同一内容を扱う文書間の差異の検出に向けて

野呂 智哉　　徳田 雄洋

東京工業大学 大学院情報理工学研究科

概 要

現在，様々な機関が大量のニュース記事をインターネット上で配信している．Google News や Yahoo! News など，多数のニュースサイトから記事を収集して表示するサイトでは，この膨大な量のニュース記事をいくつかのカテゴリに分類したり，同一 (類似) 内容の記事をまとめたりすることで効率的な記事の提示を行っているが，それだけでは十分とは言えない．例えば，アメリカ版 Google News は 4,500 ものニュースサイトからニュース記事を収集しているが，1 つの同一内容記事群に数百から数千の記事が含まれ，まだ人手で網羅できる量ではない．我々は，同一内容と判断された記事群に潜む差異を検出し，それを効率的に提示する手法を検討し，新たなニュース索引システムの構築を目指している．本稿では，その構想について述べる．

1　はじめに

インターネット環境の普及により，様々な情報を，即座に，簡単に入手できるようになった．ニュース記事 (以下,「記事」と略す) においても，従来の新聞社，通信社，放送局だけでなく，多種多様な機関が，膨大な量の記事を Web 上で配信している．これらの情報を網羅することにより，世の中の情勢を掴むことが可能になっている．

ところが，読者がこの膨大な量の記事を片端からすべて読み，網羅することは困難であり，有用な情報を効率的に取得できるような記事の提示が重要となる．1 つの解決法として，記事をいくつかのカテゴリに分類したり，同一 (類似) 内容記事をまとめたりするなどの手法がある．実際，Google News や Yahoo! News など，多数のニュースサイトから記事を収集して表示するサイトでは，このような手法で効率の良い記事の提示を実現している．

しかし，現在 Web 上で配信される記事の量は膨大であり，単にカテゴリに分類したり同一内容記事をまとめたりするだけではまだ不十分である．例えば，アメリカ版 Google News では 4,500 ものニュースサイトから記事を収集し，カテゴリ分類や同一内容記事判定を自動的に行っているが，たとえ膨大な量の記事を同一内容を扱う記事ごとにまとめたとしても，1 つのグループに入る記事数が数百件から数千件にのぼる．これだけの量の記事を人間が網羅することは困難である．

同一内容を扱う記事であっても，その内容がまったく同じであることは，一方が他方から記事を入手したり，両方が同じ機関から記事を入手したりして，そのまま配信している場合を除き，あり得ない．この，同一内容を扱う記事間に潜む差異に，読者にとって有用な情報がある可能性がある．

例えば，航空機が飛行中に乱気流に巻き込まれたという記事が複数のニュースサイトで配信されていたとする．事故の日時や場所，航空会社名，負傷者数などの主要情報はどの記事にも共通して書かれているが，乗客の証言や過去の類似事故などの補足的情報の有無は，配信元によって大きく異なる．システム側が，このような補足的情報を効率的に提示することができれば，読者にとって有益な情報をもたらす可能性を持っている．逆に，記事を読む時間が限られている読者に対しては，主要情報である共通部分のみを提示することにより，効率的な情報提供が可能となる．共通部分と差異部分のどちらを重要と見るかは読者の目的による．

現在，我々は世界中のニュースサイトから記事を収集し，ニュース索引システムを構築中である [1, 2]．本システムでは，Web 上から収集した話題語をもとに作成したディレクトリを利用して記事を分類し，それを国・地域別にまとめる．それにより，ある特定の話題に対する世界各国・地域の関連度合を見ることができる．現状では，話題語や国・地域別に記事を分類しているだけであるが，同一内容記事ごとにまとめ，さらにその中の差異を検出し，提示する機能を導入することにより，システムの拡張をはかる．本稿では，その構想について述べる．

