修 士 論 文

論文間の参照情報を考慮した学術論文要約システムの開発

指導教官　奥村学 助教授

北陸先端科学技術大学院大学
情報科学研究科情報処理学専攻

難波 英嗣

1998 年 2 月 13 日

要旨

本研究では、ひとつの研究分野に関する複数の学術論文の差異に注目し、論文間の参照情報を考慮して複数の関連する論文との違いを明確にする要約を自動的に作成することを試みる。本研究では、論文間の関係を解析する際、論文の参照情報に着目する。ある論文が他の論文を参照する場合、参照論文について記述してある箇所 (参照箇所) が存在する。その箇所を読むことで、著者がどのような目的で参照しているのか明らかになる。このようにして参照箇所から得られる情報を参照情報と呼ぶ。参照情報を収集し整理することで、ある分野の複数の論文間の関係が明らかになり、またそれらの参照情報が要約生成に利用できると考えられる。

本研究で開発する要約生成システムの枠組を説明する。ある特定の分野の複数の論文を要約対象とする。まず論文データベース中から入力論文 (target papers) を参照している論文を検索する。次に検索された論文から入力論文について記述している箇所 (参照箇所) を自動抽出する。参照箇所から、著者がどのような目的で他の論文を参照しているのか (参照タイプ) を自動的に判別し、参照タイプ毎に参照箇所を分類する。これらを要約の原型として処理を施した後、最終的な要約文書を出力する。

本研究では上記の枠組において、まずはひとつの入力論文に関する要約を生成システムを実装し、参照情報を利用することがある分野の要約を生成するのに有効であることを示した。このシステムをさらに改良することで、ある分野の研究の動向が明確な要約が生成可能になると考えられる。

目 次

図 目 次

表 目 次

第 1 章

はじめに

1.1 研究の背景と目的

1.1.1 複数テキスト要約生成の重要性

近年、インターネットの整備とともに、オンラインで数多くの電子化されたテキストを入手できるようになった。電子化された多くの情報の中から求める情報を探し出す際、様々な検索方法がある。例えば WWW 上の場合、Altavista[1]、goo[2]といった検索エンジンを用いてキーワード検索をすることが可能である。また Yahoo![3]や NTT DIRECTORY[4]、CSJ インデックス[5]のようにあらかじめ分類された Web page を、ディレクトリをたどることで目的の Web page を検索する方法もある。このように、目的の情報にアプローチするための手段が増え、益々便利になる一方でいくつかの問題点も浮き彫りになってきている。問題点は大きく以下の 2 つが考えられる。

(1) 検索結果が何万件にも及ぶような場合、その中から目的の情報を捜し出すのが困難である。

(2) WWW のように情報が複数の情報源から発信される場合、重複した内容のテキストが検索される可能性がある [船坂 96]。

[1] http://altavista.digital.com/

[2] http://www.goo.ne.jp/

[3] http://www.yahoo.co.jp/

[4] http://navi.ntt.jp/

[5] http://www.csj.co.jp/csjindex/

(1) のひとつの対処方法として、検索するためのキーワードを拡張することで、目的の情報に絞り込んでいく方法がある。別の対処方法として要約技術を用いる方法がある。[佐藤 96] はネットニュースのダイジェスティングにより情報を効率的に提示可能にしている。同じネットニュースを扱った研究に [McKeown96] がある。ネットニュースの場合は、ひとつのニュースグループ内では、記事が重複することはほとんどないと考えられるが、情報源が複数存在する場合、(2) に示した重複した内容のテキストが存在することを考慮して要約を生成しなければならない。このような場合、複数テキストからひとつの要約を生成する技術が必要とされる。

1.1.2 学術論文調査における参照関係の利用

学術論文調査において、学術論文データベースを利用するという方法がある。例えば科学技術振興事業団 (JST)[6]や学術情報センター (NACSIS)[7]等から、様々なデータベースが利用可能である。また、日外アソシエーツの雑誌記事索引ファイルで、幅広い分野の国内の学術論文を調査することができる。論文検索において、よく使用される属性は、著者名、標題、情報内容を表現する語句 ·記号などであり、索引語、索引キー、アクセス · キーなどと呼ばれている。近年では、参照文献も重要な索引語として使用されるようになってきている。

参照文献は、通常、学術論文の中になんらかの形でリストが記される。これは、研究過程においてさまざまな形で利用された文献を明示するためのものである。こうした参照文献群は、その文献が、それ以前のどのような研究をふまえたものかについての情報を与えるばかりでなく、学術情報利用者のための重要な情報源となる。参照文献は、また、*Citation Index* の出現によって、二次情報の流通のうえでも、1 つの役割を果たすようになっている。

ある分野の研究の動向調査をする場合、調査の手順としてまず、対象となる分野の研究論文を収集し、次にその分野の論文の中で互いに類似するものをグループ化することが必要となる。類似する文献を分類する方法は、付与された索引語の共出現、あるいは標題や抄録中に出現する語によるクラスタ化等、様々な方法がある。他に共引用を利用するという手法がある。共引用とは、2 つの文献が同一の文献に引用されている状態を示す。一般

[6]http://www.jst.go.jp/

[7]http://www.nacsis.ac.jp/nacsis.index.html

に単純な参照関係が文献間の類似性を表現するとは言い切れないが、類似性を示す尺度として妥当性が高いものと考えられる。[神門 91] では、共引用分析を用いて情報検索の分野の論文について調査を行っている。[神門 91] では、以下に示す式 (1.1) を用いて論文間の類似度を算出し、これを元に共引用マップを作成 ·分析することで情報検索分野の研究調査をしている。本研究では、この論文間の参照関係というものに着目する。

$$\text{文献 } A \text{ と } B\text{の類似度} = \frac{A \text{ と } B\text{が共引用された回数}}{\sqrt{A \text{ の被引用回数} \times B\text{の被引用回数}}} \tag{1.1}$$

動向調査の際、もし特定分野のサーベイが存在するならば、調査を非常に効率的に行うことができる。一方で、そういったサーベイは必ずしも存在するわけではない。そこで、ある分野の複数の論文からひとつの要約を自動的に生成することを試みる。通常、ある論文が他の論文を参照する時、論文中で参照論文について述べている箇所がある。本研究ではその箇所を参照箇所と呼ぶ。参照箇所に着目し、解析することで、論文間の関係が明らかになりそれらをまとめることにより、特定分野の要約生成が可能になると考えられる。

1.1.3 本研究の目的

本研究では、ひとつの研究分野に関する複数の学術論文の差異に注目し、論文間の参照情報を考慮して複数の関連する論文との違いを明確にする要約を自動的に作成することを試みる。論文間の関係を解析する際、論文の参照情報に着目する。ある論文が他の論文を参照する場合、参照箇所を読むことで著者がどのような目的で参照しているのか明らかになる。このようにして参照箇所から得られる情報を参照情報と呼ぶ。参照情報を収集し整理することで、ある分野の複数の論文間の関係が明らかになり、またそれらの参照情報が要約生成に利用できると考えられる。

本研究で開発する要約生成システムの枠組を説明する。ある特定の分野の複数の論文を要約対象とする。まず論文データベース中から入力論文 (target papers) を参照している論文を検索する。次に検索された論文から参照箇所を自動抽出する。参照箇所から、著者がどのような目的で他の論文を参照しているのか (参照タイプ) を自動的に判別し、参照タイプ毎に参照箇所を分類する。これらを要約の原型として処理を施した後、最終的な要約文書を出力する。

本研究では上記の枠組において、まずはひとつの入力論文に関する要約を生成システムを実装し、参照情報を利用することがある分野の要約を生成するのに有効であることを示した。このシステムをさらに改良することで、ある分野の研究の動向が明確な要約が生成可能になると考えられる。

1.2　本論文の構成

本論文では、2 章で複数テキストの要約の関連研究を紹介し、本研究との比較を行う。3 章では、複数の関連論文をまとめてひとつの要約を生成する際に用いる　参照情報」というものの定義とその有効性について説明する。4 章では、3 章で定義した参照情報を用いて複数の論文から要約を生成する方法について述べる。5 章では、4 章で述べた手法の評価実験と実験結果を示す。6 章では 5 章の実験結果について考察する。7 章で本研究のまとめと今後の課題について述べる。また、付録 A として今回作成した要約システムが生成した要約例を載せた。

第 2 章

複数テキストの要約

これまでに、単一テキストの要約生成に関するする様々な手法が提案されてきた。[Kupiec95, Teufel97, Watanabe96, Paice90]。しかし要約対象のテキストが多く、また対象テキスト間で重複する記述がある場合、個々のテキストから重要箇所を抽出して並べただけでは、重複箇所が冗長であり、要約として適切ではない。そこで、関連するテキストをまとめてひとつの要約を生成する必要性がある。

2.1 複数テキスト要約の関連研究

本節では、複数テキストの要約生成のいくつかの手法について述べる。

2.1.1 船坂 ·Yamamoto の手法

機械可読な新聞記事から検索を行う時、検索の対象が長期に及ぶ事件や政治問題といった場合、検索の結果数多くの記事が現れそれらすべてに目を通すには多大な時間を要する。このような場合、関連する記事をまとめてひとつの要約を生成する手法を開発することは有用であると考えられる。ある記事で述べられていることは別の記事で述べられていることが多い。また、新聞記事には事実文と推量文があるが、事実文の方が重要であると考えられ、[船坂 96][Yamamoto95] では、関連新聞記事を冗長な部分と推量文を削除することにより要約する手法を提案している。

基本的な処理は、まず記事の第 1 段落のみを残し、第 2 段落以降は削除する。次に、形態素解析の結果から得られる表層的な情報を用いて不用な箇所の削除の処理を行ってい

る。[船坂 96] では不用な箇所の削除方法をいくつか提案しているが、ここでは 2 つ紹介する。

ひとつは**推量文の削除**という処理で、「〜見通し」、「〜模様だ」といった新聞特有の推量表現 20 個に着目し、これらが文末に出現した文を削除の対象にしている。

もうひとつは**導入部の削除**という処理である。ある事項についての一連の記事では、古い記事で述べられていることを新しい記事で再び述べている部分がある。このような要約には冗長な箇所を導入部とし、削除の対象としている。この処理も先程の推量文の削除の場合と同様に、「〜したが」、「〜事件で」といった導入部特有の表現に着目している。これらの語の出現した導入部で、導入部中の名詞の 7 割以上が古い記事中の名詞と一致すればその箇所を削除する。

