

# 卒業論文

## Web内容マイニングによる 複数キーワードに対する 3D有向グラフを用いた発想支援

Idea Generation Support Using 3D Directed Graph  
for Multiple Keywords by Web Content Mining

富山県立大学 電子・情報工学科

1615052 山元 悠貴

指導教員 奥原 浩之 教授

令和2年2月17日



# 目 次

## 圖一覽

## 表一覽

# 記号一覧

以下に本論文において用いられる用語と記号の対応表を示す.

| 用語                                 | 記号                             |
|------------------------------------|--------------------------------|
| 先行作業                               | $i$                            |
| 後続作業                               | $j$                            |
| プロジェクトの総作業数                        | $n$                            |
| 従事者グループ                            | $k$                            |
| 依頼候補の従事者グループ数                      | $w$                            |
| 作業の所要時間                            | $t_{ij}$                       |
| 作業を従事者グループに依頼したときの費用 (ファジィ・ランダム変数) | $\tilde{c}_{ik}$               |
| クリティカルパスを選択する 0-1 変数               | $x_{ij}$                       |
| 依頼する従事者グループを選択する 0-1 変数            | $y_{ik}$                       |
| ファジィ・ランダム変数のメンバシップ関数               | $\mu_{\tilde{c}_{ik}}$         |
| 中心値 (平均値)                          | $\bar{d}_{ik}$                 |
| 左右の広がりパラメータ                        | $\beta_{ik}, \gamma_{ik}$      |
| ファイジィ目標                            | $\tilde{G}$                    |
| ファジィ目標の最良値                         | $g^0$                          |
| ファジィ目標の最悪値                         | $g^1$                          |
| ファイジィ目標のメンバシップ関数                   | $\mu_{\tilde{G}}$              |
| ファジィ目標は満たされる可能性の度合い (可能性測度)        | $\Pi_{\tilde{c}_{ik}}$         |
| 満足基準値                              | $h$                            |
| 擬逆関数                               | $L^*(h), \mu_{\tilde{G}}^*(h)$ |
| 確率変数の分布関数                          | $F$                            |



## はじめに

### § 1.1 本研究の背景

近年、人工知能（AI）などの情報処理技術はめざましく発展しており、記憶や認識、データ収集やその処理などは人間を凌駕しつつあり、人間から AI に代替される事が多くなると考えられる。そのような状況の中で、AI による代替が難しいとされる「人間が発想する」ことの重要性がより高まってゆくと考えられる。そこで、人間が発想することを支援する発想支援システムの重要性も高まっている。

現在、情報処理技術の発達に伴って、コンピュータが人間の創造的な問題解決・思考活動を支援する発想支援システムの研究が進んでいる。よって、これからの時代はより創造的な問題解決であるアイデア発想が重要になってくると考える。創造活動をする際、人間は言葉で表現することが多く、その言葉を用いて自分の発想を整理して修正を行う。実際、認知心理学では、人間にとって思考と言語は深い関連があると言われている [?]. そのため、コンピュータが発想支援を行うためには、人間の言葉である自然言語を知る必要がある。

しかし、自然言語を機械に理解させるのは困難であり、様々な機械独特の自然言語の分析方法が用いられている [?]. 典型的には発想支援システムはコンピュータを中心にしたシステムとして実現される。文章を書いたり、図を使ったり、記憶を呼び起こしたりすることは創造的に考えるための条件だといってよいだろうが、これらの行為を実行するうえでの人間の認知的な制限をコンピュータの記憶装置や出力画像などを用いて軽減することができる。

「発想支援システム」とは、「AI コンピュータ、ニューロコンピュータ、ファジーコンピュータや光コンピュータなどの最先端コンピュータを用い、人間の顕在的能力をも活用できる対話的な創造的な問題解決支援環境を持つシステム」[富士通国際研 91] のことである。つまり、発想を支援することは、人間の創造的な問題解決を支援することと考えられる。そのため、機械が発想支援を行うには、人間が考えるために使う手段である自然言語の分析方法を利用する必要がある。

また、現状の発想支援には一つのキーワードに対して共起ネットワークなどを用いて関連する単語をネットワーク化して発想を広げていくシステムがある。これらの特徴としてコンピュータの得意なビッグデータを扱うことで発散的思考により多くの情報から人間が発想支援に役立てる事ができる。また、他には作曲支援システムや、キーワードに対して無関係なワードをランダムに提案して発想支援につなげるシステムもある。



発想支援システムの構築は計算機の最も高度な知的利用技術のひとつで、計算機が出現して以来の計算機科学者、人工知能研究者の究極の夢のひとつであり、これについて心理学、社会学、工学などの各方面から様々な試みが行われている。情報技術の進歩により従来は実現が困難であった計算機によるアイデア発想支援が具体化できるようになってきている。また産業界においても個人個人の独創性を引き出せるような知的環境の構築が望まれており、様々な発想支援システムの開発が試みられ、一部は既に実用化されているものもある。

## § 1.2 本研究の目的

情報処理技術の発達に伴って、コンピュータが人間の創造的な問題解決・思考活動を支援する発想支援システムの研究が進んでいる。これからの時代はよりアイデア発想が重要になってくると考える。創造活動をする際、人間は言葉で表現することが多く、その言葉を用いて自分の発想を整理し、修正を行う。実際、認知心理学では人間にとって思考と言語は深い関連があると言われている。そのため、人工知能が発想支援を行うためには、人間の言葉を知る必要がある。しかし、自然言語を機械に理解させるのは非常に困難であり、機械独特の自然言語の分析方法が用いられている。

発想を支援することは、人間からアイデアが出やすくすることである。そのため、機械が発想支援を行うには、人間が考えるために使う手段である自然言語の分析方法を利用する必要がある。従来の発想支援では、単語の共起ネットワークを用いたものやKH coderを用いた階層的クラスタ分析を用いて単語の関連性を分析した研究がある。これらは入力の一つの単語に対してネットワークを作成するシステムが多かった。

