Vol.2012-HCI-149 No.6
2012/7/19

GhostTweet: 短い文の入力のための新たなパラダイム

綾塚 祐二[1,†1,a)]　那和 一成[1,b)]

概要：本稿では、Twitter に発信するメッセージのような短い文の入力をターゲットとした新たな作文手法「完成文変更方式」を提案する。既存の文章入力システムでは、全ての文字をタイプするなどにより文を先頭から入力し、文末まで入力が完了したところで文が完成するが、我々の提案する手法は自動的に生成された文をベースに変更を加えていくという方式で、常に発信可能な完成した文があるという状態を保つことにより、その時点で掛けられる手数によらず発信文を生成することができる。その実装へ向けた第一段の試作として、位置情報に基づき自動的に複数の候補文を生成してユーザに提示し、発信のできるシステム GhostTweet を試作した。

GhostTweet: a New Paradigm for Short Message Composition

AYATSUKA Yuji[1,†1,a)]　NAWA Kazunari[1,b)]

Abstract: We propose a new text input paradigm for short messages, named *Composition by Altering Complete Sentense*. A system with a method under this paradigm provides a complete sentense. A user can alter a part of the provided sentense or request another sentence to make it suitable to his/her intention. Unlike traditional text input methods, with which a sentense is imcomplete until a user type or select the last word, the system always shows a complete sentense as a candidate according to user's input. This means our method is available even for a user who is only able to share little resource for input. We implemented a prototype application, named GhostTweet, to investigate the paradigm. GhostTweet shows some automatically generated candidate sentense to tweet, according to location information.

1. はじめに

携帯情報端末が普及し、SMS (Short Message Service) や Twitter などの短い文章を用いたコミュニケーションサービスが広まるに連れ、ちょっとした隙間時間に短い文を入力したい、という需要が増えてきている。既存の文章入力システムでは、全ての文字をタイプするか、あるいは PoBOX[2] のような予測変換手法を用いたとしても、単語や文節などの最初の数文字をタイプし、そこで挙げられた単語等の候補を選択して文を組み上げて完成させる必要がある。これらの入力の手数を減らすことができれば、短い文を入力したいユーザにとっては大きな利益となる。

メッセージを発信するための別のアプローチとしては、自動、もしくはボタンなどによる確認のみで位置情報を含めた定型文を発信するシステムもある (「今ココなう!」[*1] や「AQUA SOCIAL DRIVE」[*2] など)。これは入力の手数は最低限で済むが、発信できる文が固定的になってしまうという欠点がある。

本稿では、短い文の入力をターゲットとし、これらの既存の手法の間を埋める、すなわち手数を予測変換よりも少

[1] 株式会社トヨタ IT 開発センター
Toyota InfoTechnology Center Co., Ltd.
[†1] 現在，株式会社電通国際情報サービス オープンイノベーション研究所
Presently with Open Innovation Laboratory, Information Services International-Dentsu, Ltd.
[a)] ayatsuka@acm.org
[b)] nawa@jp.toyota-itc.com

[*1] http://imakoko-gps.appspot.com/
[*2] http://aquadrive.jp/

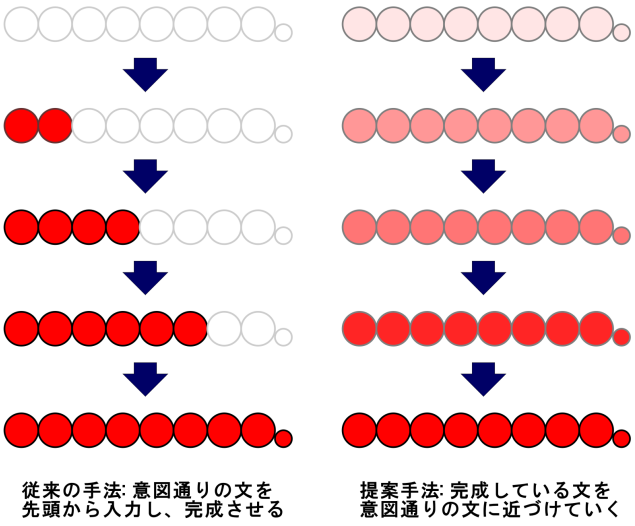

図 1 従来の手法と提案手法の概念比較

なく抑え、文のバリエーションをなるべく多くする作文手法を提案する。ユーザが一から文を作成するのではなく、最初に自動的に生成された候補の文が提示され、ユーザはそれを変更していくというアプローチにより、その時々で掛けられる手間・手数に応じたバリエーションを持つ文を生成する。

