「電子出版インデックス情報データベース」

仕様書案

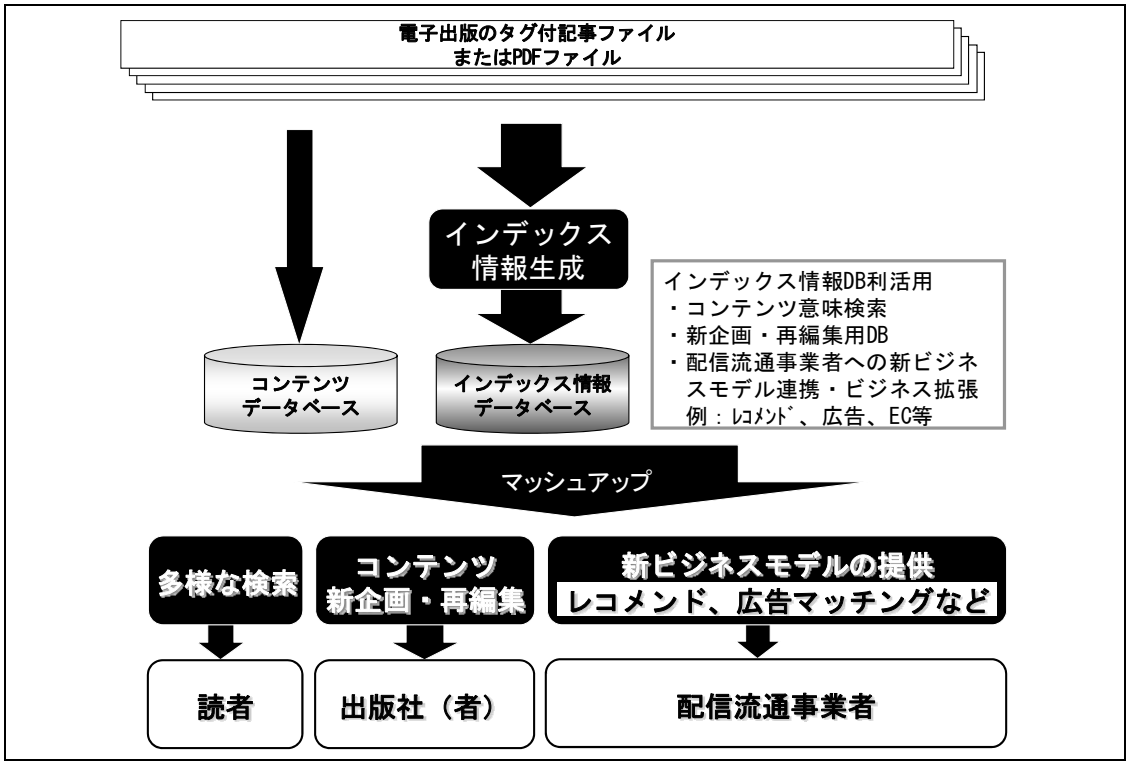
2011年5月版

社団法人日本雑誌協会

1　システム概要

インデックス情報データベースとは、電子出版コンテンツ流通管理コード（仮）体系に基づいて格納されたコンテンツに対して、読者（国民）の検索容易性・本文到達性向上を図ることを目的としたデータベースである。

コンテンツは、現状は配信フォーマットで流通されているが、それだけでは、そのコンテンツ記事のもつ意図を機械的に判断することが難しい。したがって、コンテンツに対して予め構造解析を施し、そのコンテンツの持つ意図を反映したインデックス情報（目次タイトル、ジャンル情報、記事から抽出したキーワードなど）を付与することで、読者（国民）に対して品質の高いコンテンツ記事への到達手段を提供することが出来ると考えている。



資料1　システム概念図（例）

2　システム構成

(1)　全体概要

インデックス情報データベースは、大きく 2 つの機能から構成される。

まず 1 つ目は、インデックス情報解析機能である。この機能は、コンテンツ提供者が提供するタグ付記事をもとに構造解析を行い記事の意図や希少性を反映したキーワードを抽出しインデックス情報データベースに蓄える。

具体的には、以下の処理を行う事になる。

1)　言語解析

・記事の持つテキスト情報から単語やその単語が属するカテゴリを抽出する。

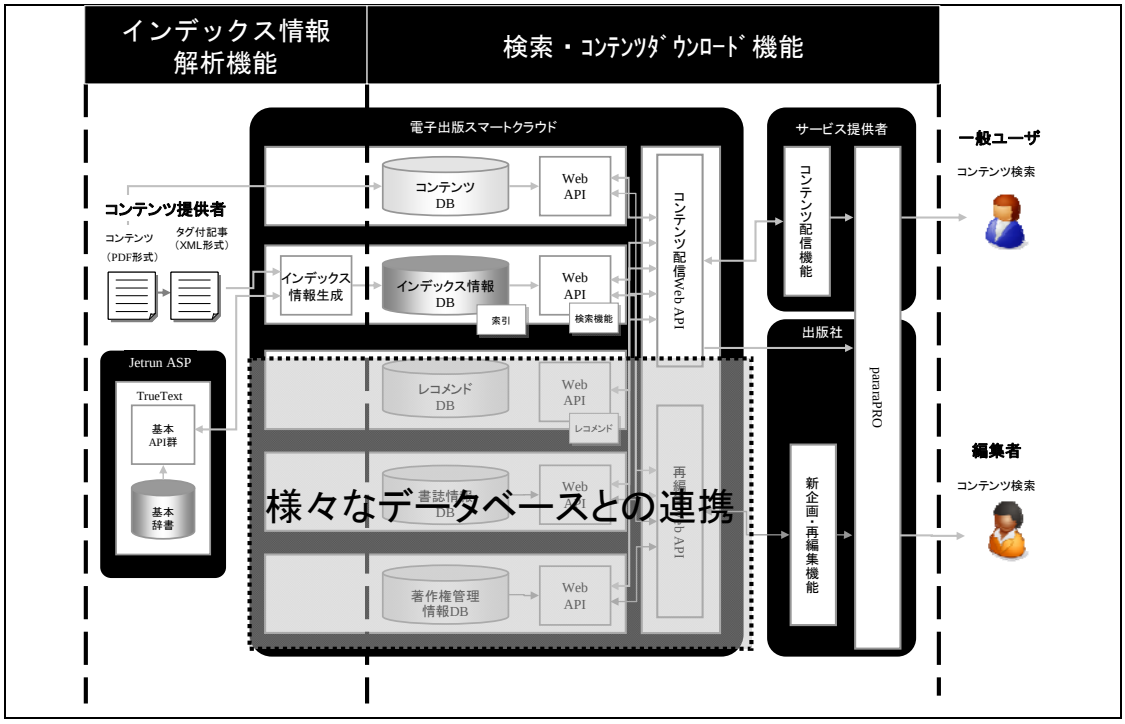
2)　意味解析

・タグ付記事（タイトル、サブタイトル、リード、大見出し、キャプション、本文など）に現れるすべての単語とカテゴリの占有率を取得する。

・抽出した全単語を出現位置や頻度を考慮して並び替える。

・記事の特徴を表す単語を生成する。

2 つ目は、検索・コンテンツダウンロード機能である。この機能は、構造解析されインデックス情報データベースに蓄積されたマイクロコンテンツ単位での記事の全文検索と、目次・記事単位の持つ意図を反映した意味情報による検索機能を実現する。これにより編集者及び一般ユーザーが一定のレベルで求めている記事にリーチすることを目指している。



資料 2　システム構成図

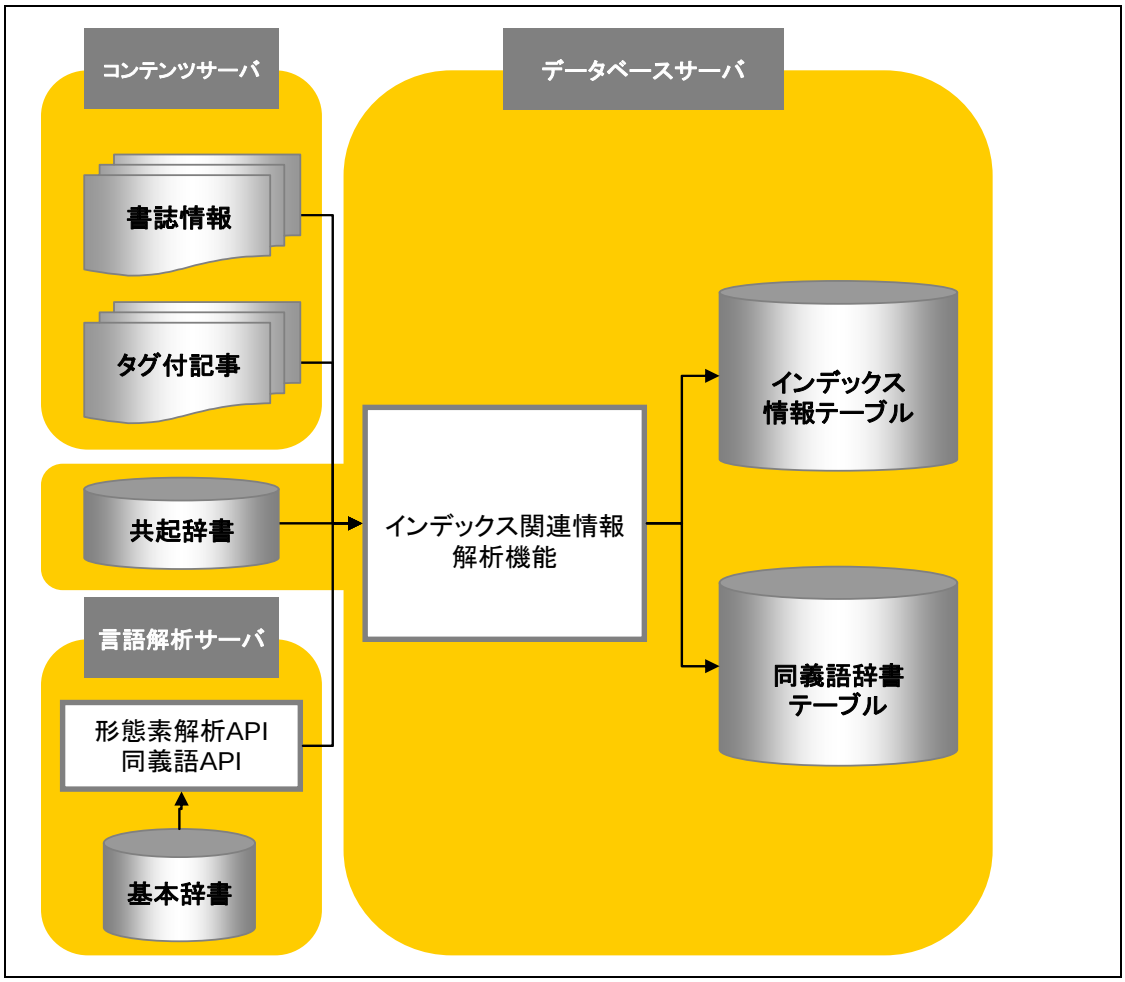
(2) 機能概要

1) インデックス情報解析機能

・インデックス情報解析機能は、コンテンツ提供者が提供する書誌情報とタグ付記事を、記事の意図や希少性を反映したキーワードを付与したうえでインデックス情報テーブルを作成する。記事の文章を解析するに当たり、形態素解析 API を利用し、キーワードを付与するために共起辞書テーブルを参照する。

