脳認知モデルに基づいた大規模情報処理システムに関する研究

明治大学大学院理工学研究科

2014年度

博士学位請求論文

脳認知モデルに基づいた
大規模情報処理システムに関する研究

The study on Large Scale Information Processing System

based on Cognitive Model of Human Brain

学位請求者　基礎理工学専攻

高　田　朋　貴

脳認知モデルに基づいた大規模情報処理システムに関する研究

高田　朋貴

内容梗概

近年，WEB の発達により人間が扱うことができないほど情報が増大しており，SNS やスマートフォンの出現によりその情報量はますます増えることが予想される．その中で，特に人間によって生成されたテキストデータを計算機で自動的に情報処理することは必要不可欠の課題である．例えば，ユーザの趣味嗜好の情報は情報推薦やターゲティング広告配信において重要な情報であり，そのような情報を得るためにはテキストデータの解析が欠かせない．また，情報検索では文書の内容を示す特徴的な語を文書中から抜き出すことにより，文書の内容を表現している．しかし，現状の計算機では人間の言葉を正確に理解することができない問題がある．人間は文書の内容を抽象化し，端的に表すことができるなど，さらに高度な情報処理を行うことができる．したがって，本研究では計算機によるテキストデータの意味理解へのアプローチとして意義ある研究であり，本研究では人間の脳モデルを用いて大規模な量の文書の内容を適切な語彙群で置き換えることで，より人間に近いテキスト情報処理システムの実現を目指している．

具体的には，人間の脳の認知に基づいた Confabulation theory を基盤として，文書にキーワードを付与するシステムを実現した．Confabulation theory は，人間の脳の柔軟かつ汎用性が高い構造をモデル化していると同時に，モデルがシンプルである故，汎用性が非常に高く，様々な分野への適用が期待できる．

本研究では，Confabulation theory を自然言語処理に応用し，現在人手による作業が必要なキーワード付与作業を完全に自動化し，大規模なデータに対して高精度かつ高速なキーワード付与システムを実現した．実験では定量的評価，定性的評価，実行速度評価を行い，定量評価においては比較手法よりも約 10%の精度向上，定性評価でも比較手法よりも優れ，実行速度も約 19 倍高速に処理することが可能であることを示した．

目次

第１章　序論

1.1 はじめに

近年 WEB の発達により，ビッグデータと呼ばれるように大量のデータが増加し続けている．特に，SNS と呼ばれるようなソーシャルサイトの出現により，ユーザにとって WEB の「情報」とは，今までマスメディアによる一方的で受動的なものから，ユーザ自身による双方向で能動的なものとなりつつあり，ますます WEB 上のデータが増加する起因となった．例えば，twitter，Facebook など多種多様なソーシャルサイトを用いることで，ユーザは趣味の話から仕事やビジネスの話まで，容易に情報を作成することや，情報発信が可能となり，短時間でかつ多くの人にその情報を届けることが可能になった．図 1.1 は近年の SNS の利用者の動向を示しており，今後も SNS の利用者数はますます増加することが見込まれている．

図 1.1　日本における SNS 利用者数（ICT 総研調べ）

更にスマートフォンの出現が非常に大きい．日本における携帯電話の普及率は非常に高く，図 1.2 で示すように，スマートフォンも年々普及率が増加しつつある．スマートフォンはいわば小型計算機であり，携帯電話に比べ，WEB へのアクセスがパソコンとほぼ同様にかつ高速に行うことができる．その為，ユーザはいつでもどこでも，ほぼリアルタイムに近い状態で情報の発信から取得までが可能となる．したがって，WEB を利用するユーザはますます増加し，ソーシャルサイトなどを通じて更にデータが増加していくことは容易に想定することができ，WEB の世界が今後ますます発展し続けることはほぼ疑いようがないだろう．

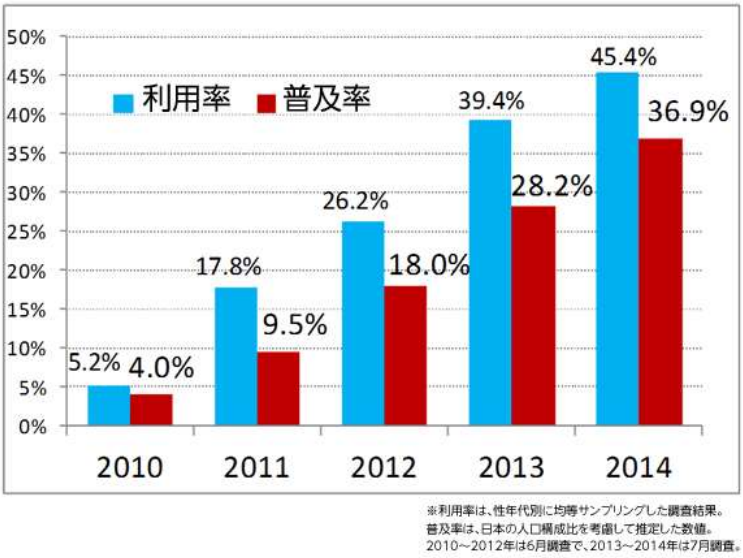

図 1.2　スマートフォンの利用率と普及率の推移（日経 BP コンサルティング調べ）

このデータの増加に伴い，近年ではますます自然言語処理の技術が注目されつつある．自然言語処理（Natural Language Processing）とは，人間の扱う自然言語を計算機上で処理させるための技術であり，その最終的な目的は計算機に人間のことばを理解させることである．自然言語は，プログラム言語やデータベースの数値のように明確な意味構造を持たないことから，計算機で扱うことは非常に困難なタスクである為，古くから研究がなされている．では，なぜこの自然言語処理技術が近年，ますます注目を浴びているのであろうか？

一つ目の要因としては，スマートフォンのデバイス的特徴の為であると考えられる．ユーザはスマートフォンの小さな画面上で文字入力等の種々の操作を行わなければならない．前述したとおり，スマートフォンは小さな計算機といって良いほど高機能となり，パソコンにだいぶ近い情報処理を行えるようにはなったが，パソコンと比べ，マウスやキーボードがない為に，入力のインターフェースとしては非常に不便である．その為，スマートフォンの不自由な入力のインターフェースを補うために，文章を入力する際に文字を予測する技術や，Siri やしゃべってコンシェルなどの音声入力の技術が普及することになった．これらの技術には自然言語処理の技術が欠かせない．

また，スマートフォンにて情報検索を行う必要がある場合にもその入力インターフェースの不自由さが際立ってしまう．その為，近年では Gunocy をはじめとしたキュレーションサイトが情報収集ツールとして注目されている．例えば Gunocy は，twitter や facebook といった SNS からユーザの趣味・嗜好を分析し，その興味はあったニュースや記事を推薦するサービスである．これらのソーシャルサイトにある情報は基本的にはテキスト情報である為，ユーザの趣味や嗜好を分析するためには，やはり自然言語処理の技術が必要不可欠となる．

二つ目の要因としては，ソーシャルサイト等のサイトを通じて，ユーザの行動や趣味嗜好を解析し，その結果を積極的にビジネスに活用しようという流れが近年活発な点である．

具体的には，SNS 解析や，EC，広告などのサービスが該当する．

SNS 解析の背景では，SNS をメディアとしてみた時に，その影響力は既に無視できないほど大きいものとなっていることがある．その為，SNS 上での情報解析や，ある出来事の影響度の解析のニーズが大きくなっている．

広告業界においては，従来型の広告では，テレビをはじめとしたマスメディアが最も主流であった．しかし，近年では図 1.3 で示すように，ターゲティング広告やリスティング広告，アフィリエイト広告など，WEB 上での広告であるインターネット広告が伸び始めている．マスメディアは多くの層に対する，言ってしまえば大雑把な広告であるのに対し，インターネット上での広告は個々人をターゲットとした肌理細かな広告配信が可能となる．具体的には，SNS などを通じたユーザの属性分析は，ある商品はどんな人にうけて，どんな人が外れているかのマーケティング分析にも利用される．その結果として，各ユーザに適した広告を配信することができる．また，インターネット広告は広告費の単価がテレビなどに比べ安いことや，不完全ながらもテレビよりも広告を出したことに対する効果測定が定量的に行うことができるため，近年注目を浴びている．

図 1.3 2013，2014 年度の 7-9 月における前年同月比（経済産業省調べ）

オンラインショッピングのような”e コマース”（EC）においては，百貨店などでの店頭販売の代わりに台頭しつつある．例えば，Amazon や楽天などのネット通販サービスが以前よりもユーザに受け入れられ，手軽に利用されるようになり，EC 市場における市場は図 1.4 で示すように年々拡大しつつある．EC サイトでは，ユーザが検索したものに商品に当てはまるようなものを提示する情報検索技術や，テキストからの商品の属性抽出，商品情報からユーザの趣味・嗜好に適合した商品をお薦めする推薦エンジンなど，多くの自然言語処理技術が活用されている．

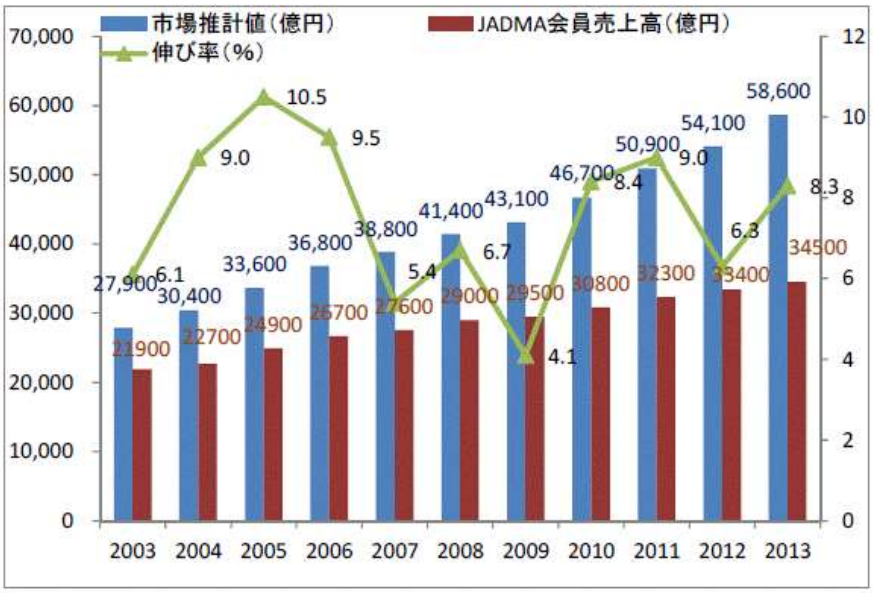

図 1.4　2013 年度通信販売売上高について（公益社団法人　日本通信販売協会調べ）

しかし，現状の計算機は人間の言葉を正確に理解することができない問題がある．その一つの例として「語義曖昧性」の問題が挙げられる．語義曖昧性とは，文中に出現した単語の語義が曖昧であることであり，自然言語処理の分野ではこれを解消するための研究がある．例えば，Java という単語はプログラミング言語のことかもしれないし，コーヒーのことかもしれないし，島の名前のことかもしれない．人間は文書から文脈情報を読み取ることで，注目している言葉が何を意味しているのかを理解することができる．しかし，計算機は真に言葉や文章の意味を理解できていない為，人間が容易に理解できるような事も計算機では非常に難しい問題となってしまっている．

もう一つの問題としては「不自然言語処理」が挙げられる．従来の自然言語処理では新聞などのいわゆる「きれいな日本語」を対象として研究がなされてきた．しかし，ブログや SNS 上の言葉はよりカジュアルな日本語が用いられているため，従来の自然言語処理の枠に収まらず，正確な言語解析ができないことが問題となっている．

1.2 本研究の目標

以上で述べたように，いまテキストデータを計算機で自動的に情報処理することは必要不可欠の課題である．更に，人間のような高度な知的情報処理を計算機が行う必要性もあると考える．

本研究では，計算機によるテキストデータの意味理解へのアプローチの一つとして，より人間的なテキスト情報処理を行うシステムの実現を目指し，人間によって生成された文書の意味を適切な語彙群で言い表すシステムの実現を目標とする．

具体的には，より人間的なテキスト情報処理を行う為に，人間の脳の認知モデルを用い，文書に適切なキーワードを付与するシステムの実現を目指す．

1.3 本論文の構成

本論文は，全 6 章で構成されている．

まず，2 章では既存研究における計算機上での文書の処理方法について概観する．この章では，「文書を単語で表現する」という観点を主軸とし，関連研究でのその役割を整理することで，文書の内容を適切なキーワードで表す”keyword annotation”の重要性について述べている．

続いて 3 章では脳モデルに関する関連研究について言及する．近年提案されている脳モデルを紹介し，テキスト処理におけるそれらの脳モデルの適用の問題点について述べている．

4 章では本研究の基盤となる考えである人間の脳の認知をモデル化した Confabulation theory について述べ，4 つの大きな特徴について言及し，モデルの優れた点を説明する．

5 章では本研究で提案する自動的に文書にキーワードを付与する Automatic Keyword Annotation System について説明する．そして，新聞を用いたキーワード付与実験によるシステムの評価を行い，提案システムの有効性を示す．

そして最後に 6 章では本研究の結論と，今後の研究課題について述べる．

第 2 章　Keyword annotation

2.1 はじめに

この章では，本研究の目標である文書に適切なキーワードを付与する”Keyword annotation”というタスクの重要性を，関連研究を概観することで言及する．Keyword annotation とは，文書の内容を適切な語彙群で言い表すタスクである．1 章で述べたように，近年の WEB の発達により，大量の情報が増大しており，既に人手では処理しきれない膨大な量となっている．その為，ユーザは自分に適切な情報取得が非常に困難であり，計算機によるサポートが必要不可欠である．

本研究では，特に計算機によるテキスト情報処理に焦点を当てている．計算機が文書データを取り扱い，応用的な情報利活用をする為には，計算機が文書の意味を正確に認識できる必要性がある．文書を対象にした関連研究には，情報検索や情報抽出，文書分類，文書要約，情報フィルタリングなどが挙げられる．例えば，情報検索の分野では，ユーザによって入力されたクエリの意味を捉え，その意味と適合するような文書を検索する為には，検索対象となる文書の内容を計算機が認識できなければならない．また，文書分類の分野においては，文書に付与すべきカテゴリを考え，そのカテゴリに基づき分類を行うが，カテゴリの決定には文書の内容が何を表しているのかがわかる必要性がある．

以上のように，Keyword annotation は多くの関連研究に共通した課題である．したがって本章では，関連研究を概観し，現状の計算機による文書の取り扱いの考え方について整理する．また，一般的な評価方法についても言及する．

2.2 情報検索

2.2.1 一般的な情報検索

情報検索とは，広義な意味では「ユーザの持つ問題（情報欲求）を解決できる情報を見つけ出す」ことであり，狭義な意味では「ユーザの検索質問（query）に適合する文書（document）を文書集合（document collection）の中から見つけ出す」ことを意味する．ここでいう検索質問とは，ユーザの情報欲求を具現化したものといえる．

現在大量の情報がインターネット上で増加し続けており，ユーザにとって有益な情報を発見することはますます困難になりつつある．その為，Google や Yahoo!のような情報検索システムは日に日に重要になりつつある．

情報検索では，現実世界の情報が文書で表現されていることを仮定し，文書をコンピュータで扱

えるような内部表現に変換する．また，ユーザの情報欲求も検索質問という形式で表現し，これもコンピュータで扱えるような内部表現に変換する．そしてこれらの内部表現を比較することで，ユーザの情報欲求に適した情報を見つけ出す．

したがって，情報検索においては文書の構造や内容をどのようにして，またはどうやってコンピュータで扱えるような内部表現に変換するかが非常に大切である．このことは，コンピュータによって人的に生成された文書データをどのようにして理解させるかを考えていることとほぼ同値であるとも考えられる．

コンピュータが扱えるような内部表現を得るためには，文書の内容情報の言語表現からその意味を抽出する必要がある．このような処理は自然言語処理（natural language processing）と呼ばれるが，現在の技術では言語表現の意味を計算機に完全に理解させることは難しい．近年では，この言語表現の意味を計算機上で表現する研究[1]も進め始められている．現在情報検索において一般的に行われているのは，文書からその内容を表していると考えられる単語を抽出し，その単語集合によって文書の内容を表現する方法である．このような文書の内容を表す特徴的な語のことを索引語（index term）と呼ぶ．つまり，文書の中から特徴的な語である索引語を抽出し，この抽出した索引語の集合を用いることで，文書の内容を近似する手法が一般にとられている．この索引語は情報検索において最も基本的なものである．なぜならば，もし索引語の精度が非常に乏しければ，情報検索システムの精度も悪くなってしまうからである．

文書から索引語を抽出する処理のことを索引付け（indexing）と呼ぶ．索引付けを行う上で重要なことは，文書中からその文書を特徴付ける索引語を漏れなく抽出することである．文書の特定性を高くする為には，その文書に現れるが，他の文書には現れないような索引語を選択すれば良い．しかし，あまりに文書に特化しすぎた索引語を選んでしまうと，検索質問でその索引語が用いられる可能性も低くなってしまい，その文書が検索されないという問題も起きてしまう．

反対に一般によく用いられる語を索引語として選択すれば，多くの文書を検索することが可能となるが，検索されたすべての文書が必ずしもユーザが必要とする文書であるとは限らない．

索引付けには，人手で行うか，コンピュータが自動的に行うかの二種類の選択肢がある．人手による索引付けは，文書の内容を人間が読んで理解した上でおこなうので，精密さという点では優れている．しかし，実際にはどのような索引語を選択するかの基準は，索引語を付与する人間によって異なり，索引付けの一貫性を保つのが困難という問題がある．Cleverdon の実験によれば，同じ文書に複数の人間が索引付けを行った場合，その一致率は 30%程度しかなかったという[2]．また，実際にその索引語を手がかりに検索を行うユーザと，索引付けを行った人間の間での一貫性も保証されているわけではない問題もある．

一方，コンピュータによる自動索引付け（automatic indexing）は，同じ文書を同じ索引付けプログラムに何度入力しても同じ結果が得られるという意味では索引付けに一貫性があるといえる．ただし，索引付けプログラムは人間のように文書の内容を理解している訳ではない為，意味のない索引付けを行ってしまう可能性もある．

2.2.2 索引語の重み付け

索引語の中には，文書の内容と密接な関係にあるものや，文書の内容と関係が薄いものも存在する．つまり，抽出された索引語が文書の内容を表す上でどれだけ重要かどうかを定量的に計測することができれば，より精度の高い検索を実現できる．このように，索引語の重要度を付与することを索引語の重み付け（term weighting）という．以下では代表的な TFIDF による索引語の重み付け手法について言及する．

TFIDF

有名な索引語の重み付けとしては TFIDF（term frequency × inverse document frequency）と呼ばれる尺度による重み付けがある．この手法における重み付けでは，「現在注目している文書に多く出現する単語であり，かつ他の文書にはあまり出現しない単語」が文書にとって特徴的な単語である，という考え方に基づいた重み付けであるといえる．

まず，注目する文書に閉じた局所的な重みである TF について説明する．ある文書 d 中に出現する索引語 t の頻度を索引語頻度（term frequency）と呼び，$tf(t,d)$で表す．この$tf(t,d)$を文書 d における索引語 t の重みw_t^dと考えることができる．

$$w_t^d = tf(t,d) \tag{2.1}$$

索引語頻度に基づく重み付けは，「何度も繰り返し言及される概念は重要な概念である」という仮定に基づいている．しかし，一般にあまりにも頻度が高すぎるような語は他の文書にも出現する可能性が高いため，文書を特徴付ける上では役に立たない．そこで後述するような大域的重みも重要となってくる．また，文書が長くなってくると平均的に語の出現頻度も高くなってしまう傾向にある．つまり，単純にその文書における索引語の出現頻度をその索引語の重みとして採用してしまうと，同じ索引語でも長い文書に現れる索引語の方がより重みが大きくなってしまう．結果として，文書長が大きい文書ばかりが検索されてしまう問題が起きてしまう．そこで，索引語の出現頻度を文書中の全ての索引語の出現頻度の総和によって割ることで，文書長による影響を抑えることができ，相対的な頻度の重みとして用いることが可能となる．ここで，s は文書 d に出現した全ての索引語を表す．

$$w_t^d = \frac{tf(t,d)}{\sum_{s \in d} tf(s,d)} \tag{2.2}$$

上述した局所的な重みでは，文書内での頻度を考慮することはできるが，検索対象の文書集合内の他の文書の索引語の分布については考慮できない．つまり，どの文書においても高頻度に出

現する単語は文書を特徴付ける単語とは言い難く，索引語頻度に基づく重み付けだけでは不十分であるといえる．そこで，検索対象の文書集合全体での観点から大域的な重みを付与することも考えられる．このような大域的な重みとして以下の式で示すような IDF（inverse document frequency）がある[3]．IDF はある索引語が全文書中のどれくらいの文書に出現するかを評価する尺度である．なお，N は文書集合中の全文書数，$df(t)$は索引語 t が出現する文書数を示す．

$$idf(t) = \log \frac{N}{df(t)} + 1 \tag{2.3}$$

式からわかるとおり，IDF はどの文書にも出現するような文書の内容の特定性の低い索引語に対しては小さな重み付けを行い，少数の文書でしか出現しない文書の特定性が高い索引語に対して大きな重み付けを行うことができる．

この局所的重みと大域的重みの二つの尺度を用いることで，より文書の内容を特徴付ける重み付けである TFIDF が現在では一般的に用いられている手法である．TFIDF は以下の式のように求めることができる．

$$w_t^d = tf(t,d) \times idf(t) \tag{2.4}$$

2.2.3 新聞記事による記事検索

新聞記事も検索される対象となる文書のひとつである．新聞記事を検索文書対象と見なした場合，ユーザが読みたい新聞記事を素早く検索する必要がある為，情報検索技術は非常に重要である．なぜならば，新聞はほぼ毎日出版されており，その結果，日々非常に大量の文書が生成されることになるからである．

新聞の一般的なデータベース検索方式には，「全文検索」と「キーワード検索」がある．全文検索は前述したように文書内の語を索引語として文書検索を行う方式である．一方，キーワード検索は人が記事中から任意にキーワードを選んだり，内容を考慮した上で記事内には存在しない単語もキーワードとして付与したりする．浜口[4]によれば，特に新聞ではこの検索方式により文書のヒット率が大きく変わってしまうという．例えば，「交通事故」をキーワードとして検索した場合，全文検索方式とキーワード検索方式とでは，後者のヒット率が 2 倍ほど大きかった．これは，実際の記事には「交通事故」という単語が記事中に含まれない事が多く，記事中に含まれていなければ検索できない為である．その為．新聞記事の検索において，キーワード検索による検索は非常に重要となる．

また，日本経済新聞では，日経シソーラス呼ばれる辞書[5]を参照し，その内容と一致する単語を記事の文書中から抽出されている．この処理の後，編集者によって付与されたキーワードは吟味され，場合によっては，索引語として他の新しいキーワードが付加される場合もある．このように新聞に対してキーワードを付与する際に，各新聞社独自のシソーラスなどを用いて文中の単語を拡

張し，キーワード付与をする取り組みも多く行われているが，計算機が言葉の意味を理解してキーワードを付与していないため，最終的には人間が付与されたキーワードを確認する必要がある．例えば，「アップル」という単語に対してキーワードを拡張しようとした時，文書の内容を考慮し，文脈を理解できなければ，会社名のことなのか，それとも果物のことを指しているのかを計算機は判断することができない為，正しくキーワードを拡張することができない．

したがって，現状では人手なしに有益なキーワードを自動的に付与することができないという問題が新聞のキーワード付与にはある．

2.3 情報抽出

情報抽出（Information extraction）とはあらかじめ指定された情報を文書中から抽出することを目的としている．情報抽出においては特に，どのような情報を抽出するかをより詳細に指定する必要性がある．例えば，ある文書から誰が，いつ，どこで，何を，どれを，どのように，といったような情報を抽出するのは典型的な情報抽出である．このように情報抽出はある事実，あるいは事実と事実の関係を同定することになるので，事実検索（fact retrieval）と呼ばれることもある．

情報抽出の課題の例として，MUC（Message Understanding Conference）という国際会議の課題がある．MUC では，文書から抽出すべき情報が，テンプレート（template）と呼ばれる一種のフレーム構造で定義されている．あらかじめ与えられたテンプレートを埋めることのできる文書を検索し．その文書から情報を抽出してテンプレート中のスロットを埋めることがこの課題の目的となる．テンプレートを埋めるための情報を探すという前半の作業は情報検索と同じであり，情報抽出は情報検索の結果をより簡潔にまとめてユーザに提示する技術であるということもできる．

情報検索における言語処理の役割を比べると，情報抽出における言語処理の役割はより本質的である．情報検索では，自然言語処理技術を利用して索引付けをより精密にすることで，文書の内容をよりよく表現し，情報検索の性能を改善する試みが行われてきた．例えば，単語による索引付けだけでなく，句構造を用いたり，同義な語義に置き換えたりなどの工夫がされている．

一方で，情報抽出では言語処理を導入し，語と語，あるいは概念と概念の間の関係を同定できないと解くことができない課題である．この意味では，情報抽出は情報検索よりも言語処理の比重が大きい．Cowie と Lehnert は情報抽出が言語処理の応用として適している理由を以下のように述べている[6].

