応用ソフトウェアを対象とした操作手順のアウトライン生成

島崎 聡　　藤井 敦
東京工業大学大学院情報理工学研究科計算工学専攻

1 はじめに

わかりやすい文章を作成する方法の一つに，文章を話題ごとに分割して見出しを付ける事で，文章の骨格や構造を作る方法がある．この骨格や構造をアウトラインと呼ぶ．例えば本論文では,「1. はじめに 2. 関連研究 3. 提案手法 4. 実験 5. おわりに」という章立てになっており，この構造がアウトラインにあたる．アウトラインがあると，読む前に文章の大枠を把握する事や，文章の一部分だけを選んで読み返す事ができるようになり，文章の内容をより容易に理解する事が可能となる．

しかし，現存する全ての文章においてアウトラインが生成されている訳ではない．例えば，手順テキストを考える．手順テキストとは，何らかの目的を達成するための手順が書かれたテキストであり，web ページやマニュアルといった形で大量に存在し，現在も増え続けている．これらの手順テキストにもアウトラインが生成されていないものは多数含まれている．そこで本研究では，アウトラインが生成されていない手順テキストから，自動アウトラインを生成する事を目的とする．

ここで，アウトラインを生成するためには，段落を分ける基準を規定する必要がある．一般的なテキストでは話題の境界で段落を分割するとされている．では，何らかの目的を達成するための操作系列が書かれている手順テキストにおける話題とは何なのかを考える．

手順テキスト全体の目的は，いくつかの部分目的に細分化できる．さらに，その部分目的も再帰的に細分化する事が可能であり，最終的には最小単位である１つ１つの操作にまで細分化する事ができる．手順テキストでは，この部分目的が話題に相当し，部分目的が変化する箇所で段落を分割すれば良いと考えられる．しかし，部分目的には様々な粒度のものがあり，どの部分目的を話題とするのかを選択する基準が必要になる．

そのため，手順テキストのアウトラインを生成するためには，どの部分目的を話題にするかを定める事のできる知識が必要だといえる．そして，手順テキストの分野によって必要になる知識が異なるため，その分野に応じた知識が必要になる．

ここで，応用ソフトウェアの操作手順テキストを考える．応用ソフトウェアでは，ある目的をソフトウェアの機能をいくつか使用する事で達成する．そして，それぞれの機能はいくつかの操作を行う事で発揮される．そのため，ソフトウェアの機能が部分目的に該当し，話題として扱う事ができる．例えば，Microsoft Word の操作手順テキスト中に,「ファイル」,「名前をつけて保存」,「保存」の順でクリックする，という操作があった時，これら 3 つの操作は「ファイルを保存する」という部分目的を持っている．

本研究では，対象を応用ソフトウェアの操作手順テキストに限定し，自動で手順テキストのアウトラインを生成する手法を提案して，実験を行う事で性能を評価する．

2 関連研究

浜田ら [1] や苅米ら [3] は料理レシピから動作を抽出し，動作間の繋がりを解析して料理手順のフローグラフを作成する事で，手順テキストの構造を解析した．

Nakao[4] は要約対象の文書を話題ごとに分割し，各話題に対して要約を作成する事で，自動要約を行った．この操作はアウトライン生成とほぼ同じ処理である．しかし，これらの手法は新聞記事や報告書を対象としており，手順テキストに適用するのは適切ではない．

西原ら [5] は，料理レシピを対象とし，各段落から教師あり機械学習を用いて見出し語句を抽出することで，アウトラインを生成した．彼らの研究は本研究と同様に，段落に対して見出しを付ける事でアウトラインを生成している．しかし，段落はレシピの著者が作成した段落を用いている．段落が作成される基準は人によって異なっており，操作内容に即した基準によって作成されているとは限らない．例えば，レシピの内容に即した基準で作成された段落の他に，文字数を基準として全ての段落の長さがある程度均等になるように作成された段落，1 つの段落が 1 つの行動のみを含むように作成された段落等がある．