以下の節では、まず提案する手法のコンセプトの詳細を説明し、その実現へ向けた試作アプリケーションを紹介する。

2. 自動生成と変更をベースにした作文

前述のように、既存のシステムでの文の入力は、ユーザが先頭から作成していくか、既に出来上がっている文を選択するという両極端な手法によって行われる。この二つの中間に位置し、その時々に掛けられる手間に応じた入力を許す手法に求められる性質は、

(1) 手数に応じてバリエーションが増やせ、
(2) 任意のタイミングで (少しの手数を掛けた段階でも、多くの手数を掛けても) 発信可能、すなわち文として「完成」している

という二つであると考えられる。

これらの性質を満たすためには、ユーザのそこまでの入力に応じ、常に文が (文末まで) 自動生成されていればよい。POBox などの予測変換入力手法では、ユーザが何らかの入力を始めると続きの語やフレーズが提示されるので、入力途中の文は (意図通りであるが) 未完成な状態であり、文が完成している状態になるのは最後のみである[*3]。それに対し、ここで提案しているのは、ユーザが何も入力せずとも文末まで完結した文 (意図とは必ずしも充分に一致しない) が提示され、ユーザの入力によって文は変更され (意図通りのものに近づけられ) るが、常に完成した状態で提示される、という手法である。従来手法と提案手法の違いを模式化したのが図 1 である。

[*3] 文を完成させた後や途中に推敲・編集を行う場合もあるが、簡単化のためにここでの議論には含めない。

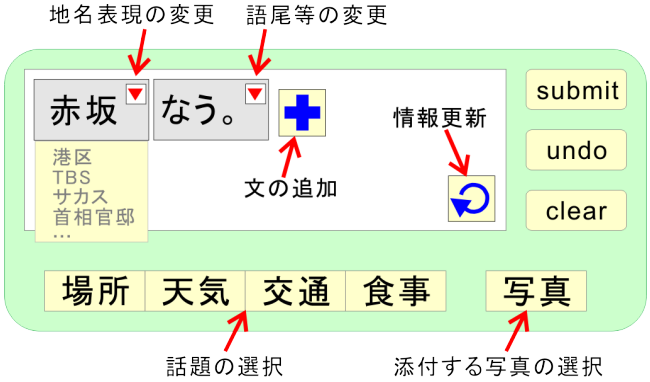

図 2 完成文変更方式を用いたアプリケーションのインタフェースイメージ

ユーザによる文の変更を、文字単位で入力するのではなく、語や文節などの単位で他の候補を出し選択させることで、上述のような「少ない手数でかつ常に完結した状態を保った変更」が可能になる。この提案手法をここでは**「完成文変更方式作文」**と呼ぶことにする。図 2 はこの手法の実装イメージの一例である。このイメージでは生成する文の話題の選択や、文の追加などの機能も想定している。図では明示していないが、もちろん文字単位での修正手段も用意しておくことで、手間を掛けられるときにはより柔軟な文にすることもできる。

この手法は絵画と対比することができる。最初に提示される自動生成の文は、構図と描かれている物体のおおよその形が示されたラフスケッチのようなものである。その後、ラフな描画を、必要な部分から詳細化していくかのように、文の部分々々を変更し、ユーザの意図に近いものへとカスタマイズしてゆく。最初の状態や途中状態でも全体像は見えており、(もちろん、描画の手腕にもよるが) どの時点でも「絵」としては成り立っているようにすることができる。[*4] 対して従来の手法は、白紙の上に端から細かく絵を描いていくことになぞらえられる。

