

Title	Webでの文章入力時における情報補完
Author(s)	中村, 和正
Citation	
Issue Date	2007-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/3589
Rights	
Description	Supervisor: 鳥澤 健太郎, 情報科学研究科, 修士

修 士 論 文

Webでの文章入力時における情報補完

北陸先端科学技術大学院大学
情報科学研究科情報処理学専攻

中村 和正

2007年3月

修士論文

Webでの文章入力時における情報補完

指導教官 鳥澤健太郎 助教授

審査委員主査 鳥澤健太郎 助教授

審査委員 東条敏 教授

審査委員 白井清昭 助教授

北陸先端科学技術大学院大学
情報科学研究科情報処理学専攻

410092 中村 和正

提出年月: 2007年2月

概要

本稿では、Blog 記事などの文章入力中に、その文章作成の参考となる情報を自動的に表示し、また必要に応じて本文に挿入することもできる文章作成支援システムについて述べる。

近年コンピュータを用いて文書を作成する際には、作成しようとする文章の参考となる関連情報を Web から収集する機会が非常に多い。例えば Blog で、その日に自分が関与した事物（訪れた場所、買った商品など）に関する意見や情報の記述を行う際にも、記述の正確性・詳細性を高めるため、訪れた場所のアクセスの方法、商品の仕様など事物の様々な側面の情報を確認したり付加したりするなど情報要求が頻繁に生じる。このような情報要求が生じると、ユーザは能動的に商用の全文検索エンジンを利用するなどして、書く作業を中断して参考情報を収集する必要がある。

本研究では、ユーザが文書を作成する際に頻繁に発生する情報要求に伴う「調べる」作業を極力抑えるために、1) ユーザの情報要求をこれまで知識獲得に使われていた語彙統語パターン [1] を利用して発見し、2) その情報要求タイプに応じた知識を Web からオンデマンドで獲得し、ユーザに提示することを目指すシステムを提案する。

ユーザの情報要求は、様々なタイプのものが存在すると思われるが、本研究では事物として具体物である固有名詞に焦点を当て、吉永と鳥澤による既存研究 [2] を用いて、具体物の様々な側面、すなわち属性とその属性値を Web から獲得し、それらを参考情報としてユーザに提示することで、ユーザへの文章作成の支援を行う。また、ユーザの情報要求の発見および、Web から獲得した参考情報の表示には AJAX を用いることで、画面遷移を伴うことなくシームレスに行い、極力ユーザの手間が発生しないようにした。

本システムの提示する参考情報が、ユーザにとってどれぐらいの精度で参考になるか評価実験を行ったところ、情報要求を検知できた内、50%以上の確率で何らかの情報を提示することができ、その内半数近くの情報が実際に参考となったことが確認できた。

目次

第1章	はじめに	1
1.1	研究の背景と目的	1
1.2	本論文の構成	2
第2章	関連研究	3
2.1	AJAXによるインターフェース	3
2.1.1	AJAXの非同期通信によるデータの更新	4
2.1.2	AJAXインターフェースによるインクリメンタルサーチ	6
2.1.3	既存の文章作成支援システム	8
第3章	提案システム	10
3.1	本システムにおけるAJAXについて	10
3.2	本システムの構成	16
3.2.1	本システムのクライアントとサーバー	16
3.2.2	本システムの処理の流れ	22
3.2.3	本システムの利用例	24
3.3	語彙頭語パターンに基づく情報要求の発見	24
3.4	Webからの属性・属性値の抽出	25
第4章	まとめ	27
4.1	評価実験	27
4.2	考察	28
第5章	おわりに	32
5.1	まとめ	32
5.2	今後の課題	32

第1章 はじめに

1.1 研究の背景と目的

本研究では，ユーザが文書を作成する際に，作成する文章に関連する情報を Web から自動的に収集し，提示することにより，文書の作成支援を行うシステムを提案する．近年コンピュータを用いて文書を作成する際には，作成しようとする文章の参考となる関連情報を Web から収集する機会が非常に多い．例えば Blog で，その日に自分が関与した事物に関する意見や情報の記述を行う際にも，記述の正確性・詳細性を高めるために，例えば以下のような情報を求めることが考えられる．

- 事物の正式名称や意味
- イベントの正確な日時
- 訪れた場所のアクセスの方法
- 商品の仕様
- 事物に関する第三者の意見

このように，文書作成中には事物の様々な側面（属性）の情報を確認したり付加したりするといった情報要求が頻繁に生じる．このような情報要求が生じると，ユーザは能動的に商用の全文検索エンジンを利用するなどして，書く作業を中断して参考情報を収集する必要がある．本研究では，ユーザが文書を作成する際に頻繁に発生する情報要求に伴う「調べる」作業を極力抑えるために，1) ユーザの情報要求をこれまで知識獲得に使われていた語彙統語パターン [1] を利用して発見し，2) その情報要求タイプに応じた知識を Web からオンデマンドで獲得し，ユーザに提示することを目指す．ユーザの情報要求の発見および，Web から得られた知識の提示は，AJAX¹を用いることでシームレスに行い，極力ユーザの手間を発生させないように工夫する．ユーザの情報要求は，実際には様々なタイプのものが存在すると思われるが，本研究では事物として具体物（例えばドイツやタイタニックなど）である固有名詞に焦点を当て，吉永と鳥澤による既存研究 [2] を用いて，具体物の様々な側面，すなわち属性とその属性値を獲得し，ユーザに提示する．

¹AJAX (Asynchronous JavaScript+ XML) とは JavaScript や XML などを用いたユーザーインタフェース構築技術の総称．

これにより、例えば「日本酒である太平山の原料米」に関して文章を作成する場合、ユーザがそれを知らなかったとしても本システムが「太平山の原料米は山田錦である」と情報を提示できれば、ユーザはそれを参考に Web から調べることなくスムーズに文章作成を続けることができるようになる。本システムではそうしたユーザの情報要求を検知し、画面内に参考となる情報を自動的に表示することにより、「調べる」ために中断することなく文章の作成の続行を可能にする文章入力支援を行う。

1.2 本論文の構成

以下本稿では、提案する本システムのクライアント側において、AJAX によるインターフェースが肝となっているため、第 2 章で AJAX の利用例および文章入力支援システムの先行研究について述べる。第 3 章では、本システムにおける AJAX の利用法、本システムの構成、ユーザからの情報要求の検知、情報抽出に関して述べる。第 4 章では、本システムの有効性に関する評価実験について述べ、第 5 章では今後の課題について述べる。

第2章 関連研究

本研究で開発するシステムはユーザの文章入力を支援するシステムである．本章では，本システムを開発する上で不可欠だった AJAX に関しての利用例と本システムとの関連について，また先行研究として開発された文章作成支援システムについて，その概要と本システムとの違いについて述べる．

2.1 AJAX によるインターフェース

近年，AJAX と呼ばれる既存技術の合わせ技により，ブラウザで表示される Web ページのインターフェースが大幅に向上してきている．

AJAX とは，スクリプト言語の JavaScript や Web 記述言語の XML といったオープンな技術を組み合わせた開発手法を指すのであって，AJAX という名称の技術があるわけではない．動的に Web ページの内容を変化させる DynamicHTML (JavaScript + CSS) と，表示内容を適宜サーバーから通信を介して取得する XMLHttpRequest オブジェクトという，従来からある標準的な技術を組み合わせて，より効果的な利用方法を提唱した手法である．

AJAX を用いると，Web ページ内の画像などのスクロール操作をマウスドラッグによって行ったり，ブラウザの画面遷移を発生させることなく動的に Web ページの内容を書き換えるといった，これまでにない Web ページでのユーザインターフェースが実現可能となる．

また，ユーザが行おうとする操作（例えばマウス操作や文字入力）を推測し，AJAX によって非同期通信を行い，システムが先に結果を提示するといった，これまでになかった新しい Web アプリケーションの開発も可能となった．

本システムにおいても，ユーザインターフェースの向上と，非同期通信を行う目的で AJAX を用いている．本節では AJAX によるインターフェースの構築を行った例について説明する．

2.1.1 AJAX の非同期通信によるデータの更新

本節では，AJAX によるインターフェースの存在を世に知らしめた代表的なアプリケーションとして Google マップについて述べる．

概要

従来の地図サイトでは地図データの更新のたびに画面遷移が発生した．例えば現在表示されている地図から，東方向の地図に切り替えたい場合，多くの地図サイトでは東方向への移動ボタンをクリックするか，画面上に表示されている地図上の東方向をクリックすることにより，該当する地図に更新し表示していた．その際，ブラウザによる画面遷移が発生し，地図データを更新するたびに Web ページ全画面を読み込みなおしていた．

2005 年に発表された，Google¹によるオンライン地図情報サービス Google マップは，ホイールで地図を拡大縮小したり，マウスドラッグによる操作で地図をスクロールさせるといった操作をブラウザ上で直感的に，そして画面遷移を伴うことなく新たな地図データに更新できるというインターフェースをもって提供された．これは，マウスドラッグの検知及び移動方向，移動距離を算出し，必要な地図データを逐次読み込むという AJAX による非同期通信により行われている．また，画面内に表示される必要な地図データ以外の，現在の中心点の座標周辺の画面外の地図データを先に読み込みこんでおくことで，マウスドラッグによる素早い座標移動（地図データの更新）時においても先読みにおけるブラウザのキャッシュ効果で，シームレスな地図データの表示が実現している．

また，地図に表示されている地域の店舗や飲食店などの情報がオンデマンドで表示され（図 2.1），さらには通常の地図表示のほか，衛星写真を用いての写真表示や，地図と写真を複合的に表示することもできる．これらも AJAX によるインターフェースの構築によって可能になった．