2 構想

2.1 ニュース索引システム

前節で述べたように，我々はニュース索引システムを構築中である．本システムは，世界 20ヶ国 40 サイト以上の英語ニュースサイトから記事を自動収集し[1]，約 17,000 の話題語で分類する．記事は国・地域別にも分類されているため，話題語を 1 つ選ぶことにより，その話題と関連のある国・地域 (その話題語と共起する国・地域名) が分かるようになっている．逆に，先に国・地域名を選ぶと，その国・地域と関連のある話題が分かる．

しかし，本システムは，該当する記事を配信日順に並べて表示するだけであり，同一内容の記事をまとめたりする機能はない．我々は，ベクトル空間法による一般的な手法により記事を同一内容ごとにまとめた上で，同一内容と判断された記事間の差異を検出し，提示する手法を考える．

2.2 同一内容記事間の差異検出

同一内容を扱う記事間から差異を検出するためには，記事をいくつかの小単位に分割し，比較する必要がある．本研究では，以下の 3 通りの分割を考える．

1. 段落単位

   すべての記事は 1 つ以上の段落から構成される．段落単位で内容の同一性判定を行うことにより，共通した内容を持つ段落と一方にのみ含まれる内容の段落に分類できる．

2. 文単位

   比較対象の記事に段落が 1 つしかない場合，段落単位の比較では，その記事を共通部分と異なり部分に分けることができない．また，一方の記事では複数段落に分けて記述されている内容が他方では 1 段落にまとめられていたりするなど，段落の分け方に違いがある場合，段落単位の比較は適さない可能性がある．また，段落内に存在する小さな差異も，段落単位では検出できない．その場合，段落単位よりも小さい文単位での比較を行うことで，共通部分と異なり部分に分類できる．

3. 用言を中心とした係り受け構造単位

   段落単位での比較における問題と同様の問題が，文単位での比較においても存在する．一方の記事では非常に長い 1 文にまとめて記述されている内容が，他方では複数の短い文に分割されている場合などである．そこで，文単位よりさらに小さい単位を考える．ここでは，用言を中心とした係り受け構造を抽出し，それを単位に比較を行う．

