設計における談話の分析と構造化

塩崎敏也・鷹合基行・武田英明・西田豊明

奈良先端科学技術大学院大学 情報科学研究科

〒 630-0101 奈良県生駒市高山町 8916-5

toshiy-s@is.aist-nara.ac.jp

あらまし　設計中の会話を録音したプロトコルデータのような議事録には、設計ノウハウやアイデアなどの設計知識が含まれており、それを構造化しユーザに読みやすく提供するような方法が望まれている。そこで本研究では、キーワードベクトル法を利用したシステムを提案しユーザへの設計情報の提示を目指す。具体的には文書の全ての行についてその前後を特定の長さだけ切り出し、それぞれのキーワードベクトルを比較することによって文書全体での話題の変化を捉える。この情報から文書の区切りを発見し文書をブロック化する。さらに得られた文書ブロックでキーワードベクトルを計算し比較することによって関係を発見し、構造化を実現する。その結果、本手法によって議事録の中に設計過程を読みとることができた。また、自転車の荷台の設計におけるプロトコルを例題に実験を行ない、本手法におけるパラメータや用いるキーワードに対する調査を行なった。

キーワード　プロトコルデータ, キーワードベクトル法, 文書の構造化

A study of document structurization method for design protocols based on keyword vector

Toshiya Shiozaki, Motoyuki Takaai, Hideaki Takeda and Toyoaki Nishida

Graduate School of Information Science,

Nara Institute of Science and Technology

8916-5, Takayama, Ikoma, Nara 630-0101 Japan

toshiy-s@is.aist-nara.ac.jp

Abstract　The aim of this research is to develop a system which can make unstructurized design information understandable for designers. We propose a method based on keyword vectors to divide the protocol data recorded during the design into important blocks of information and link them. By calculating cosine value of keyword vectors of adjacent parts of text blocks,we can detect points where topics shift to another. As a result,we can obtain text blocks each of which is related to some topic. Furthermore,we can find links between those blocks by caluculating cosine values of keyword vectors between them. We made experiments against a test case of bicycle design and analyzed the effectiveness of the method.

key words　protocol data , keyword-vector method , document structurization

1 はじめに

現在、高度な生産技術の出現によって大量に物が生産されるようになって久しい。その結果、消費者一人一人はどこにでもあるような製品にはあまり興味を示さなくなり、より消費者の立場に立って生産された製品を望むようになった。消費者が持つこのような要求に応えるために、生産者は従来の製品にない新しいアイデアを含む製品を素早く開発する必要に迫られている。このような状況において生産者は効率良く製品を設計し生産する必要がある。特に設計に着目した場合、過去の設計事例や経験を再利用することによって生産性や品質が向上するであろう。ここで過去の設計事例として、設計における談話（プロトコルデータ）の利用が考えられる。プロトコルデータとは、設計者が設計中に考えたことを全て声にだしたものを録音する方法によって得られたデータをテキストデータ化したものである。したがって、プロトコルデータには設計におけるひらめきやアイデアなどの情報が含まれており、知的な設計を支援する上で重要な役割を果たすと考えられる。しかし、そのような文書は一般的に全体が大きすぎる上、部分部分では、内容の薄い部分が多く、有益な情報を含むところを探すことが難しいという状況にある。

そこで本研究では、プロトコルデータのような構造化されていない文書の再利用を目指し、その分析と設計者が理解しやすいような形で構造化する方法を提案する。本研究で提案する構造化とは文書を、

- 文書からブロックを切り出す
- ブロック間に関係を発見する

ことを指す。またその提案した手法を実際のプロトコルデータに適用し有効性を調べる。

そして、このようなドキュメントを構造化することによって理解が促進し、有益な知識を容易に取り出せるようにし、設計をする上でより効率的で、創造的な設計につなげることを目指す。

本研究で扱うプロトコルデータは、自転車にバックパックを接続するための荷台の設計に関して、I,J,K の三人の設計者による設計中の会話文である [3]。一部を図 1に示す。このプロトコルデータを設計者の名前に基づき ijk データと呼ぶことにする。

2 文書の構造化

2.1 人間による分割

本研究では、計算機によるプロトコルデータの分析・構造化が目標である。計算機による本研究の手法の効果を確かめるために、人間がプロトコルデータを読んだ後、その中でまとまった話題と思われる部分ごとにプロトコルデータを区切る。その結果区切られた文章は人間が考えて区切ったものであるので、それぞれが意味のある文章だと言える。計算機を利用して分割するブロックも、“話題を含む意味あるブロック” として切りだす。双方を比較することが、計算機によってプロトコルデータを構造化した結果を評価する手段の一つとなり得る。以下では、人間が区切った文章にそれぞれタイトルを付けたものに「」で示した。

2.2 文書の構造化手法

本研究では、議事の流れをとらえることによってプロトコルデータの構造化を目指す。具体的には ijk データを意味のあるブロックに分割し、それらのブロックの意味上の関係を発見することによって、議事を時間的ではなく意味上の流れの形成を目指す。