これらの手法により、[船坂 96] では記事の第 1 段落で、文字削減率 14.5%を達成している。

2.1.2 柴田の手法

[柴田 97] では、Fit という検索システムに文章融合機能を埋め込み、自動分類されたテキスト (新聞記事) の融合を試みている。文章融合では関連文章から共通部分を同定する必要がある。以下、2 つの文章 (文章 1 と文章 2) を融合することを想定して話を進める。重複文を削除する際、まず文章全体に形態素解析を行い各形態素の出現頻度を調べる。この時、汎用的に用いられる頻度の高い形態素は除外しておく。次に、文章 1 と文章 2 の各文の組み合わせについて、一致する形態素を洗い出し、出現頻度に応じて以下の式 (2.1) を用いて評価を行う。

$$\frac{100}{(\text{文章 1 中の出現回数}) \times (\text{文章 2 中の出現回数})} \tag{2.1}$$

この評価方法は、出現頻度の低い形態素が異なる文章で出現した場合評価値が高くなり、そのような形態素が重複文の特定に有効であると [柴田 97] では考えている。実際、重複文の検出結果として、再現率 96%、適合率 96%が得られている。

また重複文検出後の融合文章生成では、3 タイプの文章 (AND/OR/PREFER) の生成を試みている。その中のひとつ、AND タイプでは、関連文献の中から最も短い文章を選

び、重複文だけ残したものを雛型とする。重複文は、係り受けを考慮して、形態素レベルで他の文章で用いられていないものを切り取る。例えば、以下の例文で、「届いた」が用いられていないとすると、これを切り取る。「ユーザに」は「届いた」に係っているので一緒に切り取ることになる。

文章 1 ユーザに届いた電子メールを読み上げる。

文章 2 電子メールの本文を読み上げる。

結果 電子メールを読み上げる。

2.1.3 Mani の手法

[Mani97] でも、関連した複数の新聞記事を要約の対象としている。[Mani97] では、まず、個々のテキスト中で意味的に近い語や関連語句にリンクを張る。次にテキスト間の類似箇所をこれまで紹介してきたような文や節単位ではなくグラフ単位で比較を行うことで、類似箇所と相違箇所の抽出を行っている。

グラフの作成方法について説明する。

1. テキスト中の関連語句 (e.g. "Bill Gates" と"President of Microsoft") 間にリンクを張る。その際、MUC6 で開発された SRA の NetOwl というシステムを用いて関連語句の抽出を行う。
2. TREC のコーパスを利用して、tf*idf によりテキスト中から特徴的な語を抽出する。
3. 語句間の意味的な近さを Wordnet 上の距離と考え、意味の近い語句 (以下、ノード) 間にリンクを張る。

上記の過程で作成されたグラフ間の類似箇所、相違箇所の抽出を行う。今、比較を行う一方のグラフ中のノード集合を G1、他方を G2 とする。G1 と G2 で共通のノード集合を *Common*、それ以外のノードを *Differences* と定義する。これらの 2 つの集合を用いて、グラフ間の共通箇所、あるいは相違箇所の抽出を行う。抽出の際、式 (2.2) を用いて各文のスコア計算をする。

$$Scores(s) = \frac{1}{|c(s)|} \sum_{i=1}^{|c(s)|} weight(w_i) \tag{2.2}$$

ここで c(s) は、共通箇所を抽出する場合は *Common* の中で個々の文中に含まれる語句の集合、相違箇所の場合 *Differences* の中で個々の文中に含まれる語句の集合を表す。また $weight(w_i)$ は tf*idf で計算される個々の語 (w_i) の重みである。ユーザの指定した抽出文の数に応じて、式 2.2で計算されるスコアの高い文から出力する。

2.1.4 McKeown の手法

[McKeown95] は、MUC-4 で生成されたテンプレートによる出力結果を用いて、複数の新聞記事の要約を試みている。記事はテロリストに関するもので、テンプレートにより犯人、犠牲者、事件のタイプなどの計 25 の情報を抽出する。[McKeown95] の **SUMMONS** というシステムは、まず抽出された情報を 7 種類のオペレータで取捨選択し、次に要約で用いる語彙を選択、大規模な文法辞書を用いて要約文書を生成する。

以下に情報選択のための 7 種類のオペレータを示す。

- **change of perspective**
  始めのレポートが誤りを含んでいたり、不完全な情報である時、変更点は普通要約に含まれる。このオペレータを利用するためには、元のフィールドは同じでなければならない。

- **contradiction**
  2 つの元になるレポートで同じイベントについての情報が相容れない時、矛盾が生ずる。要約はどちらを真実としてレポートすることも出来ないが、事実が明確でないことは示せる。

- **addition**
  後のレポートが知られていない事実を加えた時、これは要約に含まれる。始めのレポートの後に起こったイベントや、知られていなかった府か的な情報が加えられる。

- **refinement**
  後のレポートにおいて、より詳しい事実が詳細に論じられる。例えば、始めは 匕ュー

ヨーク市」とレポートされていたものが、後に 「ニューヨーク市の自治区」とされる。更新される情報は重要なので、要約に利用される。

- **agreement**
  もし 2 つの情報源で同じものがあれば、これは読み手の信頼を高め、レポートに使用される。

- **superset**
  もし同じイベントが違う情報源でレポートされ、それらが不完全な情報であるならば、それらを組み合わせてより完全な要約を生成することが出来る。

- **trend**
  2 つ以上のメッセージに類似点があれば、傾向があるといえる。例えば 3 件の爆弾事件が同じ場所で起こった場合、これをひとつの文で表せる。

- **no information**
  我々は情報源を信じているので、例えば、確かな新聞通信社がテロリストについて報道し、ある国ではそれを発表しなかった時、このオペレータを使用する。

ここに示した 7 種類のオペレーションにより情報を選択し、要約を生成する。生成の際、抽出された情報と要約コーパスから発生させたフレーズを用いる。

2.2 関連研究と本研究との比較

[Mani97]、 [船坂 96]、 [Yamamoto95]、 [McKeown95]、[柴田 97] はいずれも、新聞記事を要約対象にしている。新聞記事を要約対象にした場合の利点を以下に挙げる。

1. 電子化されたデータが容易に、また大量に入手可能。
2. 記事中で使われる語句がある程度統一されているため、記事間の言い回しの違いにそれほど注意を払わなくても比較的文体の整った要約生成が可能である。
3. 新聞記事では、概して客観的な事実しか述べられていない。したがって、記事間で重複した箇所を特定するのが比較的容易である。

4. 正解データが作りやすい。また、誰が作ってもそれほど大きく異なるわけではない。

また、新聞記事を対象にした場合の問題点を以下に示す。

- ある事件に関する要約は、雑誌やテレビの特集、あるいは書籍といった他のメディアでも得られる場合が多い。特に、事件が大きければそれだけリアルタイムに他のメディアで情報が得られる可能性が高い。
- 新聞記事以外のアプリケーションが非常に限られてくる。

次に、学術論文を要約対象にした場合の利点を挙げる。

- Web 等を利用することで、電子化された論文データが大量に入手可能。
- 学術論文は、参照関係という点でみるとハイパーテキストの構造をしており、Web を始めとする他のハイパーテキスト構造をしたメディアへの適用が比較的容易であると考えられる。
- 関連論文を収集する際、既存の数多くの論文データベースが利用できる。
- 論文には、著者名、論文題目、索引語等たくさんの属性情報があり、これらを参照関係と合わせて利用することで、論文間の関係を明らかにしやすい。

学術論文を要約対象にした場合の問題点を挙げる。

- 論文の著者毎に言い回しが異なり、要約生成の際、新聞記事と比較して処理が大変。
- 要約文書が作成者毎に大きく異なるものと考えらるため、システムの生成した要約の評価が困難。

以上をまとめると、複数テキストの要約は、対象テキストが学術論文の場合、新聞記事を要約する場合と比較して、要約生成の際考慮すべき点が多い。また、評価方法が難しい。しかし、参照関係を考慮した学術論文の要約が可能になれば、論文以外の他の多くのテキストへの適用が可能になると思われる。

第 3 章

複数テキスト要約における参照情報の利用

本研究では、論文間の関係を解析する際、論文の参照情報に着目する。ある論文が他の論文を参照する場合、参照論文について記述してある箇所 (参照箇所) が存在する。その箇所を読むことで、著者がどのような目的で参照しているのか明らかになる。このようにして参照箇所から得られる情報を**参照情報**と呼ぶ。参照情報を収集し整理することで、ある分野の複数の論文間の関係が明らかになり、またそれらの参照情報が要約生成に利用できると考えられる。

図 3.1は論文間の参照関係を示したモデルである。図は要約対象の論文 (target papers)3 本と、それらを参照している 2 本の論文から構成されている。 target papers に関する記述が図の上の 2 本の論文中、参照箇所に記述されている。この参照箇所を解析することで、2 本の論文がそれぞれどのような目的で target papers を参照しているのかがわかる。いいかえれば、参照の目的を明らかにすれば論文間の関係が明確になると言える。

参照目的を把握するためには、その前処理として論文中から参照箇所を抽出するという作業が必要となる。参照箇所とはどういったものであるか、またどのように参照目的を分類すれば良いかを順次述べていく。

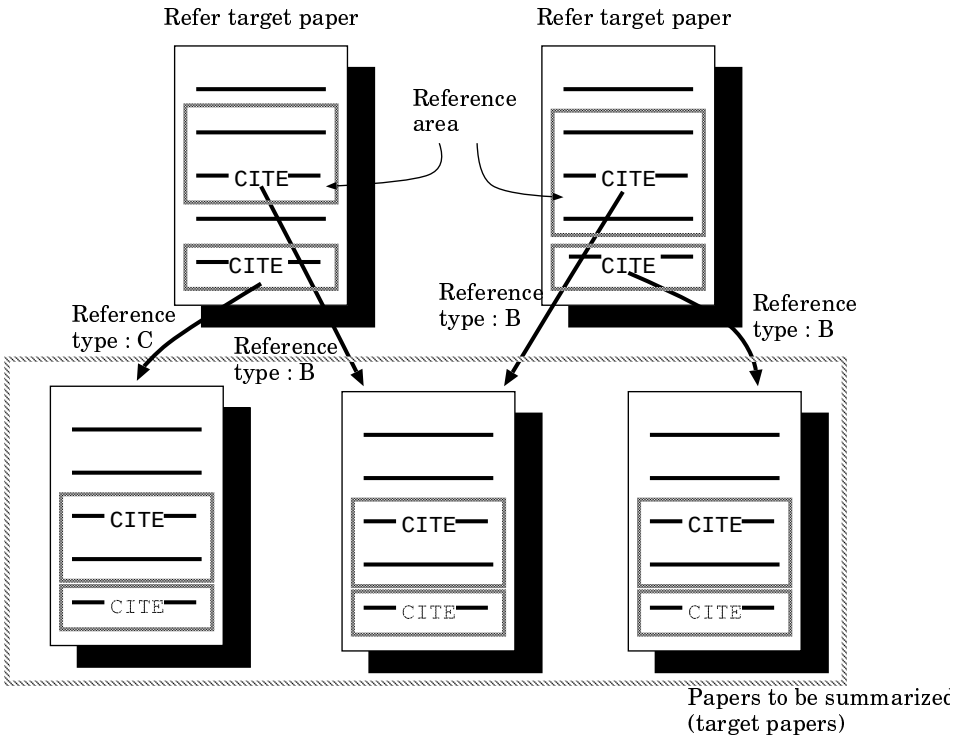

図 3.1: 論文間の参照関係

3.1 参照箇所とは

1. **Recently, rule-based approaches are re-studied to cope with the limitations of statistical approaches by learning the tagging rules automatically from the corpus [Brill94].**