[1] 日本語ニュースサイトからの記事の収集も検討中である．

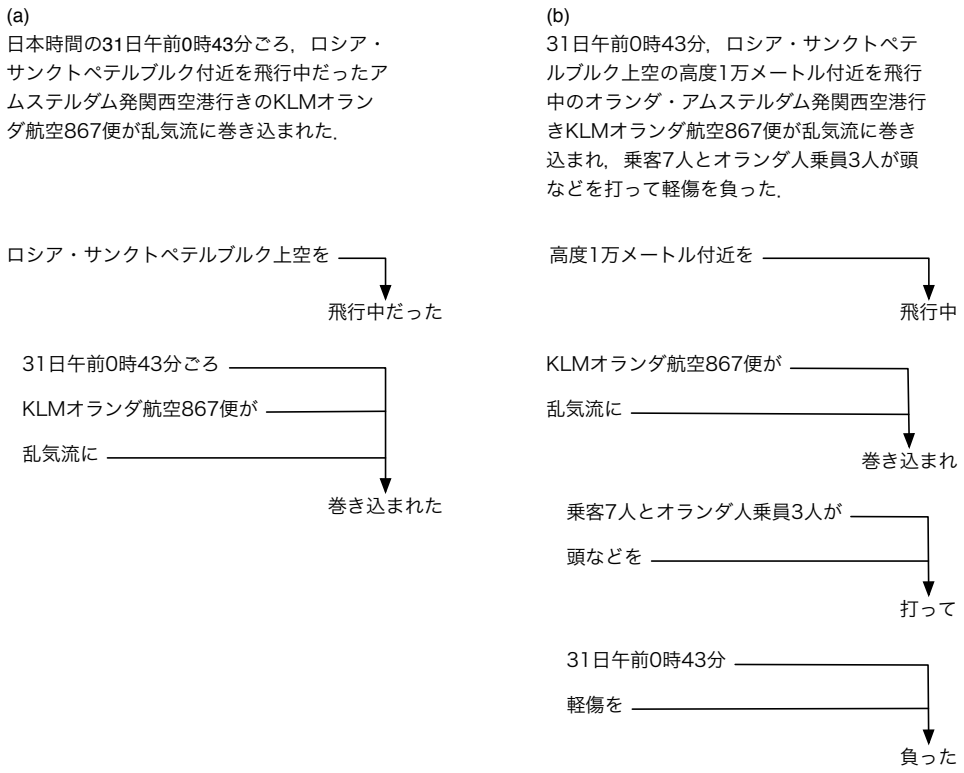

図 1: 用言を中心とした係り受け構造による比較

例を図 1 に示す．各文に対して係り受け解析を行い，用言とそれに係る文節を抽出する．文 (a) からは 2 つの係り受け構造が，文 (b) からは 4 つの係り受け構造が抽出できる．そして，「巻き込まれる」を中心とした構造が類似していることから，これが共通部分であると判断する．一方,「打つ」や「負う」を中心とした構造は文 (b) にしかないため，差異部分と判断する．

比較単位が小さいほど詳細な比較が可能となるが，代名詞や省略の存在などにより，照応解析などを行わないと，比較に必要な情報が欠落し，十分な比較ができなくなる可能性があるというトレードオフがある．この問題の解決方法についても検討する必要がある．

3 予備実験

今後の方針の検討のため，予備実験を行った．

3.1 英語記事の場合

我々のニュース索引システムが日常的に収集している記事のうち，2008 年 1 月から 5 月までの約 47 万記事 (約 1 億語) を利用し，出現単語の DF を計算した．ただし，TreeTagger [3] で形態素

表 1: 段落単位の比較 (イラク自爆テロ)

		記事 2							
		0	1	2	3	4	5	6	7
	0	**0.6012**	0.0785	0.0773	0.0000	0.0000	0.0000	0.0000	**0.1966**
	1	0.0000	**0.4919**	**0.2174**	0.0241	**0.1370**	0.0924	0.0236	**0.2664**
	2	0.0000	**0.2629**	0.0000	0.0166	**0.2721**	0.0000	0.0000	0.0804
	3	0.0000	**0.1563**	0.0135	0.0135	0.0542	0.0000	0.0000	0.0446
	4	0.0000	**0.1288**	0.0608	0.0458	0.0914	0.0001	0.0190	0.0002
	5	0.0000	0.0692	**0.1160**	0.0001	0.0003	0.0418	0.0513	0.0957
	6	0.0000	0.0637	0.0355	0.0797	**0.1403**	0.0178	0.0460	0.0257
	7	0.0000	0.0849	0.0797	0.0619	0.0073	0.0000	0.0083	0.0956
記	8	0.0000	**0.3091**	0.0178	0.0002	0.0005	**0.1472**	0.0137	0.0003
事	9	0.0000	**0.1990**	0.0664	**0.2472**	0.0003	0.0001	0.0001	**0.1064**
1	10	0.0000	**0.1988**	0.0986	0.0001	0.0312	0.0001	0.0188	**0.2103**
	11	0.0348	0.0210	0.0000	0.0000	0.0517	0.0000	0.0000	**0.1131**
	12	0.0000	**0.1131**	0.0792	0.0582	**0.1105**	0.0002	0.0082	0.0838
	13	0.0000	0.0442	0.0258	0.0173	0.0079	0.0000	0.0238	0.0000
	14	0.0000	**0.2688**	0.0488	0.0194	0.0006	0.0002	0.0002	0.0004
	15	0.0000	**0.1910**	0.0940	0.0302	0.0378	0.0000	0.0455	**0.2347**
	16	0.0421	**0.3045**	**0.1557**	0.0000	**0.1333**	0.0000	0.0754	**0.4946**
	17	0.0000	**0.2861**	0.0715	0.0199	0.0429	0.0136	0.0404	**0.1910**
	18	0.0000	0.0002	0.0000	0.0002	0.0005	0.0002	0.0300	0.0003

