blog記事の内容を反映したキャッチコピー自動生成システム “きゃっちふれいざ” の試作

Prototyping of “Catch-phraser” :
An automatic catch phrase generating system that reflects contents of blog articles

松田幸子 † 犬塚敦史 †† 林貴宏 ††† 尾内理紀夫 †† †††
Sachiko MATSUDA† Atsushi INUZUKA†† Takahiro HAYASHI††† Rikio ONAI†† †††

† 電気通信大学電気通信学部 情報工学科卒，現在 (株)NTT データシステムサービス
Department of Computer Science, The University of Electro-Communications,
NTT Data System Service Corporation (now)

†† 電気通信大学大学院電気通信学研究科 情報工学専攻
Department of Computer Science, Graduate School of Electro-Communications,
The University of Electro-Communications

††† 電気通信大学電気通信学部 情報工学科
Department of Computer Science, The University of Electro-Communications

概要

検索エンジンの検索結果から，ユーザはタイトルや要約文を頼りに閲覧するページを選択する．しかし要約文は長く読みにくい．そこで，ページ内の文章からインパクトのあるキャッチコピー（週刊誌の広告における見出しのような表現）を生成し，要約文の代わりにユーザに提示することにより，ユーザの閲覧を喚起する検索システム “きゃっちふれいざ” を試作した．“きゃっちふれいざ” により，検索の出力結果は，単なる要約文からキャッチコピーとなり，blog 記事の検索・選択を新たな尺度で行うことや，キャッチコピーそのものを閲覧して楽しむことが可能になった．

1 はじめに

現在，一般的な情報検索の手段としてテキストによる全文検索が広く用いられている．例えば，Google[1] などに代表される Web 検索などである．これらの検索エンジンの検索結果として，ページのタイトル・ページの冒頭や検索キーワード周辺などを抜粋した要約文・ページ作成日時などがテキスト情報として出力される．ユーザはタイトルや要約文を頼りに閲覧するページを選択するが，要約文は長くなりがちで検索結果全ての要約文を読むことはユーザにとって負担となる．

そこで，ページの内容を表す要約文よりも短い文を提示することで，ユーザがページを選択する際のユーザの負担を軽減できるのではないかと考えた．

この負担軽減に向け，本研究ではページ内の文章からインパクトのあるキャッチコピー（週刊誌の広告における見出しのような表現）を生成し，要約文の代わりにユーザに提示することにより，ユーザの閲覧を喚起する検索システム “きゃっちふれいざ” を構築した．なお，対象とする web ページはグルメ系ジャンルを標榜する blog とした．これは，グルメ系

[1] http://www.google.com/

blog においては「さっぱりしているのにコクがある」「一度は食べておきたい一品」等の定型的表現が頻出し，ページ間の差異が少なく差別化が難しいため，キャッチコピー化によりその差異を際だたせる効果があると考えたためである．また，タイトルに食べた物や店名に関する情報を明記する傾向が強く，キャッチコピー生成時に有効に活用できるからである．

以下，2 章では要約文とキャッチコピーについて，3 章ではシステム概要について，4 章では blog 記事取得モジュールについて，5 章では代表文選択モジュールについて，6 章ではキャッチコピー生成モジュールについて，7 章ではキャッチコピー出力モジュールについて述べ，8 章で考察を行う．

2　要約文とキャッチコピー

2.1　要約文とは

要約とは「あるひとまとまりのテキストが表している意味内容を，非常に短いテキストで簡潔に表現すること（あるいは，そのような表現自体）」と定義される [1]．人間がテキストの要約を行う際は，対象となったテキストを読んで意味内容を理解し，それを再構成し直して文章を生成するという「理解–再構成–文章生成」の手法をとる．要約文を生成することで，対象となる原テキストの意味内容をより短時間で効率的に理解可能になる．しかし，人手によるテキストの要約作業は多大な労力を要する．そのため，機械によるテキストの自動要約に対する期待は高く，関連する研究報告も数多くなされている．

機械によるテキストの要約では，基本的に対象となったテキストの不要な部分を削除し，重要な箇所だけを残すことによって要約文と見なす手法をとる．よって，機械によるテキスト要約は原テキストの重要箇所同定と近似できる．

