

Web 情報を用いたキーワード抽出による タグづけ支援

田中英人^{†1} 丸山一貴^{†2} 寺田実^{†1}

膨大な文書の管理を行なうにあたって、文書中の特徴的なキーワードを自動で抽出したものをタグとして付加することで文書整理の効率化を図る手法が考えられる。しかし、メモのような短い文書には含まれる単語数が少ないため、満足なキーワードが抽出されにくい。

本研究では、文書中に存在しない語を Web から取得してキーワード候補に含めることで、短い文書に対するキーワードづけを実現し、より適切なタグ候補の提示を行なう手法を提案する。

Support of Tagging by Keyword Extraction using Web Information

HIDETO TANAKA,^{†1} KAZUTAKA MARUYAMA^{†2}
and MINORU TERADA^{†1}

To manage a huge amount of documents, adding a keyword which is extracted from the document as a tag may be efficient. But a short document like a memo is hard to extract a keyword, because the number of words in the document is limited.

In this paper, we realize the keyword extraction from short documents by using web information, and propose the method that shows more appropriate tag candidate.

1. はじめに

思いついたこと等を簡潔に記しておく際に、我々は「メモ」をとる。現在では PC を利用することによって、より多くの文書が管理できるようになった。しかし、蓄積された文書の数が増えると、検索の必要性が生じる。

一方で、livedoor Blog^{*1}などのブログエントリやニュースサイト mediajam^{*2}などのニュース記事のように、Web 上のコンテンツは「タグ」を付加することによって管理されることが多い。文書の話題に即したタグづけを行えば、それを見ただけで文書の大まかな内容が把握でき、所望の情報にたどり着くことが可能である。また、類似した話題に関する文書同士には、同一のタグを付加することで、文書をカテゴリ分けすることができる。

しかし、例えば自分のよく知らない話題に関するメモであったり、特に明確なテーマを決めずにメモを書いた場合など、「タグを考える」ということが容易でないケースが考えられる。また、たとえタグを思いついたとしても、それより適切なタグ候補が存在する可能性もある。そこで本研究では、文書から特徴的なキーワードを自動で抽出し、タグ候補としてユーザに提示する手法に着目した。

一般的に、キーワード抽出とは、文書中の単語に対して重要度の計算を行ない、キーワードを決定する。しかしながら、メモのように短い文書には含まれる単語が少ないため、キーワード候補が少数に限られてしまい、満足なキーワードが抽出されない可能性が考えられる。つまり、文書中の単語のみについて考えるのではなく、それらの単語から連想される関連語などもキーワード候補に含める必要がある。そこで、文書に存在しない単語やフレーズをタグ候補として提示するにあたっての問題点を以下にまとめる。

● どのようにキーワード候補を追加するか

前述の通り、本研究では文書中には存在しない単語を含めたキーワード抽出を実現したい。ゆえに、どのようにして新たにキーワード候補を追加するかが問題となる。

● どのような指標を用いて重要度の計算を行なうか

一般的なキーワード抽出法では、文書中に存在する単語を対象に重要度の計算を行なう。したがって、文書中に存在する・しないに関わらず適切な計算を行なうための手法を考えなければならない。

^{†1} 電気通信大学情報理工学研究科情報・通信工学専攻

Graduate School of Informatics and Engineering, The University of Electro-Communications

^{†2} 東京大学 情報基盤センター

Information Technology Center, The University of Tokyo

*1 <http://blog.livedoor.com/>

*2 <http://mediajam.info/>

2. 関連研究

2.1 キーワード抽出に関する研究

キーワード抽出に関しては、これまでに多くの研究がされてきた。ここでは、キーワード抽出手法についての研究および既存のキーワード抽出システムについて記述する。

tf-idf 法¹⁾

tf-idf 法は、文書中において重要なキーワードを抽出するアルゴリズムである。単語 i のスコア $tfidf_i$ は、出現頻度 tf_i および逆出現頻度 idf_i の 2 つの指標を用いて、以下のように計算される。

$$tfidf_i = tf_i \times idf_i \quad (1)$$

$$tf_i = \frac{n_i}{\sum_k n_k} \quad (2)$$

$$idf_i = \log \frac{|D|}{|\{d : d \ni t_i\}|} \quad (3)$$

ここで、 n_i は単語 i の出現頻度であり、 $|D|$ は総ドキュメント数、 $|\{d : d \ni t_i\}|$ は単語 i を含むドキュメント数を表す。計算式において、 idf は一般語のスコアを下げる役割を果たし、より特徴的なキーワードを抽出する。