・同義語 API を使用して同義語辞書テーブルを作成する。

上記 2 つの作成されたテーブルにより記事の検索を実現し、検索容易性と記事到達性の向上を図る。



資料 3 「インデックス情報解析機能」概要図

インデックス情報テーブルの内容は書誌情報と記事内容、またタグ付記事からなる。書誌情報と記事内容はコンテンツの書誌情報と記事の内容がそのままの形で入力される。タグ付記事は記事の内容をもとに抽出した名詞や、記事に関連する名詞など、記事の検索の利便性を向上させるための付加的情報を保持する。

また同義語辞書テーブルを作成し、検索時の表記ゆれを吸収する。

2) 検索機能

検索機能は、インデックス情報解析機能により作成されたインデックス情報データベース（書誌情報・記事情報・タグ付記事・同義語辞書・他）から、目的の記事の検索をする。

検索は、インデックス情報データベースコンセプトガイドラインより、インデックス情報解析機能を利用して生成されたインデックス情報データベースの有効性を評価ならびに出版社（者）・流通業者・国民が必要な記事を選択する場合に共通で必要となる機能となる。

検索機能の実現において、インデックス情報データベース内に蓄積されるデータ量が大量であること、そして記事に対して検索の対象とするフィールドが沢山あるため SQL による問合せが非効率であることから検索エンジンのパッケージを利用している。検索エンジンは、株式会社ゼロスタートコミュニケーションズの“ZERO-ZONE Search”を利用している。

インタフェースは、電子出版スマートクラウド・コンセプトガイドラインの将来像を想定して Web API 連携を想定している。Web API へリクエストを行うコール元は、出版社（者）・流通業者・読者（国民）を想定している。国民においては、出版社（者）・流通業者を経由して行われることを想定している。

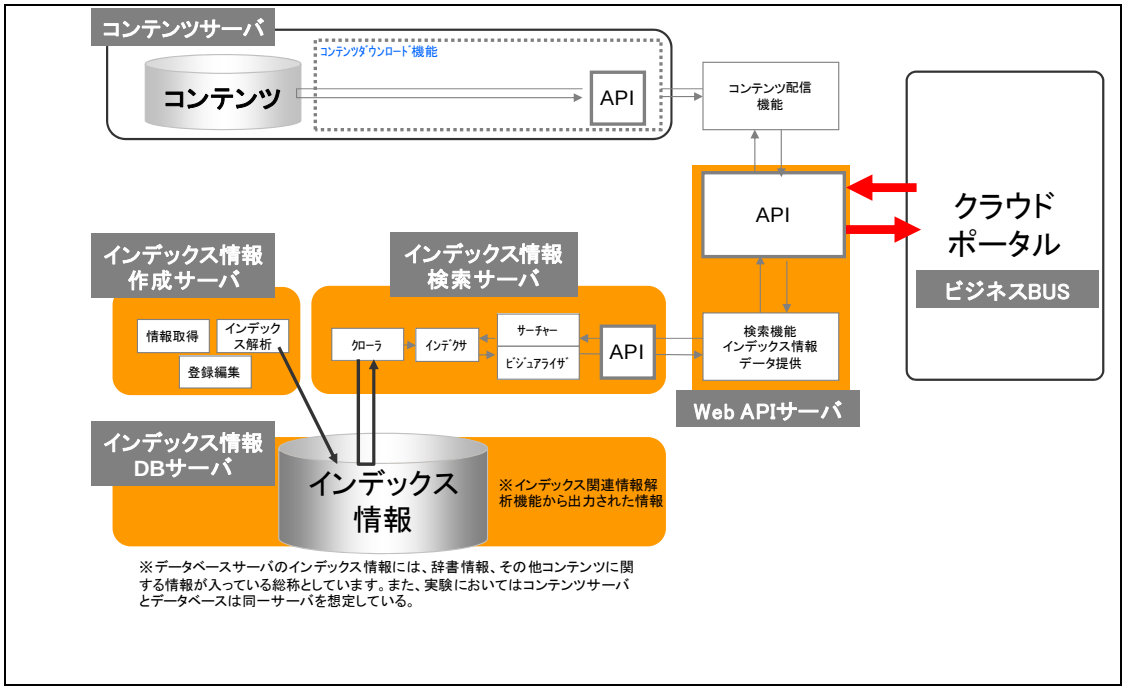
検索として必要な機能について、出版社（者）・流通業者・読者（国民）を想定した場合に個々に記事の検索目的が異なることが考えられる。検索目的がことなることから、検索目的に応じてインデックス情報データベース内の検索対象のカスタマイズ、属性情報による絞り込みが自由に選択可能にすることで個々の要求に対応できるようにする。また、検索結果の表示順序についても個々により検索目的がことなることから重要視する項目と検索キーワードのマッチ度が高いものを上位に表示する重み付け機能を有する。

3) コンテンツダウンロード機能

コンテンツダウンロード機能は、電子出版コンテンツ流通管理コードを包含した書誌情報と連動して電子出版コンテンツ流通管理コードにより目的の記事データのダウンロードをする。電子出版コンテンツ流通管理コードによって紐付き記事の配信が可能であることを確認するために実装されている。

インタフェースは、前述の検索機能と同様に Web API 連携を想定している。Web API へリクエストを行うコール元は、出版社（者）・流通業者を想定している。国民においては、記事データは、最終的に出版社（者）・流通業者を介して必要な著作権保護処置が施された上で配信されることを想定している。

検索機能およびコンテンツダウンロード機能のフローについて、電子出版クラウドを踏まて、各社が共通クラウドプラットフォーム上で機能連携が行えるように API 形式の選択を行い以下の構成（資料 4）を作成した。



資料 4　「検索・コンテンツダウンロード機能」フロー概要図

今回は、クラウド環境を想定して検索とコンテンツダウンロードの 2 つの機能をそれぞれ API 化しているそれぞれの API 機能は個別のクラウドサーバであることを想定している。

・インデックス情報検索サーバ
　検索機能 API の実装
・コンテンツサーバ
　コンテンツダウンロード機能の API 実装

インデックス情報作成サーバは、前項のインデックス情報解析機能にあたる。生成されたインデックス情報はインデックス情報作成サーバより直接インデックス情報 DB に出力される。

インデックス情報の検索ならびにコンテンツダウンロードをクラウドポータルが要求する場合に、各要求を取りまとめて必要な機能 API へ振り分けるために Web API サーバを設けている。これにより要求先を 1 箇所でまとめて受け付け要求内容に応じて必要なクラウド上の API に対して要求を振り分ける。

・Web API サーバ
　①クラウドポータルへの最初の機能要求を受付
　②機能要求により要求を振り分け

インデックス情報検索サーバは、インデックス情報データベース上の記事データを検索エンジンのクローラが取り込み行う。

検索機能は、クローラにより取り込まれたインデックス情報を検索エンジンのインデクサにより検索用のインデックスファイルが生成される。

Web API への検索要求に対してインデックス情報検索サーバの API へ要求を渡し、インデックス情報検索サーバのサーチャーによりインデックス内から検索要求に該当するデータを抽出し要求元へ検索結果を回答する。

コンテンツダウンロード機能では、Web API へのコンテンツダウンロード要求に対してコンテンツサーバの API へ要求を渡し、要求記事のデータを要求元へ送信する。

3　機能説明

(1)　インデックス情報解析機能

インデックス情報解析機能は、基本的な全文検索を実現するための書誌情報と記事の格納、記事を解析し名用を反映したキーワードを付与する意味解析と、検索時の表記ゆれを吸収するための同義語辞書作成の３段階に分けられる。

1)　検索対象である書誌情報と記事の格納

タグ付記事の各雑誌標準タグからテキストやコンテンツ管理コードを抽出する。

書誌情報と抽出したテキストをコンテンツ流通管理コードで紐付けてインデックス情報テーブルの書誌情報部分と記事情報部分に格納する。

2)　インデックス解析

タグ付記事から抽出したテキストを解析し、抽出ワード（テキストに現れる名詞）と連想ワード（記事に現れないが記事に関連する名詞）を取得し、インデックス情報テーブルのインデックス情報部分に格納する。連想ワードの付与により記事中には現れない名詞で検索を行ったときでも記事に到達できるようになる。