3. 文の自動生成

完成文変更方式が実用的に使えるかどうかは、文の自動生成や、置き換える候補の単語・文節のリストの生成が如何に適切に行えるかに左右される。「いつ・どこで・だれが・何を・どうしたゲーム」「5W1H ゲーム」等の名前で呼ばれる遊びのように、ほぼランダムに言葉を用意して文を生成するのでは、奇妙で面白い文は得られるが、状況に応じた適切な文は確率の低い偶然でしか得られない。いくつ

[*4] 慶應大学・山中俊治氏の Twitter 上での発言「…これができる人の絵は、どの段階で止めてもそれなりの空間表現がなされている。つまりいつも完成しているのだ」を参照されたい https://twitter.com/Yam_eye/statuses/75182470198333441

上野動物園		東京タワー	
ない	105	ない	66
いい	49	綺麗	65
可愛い	34	良い	48
楽しい	34	いい	46
みたい	23	好き	38
好き	18	きれい	36
らしい	15	みたい	34
よい	13	申しわけ	21
面白い	13	高い	20
かわいい	12	楽しい	17

図 3 抽出した場所名と修飾語の例

かの単語からそれを用いたもっともらしい文を自動生成する、といった研究もある [1] が、元となる単語を如何に用意・選択するかが問題となる。

3.1 位置情報の利用

「状況」としてシステムにとって検知・利用しやすく、また効果も高いと思われるのは現在位置の情報である。たとえば図 2 の中でも挙げられている「○○ (地名や施設名) なう」という文は Twitter 上などでよく見られるものである。位置情報は GPS などにより携帯型端末でも緯度経度の数値として取得可能であるが、それを適切な地名や施設名 (以下、これらをまとめて「場所名」と呼ぶ) などに変換する必要がある。計測精度の問題や、どの粒度の名称を用いるかなどの問題、あるいは単に近隣を通過する場合などを考慮すると、適切な名称をシステムが一意に決定することは難しいが、提案手法では複数の変更候補をユーザに提示し対応することが可能である。

これに加えて、本稿では場所名に関連した修飾語のリストを用意し、利用することを提案する。場所名と、この関連した修飾語をテンプレート (複数用意しランダム、もしくは何らかの基準で選択する) に当てはめることで候補文を生成すれば、自動生成文及びそのカスタマイズのための語の候補のバリエーションを確保できるであろう。

3.2 場所名と修飾語の抽出

場所名など、場所に関連した語の抽出には、位置情報付きのメッセージを収集し、解析することが考えられる。しかし、システムを試作するために位置情報付きの Twitter 上のメッセージ (ツイート) を実際に収集してみたところ、位置情報付きの日本語のツイートは少なく、解析に充分な量は得られなかった。地図情報データベースを用いたリバースジオコーディングと呼ばれる手法により緯度経度情報を地名に変換することはできるが、施設名などは得にくい。そこで、今回の試作では、場所名として Foursquare[*5] のスポット名を利用し、その名称を含むツイートから修飾語を抽出するという手法を採った。

[*5] http://foursquare.com/

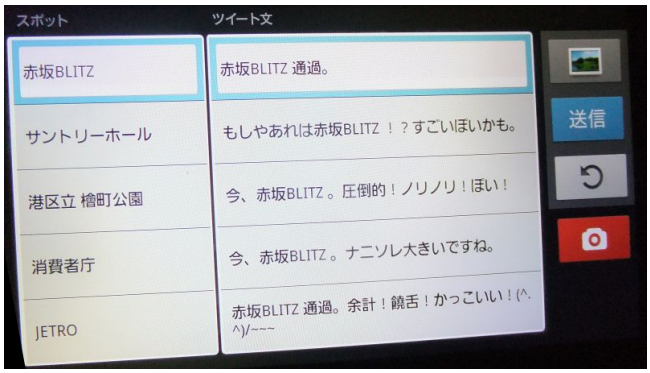

図 4 GhostTweet の画面: 東京都港区赤坂付近での使用例

修飾語の抽出は、個々のツイートを形態素解析し、形容詞などに相当する部分を取り出すことで行った。これを頻度付きのリストとしてデータベース化する。図 3 に挙げたのは場所名と、修飾語の例である。