3 提案手法

3.1 概要

本研究では以下の 2 段階の操作によって，アウトラインを生成する．

1. 操作内容に即した基準で段落を生成する
2. 各段落の内容を端的に示す見出しを生成する

1 章で説明したように，応用ソフトウェアはいくつかの機能を持っており，人間が操作する事でそれらの機能が発揮される．つまり，ユーザーがソフトウェアに対して行う操作 1 つ 1 つには，それぞれ目的となるソフトウェアの機能が存在する．

本研究では，その機能を手順テキストの話題にあたる部分目的と考え，同じ機能を目的とする操作系列が書かれている部分をまとめることで段落を生成し，その段落に対して，目的となる機能を示すフレーズを添付する事で見出しを生成する．本研究では今後，この考えに基づく段落を「機能段落」，見出しを「機能名」と呼称する．

本章では，この考えに基づくアウトライン生成の方法について説明する．

3.2 メニュー項目

本研究では，1 つ 1 つの操作と目的となる機能の対応付けを行うために，メニュー項目の情報を使用する．

メニュー項目とは，応用ソフトウェアの GUI 上に出現する項目の事である．メニュー項目は，ボタンやテキストボックスなど様々な形を取る事が可能であり，ダイアログボックスのタイトルやタブ名などの形で出現して，画面遷移の状態を指し示す事もある．しかし，どんな形であっても，木構造に近い階層構造を持っており，応用ソフトウェアに対する操作と指示の殆どは GUI 上のメニュー項目のいずれかに対して行われるという特徴がある．実際に Microsoft Word 2010 のメニュー項目構造の一部を図 1 に示す．

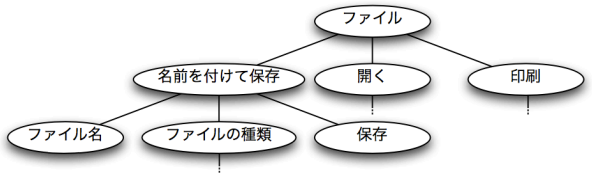

図 1: メニュー項目の木構造

また，1 つ 1 つのメニュー項目は様々な情報を保持しており，本研究では，状態遷移先，相反する状態の 3 つの情報を利用している．わかり辛い項目について補足すると，状態遷移先はその項目を操作すると状態遷移が起きる時，その遷移先の状態を指し示すメニュー項目が入る．また，相反する状態は，その項目が状態を指し示しており，かつ，その状態に遷移する事で，ある状態ではなくなる時に，その解除される状態を指し示すメニュー項目が入る．

本研究では，メニュー項目の情報を事前に作成しておき，応用ソフトウェアに対する操作とメニュー項目の対応付けを行い，その後，対応づいたメニュー項目の構造や，1 つ 1 つの項目が持つ情報を使用してアウトライン生成を行う．なお，メニュー項目の情報は，応用ソフトウェアが必然的に持っている情報であるため，知的作業を行う事無く抽出して利用できる．

3.3 機能段落生成

応用ソフトウェアに対する操作は，まず木構造上で浅いレベルのメニュー項目を操作し，徐々に深いレベルのメニュー項目を操作していき，最終的に何らかの機能を発揮する項目を操作する．という手順を繰り返す事で行われる．そのため，浅いレベルから深いレベルのメニュー項目へ順番に行われる操作を，目的となる機能が同一の操作系列としてまとめ，逆に操作対象が深いレベルから浅いレベルに変化する時点で段落を分割する事で，同一の機能を目的とする機能段落を生成する．

例えば,『「ファイル」「名前を付けて保存」「保存」をクリックする.』という文章があった時，図 1 のメニュー項目構造によると，これは浅いレベルから深いレベルに順番に操作しているため，1 つの機能段落になる．そして,「保存」をクリックすると，ソフトウェアはファイルを保存する機能を発揮するため，機能名は「ファイルを保存する」となる．

2 章で説明したように，手順テキストの著者によって生成された段落は内容に沿った物とは限らない．しかし，関連の無いテキストが同じ段落に含まれている可能性は低いと考え，本研究では手順テキストの著者による段落の境界から機能段落の境界として適切な物を選択する事で，機能段落を生成する．