テキストの重要箇所を同定する方法としては，次の二つが有力である．

第一は，テキスト中の重要語を抽出してこれを基に各文の重要度を計算し，重要度の高い文のみを抽出して要約を生成する方法である．重要度の計算には，キーワードの出現回数，前文との接続関係等が多く用いられる．砂山ら [2] は，出現頻度の高い単語を手掛かりとして重要語を抽出し，重要度を決定してテキスト要約を行っている．

第二は，対象テキストの構造を利用して重要文を決定し，要約を作成する方法である．論文の要約を例に考える．論文は一般に，序論・本論・結論という構造から成っている．論文で最重要となる部分は研究成果を示す「結論」であると考えると，結論部分の冒頭文章から数文を抜き出すことにより，要約を生成することが可能になる．

重要文抽出によるテキストの自動要約では，テキスト原文をそのまま用いるために，文と文の前後関係が崩れたり，先行詞の存在しない代名詞が出現したりして要約文が読みにくくなることも多い．難波ら [3] は，このような読みづらさの問題に対して実際に心理実験を行い，読みにくさの原因が「表現（または接続詞）の過不足による文間のつながりの悪さ」「構文的な複雑さ」「冗長・不自然な繰り返し」「情報の欠落」「副助詞の過不足による文間のつながりの悪さ」の 5 種類であるとし，問題となる点を書き換えることによって読みにくさの解消に取り組んでいる．

2.2　キャッチコピーとは

キャッチコピー（catch + copy）とは，辞書的に「消費者の心を強くとらえる効果をねらった印象的な宣伝文句」と定義されるが [4]，本研究においては「人の注意を惹くように工夫が凝らされた，簡潔な一文（または複数文）」と定義する．主に企業の運営方針を示す時や，消費者向けの宣伝広告に用いられることが多い．

人間がキャッチコピーの作成を行う際は，対象物に関する情報と既存の知識とを融合させ，極力簡潔かつ印象深い表現を駆使して文を作成する．

2.3　要約文とキャッチコピーの違い

要約文とキャッチコピーの主な違いとして対象物に関する情報の網羅率が挙げられる．原テキスト（対象物）から人間が読みとれる情報量を 100%と仮定する．要約文は原テキストから再構成あるいは抽出

された文章である．従って要約文が与える原テキストの情報量は要約文の長さに比例し，おおよそ 60〜90%程度の範囲に収まると考えられる．一方キャッチコピーは，それを見た人の興味をいかに引きつけるかということに重点がおかれており，必ずしも対象そのものについての情報を盛り込む必要がない．従ってキャッチコピーが与える原テキストの情報量はキャッチコピー作成者に大きく依存し，0%に近くなる可能性も十分に考えられる．逆の側面から見れば，キャッチコピー作成者は対象物の情報以上に，幅広い見識と語彙を持ち合わせていなければならないとも考えられる．

表 1 に，同一のテキストから作成した要約文とキャッチコピーの例を挙げる．

2.4 検索結果の出力にキャッチコピーを用いる利点

前節までを踏まえて，検索結果における「記事の要約」を「記事のキャッチコピー」に置換した場合に生ずる利点について述べる．

第一に，ユーザが検索で感じていた負担の軽減である．従来の検索で記事の要約が表示されていた部分にキャッチコピーを表示させることで，検索結果画面全体に表示される文字数を減らすことが可能になる．また簡潔かつ印象深い表現で構築されたキャッチコピーを示すことで，ユーザが直感に近い形で記事を選択でき，記事検索によって生じる負担を大幅に軽減できる．

第二に，検索行為自体の面白味の増加である．ユーザの環境及びその嗜好を把握することにより，ユーザ毎に適したキャッチコピーを生成することが可能になる．よって，キャッチコピーそのものを読むだけでもユーザは面白さを感じることができる．また，従来の検索では興味の持てなかった記事に対しても触れる機会が増え，新たな知識の獲得に繋がる．