- 情報抽出は問題の定義が明確である
- 人間の性能を比較して評価することが容易である
- 実世界のテキストが処理の対象である

既に述べたように，情報抽出では抽出すべき情報が詳細に指定されるので，システムの出力が正しいか否かの判定が情報検索と比較すると容易である．情報検索では，検索として正しい文書

かどうかの判断はユーザによって異なってしまうことが考えられる為，適合性の判断が一般に難しい．しかし，情報抽出では実際に抽出された情報が正しいかどうかは抽出の対象となった文書を見れば判定できることが多い．また，同じ課題を人間が行う事ができるので，人間の性能とシステムの性能の比較が容易である．情報検索では，検索質問に対する適合文書集合を人手で作るのは非常に大変な作業であることから，システムと人間との性能比較は困難である．

2.3.1 固有名抽出

情報抽出の中の要素技術として固有名抽出（Named entity recognition; NER）というタスクがある．このタスクの目的は，人名，組織名，地名，時間表現などの固有名を文書の中から抽出することを目的としている．固有名抽出に関連する研究については関根の論文[7]がよくまとまっている．抽出には知識ベースの手法と統計ベースの手法がある．

知識ベースの手法では，人手によって固有名に関連する抽出規則を作り，その規則に基づいて固有名を抽出する．例えば，人名には「さん」や「氏」という語が付随し，地名では「市」や「町」などが後に続く．また，英語の場合，文頭以外にて大文字で始まる単語は固有名詞とみなすことができる．したがって，このような手がかり語をもとに固有名を抽出するのが知識ベース手法である．しかし，このような規則を全て人手で作るのは容易ではない．

一方，統計ベースではタグ付の訓練データを機械学習に与え，データから抽出規則を自動的に学習することで固有名を抽出する手法である．しかし，タグ付の訓練データを用意することは困難であり，また，ドメインや言語依存になってしまう問題もあることから，タグなしコーパスとタグありコーパスを組み合わせた半教師あり学習による手法も提案されている[8]．

また，いわゆる固有名ではなく，”generalized names”と呼ばれる病名やウイルスの名前を自動抽出する研究もある[9]．これらの単語は固有名抽出に比べ，手がかり語となるようなものが少ないことから NER よりも難易度の高いタスクである．

2.3.2 キーワード抽出

キーワード抽出（単語認識（term recognition）とも呼ばれている）は本章で紹介している Keyword annotation と情報抽出の中で最も関連があるタスクといえる．NER のタスクでは，地名や人名などあらかじめドメインが限られたタスクであったが，キーワード抽出では，文書内容の特徴をあらわすようなキーワードを抽出することが目的である．この分野については影浦の論文[10]が詳しい．例えば，論文のテクニカルなキーワードを抽出する研究[11][12][13]がある．これらの研究で扱われる文書データも NER 同様，教師データな付与されていないことから，教師なしあるいは半教師あり学習の手法が用いられる．

2.4 自動要約

情報検索の分野では，文書の内容を索引語の集合という形式で表現するのが一般的である．一般に索引付けの手法は，文書の内容をよく表している特徴的な語を抽出することを目的としている．この意味では，索引付けは文書からコンピュータのための要約を生成する処理だと考えることもできる．

しかしながら，文書の内容を索引語の集合で表現する形式はコンピュータにとっては扱いやすいが，人間にとっては必ずしもそうではない．索引語の集合からその文書の内容を予想することは我々にとって困難である場合が多い．索引付けがコンピュータのために文書を要約することを目的としているのに対し，テキストの自動要約では人間のために文書を要約することを目的としている．したがって，自動要約ではシステムは文書が入力となり，要約文を出力となる．

テキストの要約とは，その文書内で記述されている中心的な話題を簡潔にまとめたものであると定義できる．Paice は要約が持つ機能の観点から，要約を以下のようにまとめている[14]．

(1) 判断材料としての要約（indicative）
(2) 内容情報を提供する要約（informative）
(3) 評価を含む要約（critical）
(4) 比較を含む要約（comparative）

（1）の機能は，読者にその文書を読むかどうかを判断させる情報を与える機能を指す．読者は内容を完全に把握できなくても，その文書が自分の現在の関心に関係があるかどうかを判断できればよい．したがって，索引付けの結果として得られる索引語集合を示すだけでもある程度この機能をはたすことができると考えられる．（2）の機能は文書の内容を読者に伝える機能である．（3）の機能は文書の内容に加え，その内容の評価に関する情報も読者に伝える．（4）の機能では，その文書だけでなく，その文書と関連する文書の内容までも含め，その話題に関する複数のテキストの内容をまとめた情報を読者に提示する．（4）の機能を持つ要約はその文書の話題に関する一種の概説とも言える．

通常，（1）や（2）は文書の著者によって作成されるが，（3）や（4）の要約は第 3 者によって作成されるのが一般的である．また，これらの機能は包含関係にあり，（1）の機能は（2）の機能によって実現する事が可能であることから，（4）の機能が実現できれば，全ての機能を実現する事ができるという訳である．一般に，「要約」という場合は（2）の機能を持つ事が期待されている．

要約を作成するためには，テキストの内容を理解し，中心的な話題を特定した上で，それらを簡潔にまとめるという 3 つの作業が必要となる．したがって，自然言語処理が必然的に必要となるが，現在の自然言語処理技術では，テキストの内容を完全に理解したり，高品質なテキストを生成したりすることは難しい．したがって，現在の自動要約と称されている研究は，要約ではなく抄録（extract）を作成することが目的となっている．要約は，内容の理解とテキストの再生産が必要となる

が，抄録では重要な情報を含む文章を抽出できればよい．

この抄録の作成には，文を索引語のような単位と見なし，テキストの索引付けを行うことと見なすこともできる．一般的な抄録の手順は，まず語の頻度に基づき索引語のように語の重要度を計算し，文を含む語の重要度に基づき文の計算を行うという 2 段階の処理を行う．これは，文中に閉じた状態で語の頻度を計算しても，大抵すべての語の頻度が 1 となってしまうためである．これは，Luhn が主張している「何度も繰り返し言及される概念は重要である」という考え方に基づいており[15]，また語の頻度の索引付けにも同じ事が言える考え方である．

2.5 文書分類

文書の自動分類（text categorization）とは，文書をあらかじめ決めたカテゴリに分類，もしくは文書にカテゴリを付与することを指す．例えば，ある新聞記事に対して，その内容に沿ったカテゴリを付与することであり，「政治」や「経済」などがそのカテゴリにあたる．このようなカテゴリを検索質問とみなせば，文書の自動分類と情報検索は基本的には同じであると見なすことができる．ただし，文書分類では情報検索と異なり，検索質問に相当するカテゴリ集合は固定であり，入力となるテキストは開集合である．

文書の自動分類でも情報検索の基礎技術を利用することができる．自動分類の基本的な手続きは以下のようになる．

- 各カテゴリをあらかじめ内部表現に変換する
- 入力テキストを内部表現に変換する
- テキストと各カテゴリの間の類似度を計算する
- テキストにもっとも類似したカテゴリを付与する

文書やカテゴリの表現形式や類似度の計算方式は情報検索と同様に用いるモデルによって異なる．代表的なモデルは以下のモデルである．

- ベクトル空間モデル
- 確率モデル
- 規則に基づくモデル

ベクトル空間モデルとは，文書とカテゴリを索引語の重みベクトルで表現し，その間の類似度を計算することで最も類似度が高かったカテゴリを文書に付与する．カテゴリのベクトルは，あらかじめカテゴリが付与された文書集合を訓練データとして用いて計算する．

確率モデルでは，カテゴリが付与された文書集合をあらかじめ用意し，これを訓練データとして用いることで，確率モデルのパラメータ推定を行う．

規則に基づくモデルでは，文書を各カテゴリに分類するための条件を記述した分類規則を用意し，それを用いることにより，文書をカテゴライズする．

2.6 情報フィルタリング

情報フィルタリング（Information filtering）とは，ユーザの趣味や関心を記述したプロファイル（profile）を参照して，情報源から次々と流れてくる情報のうち，ユーザの関心がある情報だけを抽出する技術である．Belkin と Croft によれば，情報フィルタリングは以下のような特徴を持つという[16].

- 扱う情報があまり構造化されていない
- 扱う情報はテキストが主であるが，音声や画像などの情報を含む場合もある
- 扱う情報の量が大規模である
- 入力が情報のストリームである
- ユーザの長期的な嗜好を表現するプロファイルと呼ばれる情報を使う
- フィルタリングは必要なものを探すのではなく，不要なものを削除する

これらの特徴は，情報検索や情報抽出，要約や分類などと共通するものが多い．例えば，ユーザのプロファイルを一種のカテゴリであると見なせば，情報フィルタリングは文書分類と同値と考えることができる．ただし，文書分類ではカテゴリは静的であるのに対し，情報フィルタリングではユーザごとにプロファイルは異なり，更に時間とともにそのプロファイルも変化していくことが考えられる．

また，情報フィルタリングを必要な情報のみを抽出するという観点から見れば，情報抽出と類似しているといえるが，情報抽出では文書の内容を解析し，個別の事実情報を扱うことを重視している点で異なっている．

情報検索とも，情報フィルタリングはユーザの情報要求を満たす情報をユーザに提供するという目的においては同じであると考えられる．ただし，情報検索が短期的な情報の取得が主な目的となっている一方で，情報フィルタリングでは長期的な情報の取得が主な目的になっている等，幾つかの相違点もある．

しかし，情報フィルタリングでは情報検索の基本的な技術を用いることができる．例えば，システムに入力される文書やユーザのプロファイルを索引語の重みベクトルで表現すれば，Cosine 尺度など用いることでベクトル間の類似度を求めることができ，ユーザの関心のありそうな文書かどうかをこの類似度を基準に取捨選択することができる．

2.7 評価尺度

ここでは主に情報検索の視点から評価尺度について述べるが，以下で示すような評価尺度は情報検索だけでなく，情報抽出や情報フィルタリングでも用いられ，近年では推薦エンジンの評価尺度としても使われるような汎用的なものである．

2.7.1 再現率と精度

ある文書集合と検索質問集合について，各検索質問に対する各文書の適合性が与えられたと仮定する．具体的にはテスト・コレクションのような，システムを評価するための人工文書データを用意することで満たされる．このとき，各検索質問に対しての検索結果の文書集合と適合文書の関係は以下の図2.1のように表すことができる．

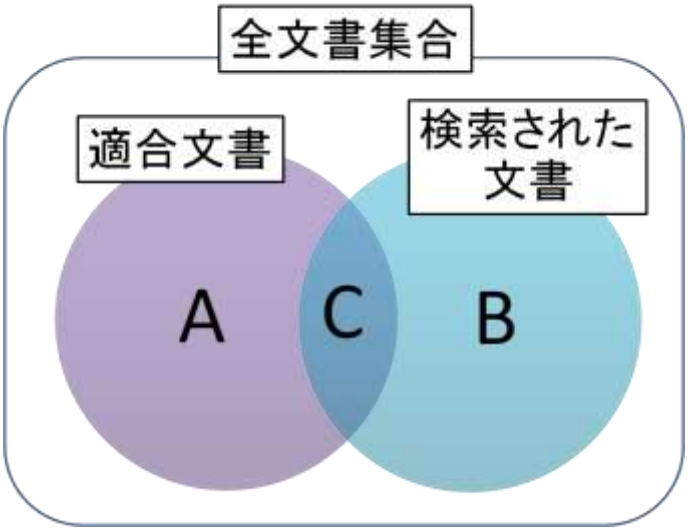

図 2.1 検索できた文書と実際に適合した文書との関係

以上の図から，適合率（Precision）と再現率（Recall）という二つの評価尺度を考えることができる．いま，集合 A を適合する文書集合，集合 B を実際にシステムによって検索された文書集合，集合 C を検索した文書中で適合した集合，つまり集合 A と集合 B の AND 集合とした時，適合率と再現率は以下のように定義できる．

$$Precision = \frac{|C|}{|B|} \tag{2.5}$$

$$Recall = \frac{|C|}{|A|} \tag{2.6}$$

適合率とは，正確性を評価するための尺度であり，検索された文章集合の中で，検索質問に適合する文書の割合を示す，検索ノイズの少なさを示す尺度である．また，再現率とは，完全性を評価するための尺度であり，検索対象となる文書集合の中の検索質問に適合する文書のうち，実際に検索された文書の割合を示す，検索漏れの少なさを示す尺度である．適合率，再現率の双方とも，その値は 0 から 1 の範囲にある．

例としてある検索質問に対してシステムが以下の表 2.1 のような検索結果を出力したとする．ここで，表の第 1 列目は順位，第 2 列目はその文書が与えられた検索質問に対して適合（○），不適合（×）であることをそれぞれ表している．このコレクション中には与えられた検索質問に対して適合する文書が全部で 5 つ含まれているとすると，ある順位までの文書を検索結果として採用したとすると，再現率，適合率はそれぞれ表の第 3 列，第 4 列のように計算できる．例えば，上位 4 文書を検索結果として採用したとすると，これらの中には適合文書が 3 つあるので（$|C| = 3$），適合率は 3/4 = 0.75 となる．一方，再現率は，5 つある適合文書のうち 3 つを検索することができたことになるので 3/5 = 0.60 となる．

表 2.1 検索結果の例

順位	適合性	再現率	精度
1	○	0.20	1.00
2	○	0.40	1.00
3	×	0.40	0.67
4	○	0.60	0.75
5	○	0.80	0.80
6	×	0.80	0.67
7	×	0.80	0.57
8	×	0.80	0.50
9	○	1.00	0.56
10	×	1.00	0.50

2.7.2 再現率－精度グラフ

表 2.1 からわかるように，一般に再現率と精度はトレードオフの関係にある．一般的に，検索結果の文書数を増やすと再現率は上がるが，精度は下がる傾向にある．再現率を横軸に，精度を縦軸にとって，検索文書数を変化させ，（再現率，精度）の点をプロットすると以下のような表ができる．このような表のことを再現率-精度グラフ（Recall-Precision graph）と呼ぶ．前述したとおり，再現率と精度はトレードオフの関係にあるので，再現率-精度グラフは一般には右下がりの曲線を描く．高い再現率でできるだけ高い精度を得ることがシステムの目的となり，曲線をできるだけ水平に近くなるようなシステムがより精度が高いといえる．

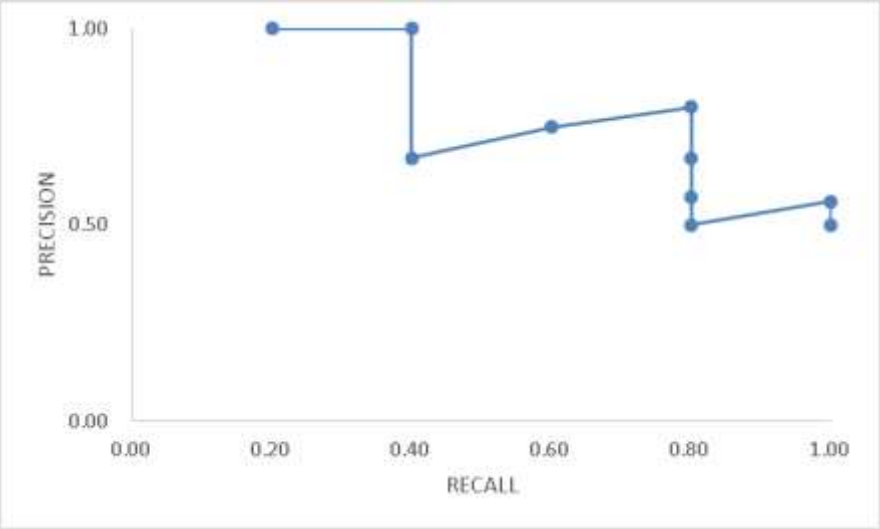

図 2.2　表 2.1 における再現率-精度グラフ

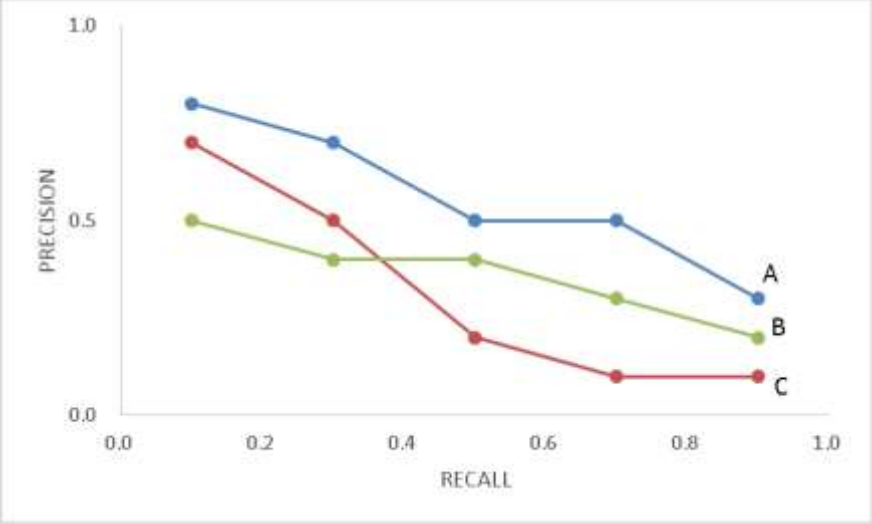

図 2.3　再現率-精度グラフによるシステムの比較

再現率-精度グラフを用いて，2 つのシステムを比較する場合は，2 つのシステムの曲線を同一グラフ上に描き，どちらの曲線が上にあるかで判断できる．曲線が上にあるシステムの方の性能が良いと言えるが，2 つの曲線が交差する場合はどちらが性能がよいかは一概には言えない．例えば，図 2.3 において，システム A はシステム B や C よりもグラフが上にあることから，システム A が最も性能が良いといえるが，システム B と C は途中で交差してしまっている為，どちらがよいとは一概にいうことができない．再現率の大きい，つまりグラフの左側についてはシステム B の方が有利であるが，再現率の低い，つまりグラフの右側についてはシステム C の方が有利である．どちらのシステムを利用するかはユーザが再現率と精度のどちらを優先するかによって決定される．

2.7.3 一般的な評価尺度

理想的には，適合率・再現率の両者を1に近づけることが望ましい．しかし，実際には両者はトレードオフの関係にあり，適合率を上げようとすると再現率が下がり，逆に再現率を上げようとすると適合率が下がるという現象が起こる．そのため，この二つの評価値を見るだけでは，どちらの方が良いとは一概には言えないという問題が起こる．そこで，両者の値を用いて，総合的な観点から評価としてF尺度（F-measure）がある．

F尺度とは，再現率と適合率の調和平均（逆数の平均値の逆数）をとったものである．この尺度は，再現率と適合率の双方の値が大きいときに，大きい値をとる．また，F尺度の取りうる値は0から1の範囲にあり，値が大きいほどよい．

$$F = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{2.7}$$

F尺度は，再現率と適合率の両方を考慮し，ひとつの値によって表すことができる．しかし，近年の大規模なテスト・コレクションでは再現率の計算に必要な検索質問に適合する文書数を得ることは非常に難しく，再現率を実際には計算することは不可能である．よって再現率は現実的ではなく理論的な指標であるといえる．これに対して，検索された文書数とその中の適合文書数は検索結果を調べればわかるため，これは現実的な尺度であるといえる．

そこで，適合率のみを用いて検索システムの評価を行う指標が提案されている．その中でも最も単純なものがP@Nである．これは検索結果の上位N個の文書の中でどれだけの文書が適合したかを表す指標である．

$$P@N = \frac{NUM(d_i)}{N} \ (d_i | i < N \ \& \ d_i \in D_R) \tag{2.8}$$

ここで，D_Rは適合文書集合，d_iは検索された文書中で適合した文書を意味する．実際，情報検索を行う場面を考えると，検索結果が100件出てきたとしても，通常であれば結果の上位だけをみて目的が果たされれば，下位の文書に目を通すことはない．つまり，P@Nはこのような面を考慮した指標であるといえる．

P@Nは検索結果の上位N個しか判断に用いないことは上述した通りであるが，再現率のような漏れの少なさもシステムの有効性として考慮されるべき点である．そこで，再現率も考慮した尺度として非補間平均精度（Average Precision; AP）である．これは，検索結果を1位から順に見ていき，適合文書があった時点における適合率を加算していって最後に算術平均をとったものである．

$$AP = \frac{1}{N} \sum_{d_i \in D_R} P@i \tag{2.9}$$

例えば，表 1 のような検索結果が得られていた場合，非補間平均精度は，(1.0+1.0+0.75+0.8 + 0.56)/9 = 0.46 となる．

非補間平均精度は検索質問ごとに値が得られるが，検索システムの評価を行う際には複数の検索質問が用意される．そこで，非補間平均精度を検索質問数で算術平均したものが検索システムの有効性の尺度としてよく用いられる．これを Mean Average Precision（MAP）と呼ぶ．なお，S は全てのクエリ集合を表し，Q_iはその中の i 番目のクエリを指す．

$$MAP = \frac{1}{N} \sum_{Q_i \in S} AP \tag{2.10}$$

ここで，一般的に適合率の高い検索質問があると MAP もあがりやすくなるという性質がある．このため，難しい検索質問の適合率に注目するために，MAP の平均の計算に算術平均ではなく幾何平均を用いた Geometric Mean Average Precision（GMAP）などがある．

$$GMAP = exp\left(\frac{1}{N} \sum_{Q_i \in S} \log(AP(Q_i) - 0.000001) \right) + 0.000001 \tag{2.11}$$

2.8 まとめ

本章では，既存研究における文書表現及び処理方法について概観した．情報検索においては「索引語」とよばれる文書を特徴付ける語を，統計量を用いることで選定することで，文書の内容を計算機上で表現をしている．一方，情報抽出や文書分類，文書要約では基本的には「索引語」の考え方を流用する場合が多く，計算機上で文書を扱うためには文書を文書の内容を表すような特徴的な単語で表現することは非常に重要である．

しかし，これらで用いられる手法では文書内に出現した単語のみから選ぶことが前提となっているが，この前提は十分でないこともある．例えば，新聞記事の検索においては，文書内に含まれていないキーワードが有益なキーワードとなる場合が多い．また，文書要約では文書外からの知識を用いることが困難であることから，現状では抄録にとどまっている．しかし，人間は文書外の知識も利用した文書要約も可能である．したがって，文書内だけの単語にとらわれない文書の単語表現は関連研究に共通した課題であるといえ，文書の内容を適切なキーワードで置き換えることは非常に重要である．

第 3 章　脳モデルに関する研究

3.1 はじめに

脳構造をモデル化し，それを情報処理システムに応用しようという考えは 20 世紀以降広まりつつある．この章では，現在までに提案されている脳をベースとしたモデルをいくつか紹介し，その利点と欠点について言及する．

3.2 ニューラルネットワーク

ニューラルネットワーク（Neural Network）とは，この言葉の通り，脳の神経回路網にヒントを得た情報処理モデルである．脳内には 100 億以上のニューロン（Neuron）と呼ばれる神経細胞が存在するといわれており，ニューロンは他の多数のニューロンと結合している．ニューロン同士は電気信号を受け取ったり，または発したりし合っている．脳科学によれば，この電気信号のやり取りにより，人間の脳内において情報処理が行われているという．また，ニューロンとニューロンを結合している部位はシナプス（Synapse）と呼ばれており，このシナプスを通じることによりニューロン間の電気信号の伝達が行われる．この信号の伝達効率はシナプスによって異なり，更にシナプスの活動状態などにより，シナプスに伝達効率は変化する．このことをシナプス可塑性（Synaptic plasticity）とよび，記憶や学習に重要な役割を持つと考えられている．ニューロンは，自分に繋がっているニューロンから受け取った電気信号の総和がある閾値を超えると，他のニューロンへと電気信号を発する．したがって，常時シナプスによってニューロン同士が物理的に接続されている訳ではない．

このようなニューロンやシナプスの特性をモデル化したものがニューラルネットワークである．

3.2.1 パーセプトロン

パーセプトロン（Perceptron）とは，1958 年にフランク・ローゼンブラット（Frank Rosenblatt）により提案された階層型ネットワークであり，最も古典的なニューラルネットワークである[17]．パーセプトロンはシンプルなネットワークでありながら，学習能力を持つ．最も有名なパーセプトロンは，以下の図 3.1 のような 3 層構造である．

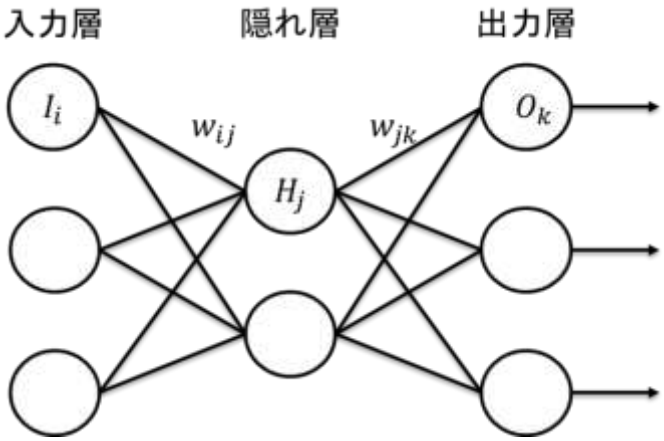

図 3.1　3 層構造のニューラルネットワーク

図中の i は入力層のノードのインデックス, j は隠れ層のノードのインデックス, k は出力層のノードのインデックスを示しており, w_{ij}は入力層の i 番目のノードと隠れ層の j 番目のノード間の重み, w_{jk}は隠れ層の j 番目のノードと出力層の k 番目のノード間の重み, I_iは入力層の i 番目のノードの値, H_jは隠れ層の j 番目のノードの値, O_kは出力層の k 番目のノードの値を意味する.

隠れ層と出力層のそれぞれのノードの値は, それぞれの層のノードが一つ前の層のノードから受け取った入力の総和が閾値を超えていた場合は 1, 下回っていた場合は 0 となる. したがって, 出力関数にはステップ関数が用いられる.

3.2.2　パーセプトロンの学習則

パーセプトロンの学習は, 隠れ層と出力層間でのみ行われる. したがって, 入力層と隠れ層のノード間の重みを表すw_{ij}は定数となる. パーセプトロンの学習は教師あり学習であり, 実際の出力値と教師信号との差が小さくなるように, 隠れ層と出力層間のノードの重みである結合荷重と閾値が調整される. つまり, パーセプロトンでの学習とは, この重みと閾値を求めることである.