本研究との関連

Google マップで店舗や飲食店などの情報表示には，AJAX による動的な Web ページの書き換えによって実現されている．こうした AJAX によって任意の場所に任意のタイミングで画面遷移を伴うことなく動的に情報を書き換える手法は，あらゆるサイトで用いられてきている．その理由として，最小限のデータのやり取りで済むことと，画面全体が再読み込みされずに情報が更新されることが挙げられ，それらはインターフェース的にユーザにとって閲覧性，操作性において向上するからだと考えられる．

¹Google <http://www.google.com>

Google マップ BETA

ウェブ イメージ ニュース マップ グループ more

コンビニ 場所例:「渋谷」 お店やサービスを検索

地図を検索 お店やサービスを検索

マップ

印刷 メール このページのリンク

Y 白山市明法島町35
076-274-4560
カテゴリ: コンビニエンス店

E ローソン 鶴来小柳町店 - 詳細
白山市小柳町る166-1
0761-93-4181
カテゴリ: コンビニエンス店

F ファミリーマート 松任山島台店 - 詳細
白山市山島台3丁目82
076-274-1053
カテゴリ: コンビニエンス店

G ローソン 松任福留店 - 詳細
白山市福留町2084
076-277-3079
カテゴリ: コンビニエンス店

H サークルK 辰口宮竹店 - 詳細
能美市宮竹町ヨ88-1
0761-51-7408
カテゴリ: コンビニエンス店

I サックス 鶴来バイパス店 - 詳細
白山市安養寺町口13-4
0761-93-2929
カテゴリ: コンビニエンス店

J サークルK 鶴来店 - 詳細
白山市明法島町西104-2
0761-93-5044
カテゴリ: コンビニエンス店

©2007 Google 地図データ ©2007 株式会社

図 2.1: Google マップによる周辺情報の表示

2.1.2 AJAX インターフェースによるインクリメンタルサーチ

本節では、検索エンジンの入力フォームに AJAX によるインターフェースを施し、インクリメンタルサーチ²を実現させた例として Google による Google サジェスト³ について述べる。

概要

従来の検索エンジンを利用する際は、検索キーとなる単語をクエリとして入力フォームに入力し、検索ボタンをクリックすることで検索エンジンに対して、検索キーに関する情報の出力を求めていた。検索エンジンを使い慣れているユーザであれば、こういった検索キー、絞り込み⁴を入力すれば、適切な情報を含んだ Web ページが出力されるかある程度想像できる。しかし時として、情報不足により曖昧な検索キーしか入力できない場合や、的外れな検索キーを入力している場合、特に検索エンジンに不慣れなユーザは、出力を望む Web ページに対してそのように適切な検索キーの入力や、絞り込みができない場合が多い。

これから述べる Google サジェストでは、その名の通り適切な検索キー及び絞り込みとなるキーをユーザに対して提案する検索エンジン（インクリメンタルサーチ）である。ユーザは入力フォームに 1 文字でも単語を入力した時点で、Google サジェストはユーザの入力中の単語の続きを推測し、他の利用者から検索の際によく用いられる（つまり出力を望む Web ページに対して、適切な検索キーである可能性が高い）検索キーの候補の一覧を、その入力フォーム周辺に提示する（図 2.2）。ユーザは、その候補から検索キーを選ぶことでそのまま検索結果を得ることができる。

さらに続けて文字を入力していくと、それに伴って検索キー候補も絞られていき、より具体的な検索キーが提示される。そのため、ユーザが曖昧にしか検索キーを入力できなかったり、適切な検索キーを入力することが難しかった場合でも、Google サジェストのシステムにより提示される検索キー候補を参考にし、適切な検索キーを用いて検索を行うことができるようになる。また、提示される検索キー候補一覧には、その検索キーで検索した場合の検索結果件数があらかじめ表示されているため、ユーザは実際に検索キーを入力（選択）する前に、その検索キーの有効性を推測することができる。

本研究との関連

Google サジェストで用いられている AJAX によるインターフェースは、本研究において画面遷移なく最小限の操作で結果を表示するという点で参考となった。また、非同期通信によってユーザからの入力単語を推測するというアプローチも、本システムの情報要求

²データを検索する方式のひとつで、1 文字入力することに検索し、徐々に絞り込んでいく手法

³Google サジェスト <http://www.google.co.jp/webhp?complete=1&hl=ja>

⁴複数の検索キー入力による、AND 検索など



[ウェブ](#) [イメージ](#) [ニュース](#) [マップ](#) [グループ](#) [more »](#)

AJA	
ajax	170,000,000 結果
ajaxとは	39,000 結果
ajax サンプル	2,280,000 結果
ajax ime	54,900,000 結果
ajax 入門	2,450,000 結果
ajax ライブラリ	1,350,000 結果
ajax php	73,900,000 結果
ajax フレームワーク	1,160,000 結果
ajax.request	157,000 結果
ajax.xml	73,100,000 結果

[検索オプション](#)
[表示設定](#)
[言語ツール](#)

つぶりの

詳細については

[Googleをホームページに!](#)

©2007 Google

図 2.2: Google サジェストによるインクリメンタルサーチ

の検知に用いている。このように、画面遷移を発生させることなくユーザからの入力に対して動的に結果を画面内に反映できるインターフェースは、操作性の面の向上において重要なことであると考えられる。

2.1.3 既存の文章作成支援システム

本節では、先行の文章作成の支援を行うシステムとして小林ら [7] が開発した、Web ベースでのドキュメント作成ツール”Web-Doctor”について、その概要と本システムとの違いについて述べる。

概要

小林らが開発した Web ベースでのドキュメント作成ツール”Web-Doctor”は、作成中の文書に引用できそうな情報をブラウザで提示し、それを簡単に文書内に挿入（引用）できるようにすることで支援を行うシステムである。一連の手順は以下の通りである、

1. ブラウザ上に表示されている文書内から、参考となる情報に関連した具体物名をマウスドラッグで反転させ、右クリックメニューから外部の検索エンジンへその単語をクエリとして送信する
2. 外部検索エンジンによる検索結果（Web ページ）がブラウザの別のウィンドウで開かれ、表示される
3. ブラウザのツールバーから貼り付けを行うことで、提示された検索結果が入力中の文書に挿入される

以上の手続きで参考情報の文書への挿入を行うことができる、文書作成支援システムとなっている。

本研究との比較

小林らが開発したシステム”Web-Doctor”は、Firefox⁵ というブラウザの拡張機能用プログラミング言語により、Firefox 専用に作られたソフトウェアである。その為、小林らのシステムを利用するには Firefox を利用することが前提で、Firefox に”Web-Doctor”が組み込まれている必要がある。しかし本研究で開発するシステムは、ブラウザに別途ソフトウェアを組み込む必要がなく、またブラウザも選ばないため、汎用性において違いがあると思われる。また、本システムは新たにブラウザのウィンドウが開かれることなく、文章入力画面内にシームレスに情報が提示されるため、操作性、ユーザビリティ的にも本システムは向上していると考えられる。

しかし、小林らのシステムでは、Web ページ全体または任意の一部を、画像を含めて挿入することができるとしている。いわばリッチテキスト⁶形式での編集環境であり、本シ

⁵Firefox <http://www.mozilla-japan.org/products/firefox/>

⁶Microsoft が策定した文書ファイル形式によって、文字の大きさや色、書式などの情報が文書中に盛り込まれたもののことで、文書内に図形や表を埋め込むことが可能となっている

システムのようなプレーンテキスト⁷形式での編集環境ではない，といった違いもある．また小林らのシステムでは挿入される文章は引用元の著作権保護のため編集ができないようになっており，プレーンテキスト形式として挿入する本システムでは，挿入された文章をも編集することが可能になっている．

また，参考情報の探し方にも違いがある．小林らのシステムではユーザの情報要求が発生した場合，ユーザはクエリとなる語句を指定し，そのクエリに関する Web ページをシステムが探し出し，そこからユーザが必要な情報を選択して挿入するという手続きになるが，本システムではユーザの情報要求もシステムが検知し，Web ページから参考情報を探し出し，必要と思われる情報のみを抽出し提示する．ユーザはその提示された情報が必要であれば参考にしたり，文章内に挿入するという操作だけで済むようになっている．

小林らのシステムと本システムでは以上のように支援方法に対するアプローチの違いがあり，本システムでは，極力ユーザの操作は必要最低限にとどめ，必要な情報が提示されれば参考にしてもらう，といった利用方法を想定して開発した．

⁷レイアウト情報や修飾情報を持たない、純粋に文字のみで構成されるテキストデータ

第3章 提案システム

3.1 本システムにおける AJAX について

本研究では、ユーザ（クライアント）からの様々な情報要求にリアルタイムで応じるために、重い処理を行う情報抽出のモジュールは Web 文書を直に持っているサーバー側で行う。本節では、このようなクライアント・サーバー形式の Web アプリケーションにおいて、ユーザーインターフェースを向上させるために近年よく用いられている AJAX について説明する。

AJAX とは、JavaScript の HTTP 通信のためのオブジェクトである XMLHttpRequest を用いて、ブラウザの HTTP による画面遷移とは非同期にサーバーとデータをやり取りし、動的にページ内容を変更する仕組みを指す。ある特定の技術を指す名称ではなく、JavaScript を中心とした、いくつかの技術を複合的に用いて構築するという手法についての総称である。この AJAX という手法を用いることにより、これまでの動的な更新を可能にしてきた既存技術（Flash¹など）を用いることなく、動的な Web ページ（またはその内容）の更新が実現できるようになった。既存技術との違いとして、AJAX の場合、ブラウザに実装されている標準の機能（JavaScript や CSS, XML など）だけで動作するという点がある。既存技術の場合、利用するブラウザごとに専用のプラグインと呼ばれるプログラムの導入を必要とすることが多い。しかし AJAX はほとんどのブラウザに標準的に実装されている機能を使うため、クライアントに対して別途プログラムの導入を要求することはない。