Yih らの研究²⁾

Yih らは、Web ページから適切なキーワードを抽出する手法について研究を行なった。単語の重要度計算には、tf-idf 値を始めとする様々な指標を用いている。ここで、指標の例をいくつか記す。

- 名詞または固有名詞であるかどうか
名詞および固有名詞がキーワードとして適していると考え、これらの品詞を持つ単語の重要度が高くなるよう計算している。
- html 文書中の出現位置
単語が html 文書の title タグに含まれているのか、あるいは body タグに含まれているのか等の情報を用いて計算をしている。また、文章中の何番目の単語であるのかといった位置情報も考慮している。
- 文章およびフレーズの長さ
単語が含まれる文章の長さを用いて計算を行なっている。また、複数の単語からなるフレーズに含まれている場合には、その長さも考慮して計算している。

本研究では、Yih らの論文で挙げられている様々な指標の中でも、特に品詞情報による重要度計算に着目した。品詞情報は、単語が固有に持つ特徴であるため、文書中に存在するかないかに関わらず、重要度の計算を行なうことができる。したがって、本研究の提案する単語の重要度計算アルゴリズムにおいて、この概念を参考にした計算指標を用いる。

キーワード自動抽出システム「言選 Web」³⁾

言選 Web は、与えられた文章からキーワードを抽出するシステムである。抽出されたキーワードは、重要度と共に出力される。また、重要度の計算には tf-idf 法を使用している。テキストファイルや PDF、Web ページからのキーワード抽出が可能であり、複合語による専門用語などの抽出に特化している。

また、このシステムを利用した Windows 用ソフトウェアとして、termex がある。言選 Web の機能に加えて、計算パラメータの設定機能や学習機能を持つなど、個人での使用に適している。

キーフレーズ抽出 Web API

Yahoo!デベロッパーネットワーク⁴⁾の提供する Web API であり、日本語の文章をパラメータとしてリクエスト URL にアクセスすると、独自の算出方法により抽出されたキーフレーズが重要度と共に返される。XML 形式等で出力されるため、プログラミングによる解析が可能であり、Web API を利用して開発されたツールも多く存在している。

2.2 Web 上の情報を用いた関連語の取得に関する研究

Google サジェスト

Google^{*3}の提供する検索支援ツールである。検索語を入力すると、その語と関連性の高いと思われる候補を補完することで検索を支援する。ユーザ個人の検索履歴を扱うのではなく、検索全体に対する人気度を用いてランクづけされた関連語を提示する。

文書中の単語の関連語を取得するにあたって、Web 上の情報は有用であると考えた。実際に Google サジェストを使用してみると、「自分では思いつかなかったが、検索しようと思っていた事柄に関連がある」といった関連語が多く提示される。このように Web 情報を用いることで、重要なキーワード候補が取得できることを期待する。

*3 <http://www.google.co.jp/>

3. 提案システム

本研究では、入力された文書に対するタグ候補を予測し、ユーザへ提示するシステムを作成した。文書長は最高でも 200 文字程度の比較的短いものを想定して実装を行なう。システムの概念図を図 3 に示す。

提案システムの流れを述べる。まず、内部キーワードとして入力文書中のキーワードを抽出する。次に、Web 上に公開されている情報を用いて、内部キーワードと関連のあるフレーズを取得し、これを外部キーワードとする。これらの内部キーワードおよび外部キーワードを合わせた語群に対して、本研究の提案するアルゴリズムによって重要度の計算を行ない、キーワードリストを作成する。このキーワードリストを基に、タグ候補を提示する。また、過去に入力された文書のうち、現在の入力文書と関連のあると思われるものを、それぞれの文書における存在単語の類似性を計算することで選別する。もし関連文書があれば、それにつけられているタグを関連タグとしてユーザへ提示する。