教師信号T_kと出力層での出力値O_kとの誤差は$T_k - O_k$で求められ, この誤差から結合荷重の修正量ΔW_{jk}は以下の式のように求められる. なお, μ は学習係数であり, y_jは隠れ層の j 番目のノードの出力値を意味する.

$$\Delta W_{jk} = \mu(T_k - O_k)y_j \tag{3.1}$$

この修正量ΔW_{jk}を用いて, 閾値を含む結合荷重W_{jk}は次のように更新がされる.

$$W_{jk} \leftarrow W_{jk} + \Delta W_{jk} \tag{3.2}$$

3.2.3 パーセプトロンの問題点

このパーセプトロンは非常にシンプルなモデルとなっているが，1969 年に発表された Minsky らの研究報告により，排他的論理和のような線形分離不可能問題を解くことができないという問題点[18]がある．

例えば，以下のような図 3.2 があり，この出力を 2 つに分類したい問題があったとする．このとき，赤と青は同じ種類であるので，この赤と青を分類する為に直線をひくことを考えると，一本ではなく，図の緑の点線のような二本の直線が必要となってしまう．このような問題を排他的論理和（Exclusive OR）と呼ばれる．現実的な問題として，このような排他的論理和の問題は非常に多く，パーセプトロンがこの問題を解けないことは非常に致命的な問題と見なされている．

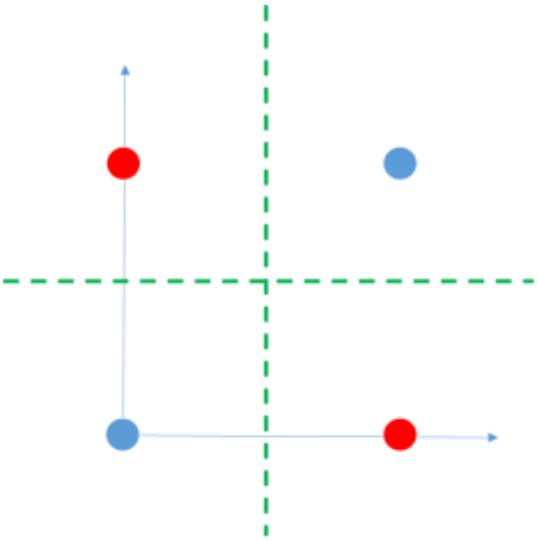

図 3.2　線形分離できない問題

3.3 バックプロパゲーション

上記の問題を解決するために，1986 年に，デビッド・ラメルハート（David Rumelhart），ジェフリー・ヒントン（Geoffrey Hinton），ロナルド・ウィリアムス（Ronald Williams）らによって，バックプロパゲーション（Back-propagation）とよばれる手法が提案された[19]．このモデルもパーセプトロンと同様なモデルであり，入力層，隠れ層，出力層から構成されるフィードフォワード型のモデルである．

パーセプトロンとの主な違いは以下の 2 点である．

I. 出力関数がステップ関数ではなく，シグモイド関数のような微分可能な連続関数を使用
II. 出力値と教師信号の 2 乗誤差を計算する

3.3.1 バックプロパゲーションの学習則

ネットワーク構造が図 3.1 と同様な時，バックプロパゲーションの学習は以下のように求められる．

(1) 隠れ層の j 番目のノードの出力値H_jを求める．但し，θ_jは j 番目のノードの閾値であり，関数 f(x)はシグモイド関数である．

$$H_j = \mathrm{f}\left(\sum_i w_{ij}\, I_i + \theta_j\right) \tag{3.3}$$

$$\mathrm{f(x)} = \frac{1}{1+e^{-x}} \tag{3.4}$$

(2) 出力層の k 番目のノードの出力値O_kを求める．但し，γ_kは k 番目のノードの閾値を意味する．

$$O_k = \mathrm{f}\left(\sum_j w_{jk}\, H_j + \gamma_k\right) \tag{3.5}$$

(3) 教師信号T_kと出力層のノードの出力値O_kから，出力層の k 番目のノードの誤差δ_kを求める．

$$\delta_k = (T_k - O_k)O_k(1 - O_k) \tag{3.6}$$

(4) 出力層の k 番目のノードにおける誤差と隠れ層の出力値H_jから，隠れ層の誤差σ_jを求める．

$$\sigma_j = \sum_k \delta_k\, w_{jk} H_j\left(1 - H_j\right) \tag{3.7}$$

(5) 隠れ層の j 番目のノードと出力層の k 番目のノード間の重みw_{jk}と，出力層の k 番目のノードの閾値γ_kを更新する．なお，α と β は学習係数である．

$$w_{jk} = w_{jk} + \alpha\delta_k H_j \tag{3.8}$$

$$\gamma_k = \gamma_k + \beta\delta_k \tag{3.9}$$

(6) 入力層の i 番目のノードと隠れ層の j 番目のノード間の重みw_{ij}と，隠れ層の j 番目のノードの閾値θ_jを更新する．

$$w_{ij} = w_{ij} + \alpha\sigma_j I_i \tag{3.10}$$

$$\theta_j = \theta_j + \beta\sigma_j \tag{3.11}$$

(7) 次の訓練データを入力層にセットし, (1)に戻る. 教師信号と出力値の平均 2 乗誤差 E が十分に小さくなるまで学習を繰り返す.

$$E = \frac{1}{2}\sum_k (T_k - O_k)^2 \tag{3.12}$$

3.3.2 バックプロパゲーションの問題点

パーセプトロンが解決できない線形分離不可能問題を解決したバックプロパゲーションであったが, このバックプロパゲーションにも 2 つの大きな問題点がある.

1 つ目は, 極小値に陥る可能性がある点である. バックプロパゲーションは, 学習を平均 2 乗誤差が小さくなるまで繰り返すことになるが, その小さくなった値が必ずしも最小値とは限らず, 極小値となってしまう可能性がある.

2 つ目は, 汎化能力不足な問題である. 一般に, バックプロパゲーションの学習では典型的な入力パターンであるものが用いられるが, 実際の入力にはそのような学習パターンには適合しないような入力も考えられる. したがって, 学習パターンから何らかの規則性などを見つけ出し, 学習パターンの中に存在しないような入力に対しても, 対応できることが求められるが, このような能力を汎化能力という. 学習をしすぎてしまうと, その学習パターンに特化しすぎた学習(過学習)を行ってしまい, この汎化能力が低下してしまうが, 反対に過学習をしないように学習したとしても, バックプロパゲーションは汎化能力があまり高くないことが知られている.

3.4 ディープラーニング

バックプロパゲーションの課題を解決したのが, 近年話題となっているディープラーニング(Deep Learning)である. ディープラーニングでは, 中間層の数を大きくすることで, より高精度な情報処理の実現を目指している. ディープラーニング自体の構想は昔から存在はしていたが, 近年の計算機の機能向上や, 多層化による過学習などの諸問題を克服するスパース・コーディングなどの新たな手法などにより, 実現が可能となった. その結果, ディープラーニングを用いた手法により, 文字や画像認識などの分野のコンテストで 2 位以下を圧倒する大差で 1 位を獲得[20]するなど, その有用性が示されている. また, Google の画像検索・音声認識や Apple の Siri などのシステムにも実用化されており, AlchemyAPI[21]のようにディープラーニングを手軽に利用できる API やツールも最近では開発されている.

ディープラーニングとは, 事前に特徴抽出してくれる Restricted Boltzmann Machine(RBM)を複数個繋げて多層化したものである. RBM は入力情報を圧縮し, 逆変換した時の誤差が最小にす

る機構を持っている. 学習済みの RBM に新たな RBM を追加し, 1 つ前の層の RBM の出力を入力にして次の層の RBM の学習をする. 学習済みの RBM に関しては何の処理もしない. RBM が多層化されたものをディープボルツマンマシン（Deep Boltzmann Machine; DBM）と呼ぶが, DBM にバックプロパゲーションを行う方法もある.

3.4.1 ディープラーニングの強み

ディープラーニングはまず, 高い精度が期待できる点が大きな強みである. 多層のニューラルネットワークにより, 一つ一つのノードにより肌理細かな認識を行う事ができるため, 非常に高い認識精度を実現可能である. その結果, 文字認識や画像認識などでの様々なコンペティションにおいて, 既存の手法よりも大幅に改良された実績を出している.

次に注目すべき点は, 特徴自体もディープラーニングが一緒に学習が可能となるという新たなパラダイムである. 例えば, 既存の画像認識においては, visual words など, システムの設計者によって特徴抽出が行われ, その抽出された特徴量を元に, 機械学習などの分類器を用いて出力をする必要性があった. 一方で, ディープラーニングではその特徴抽出自体もそのモデルの中に組み込まれている為, タスク固有な職人的知識を必要としない. 長い人工知能の研究において大きな問題とされていた適切な特徴抽出が自動的に行われることは, 非常に大きな意味がある.

3.4.2 ディープラーニングの弱み

高い精度を実現するディープラーニングにも問題は当然ある. 一つ目はパラメータとネットワーク構造である. ディープラーニングでは膨大なパラメータ量となり, 高い精度を実現する為の最適なパラメータを決定するのは非常に困難である. 具体的には, 中間層の数や, 制限つきボルツマンマシンのパラメータ, 学習の繰り返し回数等がある. 特に, ディープラーニングでは学習状況を可視化することは難しく, 学習回数の決定が非常に難しい. また, ネットワーク構造の設計も精度に大きな影響を与えるため, 精密なチューニングが必要となることもデメリットとなってしまっている.

もう一つの大きな問題は, 複雑な計算を必要とする故, 膨大な計算コストがかかることである. 計算機の性能が向上している今日でさえ, 学習にはたくさんの時間を要する. 例えば, ディープラーニングを用いて計算機にネコの画像を認識させる実験[22]では, 1000 台の計算機を用いて学習に 3 日間もかかってしまうなど, 大規模並列処理計算の為の環境やシステムを実装するためのコーディング技術などが必要不可欠な問題点もある.

また, ディープラーニングの為の自然言語処理として Bollegala が論文[23]にまとめているが, 言語処理に実際に適用しているような研究[24]はまだ例が少なく, 今後の課題であるといえる.

3.5 BESOM

脳科学の知見から，大脳皮質では脳の主要な情報処理が行われているといわれている．BESOM とは，その大脳皮質で行われている情報処理を神経科学と機械学習の観点からアルゴリズム化した確率的計算機構である[25]．現状では BESOM は未完成であるものの，生理学的な観点を重要視してモデル化された BESOM は，数少ない脳の数理モデルとして重要である．BESOM を構成する機械学習は，計算論敵神経科学の観点から妥当性が考慮されており，計算論的に導かれたアルゴリズムを実行する神経回路は大脳皮質の主要な解剖学特徴とよく一致している．

BESOM は主にベイジアンネットワークと自己組織化マップ（Self-Organizing maps; SOM），独立主成分分析，そして強化学習を組み合わせたものである．

3.5.1 ベイジアンネットワークと大脳皮質との関連性

ベイジアンネットワークは，複数の事象（確率変数）の間の因果関係をネットワーク構造化したものであり，因果関係の強さを確率値として保持する．具体的にはこの確率値は条件付確率であり，Conditional Probability Table（CPT）という表に保持されることになる．ベイジアンネットワークと大脳皮質は多くの関連性を持つといわれている[26]．例えば，大脳皮質の領野間の双方向結合がベイジアンネットワークと構造が似ていると指摘されている．その為，ベイジアンネットワークを用いた大脳皮質の神経回路モデルは既に幾つか提案されている[27][28]．

確率変数の状態の推定には様々なアルゴリズムが提案されている．確率伝播アルゴリズムはそのうちの一つである[29]．BESOM を提唱している一杉はこれを改良し，近似確率伝播アルゴリズムを提案している．これは大脳皮質の 6 層構造の解剖学的特徴と対応しており，BESOM が大脳皮質の情報処理を模していることを表す重要な証拠であるとしている．

3.5.2 自己組織化マップと大脳皮質との関連性

自己組織化マップ（Self-Organizing Maps; SOM）は，高次元数値ベクトルの形で与えられた入力データを低い次元に圧縮することができる．SOM の大きな特徴は競合学習と近傍学習とよばれる 2 つの学習により，上記の圧縮を実現している点である．

SOM は，一次視覚野に見られるコラム構造を再現させる神経回路モデルを工学的に扱いやすくするために単純化したモデルである．大脳皮質にあると考えられている競合学習と近傍学習を実現する機構がモデルに組み込まれており，生物学的に無理のないモデルとして提案されている[30]．

3.5.3 BESOM のアーキテクチャ

BESOM は，非循環有向グラフの形に結合したノードによって構成されている．また，BESOM ではノード一つ一つが SOM として扱われるので，図 3.3 のように，1 つのノードは複数のユニットから構成される．もし 2 つのノードが link で結ばれているならば，それぞれのノードに含まれるユニット同士も完全に link で結合していると見なされる．ユニット間の link は重みを持っており，この重みは学習により変化する．この link の重みは，ベイジアンネットワークの各ノードが保持している CPT の値と同値である．

BESOM の学習は，認識ステップと学習ステップを交互に繰り返すことで行われる．学習を完了した BESOM に入力が与えられると，教師ノードのユニットが決定され，そのユニットが示すものが入力に対する出力として得ることができる．

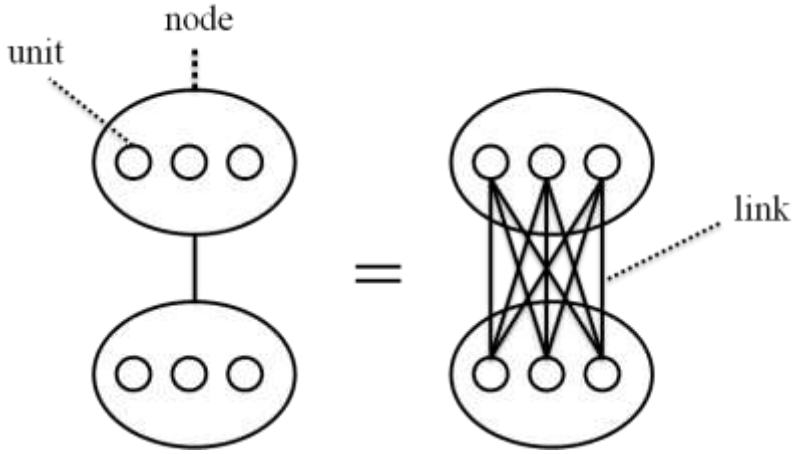

図 3.3 結合されたノード

【認識ステップ】

認識ステップでは，ノードのネットワークがベイジアンネットワークとして働く．したがって，各ノードは確率変数を意味する．また，1 つのノードの中にある各ユニットは，その確率変数が取りうる値に対応することになる．

外界からの入力と，各ノードが持つ CPT に従い，各ノードの値は推定され，Most Probable Explanation（MPE）が求められる．この MPE とは，ベイジアンネットワークにおいて，与えられた観測データを最もよく表す確率変数の値の組み合わせのことを指す．MPE は，認識ステップ終了後の学習ステップにて利用される．なお，CPT の初期値はランダムに与えられる．

【学習ステップ】

学習ステップでは，各ノードは SOM の競合層のように動作する．各ノードは，自分の子ノードから送られる入力ベクトルを学習され，SOM の学習結果は条件付確率として見なされる．更新された条件付確率は，次の認識ステップにて利用される．

ノード X は，n 個の子ノード Y（l=1,…,n）を持つとする．学習ステップでは，SOM は MPE における各子ノードの値を，1 と 0 の入力ベクトルの形で受け取る．つまり，ノードY_lが値y_j^l(j=0,⋯,s-1)を取りうるとすると，Y_lからの入力ベクトル$\mathbf{v}^l$の要素は以下のようになる．

$$v_j^l=\begin{cases}1\ (\text{MPE における}Y_l\text{の値が}\ y_j^l\text{の場合}) \\ 0\ (\text{その他の場合})\end{cases} \tag{3.13}$$

j はユニットy_j^lのインデックスであり，s はユニットy_j^lの数を意味する．

ノード X において，MPE の値を表すユニットが，競合学習における勝者となる．勝者ユニットでは，通常の SOM と同様，参照ベクトルを入力ベクトルに近づける．ここで，ノード X の勝者ユニットx_iとノードY_lのユニットy_j^lの間の結合の重みをw_{ij}^l，i をユニットx_iのインデックス，α を学習率とすれば，更新式は以下のようになる．

$$w_{ij}^l \leftarrow w_{ij}^l + \alpha\left(v_j^l - w_{ij}^l\right) \tag{3.14}$$

学習率 α の値は，更新する方法と一定値にする方法論がある．一定値の場合は，結合の重みは過去の経験を忘却し，最近の経験に比重を置いた条件付確率と解釈できる．

3.5.4 学習時の動作の流れ

図 3.4 は，学習時における BESOM の一連の動作を示したものである．

左上の Step1 が初期状態である．初期状態では，ネットワーク構造と CPT の初期値のみが与えられる．右上の Step2 は，入力と教師ノードの情報が与えられた状態を意味する．つまり，Input layer のノードのユニットと，teacher node のユニットが発火する．図 3.4 では，黒い丸が発火した状態のユニットを示している．左下の Step3 は認識ステップである．Step2 の入力と教師ノードの情報から，リンクの重みの総和が全てのユニットの組み合わせパターン毎に求められ，その中から最も総和が大きい組み合わせパターンが MPE として選ばれる．その結果，MPE として選ばれたユニットが発火することになる．右下の Step4 は学習ステップである．Step3 で求められた MPE に従い，リンクの重みが更新される．図 3.4 では，実線で示したリンクの重みが重くなるように更新され，点線で示したリンクの重みが軽くなるように更新される．以降，学習が完了するまで Step2 から Step4 までが繰り返されることになる．

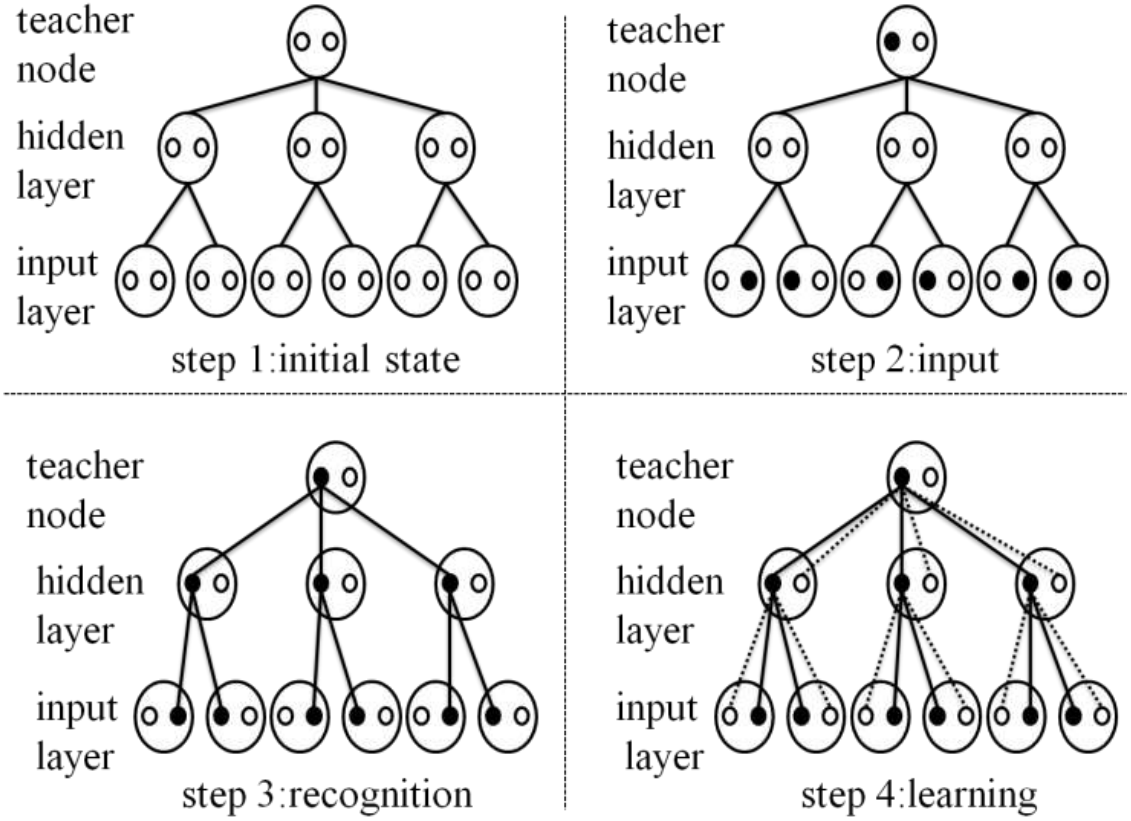

図 3.4 BESOM における学習

3.5.5 BESOM の問題点

BESOM には大きく 2 つの問題点が挙げられる.

1 つ目は, モデルとして未完成であり, 安定した動作が得られないことである. MNIST（手書き認識）などのタスクにおいて, ある程度の精度が出ている事が実証されてはいるが, まだ細かな点において改良中である為, 実用レベルで用いることができないのが現状である.

2 つ目は, 汎用性が低い点である. ディープラーニングのように特徴抽出の機構がモデルに含まれていないため, 既存の手法のようにシステムの設計者がタスク固有の特徴抽出を行う必要性があるが, BESOM のアルゴリズムに適した入力を設計することが困難である.

3.6 まとめ

最も有名である脳のモデルであるニューラルネットワークを発展させたディープラーニングは, 精度の面で非常に優秀であり, 今後期待できるモデルであるといえる. しかし, その代償として複雑さと計算速度の問題を抱えている. 確かにディープラーニングでは特徴抽出における専門的知識が不要となるが, 精度向上のためには, 大量のパラメータやネットワーク構造を最適化する必要性が

あり，この最適化の為には専門的な知識や経験がやはり必要となってしまう．これは大きな問題点として見なさざるを得ない．また，何故高精度が出るのかについて解明がなされていないのが現状であり，実システムとして用いる為にクリアしなければならない課題の一つである．

一方，BESOM も数少ない脳の数理モデルとして期待ができるモデルではあるが，モデルとして未完成であるため，残念ながら実用化には至っていない．

また，言語処理の観点からこれらのモデルをみれば，入力の次元数の問題がある．これらのモデルではモデルの構造上，基本的に入力の次元数が固定長である必要がある．この前提は画像などの分野では弊害にならないが，自然言語処理の分野では問題となってしまうことが多い．例えば，文書を処理対象とした場合，文書長は可変長であるため，単純にこれらのモデルに入力することができない．そのため，入力次元数を調整するなどの工夫が必要となるが，このような工夫はモデルとしては本質的な部分とはいえず，文書を取り扱っているモデルもまだほとんど提案されていないのが現状である．

第 4 章　Confabulation theory

4.1 はじめに

本研究では，脳の数理モデルである Confabulation theory を用いている．Confabulation theory は neuroscience から得られた知見に基づいて構築された理論であり，数学的メカニズムとしてはシンプルながらも，非常に強力なモデルである．本章では，Confabulation theory の基本的な考え方やその特徴について言及していくことにする．

4.2 Confabulation theory の概要

Confabulation theory とは，ロバート・ヘクトニールセン（Robert Hecht-Nielsen）によって提唱された人間の脳の認知機能を説明する理論である[31][32][33]．これは，Neuroscience に基づいた脳の認知モデルであり，実際の脳の働きから導出された理論である．ヘクトニールセンによれば，人間の脳には「モジュール（module）」と呼ばれる属性ごとに分割された領域と，モジュールの内部を表し，構成する「シンボル（symbol）」と呼ばれるものが存在するという．この理論では，物体は独立なシンボル間の相互作用により，物体は認識されることを仮定している．

例えば，いま図 4.1 のように，色と形と味覚を表す 3 つのモジュールがあるとする．もし，“赤”を示すシンボル，“丸”を示すシンボル，“甘い”を示すシンボルが発火したとし，その時にわたしたちがリンゴを見たとする．すると，シンボル間をつなぐナレッジリンク（knowledge）が各発火したシンボルとリンゴを表すシンボル間に独立に形成される．つまり，ここでは 3 つのナレッジリンクが形成されることになる．このシンボル間のナレッジリンクの形成により，次に私たちがリンゴを見たときに，各シンボルがナレッジリンクを通じて再び活性化し，その結果，リンゴシンボルが活性化することで，私たちはリンゴであると認識することができる．

加えて，winners-take-all という競争プロセスにより，モジュール内で複数のシンボルが発火しても，その中で最も活性化したシンボルのみが選ばれる．

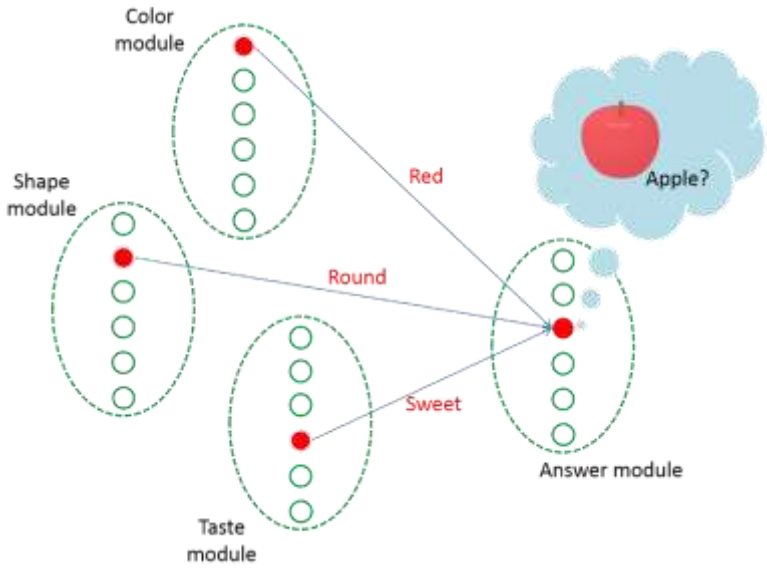

図 4.1　認知メカニズム

4.3　Confabulation theory の基本要素

Confabulation theory は以下の4つの基本要素から構成されている.

i. Thalamocortical Module
ii. Knowledge Link
iii. Confabulation
iv. The origin of behavior

これら4つの基本要素を説明しながら, もう少し詳細に Confabulation theory について説明する.

4.3.1　Thalamocortical Module

全ての認知機構（見たり聞いたり, 言葉や推論等）を含む人間の脳内の「情報処理」は大脳皮質（cerebral cortex）や視床（thalamus）によって行われているのではないかという様々な手かがりが neuroscience によって, 近年導き出されている. また, このような認知プロセスに用いられる“認知知識（cognitive knowledge）”は大脳皮質によって蓄積されているのではないかという証拠も neuroscience で発見されている. そこで, Confabulation theory では大脳皮質の構造に着目している.

図 4.2 のように, 人間の大脳皮質には約 4,000 の個々に独立したモジュールと呼ばれる集まりが存在するといわれている. 各モジュールは 1 つの属性を表現する為に用いられる. この属性とは, 我々が認知している object のことを指す. 例えば, 視覚や聴覚などを例として挙げれば, “色”や

“形”などが属性となる.

1 つのモジュールは, そのモジュールの属性の値を示す為に数千のシンボルから構成されている. 例えば, もしあるモジュールが物体の名前を表現するものだった場合, このモジュールは「リンゴ」から「情報」, 「経済」などの全ての単語を数十万のシンボルへと符号化した状態で保持されているはずであろう. 但し, ある物体の名前が想起された時, ただ一つだけのシンボルのみが活性化され, 他のシンボルは活性化されない.