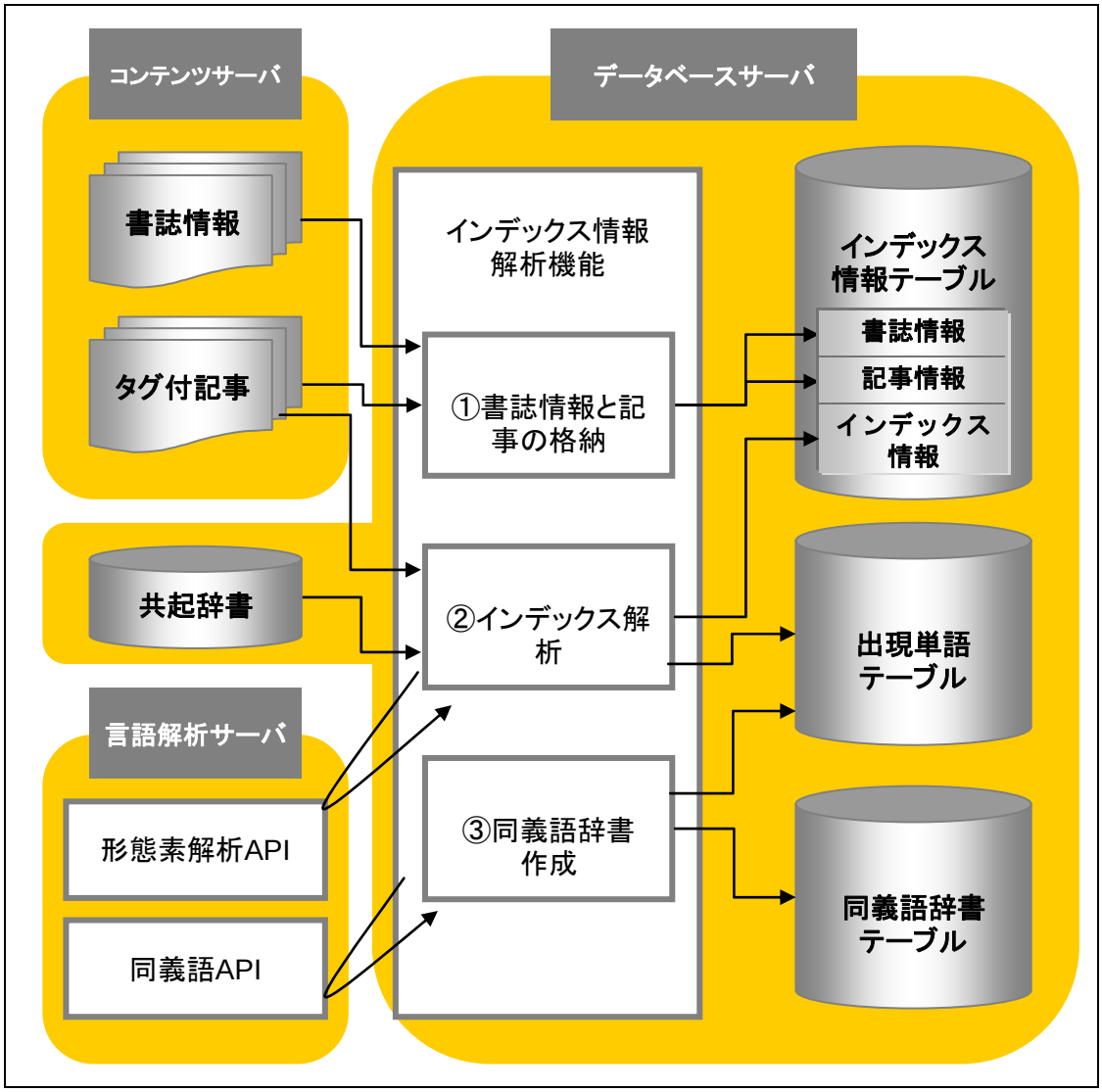
より具体的には、記事に対して以下のような操作を行う。

記事から抽出したテキストに対し形態素解析を施し、名詞とその名詞が属するカテゴリを抽出する。抽出した名詞とカテゴリを集計し重要な名詞を優先して並び替えインデックス情報テーブルの「目次・抽出ワード」項目として格納する。「目次・抽出ワード」をもとに後述の共起辞書を使って、記事中には現れないが記事に関連する名詞を探し出し、インデックス情報テーブルの「目次・関連ワード」項目に格納する。

また次の同義語辞書作成のために出現した名詞を出現単語テーブルに記録する。

3)　同義語辞書の作成

出現単語テーブルに新規に追加された名詞に対して同義語を探し出し同義語テーブルに格納する。



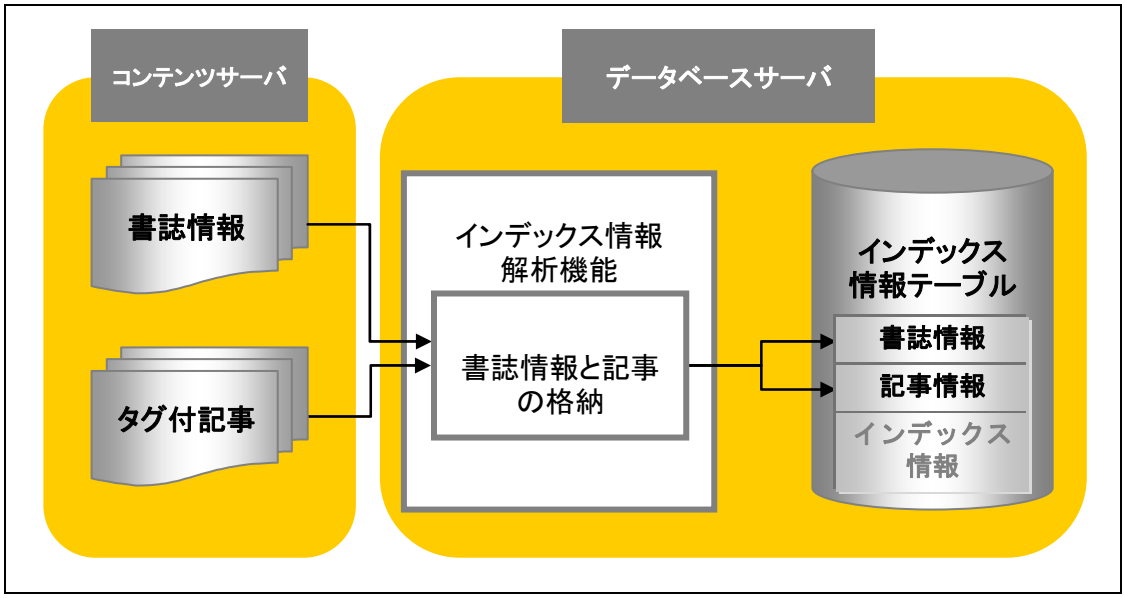
資料 5　「インデックス情報解析機能」入出力関連図

以下、3 段階のそれぞれについて機能を詳解する。

4)　書誌情報と記事情報の格納

■概要

書誌情報とタグ付記事をコンテンツ流通管理コードで紐付けて、インデックス情報テーブルに格納する。格納した内容は基本的な全文検索の機能を実現するための対象となる。



資料 6　「書誌情報と記事の格納」機能概要図

■詳細

①タグ付記事に含まれる雑誌標準タグの内容を取得する。

②単一記事に同じ雑誌標準タグが複数含まれる場合は、その内容を区切り文字で区切って連結する。

例：caption タグが”<caption>例 1</caption>”、”<caption>例 2</caption>”のように含まれる場合、インデックス情報データベースの項目「記事・キャプション」に ”例 1（改行）　（改行）　（改行）例 2” と入力する。

③記事の「コンテンツ流通管理コードタグ」にあるコンテンツ流通管理コードと一致するコンテンツ流通管理コードを持つ書誌情報を探し、紐付けた内容をインデックス情報データベースに格納する。