3 システム概要

本章においてシステム概要について述べ，4 章，5 章，6 章，7 章において各モジュールの詳細について述べる．

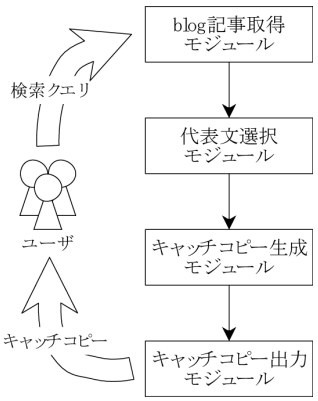

図 1: システムの構成

3.1 システム構成

システム構成を図 1 に示す．

以下で，システムの各モジュールを処理の流れに沿って簡潔に説明する．

blog 記事取得モジュール ユーザによって入力された検索クエリを基に，blog 記事の URL を取得して WWW から blog 記事本文を得る．

代表文選択モジュール blog 記事取得モジュールによって得られた blog 記事本文から重要語を抽出し，それを元に記事の代表文を選択する．

キャッチコピー生成モジュール 代表文選択モジュールによって得られた代表文を，いくつかの文章表現技法によって改変し，キャッチコピーとする．

キャッチコピー出力モジュール 生成されたキャッチコピーを，blog 記事の情報と共にユーザへ提示する．

本システムのモジュール部分は，Ruby による CGI プログラムを用いて作成した．また，入出力インタフェースにおける HTML ファイルでは特殊な処理を行っていないため，一般的なブラウザで本システムの利用が可能となっている．

表 1: 要約文とキャッチコピーの比較

原テキスト	らんちたいむの弁当も結構凶悪なボリュームだけど… 写真は柴崎駅南口の蕎麦屋「小川」のランチメニュー．最初，もりそばと冷やしたぬきうどんの二択なのかと思って注文したら何も聞かずに厨房にひっこんだので，デフォルトがそばなだけかと思ってたら…． もりそばと冷やしたぬき，両方できてきました… 炭水化物が，そば・うどん・ごはん…生協の食堂のトレイくらいのお盆にいっぱいになるくらいの量…．いや，別にまずくはないので 770 円は安いのだが．…ちゃんと食べましたよ？
要約文 （冒頭を取り出した場合）	らんちたいむの弁当も結構凶悪なボリュームだけど… 写真は柴崎駅南口の蕎麦屋「小川」のランチメニュー．
要約文 （重要度が高い文を抽出した場合）	炭水化物が，そば・うどん・ごはん… 生協の食堂のトレイくらいのお盆にいっぱいになるくらいの量…．
キャッチコピー	今日のランチは炭水化物盛り合わせ

3.2 インタフェース

本システムのインタフェースを，図 2 で示す．

ユーザは図 2 の (1) に検索クエリを入力し，検索ボタンを押す．検索結果として，(2) に blog 記事の題名，記事に含まれる画像のサムネイル，システムが生成したキャッチコピー，記事作成日時，検索実行時間が表示される．

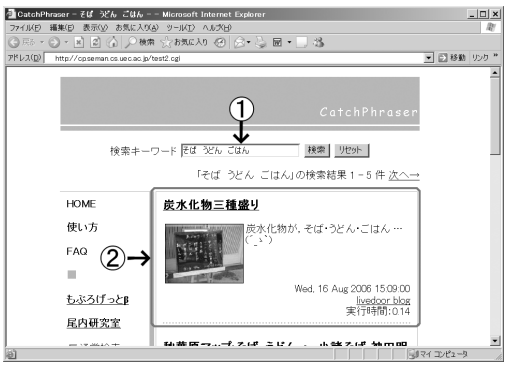

図 2: “きゃっちふれいざ” のインタフェース

4 blog 記事取得モジュール

ユーザから与えられた検索クエリを基に，もぶろげっと TM API を用いて．blog 記事検索を行い，記事の URL を得る．得られた URL にアクセスし，HTML 文書を読み込むことで blog 記事を取得する．

もぶろげっと TM もぶろげっと TM[5] は画像付き blog を検索できる検索エンジンである．通常のブラウザによるアクセスの他に，Web アプリケーション開発者向けにもぶろげっと TM API を公開している．もぶろげっと TM API は，REST タイプの Web サービスであり，もぶろげっと TM の検索結果やサムネイル画像を独自のアプリケーションに取り入れることが可能なインタフェースである．検索結果の要求は，“`http://api.mobloget.jp/`” に対して GET メソッドでパラメタを指定することによって行う．戻り値となるデータは，RSS もしくはテキストで返される．

本システムでは，ユーザが入力した検索クエリからテキスト形式で結果を受け取り，以下の要素を取得する．

表 2: 文を区切る記号

。	(句点)
.	(終止符)
!	(エクスクラメーションマーク)
?	(クエスチョンマーク)
♪	(音符記号)
\n	(改行)