第2章の関連研究でも述べたように、AJAX をインターフェースに用いた例として以下を挙げる。

- Google マップ
地図上でマウス操作により、ドラッグで地図をスクロールしたり、ホイールで地図を拡大縮小できる。必要な地図データは逐次シームレスに表示する。
- Google サジェスト
ユーザが入力しようとしている検索キーを、1文字入力されるごとに推測し、最も適切であると思われる語句を検索キー候補としてユーザに提示する。

¹Macromedia 社（現 Adobe 社）の開発により、音声やアニメーションを組み合わせたりリッチコンテンツをブラウザ上で実現した。マウスやキーボードの入力により双方向性を持たせることもできる。

- Gmail²
メールの閲覧や新規送信画面など画面の切り替え時において、全画面を読み込み直すことなく、必要な箇所だけ情報を更新する。
- Google Docs&Spreadsheet³
AJAX を多用したブラウザ上で動作するワープロソフトと表計算ソフト。Microsoft 社の Microsoft Word, Microsoft Excel と機能性, 操作性, データ形式に互換性がある。ドラッグ&ドロップによる操作, 右クリックメニュー, 文章置換, フォント装飾, 画像挿入, 表計算, 関数など, 従来のアプリケーションと見劣りしない Web アプリケーションとなっている。

こうした非同期通信を用いた部分的な Web ページの動的な更新は, AJAX のもたらすインターフェース, つまり斬新な操作性や視覚的效果だけでなく, Web2.0⁴と呼ばれる新しい Web サービスの流行により AJAX が取り込まれている場面も多く, 近年増えつつある手法である。しかし, こうした AJAX による手法にはいくつかデメリットがある。その代表的なものを次に挙げる。

²Gmail <http://mail.google.com>

Google が提供する, ブラウザ上でメールの送受信を行うことができる Web ベースのメールクライアント

³Google Docs&Spreadsheet <http://docs.google.com/>

Web ブラウザ上で動作する Microsoft Word と Excel 互換ソフト

⁴コンテンツの提供方法や技術の提供方法, サービスの利用方法など, 次世代的な手法を用いている Web サイトを漠然と指しているため, 明確な定義はない。そして, AJAX は数年前から存在している枯渇した技術の複合技であるが, その他の技術と複合することによって斬新な効果が得られたことから, Web2.0 の象徴的な技術となっている。