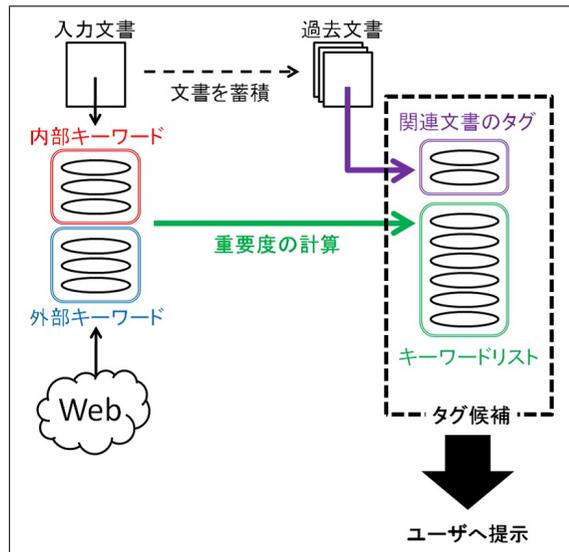


図 1 概念図

3.1 キーワード属性

キーワードの属性として、以下に示すように内部キーワードおよび外部キーワードを定義し、それらを総じた語群を入力文書のキーワードとする。

3.1.1 内部キーワード

既存のキーワード抽出システムと同様に、文書中に存在する語は重要なキーワードとなり得る。そこで、文書中から抽出されたキーワードを、内部キーワードとして扱う。

形態素解析エンジン MeCab⁵⁾を用いて、入力された文書を単語ごとに分解する。得られた単語について、表 1 に示す品詞を持つもののみを考慮し、なおかつ文書中において隣接する単語は、結合して複合語を生成する。表 1 に示した品詞リストは、出来る限り不自然な複合語を生成してしまわないよう著者が分類した結果である。

例えば「卒業論文を執筆する。」という文書が与えられた場合、形態素解析の結果は「卒業」、「論文」、「を」、「執筆」、「する」、「。」となり、一般名詞である「卒業」、「論文」、「執筆」について考える。ここで、文書中において「卒業」と「論文」は隣接していることから、「卒業論文」、「執筆」が得られる。

このようにして得られた語の集合を、入力文書に対する内部キーワードと定義し、 $I = \{i_m | m = 0, 1, 2, \dots, n - 1\}$ とする。

3.1.2 外部キーワード

本研究の目的は、文書中からは抽出されなかった語を含めたタグ候補の提示を行なうことである。そこで、Web 上で公開されているツールを利用して、キーワードを追加する。こうして得たキーワードを、外部キーワードとして扱う。

Yahoo!デベロッパーネットワーク⁴⁾の提供する「関連検索ワード Web API」を使用して、内部キーワードに対する関連検索語を取得し、これを 1 次関連語と定義する。関連検索ワード Web API は、入力したクエリと一緒に Web 検索されやすいフレーズを返す API

表 1 内部キーワードを生成する品詞リスト

品詞名	説明	単語例
一般名詞	一般的な対象を表現	「論文」「大学」「パソコン」
固有名詞	人名や地名など、特定の対象を表現	「田中」「東京」「チューリング」
サ変接続名詞	「する」に接続	「研究」「買い物」
接続詞的名詞	2 つの名詞を接続	「VS」「対」「兼」
数	数字および漢数字	「1」「百」
接頭詞	名詞や数の前に接続	「非」「第」「約」
接尾辞	名詞や数の後に接続	「くん」「メートル」

であり、本研究では関連語を取得する手段として用いた。また、取得した1次関連語に対する関連検索語をさらに取得し、これを2次関連語と定義する。本研究では、1つの内部キーワードに対して3つの1次関連語を、1つの1次関連語に対して5つの2次関連語をそれぞれ取得する。

このようにして得られた語の集合を、入力文書に対する外部キーワードと定義し、 $O = \{o_m | m = 0, 1, 2, \dots, n-1\}$ とする。

3.2 関連タグ

類似した複数の文書に同一のタグがつけられることで、検索の効率化が期待できる。ゆえに本研究では、入力文書Aとシステムに蓄積された過去文書Bとの類似度 $R_{AB}(x)$ を計算し、閾値 $R_{threshold}$ を越えた過去文書については関連文書と見なし、それにつけられているタグを関連タグとして提示する。

