

文章中のコンテキストに適合した関連動画の検索

阿辺川 武^{1,a)} 間下 亜紀子¹

概要: 我々の開発している電子書籍閲覧システムは、本文中に出現する専門用語や固有名詞に対する百科事典の項目や、ページ内容と関連する各種リソースの項目をページの脚注部に表示する機能を有する。本稿では、システムの持つ機能を Wikipedia をリソースの例にして、書籍の本文テキストからキーワードの抽出、キーワードに複数の意味があるときにおこなう語義曖昧性解消についてそれぞれ説明する。次に、書籍の本文と関連する動画を検索し提示する機能について説明し、最後に正しく本文に関連した動画が検索できているかの評価実験をおこなった結果を報告する。

1. はじめに

近年、電子書籍専用端末、タブレット端末、スマートフォンなど電子書籍を読める端末の普及にともない、電子出版されるコンテンツの点数も急速に増加し、書籍を紙上ではなく電子デバイス上で読む人が増えてきた。電子化により書籍の購入後すぐにダウンロードして読むことができたり、複数の端末でしおりやコメントを共有できるなど、紙にはない利便性が得られるようになった。

しかし、読書の観点から見ると、書籍の電子化といっても従来の紙の書籍をそのままの形で電子化したものが多く、電子デバイスで読めることの他には、本文テキストが検索できるくらいしかメリットが得られていない。そこで、我々は読者に対し、電子書籍にしかできない電子書籍ならではの機能を提供することを考え、その1つの実例として書籍本文に対し外部リソースの情報を補足して表示する自動脚注付与機能つき電子書籍閲覧システムを開発している。

自動脚注付与機能は、本文テキストに対し、外部リソースの見出し語集合を照合させ、該当する見出し語ごとに、書籍の脚注(サイドノート)部に説明文と存在すればそのキーワードに関連する画像を表示する機能である(図1)。テキストからキーワードを抽出し、他のリソースへのリンクを生成する仕組みは、すでにインターネット上のニュースサイトをはじめとして多くのサイト実現されている。また、電子書籍の世界でも、登場人物や地名などといった固有名詞のリストをボタンひとつで表示できる Amazon の X-Ray for Books がすでに端末上で実用化されている。

これらの機能と我々の開発するシステムの大きな違いは、キーワードに付随する情報を本文画面と同時に表示するところにある。既存のシステムは読者がリンクやボタンをクリックし、明示的に補足情報を呼び出す必要がある。つまり読者が「わからない」「知りたい」といった願望を持った時にはじめて表示させるものに対し、我々のシステムでは、ある意味強制的に補足情報を提示する点が異なる。

従来の我々のシステムでは、脚注部に提示する外部リソースとして Wikipedia や市販の百科事典、そして本文テキストと最も関連度の高い文書を検索する連想検索技術 [1] を用いて関連書籍や関連観光情報を提示することを実現してきた。これらのリソースでは、テキストによる説明の他に写真やイラストの画像を持つ項目もあり、脚注部には画像を優先的に表示している。文字情報だけから構成される書籍の本文に対し、脚注部で関連する画像を表示することにより、本文の伝える内容を具体的なイメージで補足することが可能になった。

我々のシステムでは、静止画像のみを持つ外部リソースを活用してきたが、本稿では、本文と関連する動画を表示する手法を提案する。これにより、例えば本文中で動物名が出現した時にその動物が実際に動く姿を、また、事件・事故に関するキーワードが出現した時にそのニュース映像を即座に視聴することができるようになる。書籍はその物理的な制約から文字と画像のみに限定されていたが、電子化が進み、インターネットに多数の動画や画像が投稿される現在、これらのメディアをうまく結合し活用していくことが本システム開発の目的である。

以降、本稿では2節で我々の開発する自動脚注付与機能の詳細を説明し、3節で関連する動画を表示する手法を説明する。4節では、表示した動画が本文に適合しているか

¹ 国立情報学研究所
2-1-2 Hitotsubashi, Chiyoda, Tokyo 101-0062, Japan
^{a)} abekawa@nii.ac.jp



図 1 システムのスクリーンショット
Fig. 1 Screen shot of our system.