2. **Some systems even perform the POS tagging as part of syntactic analysis process [Voutilainen95].**

3. **However, the rule-based approaches alone are in general not robust to handle the unknown words , and is not flexible to adjust to the new tag-sets and languages.**

4. **Also the performance is usually no better than the statistical counterparts [Brill94].**

5. **To gain flexibility and robustness and also to overcome the limited window range of statistica l approaches, we need a method that can combine both statistical and rule-based approaches [Tapanainen94].**

前ページの囲みの中で示す文は [Brill94] について記述された箇所である。本研究では、これを参照箇所 (reference area) と定義する。参照箇所を読むことで、著者がどういった目的でその論文を参照したのかがわかる。例えば、この例の場合、文 1 に [Brill94] のようなルールベースの tagging の研究がされている、といったことが書かれてある。文 3,4 では [Lee95] がルールベースの tagging の問題点を指摘している。この参照箇所から、[Lee95] は [Brill94] を既存の研究の問題点を指摘するために参照していることが分かる。

参照箇所からどのような目的で他の論文を参照しているのか明らかにすることは重要である。そこで、参照の目的をいくつかに分類する。またこれらを**参照タイプ (reference type)** と呼ぶ。

3.2 参照目的の分類

論説根拠型 ($Btype$)	ある理論を提案する場合や仮定をする場合、その根拠となる論文
問題点指摘型 ($Ctype$)	他の論文の理論や手法等の問題点を指摘する
その他型 ($Otype$)	$Btype$ にも $Ctype$ にも分類が難しい論文

以下 3 節で 3 つのタイプの詳細な説明を行う。

3.2.1 論説根拠型 (B type)

我々が、新しい理論を提唱したり、あるいはシステムを構築する際、他の研究者の研究の成果を利用する場合がある。例えば、他の研究者が提唱する概念を用いて何か別の理論を提唱する場合がある。あるいは、他の研究グループが作ったツールを自分の研究に役立てるという場合もある。また、実験用コーパスとして整備、 公開されているものを、自分の研究に用いる場合がある。このように、既存の物を利用して何か別のことをする際に参照する、こういった参照タイプを論説根拠型 (B type) とする。以下に B type の参照の例をいくつか示す。

> **"The analysis introduced in this paper has been implemented in NTT Communication Science Laboratories' Japanese-to-English machine translation system ALT-J/E [Ikehara95]."**

この引用はある論文の著者が過去の自分の論文を引用している例である。

> **"There are various definitions for TFS unification, and we base our unification algorithm on the definition given in [Carp92]."**

論文の著者が他の研究者の提案を基に、新たな提案を行う場合がある。この例では [Carp92] の unification algorithm を基に、この論文の著者が新たに TFS unification algorithm の定義を行っている。

> **"Various solutions to the problems of generating articles and possessive pronouns and determining countability and number have been proposed [Murata93,Cornish94,Bond95]."**

ある open problem を取り扱った論文として、いくつかの関連論文を引用している。

> **"The corpus we have used is the 1988, 1989 Wall Street Journal.[Liberman91]"**

実験用コーパスとして Wall Street Journal を用いている。

3.2.2 問題点指摘型 (C type)

我々が新しい理論を提案したり, あるいは新しいシステムを作る場合, 関連研究との比較を行う. こういった目的の参照タイプを問題点指摘型 (C type) と呼ぶ。

> Recently, rule-based approaches are re-studied to cope with the limitations of statistical approac hes by learning the tagging rules automatically from the corpus [brill94]. Some systems even perform the POS tagging as part of syntactic analysis process [voutilainen95]. **"However, the rule-based approaches alone are in general not robust to handle the unknown words."**

参照している文献の問題点を指摘している。この例では、ルールベースの tagging の手法では未知語の取り扱いの点で頑健さに欠けるという問題点を指摘している。

"Previous work on extraction of collocation for use in generation [Smadja91] is …. However, extracted collocations were used only to determine realization of an input concept. In our work, stored phrases would be used to provide content…."

既存の研究の問題点を指摘しつつ、同時にその問題点に対する著者らの提案を述べている。

3.2.3 その他型 (O type)

O タイプの参照は, B タイプ,C タイプいずれにも分類しがたいものである. このタイプの参照は、要約生成の際あまり有用ではないと考えられる。

"The first experiment on automatic abstracting was reported in a paer by H.P.Luhn published in 1958[Luhn58]."

この例は、要約研究の原点とも言える Luhn の論文の引用である。この種の論文は非常に多くの論文で参照されており、ある意味では関連研究における基本的な論文であると言えるが、論文の細部にまで言及されることはあまりない。

3.3 参照情報を利用した要約生成

直接参照関係にない論文でも、参照情報を利用することでひとつの要約にまとめることが可能になる。図 3.2は、機械翻訳に関する論文の参照関係の一例を示している。

図において、[Ikehara95] と [Murata93] を要約する場合を考える。この 2 本の論文は直接参照関係にはない。しかし、この 2 本を共引用する論文が存在する [Bond96]。[Bond96] の参照情報を利用することで、[Ikehara95] と [Murata93] に関する要約を生成することが

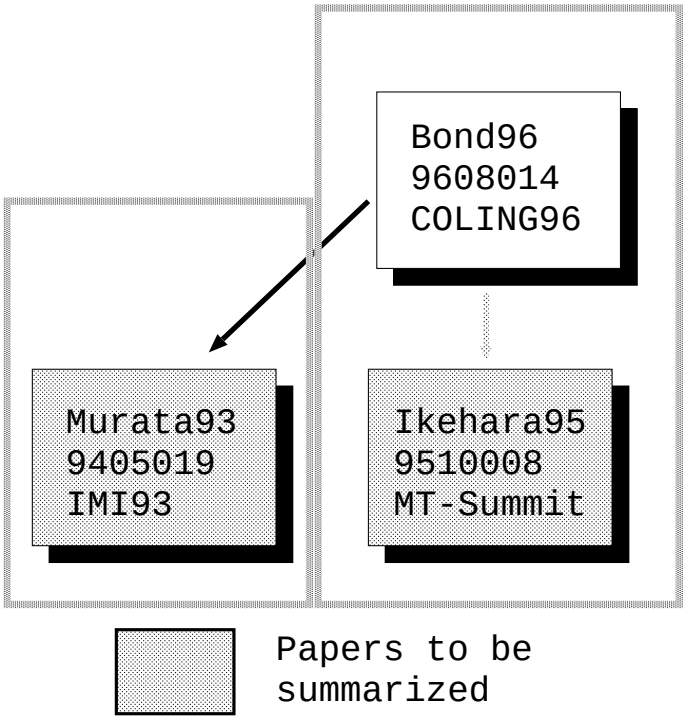

図 3.2: 論文間の参照関係の一例

可能になる。例えば [Bond96] の中では [Murata93] の問題点を指摘している (**C type**)。また [Bond96] が [Ikehara95] をベースにした研究である (**B type**) ということも [Bond96] 中からわかる。

要約生成の際に必要な情報を参照箇所から抽出する必要がある。問題点を指摘する場合、参照箇所中に逆接の接続詞や否定の副詞が出現する。このような手がかり語に着目することで、問題点指摘の箇所が抽出できるものと考えられる。本研究では、このような手がかり語を cue word と呼ぶ。B type の参照箇所中においても同様に、“base on” や”apply to” といった cue word に着目することにより、 被参照論文をどの点でベースにしているのか」に関する記述箇所が抽出可能であると思われる。

論文間の関係が明らかにされた後は、個々の論文がどのような手法を用いて、どのような結果が得られたのか等の情報を本文から抽出する必要があるが、ごく簡単に論文の abstract を利用するという方法もある。また、各論文の著者名を比較することで、同じ研究チームであるかどうか判断できる。例えばこの例の場合では、著者名の比較により [Bond96] と [Ikehara95] が同じ研究チームであることがわかる。こういった、要約生成の際、有効な情報であると考えられる。

このように、参照情報を用い、他の情報と組み合わせることによって [Bond96]、[Ikehara95]、[Murata93] の 3 本を以下のようにまとめることができる。

> 日英機械翻訳において、日本語にはない冠詞や数詞を英語に翻訳する際考慮に入れなければならない。[Murata93] では照応属性と、数の属性の分類方法の提案を行っている。一方で [Bond96] では [Murata93] の**問題点を指摘している**。[Murata93] では数詞表現を分類する上で、日本語と英語の表現上の違いを考慮にいれていない。[Ikehara95] では 1994 年から ALT-J/E というシステムを作っており。このシステムをベースに [Bond96] の提案手法を実装している。[Bond96] では日本語と英語の表現上の違いを考慮した名詞句の分類方法を提案している。

さて、ここでは非常に小さなスケールでの要約生成手法について説明したが、参照関係のスケールが大きくなると様々な問題が出てくる。この問題点については、今後の課題で触れる。

第 4 章

要約生成

要約生成のプロセスを図 4.1に示す。システムには複数の被要約論文が入力される。要約生成のプロセスは大きく以下に示す 3 つのルーチンに分割できる。

(a) 参照関係解析ルーチン

(b) 参照情報抽出ルーチン

(c) 要約生成ルーチン

システムの入力としては複数の論文 (以下、target papers)、出力は target papers と target papers を参照している論文をまとめた要約が一つ生成される。各ルーチンの詳しい説明は 4.1節で述べる。なお、本研究の最終的な枠組としては、複数の論文を入力として取り扱うが、修士研究の範囲では target paper として 1 本の論文のみを受け付け、1 本の target paper と、それと参照関係にある論文から抽出した情報を用いて生成した要約を目的の出力とする。したがって、図 4.1の図中の target paper 間の解析、及びその解析結果を用いた要約生成の手法については触れない。