解析し，一般名詞，固有名詞，動詞 (be，have を除く[2])，形容詞，外来語のみを対象とする．多品詞語は品詞ごとに区別するが，動詞の語形変化，名詞の単数形・複数形は区別しない．異なり語数は約 139 万語であった．

差異の検出には，イラクでの自爆テロに関する記事，中国と台湾の首脳会談に関する記事，ネパール国王の宮殿からの退去に関する記事をそれぞれ 2 つずつ用意し，段落単位での比較を行った．結果を表 1，2，3 に示す．1 行目，1 列目の番号は段落番号を表す (記事のタイトルを 0 番目の段落とする)．スコアが 0.1 以上のものを太字で表記する．スコアは cosine 類似度による．

結果より，比較対象の記事のどの段落との間でも類似度が低い段落 (すべての値が低い行または列) は，2 つの記事の間の差異と考えられる．例えば，表 1 において，記事 1 の第 7，第 13，第 18 段落は，記事 2 のどの段落との間の類似度を見ても，0.1 未満となっている．これらの段落は，それぞれ以下のような内容であり，記事 2 には確かに記述されていない．

- テロが起きたのは，政府や警察関係の車両しか入れない，安全と思われていた地域だった
- 同じ日，別の場所でもテロがあり，4 人が死亡した
- 将官 (general) の発言

同様に，記事 2 の第 6 段落には，負傷した警察官の 1 人の発言について書かれているが，これは記事 1 にはない．

一方，類似度が高くても，その 2 つの段落の内容は必ずしも類似しているとは言えない場合も目立つ．段落単位や文単位での比較による類似度計算は，出現単語がどの程度一致するかを尺度にしているが，段落単位や文単位では対象となる単語数が少ないため，少しでも出現単語が一致すれば類似度が高くなってしまうことが原因であると考えられる．より精密な判定を行うためには，さらに詳細な分析による類似度計算が必要である．

[2]TreeTagger の品詞セットは PennTreebank に準拠しているが，be と have には，それ以外の動詞とは異なる品詞が割り当てられている．

表 2: 段落単位の比較 (中台首脳会談)

		記事 4								
		0	1	2	3	4	5	6	7	8
	0	**0.5610**	**0.2663**	**0.4055**	0.0274	0.0532	**0.1690**	0.0000	0.1572	0.0000
	1	**0.2961**	**0.2097**	**0.2281**	**0.1267**	0.0305	**0.2157**	0.0000	**0.1214**	**0.2079**
	2	0.0307	0.0124	0.0151	**0.9643**	**0.1527**	0.0113	0.0000	0.0000	**0.3189**
	3	0.0000	0.0000	0.0000	**0.1964**	**0.3815**	0.0002	0.0003	0.0000	0.0000
	4	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	5	**0.1812**	**0.1117**	**0.1616**	0.0000	0.0000	0.0728	0.0000	**1.0000**	0.0000
記	6	0.0000	0.0000	0.0000	**0.3163**	0.0000	0.0000	0.0000	0.0000	**1.0000**
事	7	**0.2046**	**0.1003**	**0.1510**	0.0000	0.0000	0.0587	0.0000	**0.1050**	0.0162
3	8	0.0000	0.0062	**0.1023**	**0.1812**	**0.3518**	0.0000	0.0000	0.0000	0.0000
	9	**0.1264**	**0.1294**	0.0621	0.0000	0.0153	**0.1102**	0.0001	0.0432	0.0128
	10	**0.2673**	**0.1075**	**0.1312**	0.0000	0.0000	0.0982	0.0000	0.0913	0.0021
	11	0.0000	**0.2308**	0.0280	0.0000	0.0000	0.0000	0.0000	0.0000	0.0146
	12	**0.2849**	**0.1740**	**0.2020**	**0.1989**	0.0166	**0.1096**	0.0292	**0.1328**	**0.3123**
	13	**0.1965**	0.0790	0.0965	0.0000	0.0000	**0.1622**	**0.1508**	**0.1300**	0.0000
	14	**0.2495**	**0.1224**	**0.1522**	0.0000	0.0000	**0.1607**	**0.1495**	**0.2622**	0.0000
	15	0.0000	0.0758	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