双方の手法において、我々は文書の意味を捉えるための単語（あるいは熟語）の集合（キーワード）に基づく方法を提案する。特に設計に関する文書や専門文書の場合、単語の意味が文の内容に深く関係することが多く、専門用語や設計一般に関する語、物の名前などに注目することによって大きな成果が得られると考えたからである。

本研究では、プロトコルデータの中から議論の流れを捉えるために、基礎となる技術として Salton[1] らのキーワードベクトル法を用いる。そして、プロトコルデータを構造化するために大きく分けて２つの段階で処理を行なう。

- プロトコルデータからブロックを切り出す段階その際、キーワードベクトルを構成するキーワード選定について３つの実験を行なった。プロトコルデータをウィンドウ単位で見ていき、閾値を設定することによってブロックは切り出される。
- 切りだされたブロック間に関係を発見する段階ここでも、キーワードベクトルを利用する。

3 ブロックの切り出し

3.1 キーワードベクトル

キーワードベクトルは、文書中に出現する検索キーワードがベクトルの成分にあたる。そしてそれぞれの成分の値は文書中で出現するキーワード数の総和を表しているものである。本研究ではキーワードベクトルの効果は、それを構成する検索キーワードの選出に依存していること着目し、キーワードの選出について３つの実験を行なった。

- 手作業で選出
- 手作業 + オントロジーの利用
- ほぼ自動的に出現頻度順に選出

まず第 3 章第 2 節で一つ目の方法を説明し、二つ目は第 3 章第 4 節、三つ目は第 3 章第 5 節で述べる。また、人手により抽出したキーワードを用いる方法において、設計者に対する一般性を調べた。

3.2 手作業でのキーワードの選出

まず最初に、第 2 章で述べたキーワードベクトル法を利用するためにキーワードベクトルを構成する単語を ijk データから手作業で選出する。キーワードベクトルを用いてプロトコルデータ中に含まれている話題内容を捉えるためには、話題中で出現頻度が高く重要なキーワードを選出する必要がある。話題内容を捉えられるように、キーワード選定の基準として以下の単語, 熟語に注目し、収集した。

- 自転車、バックパック、接続部品などの物の名前
  - 理由:設計対象物に関する物の名前は、会話をしながら設計作業をする場合、話題の枠組を表すものであるので出現頻度も高く、キーワードとして最も重要である。

I: I guess we don't need to (inaudible)
J: that's OK so
#00:27:00
K: so we want to put it in there but I let's see if you got (inaudible)
J: yeah you'd never really be able to
K: you wouldn't be able to get your knees pedalling OK now what about maybe we ought to have a prototype that kinda has it this way
J: is a basic
I: yeah that's right this is more OK
J: would it be too funky to have it on the like projecting from the front wheel?
I: handlebars? yeah try that
J: or off this handlebar stem even because that's fixed but if it's off the handlebars you know it's like an old bike basket that way like the Wizard of Oz (laugh)
K: heavy to steer tends to
J: you could turn it long ways

図 1: プロトコルデータの一部

  - 例：handlebar,kick stand,wheel, rear wheel,backpack,strap,Velcro
- 設計の際の専門用語
  - 理由：設計対象物間の関係や対象物の性質を表す単語は、設計中の会話において物の名前ほど出現しないが、その単語に関連するところが話題の中心であるので重要である。
  - 例：specific gravity,polypropylene, centimatre,reaction
- 設計作業の際に重要と思われる語や一般的な語
  - 理由：設計者はユーザが設計対象物を使用する場合のことを考えるので、そのための単語は設計中にある程度の頻度で出現する。
  - 例：quick release,camping

名詞を中心に副詞、形容詞、動詞も含めて ijk データから全て手作業で収集した。その結果、合計で約 500 単語になった。これらのキーワード全てを使いキーワードベクトルを構成する。

3.3 キーワードベクトルを用いた文書の分割

文書検索の一手法であるキーワードベクトル法は検索キーワード群に最も近いキーワードの出現分布（キーワードベクトルの角度が小さい）を持つ文書を捜し出す。それに対して我々の文書分割の手法は、分割後の任意の隣接する文書のキーワードベクトルの角度が最も大きくなるように文書を分割する。つまりあらかじめ与えられたキーワードについて検索が行なわれた場合に検索されるべき部分とそうでない部分（キーワードベクトルの角度の違い）が特に分割後の隣接する文書間において明確に分離するような文書の分割を求めていると言える。

3.3.1 理想的な手法

文書Dを任意の分割任意の分割 (s) によって n 個の部分文章 $d_{s1}, d_{s2}, \ldots, d_{sn}$に分割し、それらのキーワードベクトルを $w_{s1}, w_{s2}, \ldots, w_{s3}$とした場合、隣合う分割された文書のキーワードベクトルの角度の変化を評価する。たとえば、

$$\sum_i (w_{si} \cdot w_{si+1}) / (|w_{si}| * |w_{si+1}|) \qquad (1)$$

が最小となる分割 s を発見する。ただし・はベクトルの内積を表す。しかしこの手法をそのまま実際の文書に適応するにはいくつかの問題点が存在する。

- 短く切った文章は内容が特殊化しているのでキーワードベクトルも特殊化してしまう。従って、分割の最小単位（一文）で切った場合にキーワードベクトルの変化が最大になってしまう。
- 任意の分割を計算することは文書の長さ $|D|$ に対して $O(|D|^n)$ 時間の計算時間を必要とし実用的ではない。