アプリケーションとして World Wide Web 上のデータベース e-Print archive[1]の中の"The Computation and Languages" のドメインの TEX ソース約 450 本を用いる。

[1] http://xxx.lanl.gov

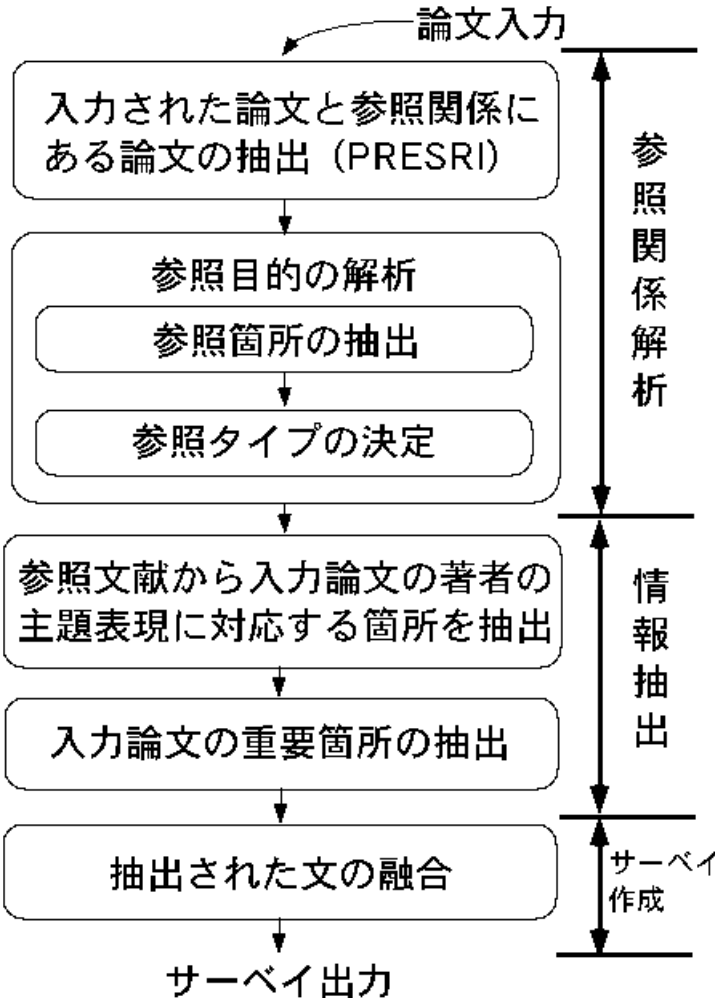

図 4.1: 要約生成のプロセス

4.1 参照関係を用いた論文検索システムの構築

本節では図 4.1の最初の処理 「入力された論文と参照関係にある論文の抽出」について述べる。ここでは、入力論文 (target papers) と参照関係にある論文をデータベースから自動的に収集することを目的とする。

まず、最初にデータベースの論文間の参照関係を解析する必要がある。 TEX ソースの bibliography を解析して、論文間の参照関係を明らかにした。以下、参照関係解析システムの構築方法について述べる。

論文の一般的な構成として、最後に参考文献を載せる。TEX ファイルでは bibliography というコマンドを用いて記述する方法が一般的である。本研究では、参照箇所から情報を

抽出する際、TEX の cite コマンドを利用する。そのためには、参考文献が bibliography コマンドを用いて記述されていることが必要条件となる。しかし、全ての TEX ソースが bibliography を用いて参考文献を記述しているわけではない[2]。そこで、ftp 経由で e-Print archive から収集した TEX ソースファイルが参照関係の解析、あるいはその後の処理の参照情報の抽出が可能であるかどうかを、(1)TEX ファイル中で bibliography というワードが出現するかどうか、(2)bibitem…の…の箇所が空になっていないか、という 2 つの情報に着目して自動判別した。

4.1.1 参照関係の解析

以下の囲みに示すのは、TEX ファイル [Bond96] の参考文献の記述の一部を抜粋したものである。[Bond96] は論文中で [Murata93] を参照している。

```
\bibitem[\protect\citename{Murata and Nagao}1993]{Murata:1993a}
Murata, Masaki and Makoto Nagao.
\newblock 1993.
\newblock Determination of referential property and number of nouns in
{Japanese} sentences for machine translation into {English}.
\newblock In {\em Proceedings of the Fifth International Conference on
Theoretical and Methodological Issues in Machine Translation (TMI '93)}, pages 218–25,
July.
```

e-Print archive には e-Print archive から得られる論文のリストファイルが存在する。以下は、上の引用に対応する e-Print archive リストファイルの一部を抜粋したものである。

[2]例えば itemize や verbatim コマンドを用いて記述される場合がある。

\\ Paper: cmp-lg/9405019

From: Masaki MURATA ¡murata@jungle.kuee.kyoto-u.ac.jp¿

Date: Thu, 19 May 94 16:21:07 JST (9kb)

Date (revised): Fri, 20 May 94 16:53:05 JST

Date (revised): Sat, 21 May 94 13:46:18 JST

Date (revised): Mon, 23 May 94 15:29:39 JST

Title: Determination of referential property and number of nouns in Japanese sentences for machine translation into English

Author: Masaki Murata, Makoto Nagao

Comments: 8 pages, TMI-93

\\

これらが同一のものであると判定できた時、初めて [Bond96] が [Murata93] を参照していると言える。そこで、TEX ファイルの bibliography から、参照されているタイトルや著者名を切り出して e-Print archive のリストファイルで検索することで、論文間の参照関係の解析を行う。

bibliography から論文のタイトルや著者名を一字一句間違えずに正確に抽出することは困難である。そこで、bibliography から著者名や文献名に含まれる単語を可能な限り抽出し、それらの単語がすべて含まれるような論文データを e-Print のリストファイルから選択することを試みる。

参考文献の記述形式はおおよそ以下の 4 つのパターンに分類できる [齊藤 93]。

1. [著者名][発行日][文献名][誌名][巻号][ページ]
2. [著者名][文献名][誌名][発行日][巻号][ページ]
3. [著者名][文献名][誌名][巻号][発行日][ページ]
4. [著者名][文献名][誌名][巻号][ページ][発行日]

参考文献のスタイルは、通常、学会毎にスタイルが定められている。そこで、処理の対象となる e-Print archive の”The Computation and Language” の分野の論文、COLING、

で、ACL、EACL、IJCAI、AAAIの5種類の論文誌各4本づつ、計20本の論文で、上記のどのパターンにあてはまるのか調査した。結果を以下に示す。

1. COLING、ACL、EACL、AAAI

2. COLING、IJCAI

1. と 2. のスタイルで、著者名、文献名は e-Print のリストファイルに必ず記述されている事項である。また発行日 (著作年) の情報もほぼ記述されている。しかし、誌名、巻号、ページについてはリストのデータに記述はあっても表記のゆれが大きく、検索の際あまりキーワードとして有効でないと考えられる。そこで、TEX の bibliography 内の個々の bibitem において、なるべく最初に記述されている単語をキーワードとして切り出して e-Print のリストファイルに検索をかけることを試みた。

ここで、表記のゆれについてもう少し触れておく。実は著者名にも表記のゆれがかなりある。例えば、“James Allen” は”J.Allen” と省略形で書かれることがある。著者名の省略形は bibliography 内の記述で用いられるケースが多い。一方、e-Print のリストファイルではあまり省略形が用いられない。したがって、J.Allen の場合、bibliography から”J”、“Allen” という 2 つのキーワードを切り出せば、リスト上の”James” の”J” と”Allen” でキーワードがマッチングする。従って著者名の表記のゆれにある程度対処可能である。

個々の bibitem の次の 3 行に含まれる単語からアルファベットと数字以外のデータはすべて除去した。また First author と 2 人目以降の著者名の” and” も除去し、残った語をキーワードとして検索を行った。先の例の場合、以下のものがキーワードとして用いられる。

“Murata”、“Masaki”、“Makoto”、“ Nagao”、“1993”、“Determination”、“of ”、“referential”、“property”、”and”、“number”、“of”、“nouns”、“in”

以上述べてきた手法で、e-Print archive の TEX ソース 400 本で参照関係を解析を行った。システムが参照関係があると判断したもののうち、100 個について人手で調べてみた。その結果、94%が実際に参照関係があった。解析に失敗したものの原因として、

(1) 同著者、同タイトルの本と論文の区別がつかなかった

(2) 論文の著者の bibliography の書き方が特徴的で、キーワードがあまり抽出できず、少ないキーワードで別の論文が検索されてしまった

などが挙げられる。

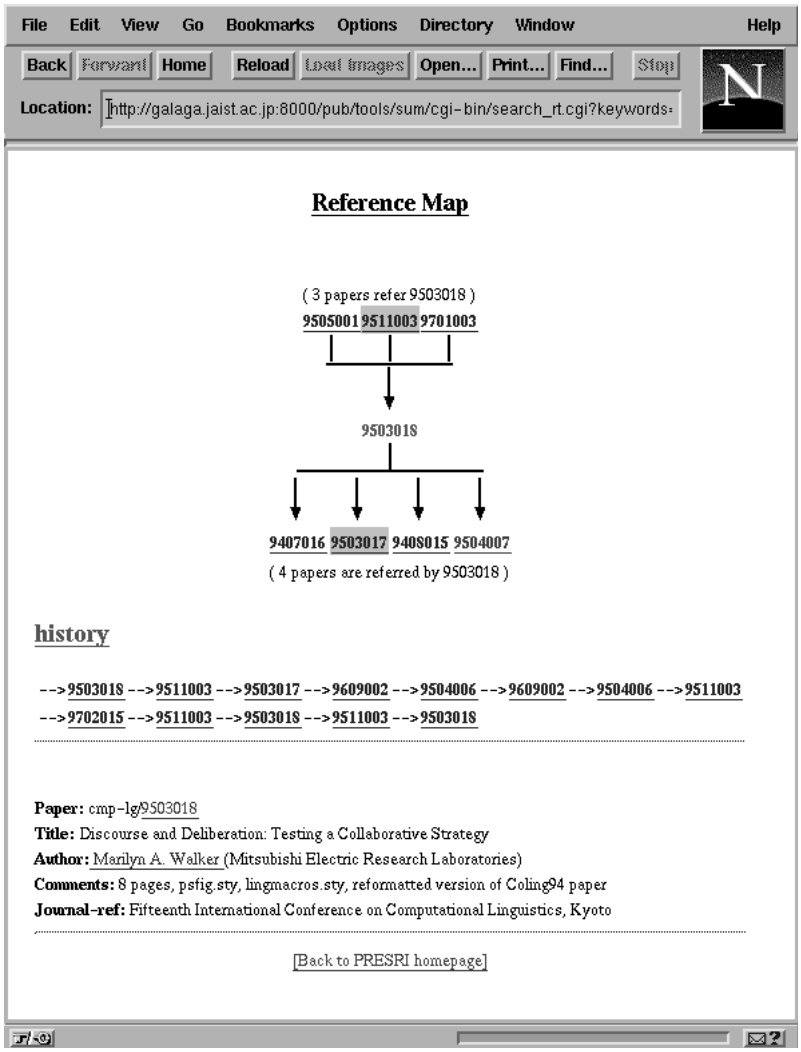

図 4.2: 参照関係による論文検索システム PRESRI

4.1.2 論文検索インターフェースの作成

前節で解析した参照関係を利用した論文検索システム PRESRI(Paper REtrieval System Using Reference Information)[3]を CGI を用いて実装した。