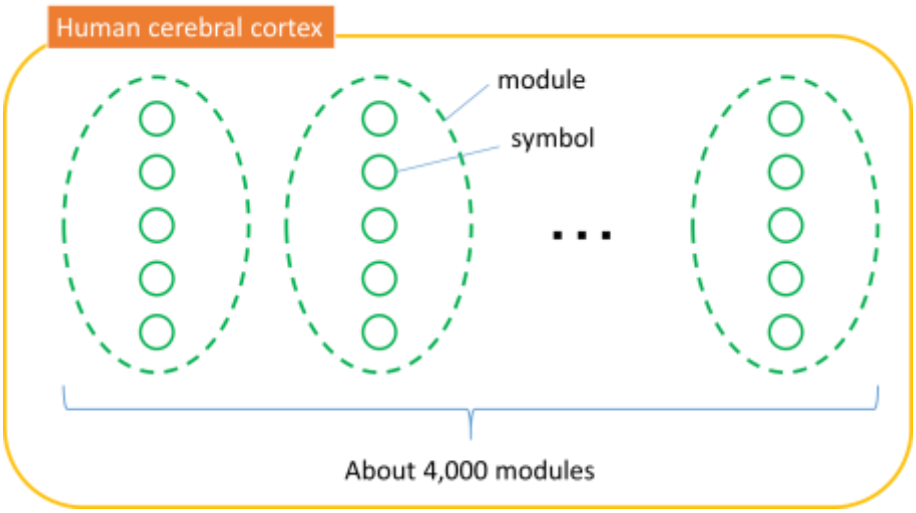

図 4.2　人間の大脳皮質内の構造

実際には, 各モジュールは脳に局所的に存在している皮質パッチとよばれるものに構成されており, confabulation theory では, 1つの属性値を示すシンボルは皮質パッチ内に存在するニューロンの集合によって表現されていると想定している. この皮質パッチの大きさは $45mm^2$ であり, この中には約 450 万のニューロンが含まれているが, 実際にシンボルを表現する為のニューロンの数はその 10%である 45 万である.

4.3.2 Knowledge Link

Confabulation theory では, 全ての認知は単純で均一である“ナレッジリンク”と呼ばれるものを利用していると仮定している. 普通の人間はこのナレッジリンクを何十億個も持っているとされており, これは生活する中で 1 秒あたりに 1 つ以上の比率でリンクを学習していることを意味している. 各個々のナレッジリンクは source symbol とよばれる一つのシンボルを表現するためのニューロンの集合と, target symbol とよばれる source symbol とは異なるモジュールに所属するシンボルを表すためのニューロンの集合とを結んでいる.

例えば, 外部刺激により source symbol を表すニューロン達が活性化すると, 軸索を通じて, source symbol が所属するモジュール外の 100 万以上のニューロン達が活性化される. この結果, target symbol を表すニューロンまで伝達され, 活性化することにより, source symbol と target symbol

が共起した状態となる．ナレッジリンクは 2 つのシンボルが初めて共起した際に即座に形成され，形成されたリンクは永続的なものとなる．形成されたナレッジリンクは，source symbol と target symbol が共起する度に強化される．これは一般に Hebb 則と呼ばれているものと同値である[34].

図 4.3 は「リンゴ」と「赤」を表すシンボル間のナレッジリンクの生成過程を示している．外部刺激により「色」を表すモジュール中の「赤」シンボルが source symbol として活性化した時，軸索を通じて，多数のニューロンが活性化された後，言語を表すモジュール中の「リンゴ」シンボルが target symbol として活性化されたとする．この時，この2つのシンボル間は共起した状態となり，ナレッジリンクが形成されることになる．つまりナレッジリンクとは，source symbol を表すニューロンの集合から target symbol を表すニューロンの集合までの伝達経路のようなものであるといえる．

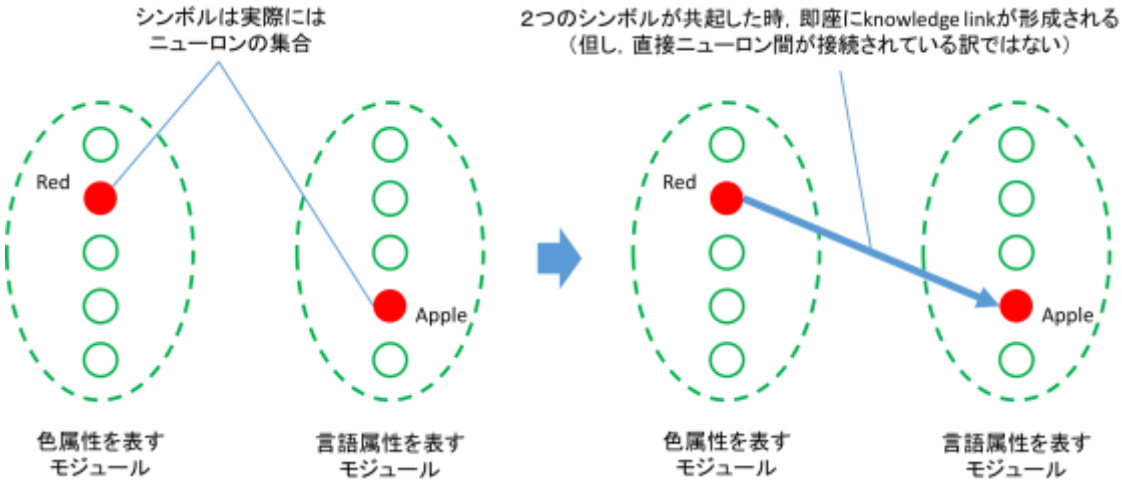

図 4.3　Knowledge Link の形成

4.3.3 Confabulation

Confabulation theory では，全ての認知活動は単純な情報処理操作によるものであると仮定しており，この操作のことを Confabulation と呼んでいる．この Confabulation は，モジュールが何らかの思考命令信号（thought-command-signal）を受け取った時にだけ動作する．なお，この思考命令信号はアナログな値（連続値）であり，デジタルな値（離散値）ではない．Confabulation の開始時はこの思考命令信号の値は 0 か非常に小さな値であるが，時間とともに急速に増幅する．

図 4.4 は Confabulation の動作例を示している．各色で円が塗りつぶされているほど活性度が高いことを意味する．思考命令信号を受け取り，Confabulation が開始した直後の時刻t_0においては，モジュール中の複数のシンボルがナレッジリンクを通じて，異なる活性度で活性化しているが，時間の経過に伴う思考命令信号の急速な増加により，全入力に対する平均活性度が最も高いシンボルのみが他のシンボルとの競争に勝ち残ることになる．この処理は非常に短時間で行われる．Confabulation theory では，このように，シンボル間で競争が行われ，その結果唯一つのシンボルのみが勝ち残るメカニズムを winners-take-all と呼んでおり，その勝ち残ったシンボルのことを

confabulation の conclusion と呼んでいる. Confabulation では多くの並列処理が行われており, 大体 80ms で終了する.

また, 各モジュールは筋肉の収縮と全く同じように単一の連続値によって制御されていることから, モジュールは muscle of thought であると見なすこともできる.

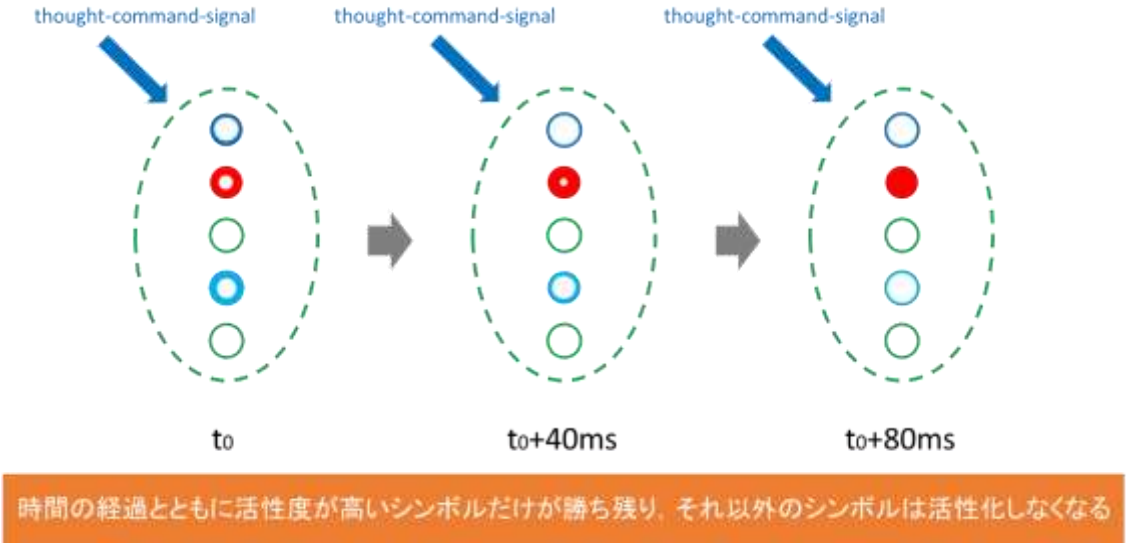

図 4.4 Confabulation の動作例

4.3.4 The origin of behavior

Confabulation theory では, 図 4.5 のように様々なモジュールで常に Confabulation が最も信頼できる結論(definitive conclusion, winners-take-all により最終的に勝ち残った唯一つのシンボルのこと)や特定のシンボルに関連した行動命令(action command)を実行していると仮定している. つまり, 我々の microbehaviors と呼ばれる瞬間的な次の動作の断片的な動きや思考プロセスは全て Confabulation の動作の結果, 結論が導かれ, その結論から生じる行動命令によるものであり, それが絶えず連続的に繰り返されることで, 我々は一連の動作を行う事ができるという.

但し, どこかに旅行に行くことを決断するといった高次の行動は頻繁に起こるようなものではなく, 大抵“suggestion”として扱われ, 行動を行う前に他の認知モジュールにより精査されることになる.

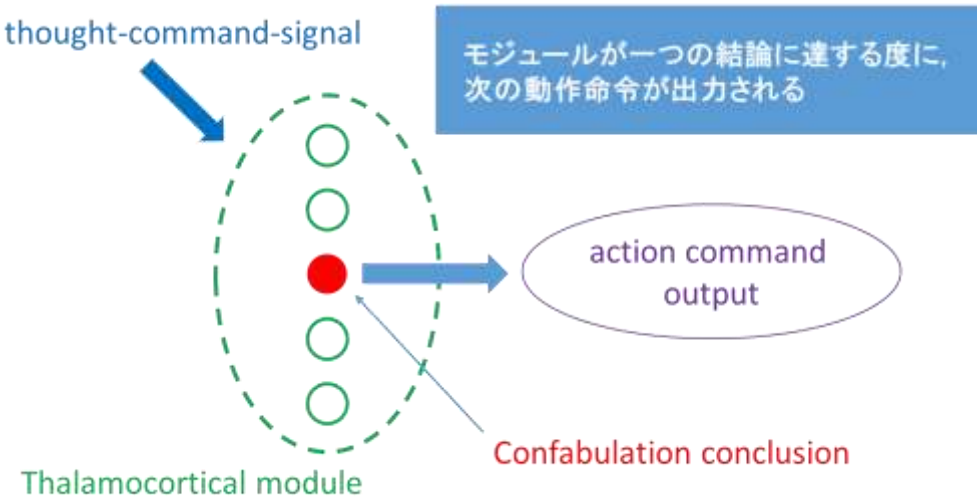

図 4.5　動作命令の生成

4.4　Cogency

Confabulation theory における認知計算の目的は，Cogency（適切さ，尤もらしさ）が最大となる結論を表すシンボルである confabulation conclusion を発見することである．Cogency とは，結論シンボルの尤もらしさを示す指標であり，確率計算によって求めることが出来る．ここでいう結論シンボルとは，Confabulation を行った結果，最終的に全ての事実（source symbols）から最も支持を受ける，つまり source symbols が活性化ときに同時に最も共起するシンボルとして尤もらしい answer モジュール中の target symbol のことを指す．

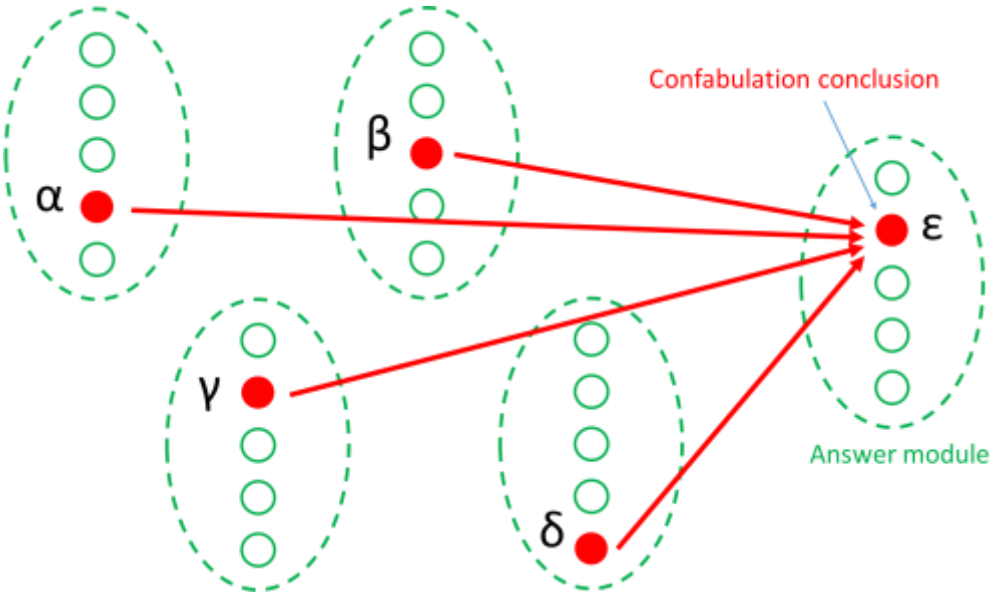

図 4.6　Confabulation

例えば，図 4.6 が示すように，入力シンボルとして異なるモジュールに所属するα，β，γ，δが外部刺激を受けて活性化したとする．各シンボルは利用できる全てのナレッジリンクを通じて，answer モジュール中のニューロンに刺激（axon）を送る．最終的に結論として選ばれる answer モジュール中の target symbol は，全ての target symbol の中で Cogency が最大となる target symbol であり，図ではεが結論を表すシンボルである confabulation conclusion となっている．

Cogency は先ほど述べたように，確率計算によって求めることが出来，具体的には$\mathrm{p}(\alpha\beta\gamma\delta|\varepsilon)$として計算が行われる．Cogency は target symbol であるεが真であるという仮定が与えられた時，仮定された事実が真であるという確率を意味している．これは一般的な考えとちょうど逆の向きになっている．一般的には，事実が生じた時に結論が起きる確率を求める，と考えるほうが直感的ではあるが，Cogency では結論が起きた時に，事実が生じる確率を求めていることと同値である．Confabulation theory では，認知に関わる各意思決定プロセスは，実際に真であることが分かっている事実を最も支持している結論を選択することであると仮定している．Cogency の最大化は人工知能や神経生物学，またはコンピュータサイエンスにおける思考という仕組みを何十年も支配してきた“Bayesian”の考え方からすると，先ほど述べたとおり，異なった考え方で計算されることになる．

Bayesian の考え方は，80 年前にロナルド・フィッシャー（R.A. Fisher）の研究で提唱されたものである．物体の特性を計測するシステムや最適なパターン分類などの為に，Bayesian は考案された．これらの分類器は，基礎確率理論からのベイズの定理を用いて条件付確率が導出される為，“ベイズ分類器”として知られるようになり，人間は優れたパターン分類器であるので，認知とは”Bayesian”に違いないと信じられるようになった．このような見解は後に，どんな状況においても正解である確率が最も高い確率を持つものが，最も妥当な結論であるという原理に拡張されることになった．この”Bayesian”の見解は直感的には良いものであり，価値ある応用技術を多く生み出してきたが，Confabulation theory においては認知モデルとしてはふさわしくないと主張している．なぜならば，”Bayesian”のアプローチでは neuroscience から得られた知見と一致しないからである．

4.5 Cogency の計算方法

いま，source symbol としてα，β，γ，δが活性化したときに，answer モジュール中のシンボルεが活性化したと仮定する．また，$\mathrm{p}(\alpha\beta\gamma\delta\varepsilon) > 0$とし，ε以外の answer モジュール中のシンボルλは全て$\mathrm{p}(\alpha\beta\gamma\delta\lambda) = 0$とする．従って，$\mathrm{p}(\alpha\beta\gamma\delta|\varepsilon) = \mathrm{p}(\alpha\beta\gamma\delta\varepsilon)/\mathrm{p}(\varepsilon) > 0$であり，$\mathrm{p}(\alpha\beta\gamma\delta|\lambda) = \mathrm{p}(\alpha\beta\gamma\delta\lambda)/\mathrm{p}(\lambda) = 0$である．このとき，以下の定理が成立する．

定理 3.1：もし$\alpha\beta\gamma\delta \Rightarrow \varepsilon$であるならば，cogency の最大化は一意となり，ε のみが解となる

上記の定理により，アリストテレスの「論理的な情報環境」下において，cogency の最大化は論理的な解を導くことを示している．また，Cogency の最大化は duck test のように動作するという．もし，アヒルと同じサイズの生き物がアヒルのように鳴いたり，歩いたり，泳いだり，飛んだ場合，我々はそ

れをアヒルと見なすであろう. 実際には, この生き物がアヒルであるという論理的な保証はどこにもない. にもかかわらず, Cogency の最大化により, 結論を導くことが出来る.

次に, 実際に Cogency の最大化の為の式の導出を行う. いま, Cogency に対して確率のチェインルール（$\mathrm{p}(abcde) = \mathrm{p}(a|bcde)\mathrm{p}(b|cde)\mathrm{p}(c|de)\mathrm{p}(d|e)$）を適用すると, $\mathrm{p}(\alpha\beta\gamma\delta|\varepsilon)$は以下の 4 通りの式のように変形が可能である.

$$\mathrm{p}(\alpha\beta\gamma\delta|\varepsilon) = \frac{\mathrm{p}(\alpha\beta\gamma\delta\varepsilon)}{\mathrm{p}(\varepsilon)} = \mathrm{p}(\alpha|\beta\gamma\delta\varepsilon)\cdot\mathrm{p}(\beta|\gamma\delta\varepsilon)\cdot\mathrm{p}(\gamma|\delta\varepsilon)\cdot\mathrm{p}(\delta|\varepsilon) \tag{4.1}$$

$$\mathrm{p}(\alpha\beta\gamma\delta|\varepsilon) = \frac{\mathrm{p}(\beta\gamma\delta\alpha\varepsilon)}{\mathrm{p}(\varepsilon)} = \mathrm{p}(\beta|\gamma\delta\alpha\varepsilon)\cdot\mathrm{p}(\gamma|\delta\alpha\varepsilon)\cdot\mathrm{p}(\delta|\alpha\varepsilon)\cdot\mathrm{p}(\alpha|\varepsilon) \tag{4.2}$$

$$\mathrm{p}(\alpha\beta\gamma\delta|\varepsilon) = \frac{\mathrm{p}(\gamma\delta\alpha\beta\varepsilon)}{\mathrm{p}(\varepsilon)} = \mathrm{p}(\gamma|\delta\alpha\beta\varepsilon)\cdot\mathrm{p}(\delta|\alpha\beta\varepsilon)\cdot\mathrm{p}(\alpha|\beta\varepsilon)\cdot\mathrm{p}(\beta|\varepsilon) \tag{4.3}$$

$$\mathrm{p}(\alpha\beta\gamma\delta|\varepsilon) = \frac{\mathrm{p}(\delta\alpha\beta\gamma\varepsilon)}{\mathrm{p}(\varepsilon)} = \mathrm{p}(\delta|\alpha\beta\gamma\varepsilon)\cdot\mathrm{p}(\alpha|\beta\gamma\varepsilon)\cdot\mathrm{p}(\beta|\gamma\varepsilon)\cdot\mathrm{p}(\gamma|\varepsilon) \tag{4.4}$$

上記の 4 式を全て掛け合わせると,

$$\begin{aligned}[\mathrm{p}(\alpha\beta\gamma\delta|\varepsilon)]^4 = {} & [\mathrm{p}(\alpha|\beta\gamma\delta\varepsilon)\cdot\mathrm{p}(\beta|\gamma\delta\varepsilon)\cdot\mathrm{p}(\gamma|\delta\varepsilon)]\cdot[\mathrm{p}(\beta|\gamma\delta\alpha\varepsilon)\cdot\mathrm{p}(\gamma|\delta\alpha\varepsilon)\cdot\mathrm{p}(\delta|\alpha\varepsilon)] \\ & \cdot[\mathrm{p}(\gamma|\delta\alpha\beta\varepsilon)\cdot\mathrm{p}(\delta|\alpha\beta\varepsilon)\cdot\mathrm{p}(\alpha|\beta\varepsilon)]\cdot[\mathrm{p}(\delta|\alpha\beta\gamma\varepsilon)\cdot\mathrm{p}(\alpha|\beta\gamma\varepsilon)\cdot\mathrm{p}(\beta|\gamma\varepsilon)] \\ & \cdot[\mathrm{p}(\alpha|\varepsilon)\cdot\mathrm{p}(\beta|\varepsilon)\cdot\mathrm{p}(\gamma|\varepsilon)\cdot\mathrm{p}(\delta|\varepsilon)]\end{aligned} \tag{4.5}$$

更に, 最初の 4 つのカッコの条件付確率の式にベイズの定理を適用すれば,

$$\begin{aligned}[\mathrm{p}(\alpha\beta\gamma\delta|\varepsilon)]^4 = {} & [\mathrm{p}(\alpha\beta\gamma\delta\varepsilon)/\mathrm{p}(\beta\gamma\delta\varepsilon)\cdot\mathrm{p}(\beta\gamma\delta\varepsilon)/\mathrm{p}(\gamma\delta\varepsilon) \\ & \cdot\mathrm{p}(\gamma\delta\varepsilon)/\mathrm{p}(\delta\varepsilon)]\cdot[\mathrm{p}(\beta\gamma\delta\alpha\varepsilon)/\mathrm{p}(\gamma\delta\alpha\varepsilon)\cdot\mathrm{p}(\gamma\delta\alpha\varepsilon)/\mathrm{p}(\delta\alpha\varepsilon)\cdot\mathrm{p}(\delta\alpha\varepsilon)/\mathrm{p}(\alpha\varepsilon)] \\ & \cdot[\mathrm{p}(\gamma\delta\alpha\beta\varepsilon)/\mathrm{p}(\delta\alpha\beta\varepsilon)\cdot\mathrm{p}(\delta\alpha\beta\varepsilon)/\mathrm{p}(\alpha\beta\varepsilon)\cdot\mathrm{p}(\alpha\beta\varepsilon)/\mathrm{p}(\beta\varepsilon)]\cdot[\mathrm{p}(\delta\alpha\beta\gamma\varepsilon)/\mathrm{p}(\alpha\beta\gamma\varepsilon) \\ & \cdot\mathrm{p}(\alpha\beta\gamma\varepsilon)/\mathrm{p}(\beta\gamma\varepsilon)\cdot\mathrm{p}(\beta\gamma\varepsilon)/\mathrm{p}(\gamma\varepsilon)]\cdot[\mathrm{p}(\alpha|\varepsilon)\cdot\mathrm{p}(\beta|\varepsilon)\cdot\mathrm{p}(\gamma|\varepsilon)\cdot\mathrm{p}(\delta|\varepsilon)]\end{aligned} \tag{4.6}$$

もし, 右辺の最初の4つのカッコ内の式の値のどれかが 0 となるが, 最後の 5 番目のカッコの式の値が 0 とならない場合は, 例外的なケースと言われる. 右辺の各カッコ内の最初の確率式は全て等しく$\mathrm{p}(\alpha\beta\gamma\delta\varepsilon)$であるので, 約分をして式を整理すると, 次のような式と定理を導くことが出来る.

