

マルチプルアライメントによる放送に対する反響の抽出

† NHK 放送技術研究所 ‡ 東京大学大学院 情報理工学系研究科

†† 小早川 健

kobayakawa.t-ko@nhk.or.jp

† 加藤 直人

katou.n-ga@nhk.or.jp

‡ 辻井 潤一

tsujii@is.s.u-tokyo.ac.jp

1 はじめに

一般的な評判分析では、意見を述べる対象となるもの（例えば、商品名）が予め特定されているため、意見の対象となる（固有）名詞を事前に登録することが、分析の性能向上に役立つ。それに対して、書籍・映画・放送の中身に対する意見を分析する場合は、意見が述べられている対象が単純な名詞にとどまらず、句や節となることがあるために、事前にその可能性を列挙して登録しておく事が難しい。このため、放送に対する意見を分析する先行研究では、意見が述べられている対象を特定せずに、意見を述べている部分だけを抽出し教師あり学習すること [1] や、教師なし学習により意見が述べられている対象を検出すること [2] を試みてきた。

教師あり機械学習によって文書に特徴的な形態素の並びを学習し、文書分類を行う単純な方法でも、一定の分類性能は得られるが、ある特定の放送回に対する反響で学習した分類器を別の放送回に適用するには頑健性に欠けるという課題を抱えていた [1]。その原因のひとつは、単純な n -gram を扱う場合、データスパースネスの問題から n は 3 程度の局所的な情報に限られていたためと考えられる。

そこで、本研究では、より大域的な情報を扱って文書分類を行うことを目指す。ここでいう大域的な情報は、同一カテゴリに分類される文書を形態素解析し、文書毎のアライメント（整列）を行った時に得られる表現テンプレートに含まれるものである。同種の文のアライメントによる先行研究 [3, 4, 5] では、推論を行うための表現テンプレートを抽出している。

2 つ以上のアライメントをすることは、マルチプルアライメントと呼ばれる。先行研究 [3, 4, 5] と同様に動的計画法を用いたマルチプルアライメントを用いて、同種の文をアライメントし、多用されている表現を抽出する。2 節では、アライメントを定式化し、従来法の問題点を指摘し、曖昧性を保持したアライメントを提案する。3 節では、実験により提案法の有効性を検証する。4 節では、全体をまとめる。

2 動的計画法によるマルチプルアライメント

2.1 2 系列データのアライメント

2 つの系列データのアライメントは、編集操作とコストの組である編集距離を定義することにより、動的計画法で厳密に求めることができる [6]。系列データの長さを $\mathcal{O}(n)$ とすると、2 系列データのアライメントの計算量は、 $\mathcal{O}(n^2)$ となる。

本研究では、2 系列データのアライメントを次のように定式化する。系列を表すのに $\{ \}$ を使い、系列 $\{w\}$ の要素を w_i で

表す。アライメントは、2 つの系列 $\{w^{(1)}\}, \{w^{(2)}\}$ を入力し、1 つのアライメント列 $\{a^{(j)}\}$ を出力する関数である。入力系列の要素の集合は Ω とし、アライメント列の要素は、 Ω の要素または空要素 ($-$) からなる。

$$w_i^{(1)}, w_{i''}^{(2)} \in \Omega \quad (1)$$

$$a_i^{(j)} \in \Omega + \{-\} \quad (j \in \{1, 2\}) \quad (2)$$

と表される。また、空要素を含む系列から、要素の順番を維持しながら空要素を取り除く関数 $f_{omit_}$ を次のように定義*1 する。

$$f_{omit_} : (\Omega + \{-\})^* \rightarrow \Omega^* \quad (3)$$

すべての $b_i \in \Omega + \{-\}$ に対して、

$$f_{omit_}(\{b\}) = \{c\} \quad (4)$$

とは、ある単調増加列 $\psi(i)$ と $c_j \in \Omega$ に対して

$$b_i = \begin{cases} c_{\psi(i)} & (b_i \in \Omega \text{ のとき}) \\ - & (b_i = - \text{ のとき}) \end{cases} \quad (5)$$

となることである。アライメント列 $\{a^{(j)}\}$ を求める関数 f_{align} は、拘束条件

$$\{w^{(j)}\} = f_{omit_}(\{a^{(j)}\}) \quad (6)$$

の下で、2 要素 v, w の編集距離 $\text{Edit}(v, w)$ を用いて、

$$\{a^{(j)}\} = f_{align}(\{w^{(1)}\}, \{w^{(2)}\}) = \underset{\{a^{(j)}\}}{\operatorname{argmin}} \sum_i \text{Edit}(a_i^{(1)}, a_i^{(2)}) \quad (7)$$