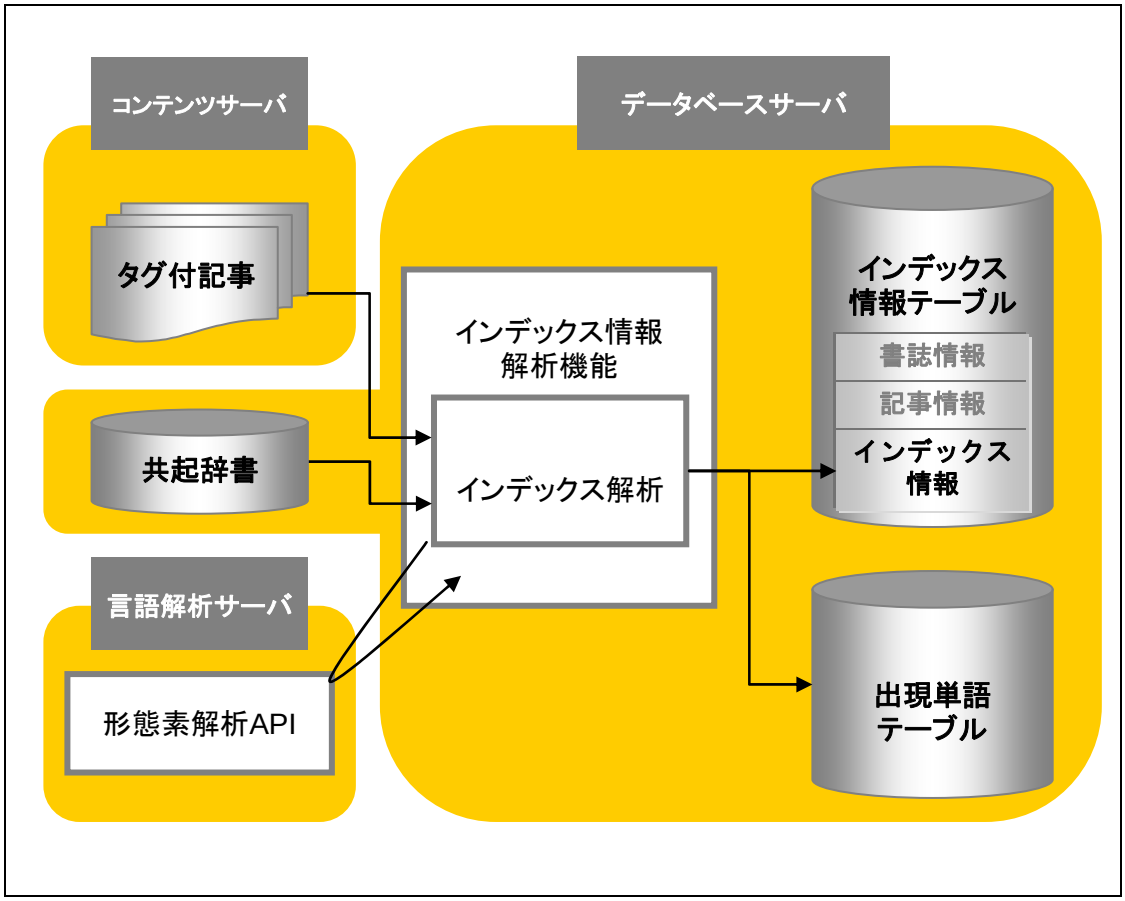
■チューニングポイント

① 本方式以上に詳細な文書構造を保ったままデータを格納する方法も存在しうる。詳細な文書構造を保持した検索が可能ならば検討すべきだろう。

5)　インデックス解析

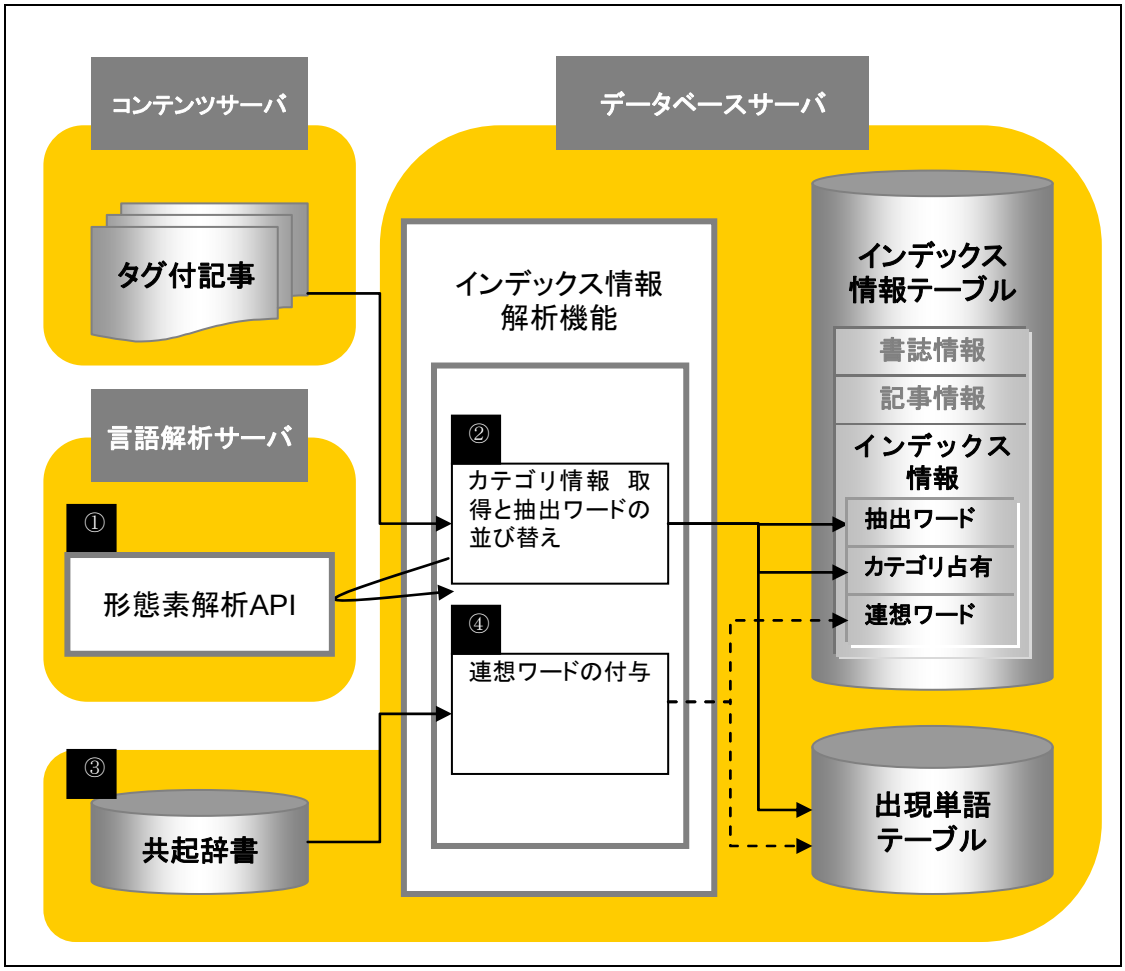
インデックス情報関連テーブルのインデックス情報部分への入力を行う。記事に現れる名詞を抽出ワードとして、また記事に現れないが記事に関連する名詞を探し出し連想ワードとして格納する。この抽出ワードと連想ワードを利用することで、基本的な全文検索に検索の容易性と記事到達性の向上を図る。

基盤技術として形態素解析があり、参照データとして共起辞書を用いる。



資料 7　「インデックス解析」機能概要図

インデックス解析は、形態素解析をもとにした記事に現れる名詞の取得と並び替え、共起辞書を参照した連想ワードの取得に分かれる。それぞれ前提と実際の解析について詳解する。



資料 8　「インデックス解析」データ入出力

■形態素解析

〇機能概要

形態素解析は自然言語をコンピューターで解析する際の基礎技術のひとつである。

特定の言語の文法と辞書を組み合わせて利用し、文章を形態素（その言語で意味をもつ最小単位）に分解し各々の品詞を判別する。

例として「東京へ行ってきた」という文章の解析結果を挙げる。

形態素	品詞	読み	活用の種類
東京	名詞	トウキョウ	
へ	助詞-格助詞	エ	
行っ	動詞-自立	イッ	行く・五段活用
て	助詞-接続助詞	テ	
き	動詞-非自立	キ	来る・か行変格活用
た	助動詞	タ	

資料 9　形態素解析の例「東京へ行ってきた」

現在では確率的言語モデルを利用したアルゴリズムが主流である。形態素同士のつながり方を隠れマルコフモデルを使って統計的にモデル化する。形態素の出現確率からその形態素の生起コスト、形態素同士のつながり方については連結コストとして数値化しておく。

文章の解析では以下のようなアルゴリズムでもっともらしい形態素単位の分かち書きを行う。

①形態素単位の分割方法を網羅する。

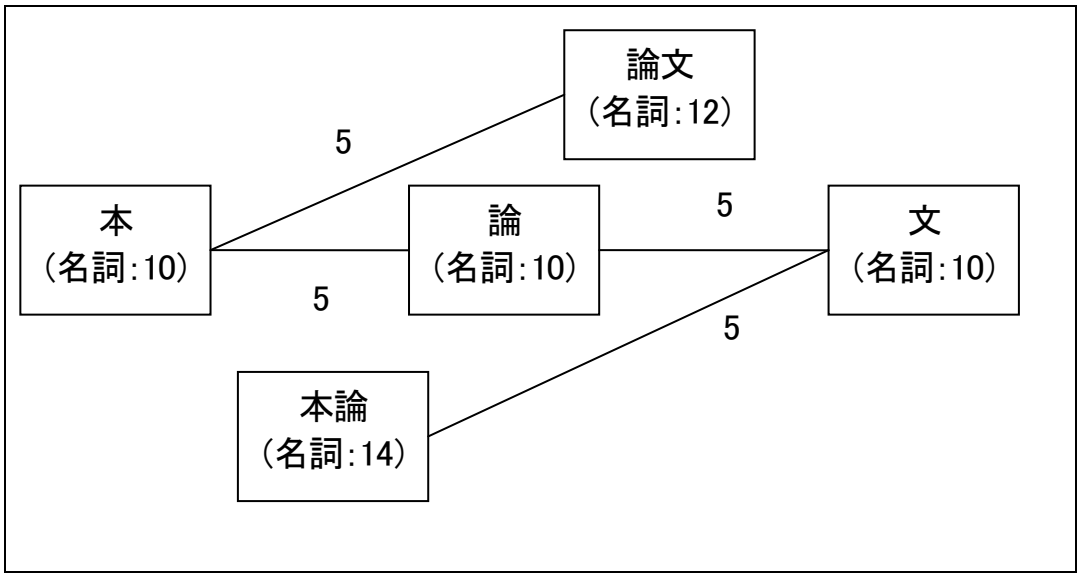
②各形態素の生起コストと連結コストをもとに、それぞれの分割方法について文章全体での総コストを求める。

③この総コストが最も低い分割方法を正とする。

具体的な解析例をあげる。

・例１「本論文」

形態素単位の分割パターンを網羅する。



資料 10　形態素解析詳細のサンプル「本論文」の各コスト

それぞれの分割パターンについて形態素の生起コスト、形態素同士の連結コストを合計する。

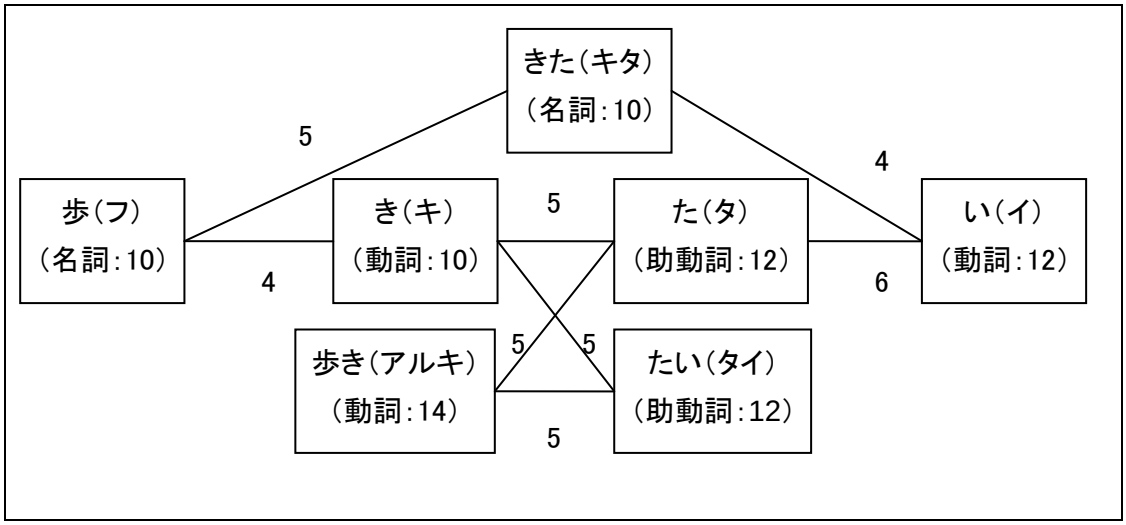
分かち書き	総コスト	総コストの昇順
本 - 論 - 文	10 + 5 + 10 + 5 + 10 = 40	3
本 - 論文	10 + 5 + 12 = 27	1
本論 - 文	14 + 5 + 10 = 29	2

資料 11　形態素解析詳細のサンプル「本論文」の総コスト

総コストが最も小さい「本 - 論文」が形態素解析の結果となる。

・例２「歩きたい」

形態素単位の分割パターンを網羅する。



資料 12　形態素解析詳細のサンプル「歩きたい」の各コスト

それぞれの分割パターンについて形態素の生起コスト、形態素同士の連結コストを合計する。

分かち書き	総コスト	総コストの昇順
歩 - きた - い	10 + 5 + 10 + 4 + 12 = 41	2
歩 - き - た - い	10 + 4 + 10 + 4 + 12 + 6 + 12 = 58	5
歩 - き - たい	10 + 4 + 10 + 5 + 12 = 41	2
歩き - た - い	14 + 5 + 12 + 6 + 12 = 49	4
歩き - たい	14 + 5 + 12 = 31	1