- blog 記事の URL
- blog 記事の題名
- blog 記事の序文（本文の先頭部分）
- blog 記事の作成日時
- blog 記事のサムネイル画像への URL

blog 記事のサムネイル画像への URL も同時に取得することで，より直感的に理解しやすい出力インタフェースの作成が可能となる．

5 代表文選択モジュール

代表文選択モジュールは，キャッチコピー生成の土台になる blog 記事本文の中から最も重要と思われる一文を代表文として選択する．

blog 記事取得モジュールで得られた blog 記事から記事本文を抽出し，一文ごとに分割する．本システムでは，表 2 に示す記号で区切られた文字列を一文とした．

分割された各文に対して MeCab[2] を用いて形態素解析を行い，TF-IDF 法によって文中に登場する単語の重要度を算出する．文中における全単語の重要度の合計値を，その文の重要度とする．全文に対しての重要度算出が終了した時点で，最も重要度の高い一文を代表文とする．

[2] http://chasen.org/~taku/software/mecab/

5.1 TF-IDF 法による方法

TF-IDF 法は，テキストから重要語を抽出する方法の一種である [6] .

TF（term frequency）とは，対象テキスト内において，ある語がどれくらいの頻度で出現するかを表したものである．

IDF(inverse document frequency) とは，ある語が出現するテキスト数が少ないほど対象テキストを特徴付けているとしたものである．値の算出には，電子化された大量のテキストデータの集合であるコーパス（corpus）を用いる．式 1 に IDF 値の計算式を示す.

$$IDF(w) = \log_2 \frac{N}{N(w)+1} \tag{1}$$

$IDF(w)$	単語 w の IDF 値
N	コーパス内のテキスト総数
$N(w)$	コーパス内の単語 w を含むテキストの総数

TF-IDF 値は上記 TF と IDF の積によって表されるため，ある単語が特定のテキスト内においてのみ出現し，かつそのテキスト内で頻発する場合に大きな値をとる．従って，TF-IDF 値の大きな単語は，そのテキストをよく特徴付ける重要語と判定することが可能になる．

TF-IDF 法が抱える弱点として，文字数の少ない文章中の単語に対して適切な TF-IDF 値を与えられないという点が挙げられる．一般的な Web サイトで見られる文章と比べて，blog 記事では文の数が少ない．そのために，一つの記事内で一回しか出現しない単語が大量に存在する．これらの単語の TF 値はいずれも 1 となり，この状態で TF-IDF 値を算出しても IDF 値を計算していることと変わりがない状態となってしまう．このような状態では，各単語の重要度を正しく判別するのは難しい．

5.2 TF-IDF 値算出法の改良

本システムでは，blog 記事を説明するのに相応しい一文が代表文となるよう，TF-IDF 法による TF 値算出の際に以下で述べる改良処理を施す．

記事題名中に登場する単語に対しての処理　一般的なグルメ系 blog では，記事題名に「食べた物の名称（及び店の名前と値段）」を記述する傾向が強い．即ち，記事題名は記事本文の内容を把握する上で重要な固有名詞を含んでいる可能性が高い．よって，記事題名中で使用された単語が記事本文中で登場した場合，その単語の TF 値を通常の 3 倍として計算する．

グルメ関連単語に対しての処理　扱う blog 記事のジャンルをグルメ系 blog と固定していることから，予めグルメ関連用語を辞書化しておき，それらへの重み付けを高くすることでより信頼性の高い重要度計算が可能になると考えられる．よって，グルメ関連用語と判定された単語については TF 値を通常の 2 倍として計算する．この処理には，後述の 6.1 と共通のグルメ用語辞書を用いる．

blog 記事の要旨に関わりのない単語に対しての処理　blog 記事本文内には，広告文や blog ランキング投票リンク等の本文に無関係な記述が存在する場合がある．これらの記述には，「ランキング」「クリック」等で代表されるような特定の単語が含まれている場合が多い．よって，blog 記事の要旨に関わりのない単語と判定された単語については，TF 値を通常の 0.1 倍として計算する．

5.3　グルメ blog 特有の表現への対応

本システムでは TF-IDF 法を用いるために文を品詞単位に分解する形態素解析を行っている．形態素解析器として MeCab を利用している．

MeCab は解析用に標準の辞書を持っているが，グルメ系 blog に頻出する固有名詞の登録が少なく，正しく形態素解析できない場合がある．そのため TF-IDF 値の算出も正しく行えず，辞書に登録されている単語と登録されていない単語の間で不平等が生じる．これを解決するため，グルメ関連用語と blog 関連用語について約 700 語を独自に追加登録し，形態素解析の失敗が少なくなるようにした．