そこで現状の発想支援について複数のキーワードからの発想支援に着目した。複数のキーワードからの発想支援にすることで一つのキーワードの場合より詳細な情報を入力することができるので実用性が高まると考えた。また、共起ネットワークを用いるネットワーク図は無向ネットワークなのでノード間の関係はエッジの大きさだけなのに対して、今回用いるネットワークは単語の順序をから作成しているので、有向ネットワークである。有向グラフはノード間の関係がエッジの大きさだけでなくエッジの向きも考慮しているので情報の量がより多くなっている。

本研究では、Google 検索を用いた複数のキーワードからの発想支援について考える。既存の発想支援システムでは一つのキーワードから発想を広げていくものが多かった。そこで、複数のキーワードから関連する単語を用いることで、より効率が良く実用性が高い発想支援につながると考えた。また、複数のキーワードを用いることで詳細なデータを分析できる。複数のキーワードに対して、スクレイピングによりそのキーワードに関するインターネット上のテキストデータを収集して、自然言語処理によりテキストデータを重要だと思われる単語群にし、3D 有向グラフに適応することで発想支援につなげることを目的とする。

## § 1.3 本論文の概要

本論文は次のように構成される。

- 第1章** 本研究の概要と目的について説明した。概要は、人間が発想することの重要性和コンピュータが発想支援するために自然言語処理が使われていることを示した。目的は、複数のキーワードに対して関連性のある単語を3D有向グラフとして表示することであった。
- 第2章** 既存の発想支援システムと課題について述べる。AIブレストパークなどの発想支援システムの紹介とそれらのシステムに使われる技術とそれらの課題について述べる。
- 第3章** 本研究の発想支援システム構築に用いる技術について述べる。用いる技術としてはサイバー空間からのテキストデータを収集するためのクロール、スクレイピング、収集したテキストデータを分析する自然言語処理を用いたテキストマイニング、などである。
- 第4章** ベイジアンネットワークを用いた提案手法についてまとめる。提案手法として2つのキーワード間の関係について調べる。名詞のみにしたテキストデータの遷移確率を用いてネットワークを生成してそのネットワークのキーワード同士の最長経路を求め発想支援する。
- 第5章** まとめと今後の課題を述べる。



# 発想支援システムの概要と課題

## § 2.1 発想のプロセス

人間が発想する際にはいくつかのプロセスが存在する。発想支援システムを作成する際に必要な発想のプロセスを説明する。人間の発想のプロセスを次の3つのステップに分けて考える。

### 情報の収集

発想する際、まず情報を集めることから始める。これはインターネット上のTwitterやFacebookなどのSNSや書籍や動画など多岐にわたる。今回のシステムではキーワードに関連するURLからインターネット上のテキストデータという情報を収集する。

### アイデアを発散させる

このプロセスでは発想の元となるキーワードに対して連想して関連する言葉などをたくさん出すことである。今回のシステムではキーワードに関連するURLからインターネット上の収集したテキストデータに対して

### アイデアを収束させる

#### Step1 :

まず、入力された2つのキーワードに対してGoogle検索からURLを検索結果の1ページ分収集してURLリストとしてテキストファイルにして保存する。収集したすべてのURLに対してBeautifulSoupというライブラリを使ってそれらサイトのHTMLを収集する。

#### Step2 : 自然言語処理の前処理

次に、自然言語処理における前処理を行う。収集したHTMLに対してすべてのHTMLのタグを消去してテキストデータのみ抽出する。抽出したテキストデータに対してMeCabを使ってクリーニング処理と形態素解析を行い不要な記号や数字の除去し、名詞・動詞・形容詞を取り出す。

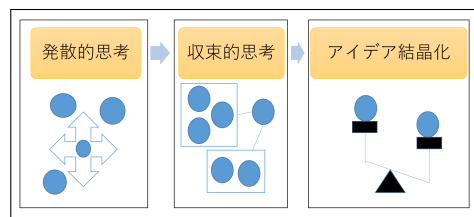


図 2.1: 発想のプロセス

#### Step3: 名詞群からの隣接行列の作成

取り出した単語群の順序関係を隣接行列にする。このとき、すべての単語で隣接行列を作成すると行列の大きさが必要以上に大きくなるので Pagerank のアルゴリズムによってより重要な単語だけを取り出し隣接行列を作成する。

## § 2.2 発想支援システムとは

「発想支援システム」とは、「AI コンピュータ、ニューロコンピュータ、ファジーコンピュータや光コンピュータなどの最先端コンピュータを用い、人間の顕在的能力をも活用できる対話的な創造的問題解決支援環境を持つシステム」[富士通国際研 91] のことである。つまり、人間がアイデア発想をする際に、発想のプロセスを支援するコンピュータシステムのことである。発想支援の解釈を以下に示す。

### 発想支援システム

発想支援システムは文字通り人間の創造的問題解決における発想のプロセスを支援するコンピュータシステムである。この発想のプロセスでは、ある問題に対して、発散的思考・収束的思考・アイデア結晶化の3つのステップの流れで行われる。

- (1) 関係のあると思われることがらを全て洗い出す (発散的思考)
- (2) それらの関係の整理・本質の追求 (収束的思考)
- (3) 評価・決断 (アイデア結晶化)

が行われる。これまでにこれらのプロセスを支援するためのさまざまな発想支援システムが研究されている。本研究のシステムでは発散的思考で Google 検索からテキストデータを収集して、収束的思考で PageRank で絞ることで発想支援に生かしたい。

発想支援システムの構築は計算機の最も高度な知的利用技術のひとつで、計算機が出現して以来の計算機科学者、人工知能研究者の究極の夢のひとつであり、これについて心理学、社会学、工学などの各方面から様々な試みが行われている。情報技術の進歩により従来は実現が困難であった計算機によるアイデア生成支援が具体化できるようになってきている。また産業界においても個人個人の独創性を引き出せるような知的環境の構築が望ま

