特許請求項への改行の自動挿入

Automatic Insertion of Newline Characters to Patent Claim

豊橋技術科学大学工学部 情報・知能工学系研究員 **太田　貴久**

PROFILE 2006年豊橋技術科学大学大学院博士前期課程知識情報工学専攻修了。現在、同大学研究員。修士（工学）、言語処理学会会員、日本知財学会会員。2014年度から特許版・産業日本語委員会委員。

t-ota@t-ota.com

1 はじめに

特許出願の際に提出する書類の１つである【特許請求の範囲】は、特許を受けようとする発明を特定するために必要な事項が記載され、特許権の及ぶ範囲を定める重要な書類である。【特許請求の範囲】では、発明が請求項ごとに記載されている（図１）。

> ユーザとの対話の中で、ロボットが出力する、前記ユーザの発話に対する応答としての応答文を生成する情報処理装置において、<nl>
> 前記発話から、前記対話における、前記ユーザの前記ロボットに対する状態を検出する検出手段と、<nl>
> 前記検出手段により検出された前記状態に対応する言葉遣いで、前記応答文を生成する生成手段とを備えることを特徴とする情報処理装置。

図１　特許請求項の例（<nl> は改行）

図１のように、請求項は基本的に「発明を１文で記述する」という特許独特の言語的な特徴がある。そのため、文長がきわめて長く、かつ、構文が複雑になる。特許における産業日本語では、文を短く区切ることが推奨されているが [1]、現在の請求項の性質上、文を分けることは望ましくない。弁理士によっては、これを少しでもわかりやすく（読みやすく）するために、発明の構成要素等の記述のまとまりごとに、請求項中に明示的な改行を挿入することがある。改行を挿入することで、長文の中の意味のまとまりを把握しやすくなる。しかしながら、すべての【特許請求の範囲】で改行が挿入されているわけではない。

そこで、本研究では、請求項を読みやすくするために、明示的な改行が存在しない請求項に、改行を自動的に挿入することを目的とする。本研究と同様に、特許請求項の可読性向上を目的とした研究として新森らの研究 [2] がある。新森らの研究では、請求項に頻出する特徴的な表現（「を備える、」など）をあらかじめ人手で用意する必要があるが、本研究はそれらを必要としない手法を提案する。

2 改行の直前の表現について

明示的な改行を含む請求項において、ほとんどの場合、改行は記述のまとまりごとに挿入される。ここで、「記述のまとまり」とは、発明の構成要素（図１の２行目）や公知技術（同１行目）である。これらの改行の直前に着目すると「において、」や「と、」といった表現がある。本研究では、このような改行の直前に現れる、様々な分野の特許請求項で数多く現れる表現を「デリミタ」と呼ぶ。さらに、図１の３行目の途中に存在する「とを備えることを特徴とする」のような、発明の最後の構成要素と発明全体を表す表現をつなぐ表現もデリミタとする。ここで、このデリミタを「最終デリミタ」と呼ぶ。なお、最終デリミタは、図１のように必ずしも改行されるとは限らない。また、最終デリミタ以外のデリミタを「行末デリミタ」と呼ぶ。

本研究では、明示的な改行を含む請求項について予備調査を行い、デリミタの形式的な特徴を分析した。その結果、デリミタには以下のような特徴があることがわ

かった。

●先頭形態素は、助詞・助動詞・読点のいずれかである

●末尾形態素は、動詞・助動詞・読点のいずれかであり、かつ、文節の末尾と一致する

本研究におけるデリミタは、これらの2つの条件を満たす形態素列とする。なお、「デリミタ」という言葉はSuzukiらの研究[3]でも用いられているが、本研究のデリミタとは異なるものである。

提案手法では、明示的な改行を含む請求項を用いてデリミタの特徴を学習し、その結果を用いて、明示的な改行を含まない請求項のデリミタを抽出し、改行を自動的に挿入する。以後、本稿では、明示的な改行を含む請求項を「改行形式請求項」、含まない請求項を「非改行形式請求項」と呼ぶ。