		記事 4								
		9	10	11	12	13	14	15	16	17
	0	**0.1667**	**0.1807**	**0.1096**	**0.2319**	0.0000	**0.3183**	**0.1705**	**0.2165**	0.0000
	1	**0.2927**	**0.1169**	**0.1523**	**0.3214**	0.0788	**0.4094**	**0.1405**	**0.2342**	0.0674
	2	0.0090	**0.1764**	0.0110	0.0000	0.0000	**0.1936**	0.0000	0.0000	0.0000
	3	0.0000	**0.4926**	0.0002	0.0000	0.0000	0.0001	0.0000	0.0000	0.0000
	4	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0419	0.0891
	5	0.0986	0.0000	0.0432	0.0913	0.0000	**0.1328**	**0.1300**	**0.2622**	0.0000
記	6	0.0152	0.0000	0.0128	0.0021	0.0146	**0.3123**	0.0000	0.0000	0.0000
事	7	**0.9391**	0.0000	0.0598	**0.2436**	0.0850	**0.1563**	0.0758	**0.1986**	0.0000
3	8	0.0000	**1.0000**	0.0000	0.0000	0.0000	0.0607	0.0000	0.0000	0.0000
	9	0.0561	0.0000	**1.0000**	0.0816	**0.1751**	0.0679	0.0600	0.0594	0.0000
	10	**0.2288**	0.0000	0.0816	**1.0000**	**0.1869**	**0.2708**	**0.1268**	**0.2970**	0.0000
	11	0.0799	0.0000	**0.1751**	**0.1869**	**1.0000**	0.0770	0.0000	**0.1037**	0.0000
	12	**0.1468**	0.0607	0.0679	**0.2708**	0.0770	**1.0000**	**0.2298**	**0.1906**	0.0506
	13	0.0712	0.0000	0.0600	**0.1268**	0.0000	**0.2298**	**1.0000**	**0.2076**	0.0000
	14	**0.1865**	0.0000	0.0594	**0.2970**	**0.1037**	**0.1906**	**0.2076**	**1.0000**	0.0000
	15	0.0000	0.0000	0.0000	0.0000	0.0000	0.0506	0.0000	0.0000	**1.0000**

表 3: 段落単位の比較 (ネパール国王)

		記事 6								
		0	1	2	3	4	5	6	7	8
	0	0.0000	**0.2878**	0.0000	0.0000	0.0000	0.0000	**0.1349**	**0.2313**	0.0104
	1	0.0288	**0.4628**	0.0000	**0.1363**	0.0114	**0.1350**	**0.1526**	**0.3293**	0.0854
記	2	0.0000	**0.1336**	0.0000	**0.2656**	0.0000	**0.2263**	**0.1308**	**0.1992**	**0.1099**
事	3	0.0000	0.0105	0.0257	0.0000	0.0000	0.0000	0.0002	0.0393	0.0000
5	4	0.0000	**0.1382**	0.0002	0.0048	**0.1038**	0.0449	0.0001	**0.1964**	**0.3769**
	5	0.0000	**0.2143**	0.0000	**0.1813**	0.0000	**0.1730**	**0.2029**	**0.2059**	**0.1136**
	6	0.0115	**0.3158**	0.0000	**0.4308**	0.0000	**0.3264**	**0.2898**	**0.2941**	**0.1622**
	7	0.0000	**0.1254**	0.0002	**0.1894**	0.0578	**0.1761**	**0.2543**	**0.2434**	**0.1194**

