June 20, 2018

Smart linkification of content within applications

Jan Althaus

Thomas Binder

Jacek Jurewicz

Lukas Zilka

Sebastian Millius

See next page for additional authors

Inventor(s)
Jan Althaus, Thomas Binder, Jacek Jurewicz, Lukas Zilka, Sebastian Millius, Matthew Sharifi, Regina O'Dell, Abodunrinwa Toki, and Evelyn Kao

Smart linkification of content within applications

ABSTRACT

When a user initiates text selection within an application, the operating system can examine text content displayed within the application to predict text selection bounds along with a possible destination application for the selected text. Until the user initiates selection, there may be no indication that a piece of text content might be actionable. Further, the functionality may not work as intended in cases where application developers implement a custom operation for the input mode utilized for passing the text content and associated action from one application to another. With user permission, this disclosure applies regular expression parsing and neural network processing to the text portion of the on-screen content to detect text entities that might be actionable by the OS or other applications on the device. After merging the actionable text entities identified via either of the two techniques, the corresponding text is presented, e.g., by underlining the corresponding text and linking it to invoke the corresponding action.

KEYWORDS

- Linkify
- Content linkification
- Link generation
- Link creation
- Text parsing
- Mobile OS
- Actionable text
- Automatic text selection

- Neural network
- Unstructured text

BACKGROUND

When using devices, such as mobile devices, people often utilize applications native to the operating system (OS) of the device. Such applications typically serve a specific function, such as texting, providing driving directions, etc. In many cases, a user need involves transfer of data from one application to another.

For example, a user may wish to look up an address received via a text message in a mapping application. In such cases, the user needs to select the content of interest from one application and perform the relevant actions necessary to act upon the content in the other application. For instance, in the use case of looking up the address, the user selects and copies the text of the address received via the text message and pastes it in the mapping application.

Some operating systems include capabilities to examine the content displayed within an application to predict text selection bounds when a user initiates text selection along with a possible destination application for the selected text. When the selection capabilities are triggered upon user action, there may be no indication that a piece of text content might be actionable until the user initiates text selection. Moreover, the functionality may not work as intended in cases where application developers implement a different custom operation for the input mode, such as long press, that is utilized by the OS for passing the text content and associated action from one application to another.

DESCRIPTION

This disclosure examines the text of the content presented on the screen of a device, when the user and/or the mobile application as configured by the user permits the operating system to perform such examination. Text entities within on-screen text content that might be actionable by the OS or other applications on the device are detected. Such detection is user configurable, and is performed upon specific user permission. Users are provided with options to turn off the detection and linkification features described herein entirely, limit their use to specific applications, etc. Detection and linkification of text is performed only for the users and in the contexts in which the users have permitted such actions.

Detection of actionable text entities can be carried out using a combination of two techniques:

(i) identification of common text patterns, such as email addresses, airline confirmation codes, via regular expressions; and

(ii) discovery of various kinds of text entities of variable length spans via neural network processing of unstructured text content.

Such detection is performed using regular expression matching and neural networks implemented on the mobile device when the user permits use of such techniques. For users that do not provide permission, no detection is performed and the linkification functionality is turned off. When actionable text is detected, text entities identified via either of the two techniques are ‘linkified,’ e.g., presented by underlining the corresponding text to mark it as a link that can be clicked, tapped, or otherwise selected to invoke the corresponding action.

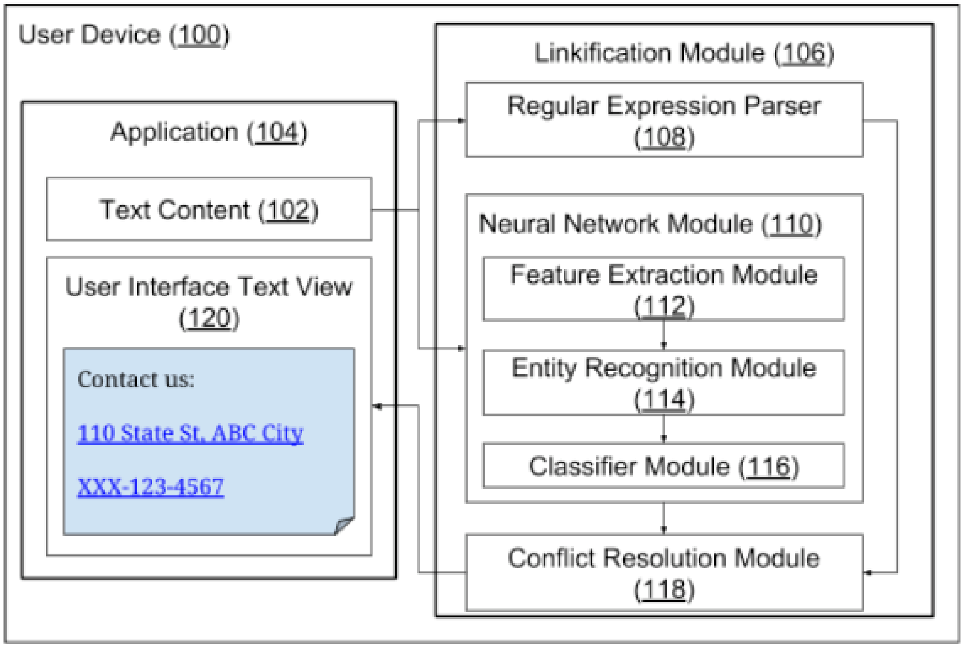
User Device (100)
Application (104)
Text Content (102)
User Interface Text View (120)
Contact us:
110 State St, ABC City
XXX-123-4567
Linkification Module (106)
Regular Expression Parser (108)
Neural Network Module (110)
Feature Extraction Module (112)
Entity Recognition Module (114)
Classifier Module (116)
Conflict Resolution Module (118)