機能段落生成は，以下の 3 段階で行う．

1. **メニュー項目候補語句抽出**

   操作手順テキストから，メニュー項目を指し示している可能性が高い文字列を抽出する．ソフトウェアの操作手順テキストにおいて，多くの場合メニュー項目を示す文字列は何らかの方法で強調されている．そのため本研究では，操作手順テキストから，太字になっているテキスト，ダブルクォーテーションで囲まれたテキスト，単語の頭文字が大文字になっているテキストを，メニュー項目候補語句として抽出する．

2. **メニュー項目識別**

   抽出したメニュー項目候補語句をメニュー項目と対応付ける．まず，ソフトウェアを開いた直後の状態を現在の状態として，最初のメニュー項目候補語句を見る．この候補語句のテキストと一致するラベルを持っており，かつ今の状態で操作する事ができるメニュー項目を検索し，該当するメニュー項目が存在したら，その候補語句とメニュー項目とを対応付ける．その後，そのメニュー項目の「遷移先状態」を現在の状態に追加し,「相反する状態」を現在の状態から取り除く．という操作を繰り返して，候補語句とメニュー項目とを対応付けていく．

   ただし，メニュー項目候補語句と一致するラベルを持つメニュー項目が複数見つかった場合は，その後で出現する候補語句を見ていき，最も先まで対応付けが可能になるようなメニュー項目を選択する．

   逆に，メニュー項目候補語句と一致するラベルを持つメニュー項目がその状態で操作できる中に存在しなかった場合は，操作手順テキストから一部の手順が省略されている可能性がある．省略は，人間が見れば，次に行う操作内容が自明である時に行われるため，対応付け可能なメニュー項目が発見できず，かつ現在の状態から遷移する事が可能な状態が 1 つしか無い場合は，その状態に遷移したものとして対応付けを行う．もし，この補完を行ってもなお候補語句と対応付けられるメニュー項目が見つからない

場合は，状態を無視して全てのメニュー項目から候補語句と一致するラベルを持つメニュー項目を検索して対応付けを行い，それでも対応するメニュー項目が発見できない場合は，その候補語句は誤抽出された語でありメニュー項目では無かったとして候補語句から除外し，状態を変化させずに次の候補語句の対応付けを開始する．

3. **機能完了箇所分割**

何らかの機能が使用された直後で機能段落を分割する．まず，章の初めで説明したように，手順テキストの著者による段落の境界を機能段落境界の候補とし，そこから，直後の段落にメニュー項目と対応付けられたテキストが含まれて無い物を除外する．

残った機能段落境界候補を見て，直前に出現するメニュー項目の状態遷移先がその項目の先祖である場合，あるいは境界候補の直後に出現するメニュー項目がソフトウェアを起動した直後の状態で操作可能である場合に，その境界候補を機能段落の境界とし，機能段落を生成する．

実際の手順テキストでは必要な操作を順番が書かれた文章だけでなく,「これから〜をします．具体手には A して B して...」のように，前後で行う操作内容を改めて宣言する文章や,「必要に応じて〜する」のように任意で内容を実行する文章も存在する．本手法はこれらのパターンでも問題無く処理が可能である．

前後の操作を改めて宣言する文章では，前後の操作と同じメニュー項目を操作するため，宣言する文章は対応する前後の操作と同じ段落に含める事ができる．任意で実行する操作の文章では，その任意の操作が前後と同じ目的を持っていたら同じ機能段落に含め，異なる目的を持っていたら分割すれば良いだけなので，通常の処理と同様に扱う事ができる．

3.4 機能名生成

多くのメニュー項目には文字列でラベルが付属しており，このラベルはそのメニュー項目の機能や目的を端的に表している．これを利用し，機能段落に含まれるメニュー項目のラベルから適切な物を抽出することで機能名を生成する．