M 種類の単語からなる文書Aと、 N 種類の単語からなる文書Bについて、 x 個の単語が一致する場合の文書間の類似度 $R_{AB}(x)$ を式(4)のように独自に定めた。ただし、 $0 \leq x \leq M, 0 \leq x \leq N, M \neq 0, N \neq 0$ であるとする。

$$R_{AB}(x) = \frac{x}{M \times N} \times \sqrt{\frac{M^2 + N^2}{2}} \quad (4)$$

文書間に一致する単語が存在しないとき、すなわち $x = 0$ であるとき、 $R_{AB}(x) = R_{AB}(0) = 0$ となる。また、両文書の持つ単語の種類数が等しく、かつその全てが一致するとき、すなわち $M = N = x$ であるとき、 $R_{AB}(x) = R_{AB}(M) = R_{AB}(N) = 1$ となる。

3.3 重要度の計算アルゴリズム

得られたキーワードの重要度を計算し、入力文書に対するキーワードリストを作成する。計算には以下の3つの指標を用いる。

- キーワード属性
- 品詞情報
- tf-idf

また、内部キーワードの集合 I と外部キーワードの集合 O を足し合わせて、キーワード全体の集合 $P = \{p_m | m = 0, 1, 2, \dots, n-1\}$ を定義する。そして、 p_m をさらに形態素解析して得られた i 個の単語を t_{mj} として、 $p_m = \{t_{mj} | j = 0, 1, 2, \dots, i-1\}$ と表すことにする。

3.3.1 キーワード属性

得られたキーワード p_m の属性により、初期スコア $Score(p_m)$ を決定する。内部キーワー

ドから1次関連語を、1次関連語から2次関連語をWeb上から取得していくにつれて、文書の内容とは関連性の低いキーワード候補も増えてしまう。そこで、得たキーワードと文書との間の一般的な関連度は、内部キーワード、1次関連語、2次関連語の順に低下していくものと考えた。ゆえに、 p_m が内部キーワードであれば $Score(p_m) = 100$ 、1次関連語であれば $Score(p_m) = 75$ 、2次関連語であれば $Score(p_m) = 50$ と定めた。

3.3.2 品詞情報

p_m を構成するうちの一つの単語 t_{mj} のスコアを $s(t_{mj})$ として、以下のように定義する。

$$s(t_{mj}) = Score(p_m) \times Pos(t_{mj}) \quad (5)$$

ここで、 $Pos(t_{mj})$ は単語 t_{mj} の持つ品詞情報により決定される値であり、以下の表2のとおりに定義した。

Web上のタグづけされた文書について独自に調査を行なった結果、半数近くのタグが一般名詞または固有名詞を含んでいるように見受けられた。ゆえに、タグとしての適切さは高いと考えられる。また、接頭詞および接尾辞については、原則的に独立して存在することはなく、かつ接続する単語の意味をより具体化する役割を果たす。以上の理由から、一般名詞、固有名詞、接頭詞、接尾辞の品詞を持つ単語 t_{mj} に対する $Pos(t_{mj})$ は1.0とし、これを $Pos(t_{mj})$ の最大値に定めた。以降、この値を基準にして考える。

サ変接続名詞は「する」に接続する単語につけられる品詞であるため、行動や思考、状態などを表す場合が多い。したがって、タグとして有用であると考えた。しかし、一般的には「誰か」あるいは「何か」が存在してこそその行動や状態であるため、それらを表す一般名詞および固有名詞と比較すると重要性はやや低い。また、接続詞の名詞は、名詞同士を接続する役割を持つ単語につけられる品詞であり、主に一般名詞か固有名詞に接続することが多い。本研究で形態素解析する際に用いているMeCabの持つ辞書において、接続詞の名詞は「VS」「対」「兼」の3つしか登録されておらず、たとえば「A対B」というフレーズを含む文書については、「A」や「B」とタグづけするほうが検索する際の柔軟性が高く妥当であると判断した。以上の理由から、サ変接続名詞および接続詞の名詞の品詞を持つ単語 t_{mj} に

表2 品詞情報を用いた重要度計算

単語 t_{mj} の品詞	$Pos(t_{mj})$ の値
一般名詞, 固有名詞, 接頭詞, 接尾辞	1.0
サ変接続名詞, 接続詞の名詞	0.75
数	0.5
その他	0.25

