Vol.2009-NL-191 No.19
Vol.2009-SLP-76 No.19
2009/5/22

文字列検索に基づく同義語・類義語抽出ツールとその性能評価

吉田和弘[†]　吉田稔[†]　中川裕志[†]

同義語・類義語抽出ツール kiwii は、接尾辞配列による文字列検索を用いて、ユーザーのクエリに対してインタラクティブに対象文書中の同義語を提示するシステムである。文書からの同義語・類義語抽出としては、抽出対象の語句を、出現位置の周辺から取得した特徴量を用いてベクトル空間モデルで表現し、ベクトル間の距離の近い語句の対を抽出する手法が広く行われているが、kiwii はそうした手法にはない利点を多く備えている。本稿では、この kiwii システムの同義語抽出精度やクエリに対する応答性を、青空文庫などを検索対象に用いて検証する。

Performance Evaluation of a Synonym Extraction System Based on String Search

Kazuhiro Yoshida[†]　Minoru Yoshida[†]
and Hiroshi Nakagawa[†]

kiwii is a synonym extraction system that is based on string search using suffix arrays. The system can interactively extract synonyms of a given query from target documents. Synonym extraction from documents is a widely studied topic, and the most popular approach is to use vector space models, where the vectors are calculated from distribution of various features around words, and similarity of the vectors are used as similarity of the corresponding words. The kiwii system has several advantages that are difficult to achieve in conventional vector space models. In this paper, we evaluate the accuracy and interactivity of the system, using the text documents obtained from digital libraries such as Aozora Bunko.

1. はじめに

kiwii は、情報大航海プロジェクト共通技術「意味の似ている言葉の抽出」において開発された、同義語抽出支援のためのソフトウェアである。このソフトウェアは図 1 に示すような検索エンジン風の GUI を持ち、ユーザーがクエリを入力すると、検索対象文書（あらかじめインデックス付けしておく）から、クエリと似た文脈で使われている語句、すなわちクエリの同義語候補を発見し、ユーザーに提示する。このソフトウェアの応用分野としては、人手による同義語辞書作成の支援が挙げられる。テキストからの情報抽出システムを開発するために同義語辞書を必要としている場合などに、対象分野のテキストからインタラクティブに同義語を抽出できるこのシステムは、辞書作成者の助けになると思われる。

図 1　kiwii の GUI（「テーブル」の同義語を検索したところ）

文書集合から同義語・類義語を抽出する手法としては、文書集合に単語区切りや構

[†] 東京大学情報基盤センター
Information Technology Center, The University of Tokyo

Vol.2009-NL-191 No.19
Vol.2009-SLP-76 No.19
2009/5/22

文解析などの処理を施したうえで、抽出対象とする語句の集合を定めておき、語句の出現位置の前後の単語分布や係り受け情報に基づいてベクトル空間モデルを作る方法があり、盛んに研究が行われているが、欠点もある。この手法を適用するには、抽出対象とする語句をあらかじめベクトルに変換しておく必要があるため、ベクトル空間を作る段階で、対象をある程度制限しておく必要がある（たとえば、一定頻度以上の名詞のみを対象にするなど)。kiwii のようなインタラクティブなシステムにおいては、ユーザーが入力する任意のクエリを受け入れられること、また、様々なクエリに対して、それに応じて適切な語句が検索対象となることが望ましい。

kiwii のアルゴリズムは、文献1などで開発してきたもので、接尾辞配列（文献2）による文字列検索を用いることで、任意のクエリに対する同義語検索を、インタラクティブ性を損なわない応答速度で実現している。kiwii は、このアルゴリズムに基づき、PC などでも利用可能な独立のプログラムとして同義語検索を実装したものであり、情報大航海プロジェクトの成果物として公開されている。

アルゴリズムの性能は、分野を限った文書などを用いて既に検証しているが、より規模が大きい、一般的な分野の文書を対象とした性能検証が待たれていた。本研究では、このシステムのデモ用に作成した、青空文庫などからなるテキストデータを対象として、kiwii プログラム性能を検証する。