Fig. 1: Linkification of text by regular expression and neural network processing

Fig. 1 shows an example of a user device that implements the techniques of this disclosure. When a user device (100) has text content (102) presented within an application (104) executing on the device, the text content is passed to the linkification module (106), e.g., prior to being displayed to the user.

Within the linkification module the received text content is processed in parallel by a regular expression parser (108) and/or a neural network module (110). The regular expression parser is configured to examine the input text and identify common entities with a well-defined structure, such as email addresses, phone numbers, street addresses, airline confirmation codes, etc. that are actionable by the OS or by one or more other applications on the device.

In parallel, the input text is converted by a feature extraction module (112) into a dense feature representation that is suitable for processing by a neural network. For instance, such a process may involve the use of word embeddings or chargram embeddings, such as hash chargram embeddings. For text snippets of different possible length spans between one and a given maximum length within the feature representation of the input, a score is assigned by the entity recognition module (114) to indicate the likelihood of the span being an entity belonging to one of the types of entities that the model is capable of recognizing. Each text snippet with a score higher than a specified threshold value is passed to a classifier module (116) to be assigned the type of entity corresponding to the text snippet.

After the text input has been processed in parallel by the regular expression parser and the neural network module, the outputs of the two processes are merged by conflict resolution module (118) to resolve conflicts between the two sets of entity annotations and yield a single consistent set of actionable text entities within the original input text content. This set is relayed back to the application for displaying within its user interface (UI) text view (120) by marking all actionable text entity spans as hyperlinks that the user may click or tap. In the example shown in Fig. 1, the address “110 State St, ABC City” and phone number “XXX-123-4567” are detected as actionable text, and are shown as hyperlinks, while the text “Contact us:” is not hyperlinked.

Interacting with the hyperlink provides shortcuts for quickly performing relevant actions corresponding to the entity assigned to text within an application capable of handling the entity. For example, interacting with a hyperlink where the underlying text is detected to be a street address can open the mapping application and display a pin at the location on the map corresponding to the particular address.

The linkification process can be repeated each time the text content within the application view is modified. The developer of an application can choose to implement the linkification capability for the entire application or only for certain specified views within the application. The threshold value for the likelihood level used to mark a piece of text as corresponding to an entity of interest can vary based on the type of entity or application. Further, the threshold value may be set by the neural network module or may be configurable by the developer or the user.

The neural network model is trained offline using a set of documents that contain a variety of entities of interest. For instance, the neural network model may be trained with text entities extracted from webpages. Alternatively, in federated settings, the neural network models may be trained using federated learning. Several iterations may be required for the neural network to provide output text with entity annotations. Therefore, the classifier operation is typically batched across several possible text snippet spans. Owing to the precise nature of operation with precisely specified text structure, the regular expression parser is often quicker than the neural network processing, with text entities detected in a single pass.

Since the regular expression parser and neural network processing operate differently, each may detect different number and types of entities within the input text content. As a result, merging the output of the two approaches, after resolving any conflicts, can provide a larger and richer set of entity annotations than that obtained via either of the approaches alone. Both approaches can support languages other than English. The language of operation may be obtained from the language chosen by the user for the application or the OS. Alternatively, the language may be detected by processing the text content of the application using a language identification model.

Further to the descriptions above, a user may be provided with controls allowing the user to make an election as to both if and when systems, programs or features described herein may enable collection of user information (e.g., information about a user’s social network, social actions or activities, profession, a user’s preferences, or a user’s current location), and if the user is sent content or communications from a server. In addition, certain data may be treated in one or more ways before it is stored or used, so that personally identifiable information is removed. For example, a user's identity may be treated so that no personally identifiable information can be determined for the user, or a user's geographic location may be generalized where location information is obtained (such as to a city, ZIP code, or state level), so that a particular location of a user cannot be determined. Thus, the user may have control over what information is collected about the user, how that information is used, and what information is provided to the user.

CONCLUSION

With user permission, this disclosure applies regular expression parsing and neural network processing to the text portion of on-screen content to automatically detect text entities that might be actionable by the OS or other applications on the device. After merging the actionable text entities identified the two techniques, the corresponding text is presented, e.g., by underlining the corresponding text and linking it to invoke the corresponding action. In this manner, the OS or application can provide a user benefit by enabling a user to perform a click, tap, or other selection operation to take actions on detected actionable text, without first having to select the text.