この検索システムでは、e-Print archive の検索システムと同様、論文のタイトル中の語、著者名からキーワード検索が可能である。検索結果の中で、e-Print archive 上の他の論文と参照関係にある場合、検索結果のリストに [–)Reference Map] という表示が現れ、これをクリックすることで、図 4.2のような参照関係のグラフが表示される。このグラフをた

[3]http://galaga.jaist.ac.jp:8000/pub/tools/sum/

どることにより、参照関係にある論文を検索することができる。

4.2 参照箇所抽出

本節では図 4.1の 「参照箇所抽出の手法」について述べる。参照箇所とは、論文中で他の論文について記述している箇所である。

次に、参照タイプを決定するためのステップとして論文中から参照箇所を抽出することを試みた。

参照箇所抽出の際、文間の結束性に着目した。それらの結束性は大まかに、以下の 5 種類に分類される。

- 照応詞
- 接続詞
- 1 人称代名詞
- 3 人称代名詞
- その他結束性のある語

これらの結束性を基に、7 クラス、86 個の cue word list を作成した。cue word list は、参照箇所コーパス 200 箇所から抽出した uni-,bi-,tri-gram を、人手で分類整理して作成した。7 クラスの cue word list を表 4.1 に示す。

また、論文中で用いられているシステム名にも着目した。システム名の自動抽出は、論文のアブストラクト中で、 「大文字のみから構成される単語をシステム名とみなす」という [Kupiec95] のヒューリスティックスを用いている。

cue word を用いて、次に示す 11 種類のルールを作成した。これらのルールにより参照箇所抽出を試みた。参照箇所は TEX の cite コマンドの前後の文を抽出する。ルールを適用して抽出を行うわけであるが、抽出過程で候補として選択されている文の中で一番最初にあるものを FIRST SENTENCE、一番最後にあるものを LAST SENTENCE とする。最初は cite コマンドの出現する文のみを参照箇所とみなす。

1-1 FIRST SENTENCE が this.cue で始まる場合, 前の文も抽出する。

1-2 FIRST SENTENCE が but.cue で始まる場合, 前の文も抽出する。

1-3 FIRST SENTENCE が and.cue で始まる場合, 前の文も抽出する。

1-4 LAST SENTENCE の次の文が but.cue で始まる場合、次の文も抽出する。

1-5 LAST SENTENCE の次の次の文が but.cue で始まる場合、次の次の文まで抽出する。

1-6 LAST SENTENCE に we.cue が含まれなくて、次の文に we.cue が含まれる場合、次の文も抽出する。

1-7 LAST SENTENCE に we.cue が含まれなくて、次の次の文に we が含まれる場合、次の次の文まで抽出する。

1-8 LAST SENTENCE に we.cue が含まれなくて、次の文に大文字のみシステム名が含まれる場合次の文も抽出する。

1-9 LAST SENTENCE の次の文が and.cue で始まる場合、次の文も抽出する。

1-10 LAST SENTENCE の次の文に they.cue が含まれる時、次の文も抽出する。

1-11 LAST SENTENCE の次の文に this.cue が含まれる場合、次の文も抽出する。

4.3 参照タイプ決定

つぎに図 4.1の 「参照タイプの決定」について説明する。参照タイプを決定するのに、先に述べた cue word list の並びを考慮してタイプ決定を行うことを試みた。表 4.2にタイプ決定ルールを示す。

これらのルールのうち、先頭の 12 個は cue word の並びで参照タイプを決定する。また残りの 2 つは、先頭の 12 個のルールで決定されなかったものに対して適用される。

4.4 要約生成

要約生成の手法については、例を挙げて 3 章で述べたが今一度まとめる。以下は、複数の参照関係のない target papers から target papers を共引用する論文の参照情報を用いて要約文書を生成する過程を示す。

1. C type の参照箇所から but.cue の出現する文を抽出。これを、被参照論文の問題点の指摘箇所と考える。

2. B type の参照箇所から base.cue の出現する文を抽出。これを、 「被参照論文をどの点でベースにしているのか」について記述されている箇所と考える。

3. 個々の target paper から 「どのような問題を取り扱った論文か」 「どのような手法を用いたのか」 「どのような結果がえられたのか」といった情報が記述されている箇所を抽出する。当面の間は、論文の abstract を利用する予定である。

4. 上記の抽出された箇所を組み合わせて要約を生成する。

抽出箇所を組み合わせて要約を生成する際、生成される要約の文脈を考慮する。

1. どのような領域の論文の要約であるのか
2. C type で参照される論文に関する記述
3. 上記の論文の問題点の指摘
4. その問題点に対する参照論文の解決方法
5. 参照論文が B type で参照している論文の記述

現状は、まだ参照タイプの決定の処理までしかできていない。今回はひとつの target paper について、target paper を参照している論文から参照箇所を切り出し、決定された参照タイプ毎に並べるシステムを構築した。

表 4.1: 7 クラスの cue word list

list name	a part of cue word
we.cue(9)	we,We,our,Our,us,I,my,My,me
this.cue(10)	In this, On this, In these, On these
base.cue(16)	base, basis, adopt, apply
and.cue(8)	and, furthermore, additionally,
but.cue(15)	however, but, In spite of
they.cue(18)	they, their, them, he, his, him
other_C(10)	difference between, different

表 4.2: 参照タイプ決定のルール

No.	適用ルールの内容
B-2	cite の文に base.cue がある場合 B
B-3	cite の文に we.cue がある場合 B
B-4	cite の文に this.cue がある場合 B
B-5	cite の文にシステム名がある場合 B
C-1	cite の文に however.cue がある場合 C
C-2	cite の文に other_C.cue がある場合 C
C-3	cite の文以降 however.cue がある場合 C
C-4	cite の文以降 other_C.cue がある場合 C
B-6	cite の文より前に base.cue がある場合 B
B-7	cite の文より前に we.cue がある場合 B
B-8	cite の文より前に this.cue がある場合 B
B-8	cite の文より前にシステム名がある場合 B
O	参照箇所が 1 文で cue word がない場合 O
B-1	適用ルールがない場合は B

第 5 章

実験

5.1 参照箇所抽出実験

評価を以下に示す Recall と Precision で行う。

$$Recall = \frac{\text{抽出された文のうち正解のものの数}}{\text{参照箇所コーパスの抽出すべき文の総数}} \tag{5.1}$$

$$Precision = \frac{\text{抽出された文のうち正解のものの数}}{\text{参照箇所抽出ルールにより抽出された文の総数}} \tag{5.2}$$

実験用コーパスとして 100 個の参照箇所、評価用 50 個を用意した。まず、実験用コーパスを用いて実験を行った。rule 1-1 〜 rule 1-11 を用いて実験した結果を表 5.1に示す。

表 5.1: 参照箇所抽出精度 (実験用コーパス)

Recall	Precision
0.909	0.769

5.1.1 参照箇所抽出ルールの組み合わせの最適化

これらの 11 種類のルールの中には抽出精度に貢献せず、むしろ精度を下げるだけのものも存在すると考えられる。そこで、11 種類のルールの組み合わせ 2^{11} 通りについて実

験を行い、どの組み合わせの時に Recall が最も高くなるか調べた。その結果、rule 1-11 以外を適用した場合精度が最も高くなった。結果を表 5.2に示す。

表 5.2: ルールの組み合わせが最適化の状態での参照箇所抽出精度 (実験用コーパス)

Recall	Precision
0.909	0.772

また、使用頻度の高かったルールを以下に示す。

- rule 1-9(47)
  LAST SENTENCE に"we","our","us"が含まれなくて、次の文に"we","our","us"が含まれる場合、次の文も抽出する。
- rule 1-10(17)
  LAST SENTENCE に we,our,us が含まれなくて、次の次の文に we が含まれる場合、次の次の文まで抽出する。
- rule 1-1(1)
  FIRST SENTENCE が、"This","These","In this","In these","For this","For these","On this","On these"で始まる場合、前の文も抽出する。
- rule 1-11(13)
  LAST SENTENCE に"we","our","us"が含まれなくて、次の文に大文字のみ 4 文字以上から構成される語が含まれる場合それはシステムの名前とみなし、次の文も抽出する。
- rule 1-13(10)
  LAST SENTENCE の次の文が"And","Furthermore","Additionally","Again","Because","So"で始まる場合、次の文も抽出する。

5.1.2 参照箇所抽出実験結果

評価用コーパスを用いて実験を行った結果を表 5.3に示す。

表 5.3: 評価用コーパスの参照箇所抽出精度

Recall	Precision
0.796	0.763

5.2 参照タイプ決定実験

参照タイプ決定実験の評価方法として、式 (5.3) を用いる。

$$accuracy\dagger = \frac{\text{ルールにより正しく参照タイプが決定できた参照箇所の数}}{(\text{参照箇所の数})} \tag{5.3}$$

5.2.1 参照タイプ決定ルールの適用順序

参照タイプ決定のルールは適用順番で精度が変動する。そこで、順序を入れかえて、どの順序でルールを適用するのが最適であるか調査した。

上位 12 種類のルールは以下の 4 クラスに分けられる。クラス内ではプログラム上、 順序を入れかえても精度に変化はない。したがってこれらの 4 クラスの順序を入れかえた場合 (4!=24 通り) の精度の変動を調べた。