に対する $Pos(t_{mj})$ は 0.75 とした。

数値や日付など、数を含むフレーズはタグとしての重要性は低いと考える。何を意味する数値なのか、何を表す日付なのかということが重要であるため、数の品詞を持つ単語 t_{mj} に対する $Pos(t_{mj})$ は、一般名詞や固有名詞の半分である 0.5 とした。また、それ以外の品詞については、 $Pos(t_{mj})$ の値を 0.25 としてある。

3.3.3 tf-idf

キーワード全体の集合 P に対して、tf-idf の計算を行なう。Perl 用キーワード抽出モジュール Lingua::JA::Summarize⁶⁾ の概念を参考に、idf の計算には形態素解析エンジン MeCab の単語生起コストを用いた。単語生起コストとは、ある単語の出現しやすさを表す数値であり、MeCab の扱う辞書に予め定義されている。 p_m を構成するうちの一つの単語 t_{mj} の単語生起コストを $c(t_{mj})$ として、キーワード p_m の生起コスト C_m を以下のように独自に定義した。

$$C_m = \sqrt{\sum_{t_{mj} \in p_m} \{c(t_{mj})\}^2} \quad (6)$$

したがって、キーワード p_m の tf-idf 値 $tfidf_m^*$ は以下のように求める。ただし、 $N(p_m)$ は集合 P におけるキーワード p_m の重複数とする。

$$tfidf_m^* = tf_m^* \times idf_m^* \quad (7)$$

$$tf_m^* = \frac{N(p_m)}{|P|} \quad (8)$$

$$idf_m^* = C_m \quad (9)$$

3.4 タグ候補の提示

3.4 節に示した重要度の計算により、キーワード p_m の重要度 S_m は以下の式 (10) で算出される。

$$S_m = \frac{\sum_{t_{mj} \in p_m} s(t_{mj})}{i} \times tfidf_m^* \\ = \frac{\sum_{t_{mj} \in p_m} \{Score(p_m) \times Pos(t_{mj})\}}{i} \times \frac{N(p_m)}{|P|} \sqrt{\sum_{t_{mj} \in p_m} \{c(t_{mj})\}^2} \quad (10)$$

算出した重要度を基に、入力文書に対するキーワードリストを作成し、スコアの高い順にユーザへ提示する。

大相撲秋場所 14 日目 (25 日、両国国技館) 横綱白鵬が 4 場所連続 16 度目の優勝を果たした。13 日目まで 2 敗で追っていた豪風と嘉風がともに敗れたため自らの取組を待たずに優勝が決まった。結ひの一番では琴歐洲を下し、連勝を 6 に伸ばした。魁皇は稀勢の里を下し勝ち越しを決めた。

図 2 入力文書

【自動抽出されたタグ候補(重要度の高い順)】
赤:内部キーワード 青:1次関連語 緑:2次関連語
横綱白鵬 ブログ 白鵬 里海 両国国技館 大相撲 北原里英 相撲 豪風 琴歐洲 豪風 魁皇 福田町駅 座席表 連勝 記録 連勝記録 ロダ イナズマイレブ 白鵬
【類似した文書につけられている関連タグ】
相撲 白鵬 優勝

図 3 出力結果

3.5 動作例

提案システムの動作を確認するために、簡単なインタフェースを作成した。動作の一例として、システムに与えた入力文書を図 2 に、それに対する出力結果を図 3 にそれぞれ示す。

出力の結果はキーワードの属性によって色分けされており、図 3 を見ると文書中から抽出されたものではない外部キーワードも多く提示されていることが分かる。また、関連タグは他のキーワードとは別に表示している。なお、関連文書については著者が予め用意した。

4. 評価実験

提案アルゴリズムによるタグ候補および類似した過去文書による関連タグの妥当性について、評価実験を行なった。

200 文字程度のニュース記事に対して提案アルゴリズムによるタグ候補および関連タグを抽出し、それぞれの候補語が「タグとして適切かどうか」を被験者に判断してもらい、どれだけ選択されたかを考察する。また、既存のキーワード抽出システムを利用して、文書中から抽出されたキーワードのみをタグ候補として提示した場合の結果との比較も行なう。また、実験には本学の学部 4 年生、修士課程 1, 2 年生、博士課程 3 年生の合計 10 名に参加してもらった。全員、日常的にパソコンを使用している。

