Support Vector Machine に基づく Web 新聞記事の自動要約

大森 岳史† 増田　英孝† 中川 裕志‡

東京電機大学工学部† 東京大学情報基盤センター‡

1　はじめに

本研究では、Web 記事から携帯端末に適した長さの自動要約を生成することを目的としている。現状では、Web 記事などの文字数の多いコンテンツを携帯端末で読んだ場合、携帯端末は小画面なために読みにくい。また、携帯記事を人手によって作成した場合、コストがかかってしまう。Web 記事から携帯端末向けの記事を自動生成することにより、記事作成のコストを削減できる。

我々は以前、あるジャンル 1 日当たりの記事全体を文書集合とみなした場合の TF・IDF に基づいた要約手法を提案した [1]。本稿では、Support Vector Machine を用いて名詞の TF・IDF、係り受けの深さなどの素性に基づき、要約結果として残す文節を判定する手法について述べる。本手法と TF・IDF に基づく手法の要約結果を、それぞれ携帯端末向けに配信している新聞記事との比較で評価を行なう。

2　対象とする新聞記事データ

2.1　対応付け記事コーパス

自動要約結果を評価するために正解データが必要となる。そこで、Web 記事と携帯記事を用いて Web 記事と携帯記事で同じ内容の記事の対を作成した。これを本稿では「対応付け記事」と呼ぶ [2]。

2.2　対応文コーパス

SVM で学習を行う際に使用するため、対応付け記事を元にして対応文を作成した。携帯記事は 1 記事が最大 3 文 (文：句点で区切られる単位) で構成されており、全体として 50 文字程度でまとめられている。対応文作成時は携帯記事 1 記事を、複数文であっても 1 文とみなす。また、Web 記事は複数の文で構成されている。

携帯記事文は少ない文字数で簡潔に事柄を伝えていることから、携帯記事文で使用されている名詞類 (一般名詞, 固有名詞, サ変名詞, 未知語など) は記事の内容を伝える上で重要であると考えられる。そこで、名詞類に注目し、記事間の対応がある携帯記事文 1 文を、複数ある Web 記事文のうち、上記名詞類を一番多く含む文と対応付ける。これを本稿では「対応文コーパス」と呼ぶ。

3　Web 記事の自動要約

Web 記事の第 1 段落には、かなり精選された文が集中している。したがって、これらの文をできるだけ活かす方が要約文の質を保つことができると考えられる。本稿では SVM により、これらの文の不要箇所の特定を行う方法で要約する。

図 1 に示すように自動要約の対象の Web 記事を記事 A とし、A の第 1 段落の文を A1,A2,…,Am とする (図 1　m：第一段落の構成文数)。Web 記事ではほとんどの場合、重要なことは先頭から 3 文以内に書かれているので、本稿では A1,A2,A3 に着目する。また、形態素解析器として「茶筌」[3] を、係り受け解析器として「南瓜」[4] を使用する。

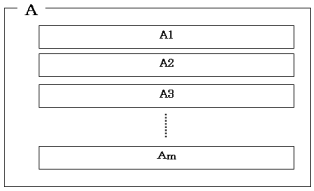

図 1: 要約対象記事の構成

3.1　SVM に基づく要約

SVM とは，1995 年に Vapnik によって開発された 2 クラスのパターン認識問題を解くための機械学習法である。学習データを正例と負例に分け、且つ正例、負例のマージンが最大となるような分離超平面を求めることが可能である。本稿では「TinySVM」[5] を使

用した。

2.2 節で携帯記事の名詞類は重要であるとした。名詞類が含まれている Web 記事対応文の文節も同様に重要であると仮定して「重要文節」とする。一方で、携帯記事の名詞類が含まれていない Web 記事の文節を「非重要文節」とする。そして、「重要文節」を正例、「非重要文節」を負例と定義して学習データの対応文で学習を行い、テストデータの対応文で認識を行う。

3.1.1 学習モデルの作成

対応文の携帯記事から名詞類を抽出する。対応している Web 記事文の「南瓜」の結果を使用して、文節ごとに素性値を割り当てる。素性の数は次の 6 項目、合計 131 素性である。

(1) ***TF・IDF* 値**
文節ごとに算出された TF・IDF 値で、算出方法は 3.2 節と同様の方法で求める。

(2) **文末記号からの距離**
文末記号からある文節ににたどり着くまでの距離を文末記号からの距離とする。図 2 の例では、文節 1 と文節 3 は文節 4 と文節 5 を経由して文末文節にたどり着くので、文末の文節からの距離は 3 となる。また、文節 2 は文節 3 と文節 4 と文節 5 を経由して文末記号にたどり着くので、文末の文節からの距離は 4 となる。

(3) **係り元文節の数**
素性を求めている文節に直接係っている文節の数をいう。図 2 の文節 4 は文節 1 と文節 3 が係っているので係り元文節の数は 2 となる。