- class a:cite の文に base,we,this.cue, システム名が出現したら B type(rule 1 2 3 4)
- class b:cite の文に however,other_C.cue が出現したら C type(rule 7 8 9 10)
- class c:cite の文以降に however,other_C が出現したら C type(rule 5 6)
- class d:cite の文以前に base,we,this,cue, システム名が出現したら B type(rule 11 12)

その結果、class c のルールを最初に適用した場合は、あとの a,b,d の順序に関係なく精度が最大となった。したがって、ルールの適用順序は class c、後は a,b,d とした。

5.2.2 参照タイプ決定実験

ルールを用いたタイプ決定精度を表 5.4、5.5に示す。なお、この実験では、まず cite の含まれる段落に参照箇所抽出ルールを適用している。次に参照タイプ決定ルールを適用して精度を算出している。

表 5.4: 14 種類全てのルールを用いた参照タイプ決定実験結果 (その 1)

		正解タイプ			
No.	適用ルールの内容	B	C	O	精度
C-3	cite の文以降 however.cue がある場合 C	1	6	4	0.55
C-4	cite の文以降 other_C.cue がある場合 C	0	0	0	0.00
B-2	cite の文に base.cue がある場合 B	27	4	3	0.79
B-3	cite の文に we.cue がある場合 B	43	4	3	0.91
B-4	cite の文に this.cue がある場合 B	0	0	0	0.00
B-5	cite の文にシステム名がある場合 B	13	1	3	0.76
B-6	cite の文より前に base.cue がある場合 B	2	1	1	0.50
B-7	cite の文より前に we.cue がある場合 B	11	4	2	0.65
B-8	cite の文より前に this.cue がある場合 B	2	1	0	0.67
B-9	cite の文より前にシステム名がある場合 B	2	1	0	0.67
C-1	cite の文に however.cue がある場合 C	6	20	4	0.67
C-2	cite の文に other_C.cue がある場合 C	0	0	0	0.00
B-1	適用ルールがない場合は B	41	9	26	0.54
O	参照箇所が 1 文で cue word がない場合 O	13	5	16	0.47

表 5.5: 14 種類全てのルールを用いた参照タイプ決定実験結果 (その 2)

参照タイプ決定精度	参照箇所抽出精度	
Accuracy†	Recall	Precision
0.689	0.746	0.775

5.2.3 適用ルールの選択によるタイプ決定精度の向上

形式処理だけでは、ほとんど限界に来ていると考えられた。そこでさらに解析精度を向上させるために、適用ルールの選択を考えた。現在用いているルールは 14 種類あるが、そのうちルール O とルール B-1 は他の 12 種類のルールで分類できなかったものを無理矢理分類するものである。そこで、無理に分類するのであれば、多少の coverage を犠牲にしても accuracy を高めようというのが、適用ルール選択の目的である。

$$coverage = \frac{\text{ルールを用いて分類された参照箇所の数}}{\text{全ての参照箇所の数}} \tag{5.4}$$

$$accuracy = \frac{\text{ルールでタイプが分類されたもののうち正解の数}}{\text{ルールを用いて分類された参照箇所の数}} \tag{5.5}$$

12 種類のルールを用いた参照タイプ決定の精度を表 5.6に示す。

表 5.6: 参照箇所決定ルールを用いた後、参照タイプを決定した場合

参照タイプ決定精度			参照箇所抽出精度	
Accuracy†	Coverage	Accuracy	Recall	Precision
0.689	0.562	0.784	0.746	0.775

また、参照箇所として人手で作成したものを与えた場合の精度を算出した。さらに、参照箇所抽出の有効性を調べるため、参照箇所として cite の出現する段落全体を与えた場合についても調べてみた。

その結果、参照箇所として人手で作成したものを与えた場合 (表 5.7) も、参照箇所抽出ルールを用いた場合 (表 5.6) も、coverage と accuracy に違いが見られなかった。また、参照箇所として cite の出現する段落を与えた場合 (表 5.8)、coverage、accuracy ともに低下した。これより参照タイプ決定の精度向上には参照箇所抽出を行うことは有効であり、またある程度の参照箇所抽出精度が得られれば、参照タイプ決定の精度にあまり影響を及ぼさないと考えられる。

表 5.7: 正解の参照箇所を与えて、参照タイプを決定した場合

参照タイプ決定精度			参照箇所抽出精度	
Accuracy†	Coverage	Accuracy	Recall	Precision
0.641	0.562	0.784	1.000	1.000

表 5.8: 参照箇所として cite の出現する段落全体を与えた場合のタイプ決定精度

参照タイプ決定精度			参照箇所抽出精度	
Accuracy†	Coverage	Accuracy	Recall	Precision
0.630	0.730	0.731	1.000	0.364

5.3 要約の出力

要約の出力例を APPENDIX A に示す。

第 6 章

考察

6.1 参照箇所の抽出

参照箇所の抽出は、TEX の cite コマンドの含まれる段落の前後数文から行っている。11 種類のルールを用いた結果、約 80%の精度で参照箇所が抽出できることがわかった。そのタスクの難度について考察する。cite コマンドの含まれる文のみを参照箇所として抽出した場合、cite の文は必ず参照箇所として抽出されるため、Precision は必ず 1.00 になる。一方で、cite が含まれる段落そのものを参照箇所として抽出した場合、参照箇所として抽出されうる文はすべて含まれてしまうため Recall が必ず 1.00 になる。したがって、この 2 点を結んだ直線を参照箇所抽出の際のベースラインと考えられる。このベースラインに対する訓練用、および評価用データでの抽出実験結果を図 6.1に示す。

図 6.1中の直線がベースラインであり、Recall = Precision = 1.0 の点で、抽出精度が最も高くなる。図 6.1より、訓練用データの場合も、評価用の場合もベースラインを上回っており、参照箇所抽出ルールの有効性が示されている。

11 種類の抽出ルールのうち、以下に示すルールは Recall 向上に貢献せず、逆に Precision 低下にだけ作用したため、抽出ルールから除去した。ただし、Precision の低下といっても、実験用の参照箇所 1ヶ所に対して作用したにすぎないので、さらに大きなデータセットで実験した場合は Recall 向上に貢献する可能性も考えられる。

1-2 FIRST SENTENCE が but.cue で始まる場合, 前の文も抽出する。

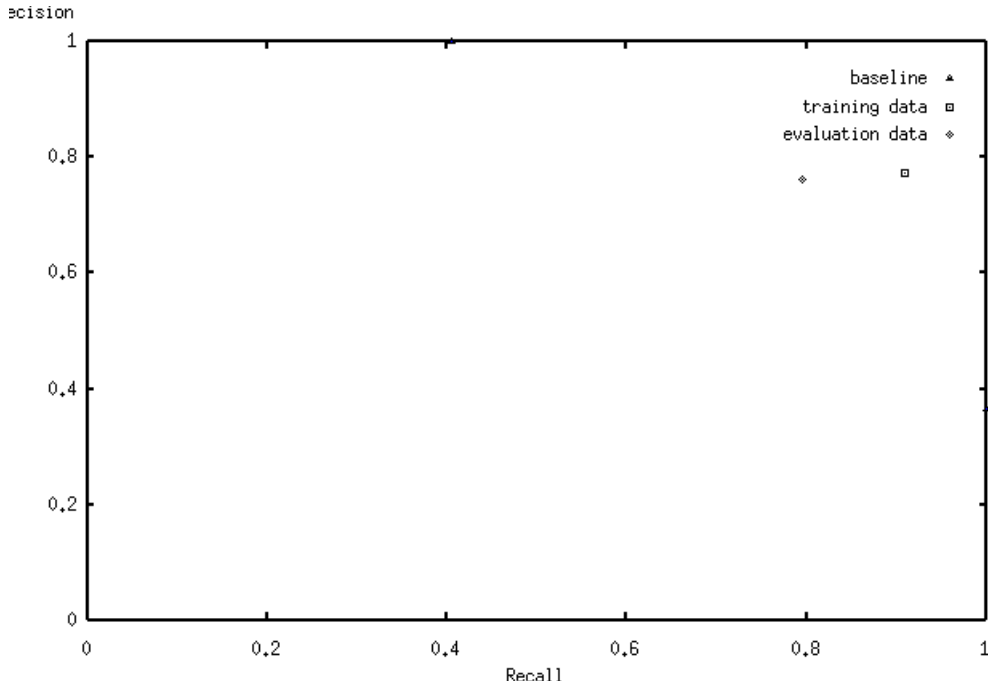

図 6.1: 参照箇所抽出の評価

照応詞の問題に関して。cite コマンドの前後の文で this.cue に含まれる語が出現した場合、その前文に先行詞があると過程してルールを作成した。しかし、[Paice90] でも指摘されているように、先行詞は前文ばかりでなく、後方照応や同一文内に先行詞がある場合、あるいは先行詞が複数文や前の段落全体にわたるケースもある。[Paice90] では、この問題に対して多くのルールを作成して照応解析を試みているが十分な精度が得られていない。本研究では、参照箇所というテキスト中で極めて限定された箇所を文抽出の対象としているため、照応問題について [Paice90] ほど一般的に処理する必要はないものと考えられる。したがって、今回作成したルールのように「this.cue の先行詞は前文である」という単純な仮定でも、比較的良い抽出精度が得られたものと考えられる。

6.2 参照タイプ決定

本節では、参照タイプ決定ルールの coverage について述べる。実験で用いた参照箇所コーパスのうち 58.5%の参照タイプが決定可能となった。しかし、残りのものについては決定できず、現段階では要約生成処理の対象から外れる。ある分野の要約を作る時に、そ

の分野の中心的な論文が存在する。 中心的」の定義は色々あると思われるが、定義のひとつに 参照される回数の多いものがその分野の中心的な論文である」という考え方がある。[神門 91] では、論文間の引用関係 (参照関係) に着目した情報検索の分野の技術論文の動向調査を行っている。本研究において、参照関係の多いその分野で中心的と考えられる論文が、参照タイプ決定ルールを用いてタイプ決定できなかった場合その論文が生成される要約からもれてしまい、ある特定分野の要約としてはあまりふさわしくないと考えられる。本研究での要約対象論文は、現在は e-Print archive 上のものだけであるが、将来的には他のデータソース[1]からの情報収集ということも考えている。coverage の低さは、論文ソースの拡大によってある程度まではカバーできるが、重要な論文が要約対象に成りうるかどうかという点で現在の手法を検討していく必要があるものと考えられる。また後でも述べるが要約の評価で、要約対象となる論文の種類 (品質) という点についても今後考えていく必要がある。