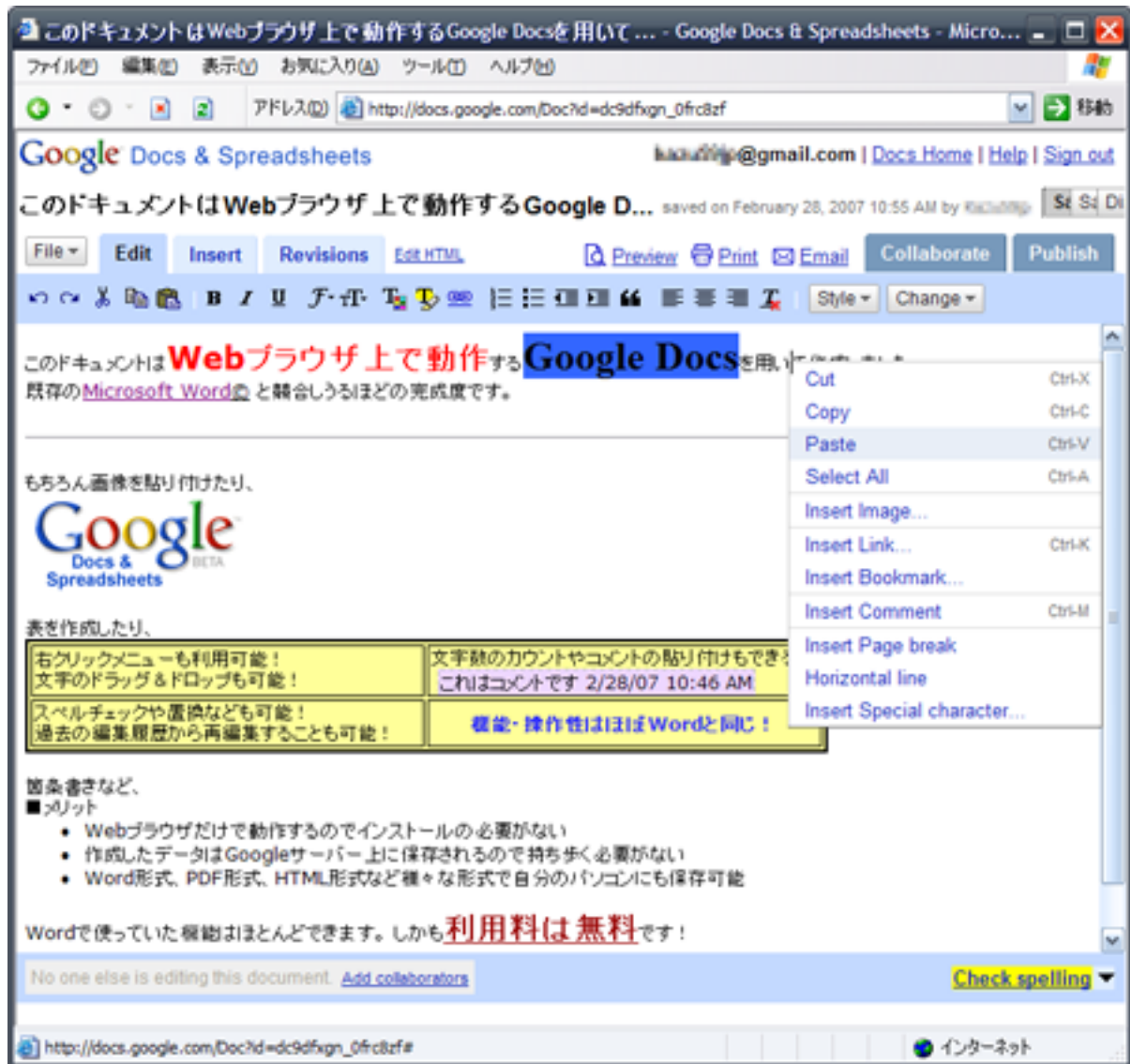


図 3.1: GoogleDocs の動作画面

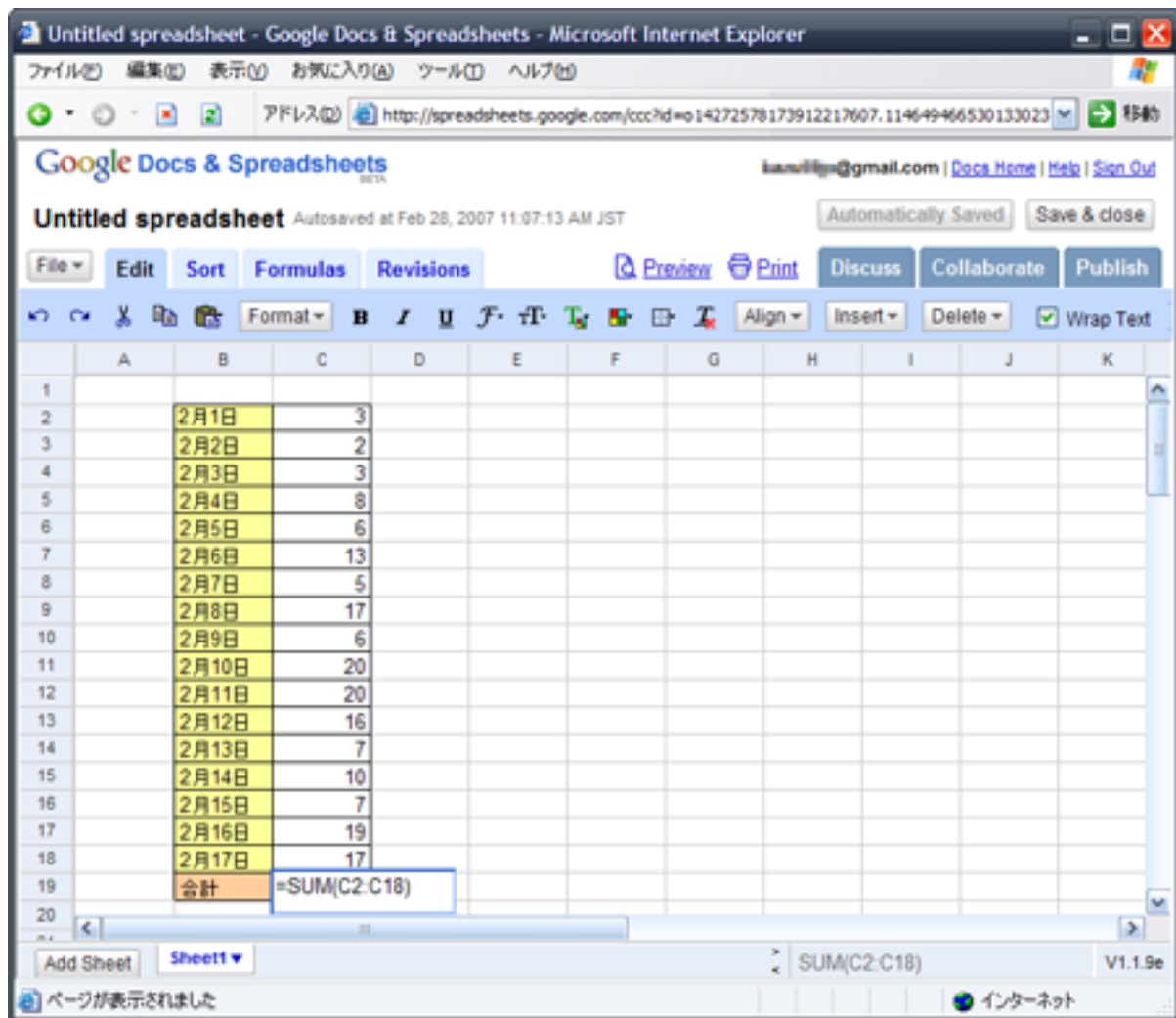


図 3.2: GoogleSpreadsheet の動作画面

1. 情報が更新されても URL が変わらない

AJAX による Web ページの動的な更新は、設計方法によっては従来のように「この情報はこの URL に割り当てられている」(図 3.3) というように、情報と URL が一意に結びつかない(図 3.4)。つまり、特定の情報をダイレクトに呼び出すための URL がユーザには明示的に提示されないということである。そのような設計で開発された AJAX の場合において、動的に表示された情報を再び呼び出すには、最初に呼び出した同じ手順を繰り返すか、クライアント側またはサーバー側に、その呼び出したという履歴情報を保存し、何らかの手段でその履歴情報を参照して呼び出せるようにするか、または情報を更新するたびに一意の URL を出力する、といった工夫が必要になる。またそれに伴い、ブラウザの「戻る」ボタンなどで前画面に戻った後、再び「進む」ボタンで再表示したとしても、そうした工夫がされていなければ、AJAX による動的な情報の更新がされる前の初期画面状態になってしまう。

そしてブラウザの仕様上、全画面を読み込むのではなく部分的に更新するということは、ユーザには一見、AJAX によって情報のやり取りがなされている最中なのかどうかわかりにくい。全画面を読み込む場合はブラウザが何かしらの方法で(例えばアイコンなどのアニメーション)通信中であることを示すが、AJAX を用いての部分的な更新においては、ブラウザは読み込んでいることを検知しない。そうした場合に、何らかの理由により AJAX によるサーバーとの通信が止まってしまった際に、再度通信を行わせるために従来のようにブラウザの「更新」ボタンで画面全体を再読み込みしてしまうと、前述のように初期画面状態に戻ってしまいかねない。

Web ページを動的に部分的に更新することによって、「情報」と「URL」が一意に定まらない設計でシステムを開発してしまうと、こうした弊害が出てしまい、ユーザビリティの向上どころか、逆にシステムの使い勝手を悪くしてしまう可能性がある。

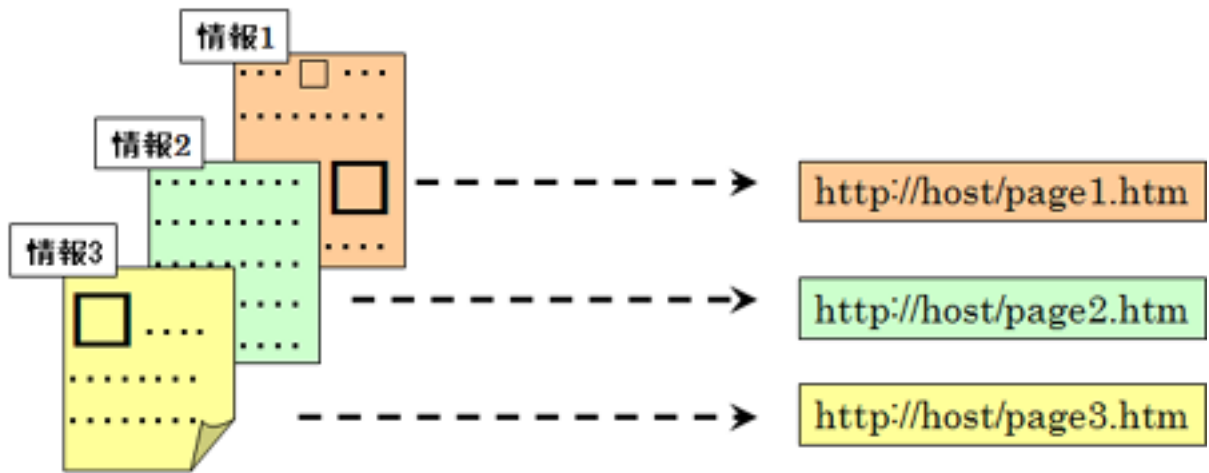


図 3.3: 従来の Web ページと URL の関係

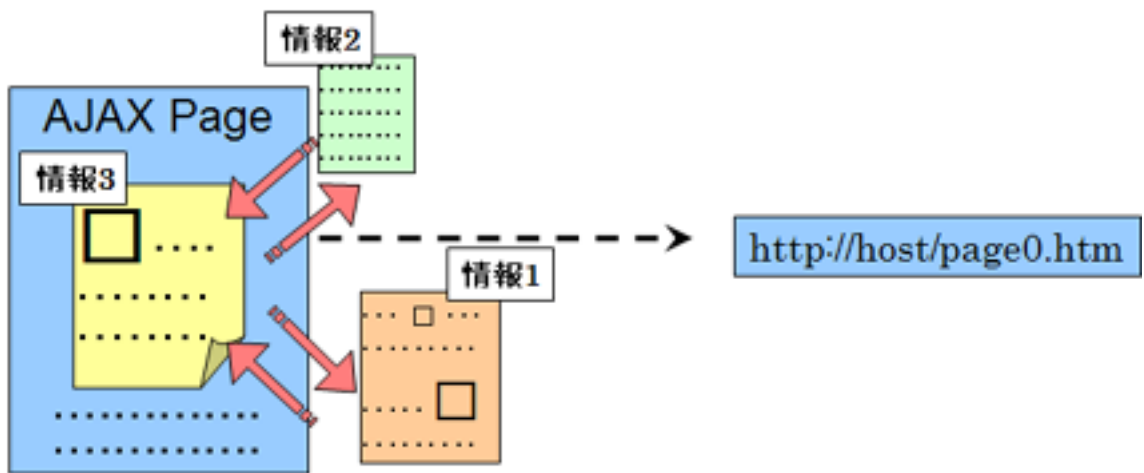


図 3.4: AJAX で情報を更新しても URL は変わらない

2. 開発に関する問題

AJAX による非同期通信には前述の通り HTTP 通信のためのオブジェクトである XMLHttpRequest を用いているが、この XMLHttpRequest がブラウザごとによって仕様が違うため、AJAX を用いたシステムを作成したとしてもブラウザによっては挙動が変わったり、正常に通信できなかつたりする。また、XMLHttpRequest だけでなく、インターフェースの構築に用いるための DynamicHTML (JavaScript + CSS) にもブラウザごとに互換性の問題があるため、ブラウザによってはインターフェースが狂ったり見栄えが変わったりする。こうしたブラウザ間の互換性問題は古くからあり、現在に至っても数多く残っている。できるだけ多くのブラウザで期待する動作させる必要があるならば、様々なブラウザで (さらにはそれぞれのバージョンにおいても) 動作検証する必要性が出てくる。

また、AJAX (JavaScript) によるアプリケーションの開発はソースコードが誰でも閲覧可能になってしまうという問題もある。JavaScript はクライアント側で動作するスクリプト言語であるため、斬新な AJAX による Web アプリケーション、Web サービスを開発したとしても、ソースコードを閲覧されてどのように実装しているのか見破られ⁵、類似したアプリケーションやサービスが簡単に作られてしまう可能性があり、差別化が困難になることが考えられる。

さらには、AJAX によって通信を要求されるサーバー側の負荷も考えなければならない。従来の全画面を読み込むという手法に比べ、部分的に必要な情報だけ読み込むことができる AJAX では、無駄なデータのやり取りが減り、通信量は減少すると思われるが、従来に比べてサーバーの負荷が高くなる可能性が高い。AJAX を用いたシステムは、ユーザの操作に敏感に反応し、頻繁に情報を更新しようとするほど、必然的にサーバーとの通信回数が増えるからである。

3.2 本システムの構成

本節では本システムの構成と、処理の一連の流れについて述べる。

3.2.1 本システムのクライアントとサーバー

本研究で開発する文章作成支援システムを図 3.5 に示す。本システムはインターネットに接続されたクライアントと、同じくインターネットに接続されているサーバーで構成されている。以下、クライアントとサーバーについて述べる。