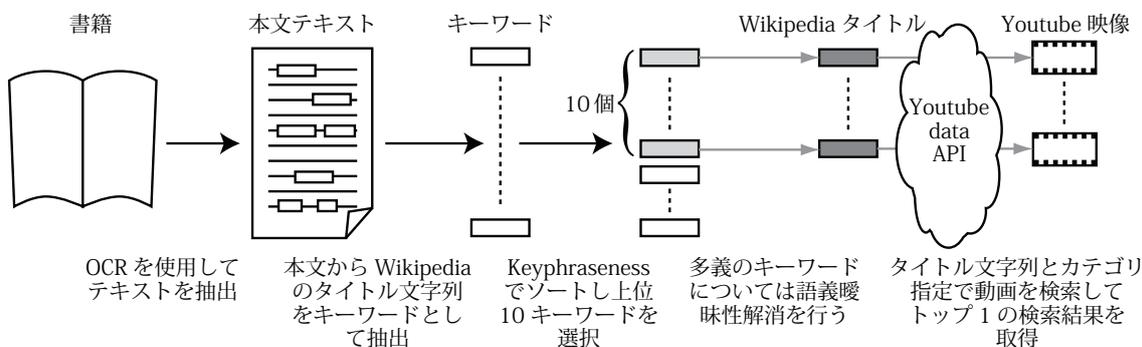


図 2 処理のながれ
Fig. 2 Flowchart of our system.

の評価実験をおこなう。最後に 5 節で本稿のまとめをおこなう。

2. 自動脚注付与機能

2.1 システムの概要

現在開発している自動脚注付与機能は、電子書籍閲覧システムの一部であり、現在閲覧している書籍のページに対し、辞書や百科事典などの外部リソースの見出し語を自動的に抽出し、その意味を脚注部に表示する機能である(図 1)。システムの動作は、最初に閲覧システムで表示されるページの本文を取得し、ページ本文中で外部リソースの見出し語が表出していれば、その見出し語と表示位置を抽出する。次に抽出されたすべての見出し語について、ページ上でハイライト表示し外部リソースへのリンクを付与する。このときリソースが Wikipedia であるときは、本文の文脈と照合し、複数の意味を持つ見出し語に対し、曖昧性解消の処理を施す。最後に見出し語をあらかじめ定義した重みによりランキングし、上位 10 個の見出し語について

は、ページ左右の脚注部に説明文と関連する画像があればそれを表示するものである(図 2)。以降でそれぞれの機能についての詳細を説明する。

2.2 キーワード抽出とランキング

キーワード抽出については、各リソースの見出し語集合を本文テキストと照合し、最長一致検索でマッチする文字列をキーワードとする。後述する動画表示では、検索クエリで使用するキーワードに Wikipedia のタイトル語を用いるため、本節では外部リソースに Wikipedia を用いた手法に限定して説明する。

マッチングに用いる表層文字列には、Wikipedia のページタイトルの他、リダイレクト元文字列と Wikipedia のパイプリンク文字列*1を用いる。パイプリンクから得られるリンク先の例を表 1 に示す。抽出したキーワードは書籍本文中の文字をハイライトとする形で表現される。