6.3 要約生成

色々な target paper について要約を出力させた所、いくつかの問題点が明らかになった。そのひとつは、ある論文中で複数回参照されている場合である。今回用いた手法では、関連ある文章の断片を集めて並べているだけなので、要約全体として一貫性に欠けるのはある程度仕方がないと思われる。しかし、ある論文中で複数回参照されている場合、同じような言い回しで何度も参照されている場合があり冗長な箇所が出現するため、要約としてあまり適切であるとは言えない。この点について、いくつかの対処方法が考えられる。例えば、複数回参照されている箇所が、論文のどういった section であるのかに着目するという手法である。例えば introduction で参照してある場合では、論文間の関係について参照箇所に記述されている可能性が高いし、逆に experiment や related work といった章での参照では参照箇所に、前者の場合と比較して論文間の関係について記述されている可能性は低い。そこで、section と参照箇所の記述内容の関係について調査し、複数の参照箇所から適当な参照箇所を選択するという方法を検討していく予定である。

[1] 例えば Unified Computer Science TR Index(Indiana university) (http://www.cs.indiana.edu:800/cstr/)

また、他の問題点として照応問題がある。先程、 参照箇所抽出の考察」でも述べたが、参照箇所中の照応詞は前文に先行詞があると考えている。したがって、そうでない場合についても対処する必要がある。例えば、参照箇所中でしばしば” In this paper” や”This paper” という表現が出てくる。この場合の” This” の先行詞はその論文全体であるが、こういった定型表現は先行詞が自明であるため、要約作成の際”In this paper” は削除、“This paper” は”They” と置き換えても、要約文書としてある程度意味は通じる。ただし、このように先行詞が自明であるケースは稀であり、その他のケースについても意味を変えない程度に何らかの処理を施す必要がある。

6.4 要約文書の評価方法について

要約生成モジュールで生成された要約の定量的な評価は難しい。評価すべき点は 2 点あると考えられる。ひとつは要約に必要な情報が生成された文章に含まれているかどうか。もうひとつは、生成された文章が自然なものであるかどうか。評価として難しい点は要約は必ず正解がひとつに限られるわけではない。どの点にポイントを置くかによって、人間の作成する要約は非常に異なってくる場合もある。したがって、要約作成のポイントのひとつとして、どういった観点で要約を作成したのか明らかにしておくことは価値あることだと考えられる。文章の自然さについては、特に生成された文章が英語である場合、native speaker でなければなかなか評価が難しく、また、non native speaker が評価した場合は、それほど信頼がおけるものでもないと考えられる。

第 7 章

まとめ

本研究では、複数のある特定の分野の論文をまとめてひとつの要約を生成するというタスクにおいて、論文間の参照関係に注目し、参照情報を利用することで要約生成の可能性を示した。その手順として、まず論文中で他の論文を参照している箇所 (参照箇所) の自動抽出ということを行った。この際、文間の結束性に着目した。次に cue word の並びを考慮してどのような目的で論文を参照しているのか (参照タイプ) を決定するルールを 12 種類作成した。このルールにより参照タイプを 80%近い精度で決定することが可能になった。

第 8 章

今後の課題

3 章で、参照情報を利用した要約生成の手法を説明した。3 章で用いたモデルでは、1 つの論文が 2 つの論文を参照する、という非常に小規模な参照関係であったが、ここでは大規模な参照関係の場合の要約生成方法について言及する。その際、参照箇所が増えると要約生成の際、どのような問題点を考慮しなければならないかを明らかにする。また、本手法の学術論文以外の適用についても簡単に触れる。

8.1 サーベイ自動生成実現に向けて

本研究の最終的な目標は、論文間の参照関係を網羅し、ある分野の一種のサーベイを自動的に生成することにある。サーベイ生成に用いる基本的な技術は 3 章で述べたごく小規模な参照関係にある論文の要約を生成する時に用いる手法である。しかしこの手法だけでは十分であるとは言えない。本節では、サーベイ作成の際考慮すべき問題点を明らかにする。

図 8.1は、3、4 章で例に挙げた [Murata93][Ikehara95] を参照している論文との関係を示したものである。3 章の例では、[Murata93][Ikehara95] を参照する論文として [Bond96] を例に挙げて説明した。しかし、e-Print archive 上の論文で、実際に先の 2 本を参照している論文は 6 本ある。参照情報を利用して要約する際、大きく 2 つの場合を考えなければならない。

1. 被要約論文を共引用する論文が複数存在する場合 (この場合は [Bond96]、[Bond96a]、[Bond94])、どの参照情報を利用するのか。

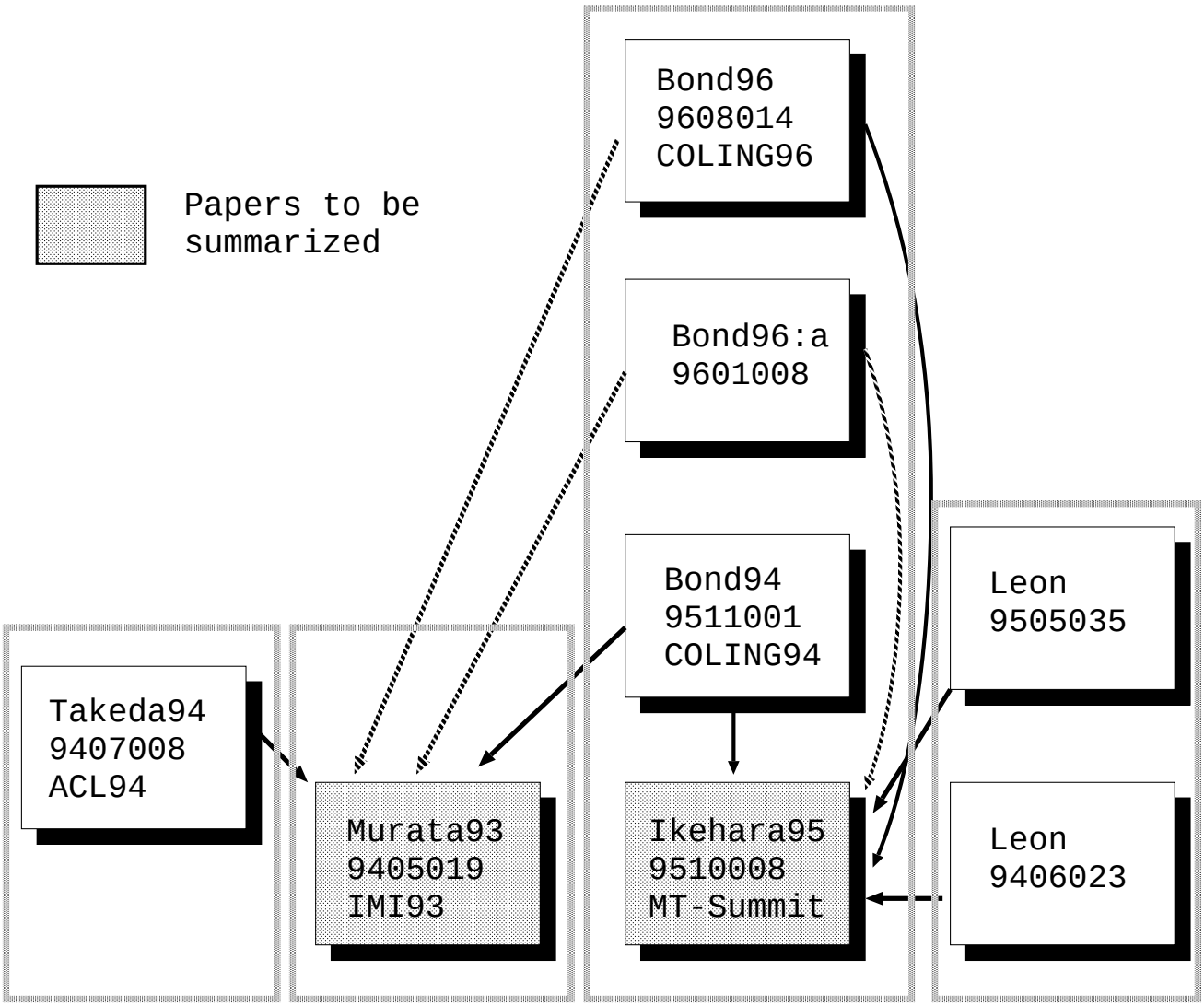

図 8.1: 論文間の参照関係の一例 (その 2)

2. 複数の被要約論文の中でひとつだけ参照しているもの ([Takeda94][Leon94][Leon95]) を要約生成の際利用するかどうか

前者の問題点で、ひとつの対処として共引用している複数の論文の中で最も新しい論文の参照情報を用いるという方法である。これは、論文の比較を行うならば、より新しい観点で行う方が良いという考え方である。特に、図 8.1のように同じ研究チームが別の論文で共引用を行う場合に有効であると思われる。後者の問題に関して、被要約論文をひとつだけ参照している論文から得られる情報についてはユーザの要求する要約の長さに応じて使いわける、という方法が考えられる。例えば、ユーザが要約文書としてなるべく詳

しく述べてあるものを要求する場合、参照情報を要約に組み入れ、ユーザがなるべく短い要約を要求する場合は、要約に組み入れないという方法である。このような手法により、ユーザの要求にある程度対応可能な要約文書の生成が可能になると考えられる。

8.2 本手法の学術論文以外の適用

本研究では e-Print archive という小規模なデータベースを用いているが、今後は Web から自動的に論文ファイルを収集して[1]、さらなるデータの充実をはかる予定である。

学術論文は参照関係という点でみればハイパーテキスト構造をしている。そこで、今後学術論文以外のハイパーテキスト構造への本手法の適用についても検討していく予定である。

[1] 例えば http://www.cs.indiana.edu/ucstri/sitelist.html

謝辞

本研究を進めるにあたり、終始熱心な御指導を賜わりました奥村学助教授に心から感謝致します。

中間審査などの折には、諸先生方から貴重な御意見を頂きました。感謝致します。

参照関係を用いた論文検索システム PRESRI の一般公開を快く承諾して下さった e-Print archive administrator の方々に感謝致します。

さらに、貴重な御意見、討論をしていただいた島津明教授、自然言語処理学講座 Thanaruk Theeramunkong 助手、ならびに研究室の皆様に感謝致します。

最後に、多くの方々の御援助によって本研究を行うことができましたことを厚く御礼申し上げます。

参考文献

[Kupiec95] Julian Kupiec , Jan Pedersen , Francine Chen. “A Trainable Document Summarizer”. SIGIR’95. pp. 68–73. 1995.