⁵Google などはソースコードを頻繁に変更したり、インデントや改行を省くなどの難読化を行い、解読の対策を施しているが、それでも限界がある

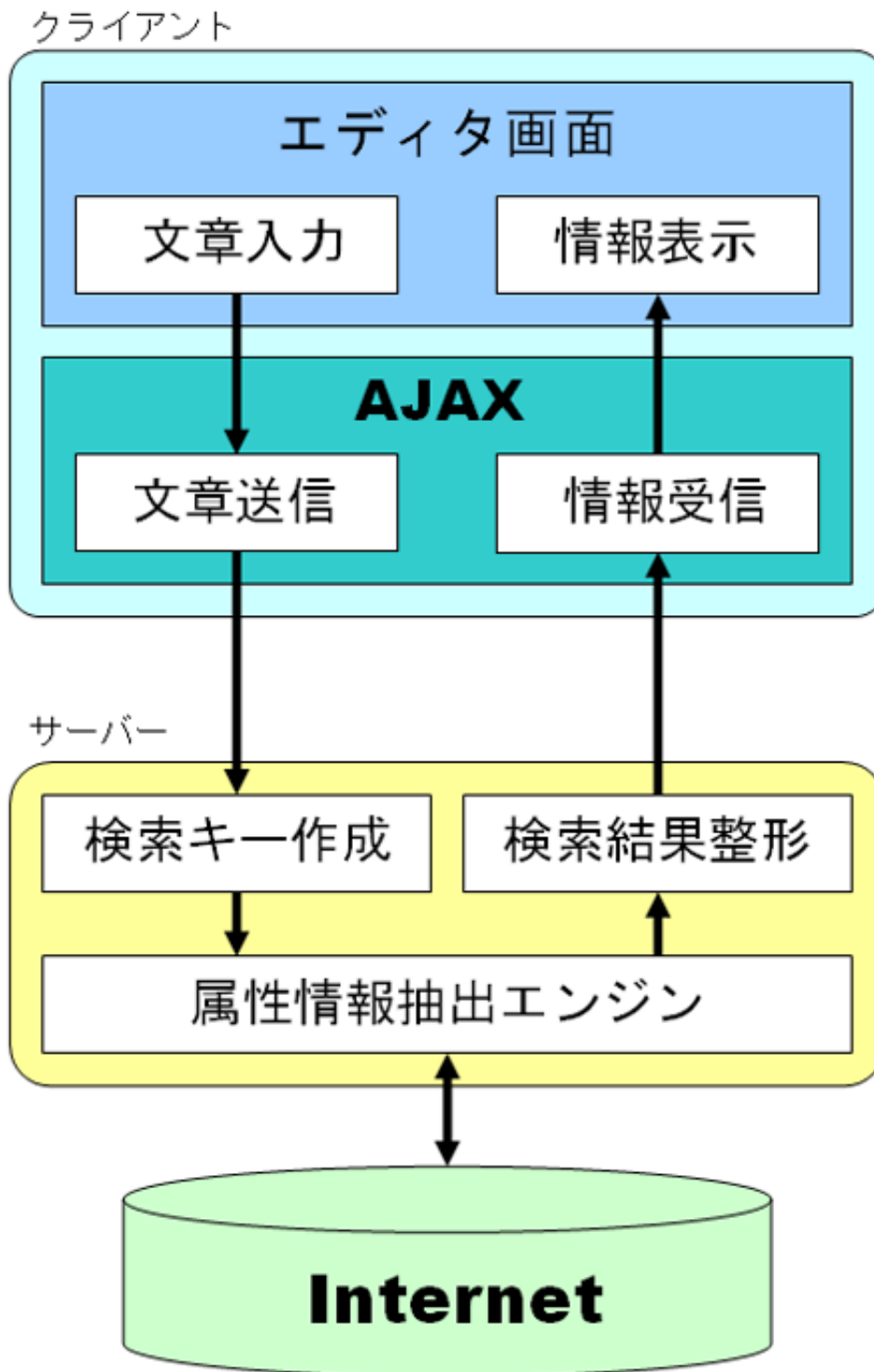


図 3.5: 本システムのクライアントとサーバーの構成

・クライアント

本システムのクライアント部に含まれるエディタ画面を図 3.6 に示す。本システムはブラウザ上で動作する Web アプリケーションであり、クライアントにはブラウザ以外のソフトウェアの導入は必要ない。クライアント部に表示されるエディタ画面（Web ページ）自体に、バックグラウンドでサーバーと通信するための AJAX コード⁶が埋め込まれているからである。

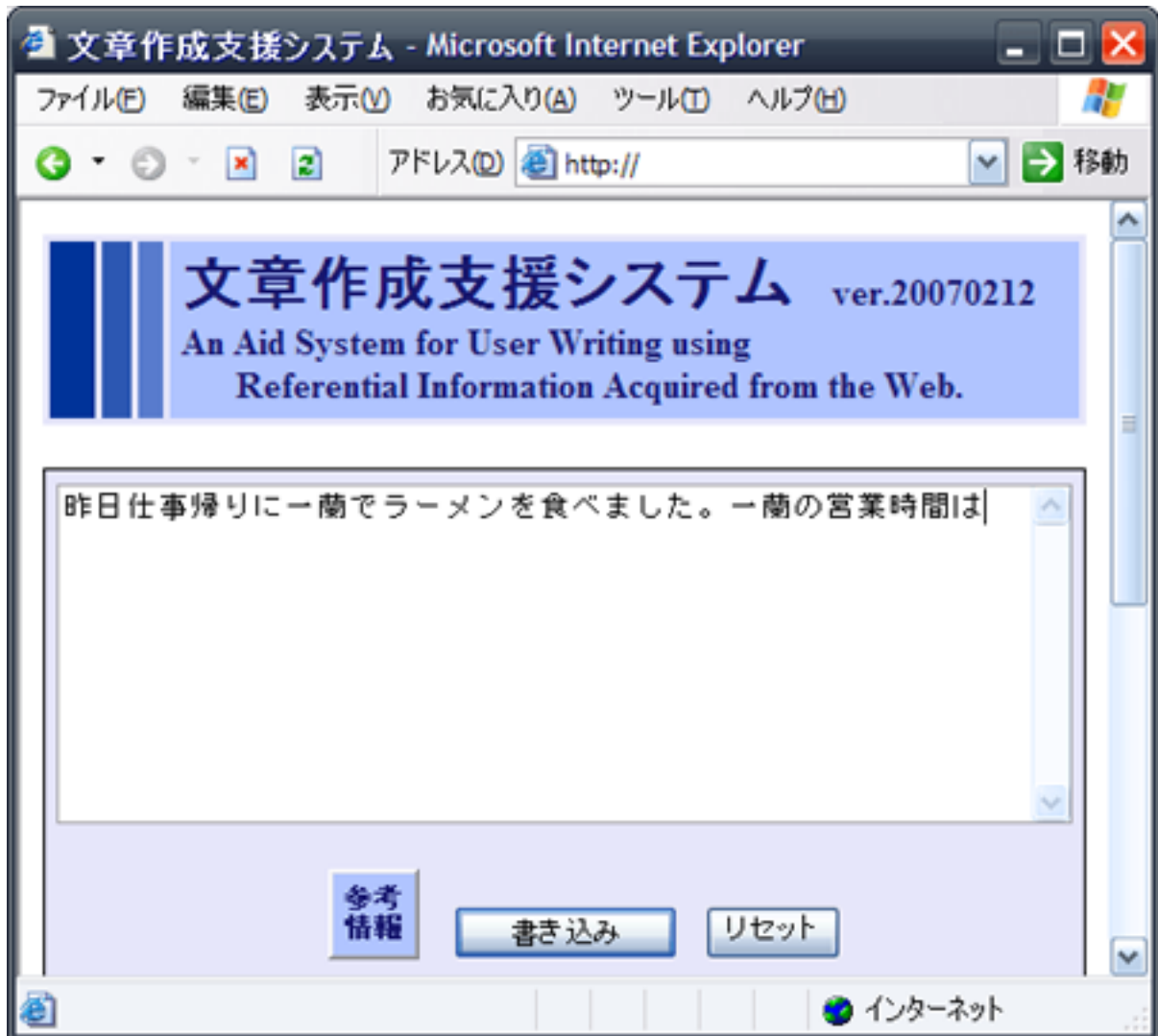


図 3.6: 本システムで文章入力を行うエディタ画面

⁶厳密には JavaScript のコード

クライアントは一定時間ごとにサーバーに入力中の文章を送り続けることにより、サーバーはほぼリアルタイムにユーザの入力する文章を受け取ることができる。しかしユーザが文章を書き終えるまでの間、またユーザの文章入力が止まっている間も延々と送信し続けるとサーバーに負担がかかってしまう。そのため、本システムでは入力された文章に変化があったかどうか、つまりユーザが文章を書き換えたかどうかをクライアント側で検知できた場合のみ、サーバーに送信するようにした。これにより、重複した文章を送信することもなくなり、またユーザが文章入力を中断したとしても、その間サーバーに送信されることはなく、サーバー側に余計な負荷はかからない。

また、ユーザの任意のタイミングでサーバーに入力中の文章を送信できるようにもした。具体的には **Tab** キー押下または **参考情報** ボタンをクリックすることで、それまでに入力した文章を送信する。一定時間ごとにサーバーに入力中の文章を送り続ける場合、文章の入力速度によってはサーバー処理時間やネットワークでの遅延などが原因で、参考情報の提示がスムーズに行われなかったといったことが起こる可能性がある。そこで、ユーザが参考情報を要求したいときのみ入力中の文章を送信することで、不必要にサーバ処理や通信が発生することなく、スムーズに参考情報を得ることができるようになると考えている。

エディタ画面である図 3.7 の左側の入力フォームに情報要求が含まれる文章を入力していくと、それに関連した参考情報を本システムが自動的に収集し、ユーザに提示する。ユーザはそれを参考に文章入力を行い、必要であれば **本文に挿入** ボタンをクリックすることで、その情報が入力中の文章に挿入される。



図 3.7: 参考情報が提示された状態のエディタ画面

・サーバー

本システムのサーバーには、クライアントからデータを受け取り、情報要求を検知し、検索キーとなる単語を抽出し、関連情報をインターネットから収集するためのプログラムが実装されている。クライアントからバックグラウンドで送信された入力中の文章をサーバーが受け取ると、サーバーはMeCab⁷によって形態素解析を行い(図3.8)、特定の語彙統語パターンを用いて、情報要求が含まれているかどうかチェックする。その際、文末の品詞から文頭へと逆順にチェックすることで、ユーザの最新の情報要求を検知する。

情報要求が検知できた場合、サーバーは情報抽出エンジンの入力としてクエリの作成を行い、情報抽出エンジンへクエリを送信し、その出力として関連情報を得る。情報要求の検知とクエリの作成、情報抽出エンジンについては次節以降で詳しく述べる。