表 4: 段落単位の比較 (航空機事故)

		記事 B			
		1	2	3	4
記事 A	1	**0.4704**	**0.4426**	0.0289	0.1257
	2	**0.3770**	0.0994	0.2236	0.1623
	3	**0.3314**	**0.4289**	0.1608	0.1668
	4	0.1067	0.0949	0.1424	0.1476
	5	0.1257	0.0994	0.0000	0.0232

3.2 日本語記事の場合

同様の実験を日本語記事についても行った．ただし，現在のところ日本語記事を収集していないので，出現単語の DF は計算していない．形態素解析，文節係り受け解析には，MeCab と CaboCha，Juman と KNP の 2 組を併用する．具体的には，以下の手順で解析を行う．

1. MeCab と CaboCha で Layer 2 まで解析
2. Juman と KNP による解析結果から並列関係の係り受け (P，A，I) を抽出し，CaboCha による解析結果に反映
3. CaboCha で Layer 2 から再解析
4. KNP による解析の際に出力される文節の素性の一部 (格，連体修飾，用言などの情報) を反映

KNP と CaboCha で文節区切りが一致しない部分は，CaboCha の区切りを採用する．KNP は並列関係の解析に強いとされている．一方，CaboCha は全般的な解析精度が高いとされ，さらに，地名，人名などの固有表現の分析も行う．そこで，上述のように 2 つの解析器を併用し，双方の長所を取ることとした．

類似度計算には，名詞，動詞，形容詞のみを利用する (非自立語，接頭辞，接尾辞，数は除外する)．固有表現は 1 語として扱う．DF を計算しない代わりに，固有表現の重みを 2，それ以外を 1 とする．

差異の検出には，乱気流による航空機事故に関する記事を 2 つ用意し，段落単位，文単位での比較を行った．結果を表 4，5 に示す．段落単位の比較では 0.3 以上，文単位の比較では 0.1 以上のスコアを太字で表記する．

表 4 より，段落単位で比較すると，記事 A の第 1〜3 段落と，記事 B の第 1，2 段落の内容が類似しているという結果になっている．実際，記事 A の第 1，2 段落と記事 B の第 1 段落には，事故が起きた日時や場所，航空機の便名，けが人の数といった基本情報に関する記述があり，記事 A の第 3 段落と記事 B の第 2 段落には，事故が起きたときの状況に関する記述がある．一方，記事 A の第 4，5 段落には，それぞれ，乗客に対するインタビューの内容，空港での救急隊員の対応に関する記述があり，記事 B の第 3，4 段落には，事故機に乗っていたある旅行会社が企画したツアーの参加者に関する記述，過去の類似事故に関する記述がある．これらの間の類似度は低くなっている．

表 5 より，文単位で比較すると，それぞれの記事の第 1 文から第 6 文までは 1 文ずつ順番に対応し，第 7 文から内容の差異が大きくなっていることが分かる．記事 A の第 7 文は第 3 段落の最後の文，記事 B の第 7 文は第 2 段落の最後の文に相当する．前者には事故機の航空会社の発言に関する記述が，後者には着陸後に医師が機内で診察を行ったことに関する記述がある．段落単位の比

表 5: 文単位の比較 (航空機事故)