6　キャッチコピー生成モジュール

キャッチコピー生成モジュールは，代表文選択モジュールによって得られた代表文に対し，以下で述べる言い換え・伏せ字・修飾語追加・倒置・文末加工の表現技法を適用してキャッチコピーへと変化させるモジュールである．

各表現技法が適用可能な状況において，実際にその表現技法を用いるか否かについては 0%から 100%までの確率で指定可能である．

6.1　言い換え

言い換えとは，意味が近似的に等価な言語表現の異形（paraphrase）を指す [7]．言い換えの種類として，語彙・構文的言い換え，参照的言い換え，語用論的言い換えの三種類があるとされている．

語彙・構文的言い換えは，意味内容を同一に保ったまま文の構造を変化させたり，同義語を用いて表現を置換したりすることによって実現される言い換えである．

参照的言い換えは，大域的文脈や話者の情報に基づいた参照表現により実現される言い換えである．

語用論的言い換えは，語用論的効果が同一である文の置換によって行われる言い換えである．語用論的効果とは，「話者がそれを発することによって達成できると期待するコミュニケーションの目的」と定義される．語用論的効果に基づいた言い換えは，現代文法理論や辞書の意味理解を超えた範囲で行われるものであり，工学的実現は難しいとされている．

言い換えを用いたキャッチコピー　参照的言い換えは，本来であれば言葉が発せられた文脈や談話の状況を把握する必要がある．この参照的言い換えをグルメ系 blog 記事に用いようとした場合，予め頻繁に出現する単語とそれに対する参照表現をデータベースとして持ち合わせていれば，実現可能であると考えられる．以下に，参照的言い換えを用いたキャッチコピーの生成例を示す．

(適用前)
海老チリ を頂く．これ，めっちゃくっちゃおいしい !!!
(適用後)
某中華料理 を頂く．これ，めっちゃくっちゃおいしい!!!

実装　言い換えの対象となる単語を集めたグルメ用語辞書と，言い換え後の単語をグループ化した置換辞書を予め作成する．

グルメ用語辞書には，グルメ系 blog 記事でよく見られる名詞，形容詞，形容動詞を 500 個程度収める．各単語はそれぞれ意味別にグループ分けを行い，一つのグループに対して一意のグループ番号を与える．

置換辞書には，グルメ用語辞書に登録されている単語に対しての言い換え表現を収める．各表現は意味別にグループ分けを行い，グルメ用語辞書内のグループ番号と同じグループ番号を与える．

代表文を形態素解析した結果得られた名詞・形容詞・形容動詞がグルメ用語辞書に存在していた場合，対応する単語をグループ辞書の中から取り出して置き換える．

表 3 に，グルメ用語辞書と対応する置換辞書に収められている単語の一例を示す．

表 3: グルメ用語辞書と置換辞書の一例

対象語	グループ語
ラーメン	極上ラーメン
らーめん	らうめん
	ら〜めん
	絶品ラーメン
ニンニク	にんにく
にんにく	にんにく卵黄
	異臭発生源
月曜日	マンデー
	週の初め
	ブルーマンデー
金曜日	華の金曜日
	フライデー
	FRIDAY
	13 日の金曜日

6.2　伏せ字

伏せ字は主に，固有名詞を公にすることを避けるための手段として用いられる．一方，文中の真に迫る箇所を伏せることで読み手の注意を引きつけるといった効果もある．そこで代表文中に出現する固有名詞を記号で伏せ，キャッチコピーを生成する手法が考えられる．

また，読み仮名の確定している固有名詞に関しては，イニシャルで置換することによって伏せ字と同等の効果を狙うこともできる．

実装　代表文に対して形態素解析を行い，固有名詞を抽出する．抽出された固有名詞を，同じ文字数の記号「○」もしくは読み仮名に適するイニシャルを示すアルファベットで置換する．以下に，伏せ字を用いたキャッチコピーの生成例を示す．

(適用前)
成田 ゆめ牧場のイチオシ
(適用後)
○○ ゆめ牧場のイチオシ

6.3　修飾語追加

ある語句の概念を限定したり，意味を詳しく説明したりする語を一般に修飾語と呼び，名詞・代名詞を修飾する連体修飾語，動詞・形容詞・形容動詞を修飾する連用修飾語がある．