2. kiwii の同義語抽出アルゴリズム

接尾辞配列を用いると、文書中での任意の文字列の出現位置の検索や、出現頻度の計算が、おおむね文書長の対数オーダーの計算量で実現できる。実際にメモリ上に接尾辞配列を構築し、単一ユーザー向けの検索システムを開発する場合、(キャッシュ効率などにもよるので一概には言えないが）一回のクエリあたり数万から数十万回の接尾辞配列への問い合わせ（すなわち、出現位置や頻度の計算）が、インタラクティブ性を損なわずに利用可能であると考えられる。

kiwii は、接尾辞配列が持つこのような機能によって、以下に示すアルゴリズムで文書からの同義語抽出を実現する。ここで、検索対象文書を D、クエリ文字列を q とする。また、^は文字列の連結、|・|は文字列の長さ、f(・)は文字列の文書中での頻度を、それぞれ表わすものとする。

1. 接尾辞配列でクエリ文字列を検索する。文書中で q の右側に隣接する文字列 x を、次のスコア関数に基づいて上位 N 個取得する：

$$f(q\hat{}x)\log(|D|/f(x))$$

これを右側文脈語集合と呼ぶ。このスコア関数は、q の文脈として高頻度で現れ（f(q^x)が大きく）かつ q に特有の文脈になっている（f(x)が低い）ようなものに高い重みを付けるようになっている。同様に、左側文脈語集合も取得する。実際には、クエリ q を文書中に現れる隣接文字列で伸長しながらこのスコアを計算していき、スコアが高いものを採用する。

2. 右側文脈語集合の左側に隣接する文字列を、再び接尾辞配列を利用して取得する。このとき、隣接する右側文脈語の個数が多いものを、上位 K 個取得する。これを右側同義語候補集合と呼ぶ。同様に左側同義語候補集合も取得する。
3. 右側同義語候補集合と左側同義語候補集合の合併の要素 s を、次のスコア関数で並べ替える（ここで R は右側文脈語集合、L は左側文脈語集合である）

$$\sum_{x\in R}\log\frac{f(s\hat{}x)|D|}{f(x)f(s)}+\sum_{x\in L}\log\frac{f(x\hat{}s)|D|}{f(x)f(s)}$$

（このスコア関数の各項で、f(x)f(s)は、頻度が f(x)の単語と f(s)の単語が偶然共起する確率に対応しており、それに対して実際の共起頻度 f(s^x)がどの程度過剰かどうかを測っている。）結果の上位を、同義語候補としてユーザーに提示する。

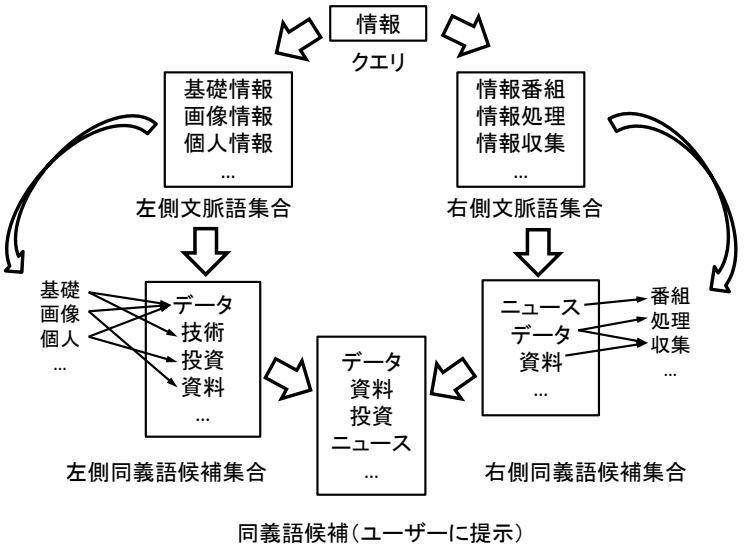

図 2　検索アルゴリズムの概要

アルゴリズムの概要を図にしたのが図 2 である。クエリが与えられると、まず接尾

Vol.2009-NL-191 No.19
Vol.2009-SLP-76 No.19
2009/5/22

辞配列を用いてクエリの右側・左側に隣接する文字列が検索され、右側文脈語集合および左側文脈語集合が得られる。(隣接文字列を適当な長さで打ち切って語句を切り取るために、スコア関数が使われている。) 得られた文脈語集合を用いて、左側文脈語集合の右側文脈、右側文脈語集合の左側文脈を検索する。こうして得られる文字列は、クエリと右側の文脈または左側の文脈を共有していることになるので、クエリの同義語候補として適切なものが得られると考えられる。