資料 13　形態素解析詳細のサンプル「歩きたい」の総コスト

総コストが最も小さい「歩き - たい」が形態素解析の結果となる。

〇既知の未解決問題

確率的言語モデルの前提からいくつかの課題が発生する。

①未知語、新語の解析

辞書には登録されていないが、人が単語と認識できる文字列を未知語という。辞書に登録されている単語で文章を区切るため、未知語の解析は規定されない。新語は当然、この未知語となるため、対応方式の検討が必要である。新語への追従するように辞書を拡充していくか、ある程度一般性を持った単語のみを登録するか２通りの方式が考えられる。

特徴のある新語は全文検索で容易に検出できることが予想されるため、検索容易性・本文到達可能性の向上の観点からは新語対応の即時性は不要と考えられる。ただし新語が定着し辞書に登録された後に記事を再解析することは効果的だろう。

②複数の形態素解析結果を持つ文章について

文章の区切り方によって複数の形態素解析結果が考えられる場合に、前後の文脈に依存して適切なものを選ぶことは難しい。

例えば「最高値」は「最高 - 値（さいこうち）」「最 - 高値（さいたかね）」のどちらにも分かちうるが、前後の文脈に依存して適切なものを選び形態素解析の結果として返却することは非常に難しい。ひらがなを多く含む文章ではさらに切り分けが曖昧になることが多く、文脈に即した形態素解析結果が得られない。

形態素解析の結果として複数の候補を残し、大域的な判断を加える方式が考えられるが、判断基準が膨大になるため実現は難しいと考えられる。

■カテゴリ情報取得と抽出ワードの並び替え

○機能概要

記事の特徴を記述することで、単独の記事の検索性が向上し、また似た別の記事を探すことにも役立つと考えられる。

すべての単語を分類するような「カテゴリ」を定め、記事に現れる単語に紐づく「カテゴリ」を集計し、「カテゴリ」の記事における占有率を特徴量として利用する。また記事に現れる単語を重要性の高い順に現れるように並び替えて記録する。

○機能詳細

記事の各標準タグの内容に対して形態素解析を施し形態素に分解する。以降、形態素の表記を「表層」と呼ぶ。「表層」に対して表記ゆれの標準を定めた「原型」、属性を現す「カテゴリ」を取得する。なお「カテゴリ」は名詞にだけ定め、一般性が高い名詞は「カテゴリ」を持たない。また名詞以外の品詞には「カテゴリ」を与えていない。

「表層」、「原型」、「カテゴリ」の３つ組の例をあげる。

表層	原型	カテゴリ
プリンター	プリンター	プリンター、スキャナー、印刷機
プリンタ	プリンター	プリンター、スキャナー、印刷機
東京	東京	東京都
とうきょう	東京	東京都
Tokyo	東京	東京都

資料 14　表記ゆれのサンプル

「プリンター」は「プリンター」と「プリンタ」の表記を持ち、どちらの表記も辞書に登録され「原型」は「プリンター」に紐づく。

形態素解析で出現した名詞に紐づく「カテゴリ」のそれぞれについて出現回数を雑誌標準タグに依存して重みをつけて数えあげ重みつきの単語数で割ることで、それぞれの「カテゴリ」の記事における占有率

を求める。

以下のように雑誌標準タグの重みを設定する。

タグ名	タグ	重み
雑誌名	magazine	0 （数え上げの対象外）
コンテンツ管理コード	c_code	0 （数え上げの対象外）
特集名	tokushu	5
記事名	kiji	4
タイトル	title	4
サブタイトル	subtitle	3
リード	lead	2
大見出し	omidashi	2
見出し	midashi	2
本文	honmon	1
キャプション	caption	1
クレジット	credit	1
その他	etc	1

資料 15　雑誌標準タグの重み

上記の重みつき占有率の上位から 3 つの「カテゴリ」をとりインデックス情報テーブル（資料 27　インデックス情報テーブルレイアウト）の項目 28（目次・カテゴリ 1）から項目 30（目次・カテゴリ 3）へ入力する。またそれぞれの重みつき占有率をパーセンテージで同テーブル項目 31（目次・カテゴリ占有率 1）から項目 33（目次・カテゴリ占有率 3）へ入力する。

またこの重みつき占有率で「カテゴリ」を並び替え、紐付く「表層」「原型」「カテゴリ」「重みつき占有率」の 4 つ組みを半角コロン区切りにし、この 4 つ組みをカンマ区切りで連結したものを、インデックス情報テーブルの項目 26（目次・抽出ワード）に入力する。

〇チューニングポイント

①「カテゴリ」の策定

「カテゴリ」を人の手で作成し単語を振り分ける方式と、一定の文書群から統計的手法を使って機械的に「カテゴリ」を作成し単語を振り分ける方式がある。

人の手で「カテゴリ」を作成する場合は、人が理解しやすい判断基準で「カテゴリ」の名称と単語の振り分け方針を定めることができる。このため特定の単語がどのカテゴリに属するか「カテゴリ」の名称を使って理解できる。ただし、運用時には「カテゴリ」の追加、削除、変更についても方式を定め、継続的に人手で管理を行う必要がある。

一方、統計的手法で「カテゴリ」を定める場合は、機械的に「カテゴリ」を作成し単語を振り分ける、運用でも人手をあまり必要としない。ただし、作成された「カテゴリ」の意味が分かりにくく適切な名称をつけることができない可能性が高い。このため、任意に選んだ単語がどの「カテゴリ」に属するのか理解することは難しい。また、「カテゴリ」の再作成を行うときに構成が大きく変わり、変更前との関連が失われる可能性が高い。

どちらの手法を使う場合も、記事を特徴付けるという目的に立ち返って、「カテゴリ」の個数を定め、振る舞いについて調査を必要とする。

②同字異義語

記事を形態素解析して単語を取り出すために、形態素解析での問題は常に発生するが、カテゴリを使用する場合はさらに別の問題が発生する。

例として「カテゴリ」として「東京カテゴリ」、「大阪カテゴリ」を定めたとする。このとき名詞「日本橋」は、東京都中央区にある地名「日本橋（にほんばし）」と大阪府大阪市中央区・浪速区の「日本橋（にっぽんばし）」のどちらを表すかは文脈に依存する。今回は、名詞に対して「カテゴリ」を一意に定めるか、もしくは一般性が高いとして「カテゴリ」を与えない方式をとっている。

統計的に同字異義語を扱う方式を組み込むことで文脈に依存した「カテゴリ」を与え、解析精度の向上を目指すことは可能と思われる。

③占有率の計算

占有率の計算は雑誌標準タグごとに重みを与えて単純な総和を求めたが、他の実装方式も考えられる。

・複数の雑誌標準タグに現れたときに重みを変化させる。

・本文のなかでも文章中の位置によって重みを変化させる。

・文章を形態素解析するだけではなく、構文解析も行うことで単語とそれに紐づく「カテゴリ」の重みを変化させる。

いくつもの方式で実装し実際の記事を解析し、より直感に合致するものを選択することが好ましい。

④出現単語の並び替え

記事の特徴量として「カテゴリ」の占有率を選び、これに従った出現単語の並び替えを行ったが、別の実装方式も考えられる。

・複数の雑誌タグに現れる単語が上位になるように全体を並び替える。

・後述の共起辞書を用いて、記事のなかで関連する単語が同時に出現している場合に上位になるよう全体を並び替える。

・形態素解析の後に構文解析を行い重要単語を推定し上位になるよう全体を並び替える。なお形態素解析の正確性、構文解析のロジック調整などが必要で理論的にも非常に難しい。

検索ロジックと連動して並び替えのロジックも調整する必要がある。

■共起辞書

〇機能概要

関連性がある単語同士は同じ文書に現れる可能性が高いことが観察される。視点を逆にして、2 つの単語が同じ文書に現れる頻度が高い場合に関連性が高いと仮定することで、単語の組み合わせの関連性を定量的に測ることができる。

共起性はある定められた文書集合の範囲での統計情報となる。文書集合の定め方によって値や傾向が変化する。このため、実際に利用する際には、定めた文書集合から得られた共起性が利用目的に合致するよう検証し必要に応じて調整を行う必要がある。

共起率を以下のように定義する。

2 つの単語を A 、 B とする。 A が現れる文書数を a 、 B が現れる文書数を b 、 A と B の両方が現れる文書数を i とする。このとき共起率 $r = r(A,B)$ を次の式で定義する：

$$r = \frac{i}{a+b-i}$$

定義から共起率は対称である。すなわち次の式をみたす。

$$r(A,B) = r(B,A).$$

なお $a+b-i$ は A と B の一方もしくは両方が現れる文書数と一致する。また共起率 r は必ず $0 \le r \le 1$ を満たす。

共起率による半距離 $d = d(A,B)$ を次の式で定める。

$$d = -\log r$$

ただし $-\log 0 = +\infty$ とする。半距離も対称であり $d(A,B) = d(B,A)$ を満たす。

なお、この半距離は三角不等式を満たさないため、距離にはならない。すなわち、次の三角不等式には反例が存在する。

$$d(A,C) \le d(A,B) + d(B,C)$$

共起率が高いことと半距離が小さいということは同等だが、後者は単語同士が近いという、より直感的な表現に合致するため、細かな検討に役立つと考えられる。

具体的な例として「東京」「大阪」「築地」の 3 単語を考える。

それぞれの単語、単語の組み合わせの出現回数を仮定する。

単語	東京	大阪	築地
出現文書数	40	20	5

資料 16　単語の出現数

単語の組み合わせ	東京、大阪	大阪、築地	築地、東京
出現文書数	10	1	3

資料 17　単語の組み合わせの出現数

この場合それぞれ共起率と半距離は次の式で求められる。

$$r(\text{東京},\text{大阪}) = \frac{10}{40+20-10} = 0.2 \qquad d(\text{東京},\text{大阪}) = -\log 0.2 = 1.609$$

$$r(\text{大阪},\text{築地}) = \frac{1}{20+5-1} = 0.04167 \quad d(\text{大阪},\text{築地}) = -\log 0.04167 = 3.178$$

$$r(\text{築地},\text{東京}) = \frac{3}{5+40-3} = 0.07142 \quad d(\text{築地},\text{東京}) = -\log 0.07142 = 2.639$$