3 提案手法

提案手法では、改行形式請求項のデリミタを利用して、非改行形式請求項のデリミタを推定し、その直後に改行を挿入する。提案手法の概要を図2に示す。

図2のように、提案手法は、はじめに改行形式請求項

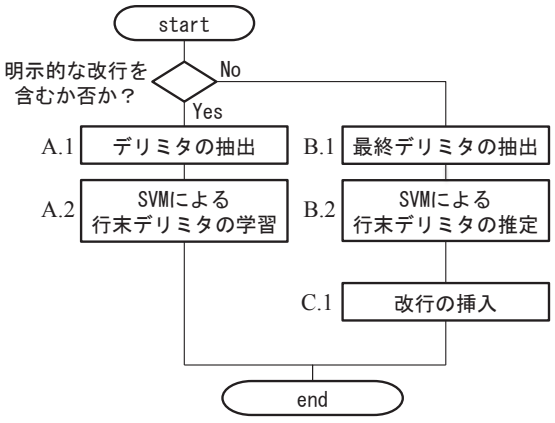

図2　改行の自動挿入手法の概要

のデリミタを抽出し、その特徴を学習する（図2のA.1とA.2）。その後、この学習結果を利用して非改行形式請求項のデリミタを推定する（図2のB.1とB.2）。最後に、非改行形式請求項に改行を挿入する（図2のC.1）。以下では、各ステップの詳細を説明する。

デリミタの抽出（A.1）

提案手法の最初のステップは、改行形式請求項から、行末デリミタと最終デリミタを抽出する。ここで、行末デリミタの抽出は、単純に先に述べたデリミタの特徴を備えた表現を抽出すればよいわけではない。例えば、改行の直前に現れる、「（構成要素）を具備し、」と「（既出の構成要素）を接続し、」という2つの表現を考える。このとき、前者の「を具備し、」はデリミタであるが、後者の「を接続し、」はデリミタではない。後者の場合、構成要素を接続すること自体が発明の特徴であるためである。この場合、デリミタは「、」（読点のみ）となる。

本研究では、情報理論におけるエントロピーを用いることで、行末からどこまでの範囲がデリミタであるかを判定する。具体的には、デリミタの条件をみたす行末に現れる表現を全て抽出し、以下に定義する表現 e の行末デリミタらしさ $D_{NL}(e)$ が最大となる表現を行末デリミタとして抽出する。

$$D_{NL}(e) = \log_2\{\mathrm{len}(e)+1\}\cdot H_L(e)$$

$$H_L(e) = -\sum_b \Pr(b \leftarrow e)\log_2\Pr(b \leftarrow e)$$

ここで、$\mathrm{len}(e)$ は表現 e の形態素数を表し、$\Pr(b \leftarrow e)$ は、改行形式請求項の行末に現れる表現 e の直前に、文節 b が出現する確率を表す。

さらに、提案手法では、最終デリミタも同様に、エントロピーを利用して抽出する。具体的には、最後の改行以降の形態素列から、以下に定義する表現 e の最終デリミタらしさ $D_{Last}(e)$ が最大となる表現を最終デリミタとして抽出する。

$$D_{Last}(e) = \frac{\mathrm{index}(e)}{N}\cdot\log_2\{\mathrm{len}(e)+1\}\cdot\{H_L(e)+H_R(e)\}$$

$$H_R(e) = -\sum_b \Pr(e \rightarrow b)\log_2\Pr(e \rightarrow b)$$

ここで、$\mathrm{index}(e)$ は抽出した表現の末尾形態素のインデックス（$\mathrm{index}(e)\geq 1$ ）を表し、N は請求項の形態素数を表す。また、$\Pr(e \rightarrow b)$ は表現 e の直後に文節 b が現れる確率を表す。

SVMによる行末デリミタの学習（A.2）

次に、前ステップで抽出した行末デリミタを用いて、請求項に現れる表現の行末デリミタらしさをサポートベクターマシン（SVM）により学習する。本研究で用いた素性の一覧を以下に示す。