上のアルゴリズムに現れる探索の手続きやスコア関数は、文字列の隣接関係の検索や文書中の文字列の頻度によっており、全て接尾辞配列を用いて効率よく計算可能である。また、アルゴリズム中、上位 N (K) 個を取得する部分は、スコア関数の上限の見積もりなどをうまく用いることによって、厳密に計算することができる。(詳細については文献 1 を参照。) 実際のプログラムでは、文字列の頻度が低くなったところで探索を打ち切るなどで高速化している。また、最終的な結果をユーザーに提示する際には、他の文字列に含まれる文字列を表示しないようにするなどしている。

このアルゴリズムの動作は、文書を文字列として見た時の文字の隣接関係だけによっており、検索結果として得られる同義語候補文字列の切り出しも、クエリに見合う文字列を取り出すことによって自然に達成される。このため、検索対象文書に事前に単語区切りなどを仮定する必要がなく、任意の言語の、任意の文字列に対して、ある程度の精度で結果を出力することができる。例えば、あとで述べる英語文書を対象として、kiwii に「I' ll」というクエリを与えると、結果は図 3 のようになる。

左側文脈語集合	抽出結果	右側文脈語集合
uess (I'll)	I will	(I'll) give you
" Then (I'll)	I shall	(I'll) tell you wh
" and (I'll)	I can	(I'll) show yo
reckons (I'll)	I must	(I'll) tell you a
…	…	…

図 3　クエリ「I'll」の検索結果

このクエリは通常の意味の単語ではなく、また検索結果に表れている「I will」などは、二語にわたっているが、アルゴリズムは正しく対応する部分を切り出すことに成功している。

既に述べた通り、既存の同義語抽出研究では、候補語の周辺の文脈における特徴量の分布をベクトル空間に対応させ、ベクトルの類似度によって単語の類似度を測る、ベクトル空間モデルが一般的である。ベクトル空間モデルは、候補語のさまざまな特徴量を統一的に扱うことができる利点があるが (文献3など)、候補語はあらかじめ、(記憶容量などの点で手に負える範囲で) ある程度制限しておく必要がある。また、候補語を増やすと、クエリに対してオンデマンドで類似度の高い語句を選び出す際にも、原理的にはクエリと候補語全ての類似度を計算する必要がある。この計算量を減らすためには、様々な工夫が可能であるが (例えば文献4)、任意の文字列に対応できるようにするのは自明でない問題である。kiwii のアルゴリズムは、このような問題点を、「文脈文字列」という、同義語抽出において特に有効な素性に特化した類似度計算を行うことによって回避しているとも考えられる。

3. 評価実験の準備

前節で述べたアルゴリズムによる同義語抽出性能は、航空分野のレポートを検索対象に用いて精度を測定するなどして検証してきている。(文献 1) しかし、そこで用いた文書はサイズが小さい一方で、同じ分野の定型的な表現を多く含んでいるために、応答性能の検証としても、精度の検証としても、システムにとって有利な状況になっていたとも考えられる。そこで本稿では、検索対象文書として青空文庫 (http://www.aozora.gr.jp/) および Project Gutenberg (http://www.gutenberg.org/) から取得した文書を用いて、より大規模で分野非依存な評価実験を行う。これらの文書は主に文学作品などから構成されており、今回の実験で用いたのは、青空文庫全体から取得した日本語文書約 128MB (utf-8) と、Project Gutenberg から適当に選択した英語文書約 37MB である。

アルゴリズムの性質からいって、何らかの検索結果を出力できるのは、クエリが検索対象文書中にある程度の頻度で出現している場合のみである。したがってシステムの性能評価を行う際にも、対象文書に出現する文字列の中から選択したクエリを用いるのが自然である。kiwii システムは任意の文字列をクエリとして受け付けるのではあるが、ここでは対象文書を既存の単語区切り・品詞解析器で解析した結果から、単語を無作為に抽出することによってクエリを作成する。解析器としては、日本語は MeCab (http://mecab.sourceforge.net/)、英語は次のサイトから入手した品詞解析器を用いる：http://www-tsujii.is.s.u-tokyo.ac.jp/~tsuruoka/postagger/。