よって我々は以下に述べるような手法を用いて近似的に文書の分割点を求める。

3.3.2 ウィンドウを用いた近似法

まとまった内容を議論していると思われる文書の長さ（ウィンドウサイズ）を l、文書 Dのある部分 x の前後 lの部分を d_{xl-}, d_{xl+}とすると、その前後２つのウィンドウ部分のキーワードベクトルを比較することによって近似的に x において文書 Dの内容がどの程度変化したかを求めることができる。

ここで文書 Dの x における内容の変化を次のようにおく。

$$c_x = \frac{w_{xl-} \cdot w_{xl+}}{|w_{xl-}| * |w_{xl+}|} \qquad (2)$$

ここで w_{xl-}, w_{xl+}は d_{xl-}, d_{xl+}のキーワードベクトルをあらわす。

ウィンドウサイズを 25 行に設定して x を 1 行づつずらしながら、プロトコルデータ全体に対して c_xを求めたものの一部を図 2に示す。25 行にサイズを設定した理由は、様々なサイズにウィンドウを変化させて見た結果、25 行近辺では行数の変化に対してあまりグラフに変化が見られず安定しており、また大きさも人間がプロトコルデータを読み区切った区切り線の平均間隔に近くて、処理するにも適当な大きさであると考えたからである。

x 軸が行番号を、y 軸が余弦を表す。グラフ中の縦線が人間が分割した部分である。全体を見ると余弦の大きさを示す線の連続が山谷の様に現れ、余弦の変化が見てとれる。

本研究では c_xが小さい部分はその前後で話題の変化が激しく、あるまとまった内容を持つわけではないと解釈し、ある閾値 tで区別することにした。つまり、文書の c_xを文書の先頭から順に計算すると t 以上の部分と t 未満の部分が交互に出現するが、t 未満の部分に文書を分割するべき場所があると考える。この方法はキーワードベクトルを比較する上での文書の長さを決められた一定の長さに固定しているので、分割した文書のキー

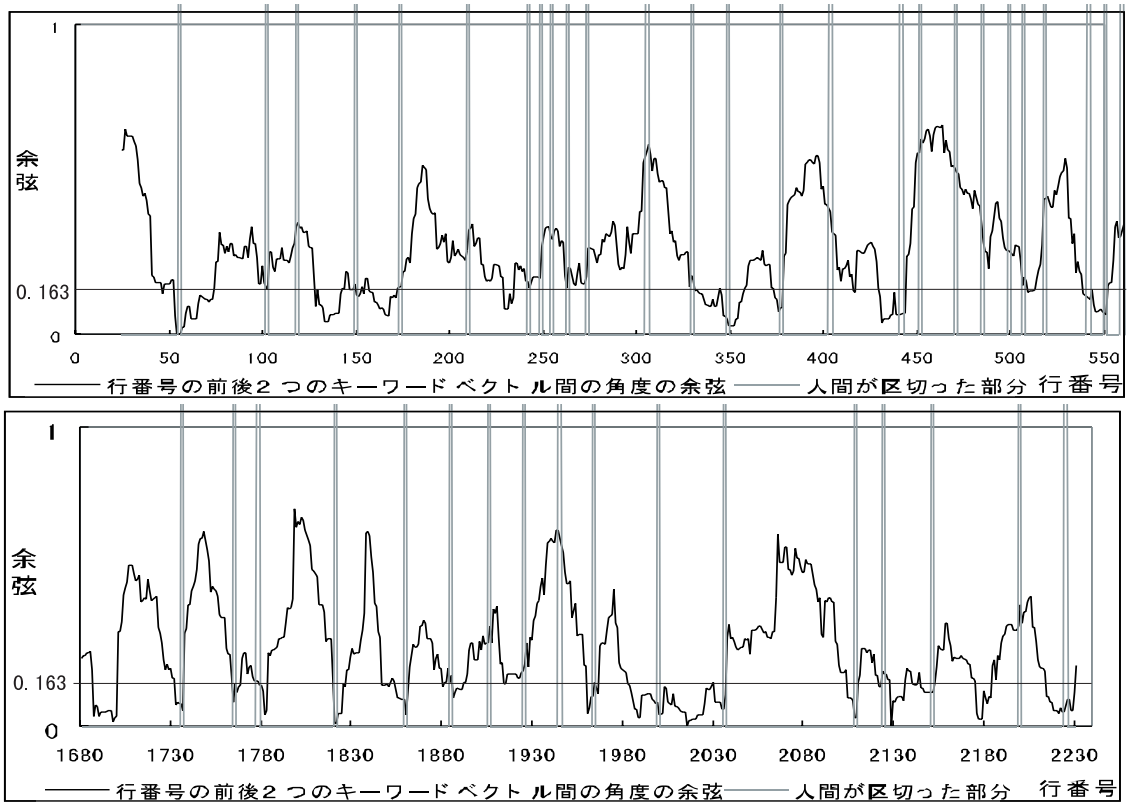

図 2: ウィンドウサイズを 25 行にした時の、各行の余弦

ワードベクトルと実際の文書の意味との誤差ををを十分小さく抑えることができる。また、文書を一回通して評価するだけでよいので $O(|D|)$ 時間の計算時間となり実用上問題ない。

