September 22, 2017

Decoding text with errors due to auto-correction

Thomas Deselaers

Victor Carbune

Decoding text with errors due to auto-correction

ABSTRACT

Auto-correction techniques implemented in software keyboards attempt to mitigate user errors in text entry by analyzing user input and automatically generating corrections. This approach can result in inaccurate corrections (auto-corruptions), e.g., when users enter words that are out of vocabulary, uncommon words, abbreviations, etc. When such phrases are sent, e.g., as messages, recipient users may not understand the sender's intent or make sense of the received text.

This disclosure describes techniques to decode received text that has errors produced by auto-correction. The techniques include reconstructing possible touch point sequences that led to the phrase, determining alternative phrases that correspond to the determined sequences, and presenting the alternative phrases on the recipient device. The techniques also enable the sender to quickly send post-corrections.

KEYWORDS

- autocorrect
- software keyboard
- virtual keyboard
- touchpoint
- beam search

BACKGROUND

It is difficult to enter text on mobile and wearable devices such as smartphones, watches, etc. due to various factors such as the small size of a virtual keyboard relative to human fingers, lack of tactile feedback, etc. Software auto-correction techniques that automatically correct user

entered text are common in such devices. Auto-correction techniques typically employ a dictionary and a language model. Different software keyboards use different language models. The language model can also be represented in different ways, e.g., as an n-gram model, a neural network, etc.

Autocorrect mitigates errors in text entry that occur due to small size of a virtual keyboard, limited range of physical motion while holding a mobile device, etc. For example, on a smartphone with a touchscreen display, auto-correction analyzes touch inputs on and maps the touch inputs to alternative text. Depending on user preferences, auto-correction can be automatically applied or provided as suggestions that a user can accept or reject.

When auto-correction is automatically applied, such techniques sometimes produce incorrect results, e.g., a word or a phrase that the user didn't intend. For example, users can sometimes type out-of-vocabulary words, uncommon words, abbreviations, etc. that the auto-correct technique might not know of. In such instances, autocorrect turns into auto-corrupt, turning user text input into incorrect text output. When such text is sent to another recipient (e.g., another human user, a website, etc.) it is often difficult for the recipient to understand what the sender intended. In some instances, auto-correction can even auto-correct a correct sentence into one that is a seemingly random collection of words.

Further, a recipient of auto-corrected text has to guess what the sender might have intended, or ask for clarification from the sender. This may lead to misunderstanding or repeated back and forth until a mutual understanding is obtained. No automatic techniques are available to decode such text at a recipient end. Auto-correct can sometimes lead to humorous results; there are several websites devoted to displaying humorous autocorrect failures.

DESCRIPTION

This disclosure describes techniques to mitigate auto-correct errors.

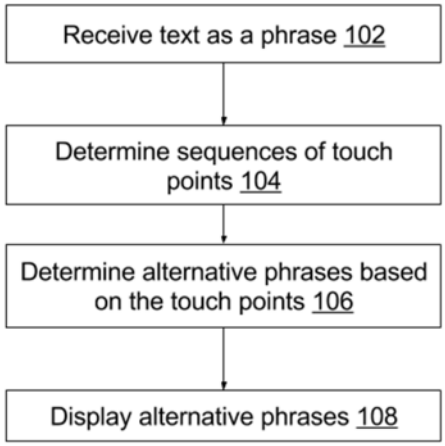

Fig. 1: Auto-corruption countermeasures

Fig. 1 illustrates a method to decode phrases caused by errant auto-correction, per techniques described herein. A recipient device receives text (102). Based on the received text, multiple alternative sequences of touch points that correspond to the received are determined (104). Based on the determined sequence of touch points, alternative phrases are determined (106). One or more of the alternative phrases are displayed on the recipient device (108).

The text can be received from a sender device, e.g., in a messaging application. The method of Fig. 1 can be user-triggered, e.g., the recipient can select or highlight a portion of the received text for which alternative phrases are then determined. Alternatively, alternative text can be determined automatically. The process of determining alternative text is also referred to as decoding.

To determine the alternative phrases, alternative sequences of touch points that lead to the text are determined, e.g., using beam-search. This uses the auto-correction technique backwards. To perform auto-correction, beam search is used to determine likely text phrases from a received sequence of touch points, e.g., on a touchscreen or gesture-enabled device that the sender uses

for text entry. To determine alternative sequences of touchpoints, beam search is used to determine likely sequences that correspond to the text. A motion smoothness model is used to rule out sequences that are less likely, e.g., due to the limited physical range of motion when the sender enters text using a virtual keyboard.

Once the touch points are determined, alternative phrases are determined based on the touch points using auto-correction. Auto-correction uses a dictionary and/or a language model to limit the alternative phrases to phrases that are likely to occur. Since auto-correction works on the recipient device in a similar fashion as on a sender device, phrases that are displayed to the recipient are like those displayed, e.g., as auto-correct suggestions, on a sender device. Based on the alternative phrases, the recipient can interpret what the sender likely intended to write. The alternative phrases can include a phrase that directly maps to touch points where auto-correction is not applied.