実験に使用したニュース記事データについて説明する。まず初めに、Yahoo!JAPAN^{*4)} のニュース記事より、「政治」「スポーツ」「芸能」などのカテゴリから、13 件の記事を選択した。次に、それぞれの記事の関連記事とされるものを 3~4 件ずつ選択し、合計 50 件のデータを得た。本研究では、これらの記事の冒頭部分を入力文書として実験に用いる。ただし、

*4 <http://www.yahoo.co.jp/>

記事を選ぶ際の条件は、「冒頭部分が 15 個以上の名詞句を含む 200 文字程度の文章であること」とした。以降、記事の冒頭部分を「本文」と呼ぶことにする。

得られた 50 件の記事について、同じ話題の記事が含まれないようランダムに 10 件ずつ分け、10 件 × 5 セットのデータセットを用意した。各実験とも、被験者 1 人につき 1 セットの記事データを評価してもらった。つまり、延べ 100 件の記事データが評価されたことになる。

4.1 実験方法

4.1.1 実験 1

それぞれの記事について、提案アルゴリズムを用いてキーワードリストを作成し、上位 20 個のキーワードを「システムによるタグ候補」とする。また、実験 1, 2 とともに Yahoo! デベロッパーネットワーク⁴⁾の提供するキーフレーズ抽出 Web API を比較対象とし、これにより抽出された最大 20 個のキーワードを「API によるタグ候補」とする。こうして得られた 2 種類のタグ候補を足し合わせた語群について、重複するものを 1 つにまとめ、ランダムに並び替えたものを被験者へ提示する。タグ候補の提示数は 25 ~ 40 個であり、平均すると 1 つの記事あたり 33.16 個であった。

被験者には、まず本文を読んでもらい、各記事について被験者自身の考えるタグを最大 3 つまで付けてもらった。次に、前述のタグ候補を提示し、それぞれの候補語が「文書に対するタグとして適切かどうか」を 2 択で選択してもらった。なお、実験 1 では、過去文書は蓄積されていないものとし、関連文書のタグは考慮しない。

4.1.2 実験 2

実験の流れは実験 1 と同様である。ただし、実験 1 で被験者にタグづけしてもらった文書を過去文書として用いて実験を行なう。

3.2 節に示した式 (4) について、閾値 $R_{threshold} = 2.0$ とする。つまり、入力文書 A および過去文書 B について $R_{AB}(x) \geq 2.0$ となるような文書 A, B を関連文書と見なし、実験 1 における入力文書 A に対するタグ候補に過去文書 B につけられているタグを追加する。これらのタグを「関連タグ」とする。つまり、「システムによるタグ候補」「API によるタグ候補」「関連タグ」の 3 種類のタグ候補を足し合わせた語群について、重複するものを 1 つにまとめてランダムに並び替えたものを被験者へ提示する。また、全 50 件の記事データについて $R_{AB}(x)$ を求めた結果、それぞれの記事に少なくとも 2 つは関連文書が存在した。

被験者には実験 1 で評価してもらったものとは異なるデータセットを提供し、提示されたそれぞれのタグ候補について「適切かどうか」を判断してもらった。

4.1.3 アンケート

各被験者には、実験 2 の終了後にアンケートに答えてもらった。設問は「提案システムに対する満足度 (5 段階評価)」「良かった点」「悪かった点」とし、その他に意見や感想などを自由に記述してもらった。

4.2 実験結果および考察

本研究では選択率、平均選択数、的中率の 3 つの指標を用いて評価を行なう。タグ候補の集合において被験者に選択されたものの割合を選択率とし、1 つの記事あたりに選択された平均のタグ候補数を平均選択数とする。さらに、提案システムが提示したタグと被験者自身が考えたタグとが一致した割合を的中率とする。

また、本研究では以下の 3 つの観点について考察する。

- システムの提示したタグ候補がどれだけ選択されたか
- 選択されたタグ候補はキーワードリスト中の何位であったか
- 過去の関連文書によるタグ候補の重要性

4.2.1 実験 1

被験者 1 人につき 10 件の記事を評価してもらったため、延べ 100 件の実験データを得た。提案システム、キーフレーズ抽出 Web API、全体の提示したタグ候補の合計および選択されたタグ候補の合計についてまとめたものを表 3 に示す。