本システムでは代表文中に名詞が出現した場合，その直前に修飾語（主に連体修飾語）を挿入する．

ただし，(i) 直前の単語が名詞である場合，(ii) 接尾辞である場合，(iii) 非自立語である場合には文法的あるいは文脈的に不自然な文となる可能性があるため，修飾語の追加は行わない．

実装　修飾語を集めた修飾語辞書を予め作成する．代表文に対して形態素解析を行い，名詞を抽出する．修飾語辞書から修飾語を一つ抜き出し，抽出された名詞の直前に置く．なお，直前に名詞を持つ名詞・接尾辞・非自立語に関してはこの機能を適用しない．

(適用前)
次の日も暇だったので
(適用後)
次の日も 超絶 暇だったので

6.4　倒置

倒置とは，文章表現における表現技法の一つであり，通常の語順を逆にして言葉の意味や印象を強める方法を指す．

倒置を代表文に適用する場合，「、」「♪」等の意味的区切りとなる記号の前後に位置する語句を入れ替える手法が考えられる．

実装　代表文中に読点「、」が出現した場合，読点前後の文字列を入れ替える．読点それ自体は，予めつなぎ表現を集めたつなぎ表現辞書から取り出された記号によって置換される．

(適用前)
ここのケーキはイチゴが多くて、本当に大好きです！
(適用後)
本当に大好きです！ここのケーキはイチゴが多くて。

6.5　文末加工

短文となるキャッチコピーにおいて，文末は文章の印象を決定する重要な要素の一つである．

そこで，代表文の末尾に様々な表現を付加する手法を考える．付加する語句としては，「感嘆詞」「顔文字」「記号」が挙げられる．

実装　「感嘆詞」「顔文字」「記号」を集めた文末表現辞書から語句を一つ抜き出し，代表文の文末に付加する．同時に，代表文の文末に最初からある句点「。」等を削除する．

(適用前)
パウンドケーキも作ってみました。
(適用後)
パウンドケーキも作ってみました (´・ω・`)

7　キャッチコピー出力モジュール

blog 記事単体は，一般に次の要素から構成される．

題名　記事の主題を端的に示す文

本文　記事を構成する主な文章

作成日時　記事が blog 上に作成された日時

コメント　記事に対して寄せられた，読者の意見・感想

トラックバック　関連ある別の記事から送信されたリンク情報

本システムでの検索結果表示インタフェースでは，上述の「題名」「作成日時」に加えて「キャッチコピー」「記事に含まれる画像のサムネイル」「検索実行時間」を表示する．本システムによる出力例を図 3 に示す．

7.1　通常検索モード

本システムでは，キャッチコピーを表示する代わりに blog 記事本文の先頭数行を表示する，通常検索も可能となっている．

通常検索での出力は，一般的な検索エンジンで使用されているスタイルである．従って，システムによって生成されたキャッチコピーによる検索結果と通常検索結果との比較が可能となっている．通常検索モードによる出力例を図 4 に示す．

8　考察

8.1　blog 記事本文の分割に関する問題

本システムでは，blog 記事本文を一文ごとに分割する際の指標として，5 で挙げた記号「。」等を利用した．しかしながら，一部の blog 記事では正しく文を分割することができなかった．以下に，代表的な失敗例とその考察を述べる．

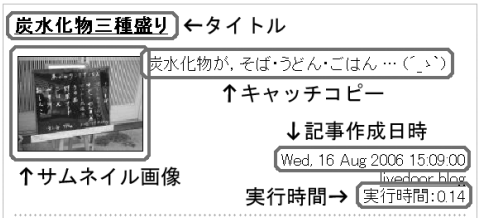

図 3: 出力インタフェース

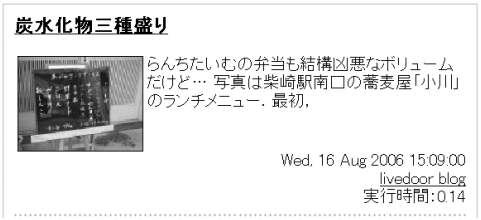

図 4: 通常検索結果

顔文字が区切り記号として使用されている

って私はバナナ評論家かっ＾＾；とりさんバナナタルト食べたことないです（＞＜）

文末に顔文字が使用されている場合は非常に多く，これらの blog 記事本文は適切に分割を行うことができなかった．

予めシステム側で顔文字辞書を作成し，区切り記号の判定で用いるという解決策も考えられるが，日々新たな顔文字が登場する昨今においては難しい．「特殊記号が一定数以上連続した場合は顔文字として判定」等の，辞書を必要としない対策が有効であるとも考えられる．