実際にブロックを切り出す際には、分割するべき場所を特定するのではなく、閾値以上のまとまった内容の部分をブロックとして切り出す。したがって、閾値を低く設定すると閾値によって除外される部分が少なくなり、まとまった意味の話題を複数含む大きな意味を持つ話題ごとにブロックを切り出せると予想できる。

逆に、閾値を高めに設定するにつれて、余弦データ中にある極小値を小さいものから順に分割するべき場所と見なすようになり、詳細な話題を一つのブロックとして切り出せるようになる。その場合、一つのブロックの大きさも小さくなってくる。また、閾値の変化によって、切り出されるブロック数が変化してくる。

図 3は、閾値とブロック数との関係を示したものである。縦軸が分割されるブロック数、横軸が閾値を表す。本研究では図 3から、切り出すブロックの数が最多に近い 62 個を切り出す値 0.163 を、余弦データを閾値に設定する。というのも、そこで切り出されるブロックは一つの主要な意味を持っていると考えられるからである。図 2で、余弦の閾値 0.163 以上にあるまとまりをブロックとして切り出す。

その結果、切り出されたブロックの内 24 個が人間が区切った部分に 5 行前後の誤差を含んで一致した。その他にも、人間が区切った複数の部分に渡って切り出されたブロックが存在した。例えば、444〜505 行に渡って切り出されたブロックの部分は、人間は５つの文章に区切っているが、大きな意味を見てみると「自転車とバッ

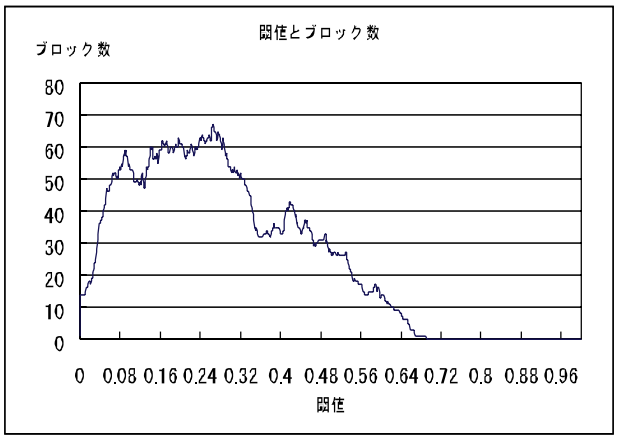

図 3: 閾値とブロック数の関係図

クパックを固定する方法の数々」という意味の部分を切り出している。このようなブロックも考えると、切り出されたブロックの内半数は切り出しが成功したと考えられる。

3.4 オントロジーを利用したキーワード選出

ここでは、キーワード選出の第 2 番目の実験について説明する。第 3 章第 2 節で述べたように、あらかじめ抜きだしたキーワードを基にキーワードベクトルを計算し文章をブロックに分割する。しかしこの手法のみだと、文章中で本来なら分割を望まない部分を分割部分と見なす危険性も存在する。例えば、「ある部品どうしをつなげる部品について」というまとまりのある話題があった場合を考える。bolt と nut は、それぞれ物をつな

げるために使う。しかし、それらが全く関係のないキーワードとして捉えられてしまうと、誤って２つのブロックを切り出してしまう可能性が存在する。このような危険を回避するために、“概念的に良く似たものをまとめる” という考え方を採り入れる必要がある。例えば、上記の例の場合 bolt,nut は、screw（ネジ）という概念でまとめておく。

このように世の中に存在するものを、ある概念がそれをより抽象的に表した概念の意味を含んでいる形で捉えたもの、つまり概念の背景にある構造を記述したものをオントロジーと呼ぶ。

そこで、オントロジーを今回用いた ijk データのように設計に関する話題に適用するために、『設計に関するオントロジー』構築を試みた。今回構築するオントロジーの機能を考え、フレームに基づくオントロジー [5] を準備した。概念はそれが文章上で表れる表現（複数可）を持つ。また、概念間は抽象-具体関係を持つ（複数可）。

オントロジー構築に当たって、抽象的な部分と、具体的な概念の部分とで別々の方法を用いた。

- 抽象的な部分

  概念が抽象的になるほど設計者自身の思考のみでオントロジーを構築することは難しくなる。そこで、この部分を構築するためには、抽象的な概念構造を詳しく収録したＥＤＲ概念辞書を利用することが効率的であると考えた。ＥＤＲ概念辞書を見ながら手作業で概念構造を構築した。