●デリミタの表層文字列

●デリミタを構成する形態素の品詞
●デリミタ先頭文節
●デリミタ先頭文節に係る文節
●デリミタ末尾文節が係る文節
●デリミタ末尾文節が係る文節までの距離
●デリミタ末尾文節を跨ぐ係り受けの数

最終デリミタの抽出（B.1）

次に、非改行形式請求項の処理を説明する。非改行形式請求項への最初の処理では、最終デリミタを抽出する。本ステップは、先の改行形式請求項からの最終デリミタの抽出と同様の手順により、最終デリミタを抽出する。

SVM による行末デリミタの推定（B.2）

次に、行末デリミタの推定を行う。本ステップでは，前ステップで抽出した最終デリミタより前に現れる A.1 で抽出した行末デリミタのいずれかと一致する表現を行末デリミタの候補として抽出する。その後、これらの候補に対して、A.2 で学習した SVM を適用し行末デリミタを推定する。

改行の挿入（C.1）

最後に、非改行形式請求項に改行を挿入する。具体的には、前ステップで抽出したデリミタのうち、表 1 に示すジェプソン形式の請求項で用いられるデリミタの前に現れるもの以外の直後に改行を挿入する。

表 1 ：ジェプソン形式の請求項で用いられるデリミタ

において　に於いて　に於て　であって にあたり　に当たり　に当り （上記表現に加え、各表現の末尾に読点を結合したもの）

4 実験

4.1 改行の復元実験

本研究では、提案手法を 2 つの実験によって評価した。1 つ目の実験では、請求項に元から存在した明示的な改行を取り除き、それを復元可能か否か確認する。本実験では、2000 年から 2006 年に公開され審査請求がされた特許から 20,000 件の公開公報をランダムに収集し、その中で明示的な改行を含む第 1 請求項 10,398 件を対象とした。

実験では、上記データを学習用と評価用に分割し、評価用の請求項から改行を取り除き、取り除いた改行をどの程度復元できるかを確認した。今回、データを 5 分割し交差確認を行った。なお、SVM には LIBSVM[4] を使用し、カーネルは RBF、パラメータは $C=2^5$、$\gamma=2^{-5}$ とした。また、本実験では、人手でデータを与えず（人間が一切手間をかけずに）にどの程度復元可能かを調べるために、提案手法 C のジェプソン形式に関する処理を行わず実験を実施した。実験の結果を表 2 に示す。

表 2 では、IPC のセクションごとに結果を示している。また、表中において、候補数とは提案手法の B.2 で抽出したデリミタ候補の数を表す。ここで、特許には一般的に複数の IPC コードが割り当てられるため、1 つの特許が複数の分野に属することがある。そのため、A ～ H の各分野の総計と全体の数は一致しないことに注意する。

表 2 より、セクション C の再現率が他の分野と比較して大幅に低いことが確認できる。セクション C は「化学；冶金」分野である。そこで、セクション C の再現率が低い理由を詳しく調査した。その結果、化学分野に多く見られる 2 種類の表現に含まれるデリミタを正しく抽出できていなかった。1 つ目は、「(1)…、<nl> (2)…<nl> (3)…からなる…」（<nl> が改行位置）や、「α≡… <nl> β≡… <nl>...」といった、構成要素や数式を箇条書きで記述している場合である。2 つ目は、「…は、<nl>…以上、<nl>…以下であり、<nl>…」のような、構成要素の条件が改行で区切られている場合である。

提案手法では、化学式や数式、箇条書きを解析していないため、構文解析に多くの誤りがある。そのため、これらの原因を突き止めるためには、化学式や数式、箇条書きの解析を実装した上でより詳細な調査が必要である。