定理 3.2: 例外でない事実と仮定されるシンボルを α, β, γ, δ, これから予測する要素を ε とすれば, $\mathrm{p}(\alpha\beta\gamma\delta|\varepsilon)$と$\mathrm{p}(\alpha|\varepsilon)\cdot\mathrm{p}(\beta|\varepsilon)\cdot\mathrm{p}(\gamma|\varepsilon)\cdot\mathrm{p}(\delta|\varepsilon)$の間には以下のような関係性が成立する.

$$[\mathrm{P}(\alpha\beta\gamma\delta|\varepsilon)]^4 = [\mathrm{p}(\alpha\beta\gamma\delta\varepsilon)/\mathrm{p}(\alpha\varepsilon)] \cdot [\mathrm{p}(\alpha\beta\gamma\delta\varepsilon)/\mathrm{p}(\beta\varepsilon)] \cdot [\mathrm{p}(\alpha\beta\gamma\delta\varepsilon)/\mathrm{p}(\gamma\varepsilon)] \cdot [\mathrm{p}(\alpha\beta\gamma\delta\varepsilon)/\mathrm{p}(\delta\varepsilon)] \cdot [\mathrm{p}(\alpha|\varepsilon) \cdot \mathrm{p}(\beta|\varepsilon) \cdot \mathrm{p}(\gamma|\varepsilon) \cdot \mathrm{p}(\delta|\varepsilon)] \tag{4.7}$$

この定理 3.2 の要となる意味を探る為に，具体例として，5 つの異なる，しかし同一の機能を表すモジュールを考えることにする．いま，各モジュールは 10,000 のシンボルを持っており，各シンボルは英単語を示すと仮定する．更に，事実を表すシンボルである α，β，γ，δ はそれぞれモジュール 1, 2, 3, 4 に所属しているとし，文章中の 4 つの連続した単語を表現しているとする．今，結論シンボルとして任意のシンボル ε を考えるとすれば，$\alpha\ \beta\ \gamma\ \delta$ というフレーズに最も適切な単語を選択することと同値となる．つまり，シンボル ε は最大の Cogency $\mathrm{p}(\alpha\beta\gamma\delta|\varepsilon)$を持つ単語として決定される．

実際に，フレーズ $\alpha\ \beta\ \gamma\ \delta$ を”the train was going”とし，シンボル ε を予期可能な”south”として考えてみることにする．すると，以下のように表現できる．

$$\mathrm{p}(\alpha\beta\gamma\delta\varepsilon)/\mathrm{p}(\alpha\varepsilon) = \mathrm{p}(the\ train\ was\ going\ south)/\mathrm{p}(the * * * south) \tag{4.8}$$

$$\mathrm{p}(\alpha\beta\gamma\delta\varepsilon)/\mathrm{p}(\beta\varepsilon) = \mathrm{p}(the\ train\ was\ going\ south)/\mathrm{p}(* \ train\ * * south) \tag{4.9}$$

$$\mathrm{p}(\alpha\beta\gamma\delta\varepsilon)/\mathrm{p}(\gamma\varepsilon) = \mathrm{p}(the\ train\ was\ going\ south)/\mathrm{p}(* * \ was\ * south) \tag{4.10}$$

$$\mathrm{p}(\alpha\beta\gamma\delta\varepsilon)/\mathrm{p}(\delta\varepsilon) = \mathrm{p}(the\ train\ was\ going\ south)/\mathrm{p}(* * * \ going\ south) \tag{4.11}$$

ここで，*は確率計算において考慮されない単語の位置を示している．例え，シンボル ε が表す単語が south から他の語（e.g. north, east etc…）に置き換えられたとしても，上記 4 つの式の分母の確率値はほとんど変わらないと考えられる．したがって，定理 3.2 における右辺の最初の 4 つのカッコの式は，シンボル ε に対してほとんど定数であるとみなすことができ，Cogency を最大化するための式としては結果として以下の式で十分であるといえる．

$$Cogency\ \mathrm{p}(\alpha\beta\gamma\delta|\varepsilon) \approx \mathrm{p}(\alpha|\varepsilon)\mathrm{p}(\beta|\varepsilon)\mathrm{p}(\gamma|\varepsilon)\mathrm{p}(\delta|\varepsilon) \tag{4.12}$$

この式は非常に重要である．なぜならば，過去の経験の中に全ての要素が完全に一致するような入力例は稀であり，ほとんどの場合は$P(\alpha\beta\gamma\delta|\varepsilon)$を計算することができないからである．例えば，赤くて丸いリンゴは一般的ではあるが，人間はリンゴが緑色をしていてもリンゴであると認識できるし，全く同じ赤色をしていたり，同じ大きさであったりするリンゴは存在しないからである．

私たちは“リンゴは赤い”といったように，ターゲットシンボルと入力シンボル間の相互関係を独立に学習している．このような分割された学習は，新たな事象の学習において重要な役割をし，私たちは過去の知識の欠片を統合することで事象を認知することができる．

実際には式(4.12)は以下の式(4.13)のように対数を用いて表現され，閾値 p_0 以上の確率値をもった入力シンボルのみが活性化される．

$$Cogency(\alpha\beta\gamma\delta,\varepsilon) = \log\left(\frac{P(\alpha|\varepsilon)}{p_0}\right) + \log\left(\frac{P(\beta|\varepsilon)}{p_0}\right) + \log\left(\frac{P(\gamma|\varepsilon)}{p_0}\right) + \log\left(\frac{P(\delta|\varepsilon)}{p_0}\right) \quad (4.13)$$

4.6 Confabulation と N-gram model

Confabulation theory は認知のモデルであることから，言語モデルと厳密に比較することは難しいが，Confabulation theory の特徴が分かりやすい為，ここでは既存の言語モデルと対比しながらその特徴を述べていくことにする．

まずは一般的な言語モデルである N-gram model について説明する．この N-gram model は，情報理論の創始者として知られるクロード・エルウッド・シャノン（Claude Elwood Shannon 1916-2001）が考えだした言語モデルである[35]. N-gram model とは，「ある文字列の中で，N 個の文字列または単語の組み合わせがどの程度出現するか」を調査する言語モデルを意味する．このモデルを扱う前提として，文字列や単語の発生確率が直前の文字列や単語に依存すると仮定し，次に予測をしたい target word の前に連続した単語を1つの単語群として扱う．例えば,図 4.7 のように“α β γ δ”のような 4 つの連続した文字列が与えられた場合、N-gram model では式(4.14)のように、”α β γ δ”という単語群が出現した後に出現する単語 ε の条件付き確率$P(\varepsilon|\alpha\beta\gamma\delta)$を求めることで 5 つ目の単語 ε を予測する．

$$n-gram(\alpha\beta\gamma\delta,\varepsilon) = P(\varepsilon|\alpha\beta\gamma\delta) \quad (4.14)$$

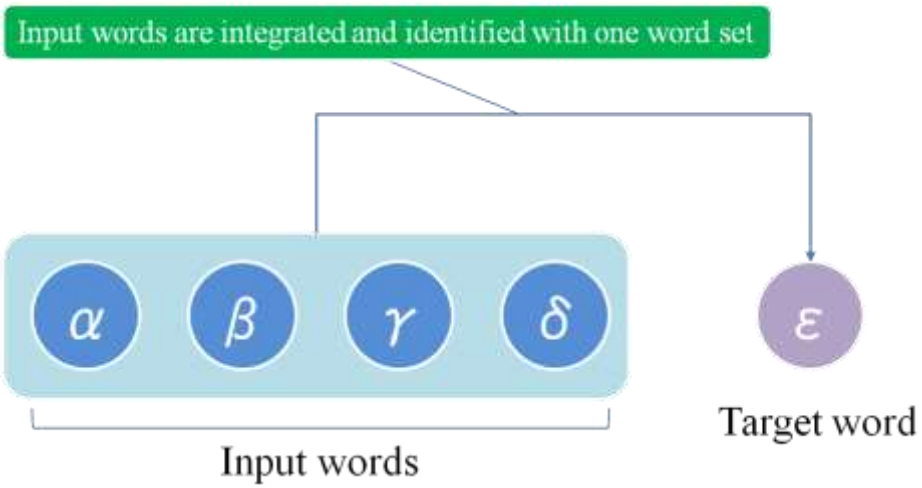

図 4.7 N-gram model

自然言語処理におけるこのモデルの有用性はすでに示されている．しかしながら，N-gram model の仕組みは非常に単純であり，実際の言葉や文字は直前の文字列だけに束縛されるわけではないことから多くの欠点も浮き彫りとなっている．例えば，有名な問題としてスパースネス問題

(0 頻度問題)が挙げられる. この N-gram model を用いて次に出現する語を予測する場合, 直前に出現した文字列全てが一致するもののみが答えとして出力されるが, その文字列の組み合わせが存在しなかった場合は, 頻度 0 となる. つまり, 全ての文字列が一致した文章が学習データの中に1つでも存在しなければ, 頻度 0 と見なされ, 解が出力されないことになる. しかしながら現実には, 学習データで頻度 0 だからといって, 世の中にその文字列の組み合わせが存在する確率が 0 であるわけではなく, また, 人間は全て同一の文字列の組み合わせを学習したことがなかったとしても, それなりに確からしい解を求めることができる点において, 言語モデルとして大きな欠点を抱えているといえる.

一方で Confabulation theory では次の図 4.8 に示すように考えられる.

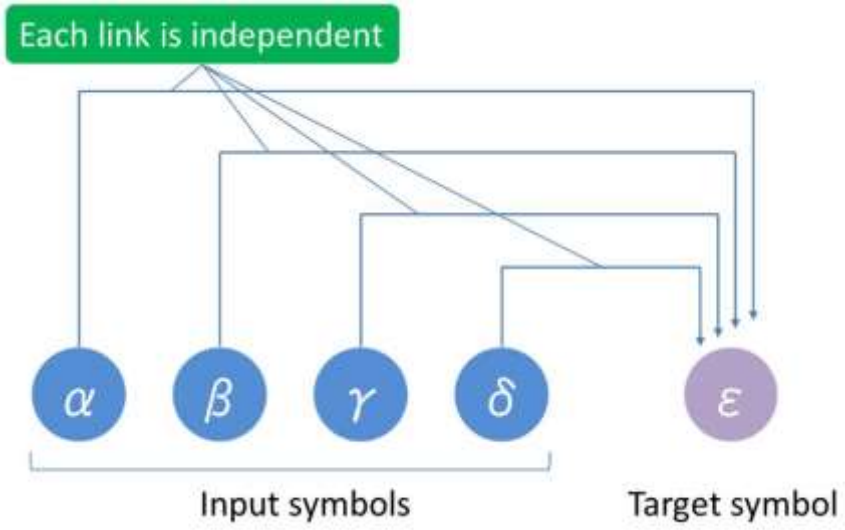

図 4.8 Confabulation theory による実装

通常の N-gram model は語列をひとつのグループとして扱うのに対し, Confabulation theory では語列中の語をそれぞれ独立に扱う.

前述した式(38)のように, Confabulation theory では, α の 4 つ後に ε が出現する確率$P(\alpha|\varepsilon)$, ベータの3つ後に ε が出現する確率$P(\beta|\varepsilon)$, というように個々の条件付き確率の乗算に比例する. これにより, 例え学習データ中に" α "という語列そのものが出現したことがなかったとしても, 異なる文章中にて, α の 4 つ後, β の 3 つ後, γ の 2 つ後, δ の 1 つ後にそれぞれ ε が出現したことがあれば, $P(\alpha|\varepsilon) \sim P(\delta|\varepsilon)$は正の値を持つことになり, 解を得ることができる.

また, N-gram model と Confabulation theory でもう一つの大きな違いは target symbol との関連性を示す条件付確率の捉え方の違いである. N-gram model は"Bayesian"にのっとり, 前向き条件付確率, すなわち$P(\varepsilon|\alpha\beta\gamma\delta)$が用いられるが, Confabulation theory では, neuroscience に基づいた結果, $P(\alpha\beta\gamma\delta|\varepsilon)$が用いられる.

以上のことより, Confabulation theory は N-gram model よりも柔軟性に単語の予測を行う事が出

来る点で非常に優れていると考えられ，より人間の思考に近い解が得られる事が期待できる．

4.7 まとめ

本章では本研究で利用した脳の認知モデルについて言及した．Confabulation theory は脳の認知機構を説明する理論であり，人間の脳の柔軟かつ汎用性が高い構造をモデル化していると同時に，モデルが非常にシンプルである故，汎用性が高く様々な分野への適用が期待できる．

Confabulation theory は 4 つの基本要素から成立している．1 つめはモジュールとであり，人間の大脳皮質には約 4,000 のモジュールと呼ばれる領域が存在し，各モジュールは 1 つの属性を表現するために用いられる．また，モジュールはその属性の値を示す数千のシンボルから構成されている．2 つめはシンボル間を繋ぐ knowledge link であり，シンボル間が共起する度に knowledge link は強化される．3 つめは Confabulation と呼ばれる認知の単純な情報処理操作であり，外的刺激により複数のシンボルが活性化しても，シンボル間で競争が行われ，その結果唯一つのシンボルのみが勝ち残る．このメカニズムを winners-take-all メカニズムという．4 つめは我々の思考プロセスは全ての Confabulation の動作の結果から導き出されたものであり，絶えず連続的に繰り返されることで，人間は一連の動作を行うことができる．

Confabulation では Cogency とよばれる尤もらしさを確率計算によって求め，Cogency 最大化により結論を導く．この Cogency 最大化は，独立な各モジュール間の結論を統合した結果で表される．このことは n-gram モデルと呼ばれる言語モデルと比較しても優れた点である．つまり，過去の経験の中で全ての要素が一致する事象がなくても，過去の知識の欠片を統合することで結論を導き出す事ができることを意味しており，人間の認知をよく説明できている．

第 5 章　Confabulation theory に基づいた Automatic Keyword Annotation System

5.1 はじめに

本章では，コーパスとして日本経済新聞を用い，自動的に記事にキーワードを付与する”Automatic Keyword Annotation System”について述べる．日本経済新聞（日経）は，日本において有名な経済新聞[36]である．この新聞記事は主に，タイトル，本文，キーワードの 3 種類から構成されている．本研究では，我々は各新聞記事に付与されたキーワードから編集者のキーワードパターンを学習し，新たな記事に対して適切なキーワードを自動的に付与する「日経アノテーター」を提案する．

5.2 モジュールの設計

4 章で説明したとおり，Confabulation theory ではモジュールの定義が非常に重要である．そこで，本節では，提案手法でのモジュールの定義について述べる．

我々が今まで行ってきた研究[37][38]では，モジュールを時系列として定義してきた．例えば，Confabulation theory を用いたイベント予測の研究では，4 日後を予測したい場合，1 日目，2 日目，3 日目をそれぞれ独立した source モジュールとして定義し，answer モジュールとして 4 日目を定義した．また，モジュール中のシンボルは，その日の新聞記事中に出現した単語と定義した．この定義では，各日に出現した単語と answer モジュール中のシンボルである 4 日目の新聞記事中に出現した単語には何らかの関係性があるとみなしている．実はこの定義は非常に緩い定義である．

各日の新聞記事では複数の単語が出現しており，answer モジュールである 4 日後の新聞記事内にも複数の単語が出現することになる．つまり，ここで大きな問題となるのは「どの単語がどの単語が出現するキーとなった単語であったか」である．しかし，これは一般に自明ではなく，更に非常に難しい問題である．なぜならば，ある現象というものは，通常複数の現象からの影響により発生しており，厳密に求めることは非常に困難であるからである．つまり“単語 A が出現後に単語 B が出現する”といったような簡単なルールで抽出することはほとんど不可能ということである．また，現実世界においては，“何日後に影響が与えたか”ということに時間的な揺らぎが発生する事が容易に想像できるため，この観点からも因果性を考えることは困難である．

以上のように，厳密な因果性を考慮することは非常に難しいことから，「出現した単語同士には何らかの関係性がある」とみなし，更にこれを拡大解釈し，「全ての出現した単語間には何らかの関係

性がある」という緩い条件を用意し，図 5.1 のように出現した単語の全ての組み合わせを生成して学習を行った．全ての組み合わせを生成してしまうことで，膨大な組み合わせが生じることになってしまうが，本当に意味のある関係性であるならば，確率値として顕著に現れるであろうと我々は想定した為である．

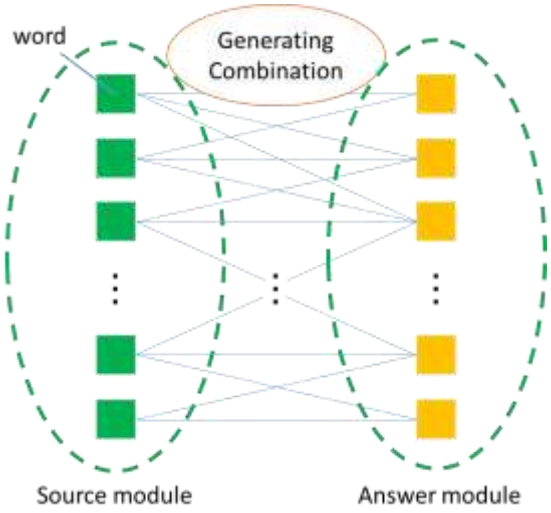

図 5.1　総組み合わせの学習

実験の結果，どの記事にもよく出現する“経済”や“政治”のような「常連語」がたくさん得られる結果となった．確かにこのような語はよく出現し，どの単語ともよく共起するため，関係性があるといえ，必ずしも間違いとは言えないが，我々が期待したようなその日独特の特徴的な単語を予測することは困難であった．

この原因は「全ての単語間には何らかの関係性がある」という定義が緩すぎるのではといった点もあるが，我々は最も大きな要因は，シンボル間の関係性以前に，そもそも「source モジュールと answer モジュールとの間に明確な関係性」が保証されていなかった為ではないかと考えた．

ここでもう一度ヘクトニールセンらが例題として扱った，文章の予測問題に着目することにする．ヘクトニールセンらは 3 単語を入力として与え，4 単語目を予測するような実験を行っている．つまり，入力の 1 単語目，2 単語目，3 単語目を source モジュールとして定義し，4 単語目を answer モジュールとして定義していることになる．したがって，この場合の学習は，1 単語目が出現した後に何が 4 単語目に出現したか，といった学習になる．図 5.2 では，入力語として”She could determine”を入力した結果，出力として”whether”が得られた実験結果を示している．

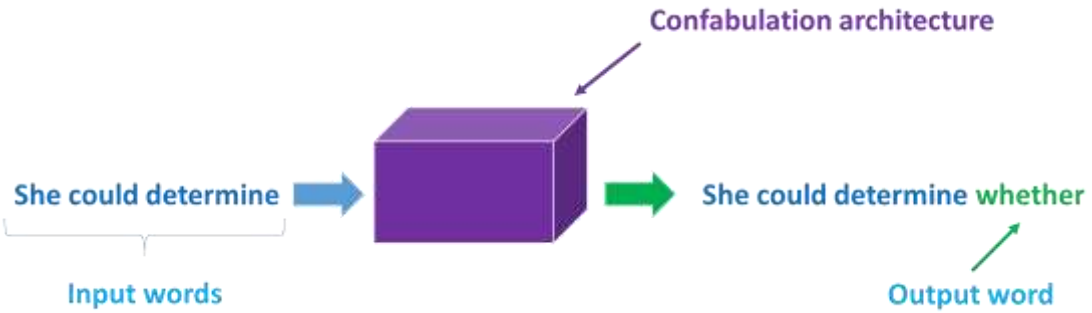

図 5.2　Sentence Cognition

ヘクトニールセンらが行った文章の場合の実験では，source モジュールと answer モジュール中の二単語間には，位置という観点から明確な関連性が保証されている点に比べ，我々のイベント予測のシステムの実験では source モジュールと answer モジュール間の間に厳密な関連性が保証されているとは言いがたい．

そこで，我々はこの反省を踏まえ，より明確な関連性が保証されるようにモジュールの定義を考えた．今回我々が用いるテキストデータは日経新聞であり，日経新聞の記事は大きく分類して以下の図 5.3 のような構成となっている．

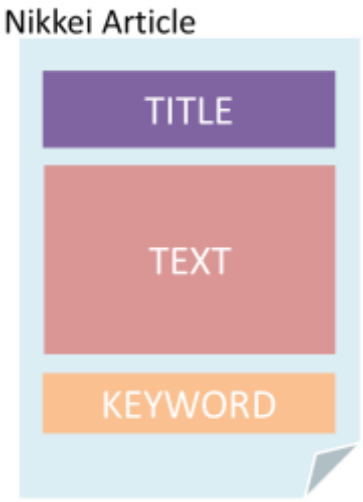

図 5.3　日経新聞の記事の構成

記事には記事のタイトル，本文，そしてキーワードが記述されている．今回のシステムの目標は新しい記事が入力として与えられた時に，この記事に付与すべきキーワードを予測することである．

ところで，森ら[39]によれば，次の図 5.4 のように一般的な新聞記事は以下の 3 つの要素で構成されている．

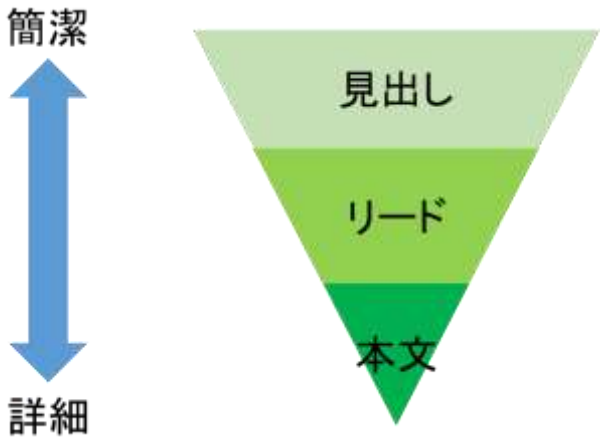

図 5.4　新聞記事の構造

森らによれば，見出しとは「記事の題名があり，読者が初めに注目する要素」であり，リードとは「本文の前に書いてある文章」，本文とは「内容を見出しやリードより詳しく書いた文章」のことを意味する．つまり，記事のタイトルとは読者に内容を簡潔に伝える機能をもち，本文は内容を詳細に伝える機能を持っていることから，この二つは大きく異なる役割を持っているといえる．

したがって，我々は以下の図 5.5 のようにタイトルと本文の 2 つの領域から入力を受け取る source モジュールと定義し，出力であるキーワードを answer モジュールとして今回定義することとした．

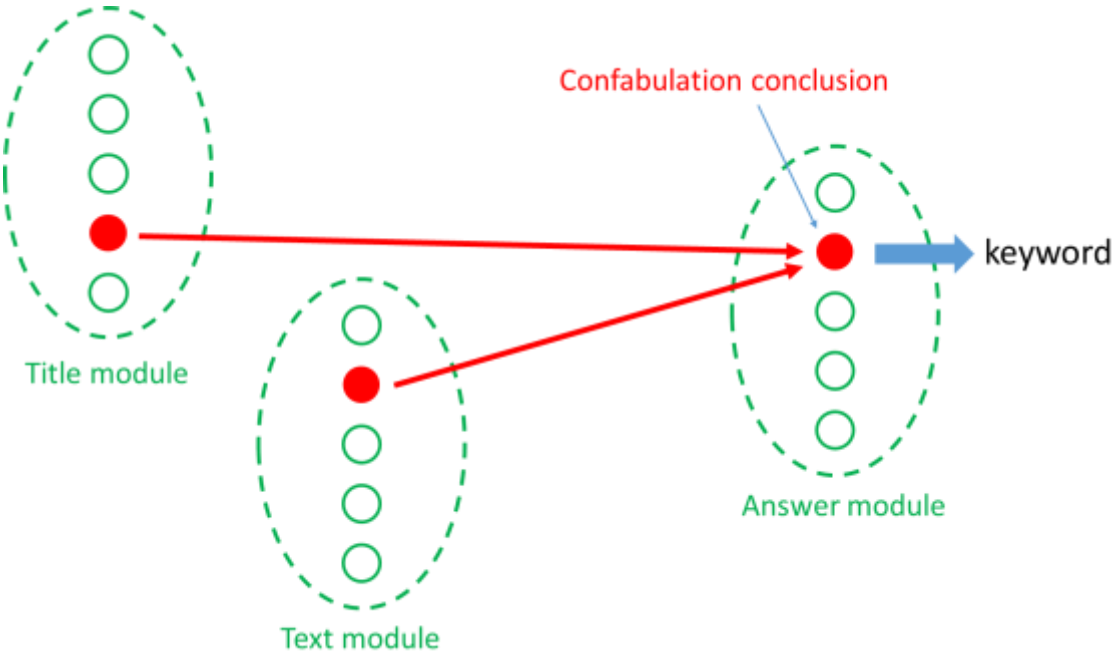

図 5.5　提案手法によるモジュールの定義図

5.3 コーパス

まず，提案手法の具体的アルゴリズムを説明する前に，前処理の部分であるコーパスからのデータ抽出について延べることにする．

下記の図 5.6 は，記事のフォーマットを表しており，提案手法に必要なデータはここから抽出される．

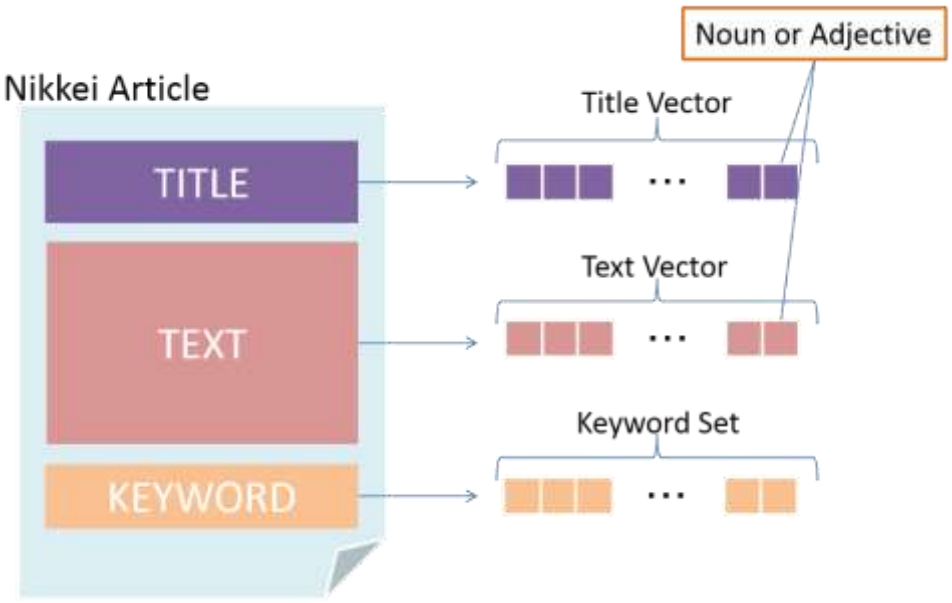

図 5.6　日経新聞の記事

まず，タイトルと本文に対して前処理を行う．全てのタイトルと本文に対して形態素解析を行い，一般名詞と形容詞のみを抽出する．この時抽出される語は二文字以上の文字列のみとした．その結果，以下の表 5.1 で示している数の単語を抽出ができた．

表 5.1　1987 年における記事の特徴

	単語種類数	1 記事の平均出現数
Title	34,717	23.48
Text	71,136	55.66
Keyword	94,025	9.16

形態素解析によって，助詞などの単語は取り除かれてしまっている為，正確ではないが，表から本文はタイトルの約 2 倍の文書長であることがわかる．また，キーワードには形態素解析を行わない．キーワードの総種類数は 94,025 キーワードと非常に種類数が多いことや，1 記事には平均 9 キーワードが付加されていることがわかった．

次に，タイトルと本文，それぞれから抽出された語を用いてベクトルを生成し，それぞれタイトルベクトルと本文ベクトルと名づけることにする．5.2 節で説明したとおり，本来モジュール中のシンボルは winners-take-all メカニズムにより，ただ一つのシンボルのみが活性化される．しかしながら，新聞記事において，ただ一つの単語のみで各モジュールであるタイトルと本文の内容を十分に表現することはできない．したがって，我々は複数のシンボルの活性化を容認し，winners-take-all メカニズムを拡張することにした．