10 名の被験者に対して提示したタグ候補の合計は 3316 個であり、そのうち 406 個の候補が選択された。ゆえに、タグ候補全体の選択率は 12.24% であり、1 つの記事あたりの平均選択数は 4.06 個となった。また、提案システムの計算結果により提示したタグ候補の合計は 2000 個であり、そのうち 275 個が選択されたため、選択率は 13.75%、平均選択数は

表 3 実験データ

	提案システム	API	全体
提示したタグ候補数	2000	1884	3316
選択されたタグ候補数	275	237	406

表 4 選択率および平均選択数

	提案システム	API	全体
選択率 (%)	13.75	12.58	12.24
平均選択数 (個)	2.75	2.37	4.06

2.75 個となった。さらに、キーフレーズ抽出 Web API の出力結果により提示したタグ候補の合計は 1884 個であり、そのうち 237 個が選択されたため、選択率は 12.58%，平均選択数は 2.37 個となった。選択率および平均選択数をまとめたものを表 4 に示す。

続いて、提案システムが提示するタグ候補の選択率および平均選択数とキーワードリスト中の順位との関連性について考察する。そこで、キーワードリストの上位から順に 5 つずつ切り出したものについて、選択率および平均選択数をまとめたものを表 5 に、グラフ化したものを図 4 に示す。

図表より、キーワードリストの 1～5 位のタグ候補と 6～10 位のタグ候補との間に大きな開きがあるのが分かる。選択率は約 18%，平均選択数は約 0.9 個減少している。さらに、選択されたタグ候補のうちの半分以上が 1～5 位の候補であることから、適切なタグ候補がキーワードリストの上位に位置づけられるような重要度の計算が実現できたと考察する。しかし、それ以降の順位に関しては、それほど大きな差が見られない。ゆえに、6～20 位のタグ候補については、現状のアルゴリズムでは明確な差をつけるのは難しいと考えられる。

表 5 上位から 5 つずつ切り出したタグ候補の選択率と平均選択数

	1～5	6～10	11～15	16～20
選択率 (%)	28.8	11	9.8	5.4
平均選択数 (個)	1.44	0.55	0.49	0.27

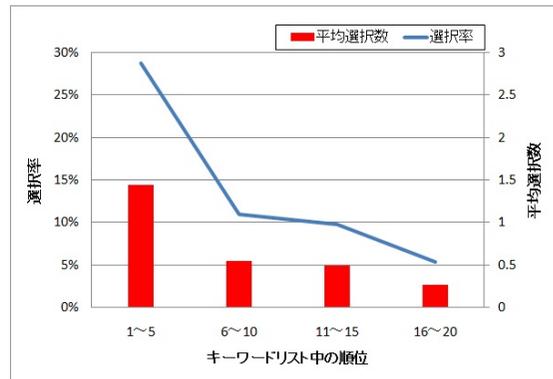


図 4 上位から 5 つずつ切り出したタグ候補の選択率と平均選択数

表 6 実験データ

	提案システム	API	関連タグ	全体
提示したタグ候補数	2000	1884	996	3934
選択されたタグ候補数	230	203	360	491

表 7 選択率および平均選択数

	提案システム	API	関連タグ	全体
選択率 (%)	11.5	10.77	36.14	12.48
平均選択数 (個)	2.3	2.03	3.6	4.91

4.2.2 実験 2

実験 1 と同様に、被験者 1 人あたり 10 件の記事の評価してもらったため、延べ 100 件の実験データを得た。提案システム、キーフレーズ抽出 Web API、関連タグ、全体の提示したタグ候補の合計および選択されたタグ候補の合計についてまとめたものを表 6 に示す。

10 名の被験者に対して提示したタグ候補の合計は 3934 個であり、そのうち 491 個の候補が選択された。ゆえに、タグ候補全体の選択率は 12.48% であり、1 つの記事あたりの平均選択数は 4.91 個となった。提案システムの計算結果により提示したタグ候補の合計は 2000 個であり、そのうち 230 個が選択されたため、選択率は 11.5%，平均選択数は 2.3 個となった。また、キーフレーズ抽出 Web API の出力結果により提示したタグ候補の合計は 1884 個であり、そのうち 203 個が選択されたため、選択率は 10.77%，平均選択数は 2.03 個となった。さらに、関連文書のタグとして提示したタグ候補の合計は 996 個であり、そのうち 360 個が選択されたため、選択率は 36.14%，平均選択数は 3.6 個となった。選択率および平均選択数をまとめたものを表 7 に示す。