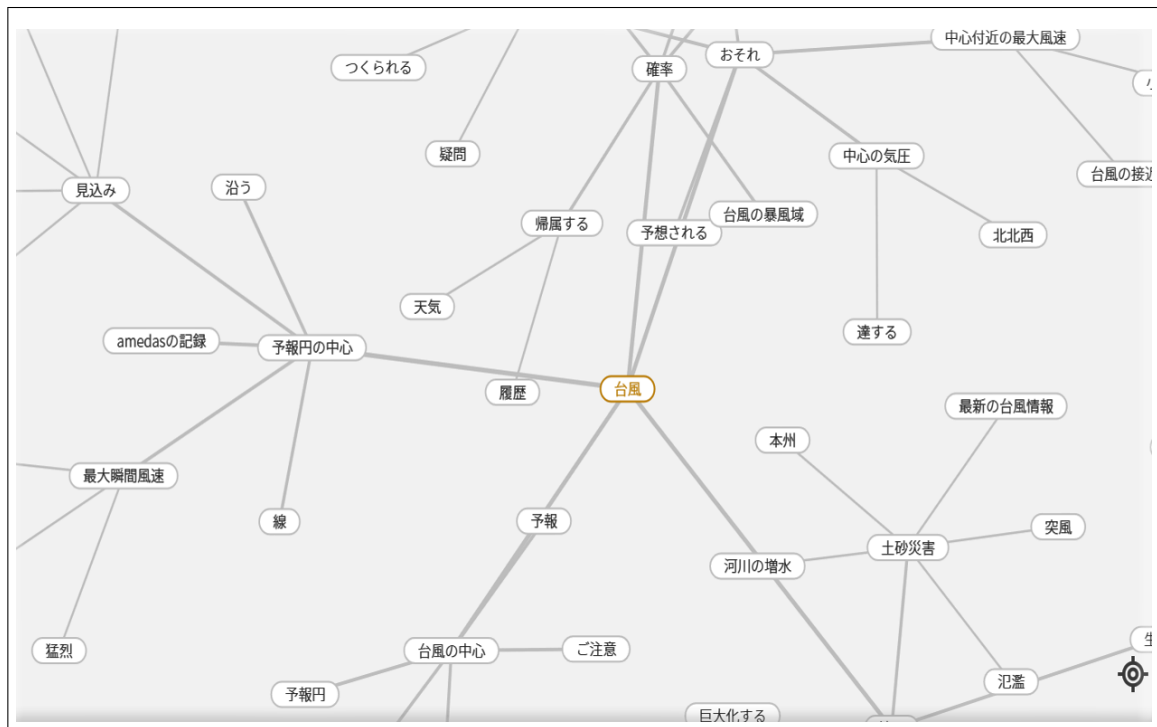


図 2.2: 既存の発想支援システム AI ブレストパークのひらめきマップの一部

れており、様々な発想支援システムの開発が試みられ、一部は既に実用化されているものもある。

既存の発想支援システムには AI ブレストスパークのひらめきマップや、ブレストアイデアがある。ひらめきマップはキーワードを入力するとその単語に共起する関連語を表示して発想支援に役立てるというものである。ブレストアイデアはキーワードを用いたワード、フレーズの生成をするシステムである。また、KH coder を用いた階層的クラスタ分析を用いて単語の関連性を分析した研究がある。

アイデアは、複数の知識の有意義な組合せである場合が多い。コンピュータで有意義な組合せを自動生成出来れば、それは「発想システム」となる。しかし、組合せの数は膨大となる。平仮名数が約 70 として、100 文字から成る文章（平仮名の順列）の総数は 70 の 100 乗（宇宙の原子の推定数より多い）となり、コンピュータでも現実的な時間内で生成するのは不可能である。優れた文章を書く人間の脳では、何らかの方法で言葉の結び付きを選択している。

しかし、その方法は明らかになっていない（従って、プログラム化出来ない）。そこで、コンピュータが複雑なアイデアを出力する「発想システム」は難しいと考え、アイデアのヒントを提示するシステムとして「発想支援システム」の開発が試みられた。

## § 2.3 既存の発想支援システムと課題

既存の発想支援システムには AI ブレストスパークのひらめきマップや、ブレストアイデアがある。ひらめきマップはキーワードを入力するとその単語に共起する関連語を表示して発想支援に役立てるというものである。ブレストアイデアはキーワードを用いたワード、



図 2.3: AI ブレストパークのブレストアイデア

フレーズの生成をするシステムである。AI ブレストパークはユーザーがキーワードを入力してそのキーワードに対する共起ネットワークを表示するシステムである。このシステムはとにかく多くの共起する関連語を表示するので、2.1 の発想支援システムにおける発散的思考であるといえる。また、他には KH coder を用いた階層的クラスタ分析を用いて単語の関連性を分析した研究がある。

上記の発想支援システム AI ブレストパークは発散的思考なのに対して、本研究では収束的思考であるので一概にどちらがいいかは評価できないと考える。発想支援システムにおいて目的によって最適なツールが変わってくるといえる。発想支援システムの例として AI ブレストパークの実行結果の例を図 2.1, 図 2.2 に示す。

本研究では、Google 検索を用いた複数のキーワードからの発想支援について考える。既存の発想支援システムでは 1 つのキーワードから発想を広げていくものが多かった。また、共起ネットワークを用いるネットワーク図は無向ネットワークなのでノード間の関係はエッジの大きさだけなのに対して、今回用いるネットワークは単語の順序をから作成しているので、有向ネットワークである。有向ネットワークはノード間の関係がエッジの大きさだけでなくエッジの向きも考慮しているので情報の量がより多くなっている。ここで、既存の発想支援システムの紹介とそれぞれのシステムのメリット、デメリットを説明する。