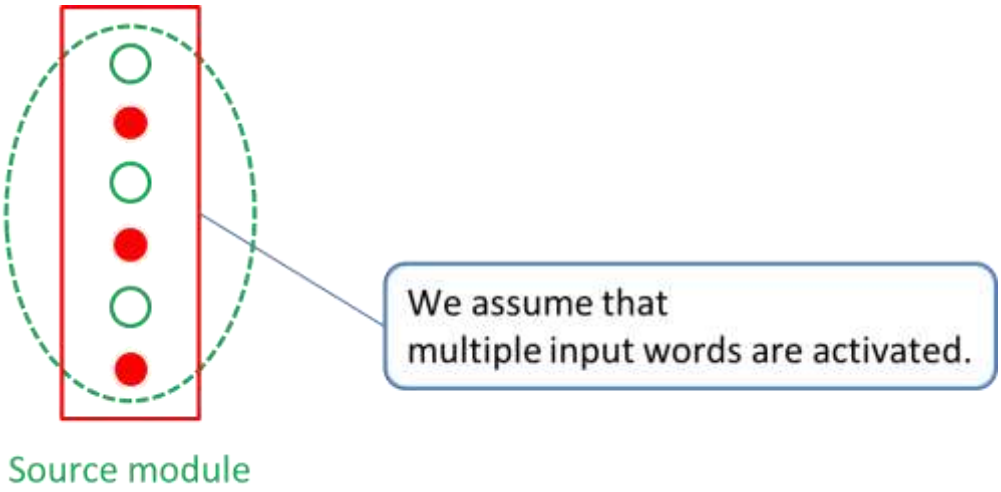

図 5.7　複数シンボルの活性化

しかし，これでは同一モジュール中のシンボルは全て同等な扱いをすることになってしまう．複数シンボルを活性化させる場合は，シンボル間の活性度に優劣をつけることができれば，winners-take-all メカニズムの自然な拡張になると考えられる．そこで，図 5.8 のようにシンボルごとに活性度を定義し，この活性度を用いることで，複数のシンボルの活性度に優劣をつけることとした．

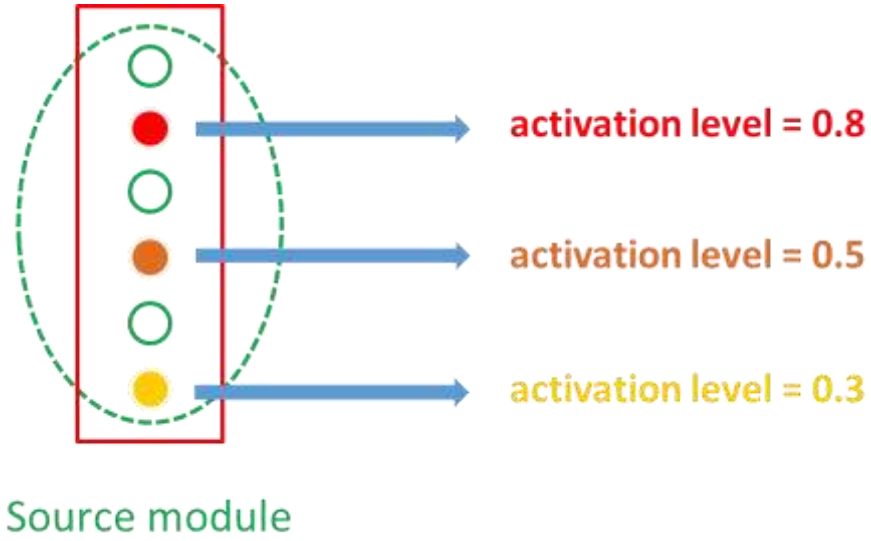

図 5.8 活性度を用いた複数シンボルの活性化

具体的には，この活性度として用いた重みは TFIDF（term frequency × inverse document frequency）であり，これらのベクトルに TFIDF を付加する．いま，w を単語，d を文書，N を総文書数とすると，TFIDF は以下のように求められる．

$$\mathrm{TFIDF}(w,d) = \frac{tf(w,d)}{\sum_{k \in d} tf(k,d)} * \log\left(\frac{N}{df(w,d)}\right) \tag{5.1}$$

この TFIDF は入力の単語を全て同等に扱わず，入力単語の中で何が重要であるかを示す為に用いられる．加えて，キーワードには TFIDF のスコアも今回は付加しないことにする．なぜなら，キーワードはプロの編集者によって付加されてものであり，全てのキーワードは等しく記事を描写している為，キーワード間に優劣はないと判断したからである．

5.4 System Description

図 5.9 は日経アノテーターのシステムの全体像を表しており，日経アノテーターは主に学習部分と推論部分から構成されている．

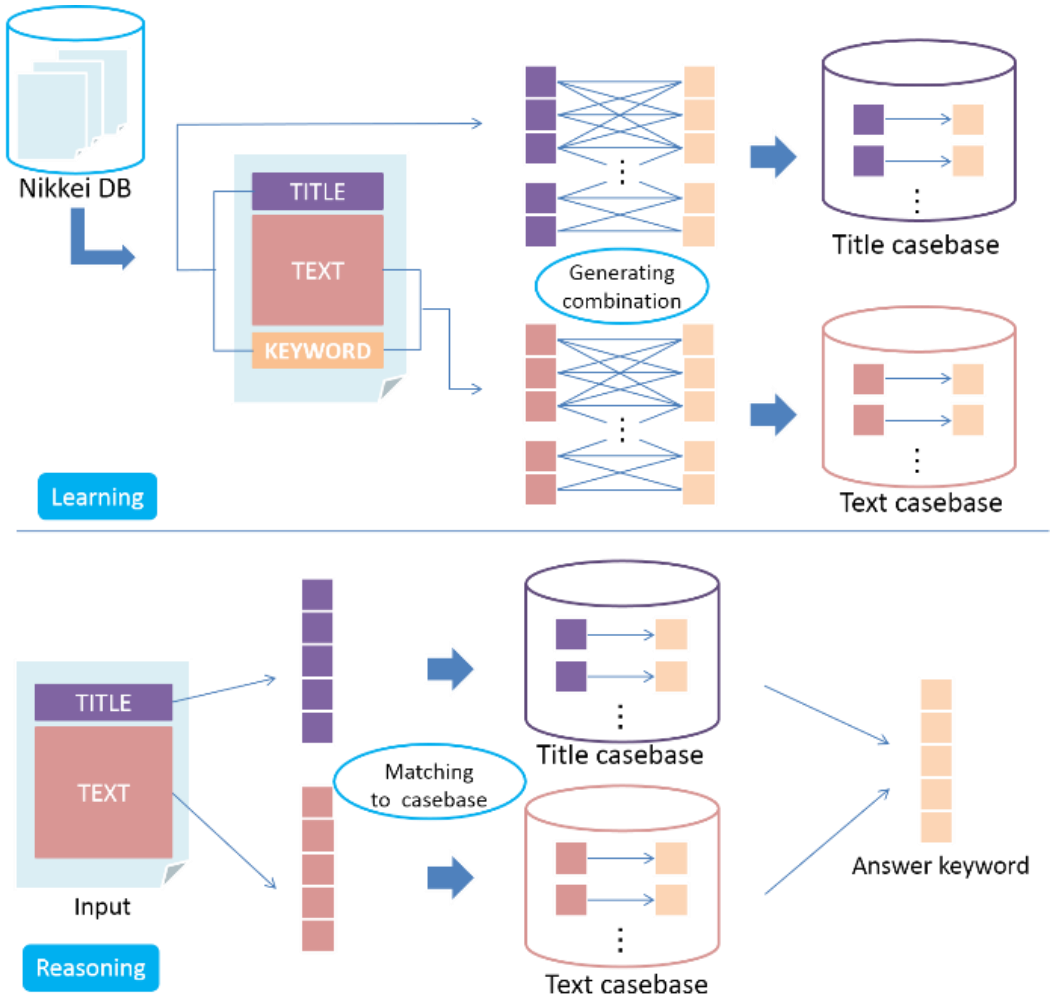

図 5.9　日経アノテーターの全体像

5.4.1 学習

学習部分では，タイトルベクトル中の単語と文書に付与されたキーワード，本文ベクトル中の単語と文書に付与されたキーワード，といった単語とキーワード間の関係性を学習する．具体的には，キーワードと各ベクトル中の単語の関係性を示す後向き条件付確率が計算されることになる．いま，キーワードを ε ，入力語を α とし，$n(\varepsilon)$を全記事におけるキーワード ε の出現数，$n(\alpha \cap \varepsilon)$を入力語 α とキーワード ε が同一記事に出現した回数とすると，後向き条件付確率は，

$$P(\alpha|\varepsilon) = \frac{n(\alpha \cap \varepsilon)}{n(\varepsilon)} \tag{5.2}$$

となる.

ここで先述したとおり, Confabulation theory に基づけば, 我々はタイトルと本文の部分を source モジュール, キーワードの部分を answer モジュールと見なすことができ, source モジュールでは各ベクトルの単語, answer モジュールでは一つのキーワードをモジュール中のシンボルであると見なすことができる.

本質的には, winners-take-all メカニズムにより, 一つのモジュールにつき, ただ一つだけのシンボルが活性化, つまりただ一つだけの単語が選ばれることになる. しかし, たった一つの単語だけではタイトルもしくは本文の内容を全て表現することができないことから, 各ベクトルから一単語のみを選択することは非常に難しい.

そこで我々は 5.2 節で述べたように,「全ての出現する単語間にはなんらかの関係性がある」という条件を導入し, 全ての出現する単語間の組み合わせを生成し, 学習を行う事にした. つまり, 同一モジュール中の複数のシンボルが活性化したことを認めることで, winners-take-all メカニズムを拡張した. 具体的には, このシステムでは同一記事中の全てのキーワードとタイトル・本文中に存在した全ての単語の組み合わせを事例として生成し, 入力語と出力キーワード間のナレッジリンクを生成する. これは, どのキーワードがどの入力単語と関連性を持っているかは自明ではない為である.

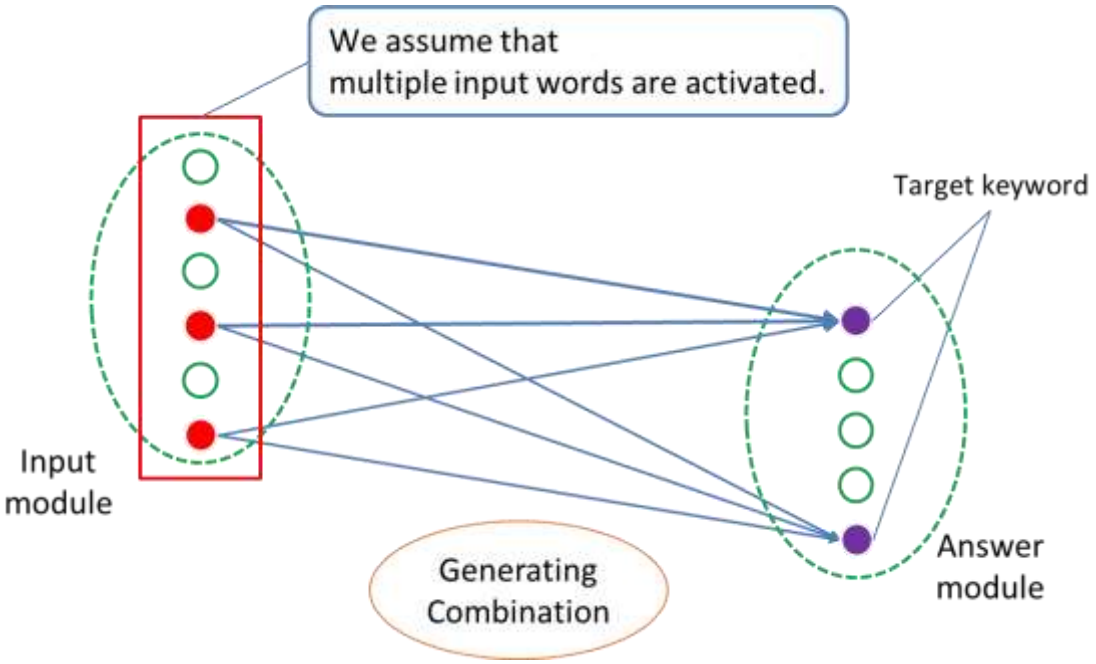

図 5.10 複数シンボル活性化における学習方法

しかしながら, この学習方法では, 5.3 節で述べたように, 複数の活性化したシンボルを全て同等に扱うことになるため, 自然な winners-take-all メカニズムの拡張となっていない. そこで, シンボルの活性度として TFIDF を用いることにし, 学習するナレッジリンクの強度の重みとして用いることにする.

以上を実現する為の基本的な考えは, あるキーワード k が出現した時の, 入力語 w の活性度を全て記憶しておくことで,「過去の事例において, 入力語 w はどの程度キーワード k の出現に影響を与えているか」を考慮することである. つまり, キーワード k が出現する時にいつも入力語 w の TFIDF 値が大きければ, 入力語 w はキーワード k の出現に大きな影響を与えていると考えることができ, 逆にいつも入力語 w の TFIDF が小さければ, キーワード k の出現に大きな影響を与えていないと考えることができる.

この重みの記憶方法は 2 種類考えられる. 一つ目は式(5.3)のように単純に総和をとって記憶する方法であり, もう一つは式(5.4)のように平均値を求めて記憶する方法である. なお, case とはキーワード k と入力語 w が同時に出現している記事の記事集合を意味している. どちらが適した重みかどうかについては後述する実験結果を踏まえた上で述べることにする.

$$weight(w,k) = \sum_{d \in case} TFIDF(w,d) \tag{5.3}$$

$$weight(w,k) = \sum_{d \in case} \frac{TFIDF(w,d)}{|case|} \tag{5.4}$$

$$case = \{d | d \ni w \cap k\}$$

5.4.2 推論

推論部分では, タイトルベクトルと本文ベクトルは学習部分と同様の方法によって, 入力記事のタイトルと本文から抽出される. この処理の後の推論では学習と同様な問題を抱えている. 即ち, 各ベクトル中の単語は winners-take-all メカニズムに従えば, 一つの単語のみしか活性化しない.

したがって, 学習部分と同様にして, 複数のシンボルの活性化を容認することにする. このとき, 入力単語の影響度には優劣があると考えられる為, その重みとして TFIDF を学習同様に用いることとした.

最後に出力キーワードの求め方について述べる. 繰り返しになるが, winners-take-all メカニズムを拡張したことにより, 各 source モジュール中の複数シンボルの活性化が容認されている. そのため, 各モジュールでの結論の出し方は少し複雑となり工夫が必要である. つまり, 本来の confabulation では, 図 5.11 の左のように, source モジュールと answer モジュールで活性化するシンボルはそれぞれ一つだけの為, 活性化したシンボル同士の関係性を求めればよかった. しかし, source モジュ

ール中の複数シンボルの活性化を容認したことにより，いま注目する answer モジュール中のキーワード k と入力モジュールである source モジュールとの関係性を評価するためには，図 5.11 の右のようにすべての活性化した source モジュール中のシンボルとキーワード k との関係性を評価しなければならない．

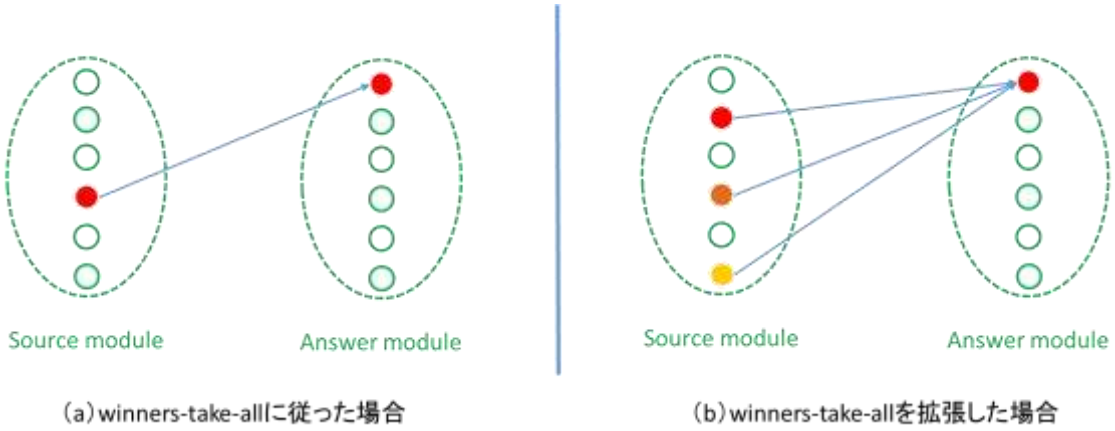

図 5.11　winners-take-all の拡張前と後の違い

その為，我々は出力キーワード k と source モジュール中で活性化した全てのシンボル間との関係性，つまり後向き条件付確率にその入力語の活性度，そしてキーワード k と入力語とのナレッジリンクの重みを掛け合わせたものの総和により，以下の式(5.5)のように，各 source モジュールにおける出力キーワード k の尤もらしさ C を求めることにした．なお，p_0は閾値，$input$ は入力記事，$Source_i$ は今注目している source モジュール中の i 番目に活性化した単語を意味する．

$$C(Source,k)=\sum_i\left(\frac{P(Source_i\cap k)}{P(k)}/p_0*TFIDF(Source_i,input)*weight(Source_i,k)\right) \quad (5.5)$$

したがって，タイトルモジュールと本文モジュールのそれぞれからキーワード k の尤もらしさを求める事ができるため，Confabulation theory における Cogency の式に当てはめれば，出力キーワード k のスコアは以下のように求められる．

$$\begin{aligned}\mathrm{score}(k)=&\log\left\{\sum_i\left(\frac{P(Title_i\cap k)}{P(k)}/p_0*TFIDF(Title_i,input)*weight(Title_i,k)\right)\right\}\\&+\log\left\{\sum_j\left(\frac{P(Text_j\cap k)}{P(k)}/p_0*TFIDF(Text_j,input)*weight(Text_j,k)\right)\right\}\end{aligned} \quad (5.6)$$

5.5 重み付けによる希少語の低減

前述したとおり，Confabulation theory における確率は後向き条件付確率を採用している．これは，人間が行動（反射運動等は除く）を起こす際には，行動を起こす前に既にその後に何が起こるかを予期しているという「結論→行動原理（the conclusion→action principle）」に基づいている．

しかしながら後向き条件付確率は，あまり出現回数が多くないような単語の場合に，私たちが予期した値と異なる値をとってしまう可能性がある．例えば，もし一度しか出現していないような希少語があり，偶然そのような語と共起をしてしまった場合，共起確率が過大評価されてしまい，その結果，得られる解にノイズが混じってしまうことになる．

更に，1/1 の確率値と 100/100 の確率値は両方とも同じく 1 となるが，100/100 となった現象の方が 1/1 の現象よりも重要視されるべきである．なぜならば，1 回しか出現していない単語が次に出現する信憑性は低く，100 回出現した単語の方が次の状況により出現し易いと考えられるからである．

したがって，我々はこれに対処するために，重みとして式(5.6)に k の出現回数に対数をとったものを掛け合わせることにした．

$$\mathrm{score}(k) = \left[\begin{array}{l} \log\left\{\sum_i \left(\frac{P(Title_i \cap k)}{P(k)}\right)/p_0 * TFIDF(Title_i, input) * weight(Title_i, k)\right\} + \\ \log\left\{\sum_j \left(\frac{P(Text_j \cap k)}{P(k)}\right)/p_0 * TFIDF(Text_j, input) * weight(Text_j, k)\right\} \end{array}\right] * \log(n(k)) \quad (5.7)$$

5.6 多義性と学習語彙外の入力に対する問題

Naive Bayes のような典型的な生成モデルでは大抵，多義性と学習語彙外の単語が入力された場合に問題が生じる．しかしながら，confabulation theory はこのような問題についても対処できることが利点の一つでもある．例えば，“apple”は果物や会社の名前などのいくつかの意味を表す単語である．その結果，このような多義性を持った単語が入力された時，出力が曖昧性を持ったものになってしまうことが考えられる．しかし，confabulation theory ではモジュールと呼ばれる領域に分割されており，各モジュールから導かれた解を統合して総合的な結論を導くことができるので，多義性の問題を解決することができる．つまり，同一モジュールの中で求められた解が例え曖昧であったとしても，他のモジュールからの解と統合することで曖昧性を減少させることができる．Naive Bayes ではこうした機構を持っていないので，どうしても多義語による曖昧性の問題は残ってしまう．

更に，学習語彙外の単語が入力される問題についても confabulation theory は問題なく動作することができる．Naive Bayes のようなモデルでは，このような問題に対してスムージングなどを用いることで対処しているが，このスムージングの手法により精度が左右されることもあり，本質的な解決策とは言いがたい．一方で confabulation theory は元々，今ある知識から，尤もらしい解を導く為の

認知モデルである. つまり言い換えれば, 未知の情報が入力されることも想定されたモデルとなっている. したがって, もし入力語の中にいくつかの未知の単語が含まれていたとしても, 問題なく解を導くことができる.

5.7 提案手法での比較実験

前節ではいくつかの計算手法による実装方法を提案した. そこでまず, どの実装方法が最適であるかについての検証を行う. 今まで提案してきた実装方法をまとめると, 以下のようになる.

A) 入力語の TFIDF のみ考慮
B) A + 希少語軽減の為の重みを乗算
C) B + ナレッジリンクへの重み(TFIDF の総和)
D) B + ナレッジリンクへの重み(TFIDF の平均)

我々ははじめに A と B の比較を行った. 表 5.2 と表 5.3 は 1987 年を学習し, 1988 年の記事に付与するキーワードを予測した結果である. 赤字となっているキーワードはその記事に実際に付与されていた正解のキーワードである. また, score は推論によって求められた解としてのキーワードの尤もらしさを示しており, この値が高い順にシステムはキーワードを出力する. DF は学習記事中で, キーワードが実際に付与されていた記事数を意味している.

まず, A の方に注目すると, 表 5.2 と表 5.3 ともに DF の値が非常に小さいキーワードが上位を占めていることがわかる. これは, 5.5 節で懸念していた, 希少語の生起確率が過大評価されてしまった結果であるといえる. つまり, これらのキーワードはたった 1 度の記事にしか出現していないが, その 1 回と共起してしまった入力語とのナレッジリンクの強度を示す確率値は 1/1 となり, 非常に強い結びつきであると判断されてしまう. そのため, このままでは正しい結びつきを持つキーワードが埋もれてしまう結果となってしまい, 正解となるキーワードを出力できていない.

そこで, B ではキーワード自身の出現回数に対数をかけたものを重みとして乗算することで, ノイズとなりやすい希少語を除去することを試みた. その結果, B の方の DF 値に注目すれば, 非常に小さい DF 値はなく, その結果, 出力の上位に正解となるようなキーワードを出力することができたことを示している. 我々は単純にキーワードの出現回数を重みとして乗算せずに, 対数をとることで, 重みづけとして絶妙なバランスがとれていると考えている. つまり, 単純にキーワードの出現回数をかけてしまうと, 解として出力されるキーワードは出現回数が高いものばかりとなってしまい, 本来入力語と正しい関係性をもつキーワードを出力できなくなってしまう. 例えば, 表 5.2 の B の 1 位にキーワードとして出力された「経営」の DF 値は 6614 であり, 表 5.3 の B の 3 番目の正解キーワードである「対外提携」の DF 値は 1712 である. これらの DF 値はそこまで大きくなく, ナレッジリンクの強度として学習した確率値を阻害しない程度の重み付けとなっていると考えることができる.