3.1 応答性の検証のためのクエリの作成

まず、システムの応答性を検証するためのクエリを作成する。アルゴリズムの性質上、クエリに対する応答時間はクエリの対象文書中での頻度に応じて増加すると考えられるので、頻度ごとのクエリを作れば、頻度と時間の関係を検証することができると考えられる。ここでは、単語区切り済みの日本語で頻度が 10～20、20～40、40～80、

Vol.2009-NL-191 No.19
Vol.2009-SLP-76 No.19
2009/5/22

100～200、200～400、400～800、1000～2000、2000～4000, 4000～8000 の単語集合を作成し、それぞれから 50 語ずつ無作為抽出して作成したクエリを用いて、頻度ごとに平均応答時間を測定する実験を行う。また、文書中で頻度が最も高い単語 20 語に対して同様に応答時間を測定し、システムの最大応答時間の見積もりを得る。

さらに、システムのスケーラビリティの検証のためには、検索対象に用いる文書の大きさと応答時間との間の関係にも興味がある。文書のサイズを小さくすると、クエリの頻度自体も低くなるため、文書サイズと応答性の関係は単純ではないが、ここでは 126MB、92MB、62MB、30MB の日本語テキストデータを作成し、頻度 400～800 および 4000～8000（いずれも頻度は元のサイズの文書で測定したものである。）のクエリに対する応答時間を測定した。

3.2 精度検証のためのクエリの作成

精度の検証の対象とするクエリ集合を適切に選択する必要があるが、精度の測定のためには、同義語・類義語関係の判定が、ある程度実際の用例に依存せずに行える必要があるため、ここではクエリを名詞に限定する。(品詞は品詞解析器の出力を用いる。なお、動詞に対しても以下で述べたものと同様の精度検証を試みたが、活用語尾の扱いなど、同義語の判定が難しい事例が多数現れたため、本稿では結果は示さない。）さらに、結果の精度と、クエリの頻度との関係を観察するために、応答性検証用のクエリと同様、頻度ごとの無作為抽出によってクエリを作成する。

作成したクエリは、英語および日本語について、以下の 6 種類であり、形態素解析のエラーに由来する単語でない文字列や、固有名詞を除外した 20 クエリずつを用いた。（最初の行は、頻度 100～200 の名詞が 3812 語あったことを示す。その中から 20 語無作為抽出し、実験用のクエリ集合を作成する。）

① 日本語低頻度名詞：日本語の名詞で、頻度 100～200 のもの（母集団 3812）
② 日本語中頻度名詞：日本語の名詞で、頻度 300～600 のもの（母集団 1754）
③ 日本語高頻度名詞：日本語の名詞で、頻度 1000～2000 のもの（母集団 623）
④ 英語低頻度名詞：英語の名詞で、頻度 100～200 のもの（母集団 873）
⑤ 英語中頻度名詞：英語の名詞で、頻度 300～600 のもの（母集団 341）
⑥ 英語高頻度名詞：英語の名詞で、頻度 1000～2000 のもの（母集団 85）

実際に使用したクエリを、表 1 に示す。

表 1 精度測定に使用したクエリ

日本語低頻度名詞	征伐、樹立、威嚇、口吻、大屋、教科書、刺青、虱、歯、境界、鉱毒、一散、雲雀、危機、仕事場、愚図、鬼神、集会、小学、城中
日本語中頻度名詞	戸外、大工、信号、旅館、苦悶、日曜、伯母、監獄、はるか、御飯、銀行、無駄、巫女、かわいそう、整理、年月、悪口、待、判事、主任
日本語高頻度名詞	うえ、意識、原、紳士、鍵、汗、河、日本人、邪魔、ごろ、表現、都合、婆さん、限り、鉄、筆、三つ、節、現実、成功
英語低頻度名詞	everyone、brass、Roman、piano、seed、tragedy、forth、expectation、succession、imitation、siege、knife、temptation、tent、obedience、encouragement、governor、wound、messenger、culture
英語中頻度名詞	wisdom、industry、fall、study、future、advice、enemy、rain、smoke、bottom、freedom、confidence、sleep、month、terror、glance、fate、living、fellow、pity
英語高頻度名詞	interest、eye、thought、subject、spirit、husband、table、hour、letter、money、kind、war、fire、year、fact、gold、age、death、rest、water