検索キー作成(MeCabを利用)					
送信された文字列	今日は学校の近くのラーメン屋真打に行きました。真打の住所は、				
形態素解析	表層形	読み	品詞	品詞細分類1	品詞細分類2
	今日	キョウ	名詞	副詞可能	*
	は	ハ	助詞	係助詞	*
	学校	ガッコウ	名詞	一般	*
	の	ノ	助詞	連体化	*
	近く	チカク	名詞	副詞可能	*
	の	ノ	助詞	連体化	*
	ラーメン	ラーメン	名詞	一般	*
	屋	ヤ	名詞	接尾	一般
	真打	シンウチ	名詞	一般	*
	に	ニ	助詞	格助詞	一般
	行き	イキ	動詞	自立	*
	まし	マシ	助動詞	*	*
	た	タ	助動詞	*	*
	。	。	記号	句点	*
	真打	シンウチ	名詞	一般	*
	の	ノ	助詞	連体化	*
	住所	ジュウショ	名詞	一般	*
は	ハ	助詞	係助詞	*	
、	、	記号	読点	*	

図 3.8: MeCab による形態素解析を文末から読み込む

⁷MeCab (和布蕪) <http://mecab.sourceforge.net>

3.2.2 本システムの処理の流れ

本システムを利用する際の手順は図 3.9 のようになる。

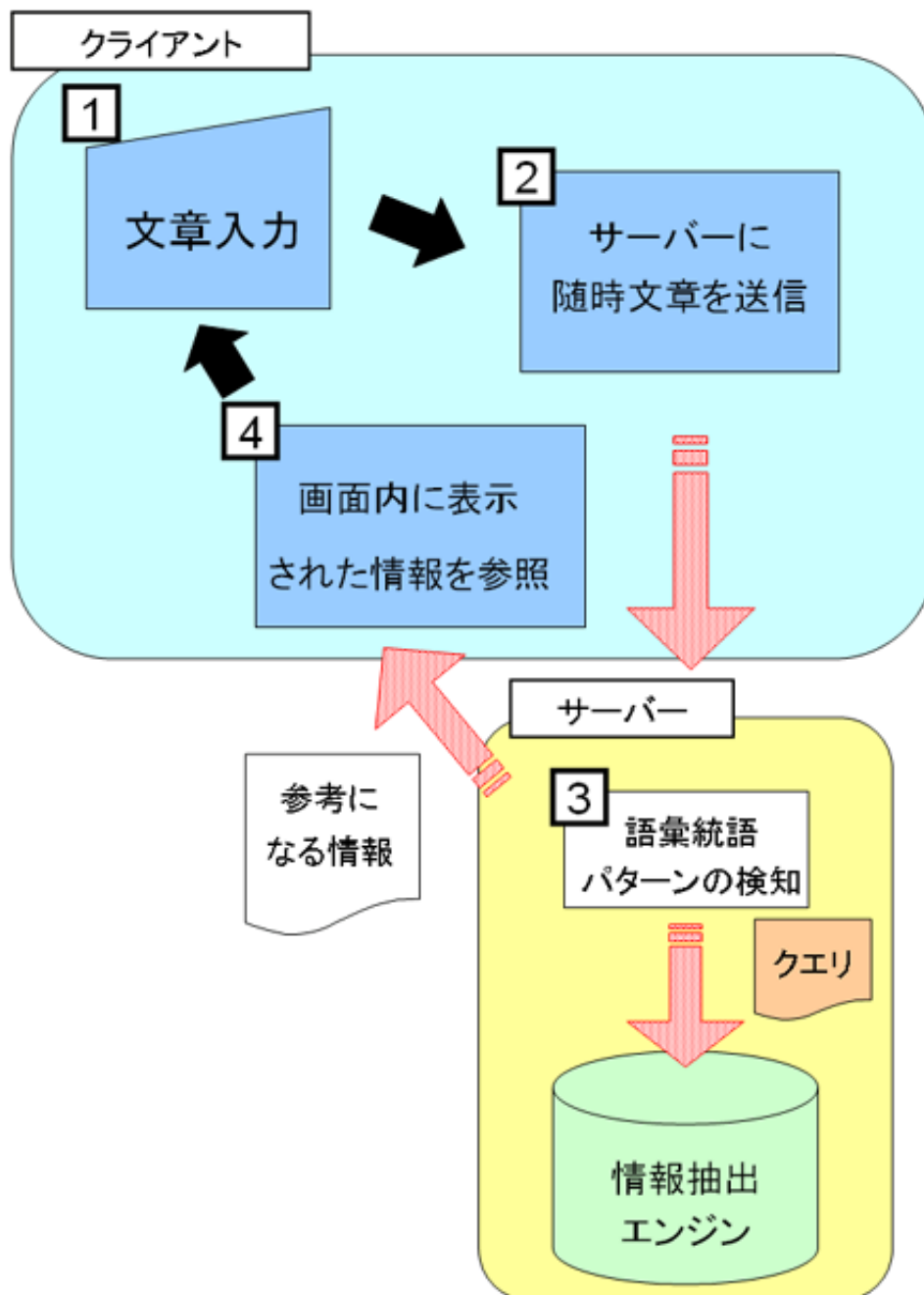


図 3.9: 本システムの処理の流れ

本システムはユーザからの入力を監視し、一定間隔で入力中の文章をサーバーに送り続ける。例えば「金沢にあるラーメン屋、真打の住所は」と文章を入力（図 3.9-①）したタイミングで、本システムが自動的に AJAX を用いてバックグラウンドで入力中の文章をサーバーへ送信（図 3.9-②）したとする。文章を受け取ったサーバーは、そこから形態素解析を行い、語彙統語パターンからユーザの情報要求の検知（図 3.9-③）を試みる。検知できた場合、情報抽出エンジンへの問い合わせを行うためのクエリを作成し、情報抽出エンジンに入力して、得られた結果をエディタ画面内に参考情報として表示（図 3.9-④）する。ユーザは必要に応じて表示された情報を参考にしたり、本文に挿入（図 3.10）したりすることができる。

以上の一連の流れのように、ユーザは本システムを利用することにより、従来のように商用の検索エンジンなどを用いて、Web から参考となる情報を収集するために文章の入力を中断することなく、必要と思われる情報をシステムから自動的に提示されることにより、ユーザはそれを参考にしながら文章入力を続行できるということになる。

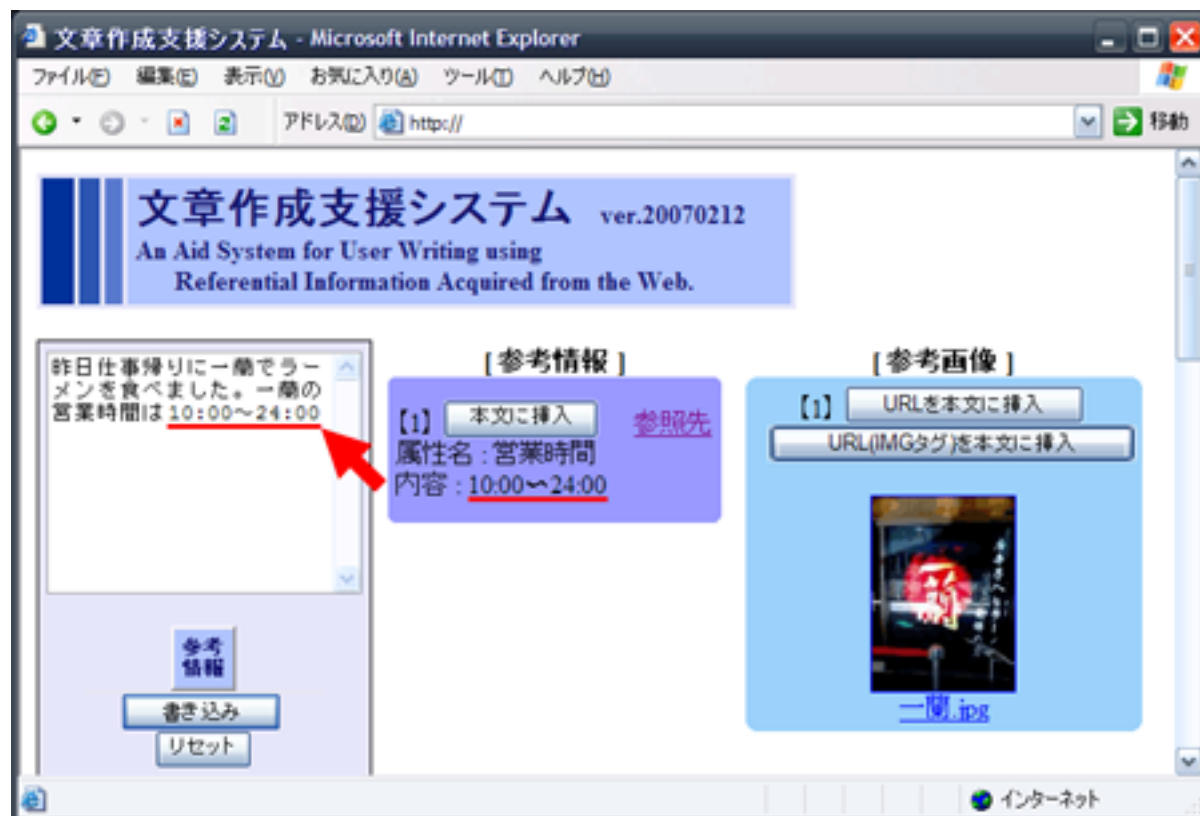


図 3.10: 本システムに提示された情報を入力中の文章に挿入

3.2.3 本システムの利用例

例えば Blog を書いているユーザが「今日は学校の近くのラーメン屋真打に行きました。真打の住所は、」という文章を入力しているとする。ここでユーザが真打の住所を知らなかったとしても、本システムが「真打の住所」を情報要求として検知できれば、真打の住所に関する情報を Web から発見し、「石川県金沢市大額 3 丁目 184」という情報を画面内に提示することで、ユーザの文章作成を支援できる (図 3.10)。