- 具体的な部分

  概念が具体的になるほど設計に関してより専門的な単語、そのオントロジーの分野を特徴付ける単語加してくる。この部分は、本実験で選んだキーワードが概念として位置するところである。専門分野の概念関係に詳しい設計者自身が構築する方が具体的な概念に関して詳しく収録していないＥＤＲ概念辞書を利用するよりも効率的である。設計者自身の思考のみで手作業で構築した。

設計に関する概念体系は、キーワードを含んで約 1100 概念から構成されたものとなった。

今回構築したようなオントロジーと類似したものにシソーラスがある。今回構築した設計に関するオントロジーは、設計に特化したシソーラスと形は同じになっている。しかし、オントロジーは、それを構築する人間の視点が深く入る。設計者によって構築するオントロジーも変化してくるであろうし、同じ人が同じものを設計するにしても視点が異なればオントロジーは変化してくる。このようなものに対して、シソーラスのような一般性が求められるようなものでは対応し切れない。

3.4.1 オントロジーの利用方法

キーワードベクトルは、本文中に出現するキーワードによって構成されるが、そのなかのあるキーワードをオントロジーを利用してある概念でまとめた場合、キーワードの代わりにその概念を概念ベクトルを構成する要素として扱う。

先述した例を用いると、nut,bolt は、screw でまとめられるので、キーワードベクトルを構成していた nut,bolt の部分は、両方とも screw に置き換わり、その結果概念ベクトルの構成要素が一つ減少することになる。この場合、本文中に nut と bolt がそれぞれ 1 回ずつ出現したとすると、screw が２回出現したと計算する。

計算機でこのような計算を行なうことができるように、多重継承も考慮に入れ（キーワード、まとめる概念、キーワードから見たまとめる概念の重み）の組をオントロジーから求めるようにした。

オントロジーを利用して ijk データから意味あるブロックを抜き出す実験として、キーワードをオントロジーにおいて全て３段上位に位置する概念でまとめることにした。その理由として、３段上位でまとめると概念ベクトルを構成する要素が約 1/5 となり効果を検証しやすい、個別に意味を考えながら取り扱うには時間と労力が必要、ということがあげられる。

この関係図を基に概念ベクトルを構成し、プロトコルデータから余弦データを求める。

オントロジーを用いた結果得られた余弦データの特徴は、全体的にオントロジーを使わないデータよりも余弦の値が大きくなる。その理由としてオントロジーを利用してキーワードをまとめた結果、概念ベクトルがここのキーワードの違いに対してあまり敏感でなくなる、ということが考えられる。

3.4.2 オントロジーを用いた結果

次にこの余弦データを意味あるブロックに分割し、オントロジーを利用する場合と利用しない場合との結果の違い比較する。その際、オントロジーを利用した余弦データを、閾値と分割ブロック数の関係のグラフから適当に求めた閾値を利用してブロックに分割してそれを比較するという方法も考えられる。しかし、両者は余弦の分布が違うので直接比較するよりも、正規化した値を出してそこから違いを読みとった方が、より正当に評価できると考えた。

図 4は、オントロジーを利用した場合と利用しない場合で得られた各行における余弦を余弦全体の平均と標準偏差から正規化したグラフの一部である。縦軸が正規化後の値で、横軸が行番号、縦線が人間が区切った部分を示している。

オントロジーを利用することによって、新たにブロックとして切り出せる部分が出現した。逆に、従来のブロックを形作る山がなだらかなるが、山と認識される部分の減少度合は、小さいと考えることができる。このグラフから、オントロジー利用した時に余弦値の変化が大きいと思われる特徴を示しているところについて、いくつか例を抜きだして分析する。

- 行番号 1470 ～ 1562

  文章中に出現する (キーワード 出現回数) が (tray 6),(screw 2),(tube 8) が (implement 15) にまとめられた。この行番号の部分は、文章タイトル「トレーの構造について」の「トレーを固定する方法について」の二つの部分とほぼ同じ位置にあたる。詳しく見ると、tray と tube は、両タイトルの文章中ににわたって出現するが、screw は後者のタイトルの文章中みに出現していることが分かる。新たなブロックの切り出しが特に後者タイトルの文章中において起こる可能性が高いと思われる。