我々のシステムでは、重要なキーワードについては画面

*1 Wikipedia 内では、本文中で他ページにリンクを貼る際、{{ リンク先ページタイトル | 表示文字列 }} という記法を用いる。

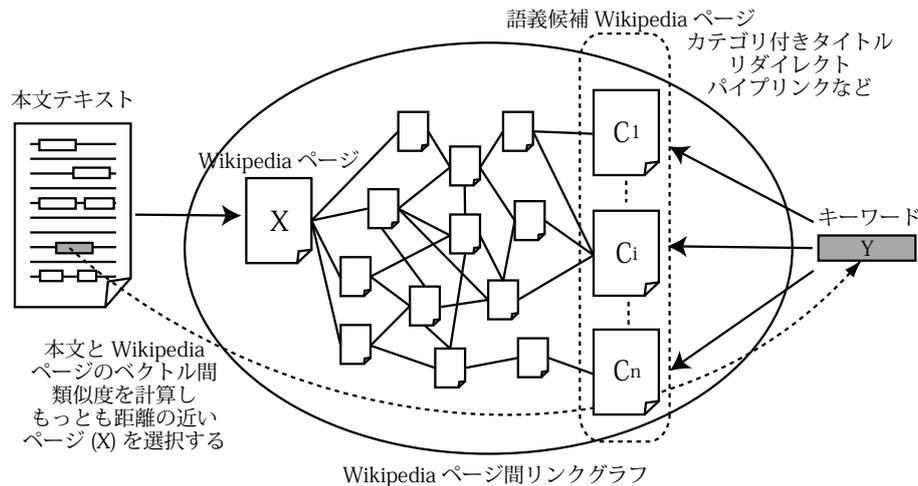


図 3 語義曖昧性解消のながれ

Fig. 3 Flow of word sense disambiguation.

表 1 パイプリンクから得られる表層文字列「オアシス」のリンク先
Table 1 Link pages of string “Oasis” or “Oasys” in Pipelink

Wikipedia ページタイトル	パイプリンク頻度
オアシス (バンド)	353
オアシス	264
オアシス (映画)	16
オアシス (アルバム)	11
オアシス (パチスロ)	6
いすゞ・オアシス	4
OASYS	2
ばらいろポップ	1
フローラルフォーム	1
オアシス (中川晃教のアルバム)	1

の脚注部にその説明と画像を表示するが、その数は、読者が一度に認知できる数および表示スペースの関係から 10 個と限定している。したがって本文ページから抽出されたキーワードが 10 個以上あるときは、なんらかの基準を用いてキーワードをランキングし選択する必要がある。本システムでは Wikipedia の見出し語集合を用いて抽出されたキーワードを、次に定義される *Keyphraseness* の値でランキングする [2]。

$$Keyphraseness = \frac{\text{キーワードがリンク付きで出現した総ページ数}}{\text{キーワードが出現した総ページ数}}$$

Keyphraseness は、Wikipedia の本文中でキーワードが出現したページのうち、他の Wikipedia ページへのリンクとなっているページの割合である。一般に人名・地名などリンク先が正しく指定できる固有名詞ほど値は高く、よく使用される一般名詞になると値は低くなる。本システムでは、*Keyphraseness* の値が 0.15 以下の表現は、脚注で表示すべきでない表現と考え、あらかじめ見出し語集合から除去しページ内のハイライトにも登場しない。

2.3 語義曖昧性解消

Wikipedia のページタイトルでは「オアシス (バンド)」「オアシス (映画)」とあるように、同じ文字列でも複数の概念が対応する場合には、括弧付きでどの分野であるかを指定することが多い。書籍の本文からキーワードを抽出した後、本文中でどの語義 (分野) として用いられるかを求める語義曖昧性解消の処理をおこなう必要がある。

本システムでの曖昧性解消処理は図 3 のようになる。あらかじめ Wikipedia の全ページについて内容語 (名詞、動詞、形容詞) を要素とするベクトルを作成しておき、現在表示している本文ページと最もベクトル間類似度の距離が近い Wikipedia ページを求め、これをページ X とする。

次に本文ページから抽出したキーワード Y に対し、キーワードが持つ語義候補ページ $C_i (i = 1..n)$ を取得する。そして Wikipedia 内のページ間リンクグラフを利用して、ページ X と C_i の最短距離 $Dist_{X \rightarrow C_i}$ を求める。ページ間リンクグラフは、ページを頂点とし、本文中で相互にリンクを持つページ間をエッジと定義する。なお本システムでは、オンデマンドによるリアルタイム処理を目標としているため、グラフ上での最短距離を求める際には、ランドマークの頂点を用いた近似解を求める手法 [3] を使用している。