また、前述したように本システムでは文章作成と情報抽出をクライアントとサーバーに分離させているため、ユーザはサーバーに入力中の文章が送信されたことを確認後、サーバーから参考情報が提示されるまでの間待ち続ける必要は無い。ユーザはサーバーの処理の完了を待たずとも文章の入力を続行することができ、サーバーの処理の完了後、つまり参考情報が提示されてから必要であれば文章を書き直す、といった使い方ができるため、文章入力 of 効率が一層高まることが期待できる。

3.3 語彙頭語パターンに基づく情報要求の発見

文書入力中のユーザの情報要求を正確に推定することは、入力途中の未完成の文からユーザの意図を推測しなければいけないため、それ自体非常に難しい問題である。本研究では、情報要求のタイプを、対象とその属性に関する情報に限定し、従来自然文からの知識獲得に用いられてきた語彙統語パターン [1] を用いてユーザの情報要求の予測を行う。

語彙統語パターンは、知識獲得を行う際に獲得対象の知識 (上位・下位関係, 属性・属性値関係等) を含む語彙, 統語的な文脈を表現したパターンである。例えば上位・下位関係の獲得であれば, "A と呼ばれる B" といったパターン [3] が属性・属性値の獲得であれば, "A の B は C である" のようなパターン [5] が用いられる。これらの語彙統語パターンは, 高い割合で目標とする知識を表現するため, 知識獲得に用いられているのであるが, 逆を返すと, このような統語パターンにマッチする表現をユーザが用いたときには, ユーザは高い確率で対応する知識に関する情報を記述することが予想できる。例えば "王貞治と呼ばれる" という入力が行われれば, ユーザは次に「野球選手」「監督」「ホームランバッター」といった「王貞治」の上位語を書くことを意図していると推測できるし, "ホテルオークラの宿泊料金は" という入力を与えられれば, ユーザは「ホテルオークラの宿泊料金」を記述しようとしているということが予測できる。

そこで本研究では, ユーザの情報要求の中でも特に頻度が高いと思われる, 具体物の属性・属性値情報に焦点を当て, 属性記述に関する語彙統語パターン "A の B" を利用して, ユーザの情報要求を予測することを試みる。具体的には作成中の文章の末尾を形態素解析し, その結果に "A (名詞句) の B (名詞句)" という語彙統語パターンを検知した場合に先行する名詞句 A を具体物名, A から "の" を介して係り受けの関係にある名詞句 B を A の属性語として, 吉永と鳥澤の属性・属性値抽出システム [2] へのクエリを生成する。

3.4 Webからの属性・属性値の抽出

本研究では，ユーザの入力文から語彙統語パターンを用いて検知された具体物とその属性に関して，情報要求を表現するクエリを生成し，吉永と鳥澤 [2] により提案された属性・属性値抽出システムの入力として与えることで具体物の属性・属性値を獲得する．

仮にユーザの情報要求を検索クエリとして適切に表現できたとしても，その検索クエリに情報抽出システムが高速に回答することが出来なければ，本システムがリアルタイムの動作を意図している以上，実用的なシステムとして運用することは難しい．今回，本システムに採用した吉永と鳥澤による属性・属性値抽出システムは，具体物の属性語を利用することで属性・属性値を含む可能性の高いWeb ページを効率よく収集し，得られた Web ページ中で入力属性語を含む範囲を特定した後，入力属性語の出現している文脈から属性・属性値の記述パターンを導出することで，属性・属性値を非常に高速に獲得することが可能となっている．

彼らが想定している入力は，具体物とその上位語であるが，上位語は実際には具体物の属性語を得ることに利用されており（本システムが検索クエリとして与える）具体物とその属性語を入力としても，彼らのシステムを運用することは可能である．

彼らは論文で 50 個の具体物とその上位語のペアを入力として，全文検索エンジンを用いて 10 件の Web ページを知識源として収集した場合，正しい属性関係が得られたページ（表 3.1 中で AllCorr または PartCorr）が少なくとも一つあった具体物は 37（74%）であったと報告しており，また 10 件程度の Web ページを知識源として属性知識の獲得を試みた場合，経験的にかかる時間は 10 秒程度であり，本システムの期待する高速性を十分達成できていると結論づけられる．

表 3.1: 属性関係の獲得実験結果

上位語	total	AllCorr	PartCorr	AttrIncorr	InsIncorr	NA
宿	41	13	4	10	0	14
株式会社	25	3	3	3	6	10
本	52	16	3	16	1	16
歌	34	6	2	8	0	18
酒	49	16	9	4	1	19
平均ページ数	40.2	10.8	4.2	8.2	1.6	15.4
		26.9%	10.4%	20.4%	3.9%	38.3%

NA : 属性関係が一切獲得できない

InsIncorr : 異なる具体物の属性関係を獲得

AttrIncorr : 不適切な属性語が獲得

AllCorr : 適切な属性関係のみ獲得

PartCorr : 一部の属性関係で属性値が冗長

第4章 まとめ

本章では，本システムの有用性を検証するための評価実験とその考察に関して述べる．

4.1 評価実験

本システムの有効性を確かめるため，数名に本システムを利用してもらい，実際に参考となる情報が提示されたか確認してもらった．実際に文章を作成し，本システムを利用してもらおうという状況下での精度を測るため，被験者には本システムの詳しい仕組みや情報要求に関する検知方法などは伝えていない．具体的には，以下のような条件で実験を行った．

入力する文章

被験者はいずれも Blog 記事経験者であるため，Blog など記事の作成状況を想定してもらい，“語句 の 語句”というパターンを含んだ文章を入力してもらった．

判定基準

“語句 の 語句”を入力した際に本システムから提示される情報に関して，それが実際に参考となるような情報であったか判定を行ってもらった．同じ入力，同じ参考情報の提示があった場合でも参考となるかどうかは，作成しようとしている文書によりその時々によってユーザの情報要求が変わるため，その都度主観によって，以下のように判定してもらった．

- a 本システムから有用な参考情報が提示された
- b 本システムから有用ではないが，何らかの参考情報が提示された
- c 本システムから何も参考情報が提示されなかった

その結果，本システムがユーザの情報要求を検知した内，27%が参考となる有用な参考情報が提示された (a)．また，有用ではないが何らかの参考情報が提示されたものが 28%であった (b)．情報要求を検知したが何も参考情報を提示できなかったものは 45%であった (c)．

4.2 考察

実験結果を見る限り，本システムがユーザの情報要求を検知した場合において，50%以上の確率で何らかの参考情報を提示でき，その内半数近くの情報が実際に参考になった，という事がわかった．判定基準の (b) ”本システムから有用ではないが，何らかの参考情報が提示された” 場合において，参考となる有用な情報と，不要な情報または全く関係ない情報が提示されていた場合も (b) に含めた．しかし ”少なくとも 1 つは参考となる情報が含まれている” という場合も (a) の ”本システムから有用な参考情報が提示された” に含まれると考えるならば，実際に参考情報を提示できたとする確率はさらに高まることが考えられる．また，吉永と鳥澤の論文 [2] にあるように，属性・属性値抽出システム自体の今後の精度向上も見込めることから，十分に文書作成の支援ができるシステムになるものと期待できる．

表 4.1 に，本システムを利用して作成された文章を示す．このように，本システムから提示された参考情報を用いることで，本来調べなければ書けなかったような，詳細な数値や住所などの情報を追記することができ，より具体的な文章の作成が実現できている．

表 4.2 に，実際に提示できた例を示す．適切な具体物名と属性名が与えられれば，時事的な情報である商品の価格や株価，ホテルなどの住所，映画の監督名や芸能人の出身地など，多岐に渡る情報の提示（属性値の抽出）ができることが確認できた．

表 4.3 に，参考情報を提示できなかった例を示す．今回の実験では被験者に情報要求に関しての検知アルゴリズムを伝えていないため，本システムが検知できる語彙統語パターン ”A (名詞句) の B (名詞句)” 以外の入力もいくつか含まれている．仮にそれらを検知するパターンの追加ができれば，さらなる精度の向上が考えられる．

表 4.1: 本システムを利用して作成された文章

SMAP のメンバー、木村拓哉の出身は東京だったことは知りませんでした。
あのパソコン用ディスプレイで有名な EIZO の本社は 924-8566 石川県白山市下柏野町 153 でした。
先週旅行に行ったドイツの首都ベルリンでは本当に楽しかったです。
東京ディズニーランドの住所が千葉県浦安市舞浜 1 となっていることからわかるように、実際には千葉県にあります。
映画、タイタニックの監督はジェームズ・キャメロンで、過去にターミネーターやエイリアン 2 など手がけています。
ターミネーターの主演はアーノルド・シュワルツェネッガーで、最新作のターミネーター 3 も主演することが決まった。
私の携帯電話、SH702iD の重量は約 89 g と、とても軽いです。
先月訪れた USJ の住所は〒 559-0034 大阪市住之江区南港北 1-13-11 です。
情報処理技術者試験の日程は年 2 回、4 月と 10 月とのこと。
道後温泉の効能は神経痛、リウマチ・胃腸病・皮膚病・痛風・貧血などがあるそうです。
博多のお土産に博多通りもん (6 個入り) を買ってきました。
スーパーファミコンの発売日は 1990 年 11 月 21 日だったようで、もうあれから 17 年近く経っている。