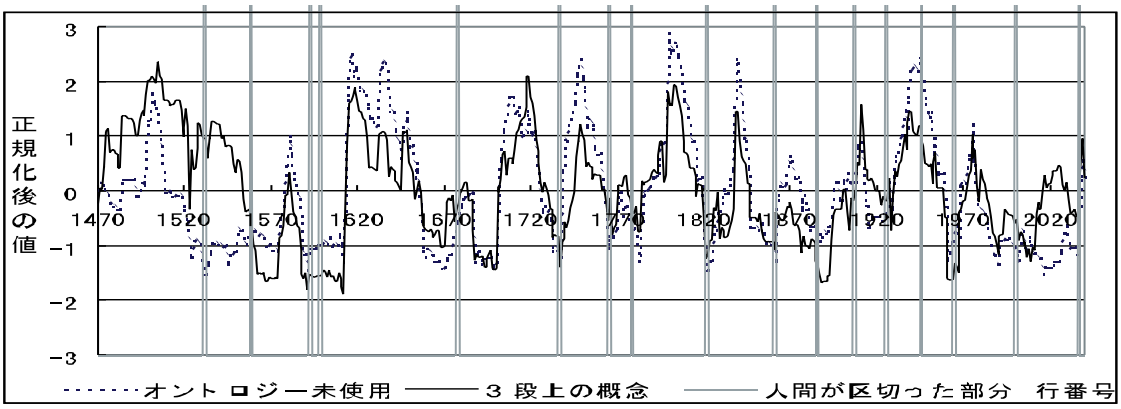

図 4: 3段上の概念でまとめた時と、オントロジーを使わない時との比較

- 行番号 2010 ～ 2038
  文章中に出現する (キーワード 出現回数) がそれぞれ、(wing nut 1),(clip 1),(cable 1),(screw 1) が (接続具 1),(implement 3) に、まとめられた。この行番号の部分は文章タイトル「盗難防止のロック機構について」の位置にあたる。様々なキーワードが一つの話題に対して共通のキーワードとしてまとめられ、新たなブロックが切り出されると思われる。
- 行番号 1560 ～ 1645
  文章タイトル「バックパックとトレーを結び付ける方法について」、「スケジューリング」、「トレーの詳細について」の部分にあたる。オントロジーを利用しても、文章中に出現する (キーワード 出現数) が (tray 5) が (implement 5)、(strap 7) が (ひも類 7)、(inch 7) が (数量 7) になるなど、文章の意味を表すキーワードについてまとめられなかった。従って、全体的な余弦値の上昇の影響でこれらのキーワードによって出現していたブロックの山が相対的に低くなってしまった。

3.5 出現頻度を基に半自動的にキーワードを選出

キーワード選出の第３番目の実験として、設計者が行なうキーワードの選定の労力削減の可能性を探るために、ijk データを構成している単語から出現頻度に着目してキーワードを選出を試みた。統計的なキーワード抽出の手法として $tf \cdot idf$ 法があるが、この方法は、分割された複数のブロックが存在する場合に重要語の選定に役立つが、今回のブロックを切り出す際には適切な手法ではないと判断し使用しなかった。

具体的には純粋にプロトコルを構成している単語を全てとりだしそこから、冠詞 a,the、代名詞 they,it、前置詞 at,on、動詞の中でかなり一般的な語 have,is、などを取り除いたもの全てをキーワード候補とした。キーワード候補数は ijk データで 1600 であった。次にそこから出現頻度順に手作業で選出した数とほぼ同数の上位 500 個取り出したものをキーワードとした。このキーワードをもとにキーワードベクトルを構成し各行において余弦を計算した。その結果と、手作業で選出したキーワードから得られたブロックとを比較する。

図 5は、出現頻度順に半自動的にキーワードを選んだ場合と、手作業でキーワードを選んだ場合で得られた各行の余弦を余弦全体の平均と標準偏差から正規化したグラフの一部である。縦軸が正規化後の値で、横軸が行番号、縦線が人間が区切った部分を示している。新たなブロックを切り出す可能性がある部分と逆にブロックが捉えられなくなったところについて見ていく。

- 行番号 1640 ～ 1700
  この部分に 7 つ出現する two という単語がキーワードとして捉えられるようになり、ブロックとして切り出されるようになった。タイトル「プラスチックの軽さと荷台の大きさについて」の話題であるが、内容と直接関係のないキーワードによって一つの話題と捉えた例といえる。
- 行番号 1787 ～ 1820
  「rear stay の高さについて」の部分であるが、細かく見ると短い範囲に (think 2),(forty 2),(plus 2) などの出現頻度の高いキーワードが出現することとで、本来のキーワードの inch の効果が減少し、細かく話題が変化していると計算機は捉えたと思われる。ブロックとして切り出せない。
- 行番号 1751 ～ 1776
  「bungee strap のコストについて」の部分であるが、ここでもまた OK,twentyfive などが、本来のキーワードである cost,strap の効果を減少させ、細かく話題が変化していると捉えられた例といえる。

半自動的に出現頻度順に選び出したキーワードから新たに切り出されるブロックについて見てきたが、ブロックとそれを切り出す要因となったキーワードとはあまり関係がなかった。また、ブロックとして捉えられなくなった原因がブロックの意味と直接関係のないキーワードによって起こっている。

3.6 オントロジーの交換

キーワードを基にした本研究の手法を同じテーマの設計者に対して一般的であるか、検証してみる。つまりキーワード、オントロジーの再利用性である。また、オントロジーの違いによって切り出されるブロックがいかに変化するかということも検証する。

我々が実験を進めるうえで使用してきた ijk データとは別に『自転車にバックパックを取り付けるための部品の設計』という同じテーマに関して、F,G,H の三人組で設計に取り組んだ時の２つ目のデータ、fgh データが存在する。そのプロトコルデータについて、ijk デー