セクション C 以外の分野は、高い精度で改行を復元することができていたが、他の分野についても誤りの原因を調査した。その結果、以下の 2 種類の請求項で誤った判定をしていた。1 つ目は、「…と、…と、…を備える…において、<nl>…は、…」のような、ジェプソン形式で記述された公知技術の構成要素の直後（下線部）を誤ってデリミタと抽出していた。これによって適合率が低下していた。2 つ目は「…し、<nl>…し、<nl>…する…」のような、順次列挙形式（書き流し形式。処理を順序的に記述する請求項の形式）で記述された請求

表２：改行復元実験の結果

IPC	文書数	改行数	候補数	適合率	再現率	F 値
A	1,201	3,796	35,212	0.78	0.82	0.80
B	2,471	7,410	66,523	0.78	0.82	0.80
C	550	1,289	9,341	0.74	0.60	0.66
D	82	251	2,141	0.74	0.87	0.80
E	429	1,126	11,673	0.74	0.80	0.77
F	1,298	3,782	35,670	0.75	0.84	0.79
G	4,136	14,559	121,895	0.83	0.85	0.84
H	3,789	12,814	105,830	0.82	0.84	0.83
全体	10,398	33,521	293,723	0.80	0.83	0.81

項の動詞連用形の直後をデリミタとして抽出できなかった。これが原因で再現率が低下していた。

１つ目の問題については、実験では提案手法のステップ C.1 の処理を除外したためであり、問題にはならない。２つ目の問題については、請求項全体がどのような形式で記述されているかという情報を素性に加える事で対応できると考える。

４.２ 人手による改行位置の評価実験

２つ目の実験では、自動挿入した改行位置が適切か否かを人間が確認した。具体的には、人間が改行を挿入し、それに対して提案手法がどの程度正しく改行を挿入できたかを確認した。なお、本実験の対象は、ランダムに取得した、1996 年から 2006 年に公開された自然言語処理に関する特許（IPC コード：G06F17/27 － 28）100 件の第１請求項である。また、SVM の学習は、実験１に用いた 20,000 件の特許を使用した。ここで、実験対象である自然言語処理に関する 100 件の特許は、この 20,000 件に含まれていない。実験結果を表３に示す。

表３：人手による評価結果

適合率	再現率	F 値
0.97	0.97	0.97

表３より、自然言語処理装置分野では、非常に高い精度で改行を挿入することができた。ただし、この結果は、自然言語処理分野の請求項の構造が比較的平易であったことが大きいと考える。具体的には、殆どの請求項が「○と、△と、…とを備えた×装置」のような、「と、」で構成要素を連結するパターンであった。今回は、筆者自身の専門分野であったため自然言語処理分野を選択したが、今後、他分野での評価が必要である。

5 おわりに

本研究では、特許請求項を読みやすくするために、改行を自動挿入する手法を提案した。提案手法では、弁理士が請求項に記載した明示的な改行に着目し、それらをそのまま学習データとすることで、人手で用意しなければならないデータを最小限にした。本手法に対して実験を行った結果、化学；冶金分野以外の特許（特に自然言語処理分野）では、高適合率・再現率で改行を挿入できることを確認した。今後、まだ十分な再現率を達成していない化学；冶金分野への対応を中心に手法の改善を行う。

参考文献

[1] 一般社団法人日本語特許情報機構 特許情報研究所, “特許ライティングマニュアル（初版）”, 2013

[2] 新森昭宏, 奥村学, 丸川雄三, 岩山 真, “手がかり句を用いた特許請求項の構造解析”, 情報処理学会論文誌 45(3), pp. 891 － 905, 2004

[3] Yusuke Suzuki, Hirofumi Nonaka, Akio Kobayashi, Hiroyuki Sakai, Shigeru Masuyama, “Extraction of Technology Terms from Patent Specifications for Technology-Effect Type Patent Map Generation,” Proc. of the 25th International Technical Conference of Circuits/Systems, Computers and Communications (ITC-CSCC 2010), pp.725-728, Pattaya, Thailand, 2010.

[4] Chih-Chung Chang, Chih-Jen Lin, “LIBSVM - A Library for Support Vector Machines”, https://www.csie.ntu.edu.tw/~cjlin/libsvm/