### AI ブレストパークのひらめきマップ

このシステムは実際に TIS 株式会社が運営する月額制の発想支援システムである。これは、ユーザーはキーワードとなる単語や言葉を入力すると、そのキーワードに共起する関連語のネットワーク図を表示するというシステムである。このシステムのメ

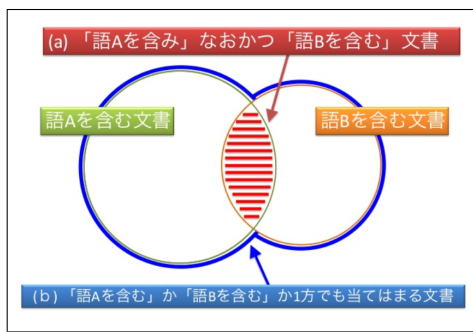


図 2.4: Jaccard 係数の仕組み

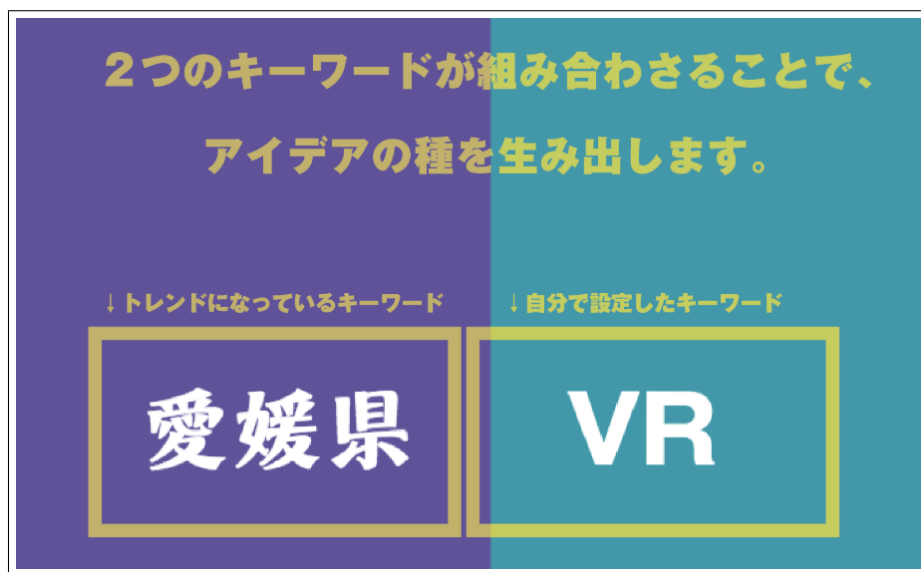


図 2.5: ひらめきエンジンの例

リットは多くの関連語が比較的早く結果が表示される点とユーザービリティに優れている点である。共起ネットワークについて、Jaccard 係数という集合の類似度を表す指標を使って作られる。このシステムのデメリットとしては、結果に関して量を優先しているので共起ネットワークの数値などは見ることができない点が挙げられる。

#### AI ブレストパークのブレストアイデア

コトバはすべて概念の塊であるので、キーワードとひらめきマップなどのコトバをランダムに結合することで、新しい概念を無数に生成することができる。予想外の新しい切り口を続々浴びることで、思考の枠が揺らされ発想が支援される。意外な結合が生む新しい価値に着目している。このシステムのワード生成はキーワードに対して、選択したカテゴリーのコトバがランダムに自動接着して新しいワードを生成、リストが表示される。フレーズ生成は選択したカテゴリーのコトバがランダムに自動接着してフレーズを生成、リストが表示される。

#### ひらめきエンジン

株式会社乃村工藝社と株式会社カヤックが共同開発した発想支援アプリケーションで



ある．リアルタイムで抽出するトレンドワードと，あらかじめ設定していたワードをランダムに組み合わせ，ビジュアル化し続けることでアイデアの種を生み出すシステムである．表示されるトレンドワードは，Google(日英)・Twitter(日英)・Yahoo!(日英)・Wikipedia(日英)・はてなブックマーク(日)のソースのAPIを使用している．設定したいキーワードは20個まで入力可能で興味のあるトレンド情報を自由に選択可能である．また，トレンドワードのかわりにCSVの辞書を設定することで，さらにカスタマイズ可能である．このシステムのメリットは，組み合わせる二つのワードのうちの一つをWebから抽出するトレンドワードにすることで意外な組み合わせで最新情報をもとにした発想支援が可能である．デメリットとしては，トレンドワードと自分で設定したワードとの関係性は全く考慮されていない点が挙げられる．

# インターネット上のテキストデータの分析

## § 3.1 スクレイピングによるテキストマイニング

アイデア発想において人間は自然言語から思考して発想することが一般的である。そこでインターネット上にあるテキストデータを自然言語処理することを考える。現代社会においてインターネット上の情報量は莫大になっており、今後も増え続けることが予想される。このインターネット上の情報を収集して分析することで発想支援に生かせる考える。発想支援において重要なことはキーワードからより関連度の高い単語をより多く表示させることである。そこで、より良いデータを多く収集するためにインターネットからテキストデータを収集することとする。

今回、キーワードごとに Google 検索から上から何件分かの URL を取得し、その URL からテキストを抽出してそのテキストに対して自然言語処理を行い、必要な単語を取り出す。その単語群から 3D グラフを作成する。以下にテキストマイニングとそれに付随する技術について説明する。

図 3.1 にデータ収集に用いるテキストマイニングの説明を示す。図 3.2 に形態素解析エンジンの MeCab, ChaSen, JUMAN, KARASI の比較表を示す。