なお、この例では三角不等式が満たされている。

〇機能詳細

カテゴリを持つ名詞に限定して共起率をもとめ、共起辞書を作成する。文書の集合として日本語版 Wikipedia のダンプデータを用いた。具体的な作成手順は以下のとおりである。

①日本語版 Wikipedia のダンプデータの各ページを 1 文書として、形態素解析を行う。

②現れる名詞の「原型」を取得する。取得した「原型」を 1 単語として、各単語の出現回数と 2 単語の組み合わせの出現回数を数え上げる。

③数え上げの結果から共起率と半距離を計算し共起辞書に格納する。

例えば「東京から大阪へ」、「東京の築地」という 2 つの文書からは、「東京」「大阪」「築地」の 3 語が抽出され、それぞれの出現回数を数える。

単語	東京	大阪	築地
出現文書数	2	1	1

資料 18　単語の出現数

単語の組み合わせ	東京、大阪	大阪、築地	築地、東京
出現文書数	1	0	1

資料 19　単語の出現数

〇チューニングポイント

①共起辞書を作るための文書集合

文書の集合として日本語版 Wikipedia を採択した理由としては、あらゆるジャンルの文書が得られることが挙げられるが、データの偏りがあるため共起辞書としては精度は十分でない。十分な量の雑誌記事を用意し文書の集合として使うことが妥当と予想される。

②共起辞書の作成で使用する単語

「カテゴリ」を持つ名詞だけではなく、すべての名詞を使う方式や、動詞、形容詞、副詞なども数え上げる方式などが考えられる。あまりに多くの文書に現れる単語はほとんどの単語と共起することから、記事の特徴づけには使えないため共起辞書からは削除する必要がある。逆に出現回数が少ないものもノイズとして可能性が高いため、共起辞書からは削除するべきである。

また「原型」を使うことで同義語を集約することはより適切な共起性が得られる。さらに類義語を集約

することで、感覚的に記事の内容を捉えるということに近づける可能性がある。

③共起率の定義式

共起率の定義は一意ではなく用途に応じて変更することができる。

以下の例が考えられる：

2 つの単語を A、B とする。A が現れる文書数を a、B が現れる文書数を b、A と B の両方が現れる文書数を i とする。

$$r(A,B)=\frac{i}{a+b-i} \qquad d=-\log r$$

$$r_m(A,B)=\frac{i}{\min(a,b)} \qquad d_m=-\log r_m$$

$$r_s(A,B)=\frac{i}{\sqrt{ab}} \qquad d_s=-\log r_s$$

代替案としてあげた r_m は出現回数が多くない単語に重きをおいた定義である。出現回数が少ない単語についてはノイズとして振舞う可能性が高い。もうひとつの r_s は r と r_m の中間の振る舞いをする。実際にさまざまな定義で共起率を求めて記事の特徴づけや検索に有効な定義を検討する必要がある。

機能概要であげた例を再掲し、計算例とする。

単語	東京	大阪	築地
出現文書数	40	20	5

資料 20　単語の出現数

単語の組み合わせ	東京、大阪	大阪、築地	築地、東京
出現文書数	10	1	3

資料 21　単語の組み合わせの出現数

	共起率			半距離		
	r	r_m	r_s	d	d_m	d_s
東京、大阪	0.2	0.5	0.3536	1.609	0.6931	1.040
大阪、築地	0.04167	0.2	0.1	3.178	1.609	2.303
築地、東京	0.07143	0.6	0.2121	2.639	0.5108	1.551

資料 22　さまざまな共起率の比較

■連想ワードの付与

〇機能概要

検索容易性・本文到達性向上の実現の方式の 1 つとして、記事に関連する単語を検索対象に追加することが考えられる。たとえばある記事が「水星」「木星」「土星」という単語を含んでいるときには「惑星」や「太陽系」の単語で検索、発見できることは好ましいと思われる。

記事には現れないが記事内容に関連する名詞を連想ワードとして付与することで、連想ワードによる検索から記事に到達できる。

抽出ワードの上位にある複数の名詞と共起する名詞は記事全体と関連性が高いと判断する。機械的にこの条件を満たす名詞を付与するために、前述の共起辞書を使用する。

〇機能詳細

詳細なアルゴリズムを記述する。

①抽出ワードに記載された「原型」の上位n件までを取得する。デフォルトで$n=30$とする。

②取得したn件の「原型」のそれぞれに対して共起率の高い名詞を上位からm件まで取り出す。デフォルトで$m=100$とする。

③取り出した重複を含む延べ最大nm件の重複を削除し、共起率の総和の降順に並び替える。

④③で得た名詞からk件以上の「原型」に共通する名詞を取得する。

デフォルトで$k=5$とする。

⑤④で取得した名詞のうち抽出ワードに含まれないものを上位からj件まで取得する。

デフォルトで$j=20$とする。

⑥取得した名詞と共起率の総和を 2 つ組にして、インデックス情報テーブルの項目 27 目次・連想ワードに入力する

例として$n=3$、$m=5$、$k=2$、$j=2$とした場合の連想ワードを考える。

アルゴリズムの例示を意図するため、データは人工的に作成する。

①抽出ワードの上位$n=3$位は以下の名詞と仮定する。

「東京」、「大阪」、「築地」

②それぞれに対して共起率の高い名詞を上位から$m=5$件取得する。

「東京」の共起語					
「原型」	大阪	千代田区	港区	大学	埼玉
共起率	0.401	0.201	0.101	0.081	0.061
「大阪」の共起語					
「原型」	京都	東京	大学	記念日	港区
共起率	0.402	0.401	0.102	0.082	0.062
「築地」の共起語					
「原型」	東京	中央区	市場	大学	記念日
共起率	0.303	0.203	0.103	0.083	0.073

資料 23　それぞれの共起語

③取り出した重複を含む延べ最大 15 件の重複を削除し、共起率の総和の降順に並び替える。

「原型」	東京	京都	大阪	大学	中央区
共起率の和	0.704	0.402	0.401	0.266	0.203
共起する単語数	2	1	1	3	1

「原型」	千代田区	港区	記念日	市場	埼玉
共起率の和	0.201	0.163	0.155	0.103	0.061
共起する単語数	1	2	2	1	1

資料 24　共起語と共起率の総和

①③で得た名詞から $k = 2$ 件以上の「原型」に共通する名詞を取得する。

「東京」の共起語				
「原型」	東京	大学	港区	記念日
共起率の和	0.704	0.266	0.163	0.155
共起する単語数	2	3	2	2

資料 25　連想ワード候補

②④で取得した名詞のうち抽出ワードに含まれないものを上位から $j = 2$ 件まで取得する。

最終的に「大学」と「港区」を得る。なお連想ワード候補のうち「東京」は抽出ワードに含まれるため、「記念日」は上位 3 位のため、取得の対象外となる。

〇チューニングポイント

①数値の調整

機能詳細の記述で一般的に記しているように件数の選び方によって機能の振る舞いが大きく変わる。特に性質に大きな影響を与えるのは、以下の 2 つの変数である。

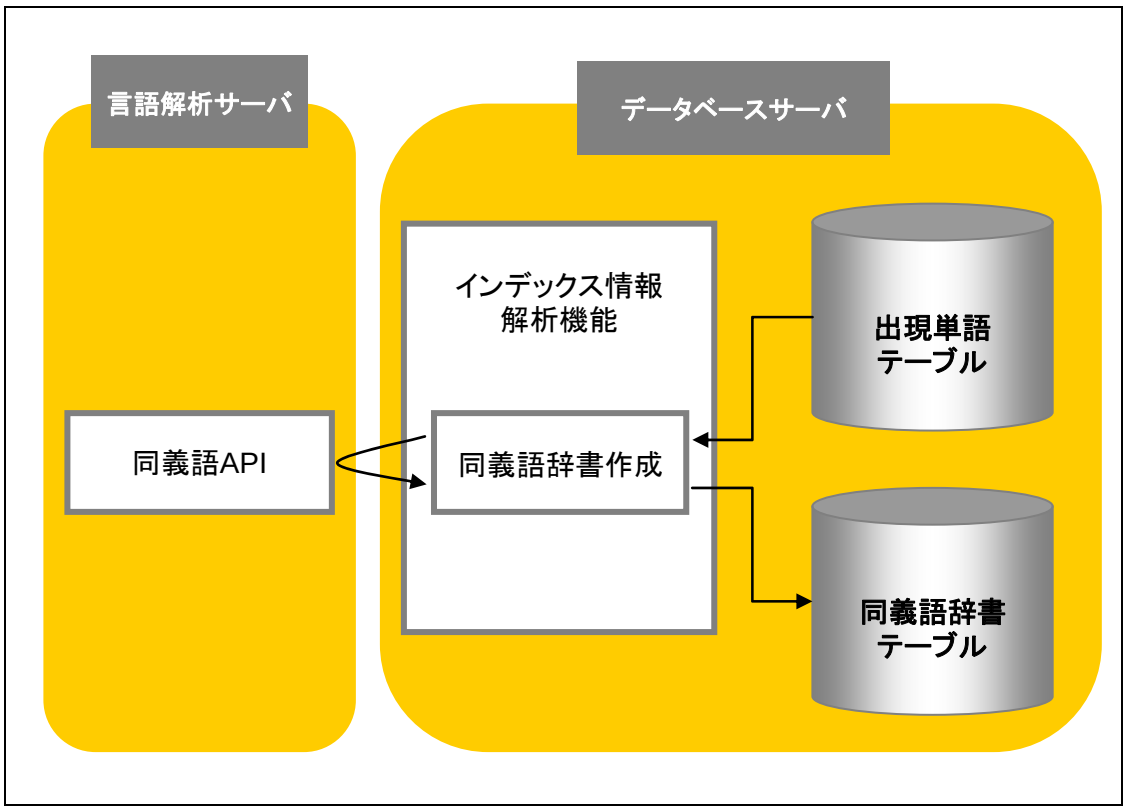
- n ：連想のもとになる記事中の単語の数。大きな値に設定すると記事中の重要でない単語も連想ワードの選択に影響を与えることになり、記事の内容から離れた単語も連想ワードとして取得しやすくなる。小さな値に設定すると振る舞いは逆になり、記事の内容により近い単語のみが連想ワードの候補になる。小さな値の場合は記事中の重要単語の選びかたに動作が大きく左右される。
- k ：連想ワードが共起する必要がある単語の数。大きな値に設定すると記事と関連性の高い単語を抽出できるが、十分な件数が得られない可能性が高くなる。小さな値に設定すると記事と部分的に関連する単語が多く得られる。