と定式化できる。また、アライメントコストを求める関数 f_{cost} は、(6) 式の下で、

$$f_{cost}(\{w^{(1)}\}, \{w^{(2)}\}) = \min_{\{a^{(j)}\}} \sum_i \text{Edit}(a_i^{(1)}, a_i^{(2)}) \quad (8)$$

と定式化できる。

ここでは、編集距離を次のように定義*2 する。

$$\text{Edit}(v, v) = 0 \quad (v \neq - \text{ のとき}) \quad (9)$$

$$\text{Edit}(v, -) = \text{Edit}(-, v) = 7 \quad (v \neq - \text{ のとき}) \quad (10)$$

$$\text{Edit}(v, w) = 10 \quad (v \neq w, v \neq -, w \neq - \text{ のとき}) \quad (11)$$

例えば、 $\mathbf{a \ b \ a \ d}$ と $\mathbf{a \ c \ d}$ をアライメントする場合は、アライメント列 $\mathbf{a \ b \ a \ d}$
 $\mathbf{a \ c \ - \ d}$ が出力される。

*1 * は、0 回以上の繰り返しを表す。つまり、 $\Omega^* = \phi + \Omega + \Omega^2 + \dots$ である。 $(\phi$ は空集合である。)

*2 空要素対の挿入を防ぐために、 $\text{Edit}(-, -) > 0$ を定義に加えておくと、数学的厳密さが増すが、以降これは省略する。

2.2 単純なマルチプルアライメント

マルチプルアライメントは、2つの系列データのアライメントを多数の系列データに拡張したアルゴリズムである。入力系列数を M とすると、 M 次元のテーブルを用いて厳密解を求めるアルゴリズムは、計算量が $O(n^M)$ となるため、現実的でない。

このため、近似解法として、2系列データのアライメントを再帰的に繰り返す方法が用いられる [7]。まず、 Ω から構成される高々 M 個の要素からなる集合を $\Omega^{(M)}$ と書き、

$$\Omega^{(M)} = \Omega + \Omega^2 + \dots + \Omega^M \quad (12)$$

と定義する。 M 系列のアライメントをとった $a_i^{(j)}$ で、 i に着目し空要素以外の要素からなる集合を W_i と書き、 $\Omega^{(M)}$ の要素と見なす。マルチプルアライメントの入力は2つのアライメント列から空要素を除去した系列 $\{W^{(1)}\}, \{W^{(2)}\}$ であり、出力は新たなアライメント列 $\{a^{(j)}\}$ である。

$$W_{i'}^{(1)}, W_{i''}^{(2)} \in \Omega^{(M)} \quad (13)$$

$$a_i^{(j)} \in \Omega^{(M)} + \{-\} \quad (j \in \{1, 2\}) \quad (14)$$

マルチプルアライメントは、拘束条件

$$\{W^{(j)}\} = f_{omit_}(\{a^{(j)}\}) \quad (15)$$

の下で、

$$\{a^{(j)}\} = f_{align}(\{W^{(1)}\}, \{W^{(2)}\}) = \operatorname{argmin}_{\{a^{(j)}\}} \sum_i \operatorname{Edit}(a_i^{(1)}, a_i^{(2)}) \quad (16)$$

と表すことができる。

ここで得られるアライメント列は、空要素を除去し再帰的にマルチプルアライメントの入力 $\{W\}$ に用いる。例えば、2入力から出力されるアライメント列 $\{a^{(j)}\}$ は $\begin{smallmatrix} a & b & a & d \\ a & c & - & d \end{smallmatrix}$ と表わされ、空要素を除去して次段のマルチプルアライメントに入力されるアライメント列 $\{W\}$ は、 $a \left(\begin{smallmatrix} b \\ c \end{smallmatrix} \right) a d$ と表される。