最後に、候補 C_i の中で次に定義する $score_i$ が最も小さい候補ページを求める語義とする。

$$score_i = Dist_{X \rightarrow C_i} - \log(freq_i)$$

ここで $freq_i$ は、与えられた表層文字列に対してパイプリンク内のリンク先として出現した回数である。 X と C_i の距離だけでなくパイプリンク内の頻度を考慮した理由は、表層文字列だけから想起されやすい語義、つまり常識で思いつきやすい語義が、本文の文脈と語義の距離とともに曖昧性解消の精度に影響するということが経験的にわかったからである。

関連研究 [2] では、本文テキストと候補 Wikipedia ページ C_i とのベクトル間類似度を直接計算することで語義曖昧性解消を行なっているが、この手法では、本文が極端に短い場合 (最小 2 語) は、候補ページ C_i の多くが同じ類似度となり比較が困難になる。本手法では、本文が短くても関連ページ X を 1 つ求めるは容易で、これにより本文ページを Wikipedia ページリンクグラフの 1 頂点に対応づけることができ、曖昧性解消問題をリンクグラフ内での最短経路探索問題に還元することができる。

3. 文章に適合した動画の検索

本システムでは、キーワードを説明する Wikipedia の画像と説明文を脚注部に表示するが、この部分に動画のサムネイルを表示することも可能を可能にした。本節では、2 節で抽出したキーワードを用いて、文脈に適した動画を検索する手法を説明する。キーワードに基づいた動画検索では、動画のクローズドキャプションや投稿コメントの情報 [4], [5], [6], [7], 音声認識結果によるテキストを検索する手法があるが、これらの手法は、もともと大量の動画やコメントが手元にあることを前提としており、我々の環境では実現できない。そこで本システムでは、既存の動画検索サービスを利用して、キーワードをクエリとして適合する動画を動的に取得する。

本システムでは、動画検索サービスとして Youtube Data API を利用し、検索オプションとして、関連度順・日本語優先 ("orderby=relevance.lang.ja") を指定する。

3.1 カテゴリの指定

本システムでは、Wikipedia の見出し語を利用して本文から抽出したキーワードに 1 つにつき、それを説明する 1 つの動画を表示することを前提条件とする。したがって検索結果でトップ 1 位にランキングされた動画のみを提示することにする。多義のキーワードについては、キーワードが出現した文脈での語義を持つ映像を検索する必要がある。Youtube に投稿された動画の多くはカテゴリが付与されているので、本文の文脈から Youtube のカテゴリを推定できれば、求める語義に適合した動画が検索できる。例えばキーワード「オアシス」では、カテゴリ指定なしの場合は、ロックバンドのオアシスのライブ動画が検索結果のトップ 1 位に出現するが、「旅行とイベント」のカテゴリを指定すると砂漠のオアシスに関連する動画がトップ 1 位に出現する。キーワード「オアシス」について、いくつかのカテゴリを指定した検索結果を表 2 に示す。

文脈に応じて適切な Youtube カテゴリを指定できればよいが、Youtube では使用できるカテゴリが 14 種類と少ないため、常に最適なカテゴリを選択することができない。そこで本システムでは、Wikipedia のカテゴリ体系と比較的一貫性のある「音楽」「映画とアニメ」「ゲーム」「旅

行とイベント」の 4 カテゴリのみを扱うことにする。2.3 節でキーワードの語義曖昧性解消を行った後、該当する Wikipedia ページが「音楽」「映画」「アニメ」「ゲーム」といった Wikipedia カテゴリに分類されている場合、地名、地域、建造物といった「場所」に関連する Wikipedia カテゴリに分類されている場合、対応する Youtube カテゴリ「旅行とイベント」を検索時に指定する。例えば、場所に関連する「オアシス」を検索したい場合は「オアシス category:旅行とイベント」といった検索クエリを生成する。4 つのカテゴリのどれとも適合しないキーワードについては、4 カテゴリすべてを除外するクエリを生成する。

4. 評価実験

検索された動画が本文の文脈にどの程度適合しているか、そして語義曖昧性解消の処理やカテゴリ指定が検索結果の精度に影響しているかを調べるために、次の 4 通りの検索クエリについて評価実験をおこなった、