### テキストマイニング

テキストマイニングとは、大量かつ多量なデータを様々な観点から分析し、役に立つ情報を取り出そうとする技術である。アンケートの自由記述や SNS の口コミ、Amazon などのレビューの分析などに用いられている。インターネット上のテキストを用いることで大量のデータを活用することができる。テキストマイニングによってテキストデータからどのような単語が多いのか、どんな特徴があるかなど分析できる。テキストマイニングの流れとしてはテキストの抽出、そのテキストに対して自然言語処理の前処理である形態素解析や単語の正規化、ストップワードの除去などをして残ったノイズのできるだけ少ないテキストデータを扱う。

まず、収集した文章に対して HTML タグや JavaScript のコードを取り除くクリーニング処理をする。

### 形態素解析

その次に形態素解析を行う。形態素解析とは文を形態素ごとに分解する技術である [3]。

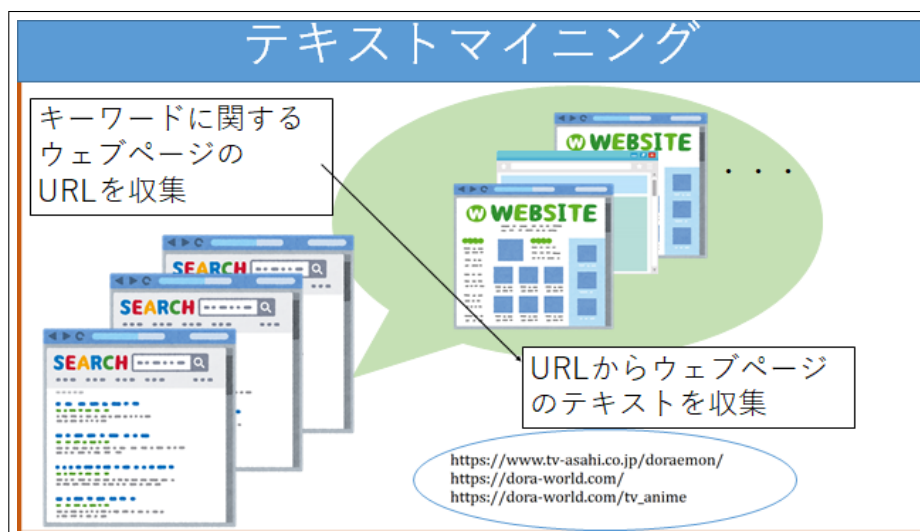


図 3.1: テキストマイニングのイメージ

|            | MeCab           | ChaSen       | JUMAN           | KAKASI     |
|------------|-----------------|--------------|-----------------|------------|
| 解析モデル      | bi-gram マルコフモデル | 可変長マルコフモデル   | bi-gram マルコフモデル | 最長一致       |
| コスト推定      | コーパスから学習        | コーパスから学習     | 人手              | コストという概念無し |
| 学習モデル      | CRF (識別モデル)     | HMM (生成モデル)  |                 |            |
| 辞書引きアルゴリズム | Double Array    | Double Array | パトリシア木          | Hash?      |
| 探索アルゴリズム   | Viterbi         | Viterbi      | Viterbi         | 決定的?       |
| 接続表の実装     | 2次元 Table       | オートマトン       | 2次元 Table?      | 接続表無し?     |
| 品詞の階層      | 無制限多階層品詞        | 無制限多階層品詞     | 2段階固定           | 品詞という概念無し? |
| 未知語処理      | 字種 (動作定義を変更可能)  | 字種 (変更不可能)   | 字種 (変更不可能)      |            |
| 制約つき解析     | 可能              | 2.4.0で可能     | 不可能             | 不可能        |
| N-best解    | 可能              | 不可能          | 不可能             | 不可能        |

図 3.2: 形態素解析エンジンの比較

発想支援において必要な名詞や動詞に分解することである。形態素解析エンジンとして MeCab, ChaSen, JUMAN, KARASI, Janome などがある。それぞれの形態素解析エンジンに特徴がある。今回は ChaSen や KAKASI に比べ高速で、解析精度に関わる新語の解析に使う辞書の設定が容易な MeCab を使って形態素解析した。

また、助詞の「は」や「が」助動詞の「です」は不要なので取り除くことが多い。そして単語群に対して正規化を行う。正規化することで文字種を統一できる。半角と全角や数字を統一することで同じ単語として扱うことができる。また、前処理として

## § 3.2 PageRank による隣接行列の絞り込み

$$r(P_i) = \sum_{P_j \in B_{P_i}} \frac{r(P_j)}{|P_j|} \quad (3.1)$$

$r(P_j)$ :page $P_j$ のページランク

$P$ :全ページの集合

$B_P$ :入リンクの集合

$|P_j|$ :ページ  $P_j$  の出リンク数の総数

自然言語処理の分野において、単語をベクトル表現する Word2Vec という技術がある。複数のキーワードの関連語がなかった場合にも発想支援できるように単語をベクトル表現することで近いベクトルの単語同士を関連語として見れるのではないかと考える。

ある単語とある単語が同時に出現することを共起するといい、文章において関係深い単語は共起することが多い。共起分析では単語同士の Jaccard 係数を比較したり、共起関係を持つ単語と単語を線で結んで描かれる共起ネットワークが利用される。文章また単語群に対して共起する単語をネットワークで表した共起ネットワークという。

### § 3.3 3D Force-Directed Graph

隣接行列から 3 により単語のつながりを可視化する。単語群から、単語の遷移をにカウントして隣接行列を作成する。隣接行列を Pandas で作成し、隣接行列から NetworkX という Python のライブラリを使用してベイジアンネットワークを作成する。

# 提案手法

### § 4.1 複数のキーワードから隣接行列を作り 3D グラフ作成

提案するモデルについて説明する．任意の2つのキーワードについて Google 検索から，クローリングしてそれぞれのキーワードのウェブサイトの URL を Google 検索結果のいくつかのページ分集めてくる．その URL のウェブサイトのテキストデータを収集して自然言語処理およびその前処理をして名詞・形容詞・動詞に絞った単語にまとめる．その単語の順序関係を用いて 3D グラフに適応できるようにして 3D グラフから発想支援に役立てる．