4. GhostTweet

提案手法のコンセプト実現へ向け、第一段階となる試作として前節のアイディアに基づいた文の生成を行うシステム、GhostTweet を作成した。GhostTweet は発信専用の Twitter クライアントアプリケーションとサーバからなるシステムであり、クライアントは Android OS を搭載したスマートフォンやタブレット上で動作する。以下では、特に断らない場合は GhostTweet クライアントを単に GhostTweet と表記する。

GhostTweet を起動し、GPS 等で位置情報を捕捉すると、それをサーバに送る。サーバは位置情報をキーとしてデータベース[*6]から周辺の場所名とそれに関連した修飾語のリストを取り出す。場所名は Foursquare でのチェックイン数などで重み付けられており、近隣 (たとえば 1km 四方) の上位の 5 つの場所名の候補と、場所名に関連した修飾語をテンプレートに当てはめた候補文を各 5 つずつが端末に返される。複数あるテンプレートはランダムに選択され、修飾語は収集したツイート中で出現頻度の高いものほど高い確率で選択される。ランダム性があるため、同じ場所でも問い合わせ毎に違う候補文が得られる。

「近隣」の範囲の広さの設定は現在の実装では固定である。範囲を広げると、多少離れていても人気の高い場所名が選ばれやすくなるが、より近くにある知名度の低い場所名が選ばれにくくなる。範囲を狭めると、知名度の低い場所名も候補として挙げられやすいが、候補の数が 5 つに満たない、場合によっては 1 つも挙げられないという状況が起こりやすくなる。ユーザが自由に範囲の設定を変更できるようにするのはもちろん、移動速度に応じて自動的に変