アライメント列の要素 $a_i^{(j)}$ は集合又は空要素であることに注意して、編集距離を次のように定義する。

$$\begin{aligned} \operatorname{Edit}(a_i^{(1)}, a_i^{(2)}) &= 0 \\ &\quad (a_i^{(1)} \text{ と } a_i^{(2)} \text{ が同じ要素を持つとき}) \end{aligned} \quad (17)$$

$$\begin{aligned} \operatorname{Edit}(a_i^{(j)}, -) &= \operatorname{Edit}(-, a_i^{(j)}) = 7 \\ &\quad (a_i^{(j)} \text{ が } - \text{ でないとき}) \end{aligned} \quad (18)$$

$$\begin{aligned} \operatorname{Edit}(a_i^{(1)}, a_i^{(2)}) &= 10 \\ &\quad (a_i^{(1)} \text{ と } a_i^{(2)} \text{ に共通の要素がないとき}) \end{aligned} \quad (19)$$

M 個の入力系列を再帰的にアライメントするとは、2つの系列を選ぶ組み合わせ ${}_M C_2$ 通りについて、それぞれ $f_{cost}(\{w^{(k)}\}, \{w^{(l)}\})$ を求め、最低コストの組み合わせ (k, l) から順に f_{align} を行っていく。

2.3 単純なマルチプルアライメントの問題点

アライメントでは、コスト f_{cost} が等しいアライメント f_{align} が複数存在しうる。単純な動的計画法では、同コストのいずれかひとつのアライメントのみが出力されていた。マルチプルアライメントのようにアライメントを繰り返す場合、1回のアライメントで選択されたアライメントが後段のアライメントに影響を及ぼす。この1回のアライメントでは、全体的に見て最適なアライメントが選択されるとは限らないので、マルチプルアライメントでは、全体として不適切なアライメントが得られやすい。

生物分野でマルチプルアライメントが用いられる場合、編集距離が要素毎にきめ細かく決められている場合が多い [7]。このため、コストの等しい候補が多数存在するケースは少ない。一方、自然言語処理では、このようにきめ細かくは編集距離が決められていないケースが通常であるため、同コストの解が多数存在することになる。

2.4 曖昧性を保持したマルチプルアライメント

単純なマルチプルアライメントには、上記の問題があるため、アライメントを拡張して曖昧性を保持するようにする。

まず、 f_{align} は集合を返す関数とし、その集合はコスト最小となるアライメントのすべてを要素に持つ。

$$f_{align}(\{W^{(1)}\}, \{W^{(2)}\}) = \{\{a^{(j)}\} \mid \min_{\{a^{(j)}\}} \sum_i \operatorname{Edit}(a_i^{(1)}, a_i^{(2)})\} \quad (20)$$

実装は、動的計画法のテーブルで保持するデータを拡張することによって実現する。動的計画法のテーブルのセルには直近の最適経路をひとつだけ保持するのが従来法であるが、それを最適経路すべての集合を保持するように変更する (図 1)。動的計画法のテーブルを埋め終わった後に最適経路をトレースバックする時に、同スコア候補のすべてのアライメント列候補を出力する (図 2)。図では、着目しているセルの左からの経路と上か

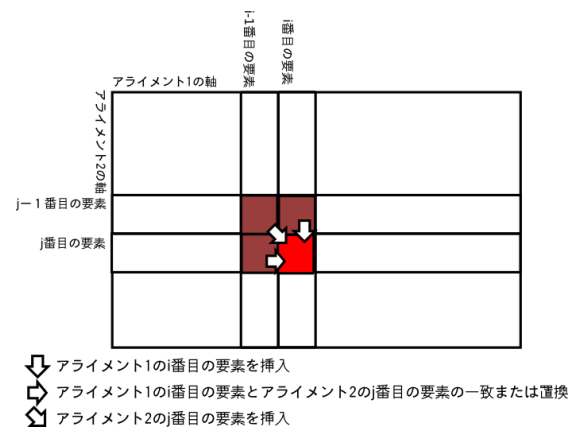
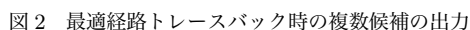