[Mani97] Inderjeet Mani , Eric Bloedorn. “Multi-document Summarization by Graph Search and Matching”. AAAI’97. pp. 622–628. 1997.

[McKeown95] Kathleen McKeown and Dragomir R. Radev. “Generationg Summaries of Multiple News Articles”. SIGIR’95. pp. 74–81. 1995.

[McKeown96] Jacques Robin, Kathleen McKeown. “Empirically designing and evaluation a new revision-based model for summary generation”. Artificial Intelligence 85(1996)

[Paice90] Chris D. Paice “Constructing Literature Abstracts by Computer: Techniques And Prospects”. Information Processing & Management. Vol.26 No.1, pp. 171–186. 1990.

[Teufel97] Simone Teufel, Marc Moens. “Sentence extraction as a classification task”. ACL’97 Summarization workshop.1997.

[Watanabe96] Hideo Watanabe. “A Method for Abstracting Newspaper Articles by Using Surface Clues”. COLING’96. pp974–979.1996

[Yamamoto95] Kazuhito Yamamoto, Shigeru Masuyama, Shozo Naito. An Empirical. “Study on Summarizing Multiple Texts of Japanese Newspaper Article”. NLPRS’95.pp. 461–466. 1995.

[神門 91] 神門典子, 野末道子, 榛田倫子, 村上匡人, 谷津真理子, 上田修一. “情報検索分野の構造 :引用調査による下位領域の発展過程の分析”. Library and Information Science No.29. 1991.

[齊藤 93] 斎藤 陽子. “引用文献の記述形式の実態と基準”. 書誌索引展望. 第 17 巻. 第 4 号. 1993.

[佐藤 96] 佐藤理史, 佐藤円. “ネットニュースグループ fj.wanted のダイジェスト自動生成”. 自然言語処理, Vol. 3, No. 2, pp19-32. 1996.

[柴田 97] 柴田昇吾, 上田隆也, 池田裕治. “複数文章の融合”. 情処研報 NL120-12. 1997.

[船坂 96] 船坂貴浩, 山本和英, 増山繁. “冗長度削減による関連新聞記事の要約”. 情処研報 NL114-7. 1996.

要約生成の手法の説明に用いた論文

[Brill94] E. Brill. “Some advances in transformation-based part-of-speech tagging”. In Proceedings of the AAAI'94. 1994. (http://xxx.lanl.gov/ps/cmp-lg/9406010)

[Lee95] Geunbae Lee, Jong-Hyeok Lee, Sanghyun Shin. “TAKTAG: Two-phase learning method for hybrid”. statistical/rule-based part-of-speech disambiguation”. (http://xxx.lanl.gov/ps/cmp-lg/9504023).

[Ramshaw3] Lance A. Ramshaw (Bowdoin College) and Mitchell P. Marcus (University of Pennsylvania). Text Chunking using Transformation-Based Learning. 13 pages, LaTeX2e, 1 included figure. Journal-ref: ACL Third Workshop on Very Large Corpora, June 1995, pp. 82-94. (http://xxx.lanl.gov/ps/cmp-lg/9505040).

[Ueberla5] J.P. Ueberla and I.R. Gransden. Clustered Language Models with Context-Equivalent States. 3 pages, latex. (http://xxx.lanl.gov/ps/cmp-lg/9606002).

[Light6] Marc Light (University of Tuebingen) Morphological Cues for Lexical Semantics. Journal-ref: Proceedings of the 34th Meeting of the Association for Computational Linguistics (ACL'96). (http://xxx.lanl.gov/ps/cmp-lg/9606003).

[Bond96] Bond, Francis, Kentaro Ogura, and Satoru Ikehara. 1996. Classifiers in Japanese-to-English Machine Translation In *Proceedings of the 16kth International Conference on Computational Linguistics (COLING '96)*, August. ((http://xxx.lanl.gov/ps/cmp-lg/9608014).

[Bond96a] Francis Bond (NTT), Kentaro Ogura (NTT), Tsukasa Kawaoka (Doshisha University). 1996. "Noun Phrase Reference in Japanese-to-English Machine Translation ". (http://xxx.lanl.gov/ps/cmp-lg/9601008)

[Bond94] Francis Bond (NTT), Kentaro Ogura (NTT), Satoru Ikehara (NTT). 1994. "Countability and Number in Japanese-to-English Machine Translation". Proceedings of the 15th International Conference on Computational Linguistics (COLING'94), pp 32–38.(http://xxx.lanl.gov/ps/cmp-lg/9511001)

[Ikehara95] Satoru Ikehara (NTT), Satoshi Shirai (NTT), Akio Yokoo (NTT), Hiromi Nakaiwa (NTT). 1995. "Toward an MT System without Pre-Editing — Effects of New Methods in ALT-J/E —". (http://xxx.lanl.gov/ps/cmp-lg/9510008)

[Murata93] Murata, Masaki and Makoto Nagao. 1993. Determination of referential property and number of nouns in Japanese sentences for machine translation into English. In Proceedings of the Fifth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI '93), pages 218–25, July.(http://xxx.lanl.gov/ps/cmp-lg/9405019)

[Leon94] Fernando Sánchez León (Laboratorio de Lingüística. "A Spanish Tagset for the CRATER Project". (http://xxx.lanl.gov/ps/cmp-lg/9406023)

[Leon95] Fernando Sánchez León (Laboratorio de Lingüística "Development of a Spanish Version of the Xerox Tagger". (http://xxx.lanl.gov/ps/cmp-lg/9505035).

[Takeda94] Koichi Takeda (IBM Research, Tokyo Research Lab.). 1994. "Tricolor DAGs for Machine Translation". ACL94.(http://xxx.lanl.gov/ps/cmp-lg/9407008)

APPENDIX A

The summary about [Brill1]

– produced by ASG Ver. 1.0 –

98/1/13 This is the summary of 4 papers, 3 of which have reference relation to [Brill1].

1. Central Topic of This Summary

The central topic of this summary is as follows:

[Brill1]

Most recent research in train-able part of speech taggers has explored stochastic tagging. While these taggers obtain high accuracy, linguistic information is captured indirectly, typically in tens of thousands of lexical and contextual probabilities. In [Brill92], a train-able rule-based tagger was described that obtained performance comparable to that of stochastic taggers, but captured relevant linguistic information in a small number of simple non-stochastic rules. In [Brill1], [Brill1] describe a number of extensions to this rule-based tagger. First, [Brill1] describe a method for expressing lexical relations in tagging that stochastic taggers are currently unable to express. Next, [Brill1] show a rule-based approach to tagging unknown words. Finally, [Brill1] show how the tagger can be extended into a k-best tagger, where multiple tags can be assigned to words in some cases of uncertainty. .

2. Availability of [Brill1]

In this section, we show the availability of [Brill1].

citation [Ueberla5] 9606002 → 9406010

Due to limitations of [Ueberla5]'s software, each word could belong to one part of speech only. Brill's rule based tagger [Bri94b] was therefore employed to assign the most likely tag t to each word in the official 20K vocabulary used in the language modeling experiments.

This resulted in 61 different tags. Table gives the results for various models using this part of speech information.

citation [Light6] 9606003 → 9406010

Only its words and part-of-speech tags were utilized. Although these tags were corrected by hand, part-of-speech tagging can be automatically performed with an error rate of 3 to 4 percent [merialdo_94,brill_94].

3. Problem of [Brill1]

In this section, we show the problems of [Brill1] which were pointed out from other wokers.

citation [Lee2] 9504023 → 9406010

Recently, rule-based approaches are re-studied to cope with the limitations of statistical approaches by learning the tagging rules automatically from the corpus [brill:simple,brill:some]. Some systems even perform the POS tagging as part of syntactic analysis process [voutilainen:syntax]. However, the rule-based approaches alone are in general not robust to handle the unknown words, and is not flexible to adjust to the new tag-sets and languages.

citation [Ramshaw3] 9505040 → 9406010

Most efforts at superficially extracting segments from sentences have focused on identifying low-level noun groups, either using hand-built grammars and finite state techniques or using statistical models like HMMs trained from corpora. In [Ramshaw3], [Ramshaw3] target a somewhat higher level of chunk structure using Brill's [Brill93] transformation-based learning mechanism, in which a sequence of transformational rules is learned from a corpus; this sequence iteratively improves upon a baseline model for some interpretive feature of the text. This technique has previously been used not only for part-of-speech tagging [Brill94b], but also for prepositional phrase attachment disambiguation [BrillResnik94], and assigning unlabeled binary-branching tree structure to sentences [Brill93a]. Because transformation-based learning uses pattern-action rules based on selected features of the local context, it is helpful for the values being predicted to also be encoded locally.

[References]

[Brill1]

Some Advances in Transformation-Based Part of Speech Tagging. Eric Brill (MIT) 6 Pages. To appear in Proceedings of AAAI94. Code available. cmp-lg/9406010

[Lee2]

TAKTAG: Two-phase learning method for hybrid statistical/rule-based part-of-speech disambiguation Geunbae Lee, Jong-Hyeok Lee, Sanghyun Shin (Department of Computer Science & Engineering and Postech Information Research Laboratory, Pohang University of Science & Technology) 10pages, latex, named.sty & named.bst, use psfig figures, submitted cmp-lg/9504023

[Ramshaw3]

Text Chunking using Transformation-Based Learning Lance A. Ramshaw (Bowdoin College) and Mitchell P. Marcus (University of Pennsylvania) 13 pages, LaTeX2e, 1 included figure Journal-ref: ACL Third Workshop on Very Large Corpora, June 1995, pp. 82-94 cmp-lg/9505040

[Mikheev4]

Unsupervised Learning of Word-Category Guessing Rules Andrei Mikheev (HCRC, Edinburgh University) 8 pages, LaTeX (aclap.sty for ACL-96); Proceedings of ACL-96 Santa Cruz, USA; also see cmp-lg/9604025 Journal-ref: – cmp-lg/9604022

[Ueberla5]

Clustered Language Models with Context-Equivalent States J.P. Ueberla and I.R. Gransden 3 pages, latex cmp-lg/9606002

[Light6]

Morphological Cues for Lexical Semantics Marc Light (University of Tuebingen) Journal-ref: Proceedings of the 34th Meeting of the Association for Computational Linguistics (ACL'96) cmp-lg/9606003