[*6] 今回の試作データベースでは東京 23 区内を対応区域とした

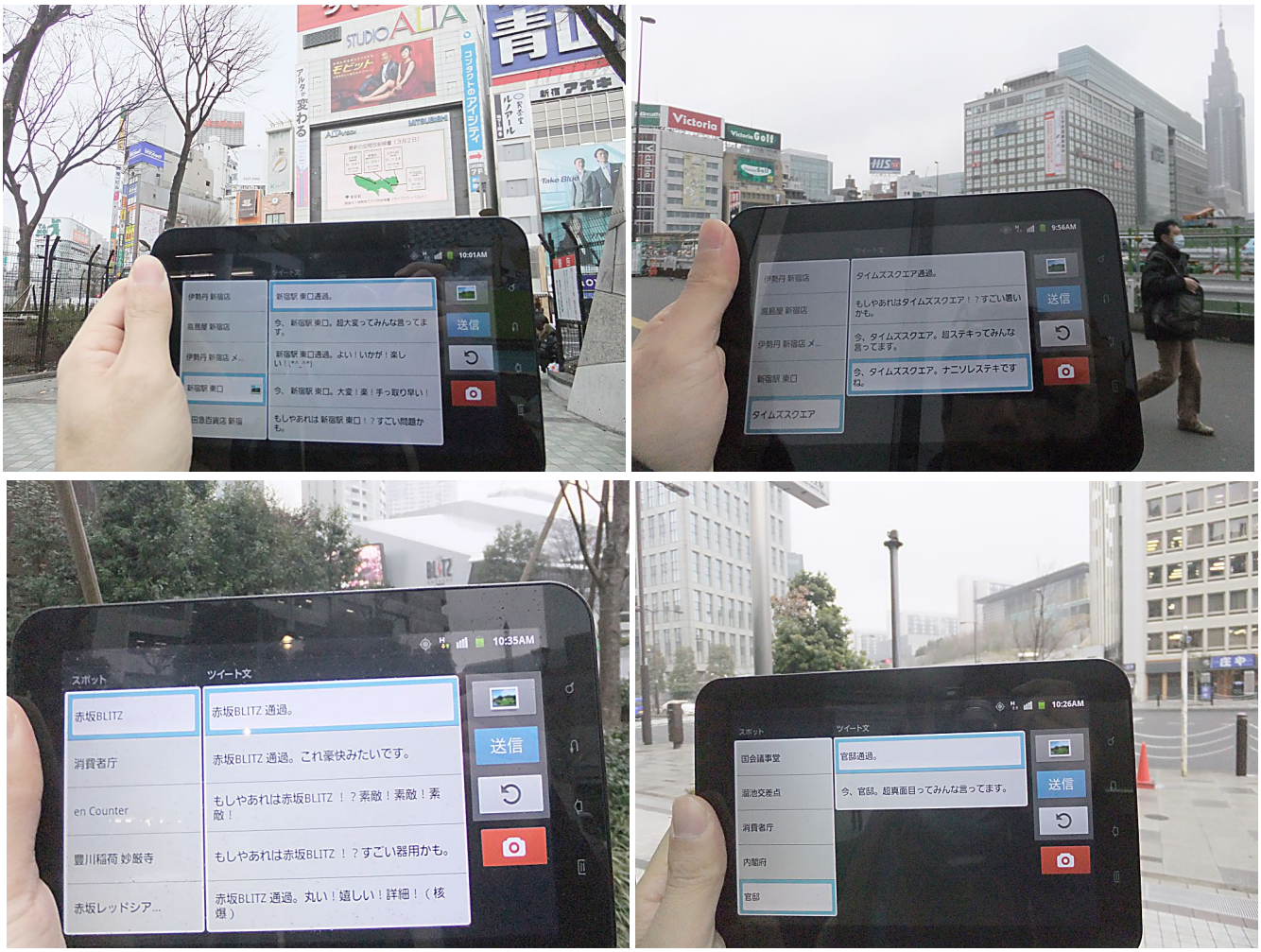

図 5 GhostTweet の試用例 (左上: 新宿駅東口付近 (東京都新宿区)、右上: 新宿駅南口付近 (東京都新宿区)、左下: 溜池交差点付近 (東京都港区)、右下: 赤坂付近 (東京都港区))

える (速く移動するほど範囲が広がる) などの方法で可変にすることも考えられる。

図 4 は GhostTweet の画面例である。左のカラムが場所名であり、場所名を選択すると (何も操作しなければ最上位候補が選択されている)、中央のカラムにその場所名に対応した候補文が表示される。5 つという候補の数はターゲットとしたタブレット型端末で表示したときに画面をスクロールさせずに表示できる量を意図している。

候補文を選択し (これも何も操作しなければ最上位候補が選択されている)、送信ボタンを押すと、選択されている文が Twitter のサーバへと発信される。場所名・候補文ともに最上位のものでよければ、ユーザは自動生成されたその内容を確認し、送信ボタンを押すという 1 ステップだけで発信が完了する。他の候補を選択する場合も、(場所の選択→) 文の選択→送信の 2〜3 ステップで送信ができる。適切な候補が見つからない場合は更新ボタン (送信ボタンの下のボタン) を押し、他の候補を表示させることもできる。

本稿では詳細を割愛するが、GhostTweet は写真に関連した機能も備えている。右下のボタンは写真撮影ボタンである。これを押すことで、端末のカメラで写真を撮影し「○○付近通過。」という文とともに発信することができる。右上のボタンを押すと Instagram から抽出した付近で撮られた写真のリストが表示され、そこから任意のものを選択し、引用してメッセージを発信することができる。

5. GhostTweet の試用

図 5 の写真はいくつかの場所で GhostTweet を実際に使用した様子である。いずれも背景として写っている景色中に、選択している場所名の建物が見えている。定量的な評価は行なっていないが、すぐに判る近隣の大きなスポットはほぼ常に候補として提示される。逆に、候補の中には、あまり広くは知られていないと思われる意外な場所名が含まれることもよく見られた。このような意外な場所名は、そのスポットや今いる場所の周囲への興味を掻き立てる効果があると思われる。