		記事 B					
		1	2	3	4	5	6
	1	**0.5695**	**0.2055**	0.0000	0.0000	**0.3394**	**0.3875**
	2	0.0000	**0.6240**	**0.1157**	0.0000	0.0636	0.0000
	3	0.0000	0.0000	**0.4692**	0.0000	0.0000	0.0000
	4	0.0000	0.0668	**0.2229**	**0.4226**	0.0000	0.0000
	5	**0.1482**	0.0962	**0.1204**	0.0000	**0.7947**	0.0000
記	6	**0.2582**	**0.1118**	0.0000	**0.1414**	0.0000	**0.3162**
事	7	0.0000	**0.1471**	**0.2863**	**0.1240**	**0.2250**	0.0000
A	8	**0.1571**	0.0000	**0.1703**	0.0000	0.0000	**0.1925**
	9	0.0000	0.0423	**0.1057**	0.0000	0.0000	0.0000
	10	0.0000	0.0435	0.0726	0.0000	0.0000	0.0000
	11	0.0000	**0.3953**	0.0000	0.0000	0.0726	0.0000
	12	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	13	0.0000	0.0000	0.0737	0.0000	0.0000	0.0000

		記事 B				
		7	8	9	10	11
	1	0.0000	0.0000	0.0000	0.0527	0.0686
	2	**0.2018**	0.0591	0.0000	**0.1332**	0.0578
	3	**0.1213**	**0.2665**	0.0000	0.0400	0.0000
	4	**0.1945**	0.0570	0.0000	**0.2140**	0.0557
	5	0.0934	0.0821	0.0000	0.0925	0.0401
記	6	0.0000	0.0000	0.0000	**0.2148**	**0.2798**
事	7	**0.1427**	**0.2091**	0.0000	0.0628	0.0409
A	8	0.0990	0.0870	0.0000	**0.1307**	**0.1703**
	9	**0.1230**	**0.1802**	0.0000	**0.1083**	0.0353
	10	**0.1267**	0.0371	0.0000	**0.1115**	0.0363
	11	0.0000	0.0000	0.0000	0.0000	0.0000
	12	**0.1400**	0.0000	0.0000	0.0000	0.0000
	13	0.0858	0.0000	0.0000	0.0566	0.0000

較では，記事 A の第 3 段落と記事 B の第 2 段落の内容が類似していると判定されたが，文単位で比較することにより，段落内に共通部分と差異部分が混在していることが分かる．

英語記事の場合と同様，日本語記事でも，内容が類似していなくても類似度が高くなってしまうことがある．比較対象記事のどの段落 (文) との間を見ても類似度が低い段落 (文) は差異部分と判断できるが，逆に類似度が高くても，それが共通部分と判断することは難しい．まだ実験を行っていないが，係り受け構造を利用することで，より詳細な比較ができる可能性があると考えている．

4 おわりに

本稿では，同一 (類似) 内容記事間の差異を検出について述べた．現在のところ，段落単位，文単位の比較についての小規模な実験のみであるが，類似度の低いものについては，それを差異と判断できると考える．一方，出現単語の一致具合を見ているだけであるため，類似度が高くても必ずしも内容が同じであるとは言い切れない．係り受け構造などの情報を利用すれば，この問題を解決できるのではないかと考えている．

提案手法は，基本的に，2 つの記事の間の差異を検出するものであるが，これを記事群内のすべての記事の組み合わせに対して適用することにより,「一部の記事に共通する差異」を検出できる．この共通の差異に注目して大量の同一内容記事群を再構成することにより，効率的に情報を提示す

ることを考えている.

参考文献

[1] Bin Liu, Pham Van Hai, Tomoya Noro, and Takehiro Tokuda. Towards automatic construction of news directory systems. In *Information Modelling and Knowledge Bases XIX*, Frontiers in Artificial Intelligence and Applications 166, pp. 208–216. IOS Press, 2008.

[2] Tomoya Noro, Bin Liu, Yosuke Nakagawa, Hao Han, and Takehiro Tokuda. A news index system for global comparisons of many major topics on the earth. In *18th European-Japanese Conference on Information Modelling and Knowledge Bases (EJC2008)*, pp. 197–213, 2008.

[3] Helmut Schmid. Probabilistic part-of-speech tagging using decision trees. In *First International Conference on New Methods in Natural Language Processing (NemLap-94)*, pp. 44–49, 1994.