4. 実験結果

前節で述べた方法で作成したデータおよびクエリを用いた実験の結果を述べる。なお、今回の実験では、2 節のアルゴリズムで示した N と K は、どちらも 500 に設定してある。また、文脈語集合、同義語候補集合の検索においては、長さ 20 を超える語句は検索対象から外してある。

4.1 応答性の検証

3.1 節で述べた方法で作成した各頻度のクエリを用いて、プログラムの応答時間を測定した結果が図 4 である。予想された通り、高頻度のクエリほど応答時間が長くなる傾向がはっきり観測できる。さらに、対象文書中でもっとも頻度が高かった 50 単語をクエリに用いたところ、平均応答時間は約 4200 ミリ秒、最長は 10000 ミリ秒であった。検索エンジンなどでは、数秒の応答時間は許容範囲内であるとされることも多いことを考えれば、この結果は kiwii システムがこの規模の文書では十分インタラクティブに利用可能であることを示していると考えられる。

さらに、頻度 400～800 および 4000～8000 のクエリに対して、文書サイズごとの応答時間を測定したところ、図 5 のようになった。全体として、文書サイズが大きくなるほど応答時間が延びる傾向はみられるが、勾配はなだらかである。原因の一つとしては、2 節のアルゴリズムで述べた、上位 N（K）個の語句集合を選択することによる枝刈りが有効に機能し、探索空間を制限していることが考えられる。

Vol.2009-NL-191 No.19
Vol.2009-SLP-76 No.19
2009/5/22

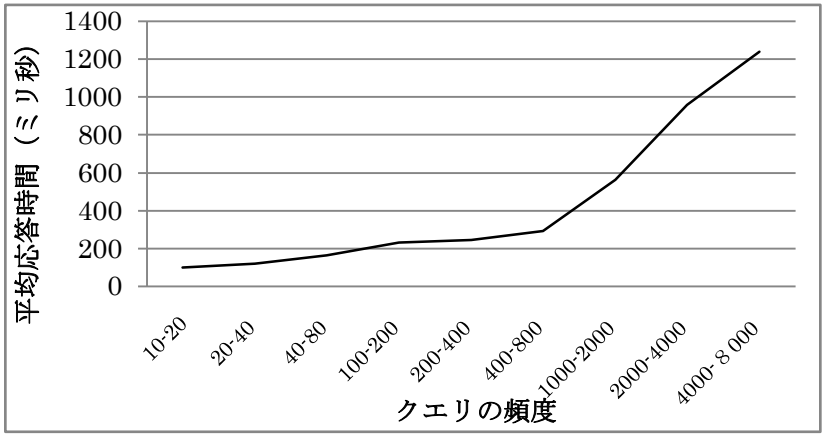

図 4　クエリの頻度と応答時間の関係

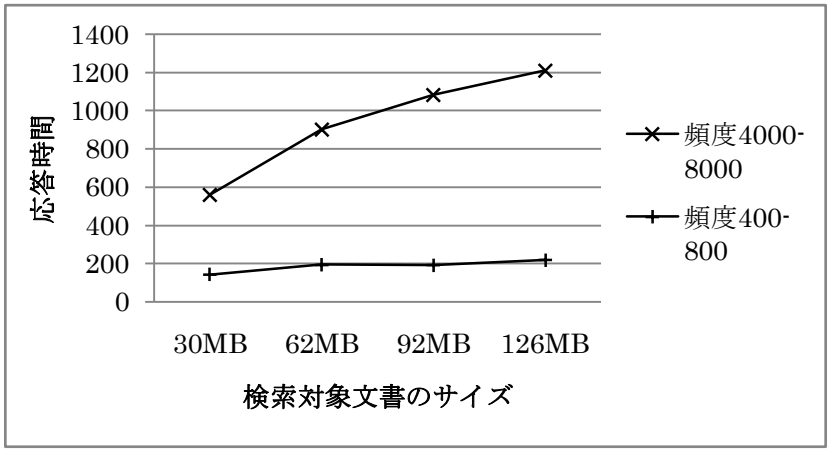

図 5　文書サイズと応答時間の関係

4.2　同義語抽出精度の検証

3.2 節で述べた方法で作成したクエリによる検索結果の、上位 20 語句を人手で評価することによって、同義語抽出の精度を検証した。人手評価の際には、次の 3 分類を用いた。

①　クエリの同義語：文脈に依存せず、似た意味で使える可能性が高い語
②　クエリの類義語：文脈によっては似た意味で使われる語（上位下位語なども含む）
③　それ以外：クエリとは無関係と判断された語