システムとしては Python で Selenium というライブラリと GoogleChrome 用の WebDriver である CrhomeDriver を使って複数のキーワードを Google で検索して URL を取得し，その URL をまとめたものをテキストデータとして保存する．例としてキーワードにトマトとプラスチックと入力した際の URL リストは図 4.1 に示す．

前処理として

### § 4.2 提案手法のアルゴリズム

最後に，本研究で提案したモデルのパラメータ設定から解法までのアルゴリズムについてまとめる（図??参照）．まず，ファジィ・ランダム多目的日程計画問題を解くために必要となるデータの収集を行い，そのデータを用いたパラメータの設定方法を以下に示す．

Step1 : クローリング

まず，入力された2つのキーワードに対して Google 検索から URL を検索結果の1ページ分収集して URL リストとしてテキストファイルにして保存する．収集したすべての URL に対して BeautifulSoup というライブラリを使ってそれらサイトの HTML を収集する．

トマト - Wikipedia  
 【みんなが作ってる】 トマトのレシピ 【クックパッド】 簡単おいしい ...  
 トマトのレシピ | キックコマン | ホームクッキング  
 農園kitchen moriy (モリー) トマトを食べよう、農園のとなりで  
 トマトとは - コトバンク  
 [トマト] おいしいトマトの見分け方と、保存のコツ | カゴメ株式会社  
 さっぱり美味しい 無限トマト 作り方・レシピ | クラシル  
 トマト とまと : 旬の野菜百科 - フーズリンク合成樹脂 - Wikipedia  
 プラスチックの基礎知識 | プラスチック | プラスチック情報館 ...  
 プラスチックとは - コトバンク  
 プラスチックっていったい何? ゼロから分かるプラスチックの基本と ...  
 プラスチックとは | 日精樹脂工業株式会社  
 プラスチックの定義 (基礎知識) | KDAのプラスチック加工技術  
 そもそもプラスチックとは何でしょうか? | プラなし生活  
 プラスチックっておもしろい | プラマテルズ株式会社  
 プラスチックとは何? Weblio辞書

図 4.1: キーワードに「トマト」と「プラスチック」と入力した際にクローリングして取得した URL タイトル

<https://ja.wikipedia.org/wiki/%E3%83%88%E3%83%9E%E3%83%88>  
<https://cookpad.com/search/%E3%83%88%E3%83%9E%E3%83%88>  
<https://www.kikkoman.co.jp/homecook/series/tomato02.html>  
<http://moriy.toyama.jp/>  
<https://kotobank.jp/word/%E3%83%88%E3%83%9E%E3%83%88-105797>  
<https://www.kagome.co.jp/vegeday/yasai/tomato/>  
<https://www.kurashiru.com/recipes/b37f24c1-8aff-4007-a631-2545d7885cce>  
<https://foodslink.jp/syokuzaihyakka/syun/vegetable/tomato.htm>  
<https://ja.wikipedia.org/wiki/%E5%90%88%E6%88%90%E6%A8%B9%E8%84%82>  
[https://www.nagaseplastics.co.jp/info/plastic/basic\\_knowledge](https://www.nagaseplastics.co.jp/info/plastic/basic_knowledge)  
<https://kotobank.jp/word/%E3%83%97%E3%83%A9%E3%82%B9%E3%83%81%E3%83%83%E3%82%AF-126128>  
<https://ecoracy.com/83.html>  
<https://www.nisseijushii.co.jp/plastics/plastics1.php>  
[https://www.kda1969.com/study/study\\_pla\\_basic.htm](https://www.kda1969.com/study/study_pla_basic.htm)  
[https://lessplasticlife.com/marineplastic/driver/what\\_is\\_plastic/](https://lessplasticlife.com/marineplastic/driver/what_is_plastic/)  
<https://www.plamatels.co.jp/special>  
<https://www.weblio.jp/content/%E3%83%97%E3%83%A9%E3%82%B9%E3%83%81%E3%83%83%E3%82%AF>

図 4.2: キーワードに「トマト」と「プラスチック」と入力した際にクローリングして取得した URL リスト

#### Step2: 自然言語処理の前処理

次に、自然言語処理における前処理を行う。収集した HTML に対してすべての HTML のタグを消去してテキストデータのみ抽出する。抽出したテキストデータに対して MeCab を使ってクリーニング処理と形態素解析を行い不要な記号や数字の除去し、名詞・動詞・形容詞を取り出す。

#### Step3: 名詞群からの隣接行列の作成

取り出した単語群の順序関係を隣接行列にする。このとき、すべての単語で隣接行列を作成すると行列の大きさが必要以上に大きくなるので Pagerank のアルゴリズムによってより重要な単語だけを取り出し隣接行列を作成する。