表 4.2: 本システムにより参考情報を提示できた例

プレイステーション 3 の価格	62,790 円 (HDD 20GB)、オープンプライス (HDD 60GB)
任天堂の本社	601-8501 京都府京都市南区上鳥羽鉾立町 11-1
東京大学の住所	東京都文京区本郷 7-3-1
松井秀喜の打率	305
アメリカの GDP	10 兆 7280 億ドル (2004 年:名目)
名探偵コナンの作者	青山剛昌
タイタニックの監督	ジェームズ・キャメロン
Google の時価総額	150,841 百万
エキスポランドの住所	大阪府吹田市千里万博公園 1-1
革命の作曲	ショパン
ドイツの首都	ベルリン 北緯 52 度 30 分 東経 13 度 22 分
ドラゴンボールの作者	鳥山明
スーパーファミコンの発売日	1990 年 11 月 21 日
木村拓哉の出身地	東京都
EIZO の本社	924-8566 石川県白山市下柏野町 153
城南区のガスト	福岡県福岡市城南区片江 3 丁目 12-36
東京ディズニーランドの住所	千葉県浦安市舞浜 1
ユニクロの営業時間	年中無休/AM9:00 ~ PM9:00
硫黄島からの手紙の出演者	渡辺謙 二宮和也 伊原剛志 加瀬亮 中村獅童
祐徳稲荷神社の住所	佐賀県鹿島市古枝乙 1855
福岡市の一蘭	福岡県福岡市中央区天神 1-10-15
ミッションインポッシブル 3 の俳優	トム・クルーズ (TOM CRUISE)
福岡市の美術館	福岡市中央区大濠公園 1-6 092-714-6051
博多のお土産	博多通りもん (6 個入り)
イチローの背番号	51

表 4.3: 本システムにより参考情報を提示できなかった例

木村拓也のドラマ
華麗なる一族の出演者
野間のたい焼き
東京ディズニーランドの営業時間
緑のポッキー
ポッキーの種類
明治製菓のお菓子
鉛筆削りの性能
洋服のサイズ
マグロの成長
中洲川端商店街のぜんざい
チャンネルシティーのコムサ
唐津バーガーのメニュー
ひつまぶしの本店
エビフィレオの値段
飛騨のさるぼぼ
ロッセリアのキャンペーン
博多のお笑い
梅田のヨドバシ
博多とおりもんの材料
ミッキーマウスの誕生日
節分の由来
ITO の会社
神座のラーメン
トマトの料理
ダチョウの卵料理のレシピ
人気のカクテル
滋賀県の湖
日本一の山
牧のうどんの本店
イチゴのあまおう
マクドナルドの営業時間
城南区のピザポケット
ナフコの電話番号
グラタンのレシピ

第5章 おわりに

5.1 まとめ

本稿では，入力中の文章に関連した情報を AJAX と既存の情報抽出エンジンを用いて提示することで，ユーザへの文章作成を支援する手法の1つとして，本システムを提案した．また，本システムがユーザの情報要求に対して，どれぐらいの確率で有用な参考情報を提示できているか評価実験を行った．

実験結果より，ユーザからの情報要求を検知した場合において，50%以上の確率で有用な参考情報を提示することができた．また，今後の情報要求の検知のために用いる語彙統語パターンの追加と，現在の属性・属性値抽出システム以外の情報抽出エンジンの併用を行うことで，さらなる精度の向上が期待できると予測され，本研究の目的である，“文章作成の中断となるような「調べる」作業を発生させることなく，スムーズに文章を作成する”ことが一層果たせるのではないかと考えている．

5.2 今後の課題

本システムを利用した結果及び評価実験の結果から，今後の課題として以下に改善すべき点を挙げておく．

情報要求を検知する語彙頭語パターンの実装が現在，1パターンしか無い

現在は，“A（名詞句）のB（名詞句）”というパターンのみである．このパターンを増やし，情報要求の検知精度を向上させ，多岐に渡る情報の提示を行いたい．

本システムが対応する情報抽出システムを増やす

現在，本システムは吉永と鳥澤ら [2] の属性・属性値抽出システムと Yahoo，goo にしか対応していない．しかし例えば，その他の商用の検索エンジンなどを情報抽出エンジンとして併用することで，より多くの複数の情報を提示できると考えている．

また、検索エンジンに限らず、既存の Web サービスとのマッシュアップ¹によって様々な情報の提示ができる可能性がある。例えばユーザからの情報要求として地域名を検知すると、その場所を Google マップで表示したり、書籍名を検知すると Amazon²の商品ページを提示する、といった情報提示も有効ではないかと考えている。

提示される情報の信頼性を確認する手段が乏しい

提示される参考情報が一見有用な情報だったとしても、その真偽の確認が必要な場合がある。首都や過去の映画監督、著者名など不変である情報または、ユーザが曖昧にも記憶していた情報が提示された場合、その参考情報の真偽は判断しやすい。しかし、例えば株価や住所、電話番号といった、記述された時期によって変化する可能性のある時事的な情報は、記述された日時によってユーザの情報要求に対して適切な参考情報かどうかの評価が変わる。そうした時事的な参考情報が提示されたとしても、“いつ取得した情報なのか”、“それは最新の情報なのか”といった、参考情報の信頼性に対する疑問がでてくる。

提示される参考情報の信頼性を高めるためにも、例えば“参考情報の取得日時などを明記する”、さらには“獲得する Web ページ内の記述から情報の記述日を抽出し、ユーザに提示する”といったこともできれば、参考情報に対する信頼性も高まるのではないかと考える。

また、“そもそも、その情報は本当に情報要求に対しての情報なのか”といった根本的な疑問がユーザに出てくる可能性もある。例えば通天閣のアクセスに関する参考情報が提示された場合(表 5.1)、それが本当に通天閣のアクセスに関する情報なのかどうか、提示された情報だけでは確認が難しい。ユーザが大まかにでも通天閣の住所などを知っていれば提示された情報が正しいかどうか判断できる可能性もあるが、まったく予備知識がない場合判断ができない可能性がある。ユーザが本システムから提示された参考情報に対して、その情報の真偽を確かめる判断材料の提示ができなければ、結果的にユーザは、他の検索エンジンなどを用いて情報を収集することでその提示された参考情報の信頼性を確認することが予想され、本システムの利用に意味が無くなってしまいかねない。


ユーザに負担をかけず、“抽出した参考情報の周辺の情報、例えば前後の文脈なども提示する”といった、信頼性を確認するための判断材料の提示(図 5.1)も、本システムのユーザビリティ的な面からみても必要であると考えている。

¹複数の異なる提供元の技術やコンテンツを複合させて新しいサービスを構築すること。複数の API を組み合わせて形成された、あたかもひとつの Web サービスであるかのような機能。AJAX の流行によりマッシュアップにより構築された Web サービスも目立つようになってきた。


²Amazon <http://www.amazon.co.jp>

表 5.1: 通天閣のアクセスに関する参考情報

通天閣のアクセス	JR 環状線新今宮駅から徒歩 10 分、 地下鉄堺筋線恵美須町駅から徒歩 3 分
----------	---



通天閣



所在地	大阪府大阪市浪速区恵美須東1-18-6
営業時間	10:00~18:00(閉場18:30)
休業日	無休
入場料	小人300円~大人600円
駐車場	有・観光バスのみ3台(無料)
アクセス	<u>JR環状線新今宮駅から徒歩10分、地下鉄堺筋線恵美須町駅から徒歩3分</u>
問合せ先	通天閣観光株式会社
TEL	06-6641-9555

昭和31年(1956年)10月に再建された2代目です。高さ103m、円形エレベーターを備えた近代的な姿は、浪速っ子の心に再び灯をともしました。2代目通天閣は大阪のシンボルとして、今日も元気にそびえています。

図 5.1: 参考情報の情報元を提示することで信頼性を高める

謝辞

本研究を進めるにあたり，日頃からの研究方針及びその手法，それに関連する先行研究などに関して，懇切丁寧なご指導を賜りました鳥澤健太郎准教授に厚く御礼申し上げます．

本研究へのご理解と適切な指摘，方向性の助言など多大なご協力を賜りました東条敏教授に厚く御礼申し上げます．

属性・属性値抽出システム他，研究全般にわたってご指導，ご協力頂きました吉永直樹博士に深く感謝致します．

また，公私にわたりお世話になりました知識工学講座の皆様方に感謝を申し上げます．

参考文献

- [1] Marti A. Hearst, Automatic acquisition of hyponyms from large text corpora. In Proc. of COLING, pp.539 - 545, 1992.
- [2] 吉永直樹, 鳥澤健太郎, Web からの具体物の属性・属性値情報の自動獲得, 言語処理学会第 13 回年次大会発表論文集, 2007.
- [3] 安藤まや, 関根聡, 石崎俊. 定型表現を利用した新聞記事からの下位概念単語の自動抽出. 情報処理学会研究報告, 2003-NL-157, pp.77-82, 2003.
- [4] 今角恭祐. 並列名詞句と同格表現に着目した上位下位関係の自動獲得. 九州工業大学修士論文, 2001.
- [5] 高橋哲朗, 乾健太郎, 松本裕治. 言語パターンと統計的共起尺度による属性関係抽出. 言語処理学会第 11 回年次大会発表論文集, pp.432-435. 2005.
- [6] 徳永耕介, 風間淳一, 鳥澤健太郎. 属性語の Web 文書からの自動発見と人手評価のための基準. 自然言語処理, 13(4), 2006.
- [7] 小林俊, 佐々木稔, 米倉達広. Web ベースでのドキュメント作成ツール”Web-Doctor”の提案. 情報処理学会インタラクション 2006 (査読付き口頭発表) 講演論文集 インタラクティブ発表, 2006.