図1 拡張マルチプルアライメントのデータ構造

らの経路が同コスト最適経路である場合が示してある。トレースバック時に終点 (テーブルの右下のセル) からそのセルまでの経路のコピーをスタックに積み、スタックの中で始点 (テーブルの左上のセル) に到達していない経路を順にトレースバックするようにアルゴリズムは動作する。一つのアライメントから得られる同コスト最適経路の数は、終点から始点の方向に経路


$$1 + \sum_{q \in \text{分岐箇所}} (n_q - 1) \quad (21)$$

同一コストの最適解の集合が f_{align} の返り値となるため、再帰的に f_{align} を適用する場合には、集合に対する f_{align} が定義されなければならない。マルチプルアライメントで得られるアライメント列の集合を $\mathcal{A}^{(1)}, \mathcal{A}^{(2)}$ とすると、それぞれの要素の組み合わせのマルチプルアライメントの和集合を集合のマルチプルアライメントと定義する。

$$f_{\text{align}}(\mathcal{A}^{(1)}, \mathcal{A}^{(2)}) = \sum_{\{a^{(k)}\} \in \mathcal{A}^{(1)}} \sum_{\{a^{(l)}\} \in \mathcal{A}^{(2)}} f_{\text{align}}(\{a^{(k)}\}, \{a^{(l)}\}) \quad (22)$$

2.5 曖昧性の枝刈り

2.5.1 アライメントコスト最小となる枝刈り

2.5.2 空要素の数最小となる枝刈り

2.5.3 最適経路候補の絶対数による枝刈り

2.5.4 集合要素の絶対数による枝刈り

2.6 最良候補の選択

記の基準は、最終候補に対して 1 回のみ適用され、最良候補の選択に用いられる。

2.6.1 低頻度辺の枝刈り数最小の基準

最終アライメント列は有向グラフに変換され、辺には同一ノードを結ぶアライメント列の数が付与される。ある一定頻度の以下の低頻度辺は枝刈りされ、それによって孤立する頂点は除去される。ここでは、低頻度辺の枝刈り数が最小となるグラフを選択する基準を導入する。

2.6.2 頂点数最小の基準

さらに、得られた有向グラフの中で、頂点の数が最小となるグラフを選択する。これは、同一の表記となる頂点になるべく少なくなるグラフが、よりよくアライメントされたグラフであるという仮定に基づくものである。

3 実験

3.1 実験コーパス

実験協力者に放送番組を視聴してもらい、放送に対する感想を集めたコーパスが存在する。対象とした放送番組は 2006 年に教育テレビで放送された「知るを楽しむ」の「武士の家計簿」というシリーズの最終回である。これは、30 分程度の教養娯楽番組で、江戸時代の武士の経済状況を扱ったものである。実際の放送局で感想を分類するのに類似した分類項目を用いて、同コーパスを人手で分類した。1000 程度集まった意見の中から、意見数の多い項目を拾い出して表 1 に列挙した。

表1 意見の分類項目と意見の数

意見の分類項目	意見の数
意外だ	69
感動した	13
考えさせられた	20
気の毒だ	13
知った	57
素晴らしい	25
よい/よくない	53
合計	250

3.2 評價指標

意見を茶筌*3により形態素解析し、意見の分類項目毎にマルチプルアライメントを行う。この場合、アライメントの要素は形態素となる。低頻度辺の枝刈り (2.6.1 節) により、全表現が一つに結合した大きなアライメントは表現の断片に分断される。この表現の断片の中に、意見を人手で分類する時に着目した表現がどの程度含まれているかを評価指標とする。ここで抽出されたものを表現テンプレートと呼ぶことにする。マルチプルアライメントは、従来手法を用いた場合と提案手法を用いた場合の表現テンプレート抽出結果を比較する。

3.3 実装と実験パラメター

8CPU Xeon, 64GB RAM, Linux 上で実装を行って実験した。実験パラメータには表 2 に示すものを用いた。

*3 Ver. 2.3.3 <http://chasen-legacy.sourceforge.jp>

表2 実験パラメター

項目	値
最適経路候補の絶対数による枝刈り (2.5.3 節)	1000
集合要素の絶対数による枝刈り (2.5.4 節)	100
低頻度辺の枝刈り (2.6.1 節)	1

3.4 実験結果

まず、マルチプルアライメントによって抽出された表現テンプレートの例を従来法と提案法を比較で図3に示す。表現テンプレートはグラフ構造で表わされ、頂点には形態素が、辺には頂点の接続が対応する。辺にある数字は、その接続を通過する意見がいくつあったかを示す。提案法は以下の2点で従来法より優れている。まず、提案法では文末の「すごい」と「すばらしい」が抽出されているのに対して、従来法ではそれらが抽出されていない。次に、従来法では「新政府の」や「の経理」の表現が余計に抽出されているのに対して、提案法ではそれらの余計な抽出がない。