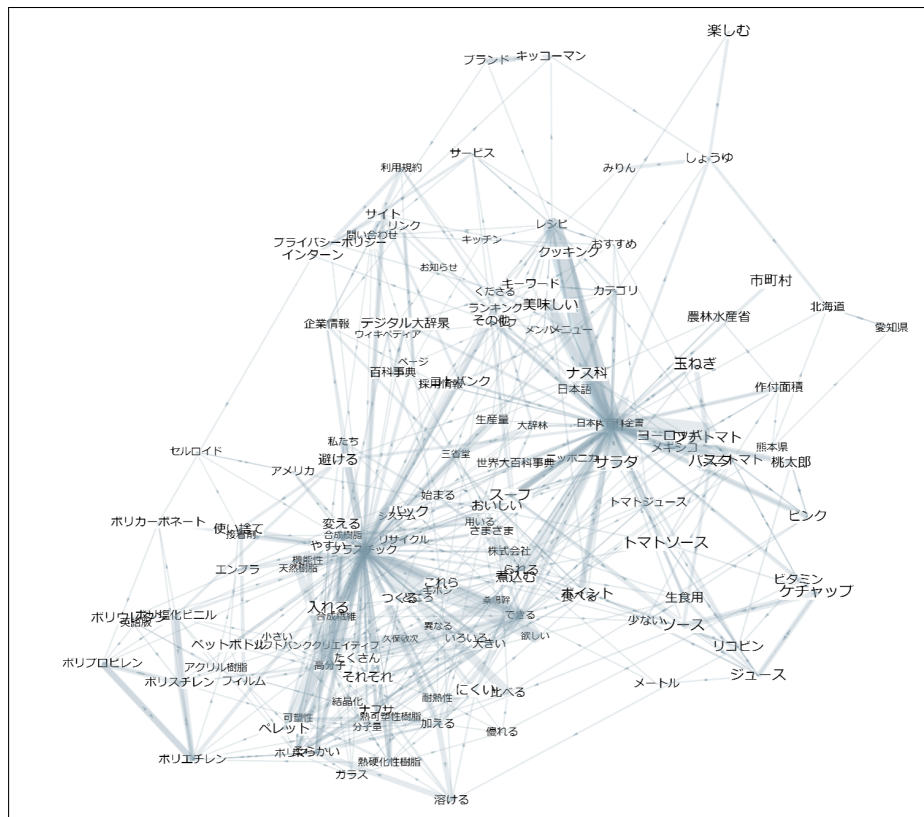


図 4.3: キーワードに「トマト」と「プラスチック」と入力して PageRank のしきい値を 0.001 にした場合の出力結果

Step4 : 隣接行列から 3D グラフへの適応

隣接行列を 3d-force-graph に適応する。

結果:

## § 4.3 結果と考察

実際にキーワードとして「トマト」と「プラスチック」を入力としたときに図のような出力結果となる。

出力結果からの考察として「トマト」→「食べる」→「やすい」→「プラスチック」というつながりから「トマトを食べやすくするプラスチック製品」というアイデアの種を発見することができる。システムの出力結果と実際に Google 検索の結果を比較した。Google の検索結果では

PageRank のしきい値を大きくすることで





### おわりに

本研究では，アイデアの発想支援を目的としてグーグル検索を用いたテキストマイニング  
本研究で提案したシステムの特徴をまとめる．一つ目の特徴は，複数のキーワードに対しての発想支援が行える点である．二つ目の特徴は，3D 有向グラフに適応した点である．  
本研究の研究成果は，  
今後の課題として，  
まだ様々な方面からアプローチし，改善する余地があると考えられる．

# 謝辞

本研究を遂行するにあたり，多大なご指導と終始懇切丁寧なご鞭撻を賜った富山県立大学電子・情報工学科の奥原浩之教授に深甚な謝意を表します．最後になりましたが，多大な協力をして頂いた奥原研究室の同輩諸氏に感謝致します．

2020 年 2 月

山元 悠貴

## 参考文献

- [1] 國藤 進, “発想支援システムの研究開発動向とその課題”, 人工知能学会誌, pp. 552-559, 1993.
- [2] イ スンジュ, “発想支援のためのテキストマイニング”, 人工知能学会 第 25 回 セッション ID: 1P2-10in, 2011.
- [3] Rada Mihalcea, Paul Tarau, Elizabeth Figa, “PageRank on Semantic Networks with Application to Word Sense Disambiguation”, In Proceedings of the 20st International Conference on Computational Linguistics, 2004.
- [4] 飯田耕司, “不確実性への挑戦 意思決定分析の理論”, 三恵社, 2006.
- [5] “PERT と CPM” , <http://d-engineer.com/industrialeng/pert.html>. 閲覧日 2018, 12, 20.
- [6] “ク リ ティカ ル パ ス で プ ロ ジェ ク ト 遅 延 要 因 を 以 前 に 把 握 セ ヨ” ,<https://pmkuma.com/critical-path-method/>. 閲覧日 2018, 12, 20.
- [7] 木瀬 洋, “スケジューリング研究の過去・現在・未来”, スケジューリングシンポジウム 2000 講演論文集, pp. 5-10, 2000.
- [8] 嘉納成男, 石田航星, “繰り返し工程に関する工程計画手法の開発”, 日本建築学会計画系論文集, Vol. 81, No. 723, pp. 1195-1205, 2016.
- [9] 吉川和広, 春名 攻, “ネットワーク手法による施工計画のシステムアプローチに関する研究-CPM 計算の簡便化-” 土木学会論文報告集, Vol. 182, pp. 41-58, 1970.
- [10] “建築現場を IT 化する「ダンドリ」ツール 6 選”, <https://mag.branu.jp/archives/2442>. 閲覧日 2018, 6, 5.
- [11] “現場コミュニケーションアプリ Kizuku”, <https://www.ctx.co.jp/kizuku2pr/>. 閲覧日 2018, 6, 5.
- [12] David L. Applegate, Robert E. Bixby, Vasek Chvatal, William J. Cook, The Traveling Salesman Problem: A Computational Study, Princeton Univ Pr, 2007.
- [13] M. K. Luhan (Ijula. and M. M , Gupta ; On fuzzy stochastic optimization. Fuzzy Sets and Systems, 81, pp. 47- 55, 1996.
- [14] M.Gen, R. Cheng, Genetic Algorithms, Engineering Design, John Wiley, Sons, 1997.
- [15] 矢野均, “多目的ファジィランダム単純リコース問題に対する対話型意思決定とその応用”, 日本経営工学会論文誌, Vol. 67, No. 4, 2017.
- [16] 矢野均, “ファイジィ・ランダム単純リコース問題の定式化”, 日本知能情報ファジィ学会誌, Vol.27, No. 4, pp. 695-699, 2015.