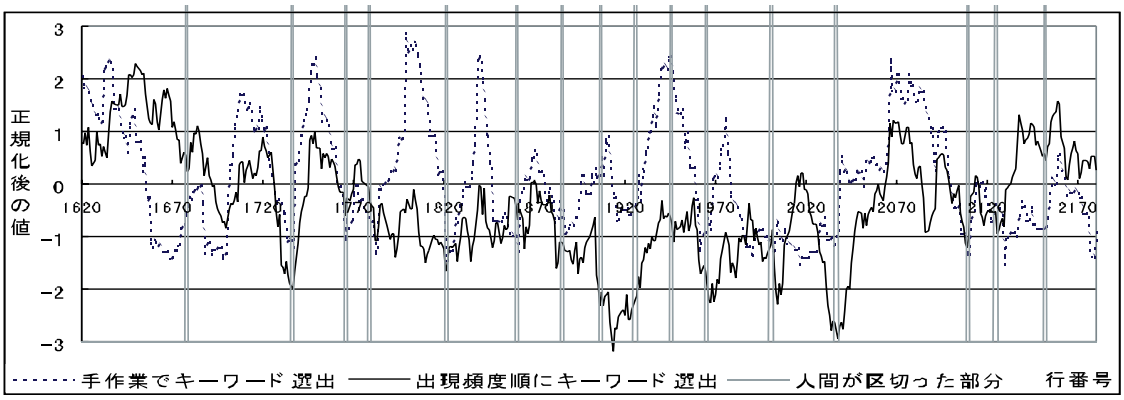

図 5: 出現頻度順にキーワード選択と手作業でキーワード選択との比較

タを分析した人とは別の人が、文章中からキーワードを約 400 個選定したあと、約 800 の概念を持つ fgh オントロジーを作成した。その fgh オントロジーを ijk データの分割に際に利用することにした。つまり fgh オントロジーが、過去に構築したオントロジーにあて、ijk データをこれから参照するの新しい設計情報にあてる。fgh オントロジーの特徴は、中段の概念が少なめで荒く階層構造が作られている。もちろん、fgh で選出したキーワードには、ijk データで選出したキーワードがかなり含まれているが、含まれていないものも数多く存在する。

fgh オントロジーの構造を考慮に入れて、fgh オントロジーを利用して fgh キーワードを一段上の概念でまとめたものと、ijk オントロジーを利用して ijk キーワードを 3 段上の概念でまとめたものとで、ブロックの分割結果を比較してみた。

グラフ全体を見ると、前半部では、両者の形状に何箇所かで相違が見られるが良く似た形状であると読みとれる。特に会話が始まって 1/4 経った辺りでは、良く似た形をしている。後半部では、両者の形状に著しい違いを読みとることが出来る個所も多くなってくる。ijk において、大きな山として捉えられるところは、形状に違いがあるが、fgh オントロジーの方も山を形作っている場合が少なくない。ijk と fgh オントロジーを利用した時とで、グラフの形状に違いが見られるところは、キーワードベクトルを構成しているキーワードの違いによるところが大きかった。

4 キーワードベクトルを用いたブロック間の関係の発見

この章ではキーワードベクトルを用いて切り出されたブロック間の意味的な関係を発見する方法について述べる．ブロック集合 $\{d_1, \cdots, d_n\}$ の 2 ブロック d_i, d_j に対して，計算したキーワードベクトルを w_i, w_j とすると，これら 2 つのキーワードベクトルを比較することによって 2 ブロック d_i, d_j の関係を近似的に求めることができる．ここではキーワードベクトルの比較方法として前述のブロックの切り出しで用いた方法と同じくベクトルの角度の余弦を求める方法を用いている．つまり 2 ブロックのキーワードベクトル w_i, w_j に対して計算されるベクトルの余弦

$$c_{ij} = \frac{w_i \cdot w_j}{|w_i| * |w_j|}$$

表 1: ブロック間の関係

ブロック番号	ブロック番号	関係の深さ
37	46	0.833
47	52	0.809
4	16	0.666
26	40	0.615
24	40	0.611
39	40	0.600
37	49	0.592
24	27	0.592
52	59	0.561
24	26	0.552
25	60	0.535
46	49	0.518
52	60	0.516
4	45	0.514
17	26	0.510
13	23	0.503
38	45	0.479
47	60	0.478
47	59	0.475

の大きいブロックほど内容上の関係が深いと考える．また関係の深いブロックについてベクトルの成分を比較することによって，どういう意味でその 2 ブロックが関係が深いのかを分析することができる．この方法を特にプロトコルデータから切り出されたブロック群について行なうことによって，プロトコルデータ全体を線形に読むだけではなく意味上の関係をたどることができ，読者はプロトコルデータの重要かつ必要な部分を早く発見し理解することができるようになる．

ijk データを構造化するために、切り出されたブロック全体に対して以下の手順で処理を進める。また、構造化するブロックとして手作業で選んだキーワードのみを使った手法（オントロジーは使わない）を選んだ。

- ２つ１組として、全組合せのキーワードベクトルを計算する
- 類似度が高いブロックの組合せから順にソートし関連付ける

ソートした結果を表 1に示す。この表から、ブロック 37 とブロック 47 の様に関係が深いブロックが多いブロックと、ブロック 3 やブロック 19 の様に関係の深