評価の結果、各分類に当てはまる語句の、検索結果中での割合を示したのが表 2 である。

表 2　検索結果のクエリに対する同義性の評価結果

	分類①の割合	分類②の割合	分類③の割合
日本語低頻度名詞	2.3%（9/400）	10.3%（41/400）	87.5%（350/400）
日本語中頻度名詞	5.3%（21/400）	17.5%（70/400）	77.3%（309/400）
日本語高頻度名詞	3.3%（13/400）	18.3（73/400）	78.5（314/400）
英語低頻度名詞	0.5%（2/400）	1.3%（5/400）	98.3%（393/400）
英語中頻度名詞	1.5%（6/400）	7.2%（29/400）	91.3%（365/400）
英語高頻度名詞	1.8%（7/400）	12.5%（50/400）	85.8%（343/400）

これを見ると、検索結果のうち、同義語と判定されたものは日本語で 2～5 パーセント、英語で 1 パーセント前後となっており、低いレベルにとどまっているが、正解を類義語まで広げると、精度は 10～20 パーセント程度となり、kiwii による類義語抽出は、人手による類義語辞書作成の支援などでは、有効に活用できる可能性を持っていると考えられる。全体に英語の精度が低くなっているのは、今回用いたデータでは、日本語に比べて英語の分量が少なかったことなども考えられるが、より精密な検証が必要である。

さらに、検索結果の並べ替えが適切なら、検索結果の上位ほど同義語・類義語の割合は大きくなることが期待されるが、これを検証するために、検索結果の、各順位までの累積正解率を、日本語、英語それぞれの類義語について（すなわち、分類①と②を正解として）測定し、グラフにしたのが図 6、図 7 である。

この結果を見ると、期待された通り、順位が高いほど精度も高くなっており、特に高順位では、英語の低頻度語を除いて、30 パーセント程度の正解率を得ている。また、全てのグラフを通して、頻度の高いクエリほど精度が高くなる傾向が伺える。データ数が少ないため、はっきりと断定はできないが、日本語名詞の場合のグラフを見ると、頻度 1000～2000 と頻度 300-600 の精度のグラフはほぼ重なっており、頻度 300 前後より高い頻度の単語なら、ある程度安定して高い類義語抽出精度を示していると解釈することもできる。なお、実験に用いたコーパス中で、頻度 100 以上の単語の単語数は約 8000 語、頻度 300 以上では約 6000 語であった。