- [17] 蓮池隆, “CPM を用いた不確実・不確定状況下におけるクリティカルパスの求解”, 数理解析研究所講究録, Vol. 1829, pp. 72-79, 2013.
- [18] 片桐英樹, 坂和正敏, 加藤浩介, 大崎修嗣, “ファジーランダム多目的線形計画問題に対する可能性測度と必然性測度を用いた満足水準最適化モデルに基づく対話型ファジー満足化手法”, 電子情報通信学会論文誌 A Vol. J87-A, No. 5, pp. 634-641, 2004
- [19] 片桐英樹, 坂和正敏, 石井博昭, “ファジィランダム変数を含む線形計画問題に対する可能性計画と確率計画に基づく意思決定 (あいまいさと不確実性を含む状況の数理的意思決定)”, 数理解析研究所講究録, Vol.1252, pp. 240-245, 2002.
- [20] 玉置 久, 森 正勝, 荒木 光彦, “遺伝アルゴリズムを用いたパレート最適解集合の生成法”, 計測自動制御学会論文集, Vol. 31, pp. 1185-1192, 1995.
- [21] 堀井宏祐, 國藤進, 松澤照男, “非同期型島モデル並列 GA の評価”, 計測自動制御学会論文集, Vol.35, NO.11, 1394-1399, 1999.
- [22] 末次峻也, 片山尋貴, 徳丸正孝, “並列 GA による感性データ解析”, 日本知能情報ファジィ学会誌, Vol. 25, NO.6, pp. 924-934, 2013.
- [23] 山中亮典, 吉見真聡, 廣安知之, 三木光範, 横内久猛, “遺伝的アルゴリズムの並列計算システム向けフレームワークの提案”, 情報処理学会研究報告, Vol. 2010, No. 8, pp. 1-7, 2010.
- [24] “Raspberry Pi とは”, <https://furien.jp/columns/58/>. 閲覧日 2018, 11, 6.
- [25] “MPIの環境構築や基本コマンドのまとめ”, <https://qiita.com/kkk627/items/49c9c35301465f6780f>. 閲覧日 2018, 11, 6.
- [26] “MPI を用いた並列プログラミングの概要”, <http://www.cenav.org/raspi4b/>. 閲覧日 2018, 11, 8.
- [27] “MPI 並列プログラミング”, <http://www.matlab.nitech.ac.jp/kazu-k/mpi.html>. 閲覧日 2018, 11, 8.
- [28] Hideki Katagiri, “Interactive multiobjective fuzzy random linear programming: Maximization of possibility and probability”
- [29] M. Sakawa, I. Nishizaki, H. Katagiri, Fuzzy Stochastic Multiobjective Programming, Springer, 2011.
- [30] 椎名孝之, “確率計画法”, 朝倉書店, 2015.



# 付録

## A. 1 Hello World を並列実行するソースコード

Raspberry Pi 3 を 8 台で Hello World を並列実行するソースコード A.1 を示す。

ソースコード A. 1: hellow.c

```
1 #include <stdio.h>
2 #include "mpi.h"
3 int main( int argc, char *argv[] )
4 {
5     int rank;
6     int size;
7     double t0,t1,t2,t_w;
8     MPI_Status stat;
9     MPI_Init(&argc,&argv );
10    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
11    MPI_Comm_size(MPI_COMM_WORLD, &size);
12    MPI_Barrier(MPI_COMM_WORLD);
13
14    t1=MPI_Wtime();
15    printf( "Hello world from process %d of %d\n", rank, size );
16    MPI_Barrier(MPI_COMM_WORLD);
17    t2=MPI_Wtime();
18    t0=t2-t1;
19    MPI_Reduce(&t0, &t_w, 1, MPI_DOUBLE, MPI_MAX, 0, MPI_COMM_WORLD);
20    MPI_Finalize();
21    printf("%f\n",t_w);
22    return 0;
23 }
```

## A. 2 円周率計算を並列分散処理するソースコード

Raspberry Pi 3 を 8 台で円周率計算を並列分散処理するソースコード A.2 を示す。

ソースコード A. 2: pi.c

```
1 #include <stdio.h>
2 #include <math.h>
3 #include <mpi.h>
4 void main(int argc, char* argv[]){
5     double PI25DT = 3.141592653589793238462643;
6     double mypi, pi, h, sum, x, f, a;
7     double t1, t2, t0, t_w;
8     int n, myid, numprocs, i, rc;
9     int ierr;
10
11    MPI_Status stat;
```

```

12 MPI_Init(&argc, &argv);
13 MPI_Comm_rank(MPI_COMM_WORLD, &myid);
14 MPI_Comm_size(MPI_COMM_WORLD, &numprocs);
15 if(myid==0){
16     printf("Enter the number of intervals \n");
17     scanf("%d", &n);
18     printf("n=%d \n", n);
19 }
20 MPI_Barrier(MPI_COMM_WORLD);
21 t1=MPI_Wtime();
22
23 MPI_Bcast(&n, 1, MPI_INT, 0, MPI_COMM_WORLD);
24 h = 1.0 / n;
25 sum = 0.0;
26 for(i=myid+1; i<=n; i+=numprocs){
27     x = h * (i - 0.5);
28     sum = sum + 4.0 / (1.0 + x*x);
29 }
30 mypi = h * sum;
31 MPI_Reduce(&mypi, &pi, 1, MPI_DOUBLE, MPI_SUM, 0, MPI_COMM_WORLD);
32 if(myid == 0){
33     printf("pi is approximately: %18.16lf Error is: %18.16lf \n", pi,fabs(pi-
        PI25DT));
34 }
35 MPI_Barrier(MPI_COMM_WORLD);
36 t2 = MPI_Wtime();
37 t0= t2-t1;
38 MPI_Reduce(&t0, &t_w, 1, MPI_DOUBLE, MPI_MAX, 0, MPI_COMM_WORLD);
39
40 if(myid==0){
41     printf("execution time = : %8.4lf [sec.] \n", t_w);
42 }
43 rc= MPI_Finalize();
44
45 }

```

---