いブロックがあまり存在しないブロックがあることに気付く。

- ブロック 47, ブロック 52

  ２つのブロックに共通して出現するキーワード inch が両者を関係の深いブロックであると認識した。ブロック 47 は留めるために必要な部品のサイズ、について述べていて、ブロック 52 は何か支えるためのもののサイズ、について述べている。内容を見てみると直接のつながりはないと思われる。しかし、「もののサイズについての話題」という観点から見ると深い関係があるといえる。２つのブロックを人間が区切った文章と比較してみると、ブロック 47 は「tray の詳細（サイズ）について」の文章の中間に位置している。ブロック 52 も「rear stay の高さについて」の文章の中間に位置している。

次に関係が深いと計算されたブロック

- ブロック 38 「トレーを使ってバックパックと自転車を結び付けることについて提案」
- ブロック 43 「トレーの構造について」の一部
- ブロック 45 「トレーの構造について」の一部
- ブロック 47 「トレーの詳細について」の一部
- ブロック 54 「トレーのコストについて」

を見る。どのブロックもキーワード “tray” が出現しているので、トレーについて関係が深いと捉えた。ブロックが位置する部分のタイトルを見るとトレーを利用することについて、トレー使用の提案からトレーの詳細な設計にいたるまでの一連の設計の流れが読みとれる。ブロックの内容を個別に見てみると、38,54 ではトレーについて詳しく述べられているので、トレーに関する情報を得やすいが、他のブロックはプロトコルデータの性質上、そこだけ読むとはっきりとした情報は得にくい。しかしながら、関係が深いブロックとして 38 の存在が分かれば、38 から得たトレーに関する情報をもとに他の３つのブロックについてより多くの前提知識を持って文章を読むことができる。また、それぞれのブロックに番号をつけているので、番号の若い順からブロックを眺めることによって、あるものに関しての設計の流れを捉えやすくなるといえる。設計の過程は、

“問題提起”→“提案”→“展開”→“評価”→“決定”

という要素で成り立っていると考えられている [4]。上記の例では、38 → 43, 45, 47 → 54 がちょうど設計の過程であるとされる “問題提起”,“提案”→“展開”→“評価”“決定” に該当すると、読みとることができる。したがって、プロトコルデータ全体にも設計の過程が表されているということが分かる。

5 関連研究

本研究で提案する手法と関連の深い研究のひとつに、TextTiling[2] がある。TextTiling では、文章を分割する際に本研究と同じくキーワードベクトルの角度の余弦を計算することによって、文書を複数の節に分割している。本研究の位置付けとして、TextTiling がキーワードベクトルを求める単位をまとまった数の token であるのに対し、我々は会話文を扱うということに着目しそれを構成するまとまった数の文を単位とした。また、TextTiling では重点を置いてなかったキーワードベクトルを構成するキーワードの選定にも、注意を払った。これによって、文書分割の手法の設計に関する文書への拡張を図った。TextTiling では、本研究で提案する文書間の関係については触れられていない。

6 まとめ

プロトコルデータのような構造化されていない文章を、その文章でキーワードと考えられるものを基にキーワードベクトルを用いてブロックを切り出した。その際キーワード選択が重要であることが分かった。また、オントロジーを利用することによってキーワードのみでは切り出せなかった意味のあるブロックを切り出せたことを述べた。また、半自動的に出現頻度順にキーワードを選択しても良い結果を得ることができなかった。

切り出したブロック間に関係を発見し、関係の深いもの同士を関連付けた。この２段階の構造化が、設計に関する議論の流れをつかむ一つの手法であることを示した。

また、プロトコルデータを構造化することによって、その中には設計過程のモデルとされるものをそのまま含んでいるということを確かめることができた。『自転車にバックパックを取りつけるための部品の設計』というテーマに対してもそれが当てはまることを示した。

以上より、今回提案したシステムは、設計における会話文のような議論に流れがある文書の構造化に有効であるといえる。

謝辞

なお、本研究の一部は、IMS GNOSIS(製造知識の体系化)、および日本学術振興会未来開拓学術研究推進事業「シンセシスのモデル論」プロジェクト
(JSPS-RFTF96P00701)、の研究費によって実施された。

参考文献

[1] G.Salton and M.J.McGill. *Introduction to Modern Information Retrieval.* MacGrew-Hill, 1983.

[2] Marti A. Hearst. Texttiling:segmenting text into multi-paragraph subtopic passages. In *Association for Computational Linguistics*, pp. 33–64, 1997.

[3] Henri Christiaans Nigel Cross and Kees Dorst. *Analysing Design Activity.* 1995.

[4] 武田英明, 冨山哲男, 吉川弘之. 知的 CAD の開発のための設計過程の分析と論理による形式化. 精密工学会誌, pp. 57(6):1047–1052, 1991.

[5] T.R.Gruber. Ontolingua: A mechanism to support portable ontologies. Technical Report 91-66, Knowledge Systems Laboratoty Stanford University, 1992.