- a. 本文に表出するキーワード文字列のみ
ex. 「オアシス」
 - b. 曖昧性解消後の Wikipedia タイトル文字列
ex. 「オアシス (バンド)」
 - c. 表出するキーワード文字列 + カテゴリ指定
ex. 「オアシス category:-Music,-Film,-Games,-Travel」
 - d. 曖昧性解消後の Wikipedia タイトル + カテゴリ指定
ex. 「オアシス (バンド) category:Music」
- b. と d. については、曖昧性解消した後の Wikipedia ページタイトルをそのまま用いた。c. については、曖昧性解消をしていないことからカテゴリはわからないので、4 つのカテゴリを常に除外する指定とする。d. については、曖昧性解消後 4 つのカテゴリのどれかに当てはまるとき、そのカテゴリのみとする指定をする。

評価には、新書を中心に電子化された 32 冊の書籍を用意し、見開き 2 ページで 500 文字以上あるページを対象とした。実験に使用した Wikipedia 日本語版は 2012/10/7 にダンプされたデータを使用した。

4.1 評価手法

1 人の評価者による 5 段階の主観評価を行った。評価の前にあらかじめキーワードの属性ごとに評価基準を定義した。定義の例を図 3 に示す。Youtube では動画であっても表示は静止画で音楽だけが流れるといったものがあるため、そのような動画は静止画として分類している。

4.2 評価結果

4 つの検索クエリによって検索された動画の評価結果について、評価の平均値と各評価値ごとの内訳を表 4 に示す。

表 2 キーワード「オアシス」の検索結果
Table 2 Search results of the keyword "Oasis".

カテゴリ指定	動画タイトル	再生回数
なし	Don't Look Back In Angeroasis LIVE 和訳付き with Lyrics	2,291,136
旅行とイベント	砂漠のオアシス「パームスプリングス」に忍び寄る危機とは？	1,159
映画とアニメ	映画「東京オアシス」予告編 小林聡美 原田知世 もたいまさこ	33,588
ペットと動物	シベリアンハスキー 散歩途中のオアシス	1,680
ゲーム	ジョジョの奇妙な冒険 黄金の旋風 第10話 グリーンディとオアシス 後編	271,041
-音楽	oasis - Whatever (カテゴリ:ブログ)	1,219,765

表 3 属性ごとの評価基準の例

Table 3 Example of evaluation criterion of an attribute.

人物・グループ
5. その人物、グループ全員の映像 (動画)
4. その人物、グループが映像の一部に映っている、もしくは説明映像 (動画・静止画)
3. その人物、グループが関わっている映像 (動画・静止画)
2. その人物、グループに関連する映像はあっても関係のない字幕や音声が入っている映像 (動画・静止画)
1. 関連のない映像 (動画・静止画)
場所・建物
5. その場所、建物がメインの映像 (動画)
4. その場所、建物のごく一部、もしくは広範囲の映像 (動画・静止画)
3. 地図などでその場所、建物の位置を示す映像 (動画・静止画)
2. その場所、建物が映像の一部に映っている映像 (動画・静止画)
1. 関連のない映像 (動画・静止画)

表 4 評価結果

Table 4 Evaluation results.

検索クエリ	平均値	5	4	3	2	1
a. キーワード	2.473	44	36	37	31	116
b. 曖昧性解消後タイトル	2.483	44	35	36	34	112
c. キーワード +カテゴリ指定	2.612	48	40	40	32	103
d. 曖昧性解消後タイトル +カテゴリ指定	2.637	48	39	41	38	96

4.3 考察

評価実験の結果、平均値は2.4~2.6と低く、その原因として、本文の文脈とは関連のない映像が多く表示されてしまった事があげられる。動画に付与されたカテゴリは、投稿者が自らタグ付けを行うため適切なカテゴリが付与されているとは限らず、検索クエリで音楽カテゴリを除外した場合でも、関連のない音楽映像の多くが検索結果のトップに残っていた。そのためカテゴリだけに依存した検索クエリではなく、キーワードとともに具体的な分野を表す語を付加する必要があることがわかった。