変数 n と k の組み合わせによって連想ワードの記事との関連性の振る舞いが変わるため、目的に合わせて値を調整する必要がある。なお、記事の長さによって値を変更する方式をとることも考えられる。

6)　同義語辞書

〇機能概要

略称や表記ゆれなど、同じものを表しているが表記が異なる言葉は多く存在する。記事に現れた表記に合致するものでなくても検索に利用し記事に到達できることが好ましい。このため、略語や表記ゆれを同義語辞書としてまとめ、検索の利便性を向上させる。



資料 26　「同義語辞書作成」機能概要図

〇機能詳細

インデックス解析で取得した抽出ワードと連想ワードに含まれる名詞の「原型」を取得する。取得したそれぞれの「原型」に紐づく「表層」を取得し同義語辞書に格納する。

〇チューニングポイント

①同義語の定義

どのような「表層」が同じ「原型」を現すかは人手で定める。異なるものを表す略語や表記ゆれに対して、複数の紐付けを行うか恣意的に一意に定めるかは検索結果に影響する。

例として「IC」という略語に対して「集積回路」や「インターチェンジ」などが考えられる。この「IC」の同義語として「集積回路」と「インターチェンジ」の一方を返すか、両方を返すかあるいは何も返さないか、検索の方式に依存して定めることができる。

7) テーブル定義

インデックス情報データベースの各テーブルの定義を記述する。

①インデックス情報テーブル

No.	項目名	属性	解説
1	レコード ID	正整数	レコードを特定するための ID
2	書誌・出版社名	文字列	発行出版社名（全角、最大 20 文字）
3	書誌・出版社名カナ	文字列	発行出版社名のカナ表記（全角カタカナ、最大 60 文字）
4	書誌・タイトル名	文字列	雑誌名※号数や月号表記は除く（全角半角、最大 30 文字）
5	書誌・タイトル名カナ	文字列	雑誌名のカナ表記（全角カタカナ、最大 80 文字）
6	書誌・巻数／号数	文字列	巻数、号数（全角、最大 12 文字）
7	書誌・底本雑誌コード	文字列	底本の雑誌コード（半角数字、半角ハイフン、半角スラッシュ、最大 11 桁）
8	書誌・底本発売日	文字列	底本の発売日（半角数字、8 桁固定） YYYYMMDD
9	書誌・配信開始日	文字列	※今回は空欄とする
10	書誌・開き	正整数	右開き／左開き（半角数字、1 桁固定） 0：右開き、1：左開き
11	書誌・コンテンツ管理コード	文字列	新たに規定した、版元が持つ流通管理のための共通の電子出版ファイルコード （半角英数字、20 桁固定） AAAAAAAABBBBCCCCCCCD
12	書誌・話名、記事名	文字列	その号の内容を表すもの。（全角半角、最大 100 文字）
13	記事・雑誌名	文字列	雑誌標準タグ magazine の内容
14	記事・コンテンツ管理コード	文字列	雑誌標準タグ c_code の内容から半角英数字以外のものを削除したもの 第 11 項の書誌・コンテンツ管理コードと一致する。
15	記事・特集名	テキスト	雑誌標準タグ magazine の内容 複数ある場合は連結している
16	記事・記事名	テキスト	雑誌標準タグ kiji の内容 複数ある場合は連結する
17	記事・タイトル	テキスト	雑誌標準タグ title の内容 複数ある場合は連結する
18	記事・サブタイトル	テキスト	雑誌標準タグ subtitle の内容 複数ある場合は連結する
19	記事・リード	テキスト	雑誌標準タグ lead の内容 複数ある場合は連結する
20	記事・大見出し	テキスト	雑誌標準タグ omidashi の内容 複数ある場合は連結する

21	記事・見出し	テキスト	雑誌標準タグ midashi の内容 複数ある場合は連結する
22	記事・本文	テキスト	雑誌標準タグ honmon の内容 複数ある場合は連結する
23	記事・キャプション	テキスト	雑誌標準タグ caption の内容 複数ある場合は連結する
24	記事・クレジット	テキスト	雑誌標準タグ credit の内容 複数ある場合は連結する
25	記事・その他	テキスト	雑誌標準タグ etc の内容 複数ある場合は連結する
26	目次・抽出ワード	テキスト	記事から抽出した名詞とカテゴリの列
27	目次・連想ワード	テキスト	第 26 項 目次・抽出ワードに現れる名詞と関連のある名詞を複数記載する
28	目次・カテゴリ 1	テキスト	現れる名詞から判定した記事の属性
29	目次・カテゴリ 2	テキスト	現れる名詞から判定した記事の属性
30	目次・カテゴリ 3	テキスト	現れる名詞から判定した記事の属性
31	目次・カテゴリ占有率 1	非負整数	項目 28 目次・カテゴリ 1 の記事における占有率
32	目次・カテゴリ占有率 2	非負整数	項目 29　目次・カテゴリ 2 の記事における占有率
33	目次・カテゴリ占有率 3	非負整数	項目 30 目次・カテゴリ 3 の記事における占有率

資料 27　インデックス情報テーブルレイアウト

②カテゴリテーブル

No.	項目名	属性	解説
1	レコード ID	正整数	レコードを特定する ID
2	カテゴリ ID	文字列	カテゴリを特定する ID
3	カテゴリ名称	文字列	カテゴリの名称

資料 28　カテゴリテーブルレイアウト

③出現単語テーブル

No.	項目名	属性	解説
1	レコード ID	正整数	レコードを特定する ID
2	表層	文字列	記事から抽出した名詞
3	原型	文字列	項目 2 表層が紐づく原型
4	カテゴリ ID	文字列	項目 2 表層が属するカテゴリの ID

資料 29　出現単語テーブルレイアウト

④同義語辞書テーブル

No.	項目名	属性	解説
1	レコード ID	正整数	レコードを特定する ID
2	表層	文字列	記事から抽出した名詞
3	表記ゆれ	文字列	項目 2 表層と同じ原型をもつ別の表層

資料 30　同義語辞書テーブルレイアウト

⑤共起辞書テーブル

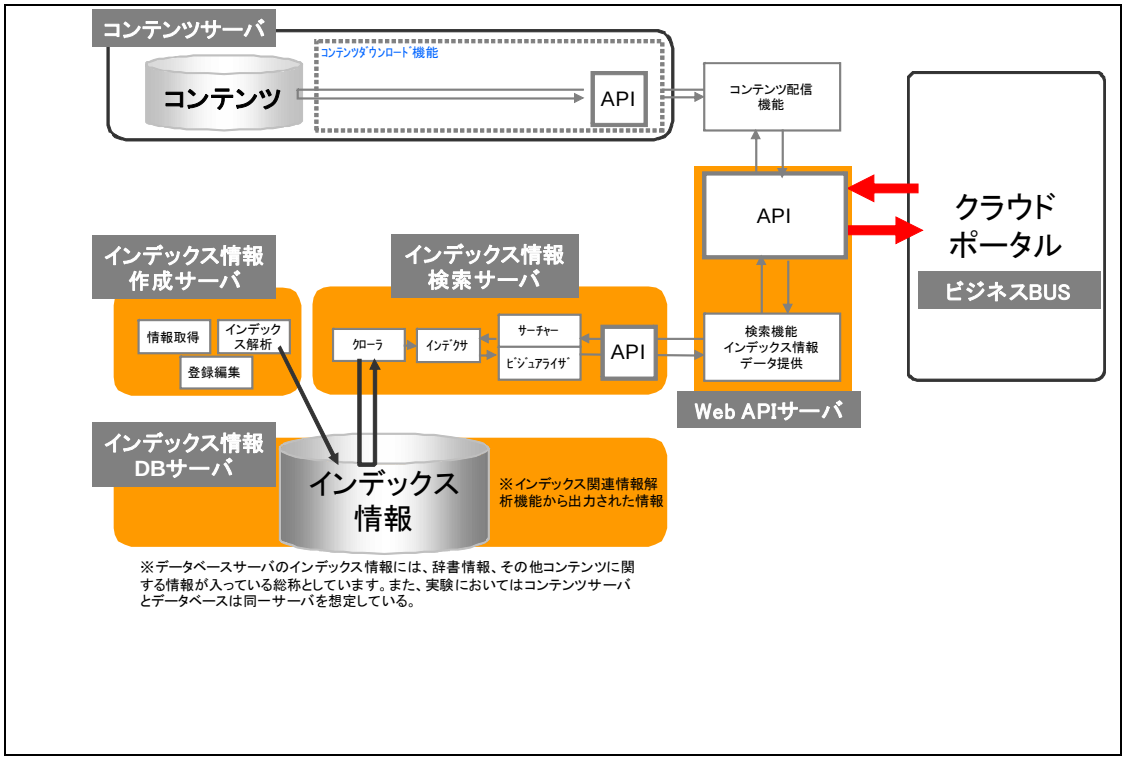
No.	項目名	属性	解説
1	レコード ID	正整数	レコードを特定する ID
2	原型 1	文字列	Wikipedia から抽出した原型の 2 つ組 UTF-8 文字コードで辞書式に並び替え一意にする
3	原型 2	文字列	
4	共起率	正実数	項目 2　原型 1 と項目 3 原型 2 の共起率（関連性） 1 以下の正実数となる
5	半距離	正実数	項目 4 共起率の自然対数の反数 正実数となり小さいほど共起率（関連性）が高い

資料 31　共起辞書テーブルレイアウト

(2)　検索・コンテンツダウンロード機能

検索機能およびコンテンツダウンロード機能は機能概要で述べたように電子出版スマートクラウドを想定して個々の機能を API 実装するとともに、機能要求を一手に受け付ける Web API の実装により構成されている。検索機能は、インデックス情報データベースに格納されている記事情報ならびにインデックス情報に対してキーワード検索・絞り込み・並び替え・検索対象設定・重み付け設定を提供している。検索機能の大きな特徴として、1 リクエスト要求で検索に必要なキーワード検索・絞り込み・並び替え・検索対象設定・重み付け設定を自由に選択し組み合わせて使用できることにある。インデックス情報データベースコンセプトガイドライン案では、対象プレイヤーとして出版社（者）・流通業者（配信 PF/電子書店/リアル書店など）・読者（国民）などそれぞれ求める検索条件や機能が異なるため本来であれば個別に API の設計を検討する必要があるが、1 リクエスト要求の中で必要な機能を選択できることで個別の要求に応えることが可能である。コンテンツダウンロード機能は、コンテンツサーバに格納されている記事データを配信する機能を提供している。