しかし, 表 5.2 の A で正解であった「プジョー」の DF 値は 23 と小さく, B の手法では大きな重み

付けがされない為，正解として上位に出力されなくなってしまったキーワードもある．

表 5.2　A と B の比較①

A．入力語のTFIDF考慮logなし

	Answer	Score	DF
1	権限分散	18.603	1
2	呉金属工業協進会	18.601	1
3	登別観光協会	18.579	1
4	佐々木一	18.579	1
5	上原利夫	18.579	1
6	登別商工会議所	18.579	1
7	野口秀次	18.579	1
8	中浜元三郎	18.579	1
9	上田邦雄	18.579	1
10	加藤雅	18.568	1
11	菱化海運	18.496	1
12	菱成産業	18.496	1
13	高野恒利	18.398	1
14	申立書	18.339	1
15	ザイモス	18.296	1
16	梶井功	18.281	1
17	山口巌	18.281	1
18	本庄谷礼介	18.146	1
19	利払い猶予	18.146	1
20	イラン中央銀行	18.142	1
21	延岡地域商業近代化推進協議会	18.063	1
22	自立の道探る南太平洋諸国	18.058	5
23	韓国原子力産業会議	18.058	1
24	プレイサー	18.056	1
25	キャンベル・レッド・レイク	18.056	1
26	ゴメス（アラン）	18.052	1
27	守田義雄	18.047	1
28	自立目指す韓国半導体産業	18.026	5
29	魚長食品	18.025	1
30	味の素（タイ）	17.991	1
31	城山	17.981	1
32	マドラ出版	17.965	1
33	宮道義幸	17.919	1
34	グラスゴー	17.911	1
35	インデペンデント・グローサーズ・アライアンス	17.907	1
36	工藤秀幸	17.890	1
37	科学都市	17.888	1
38	ピート・マーウィック・ミッチェル	17.876	1
39	経営者保険	17.876	1
40	ワン・ナム・フォン	17.876	1
41	マッキンゼー日本支社	17.868	1
42	斉藤千宏	17.865	1
43	アイフルホーム	17.864	1
44	京成金属	17.862	1
45	グループ経営	17.862	5
46	関口勇二	17.851	1
47	鯨岡昭雄	17.820	1
48	松和建設	17.820	1
49	石川操	17.820	1
50	昭和建設	17.820	1

B．入力語のTFIDF，log考慮

	Answer	Score	DF
1	経営	136.513	6614
2	米国	122.154	21094
3	産業界	114.303	6024
4	円高	111.700	6605
5	九州	111.286	8224
6	東京	110.645	14347
7	北海道	110.640	5759
8	開発	110.261	11573
9	経営者	108.240	950
10	会社設立	108.121	3934
11	中小企業	107.762	2407
12	銀行	107.722	4892
13	子会社	106.443	2699
14	国際化	106.318	1507
15	経済	105.972	3821
16	首都圏	105.947	8436
17	情報	105.866	4103
18	近畿	105.749	7511
19	事業強化	105.204	3300
20	経営計画	104.788	1059
21	商業界	104.715	2560
22	経営多角化	104.616	1526
23	工場	104.280	4744
24	調査	104.233	5067
25	設備投資	104.190	5439
26	株式	103.978	5738
27	利益	103.690	4849
28	中国	103.251	6025
29	新事業進出	102.547	2093
30	技術	102.523	4353
31	合理化	102.267	1480
32	サービス	102.174	3552
33	静岡	101.950	5089
34	金融	101.900	3417
35	経営方針	101.759	1149
36	業務提携	101.621	2967
37	日本企業	101.487	1199
38	大阪	100.877	4509
39	民間統計	100.474	4629
40	海外活動	100.383	1942
41	新潟	100.304	4284
42	輸出	100.238	4025
43	企業グループ	100.211	1055
44	日本	99.743	3132
45	活性化	99.580	2053
46	中部	99.272	4697
47	社長	99.260	1440
48	東北	99.060	4711
49	株価	98.878	6721
50	四国	98.844	3915
⋮			
77	貿易収支	97.652	395

表 5.3　A と B の比較②

A. 入力語のTFIDF考慮logなし

	Answer	Score	DF
1	プジョー・モーターサイクル	19.921	1
2	プジョーS・A	19.866	1
3	プジョー・グループ	19.608	1
4	ベル・テレホン・マニュファクチュアリング	19.524	1
5	カルベ（ジャック）	19.503	4
6	プジョー・シトロエン	19.318	1
7	GIE	19.318	1
8	NCシステム	19.312	1
9	次世代FAシステム	19.312	1
10	ユアサ・ゼネラル・バッテリー	19.245	1
11	マートレット・インポーティング	19.241	1
12	パク・スズキ・モーター	19.224	1
13	ベリー（フランス）	19.173	1
14	プジョー	19.133	23
15	モービル・データ・インタナショナル	19.108	1
16	ヨーロッパ・カー・オブ・ザ・イヤー	19.103	1
17	韓国プラスチック	19.086	1
18	韓国火薬グループ	19.086	1
19	韓国火薬	19.086	1
20	NEC（米国）	19.066	1
21	油圧サーボ弁	19.063	1
22	部分提携	19.044	1
23	コントロ	19.019	1
24	マソー・ディクストッフポンペン	19.019	1
25	キホーテ	19.011	1
26	アームストロング	18.968	1
27	味の素（韓国）	18.945	1
28	三星光機	18.926	1
29	神田予備校グループ	18.923	1
30	テクノ・ホルティ園芸専門学校	18.923	1
31	伊藤学園	18.923	1
32	明日を勝つシナリオ	18.919	7
33	メモリーIC	18.856	1
34	共栄工業（台湾）	18.831	1
35	聯城	18.831	1
36	連星	18.831	1
37	工企模具機械製造	18.831	1
38	ニュー・フロンティア・ジェネティクス	18.822	1
39	レイノルズ・アンド・レイノルズ	18.793	2
40	プラスチックネット	18.767	1
41	ネトロン	18.767	1
42	ホリー	18.726	1
43	ハーン（カール）	18.717	2
44	松下ビデオマニュファクチャリング	18.693	1
45	タイ・ブラナ・インダストリーズ	18.659	1
46	ボン・オーハシ	18.655	1
47	コンパニア・コロンビアナ・アウトモトリス	18.652	1
48	モルソン	18.644	5
49	PCI	18.638	1
50	エタニット	18.630	1

B. 入力語のTFIDF，log考慮

	Answer	Score	DF
1	米国	140.383	21094
2	工場	133.398	4744
3	業務提携	132.223	2967
4	円高	128.106	6605
5	現地生産	127.732	1430
6	自動車	126.624	3313
7	開発	124.878	11573
8	対外提携	124.781	1712
9	会社設立	124.585	3934
10	輸出	122.175	4025
11	中国	121.285	6025
12	設備投資	120.854	5439
13	産業界	119.771	6024
14	技術	119.403	4353
15	民間統計	119.359	4629
16	海外活動	119.173	1942
17	メーカー	119.051	2536
18	新製品	119.017	16330
19	合弁	118.809	1359
20	経営	118.545	6614
21	九州	118.490	8224
22	生産動向	118.361	1104
23	工場建設	117.963	1608
24	事業強化	117.727	3300
25	輸入	117.617	4169
26	建設	116.189	5650
27	子会社	115.606	2699
28	韓国	115.529	2745
29	北海道	114.898	5759
30	英国	114.782	3733
31	政府統計	114.555	4038
32	静岡	114.400	5089
33	技術開発	114.316	5022
34	首都圏	113.656	8436
35	通産省	113.477	2917
36	増産	113.387	993
37	日本	113.051	3132
38	生産管理	112.930	812
39	研究開発	112.629	3557
40	市場開拓	112.576	2287
41	半導体	112.548	2053
42	共同開発	112.254	2705
43	合弁会社	111.917	851
44	東北	111.419	4711
45	東京	111.260	14347
46	台湾	111.010	1138
47	西独	110.993	2288
48	新事業進出	110.780	2093
49	長野	110.755	3694
50	情報	110.755	4103

次に，ナレッジリンクに入力単語の活性度を用いることで重みとして扱う手法について言及する．

表 5.4 と表 5.5 は，上記と同じ新聞記事を入力とした時に，C と D の推論方法で出力された結果である．まず，正解のキーワードの数だけで比較を行うと，D の TFIDF 値の平均をとり，ナレッジリンクに重み付けを行う手法の方が良い結果となった．また，B と C を比較すると，C の方の正解数が悪化してしまう結果となった．この原因としては，単純な TFIDF 値の総和によって重み付けを行ってしまうと，DF 値が大きければ大きいほど単純に TFIDF 値が加算される機会が多くなり，結果として，DF 値が大きいキーワードが優先して出力されてしまうため，正解数が悪化してしまったと考えられる．確かに，表 5.3 の B における出力キーワード上位 50 件の DF 値の総和が 233,219 であったのに比べ，表 5.4 の C における出力キーワード上位 50 件の DF 値の総和は 265,745 と，DF 値の総和が大きくなっていることからも上記のような問題が発生していることが伺える．

次に，表 5.4 と表 5.5 での D に注目してみると，今回実装した手法の中で最も正解キーワード数が多い結果となった．また，出力されたキーワードの DF 値をみてみると，例えば，表 5.4 の正解キーワードの中で最も DF 値が低い「自立」の DF 値は 89，表 5.5 では「プジョー」であり，その DF 値は 23 であった．特に，「プジョー」というキーワードは B の手法では，キーワードの出現回数の対数をかけたことにより，正解キーワードとして出力されなくなってしまったキーワードであり，D の手法を用いれば，この問題も解決することができた．したがって， これらの DF 値は非常に小さいにも関わらず正解キーワードとして出力することができたことから，TFIDF の平均値を重みとして用いれば，ナレッジリンクへ適切な重み付けを行うことができるといえる．

また，表 5.4 の D の手法では，正解キーワードとはなっていないが，意味的に類似したキーワードも多く含まれている．表 5.4 の入力記事は，正解キーワードが「経営」，「自立」，「経営再建」，「経営不振」などから，会社の事業があまり上手くっていない記事であることが想像できる．他に出力されたキーワードを見てみると，「債務」「負債」「倒産」など，正解キーワードの概念に類似したキーワードであるといえる．一方，表 5.5 の入力記事は，正解キーワードが「自動車」，「プジョー」，「四輪車」，「スズキ」など，自動車産業に関する記事であると考えられる．同様に他に出力されたキーワードをみてみると，「自動車業界」，「乗用車」，「4WD」など，正解とはなっていないが，やはり記事に関連したキーワードを出力する事ができている．これらのキーワードは A,B,C のいずれの手法でも上位に出力されていないキーワードである．

キーワードアノテーションシステムのあり方としては，実際に編集者が付与したキーワードを再現できることが最も重要であるが，単に正解キーワードを当てるだけでなく，キーワード付与の候補として適切なキーワードを出力することも重要であると我々は考えている．

したがって，以上の考察から，我々は提案手法の実装方法として，最も優れていたと考えられる D の手法を採用することにした．

表 5.4　CとDの比較①

C. 入力語のTFIDF,Log考慮
＋ナレッジリンクの重み（TFIDF総和）

	Answer	Score	DF
1	経営	236.830	6614
2	米国	214.182	21094
3	東京	197.032	14347
4	開発	188.399	11573
5	産業界	187.287	6024
6	北海道	181.833	5759
7	九州	180.563	8224
8	円高	178.911	6605
9	首都圏	178.058	8436
10	会社設立	177.223	3934
11	銀行	177.188	4892
12	新製品	173.506	16330
13	近畿	171.210	7511
14	中小企業	169.556	2407
15	設備投資	169.147	5439
16	経営者	167.043	950
17	調査	165.516	5067
18	情報	164.819	4103
19	株式	164.183	5738
20	子会社	163.709	2699
21	事業強化	163.658	3300
22	工場	163.015	4744
23	利益	162.574	4849
24	中国	160.323	6025
25	静岡	160.085	5089
26	サービス	159.346	3552
27	民間統計	157.599	4629
28	新潟	156.708	4284
29	業務提携	156.698	2967
30	経営多角化	156.432	1526
31	新事業進出	156.411	2093
32	大阪	155.818	4509
33	商業界	155.667	2560
34	国際化	155.515	1507
35	中部	155.369	4697
36	経済	154.687	3821
37	建設	154.542	5650
38	東北	154.030	4711
39	経営計画	153.874	1059
40	株価	153.638	6721
41	金融	153.597	3417
42	社長	153.069	1440
43	技術	151.998	4353
44	決算	151.660	7541
45	四国	151.656	3915
46	技術開発	148.967	5022
47	店舗	148.700	2612
48	長野	147.860	3694
49	経営方針	147.237	1149
50	発売	146.377	6563

D. 入力語のTFIDF,Log考慮
＋ナレッジリンクの重み（TFIDF平均）

	Answer	Score	DF
1	経営	81.156	6614
2	自立	54.521	89
3	米国	51.672	21094
4	北海道	47.542	5759
5	銀行	45.169	4892
6	経営者	44.909	950
7	企業グループ	39.872	1055
8	企業再建	36.481	431
9	産業界	36.472	6024
10	サラリーマン	35.301	1333
11	九州	34.339	8224
12	造船	31.345	661
13	経営計画	31.209	1059
14	中小企業	30.665	2407
15	子会社	30.236	2699
16	合理化	29.251	1480
17	円高	28.910	6605
18	経済	28.165	3821
19	会社設立	26.969	3934
20	国際化	26.495	1507
21	東京	26.268	14347
22	来島どっく	25.859	67
23	経営多角化	25.353	1526
24	ブラジル	25.255	362
25	債務	25.174	513
26	倒産	24.878	764
27	負債	24.664	469
28	経営再建	24.496	152
29	トップマネジメント	24.446	439
30	取引停止	24.254	67
31	経営不振	24.078	347
32	全銀協	23.568	111
33	金融	23.545	3417
34	金融業界	23.413	1272
35	対外債務	23.196	195
36	造船業界	23.174	352
37	経営方針	22.609	1149
38	経営相談	21.913	144
39	産業構造	21.839	553
40	債務国	21.812	290
41	外国為替	21.568	2304
42	経営戦略	21.567	429
43	社長	21.266	1440
44	ドル	21.164	3340
45	開発	21.164	11573
46	四国	21.049	3915
47	銀行取引停止	20.683	44
48	資産	20.540	1170
49	構造転換	20.491	280
50	規制緩和	20.093	594

表 5.5　CとDの比較②

C. 入力語のTFIDF,Log考慮
＋ナレッジリンクの重み（TFIDF総和）

	Answer	Score	DF
1	米国	252.100	21094
2	工場	230.285	4744
3	業務提携	220.234	2967
4	円高	215.373	6605
5	開発	207.910	11573
6	現地生産	205.550	1430
7	新製品	204.837	16330
8	自動車	203.187	3313
9	会社設立	201.713	3934
10	民間統計	201.363	4629
11	設備投資	198.914	5439
12	輸出	196.736	4025
13	生産動向	195.532	1104
14	中国	195.397	6025
15	九州	195.199	8224
16	政府統計	193.704	4038
17	対外提携	193.254	1712
18	産業界	191.912	6024
19	輸入	191.282	4169
20	工場建設	188.834	1608
21	建設	188.359	5650
22	東京	187.624	14347
23	メーカー	186.735	2536
24	経営	186.323	6614
25	首都圏	185.189	8436
26	技術	184.623	4353
27	合弁	183.083	1359
28	海外活動	182.625	1942
29	英国	182.035	3733
30	北海道	181.598	5759
31	通産省	180.021	2917
32	事業強化	179.447	3300
33	子会社	178.972	2699
34	商品市況	177.850	6219
35	静岡	177.747	5089
36	技術開発	177.476	5022
37	韓国	176.891	2745
38	増産	175.972	993
39	日本	174.791	3132
40	東北	173.988	4711
41	発売	173.309	6563
42	生産管理	173.073	812
43	銀行	171.717	4892
44	半導体	171.502	2053
45	近畿	171.077	7511
46	新潟	170.250	4284
47	長野	169.521	3694
48	中部	168.691	4697
49	商品	168.643	4333
50	サービス	168.554	3552

D. 入力語のTFIDF,Log考慮
＋ナレッジリンクの重み（TFIDF平均）

	Answer	Score	DF
1	業務提携	87.361	2967
2	自動車	84.091	3313
3	工場	74.511	4744
4	現地生産	73.570	1430
5	米国	73.189	21094
6	対外提携	67.116	1712
7	フランス	65.628	1826
8	生産動向	65.098	1104
9	合弁	58.977	1359
10	民間統計	58.131	4629
11	自動車業界	55.462	888
12	円高	53.678	6605
13	輸出	53.379	4025
14	工場建設	52.891	1608
15	会社設立	52.204	3934
16	乗用車	51.089	1027
17	合弁会社	50.824	851
18	プジョー	50.158	23
19	四輪車	49.508	57
20	輸送用機器業界	49.353	951
21	自工会	49.104	93
22	生産管理	48.413	812
23	増産	48.058	993
24	生産量	47.433	482
25	海外活動	46.662	1942
26	4WD	46.480	172
27	生産計画	46.070	619
28	販売提携	45.014	543
29	オースチン・ローバー・ジャパン	44.813	28
30	輸入	44.123	4169
31	メーカー	44.040	2536
32	輸出動向	43.138	807
33	鈴木自動車工業	41.958	213
34	スズキ	41.958	213
35	開発	41.792	11573
36	政府統計	40.973	4038
37	設備投資	40.521	5439
38	自動車部品	39.932	716
39	EC	39.864	658
40	中国	38.469	6025
41	軽自動車	37.937	110
42	台湾	37.828	1138
43	GM	37.801	268
44	英国	37.696	3733
45	販売台数	37.500	411
46	カナダ	37.454	861
47	委託生産	37.337	234
48	生産拠点	37.095	336
49	西欧	36.996	1546
50	鉱工業生産	36.927	300

5.8 比較手法

Keyword annotation に関連した自動的に単語を抽出する研究はいくつか存在する. 我々が文書の索引語を付与したいとき, コーパスには大抵教師データが存在しない. そこで, 多くのアプローチでは教師なし学習が用いられる. しかしながら, term extraction の場合, 文書中に出現する語のみでしか索引語として用いることができない. これは新聞記事の検索において非常に大きなデメリットとなってしまう. 実際, 本研究で用いる日経新聞の場合は, 約 50%しかキーワードが本文中に出現していないので, term extraction はキーワード付与には適していないといえる. 更に, 本研究では我々が用いるコーパスには編集者によってキーワードが付与されている為, 教師あり学習に焦点を当てているので, term extraction とは目的も異なる.

Keyword annotation のタスクを教師あり学習による文書分類と考えれば, 多くの研究が存在する. 典型的な文書分類による教師あり学習は, Naive Bayes[40], Support Vector Machine[41], K-nearest Neighbor[42], Decision Tree[43], などが挙げられる. 特に, 識別モデルである Support Vector Machine は最も高精度な機械学習手法として知られている. Support Vector Machine で用いられるカーネル関数は様々であり, 例えば, グラフ型カーネル[44]や文字列型カーネル[45]などがある. しかし, 識別モデルは一般に, 学習に多くの時間を必要とする, 今回用いる日経新聞では約 10 万のキーワードが存在し, キーワードをクラスと考えるならば, 識別モデルによる学習は非現実的である. 本研究で提案する手法は, 単語の出現回数と単語とキーワードとの共起回数のみがわかればよい為, 高速に学習が可能となる点も提案手法の利点であるとも言える.

そこで, 本研究では比較手法として TFIDF と Naive Bayes を用いた. TFIDF は 2 章で説明したとおり, 統計的な尺度を用いて文書内の語から特徴的な語を重み付けする手法である. したがって, TFIDF 値が大きい語がキーワードの候補であるとみなすことで, 各文書のキーワード付けの手法とした. Naive Bayes は提案手法と類似した手法であるといえ, 文書内に出現した語とキーワード間の確率から, その文書が所属するクラスを決定する手法である. 以下に Naive Bayes の計算方法について示す.

Naive Bayes

ナイーブベイズは古典的な分類器ではあるが, 実装が容易である為, 現在でも使われており, 分類性能もそれなりに高いことで知られている. ナイーブベイズは確率に基づいた分類器であり, 事例 d に対して, $P(c|d)$が最大となるようなクラス$c \in C$を出力することで分類を行う. この確率$P(c|d)$を求める為に, 様々な工夫が施されている. まず, 以下の式のようなベイズの定理を用いる.

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)} \propto P(c)P(d|c) \tag{5.8}$$

ナイーブベイズではこの右辺が最大となるようなクラス c を出力する事が目的となるが，分母の$P(d)$はどのクラスにも依存しない為，考慮しないでも問題がない．したがって，分子の$P(c)P(d|c)$を最大化することが目的となる．上記式の一番右の式を求める事ができればよいのだが，一般に$P(d|c)$を求めることは簡単ではない．なぜならば，単語の種類数と文書 d との組み合わせ数を考えると，起こりうる d は膨大となり，あらゆる d について回数を調べて$P(d|c)$を最尤推定で求めることは非現実的である．そこで，ナイーブベイズでは文書 d を簡単化したモデルを仮定して$P(d|c)$を求めている．

モデルとして 2 種類のモデルが考えられ，一つは多変数ベルヌーイモデル（multivariable Bernoulli model），もう一つは多項モデル（multinomial model）と呼ばれる．一般に多項モデルの方の精度が良いといわれている[46]．そこで本稿では多項モデルのみ説明をする．

多項モデルは，文書中の各位置についてどんな単語が出現するかをモデル化することになる．いま，文書 d 内の単語数を$|d|$と表すことにする．多項モデルでは，語彙 V の中から一つの単語を選ぶ操作を$|d|$回繰り返すことで文書を生成することになる．

仮にクラスが c であった場合，単語 w が選ばれる確率を$q_{w,c}$で表すことにする．$q_{w,c}$は W を単語を値とする確率変数，C をクラスを値とする確率変数とすれば以下の式のように表すことができる．

$$q_{w,c} = P(W = w|C = c) \tag{5.9}$$

$|V|$個の単語をとりうる確率変数 W があり，各単語には$q_{w,c}$なる確率が与えられている．繰り返しになるが，多項モデルでは，この W に入る単語を決定する試行を$|d|$回繰り返すことで文書が生成されると考える．文書 d 内で，単語 w がそれぞれ$n_{w,d}$回出現する確率は，多項分布で表すことができるので，以下のようになる．

$$\frac{(\sum_w n_{w,d})!}{\prod_{w\in V} n_{w,d}!}\prod_{w\in V} q_{w,c}^{n_{w,d}} \tag{5.10}$$

文書長$|d| = \sum_w n_{w,d}$である．ただし，より厳密には，まず何回試行するかを決めなくてはならない．つまり，文書の長さをまず決めなければならないので，以下の式のようになる．

$$P(d|c) = P\left(K = \sum_w n_{w,d}\right)\frac{(\sum_w n_{w,d})!}{\prod_{w\in V} n_{w,d}!}\prod_{w\in V} q_{w,c}^{n_{w,d}} \tag{5.11}$$

ここで，K は文書の長さを表す確率変数であり，$P\left(K = \sum_w n_{w,d}\right)$は長さが$\sum_w n_{w,d}$となるような文書が起こる確率を表している．したがって，多項モデルのナイーブベイズは，

$$P(c)P(d|c) = p_c P\left(\sum_w n_{w,d}\right)\frac{(\sum_w n_{w,d})!}{\prod_{w\in V} n_{w,d}!}\prod_{w\in V} q_{w,c}^{n_{w,d}} \tag{5.12}$$

を最大化するようなクラス c を出力する. p_cはクラス c の生成確率である. よって,

$$\arg\max_c P(c)P(d|c) = \arg\max_c p_c P\left(\sum_w n_{w,d}\right)\frac{(\sum_w n_{w,d})!}{\prod_{w\in V} n_{w,d}!}\prod_{w\in V} q_{w,c}^{n_{w,d}} = \arg\max_c p_c \prod_{w\in V} q_{w,c}^{n_{w,d}} \tag{5.13}$$

となり, 結局最大化するクラス c を見つけるためには, $p_c \prod_{w\in V} q_{w,c}^{n_{w,d}}$がわかればよい.

次に, 多項モデルに対する最尤推定によりパラメータを推定する方法について述べる. 求めるべきパラメータは上記で述べたように, p_cと$q_{w.c}$の二つのパラメータである. 文書長は$|d| = \sum_w n_{w,d}$と表すことができるので, 対数を用いれば以下の式を最大化することになる.

$$\begin{aligned}\log P(D) &= \sum_{(d,c)\in D} \log P(d,c) = \sum_{(d,c)\in D}\left(\frac{P(|d|)|d|!}{\prod_{w\in V} n_{w,d}!} p_c \prod_{w\in V} q_{w,c}^{n_{w,d}}\right) \\ &= \sum_{(d,c)\in D} \log\frac{P(|d|)|d|!}{\prod_{w\in V} n_{w,d}!} + \sum_{(d,c)\in D} \log p_c + \sum_{(d,c)\in D}\sum_{w\in V} n_{w,d}\log q_{w,c} \\ &= \sum_{(d,c)\in D} \log\frac{P(|d|)|d|!}{\prod_{w\in V} n_{w,d}!} + \sum_c N_c \log p_c + \sum_c\sum_{w\in V} n_{w,c}\log q_{w,c}\end{aligned} \tag{5.14}$$

ここで, N_cはクラス c であった学習データ中の文書数を表す. 多項モデルでは, $\sum_{c\in C} p_c = 1$という制約に加えて, 任意の c について$\sum_{w\in V} q_{w,c} = 1$となる制約がある. したがって, この最大化問題は次のような制約つき最適化問題とすることができる.

$$\begin{aligned}&\max. \log P(D) \\ &\text{s.t.} \sum_{c\in C} p_c = 1, \sum_{w\in V} q_{w,c} = 1;\ \forall c \in C\end{aligned} \tag{5.15}$$

これはラグランジュの未定乗数法を用いて解くことができ, 未定乗数$\{\beta_c\}_{c\in C}$(= βとおく)とγを導入して, 次のようにラグランジュ関数$L(\theta,\beta,\gamma)$を定義する.

$$L(\theta,\beta,\gamma) = \log P(D) + \sum_{c\in C}\beta_c\left(\sum_{w\in V} q_{w,c} - 1\right) + \gamma\left(\sum_{c\in C} p_c - 1\right) \tag{5.16}$$

ただし, θ は求めたいパラメータ集合である$\left(\{q_{w,c}\}_{w\in V, c\in C}, \{p_c\}_{c\in C}\right)$を表す. 等式制約付きの凸計

画問題に対するラグランジュの未定乗数法によれば，$q_{w,c}$に関する偏微分が 0 になれば良い．したがって，これらの偏微分を計算すると，

$$\frac{\partial L(\theta,\beta,\gamma)}{\partial q_{w,c}} = \frac{n_{w,c}}{q_{w,c}} + \beta_c, \frac{\partial L(\theta,\beta,\gamma)}{\partial p_c} = \frac{N_c}{p_c} + \gamma \tag{5.17}$$

となる．これらを 0 とし，$\sum_{c\in C} p_c = 1, \sum_{w\in V} q_{w,c} = 1$とあわせると，

$$q_{w,c} = \frac{n_{w,c}}{\sum_w n_{w,c}}, p_c = \frac{N_c}{\sum_c N_c} \tag{5.18}$$

が最終的に得られる．$q_{w,c}$を言葉で表現するならば，以下のようになる．

$$q_{w,c} = \frac{(\textit{クラス}\boldsymbol{c}\textit{に属する学習文書全体での}\boldsymbol{w}\textit{の出現回数})}{(\textit{クラス}\boldsymbol{c}\textit{に属する学習文書全体での全単語の出現回数})} \tag{5.19}$$

しかし，このままでは未知の文書に対するクラスを予測する際に，学習データ中の語彙に含まれない単語が1つでも含んでしまうと$q_{w,c}$は 0 となってしまい，結果として，$P(d|c)$も 0 となってしまう．つまり，その新しい文書が生成される確率は 0 ということになり，これは問題である．

このような問題はゼロ頻度問題と言われ，この問題を解決する為には一般的にスムージングを用いる．ここでは，単語の出現回数に 1 を加えるという最も基本的なラプラススムージング（Laplace Smoothing）を用いることにする．したがって，$q_{w,c}$は，

$$q_{w,c} = \frac{n_{w,c}+1}{\sum_w (n_{w,c}+1)} = \frac{n_{w,c}+1}{\sum_w n_{w,c} + |V|} \tag{5.20}$$