「現在位置の場所名」をピンポイントで使うのではなく「近隣の場所名」を使うため、例えば同じ電車の路線で駅毎に発信したとしても、人ごとに場所名の選択などでも違い

図 6 JR 山手線で移動しながら発信した例

が生じやすい、という特徴もある。図 6 は JR 山手線・外回りの電車で渋谷駅から品川駅にかけて移動しながら発信した例である[*7]。

テンプレートに埋め込まれる言葉は、場所に応じて適切なもの、意外なもの、意味不明なものなどが現れる。5 つの候補文の中に常に充分に適切と思われるものが現れるとは限らないが、試用した範囲では使用に耐えないほどではなかった。適切でかつ笑いを誘うようなものなど、とてもよい候補文が提示されることもある。前述のユーザごとの違いの生じやすさや、こうしたことが、ユーザがこのシステムを使うモチベーションに繋がると思われる。

まだ著者らの周囲の数名による試用しか行なっておらず、本格的な評価は今後の課題であるが、GhostTweet のようなシステムを用いて Twitter への発信することは現実的であるとの感触を得られた。本稿で提案するような方法で自動的に生成した候補文の利用が現実的であれば、それをベースとして変更してゆく作文手法も充分に実用性を持つであろう。

6. まとめと今後の発展

本稿では、Twitter に発信するメッセージのような短い文の入力をターゲットとした新たな作文手法、完成文変更方式を提案した。位置情報などに基づき自動的に生成された文をベースに変更を加えていくという方式で、常に発信可能な完成した文がある状態を保つことが大きな特徴である。この特徴により、その時点で掛けられる手数によらず発信文を生成することができ、手数に応じて発信文のバリエーションを増やすことができる。

その実装へ向けた第一段の試作として、Twitter 上のデータから個々の場所に関連した語を抽出したデータベースを作成し、それとテンプレートを用いて生成した発信候補文を提示し、ユーザは選択操作のみで Twitter への発信を行うことができるアプリケーション、GhostTweet を開発した。数人で試用してみたところ、候補文の中に必ずしも適切なものが含まれるとは限らないが、短い文によるコミュニケーションという目的のためには充分実用的と言える感触が得られた。今後は、候補文の一部の変更の機能も実装し、より多くのユーザに試用してもらい、コンセプト全体の検証を行う。

さらなる発展としては、ユーザによる候補文や語の選択をデータベースの改良に用いるということが考えられる。ユーザが選択した候補は、その場所においてより適切であるとユーザに明示的に評価されたものとみなせるので、候補の重み付けをするためのとても重要な指標となる。また、語の部分を手動で入力できるようにすることで、積極的にその場所やテンプレートに適した語を収集することもできるであろう。それにより、提示する候補文・語の精度も向上し、システムをより有益にすることが可能となる。このように、システムが利用されることにより文の解析や生成の精度が向上され、それがまたシステムの有用度を向上させ、より多くの利用に繋がるという循環が回るのが、本稿の提案がもたらしうる理想の状態である。

[*7] 「ゴジラ上陸地点」は映画「ゴジラ」(1954 年) でゴジラが襲撃した品川区八ツ山橋付近 (大崎駅と品川駅の間の線路のそばにあたる)

謝辞

試作に当たっていろいろと議論して頂き、またプログラムやデータベース、発信文のテンプレートなどを作成を担当して頂いた、水草亜友さんを始めとするチームラボ株式会社の皆様に深い感謝を捧げます。

参考文献

[1] Uchimoto, K., Isahara, H. and Sekine, S.: Text Generation from Keywords, *COLING '02 Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, pp. 1–7 (2002).
[2] 増井俊之：動的パタンマッチを用いた高速文章入力手法, インタラクティブシステムとソフトウェア V: 日本ソフトウェア科学会 WISS'97, pp. 81–86 (1997).