文末以外の位置で意図的に改行がなされている

もちろんフライパンで レモンやオレンジ果汁と洋酒と焼く というものも 結構あって こちらもとっても美味しそう

blog 記事作成者が文末以外の位置で意図的に多数の改行を入れた場合，適切な分割ができなくなる可能性がある．

意図的な改行で区切られた文は短くなることが多く，文の長さを考慮することで文として採用しない方法も考えられる．しかし，そのような文の中にはキャッチコピーの生成にふさわしいものが含まれていることもあり，単純に長さのみで切り捨てるのは問題であり慎重な調整が必要であると考えられる．

8.2 キャッチコピー生成時のランダム性に関する問題

キャッチコピー生成において，修飾語付加・倒置・文末加工においては使用する表現を辞書ファイルからランダムに選んでいる．このため，代表文の内容と食い違う表現が使用されてしまう恐れがある．例えば “ありがとう” という肯定的な表現が登場する文に “（怒）” という否定的な内容の文末加工が行われる場合などである．

代表文の内容に適した表現を選択するには，代表文の意味を汲んだ分析を行う必要がある．一方，本システムの目標である “面白みのあるキャッチピーの出力” という側面から考えた場合，ある程度のランダム性は残すべきであると考えられ，“文章の一貫性” と “意外性によるおもしろさ” の折り合いをどうつけるか考慮していく必要がある．

8.3 パーソナライズ

現在のシステムでは，キャッチコピー生成過程においてユーザの嗜好を利用したパーソナライズは行っていない．しかし，本来キャッチコピーは読み手となるユーザに適したものとなるよう生成されるべきである．

キャッチコピー生成においてユーザ毎に変えることが望ましいと考えられるパラメタを以下に挙げる．

表現技法の使用確率　過去にユーザが “おもしろい” と感じたキャッチコピーに使用されていた各表現技法（言い換え・伏せ字・修飾語付加・倒置・文末加工）

の使用頻度を記録し，新たなキャッチコピー生成に適用することで，よりユーザによって読みやすくおもしろいキャッチピーを生成できると考えられる．

使用される表現及びキーワード　文体の種類（「だ・である」調か「です・ます」調であるのか）・年代別の流行語・顔文字使用の有無等に関してユーザが好む文章表現を把握する．これらを基に，各表現技法で使用する単語を変化させる（辞書を切り替える）ことで，よりユーザに適したキャッチコピーを生成できると考えられる．

9　おわりに

本研究では，blog 記事の内容に基づいたキャッチコピーを生成して出力するシステム “きゃっちふれいざ” を試作した．

生成されたキャッチコピーによって，blog 記事の検索・選択を新たな尺度で行うことや，キャッチコピーそのものを閲覧して楽しむことが可能になった．

今後は，ユーザ個人の情報を保存して好みの話題・単語等を利用するといった，パーソナライズされたキャッチコピー生成の実装が課題となる．また，各モジュールについての細かな調整や，使用している辞書ファイルの拡張性も改善していく必要がある．

参考文献

[1] 長尾真：自然言語処理，岩波書店 (1996).

[2] 砂山渡，谷田内正彦：文章の特徴を表すキーワードを発見して重要文を抽出する展望台システム，電子情報通信学会論文誌 D-I，Vol. J84-D, No.2, pp.146–154 (2001).

[3] 難波英嗣，奥村学：書き換えによる抄録の読みやすさの向上，情報処理学会 自然言語処理研究会，99-NL-139-9 (1999).

[4] 松村明，大辞林 第二版，三省堂 (1999).

[5] 井原伸介，林貴宏，尾内理紀夫：画像情報を含む blog 記事検索システムの開発，電子情報通信学会論文誌 D-I, Vol.J89-D, No.6, pp.1236–1247 (2006).

[6] 西尾章治郎，田中克巳，上原邦昭，有木康雄，加藤俊一，河野浩之：岩波講座 マルチメディア情報学 8 情報の構造化と検索，岩波書店 (2000).

[7] 乾健太郎，藤田篤：言い換え技術に関する研究動向，自然言語処理，Vol.11, No.5, pp.151–198 (2004).