機能名生成は以下の 2 段階で行う．

1. **代表語句抽出**

メニュー項目のラベルから，機能段落の目的である機能を最も適切に表すラベルを選択して代表語句として抽出する．

メニュー項目は，浅いレベルの項目ほど機能を大まかに示し，深いレベルの項目ほど機能を詳細に示す．そのため，機能段落に含まれるメニュー項目のうち，最も深いレベルの項目のラベルがその機段落能で使用する機能を最も詳細に示す項目であると考えられる．これより，最も深いレベルの項目が 1 つに特定できる場合は，その項目のラベルを代表語句として抽出する．また，最も深いレベルの項目が複数ある場合，その機能段落ではそれらの機能全てを使用していると考えられるため，それらの機能全体を示す項目として，最も深いレベルの項目全てに共通する先祖を集め，その中で最も深いレベルのメニュー項目のラベルを代表語句として抽出する．

この時点で，使用した機能に最も近い語句は抽出された．しかし，この語句は動詞や名詞，文など様々な形を取りうる．ここで，目的である機能を 1 語で表す事ができるとは限らない事を考えると，より表現力の高い述語と項の組の形に変換すべきだと考えられる．そのため，次の段階でこの代表語句を補完して述語と項の組の形に変換する．

2. **代表語句補完**

抽出された代表語句を述語と項の組の形にするため，代表語句の種類に応じて，表 1 の様な処理を行う．

表 1: メニュー項目が持つ情報

条件		補完方法
代表語句が述語と項を含む		補完しない
代表語句が動詞	代表語句の先祖に名詞がある	述語を代表語句，目的語を代表語句の先祖のうち最も深いレベルの名詞とする
	代表語句の先祖に名詞が無い	補完しない
代表語句が名詞	代表語句の先祖に動詞がある	代表語句の先祖のうち最も深いレベルの動詞を述語，代表語句を目的語とする
	代表語句の先祖に動詞が無い	述語を「設定する」という意味の動詞，目的語を代表語句とする

代表語句として動詞が抽出されて，その先祖に名詞が出現しない場合は，人間から見れば動詞のラベルだけで使用する機能が理解できるという事なので，名詞の補完は行わない．

また，代表語句として名詞が抽出され，その先祖に動詞が存在しない場合は，名詞だけで使用する機能が理解できるという事である．その場合には，その名詞が設定項目の名前や設定値を表していると考えられ，その設定項目を編集するか，その設定値を適用する機能である可能性が高い．そのため，述語として「設定する」という意味の述語で補完を行う．

4 実験

4.1 実験方法

Word2010 の操作手順テキストに対して本手法を適用して評価実験を行った．評価に使用する手順テキストは，200 万以上の英語の手順テキストを掲載している ehow[1] という WEB サイトから収集した．ここの tech カテゴリで「Word 2010」というクエリで検索し，検索結果上位 881 を収集した．881 件の内，人手で確認して本当に Word2010 に対する操作手順が書いていると確認で

[1] http://www.ehow.com

きたテキスト上位 10 件 (機能段落 23 件を持つ) に対して，正解データをアノテーションして評価実験を行った．

機能名生成の性能は機能段落生成の結果に左右されてしまうため，機能段落生成と機能名生成の評価実験はそれぞれ独立して行った．

4.2 機能段落生成の評価実験

機能段落生成は，メニュー項目候補語句抽出，メニュー項目識別，機能完了箇所分割の 3 段階で行われており，より性能を詳細に把握するために，3 段階それぞれが完了した時点での性能評価を行った．

メニュー項目候補語句抽出ではメニュー項目を示しているテキストを正解として，メニュー項目識別では手順テキスト中に出現するメニュー項目の ID を正解として，機能完了箇所分割では，正しい段落の境界を正解として，どの程度正解が得られたのかを，精度，再現率，F 値によって評価した．