Vol.2009-NL-191 No.19
Vol.2009-SLP-76 No.19
2009/5/22

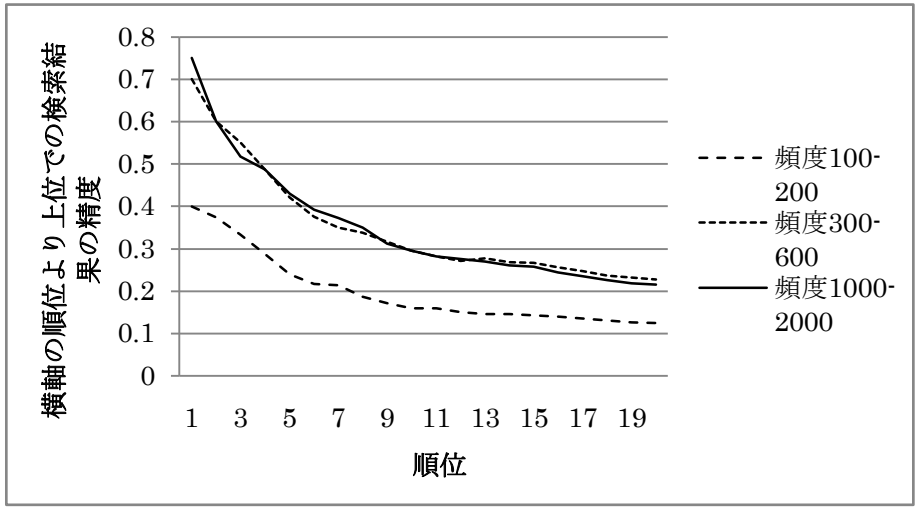

図 6　日本語名詞をクエリとしたときの、検索結果の順位と精度の関係

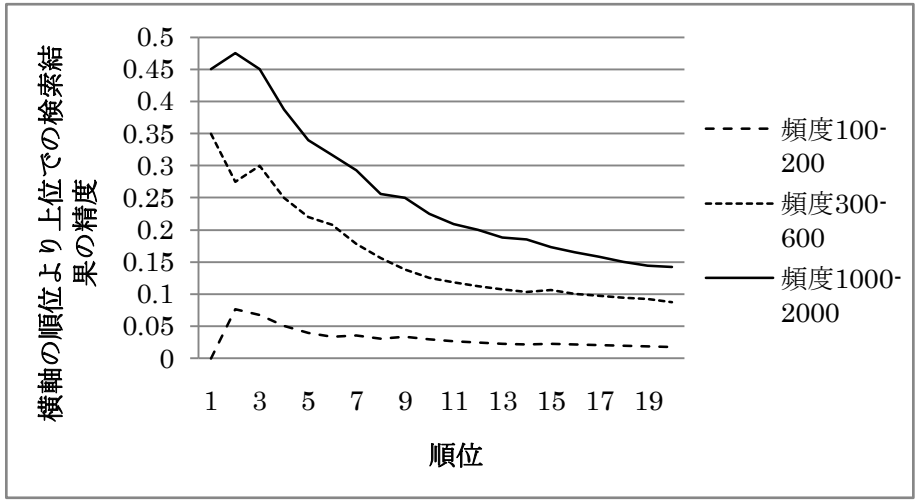

図 7　英語名詞をクエリとしたときの、検索結果の順位と精度の関係

5. おわりに

本稿で行った精度検証では、クエリの作成や精度検証を容易にするため、単語区切り済みの文書のうち、名詞から無作為抽出することでクエリを作成している。このように対象とする語句があらかじめある程度分かっている状況であれば、ベクトル空間による既存の同義語判定モデルも適用することができる。

そこで予備的な実験として、今回精度検証に用いた日本語文書中の、頻度 3 以上の語約 100,000 語に対して、隣接する 1 グラムおよび 2 グラムに基づくベクトル空間モデルを用いて、同義語候補リストを出力するプログラムを作成し、kiwii に与えたものと同じクエリに対する応答時間を計測する実験を行った。このプログラムは一つのクエリに対して、単語数（約 100,000）回の内積計算によって類似度を判定するのであるが、応答速度は kiwii と大差なかった。プログラムの文書サイズなどに対するスケーラビリティは kiwii よりも低いとは思われるが、データによっては、単純なベクトル空間モデルでも、高い応答性を実現できることになる。また、既に述べたように、この種のモデルにおける、近似計算による高速化手法も研究されている。

この結果から考えても、kiwii の優位性はかならずしも高速性にはないため、今後は任意のクエリや検索対象を扱えることの意義を何らかの形で検証していく必要がある。実際、図 3 に見られるように、単語以外のクエリや検索結果を含む、興味深い検索例もあるが、検証のためには、このような、単語以外で有用な結果が得られるテスト用クエリなどをうまく収集・作成する必要がある。

また、kiwii プログラムの応用としては、主に人手による同義語・類義語辞書作成の支援などを想定しているが、実際の同義語辞書作成などにおいて、インタラクティブな同義語抽出システムが助けになる場面があるのか、その際要求される精度がどの程度であるかなどは、未解決の問題である。さらに、文書を執筆する際に同義語辞書代わりに使用するなど、辞書作成支援以外にも有用な場面はあると思われるので、応用の可能性を探っていきたい。

参考文献

1 Minoru Yoshida, Hiroshi Nakagawa, and Akira Terada, Gram-Free Synonym Extraction via Suffix Arrays, AIRS 2008, pp.282-291, 2008.
2 Udi Manber and Gene Myers, Suffix arrays: a new method for on-line string searches, SIAM Journal on Computing, Volume 22, Issue 5, pp. 935-948, 1993.
3 Masato Hagiwara, Yasuhiro Ogawa and Katsuhiko Toyama, Selection of Effective Contextual Information for Automatic Synonym Acquisition, COLING/ACL 2006, pp.353-360, 2006.
4 Deepak Ravichandran, Patrick Pantel and Eduard Hovy, Randomized algorithms and NLP: using locality sensitive hash function for high speed noun clustering, ACL 2005, pp. 622-629, 2005.