検索機能およびコンテンツダウンロード機能の機能説明を行う前に、各サーバと API のフロー図と各サーバの役割を以下（資料 32）に示す。



資料 32　「検索・コンテンツダウンロード機能」フロー概要図（再掲）

■各サーバの役割

「検索・コンテンツダウンロード機能」フロー概要図（資料 32）にある各サーバの役割は以下のとおりである。

1)　コンテンツサーバ

①記事データ（PDF データファイル）と中間ファイルフォーマット（FMT）が格納されている。

②API を実装、Web API サーバからコンテンツの要求を受け付ける。API は、対象のコンテンツ管理コード、ダウンロード要求者の ID 情報のパラメーターを受け取ることで、要求を受け付け対象の記事データを要求元へ送信する。

③コンテンツ要求をログに保管する機能を有する。（コンテンツ管理コード、ダウンロード要求者の ID 情報、要求日時を保管）

ログ機能は、実証実験のために実装されている。

2)　インデックス情報検索サーバ

①クローラ、インデクサ、サーチャー、API により構成される。

②クローラは、インデックス情報作成サーバにより生成されたインデックス情報が格納されているインデックス情報データベースからインデックス情報を参照してインデクサへ引き渡す機能。

③インデクサは、クローラによって引っ張ってきたインデックス情報を解析して検索用のインデックスを作成する機能。

インデックスは、インデックス情報が更新される度に再構築する必要がある。

※実証実験においては、インデックス情報の更新の検知もしくは更新の通知の受け取りならびにそれによるインデックスの自動再構築を行う機能は有していません。手動によりインデックスの更新を実行している。

④サーチャーは、API を介して要求のあった検索条件（キーワード、検索対象、重み付け設定、絞り込み条件、並び替え条件）を元にインデックスから該当するデータの抽出と表示順序を判定するスコアの計算を行っている。

⑤ビジュアライザは、サーチャーが抽出したデータと算出したスコアデータを受け取り、要求元が必要とするファイルフォーマットに整形する機能。

⑥API を実装、Web API サーバから検索要求を受ける。API は、検索条件を受け取りサーチャーへ検索条件を渡す。ビジュアライザから受け取った検索結果を要求元へ回答する。

⑦検索要求をログに保管する機能を有する。（キーワード、検索対象、重み付け、検索要求者の ID 情報、要求日時を保管）

ログ機能は、実証実験のために実装されている。

3) Web API

①中心となる API、クラウドポータルから、検索もしくはコンテンツダウンロードの要求の受付を行う。

②要求に応じた必要な機能を有するクラウドサーバ上の API に対して要求の受付と回答を行う。

■API 説明

前項の各サーバの役割より Web API として要求を受け付ける検索機能とコンテンツダウンロード機能の API 仕様について説明を記述する。

4) 検索機能

検索を行う際に Web API へ要求するリクエストパラメーター（資料 33）により REST で行われる。初回検索時にキーワード、検索対象、絞り込み、重み付けのリクエストパラメーターを付け Web API へ要求を行う。

■リクエストパラメーター

名前	パラメーター名	データ型	入力値	備考
キーワード	q	Text	徳川 etc..	ユーザーの任意入力
雑誌名	target_art_magazine	Boolean	0,1	0 .. 非対象, 1 .. 対象
コンテンツ管理コード	target_art_c_code	Boolean	0,1	0 .. 非対象, 1 .. 対象
特集名	target_art_tokushu	Boolean	0,1	0 .. 非対象, 1 .. 対象
記事名	target_art_kiji	Boolean	0,1	0 .. 非対象, 1 .. 対象
タイトル	target_art_title	Boolean	0,1	0 .. 非対象, 1 .. 対象
サブタイトル	target_art_subtitle	Boolean	0,1	0 .. 非対象, 1 .. 対象
リード	target_art_lead	Boolean	0,1	0 .. 非対象, 1 .. 対象
大見出し	target_art_omidashi	Boolean	0,1	0 .. 非対象, 1 .. 対象
見出し	target_art_midashi	Boolean	0,1	0 .. 非対象, 1 .. 対象
本文	target_art_honmon	Boolean	0,1	0 .. 非対象, 1 .. 対象
キャプション	target_art_caption	Boolean	0,1	0 .. 非対象, 1 .. 対象
クレジット	target_art_credit	Boolean	0,1	0 .. 非対象, 1 .. 対象
その他	target_art_etc	Boolean	0,1	0 .. 非対象, 1 .. 対象
出版社	target_mag_publisher_name	Boolean	0,1	0 .. 非対象, 1 .. 対象
抽出キーワード	target_ind_abstract_words	Boolean	0,1	0 .. 非対象, 1 .. 対象
雑誌名 重み付け	boost_art_magazine	Decimal	0.0 ～ 10.0	
コンテンツ管理コード 重み付け	boost_art_c_code	Decimal	0.0 ～ 10.0	
特集名 重み付け	boost_art_tokushu	Decimal	0.0 ～ 10	
記事名 重み付け	boost_art_kiji	Decimal	0.0 ～ 10	
タイトル 重み付け	boost_art_title	Decimal	0.0 ～ 10	
サブタイトル 重み付け	boost_art_subtitle	Decimal	0.0 ～ 10	
リード 重み付け	boost_art_lead	Decimal	0.0 ～ 10	
大見出し 重み付け	boost_art_omidashi	Decimal	0.0 ～ 10	
見出し 重み付け	boost_art_midashi	Decimal	0.0 ～ 10	
本文 重み付け	boost_art_honmon	Decimal	0.0 ～ 10	
キャプション 重み付け	boost_art_caption	Decimal	0.0 ～ 10	
クレジット 重み付け	boost_art_credit	Decimal	0.0 ～ 10	

その他 重み付け	boost_art_etc	Decimal	0.0 ～ 10	
出版社 重み付け	boost_mag_publisher_name	Decimal	0.0 ～ 10	
抽出キーワード 重み付け	boost_ind_abstract_words	Decimal	0.0 ～ 10	
並べ替え	sort	Text	-score, -mag_publish_date	“-” .. 降順, “” .. 昇順
目次情報	selected_facets	Text	ind_abstract_words:日本	複数指定可
カテゴリ	selected_facets	Text	ind_category:その他一般ワード	複数指定可
出版社	selected_facets	Text	mag_publisher_name:出版社名	複数指定可

資料 33　検索機能リクエストパラメーター

入力パラメーターの内、テキスト情報については UTF-8 エンコードを行うこと。

■レスポンスフィールド

名前	パラメータ名	値
対象件数（ヘッダー）	numFound	125
<doc>	子セット開始	
スコア	score	1.0
電子出版コンテンツ流通管理コード	art_c_code	404a0001006201101241
キャプション	art_caption	方広寺鐘銘
クレジット	art_credit	作家 山名美和子 やまな・みわこ
その他	art_etc	特別読み物
本文	art_honmon	慶長十九年（一六一四）十月一日、
記事名	art_kiji	大坂の陣　女たちの戦い
リード	art_lead	姉妹で敵味方に分かれてしまった淀殿とお江
雑誌名	art_magazine	歴史読本
見出し	art_midashi	方広寺鐘銘事件
タイトル	art_title	大坂の陣 女たちの戦い
特集名	art_tokushu	徳川幕府誕生
抽出キーワード	ind_abstract_words	<str>徳川家康</str>
カテゴリ	ind_category	<str>日本史</str>
発行年月	mag_publish_date	2011-01-24T00:00:00Z
出版社名	mag_publisher_name	新人物往来社
</doc>	子セット終了	

資料 34　検索機能レスポンスフィールド

①キーワード検索機能

リクエストパラメーターのキーワードに検索したいワードを入れて Web API へリクエストすることで検索にヒットした記事情報をレスポンスフィールドの書式により回答される。キーワードを複数入力する場合は、半角スペースもしくは全角スペースを挿入する。最大 10 ワードまで対応している。

②検索対象機能

キーワード検索として設定したワードの検索対象項目を指定することができる。パラメーター名が“target_”で始まるパラメーターを 0 で検索非対象となり、1 とした場合は検索対象となる。

③重み付け機能

検索結果の表示順序は、検索エンジンのパッケージが持つスコアリングロジックにしたがって順位が決定される。機能概要で重み付けについて上述しているが、検索利用者により個々に求めている情報が異なるため恣意的にこの検索順位を制御したいケースで重み付け機能を利用することができる。

“boost_”で始まるパラメーター名を 0 から 10 まで変更することで検索対象ごとにスコアリングを高めたり低めたりするこることができる。

④絞り込み機能

初回検索時もしくは、検索結果に絞り込みのパラメーターを追加することで検索結果の対象件数を絞り込むことができる。例えば、キーワード「ABC」で検索した結果が 1000 件あった場合にインデックス情報・カテゴリ・出版社（者）の絞り込み候補を元に絞り込みたい条件をパラメーターに追加することで 1000 件以下のより目的の検索結果を導くことができる。パラメーター名が“selected_”で始まるパラメーターが絞り込みの対象となる。

⑤並び替え機能

初回検索時もしくは、検索結果に絞り込みのパラメーターを追加することで検索結果の表示順序の変更を行うことができる。デフォルトはマッチ度で高い順となっている。パラメーター名は“sort”となる。-score がマッチ度順、mag_publish_date が出版年月順となる。また、降順・昇順についてもパラメーターにより変更を行うことができる。

これらリクエストパラメーターは、初回検索時ならびに初回検索後にパラメーターを追加することで追加検索を行える。

5） コンテンツダウンロード機能

コンテンツのダウンロードを行う際に Web API へ要求するリクエストパラメーター（資料 35）により対象の記事データが http stream にて送信される。電子出版コンテンツ流通管理コードによりコンテンツ配信の有効性を確認するための補助的機能である。リクエストパラメーターとレスポンスは下記の通りとなる。

■リクエストパラメーター

名前	パラメータ名	データ型	入力値例	備考
コンテンツ管理コード	document_id	String	404a0001029401101241	
ユーザーID	user_id	String	test	ログインユーザーID

資料 35　コンテンツダウンロードリクエストパラメーター

■レスポンスフィールド

リクエストパラメーターの要求に応じて、記事データのファイルを http stream により要求元へ送信が行われる。

実証実験においては、各 API についてエラー制御・補正については考慮しない。