また，機能段落生成が完了した時点である機能完了箇所分割の結果に対して，ベースラインを作成して比較を行った．本研究ではベースラインとして，全段落境界候補での分割，一切分割無し，TextTiling[2] の 3 つのメニュー項目を使用しない手法と，メニュー項目の lexical chain を使用した．メニュー項目の lexical chain とは，段落境界候補のうち，直前と直後に出現するメニュー項目の組が，評価実験に使用していない 871 件の手順テキストにおいて同一段落内で隣接した事が無い場合に分割する手法である．なお，TextTiling の Pseudosentence size と Window width のパラメータは，実験の結果最も性能が高くなる値 (両方共に 1) にして適用した．

4.3 機能名生成

機能名生成も，代表語句抽出と代表語句補完の 2 段階で行われており，それぞれの段階が完了した時点での性能評価を行った．この時，機能名を添付する機能段落は正解データを使用して評価を行った．

各段階において出力された語句あるいは述語と項の組に対して，機能段落の内容に合っているかを，人手で [成功 or 失敗] の二値で評価を行った．

また，機能名生成でも，ベースラインを作成して比較を行った．ベースラインとしては，各機能段落に出現する単語に対して TF・IDF の値を計算して，TF・IDF 最も高い単語をその段落の見出しとする手法を使用した．

4.4 結果と考察

機能段落生成評価実験の結果を表 2 に，機能名生成評価実験の結果を表 3 に示す．

本手法は機能段落生成でも機能名生成でもベースラインよりも高い性能を示していた．

機能段落生成ではメニュー項目の lexical chain の性能が低い事から，ただメニュー項目の字面を利用するだけでは性能は伸びず，メニュー項目の構造や遷移の情報を利用して初めて性能が向上すると考えられる．

また機能名生成では，代表語句抽出の段階ではベースラインより性能が低く，代表語句を補完して述語と項の組にすることで性能がベースラインを超えた．つまり，メニュー項目を利用しても代表語句抽出の性能が向上することはなかった．しかし，メニュー項目を使用して抽出された代表語句は補完可能であり，補完を行う事で最終的な性能を向上させる事ができた．

表 2: 機能段落生成の評価結果

手法	精度	再現率	F 値
提案手法 (メニュー項目候補語句抽出まで)	0.95	0.94	0.94
提案手法 (メニュー項目識別まで)	0.50	0.47	0.48
提案手法 (機能完了箇所分割まで)	0.75	0.72	0.63
全候補で分割	0.30	0.6	0.36
分割無し	0.40	0.40	0.40
TextTiling	0.50	0.50	0.50
メニュー項目の lexical chain	0.26	0.38	0.28

表 3: 機能名生成の評価結果

手法	成功率
提案手法 (代表語句抽出まで)	0.26(6/23)
提案手法 (代表語句補完まで)	0.52(12/23)
TF・IDF	0.30(7/23)

5 おわりに

本研究では，メニュー項目の情報を知識として使用し，応用ソフトウェアの操作手順テキストのアウトライン生成を行った．その結果，段落生成では F 値で 0.63 の性能を得る事ができ，見出し生成では 52%の段落に正しい見出しを付ける事ができた．これより，分野に応じた知識を使用する事で手順テキストのアウトライン生成の性能が向上する事が示せた．

参考文献

[1] 浜田 玲子, 井出 一郎, 坂井 修一, 田中 英彦: “料理テキスト教材における調理手順の構造化”, 電子情報通信学会論文誌 D, Vol.85, No.1, 2002, pp.79—-89

[2] Marti A Hearst: “TextTiling: Segmenting text into multi-paragraph subtopic passages”, Computational linguistics, Vol.23, Issue.1, 1997, pp.33-64.

[3] 苅米 志帆乃, 藤井 敦: “料理レシピテキストの構造解析とその応用”, 言語処理学会第 18 回年次大会発表論文集, 2012, pp.893–842

[4] Yoshio Nakao: “An Algorithm for One-page Summarization of a Long Text Based on Thematic Hierarchy Detection”, In Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, 2000, pp.302—309,

[5] 西原 弘真, 苅米 志帆乃, 藤井 敦: “料理レシピを対象としたアウトライン型自動要約”, 情報処理学会研究報告 情報学基礎研究会報告, vol.8, 2013, pp.1–7