とすることで，ゼロ頻度問題を緩和することができる．

5.9 既存手法との比較実験

提案手法の精度を検証するために，我々は比較手法として既存手法との精度の比較を定量・定性的評価とシステムの実行速度の比較を行った．

5.9.1 定量的評価

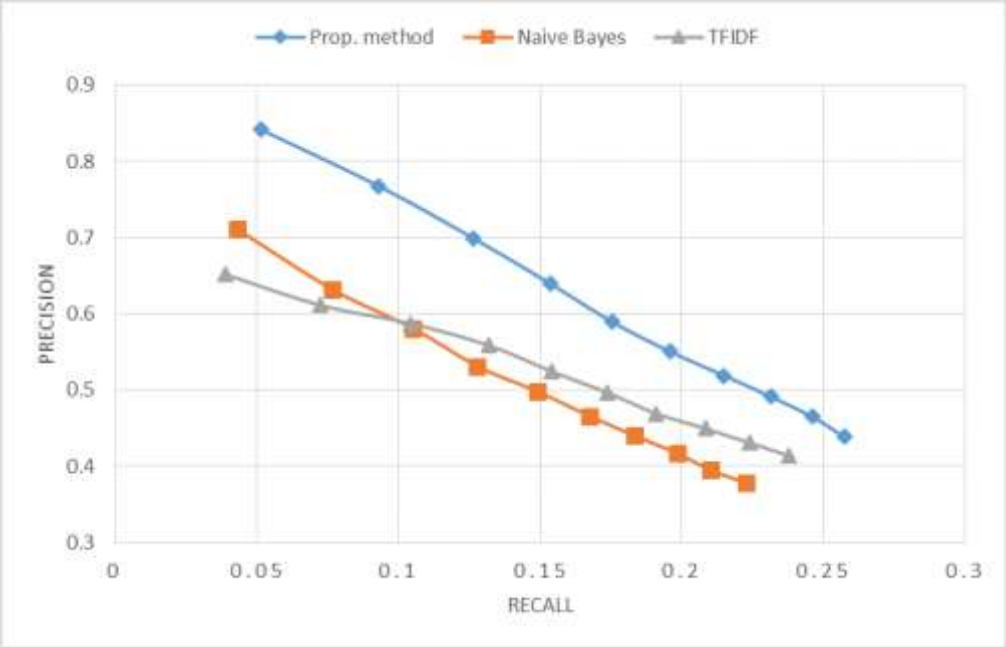

図 5.12　3 手法の Recall-Precision 曲線

今回，定量評価には，情報検索において一般的な評価尺度である，PrecisionとRecall，そしてF-measure を用いた．図 5.12 は Recall-Precision 曲線を表しており，縦軸は Precision，横軸は Recall を表している．一般に，PrecisionとRecall はトレードオフの関係にあり，大抵右下がりの曲線となる．また，曲線が上にある手法の方が良い精度であるといえる．

また，表 5.6 は各評価手法での実験結果を意味している．P@N は上位 N 件中の正解率を表す．これらの実験結果から，提案手法は比較手法よりも優れていたことがわかる．

表 5.6 各評価手法での実験結果

	P@5	P@10	F-measure
Prop. Method	**0.589**	**0.438**	**0.324**
Naive Bayes	0.498	0.378	0.223
TFIDF	0.525	0.414	0.302

まず，提案手法とナイーブベイズの比較を行う．これらの手法は，ともにキーワードと単語間の関係性として確率値を用いているという点で似ている手法といえるが，図 5.12 や表 5.6 から実験の評価値には大きな差が生じた．

この 2 つの手法には主に 2 つの大きな違いがある．一つ目の違いとしては，記事中のタイトル部分と本文部分の扱い方である．提案手法では，Confabulation theory に則り，タイトルと本文をそれぞれ独立なモジュールであると定義した為，記事に付与されたキーワードとタイトル・本文それぞれからの入力語の関係性を独立に学習している．一方で，ナイーブベイズはタイトルと本文は分割せずに同等な扱いをしている．二つ目の違いは，用いられる条件付確率である．提案手法では，後向き条件付確率を用いているが，ナイーブベイズでは前向き条件付確率が用いられる．

そこで我々は，どの要因が最も精度に大きな影響を与えているかを調べるために，両システムともに本文だけを用いて実験を行った．図 5.13 はその結果である．

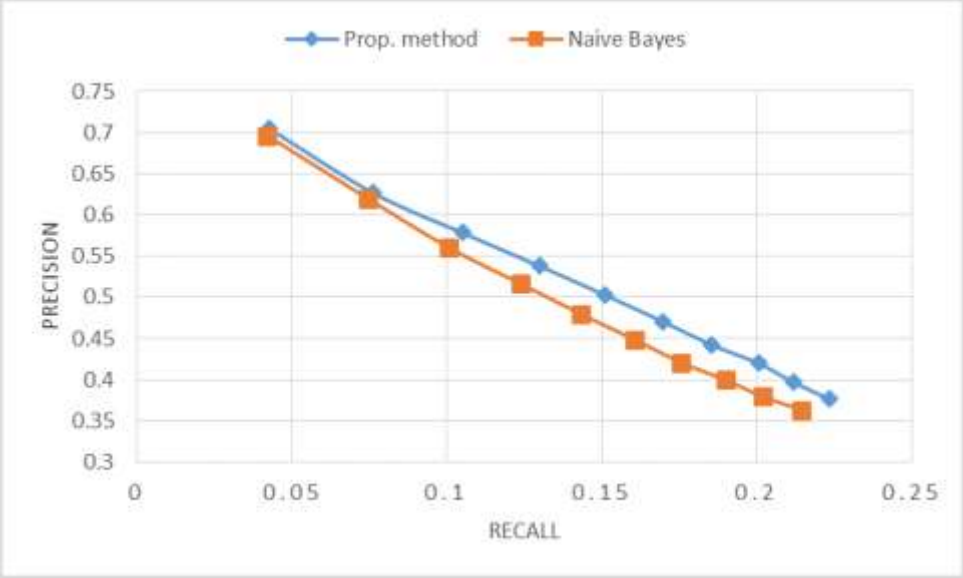

図 5.13 本文のみを用いた時の提案手法とナイーブベイズの Recall-Precision 曲線

この図から，両システムとも精度に大きな違いが現れていない事がわかる．更に，ナイーブベイズについては，もしタイトルを使わなかったとしても，そこまで精度に大きな違いが生まれていないことがわかった．つまり，確率値の違いはそこまでシステムの精度に影響を与えず，モジュールの存在がシステムの精度に大きな影響を与えているという事が実験結果からわかった．

したがって，タイトルと本文を独立なモジュールとして扱うことで，タイトルと本文の情報をより効果的に扱うことが出来たため，提案手法の方がナイーブベイズよりも優れた性能を発揮できたと考えられる．この実験結果は，Confabulation theory において非常に重視されているモジュールの重要性を裏付ける結果であるという事ができ，更には 5.2 節において我々が予想した「source モジュールと answer モジュールの間には明確な関係性」が必要であり，明確な関係性を保証する事ができれば優れた性能を引き出すことができることを示している．つまり，Confabulation theory に基づいてシステムを構築する際に，システムの精度を最大限に発揮させるためには，「モジュールの定義が非常に重要である」という知見を得ることができた．

次に提案手法と TFIDF を比較することにする．図 5.12 をみると，Recall-Precision 曲線の右にいくほど，提案手法とTFIDF の精度差は徐々に小さくなっていることがわかる．更に，表 5.6 をみると，P@10 における提案手法と TFIDF の精度差は，P@5 に比べて同様に小さくなっていることがわかる．つまり，キーワードの出力数が増えるほど，提案手法と TFIDF の精度差は小さくなるということである．以上から，出力キーワード数が小さい時には，提案手法は特に高い精度を実現する事ができるといえる．Confabulation theory では winners-take-all メカニズムにより，1 つの入力に対して，answer モジュール中のシンボルが活性化するのは本来ただ一つである．このことを踏まえれば，モデルの性格上，提案手法は少ない出力数に対しては高い精度を実現できるが，出力数が増えるほど認識が曖昧となってしまい精度が下がってしまうので，大量のキーワード出力には不向きであると考えられる．

一方で，もし入力記事中に出現した単語を全てキーワードの候補としたとしても，本文中に正しいキーワードが含まれている割合は約 50%であるため，TFIDF は最高でも 50%程度の性能しか発揮することができない．TFIDF はもともと，記事の中に出現した単語の中で特徴的なものに大きなスコアを与えるものである．直感的に，より特徴的な語がよりキーワードになりやすいと考えられる．したがって，大量のキーワードを出力する程，TFIDF によるキーワードの出力が正解する傾向にあると考えられる．

5.9.2 定性的評価

5.7 節で示したように，正解ではなかったキーワードでも出力キーワードの中には意味的に等しいものがあった．しかし，上記定量評価では正解として評価を行う事ができないため，そのような意味的に等しいキーワードを正しく評価することができないが，そのようなキーワードも正解キーワードとして評価されるべきである．そこで，上記問題を解決するため，我々はアンケートによって人間に評価をしてもらうことにした．

具体的には，12 人の被験者（大学生・大学院生）に協力をしてもらい，被験者に新聞の記事とシステムが出力した上位 N 件を提示し，正解だと思ったキーワードには○，不正解だと思ったキーワードには×をつけてもらった．そして，アンケートの結果を評価する為に，以下の式により，被験者の好ましさを示す尺度である Favorability を定義した．なお，○だった場合は 1 点，×だった場合は 0 点とした．また，w_pは p 番目に出力されたキーワードを意味し，今回 N は 20 とした．

$$\text{Favorability} = \frac{\sum_p w_p}{N} \tag{5.21}$$

更に P@N の値の違いが被験者の評価にどう影響を与えるか検証したかった為，アンケートに用いた記事は以下に示すような 3 つの記事グループに分類し，各グループから 2 つずつランダムに選択した．また，比較手法にはナイーブベイズを用いた．

A) 提案手法，比較手法ともに P@20 が 0.5 以上であり，かつその差が 0.1 未満である記事
B) 提案手法，比較手法ともに p@20 が 0.1 以上 0.5 未満であり，かつその差が 0.1 未満である記事
C) 提案手法，比較手法ともに P@20 が 0.1 未満である記事

表 5.7 はアンケートの結果を示している．T 検定の結果，有意水準 1%で有意に差があった．これは，提案手法によるキーワード出力の方が比較手法よりも意味的に正解のキーワードを出力できており，人間にとってより妥当であったことを示している．

表 5.7 アンケートの結果

	Prop.method	NB
Favorability	**0.4667**	0.4201

表 5.8 各記事グループでの評価結果

	A		B		C	
	Prop.method	NB	Prop.method	NB	Prop.method	NB
Favorability	0.5271	0.5063	0.4813	0.4438	0.3917	0.3104

表 5.8 は，各記事セット A，B，C の各手法における Favorability のスコアを示している．まず，提案手法の全ての Favorability は比較手法よりも優れたスコアを示している．記事セット A と B での Favorability では，提案手法と比較手法の間に大きなスコアの違いを示す事ができなかった．これについては，定量評価と人間の感覚との間にはさほどズレが生じていなかったとも言える．

一方で，記事セット C にてスコアの差を示す事ができた．記事セット C とは P@20 が両手法とも 0.1 未満であった記事セットであり，定量的には悪かったケースであるといえる．これは前述した予想通り，実際の記事には付与されなかったキーワードであったため，定量的には評価されなかったが，意味的には正しいキーワードを含んでいたことを示している．更に，記事セット C での提案手法と比較手法での Favorability のスコア差が最も大きかったことから，提案手法によって出力されるキーワードは，文書の意味をより正確に捉えられており，意味的に正しいキーワードを出力することができるといえる．

以上より，提案手法は比較手法よりも人間による評価が良かったことから，提案手法は人間にとってより妥当なキーワードを出力する事ができると考えられる．

5.9.3 実行速度評価

表 5.9 推論速度の比較

	Time
Prop. Method	1
Naive Bayes	18.95

我々は，提案手法と比較手法であるナイーブベイズの推論時間の計測も行った．どんなに優れた精度をシステムが導けるとしても，特に新聞のような毎日発行される規模の大きな文書データに対しては，システムの実行速度は非常に重要であると考えられるからである．

推論速度の計測は，10 記事を用意し，各記事でのキーワード付与にかかる時間を計測し，その平均時間を求めることによって行った．

表 5.9 は，両手法でのシステムの推論速度の結果を示したものであり，提案手法の推論速度を 1 と見なした時の，推論速度の割合を示している．提案手法は，比較手法に比べて約 19 倍の速度で実行可能である事がわかった．Confabulation theory ではもともと認知の計算が並列処理で行われることを前提としているため，提案手法でも特に工夫をしなくても並列処理を行う事ができる．具体的には，タイトルと本文を独立なモジュールとして扱い，それぞれのモジュールで解を求めさせることで並列処理化させることができる．加えて，提案手法では，入力語とナレッジリンクを持っているキーワードのみに焦点を当てればよいので，単純な計算量も少なくなる．

一方で，ナイーブベイズは提案手法とは異なり，タイトルと本文を別々には扱わない．また，シンプルなナイーブベイズのアルゴリズムでは，過去に出現した全てのキーワードのスコアを計算する必要がある為，ナイーブベイズで解を求めるためには膨大な計算量が必要となってしまう．

以上の理由から，提案手法はナイーブベイズよりも高速に処理が可能であると考えられる．

5.10 まとめ

本章では Confabulation theory を自然言語処理に応用し，現在人手による作業が必要なキーワード付与作業を完全に自動化し，大規模なデータに対して高精度かつ高速なキーワード付与するシステムを提案した．

まず，Confabulation theory に基づくシステムが最大限に性能を発揮する為にはモジュールの定義が重要であることを従来研究から分析し，テキストデータに適切なモジュールの定義を示した．次に，Confabulation theory の winners-take-all メカニズムに基づけば，文書の内容はただ 1 つの単語で表す必要があるが，文書の内容を 1 単語で表すことは困難であることから，複数単語での学習が可能となるように winners-take-all メカニズムを拡張した学習方法を提案し，精度の向上につながることを明らかにした．また，Confabulation theory では後向き条件付確率を採用しているが，出現回数が低いキーワードではその確率値が過大評価され，意図しない出力が得られてしまう事があることから，キーワードの出現回数に対数をとったものを重みとすることで，この問題が解決できることを示した．

実験は，14 年分の日経新聞から（1987-1988），…，（1999-2000）のペア 13 セットで実施し，約 10 万のキーワード候補から記事に適切なキーワードを自動で付与する実験を行った．評価は，定量評価，定性評価，実行速度評価の 3 つ観点から行った．その結果，定量評価においては比較手法よりも約 10%の精度向上，定性評価でも比較手法よりも優れ，実行速度も約 19 倍高速に処理することが可能であることを示した．

第 6 章　結論

本研究では，人間の脳の認知モデルである Confabulation theory を用いて，文書にキーワードを自動的に付与するシステムを提案した．文書にキーワードを付与することは，計算機上で文書を取り扱うための基本的な方法論の一つとして用いられている．一般的な方法論では，文書内に出現した単語のみで文書を表現することが多いが，新聞の記事などの情報検索や文書要約の分野では文書外の単語も用いて文書を表現することが有益となることが多い．また，統計量などを用いた手法によって付与されたキーワードは人間によって付与されたものと比べ，精密さにかける．本研究では，人間によってキーワードが付与された大規模なコーパスを学習させることで，文書内に存在しない単語もキーワードとして付与し，人間に近いキーワード付与が可能なシステムを実現した．

実験では新聞記事をコーパスとして用い，人間に近い動作を計算機に行わせるために Confabulation theory を利用した．Confabulation theory では機能を意味するモジュールと，その機能の属性を表すシンボルの定義が重要であるが，我々は新聞記事のタイトルと本文とキーワードの 3 つの領域をモジュールとして定義し，シンボルは各領域内で出現した単語として定義した．

また，Confabulation theory では各モジュール中に活性化するシンボルは唯一つであるという winners-take-all メカニズムがある．しかし，各モジュール中のシンボルを単語としたことから，唯一つのシンボルを活性化させることは困難であった．なぜならば，ただ一つの単語だけで各領域の内容を表現しきるほどの表現力が今回定義したシンボルにはなかったからである．そこで，我々は複数シンボルの活性化を許容することで，この winners-take-all メカニズムを拡張することにした．具体的には，学習時に，各記事の本文中に出現した単語とキーワード，タイトル中に出現した単語とキーワード間には関係性があるとみなして，総当りで共起数を学習して，単語とキーワード間の関係性を示す後向き条件付確率を学習することにした．

この方法論には二つの問題が生じることがわかった．一つ目は，全ての単語とキーワード間を学習することで，たまたま関係性を持ってしまったものが過大評価されてしまうことである．つまり，ある入力単語と 1 回しか出現したことがないような希少キーワードが共起した場合，その確率は 1 となり，非常に大きな関連性をもつことになってしまう．そこで，キーワードの出現回数に対数をとったものを重みとすることで，この問題を解消することにした．もう一つの問題は，総当りで関係性を学習してしまったことで，入力単語とキーワード間の正確な関係性が学習できない点である．本来の winners-take-all メカニズムに従えば，最も関連性があったシンボル間のみが活性化することになるが，総当りで学習してしまうことで，シンボル間の活性化に差異が生まれなくなってしまう．そこで，自然な winners-take-all メカニズムの拡張とするために，各シンボルが活性化したときの活性度として，文書中での単語の重要度を表す TFIDF 値を用いることで，シンボルの活性度に優劣を設け，より自然な winners-take-all メカニズムの拡張を実現した．

以上の工夫により，既存手法よりも約 10%の精度向上を実現させることに成功した．これは，既存手法よりも，より人間に近いキーワード付与を実現できたことを意味する．また，アンケートによる定性的評価においても既存手法よりも優れていたことを示した．更に，人間の脳は並列処理的に動作していることが脳科学の知見から得られており，Confabulation theory ではそのような機構がモデルの中に組み込まれていることから，計算機上においても自然に並列処理で実装することが可能であった．その結果，既存手法よりも約 19 倍高速に動作することも示すことができた．

今後は，より緻密なモジュールとシンボルの定義により，更なる精度向上を目指していく事が課題であると考えられる．特に，計算機による人間の言葉の意味理解は今後必須の課題である．本研究では，「文書の意味・内容を単語・キーワードで置き換える」というアプローチをとることで，計算機による言葉の意味理解の実現を目指した．更なる言語処理による高度な知的情報処理の為に，何を入力として与え，何を出力とすれば計算機が言葉の意味を理解できたかどうかを定義し，人間の言語処理機構を参考にしながら，計算機に適した方法論を模索し続けることは，今後も積極的に取り組むべきであると思われる．

参考文献

[1] T. Mikolov et al. “Efficient Estimation of Words Representation in Vector Space,” International Conference on Learning Representations, 2013.

[2] C. Cleverdon, “Optimizing convenient online access to bibliographic databases,” Information Service and Use, 4, pp. 37-47, 1984.

[3] K. Sparck Jones, “A statistical interpretation of term specificity and its application in retrieval,” Journal of Documentation, vol. 28, No. 1, pp. 11-21, 1972.

[4] M. Hamaguchi, “YOMIDAS REKISHIKAN, new database service of japanese newspaper and YOMIDAS YOGO JISHO, thesaurus by The Yomiuri Shimbun,” Journal of Information Processing and Management, vol. 52, No. 3, June 2009.

[5] M. Ishii, “Auto-indexing system in Nihon Keizai Shinbun, Inc.,” Information Science and Technology Association vol. 42, No. 11, pp. 1058-1064, November 1992.

[6] J. Cowie et al., “Information extraction,” Communications of the ACM, vol. 39, No. 1, pp. 80-91, 1996.

[7] S. Sekine, “Named Entity to Terminology,” the 10th NLP Workshop on “Named Entities and Terminology”, 2004.

[8] Z. Kozareva, “Bootstrapping Named Entity Recognition with Automatically Generated Gazetter Lists,” Proc. of the EACL 2006 Student Research Workshop, pp. 15-21, 2006.

[9] D. Yarowsky, “Unsupervised learning of generalized names,” Proc. of the 19th International Conf. Computational Linguistics, 2002.

[10] K. Kageura, “Methods of automatic term recognition,” Terminology, vol. 3, No. 2, pp. 259-289, 1996.

[11] Q. Zadeh et al., “Semi-Supervised Technical Term Tagging With Minimal User Feedback,” LREC, pp. 617-621, 2012.

[12] H. Nakagawa et al., “Automatic Term Recognition based on Statistics of Compound Nouns and their Components,” Terminology, Vol.9, No.2, pp. 201-219, 2001.

[13] M. Utiyama et al., “Using Author Keywords for Automatic Term Recognition,” Terminology, Vol.6, No.2, pp. 313-326, 2000.

[14] C. D. Paice, “Constructing Literature Abstracts by Computer,” Techniques and Prospects. Information Processing and Management vol. 26, No. 1, pp. 171-186.

[15] H. P. Luhn, “The automatic creation of literature abstracts,” IBM Journal, pp. 159-165, 1958.

[16] N. J. Belkin et al., “Information filtering and information retrieval: Two sides of the same coin?,” Communications of the ACM, vol. 35, No. 12, pp. 29-38, 1992.

[17] F. Rosenblatt, “The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain,” Psychological Review, vol. 65, No. 6, pp. 386-408, 1958.

[18] M. L. Minskey et al., “Perceptrons: An Introduction to Computational Geometry,” The MIT Press, Cambridge, MA, 1969.

[19] D. E. Rumelhart et al., “Learning internal representations by back-propagating errors,” Nature, vol. 323, pp. 533-536, 1986.

[20] A. Krizhevsky et al., “ImageNet Classification with Deep Convolutional Neural Networks,” Proc. Neural Information and Processing Systems, 2012.

[21] http://www.alchemyapi.com [Accessed January 13, 2015].

[22] Q. V. Le et al., “Building High-level Features Using Large Scale Unsupervised Learning,” Proc. International Conf. Machine Learning, 2012.

[23] D. Bollegala, “Deep Learning for Natural Language Processing,” Journal of the Japan Society for Artificial Intelligence, vol. 29, No. 2, pp. 195-201, 2014.

[24] R. Socher et al., “Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection,” Proc. Neural Information and Processing Systems, 2011.

[25] Y. Ichisugi, “The cerebral cortex model that self-organaizes conditional probability tables and executes belief propagation,” Proc. International Joint Conf. on Neural Networks, pp. 1065-1070, 2007.

[26] Y. Ichisugi, “The Cerebral Cortex and Bayesian Networks,” Journal of RSJ, vol. 29, No. 5, pp. 412-415, 2011.

[27] D. George et al, “A hierarchical Bayesian model of invariant pattern recognition in the visual cortex,” Proc. International Joint Conf. on Neural Networks, vol. 3, pp. 1812-1817, 2005.

[28] R. P. N. Rao, “Neural models of Bayesian belief propagation,” The MIT Press, Cambridge, MA, 2007.

[29] J. Pearl, “Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference,” Morgan Kaufmann, 1988.

[30] T. Kohonen, “Self-Organizing Maps,”Springer-Verlag, 1995.

[31] R. Hecht-Nielsen, “Confabulation Theory,” Springer-Verlag: Heidelberg, 2007.

[32] R. Hecht-Nielsen, “Cogent confabulation,” Neural Networks, Vol.18, pp. 111-115, 2005.

[33] R. Hecht-Nielsen, “Confabulation theory,” UCSD Institute for Neural Computation Technical Report #0501, 2005.

[34] D. O. Hebb, “The Organization of Behavior,” Wiley, New York, 1949.

[35] C. E. Shannon, “Prediction and Entropy of Printed English,” Bell Syst. Tech. J., vol. 30, pp. 50-64, 1951.

[36] http://www.nikkei.com [Accessed January 13, 2015]

[37] T. Takada et al., “Predicting Future Events using Time Series Linguistic Data,” World Conf. on

Soft Computing, 2011.

[38] T. Takada et al., “Reader Centric Real-time Electric Magazine Article Generator,” IEEE International Conf. on Systems Man and Cybernetics, 2011.

[39] S. Mori et al., “Survey of Relationship between Article’s Body and Title of Newspaper,” IEICE Technical Report, pp. 13-16, 2011.

[40] M. E. Maron, “Automatic Indexing: An Experimental Inquiry,” J. ACM, Vol.8, No.3, pp. 404-417, 1961.

[41] T. Joachims, “Text Categorization with Support Vector Machines: Learning with Many Relevant Features,” Proc. European Conf. on Machine Learning, pp. 137-142, 1998.

[42] B. V. Dasarathy, “Nearrest Neighbor (NN) Norms: NN Pattern Classification Techniques,” IEEE Press, 1991.

[43] C. Apte et al., “Automated Learning of Decision Rules for Text Categorization,” ACM Trans. Information Systems, Vol.12, No.3, pp. 233-251, 1994.

[44] S. Bleik et al., “Text Categorization of Biomedical Data Sets Using Graph Kernels and a Controlled Vocabulary,” IEEE/ACM Trans. on Computational Biology and Bioinformatics, 2013.

[45] D. Zhang andW. S. Lee, “Extracting Key-Substring-Group Features for Text Classification,” Proc. of the 12th ACM SIGKDD International Conf. on Knowledge Discovery and Data Mining, pp. 474-483, 2006.

[46] A. McCallum et al., “A Comparison of Event Models for Naive Bayes Text Classification,” Proc. of the AAAI-98 Workshop on Learning for Text Categorization, pp. 41-48, 1998.

謝辞

本研究に進めるにあたり，非常に多くの方からご指導・ご鞭撻を賜わりました．
大変御多忙の身でありながら熱心にご指導して下さいました，高木友博教授に深く感謝いたします．高木先生には，学部 3 回生から，卒業研究，博士前期・後期課程と 7 年間半もの長きにわたりお世話になりました．時には励ましの，時には叱咤のお言葉を頂戴したことは今の私の研究姿勢の根幹となっております．私は研究者としても人間としてもまだまだ未熟ものですが，今後とも引続きご指導・ご鞭撻の程，宜しくお願い致します．

また，常に変わらぬ温かいご指導をいただいた明治大学理工学部情報科学科の先生方に篤く感謝の言葉を捧げます．お忙しい中，副査を引き受けてくださった林先生，武野先生には特に御礼を申し上げます．

そして日頃の苦楽を共にしてきたウェブサイエンス研究室の多くの皆様に感謝いたします．後期課程がんばれよと励ましの言葉をかけて卒業していった伊藤慎一郎君，高山勇樹君，坂口貴俊君，本橋直樹君，赤穂有哉君，中津恭兵君（以上，博士前期課程同期の皆様），良き友として今でも応援してくれる伊澤諒君，精神的につらかった時期でも笑いながら語り合った島村和明君，山口幸治君，岩城基史君，笠間大祐君，初めて一緒に共同研究した EMG 班の赤塚慎也君，平優里さん，新井瑞希さん，同じ班として研究活動を一緒に頑張ってきた LCP・CL 班の皆さん，そして最終学年時に長時間の議論にいつも付き合ってくれた簇野光輝君，宮城涼君，その他にも研究生活を共に過ごしてきた全ての皆様に感謝いたします．

最後に，長い学生生活を辛抱強く支えて頂いた父母，兄姉に深く感謝いたします．