関連タグの選択率および平均選択数の高さが目立つ結果となった。提案システムおよび API の結果と比較すると、選択率は 3 倍以上であり、平均選択数は約 1.3～1.6 個多く選択されている。

また、実験 1 の結果 (表 4) と比較すると、全体の選択率および平均選択数は増加しているが、提案システムと API については値が減少しているのが分かる。これは、関連タグがより適切であり、被験者はタグ候補を相対的に評価したため、提案システムと API によるタグが選ばれにくくなったものと考えられる。このことから、関連タグの重要性が見受けられる。

表 8 的中率

	提案システム	API	関連タグ
的中率 (%)	49.1	42.7	77.7

さらに、それぞれのタグ候補についての的中率をまとめたものを表 8 に示す。表を見ると、提案システムの的中率が API を上回っていることが分かる。つまり、人間の手でタグづけする感覚により近いタグ候補を提示できたと考えられる。また、ここでも関連タグの数値が極めて高いことが見て取れる。したがって、類似した文書が蓄積されていることが前提となるものの、関連タグの概念は非常に有用であることが分かった。

4.2.3 アンケート結果

10名の被験者に対して、提案システムの提示したタグ候補に対する満足度を5段階(5が満足, 1が不満足)で評価してもらった結果、平均の満足度は4.1となり、良い評価を得ることが出来た。

次に、本システムの「良かった点」「問題点」として挙げられた意見を以下に記す。

- 良かった点
 - 自分の考えたタグがほぼ全て含まれていた
 - 正式名称や略称など様々な形で提示されるため、選択の余地が広がる
- 問題点
 - タグの冗長性を感じた
 - 明らかに関係ない候補も含まれているため、もう少し厳選してほしい

5. ま と め

5.1 結 論

本研究では、文書中に存在しない語を含めたキーワード抽出、およびそれらをタグ候補として提示した際の妥当性について実験を行なった。実験の結果、文書に含まれる語のみを用いてタグ候補を提示するよりも、Webを利用して得た関連語を含めたタグ候補を提示するほうが選択率および平均選択数が向上することが確認できた。さらに、システムに蓄積された過去文書による関連タグを候補に追加することで、飛躍的に良い結果を得ることができた。

5.2 今後の課題

5.2.1 タグ候補の提示方法

実験1で示したように、作成したキーワードリストの順位が下がるにつれて、その選択率も減少していく。また、アンケートの回答として「提示されるタグ候補が多すぎると、返って見づらくなる」という意見をいただいた。したがって、タグ候補の提示数について考える必要があると感じた。

現時点でのこの問題の解消法としては、初めにキーワードリストの上位10個のみを表示し、必要であれば次の10個を表示、といったように必要に応じてタグ候補を追加提示していくインタフェースを導入する方法などが考えられる。

5.2.2 個人利用に特化したシステム設計

システムの使用目的の一つとして、個人的なメモ帳としての利用が考えられる。使用ユーザを1名に限定することで、「ユーザ固有の言い回し」や「頻繁につけられているタグ」などの情報をシステムに学習させることができると期待している。また、各種パラメータの設定機能などを実装することで、個人的な利用に特化したシステムの設計を目指す。

参 考 文 献

- 1) G. Salton, “ Automatic text processing: the transformation, analysis and retrieval of information by computer ”, Addison-Wesley, 1988 .
- 2) Wen-tau Yih, Joshua Goodman, Vitor R. Carvalho, “ Finding advertising keywords on web pages ”, Proceedings of the 15th international conference on World Wide Web, pp.213-222, 2006 .
- 3) 前田朗, “ キーワード自動抽出システム「言選 Web」”, 漢字文献情報処理研究 第6号, pp.124-133, 2005.10 .
- 4) Yahoo!デベロッパーネットワーク <http://developer.yahoo.co.jp/> .
- 5) 形態素解析エンジン MeCab <http://mecab.sourceforge.net/> .
- 6) Kazuho@Cybozu Labs — キーワード抽出モジュールを作ってみた <http://labs.cybozu.co.jp/blog/kazuho/archives/2006/04/summarize.php/> .