国名や人種、大規模組織など、広範囲の物や人、概念を表す言葉は評価自体が難しく、カテゴリ指定の有無でもほ

とんど検索結果に差が現れなかった。また、事件や事故などの時事情報に関連した言葉は、関連映像が多く表示されながらも直近に話題になった出来事に大きく偏る傾向にあった。例えば、セシウムという言葉では物理的な性質や構造についての説明映像がより関連が強いと思われるが、Youtubeでは福島第一原子力発電所の事故に関連した映像がほとんどであった。

評価をする中で、映像での説明が有効な言葉とそうでない言葉も分かれ「前置詞」や「接続詞」など語学に関連するキーワードでは講義映像がヒットすることが多く、その言葉について詳細な説明がなされていたが、物として存在しない言葉を説明しているので一瞥しただけでは関連している映像かどうかの判断が難しい。一方、動物や植物、電車などはそれぞれが物として存在しているため、短い時間での視聴でもキーワードに関連した映像かどうかの判断がしやすく、本文の文脈と関連度の高い映像も多かった。したがって、一般名詞や抽象名詞などは動画での補足説明には向かず、最初から表示候補として採用しないといった対策が必要であることがわかった。

本システムでは、ウォーターフォール的に多くの処理をおこなっているため、前段階の誤りがそのまま次の段階に伝搬する。最後に誤りの原因となる処理を箇条書きで挙げる。

- (1) スキャンしたページのOCR認識
- (2) Wikipedia タイトル文字列の抽出
- (3) 語義曖昧性解消
 - i. 文脈と近い Wikipedia ページの選択
 - ii. ページ間リンクグラフを使った概念の近さの測定
- (4) 動画検索
 - i. 誤ったカテゴリ付与
 - ii. 検索条件の指定
 - iii. 文脈に応じた動画が存在しない

5. おわりに

本稿では、我々が開発している電子書籍閲覧システムの特徴である自動脚注付与機能について説明をおこない、関連動画の検索・表示手法について説明した。実際の書籍を用いた評価実験の結果、本文文脈と検索された動画の関連

度が5段階評価で約2.6という値となった。今後の課題として、誤りを含む可能性のあるテキストのOCR認識から動画検索までの各処理の個別の精度を求めることが挙げられる。そして本稿の実験では、ページから抽出したキーワードをランキングし、トップ10個のキーワードについて動画を検索し評価を行ったが、そもそも読者は書籍本文を読んでいる際に、どのような動画を望んでいるのかについて検討する必要がある。

参考文献

- [1] 西岡真吾：汎用連想計算エンジン GETA, コンピュータソフトウェア, Vol. 26, No. 4, pp. 87-106 (2009).
- [2] Mihalcea, R. and Csomai, A.: Wikify!: linking documents to encyclopedic knowledge, *The 18th ACM Conference on Information and Knowledge Management*, pp. 233-242 (2007).
- [3] Tretyakov, K., Armas-cervantes, A., Garca-baueles, L. and Dumas, M.: Fast Fully Dynamic Landmark-based Estimation of Shortest Path Distances in Very Large Graphs Categories and Subject Descriptors, *ACM Conference on Information and Knowledge Management*, pp. 1785-1794 (2011).
- [4] 増田智樹, 山本大介, 大平茂輝, 長尾確: オンラインアノテーションを利用したビデオシーン検索, 人工知能学会第21回全国大会 (2007).
- [5] 中村聡史, 田中克己: ソーシャルアノテーションに基づく動画検索手法, *DEIM Forum 2009* (2009).
- [6] 若宮翔子, 北山大輔, 角谷和俊: 投稿ユーザコメントとWeb ページ文脈を用いた共有動画シーン検索方式, *DEIM Forum 2010* (2010).
- [7] 江端佑介, 川村秀憲, 鈴木恵二: ユーザコメントの tf-idf 法による分析を用いたインタラクティブな関連動画の提示, 電子情報通信学会技術研究報告. AI, 人工知能と知識処理 109(439), pp. 7-10 (2010).