try to link text regions, fix any missing field/label associations, group together any field/label pairs that you think should be a single question, and group checkboxes together (if they are all part of the same question). Finally, experiment with ordering the questions in the chat and fixing the wording of text chat questions.*"

When they had finished with the two forms, participants filled out a short survey asking them about their familiarity with form creation tools and chat UIs, and rated the overall usability of FormYak as well as the difficulty of the four main actions (updating fields, grouping, changing order of questions and changing question text). Finally, participants were asked to explain which feature of the interface was most helpful (and why) and which feature of the interface was the most difficult to understand (and why).

We logged user actions on the forms in order to quantify the number of times they performed the different actions of making field/label associations, grouping questions, ordering questions, and editing text/labels.

5.1 User behaviors

Based on logs from participants' use of the system, we observed that the most frequent action was to associate fields with labels. Participants added or changed field-label associations a median of 9 times on the first form and a median of 10 times on the second form. Participants made between one and six groupings of questions on the first form (which was more complex than the second form). They edited the chat output a median of 7.5 times on the first form and 8.5 times on the second form. Finally, they created a median of 8.5 ordered groups on the first form and 1.5 ordered groups on the second form (again, likely because the first form was more complex and followed a two-column format).

5.2 Survey responses

Overall, changing the question text in the chat window had the highest usability (three ratings of "very easy" and two of "easy"). Changing the order of question groups was seen as the most difficult, with three participants rating this task as difficult and only one finding it to be easy.

In accordance with these ratings, all participants agreed that associating fields and labels, as well as grouping multiple fields into a single question were the most helpful features of the interface. One participant explained the utility of grouping multiple fields together. "*It reduces the number of questions and thus increases the usability of the chatbot.*" Similarly, participants mentioned in the open-ended question that re-ordering groups of questions was more difficult to understand, although this may have been due to the particular interaction technique chosen for ordering (clicking and dragging using a 'lasso' motion).

In terms of suggestions for future design, there were some small usability suggestions such as implementing an "undo" or "redo" feature rather than requiring users to clear the form or delete several links in order to fix a mistake. More broadly speaking, users also suggested that editing or deleting could be facilitated by not only allowing a user to jump to the appropriate chat bubble by clicking on the form, but also vice-versa: "*It should enable double-linking between the form fields and the chatbot questions, i.e., clicking either one will highlight the corresponding element on the other side.*"

6 CONCLUSION AND FUTURE WORK

Users are shifting away from desktop interfaces and toward conversational chat applications on mobile devices. The tool we built and presented here, FormYak, aides this transition, making it easier for chat creators to translate highly structured documents (forms) to conversational interfaces. In future work, we plan to improve the FormYak interface based on feedback from our user study and also extend the tool to support less structured documents.

REFERENCES

[1] 2016. SPACE10. https://medium.com/conversational-interfaces/introducing-the-conversational-form-c3166eb2ee2f. (2016).

[2] 2017. Fobi.io. https://fobi.io/. (2017).

[3] 2017. http://www.grandviewresearch.com/industry-analysis/chatbot-market. (2017).

[4] 2018. Juniper Research. https://www.juniperresearch.com/press/press-releases/chatbots-a-game-changer-for-banking-healthcare. (2018).

[5] Johnny Accot and Shumin Zhai. 2002. More Than Dotting the I's — Foundations for Crossing-based Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02)*. ACM, 73–80. https://doi.org/10.1145/503376.503390

[6] Xin Tao Yun Li Cao Shi Canhui Xu, Zhi Tang. 2013. Graph-based layout analysis for pdf documents. In *Proc. of SPIE 8664, Imaging and Printing in a Web 2.0 World IV*, Vol. 8664. 866407.

[7] Christopher Andreas Clark and Santosh Divvala. 2015. Looking Beyond Text: Extracting Figures, Tables and Captions from Computer Science Papers. In *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence.*

[8] Laurent Denoue, Scott Carter, Jennifer Marlow, and Matthew Cooper. 2017. DocHandles: Linking Document Fragments in Messaging Apps. In *Proceedings of the 2017 ACM Symposium on Document Engineering (DocEng '17)*. ACM, 81–84. https://doi.org/10.1145/3103010.3121036

[9] J. L. Meunier. 2005. Optimized XY-cut for determining a page reading order. In *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*. 347–351 Vol. 1. https://doi.org/10.1109/ICDAR.2005.182

[10] Min Wu, Arin Bhowmick, and Joseph Goldberg. 2012. Adding Structured Data in Unstructured Web Chat Conversation. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, 75–82. https://doi.org/10.1145/2380116.2380128