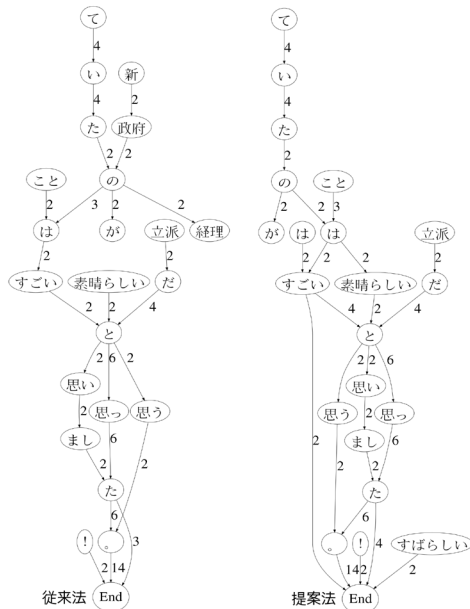


図3 マルチプルアライメントによって抽出された表現テンプレート

各分類項目に対して、抽出できた表現の比較を表3に示す。

表3 分類項目毎の抽出表現数

分類項目	従来法	提案法
意外だ	8	8
感動した	2	2
考えさせられた	2	2
気の毒だ	2	1
知った	6	7
すばらしい	9	11
よい/よくない	1	2
合計	30	33

次に、従来法と提案法の計算時間の比較と並列処理による

CPU 負荷率を表4に示す。提案法は、従来法と比較して2156倍もの計算時間を消費していることがわかる。提案法では、OpenMP による並列処理を行っており、並列化の効果はあがっている。一方、さらなるアルゴリズムの工夫も期待される。

表4 計算時間の比較

分類項目	従来法の 計算時間 (s)	提案法の 計算時間 (s)	提案法の CPU 負荷率 (%)
意外だ	0.124	543.157	396
感動した	0.010	7.200	243
考えさせられた	0.020	225.641	233
気の毒だ	0.011	78.336	241
知った	0.104	365.888	341
すばらしい	0.044	56.928	339
よい/よくない	0.360	174.115	343
合計	0.673	1451.265	339

4 おわりに

自然言語処理に用いられるマルチプルアライメントの拡張アルゴリズムを提案し、その実装と実験を通じて提案法の性能を測定した。放送に対する反響の分析に用いて多用される表現の抽出を試みた結果、従来法と比較して提案法には1) 必要な表現が多く抽出され、2) 不要な表現が少なく抽出されるという優位性が見られた。

提案法は、アライメントの計算過程で生じる曖昧性をなるべく保持した計算を行うことで真の最適解を捨てないように工夫されたアルゴリズムになっている。そのために表現の抽出性能が上がっていると考えられる。その代償として多くの計算時間を消費している。また、従来法と違って解の可能性を多く提示することから、それらを絞り込む効果的な外的基準も導入した。

提案法は、本質的に指数関数的な計算量を持つアルゴリズムである。このため各種の枝刈りを導入し、計算可能性を担保している。枝刈りをどのように行うかについては、まだ十分な検討が加えられていないので、それは今後の課題である。同時に、高速化/省メモリ化を進めていく中で、より大規模なアライメントを行って、多用表現を抽出する実験も行っていく必要がある。

参考文献

- [1] 小早川, 宮崎, 藤井, 八木: “品詞 n -gram による反響表現の抽出”, 第13回言語処理学会年次大会発表論文集, No. D2-7, pp. 428–431 (2007).
- [2] 小早川: “放送に対する反響抽出の課題”, 第6回情報科学技術フォーラム一般講演論文集, 第2分冊, No. E-058, pp. 277–278 (2007).
- [3] R. Barzilay and L. Lee: “Bootstrapping lexical choice via multiple-sequence alignment”, Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 164–171 (2002).
- [4] R. Barzilay and L. Lee: “Learning to paraphrase: An unsupervised approach using multiple-sequence alignment”, HLT-NAACL 2003: Main Proceedings, pp. 16–23 (2003).
- [5] R. Barzilay and L. Lee: “Catching the drift: Probabilistic content models, with applications to generation and summarization”, HLT-NAACL 2004: Proceedings of the Main Conference, pp. 113–120 (2004). Best paper award.
- [6] T.H., C.E., R.L.: “アルゴリズムイントロダクション”, 第2巻アルゴリズムの設計と解析手法, 近代科学社 (1995).
- [7] R.Durbin, S.Eddy, A.Krogh and G.Mitchison: “Biological sequence analysis”, Cambridge University Press (1998).