Further, similar techniques can be employed to display possible alternative phrases to a sender, e.g., when the sender notices after sending that there was an error in the text. The sender can select a phrase from the text and alternative phrases are displayed. The sender can then send quick follow up texts as corrections. In these instances, sender touch points may be saved on the sender device. Due to saved touch points on the sender device, more accurate alternative phrase suggestions can be generated by the sender device than by a recipient device that reconstructs touch points based on received text. Further, the alternative phrase suggestions could be stored instead of touch points, e.g., when there is high likelihood that auto-correction might have introduced an error.

Another way the techniques can be implemented is for the sender device to transmit both text and touch point data, if the sender permits sending of such data. Further, the sender device

can transmit already-generated alternative phrases. The availability of sender touch point data and/or alternative phrases aids a recipient device in decoding auto-corrupted text. If a user does not grant permissions, no additional data beyond the auto-corrupted text is sent.

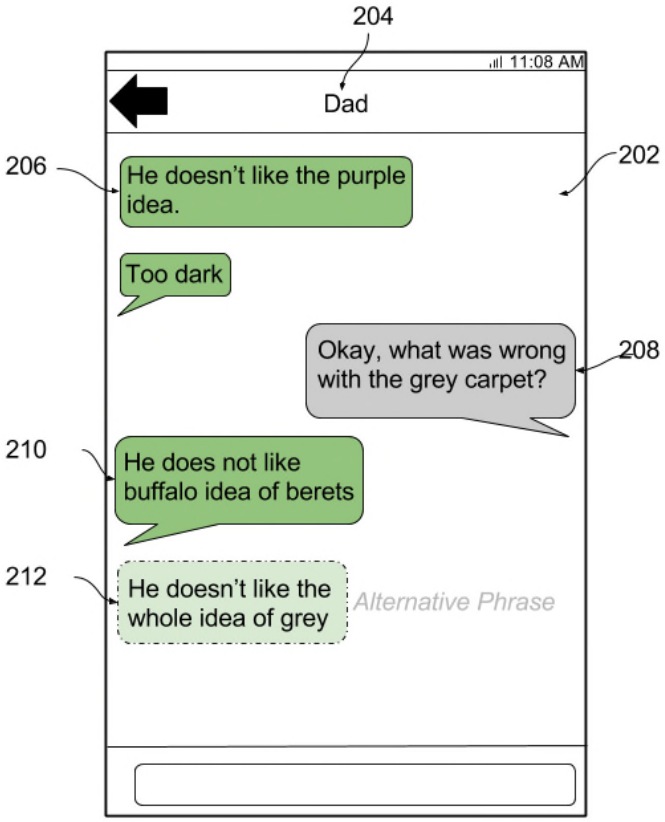

Fig. 2: Example of auto-corruption and decoding

Fig. 2 illustrates an example of auto-corruption and the subsequent decoding of text via an alternate phrase determined using the techniques described herein. A recipient using smartphone (202) has received messages (206) from a contact (204). The recipient responds with the message illustrated in grey (208). As a response to the sent message, the smartphone receives a nonsensical sentence - "He does not like the buffalo idea of berets" (210).

Using the techniques described herein, an alternative phrase (212) - "He doesn't like the whole idea of grey" - is determined and displayed. The alternative phrase can be sent back to the

sender, e.g., for the sender to indicate whether the phrase “whole idea of grey” was intended instead of “buffalo idea of berets,” avoiding the tedious sending of messages back and forth to resolve the sender’s intended text.

The described techniques can be implemented in operating systems used for devices that use software keyboards, e.g., wearable and mobile devices. The techniques can also be implemented in software keyboard applications. Further, the techniques can be included as part of software applications that provide text entry options, e.g., web browsers, word processors, etc.

Training data for the techniques described here can be gathered through a user feedback mechanism provided to the user. When users permit, scoring heuristics that highlight parts of a conversion where auto-corruption happened can be determined. For example, with user permission, it is observed when a recipient asks a sender a follow-up question, e.g., “what did you mean by that”, then the sender clarifies, and the recipient states “oh, got it” or similar patterns. Once such a detection/feedback system is in place, the data can be used to improve the auto-correction model. Alternatively, a version of the correction model applied on the recipient side can be applied on the sender side by detecting that the sent text had an auto-correction errors and notifying the sender.

In situations in which certain implementations discussed herein may collect or use personal information about users (e.g., user data, information about a user’s social network, user's location and time at the location, user's biometric information, user's activities and demographic information), users are provided with one or more opportunities to control whether information is collected, whether the personal information is stored, whether the personal information is used, and how the information is collected about the user, stored and used. That is, the systems and methods discussed herein collect, store and/or use user personal information

specifically upon receiving explicit authorization from the relevant users to do so. For example, a user is provided with control over whether programs or features collect user information about that particular user or other users relevant to the program or feature. Each user for which personal information is to be collected is presented with one or more options to allow control over the information collection relevant to that user, to provide permission or authorization as to whether the information is collected and as to which portions of the information are to be collected. For example, users can be provided with one or more such control options over a communication network. 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. As one example, a user's identity may be treated so that no personally identifiable information can be determined. As another example, a user's geographic location may be generalized to a larger region so that the user's particular location cannot be determined.

CONCLUSION

This disclosure describes techniques to decode received text that has errors produced by auto-correction. The techniques include reconstructing possible touch point sequences that led to the phrase, determining alternative phrases that correspond to the determined sequences, and presenting the alternative phrases on the recipient device. The techniques also enable the sender to quickly send post-corrections to the recipient when the sender notices an autocorrect error or when the recipient asks for a clarification.