(4) **文節番号**
Web 記事の先頭の文節から順に番号を割り当てる。この番号を文節番号の素性値とする。

(5) **文節の総数**
Web 記事文の文節の合計値である。図 2 の例では５つの文節で構成されているので、全ての文節の素性値が 5 となる。

(6) **品詞** ***(126* 素性*)***
文節に含まれる品詞の種類である。「茶筌」の 126 品詞を素性とし、当初、各素性値を「0」とする。例えば、「私は」の形態素解析の結果は、

私：名詞-代名詞-一般

は：助詞-係助詞

となるので、対応する素性値に「1」を加算する。

以上の素性を文節ごとに算出する。全ての素性値の算出が終了したら、それぞれの文節に対して対応付けた携帯記事の名詞類が含まれているかを比較する。携帯記事の名詞類が含まれていた文節は「重要文節」と定義することとし、正例とする。携帯記事の名詞類が含まれない文節の場合は「非重要文節」であるので、負例とする。この学習データを元にして学習モデルを作成する。

3.1.2 SVM による要約手順

要約対象の Web 記事を図 1 を例にして、SVM による要約手順を示す。

Step:0 Web 記事の本文から 3 文 A1,A2,A3 を取り出す。

Step:1 文 A1,A2,A3 を係り受け解析する。

Step:2 係り受け解析した文 A1,A2,A3 の文節ごとの素性値を算出する。

Step:3 学習モデルを用いて SVM に各文節の素性値を入力し、重要文節、非重要文節の判定を得る。

Step:4 文 A1,A2,A3 から非重要文節を削除して終了。

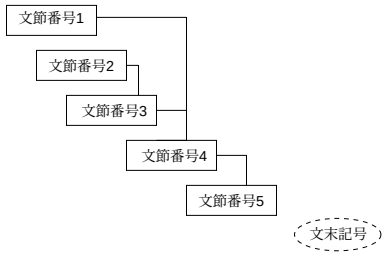

図 2: 係り受け解析の例

3.2 TF・IDF に基づく要約

次に、評価の比較に用いた TF・IDF に基づく要約の概要を示す。Web 記事はジャンルと日付によってまとまりを持つものである。また、ある日のあるジャンルの記事は一般に複数あり、その一つ一つを文書頻度

(DF 値) の算出の単位とする。以下に TF・IDF に基づく要約手順を示す。ここで、要約目標の文字長を L とする。L は 50 文字程度と 100 文字程度の 2 種類を設定した。

初めに要約対象の Web 記事から「茶筌」により名詞類を抽出する。次に、これらの名詞類の $TF \cdot IDF$ 値を算出する。記事 A を A と表し、TF・IDF 値を求める単語を W と表す。以下の式により記事 A に出現する単語 W の $TF \cdot IDF$ 値を求める。

$$TF \cdot IDF(A,W) = TF(A,W) \cdot IDF(W) \qquad (1)$$

$TF(A,W)$ は記事 A における、単語 W の生起頻度である。$IDF(W)$ は当日に収集された文書数 N と、N の中で W が一回以上生起する文書数 $DF(W)$ に関係し、次のように定義する。

$$IDF(W) = \log(\frac{N}{DF(W)} + 1) \qquad (2)$$

また、予備実験の結果、助詞「は」付き文節中の名詞は重みを 10 倍とすることにした。

次に、要約対象の Web 記事文 A1,A2,A3 をそれぞれ「南瓜」を使用して、係り受けの結果を得る。これらの文節に算出した TF・IDF 値を加算する。最後に係り受け解析結果の枝の先端の文節の TF・IDF 値を比較して、最低の値を持つ文節を削除する。以降、L に到達するまで、文節の削除を繰り返す。

以上が我々が以前に提案した、TF・IDF に基づく要約の概要である。

4 評価

4.1 名詞の一致率

要約の評価には対応付け記事を使用し、Web 記事の要約結果とそれに対応する携帯記事の名詞一致率を算出した。要約対象の記事は、SVM と TF・IDF による要約の両手法で、同一の記事集合を使用した。要約対象の記事数は、政治 84 記事、経済 248 記事、国際 191 記事、社会 238 記事の合計 761 記事である。

また、SVM の学習データには 2003 年 8 月 1ヶ月分の対応文を使用した。ジャンル別の対応文の数は、政治 403 文、経済 495 文、国際 521 文、社会 574 文である。

携帯記事と要約結果に関して精度と再現率、F 値を以下の式に従い算出した。

$$\text{精度} = \frac{\text{(携帯記事と要約結果に共通する名詞数)}}{\text{(要約結果に含まれる名詞数)}} \qquad (3)$$

$$\text{再現率} = \frac{\text{(携帯記事と要約結果に共通する名詞数)}}{\text{(携帯記事の名詞数)}} \qquad (4)$$

$$F \text{ 値} = \frac{(2 \times \text{精度} \times \text{再現率})}{(\text{精度} + \text{再現率})} \qquad (5)$$

SVM に基づく要約結果を表 1 に示す。TF・IDF に基づく要約結果を表 2 に示す。また、「統合」とは政治、経済、国際、社会の 4 ジャンルの記事集合である。

4.2 考察

本稿の SVM に基づく要約手順では、要約目標の文字数 L の指定はできない。そこで、従来の TF・IDF に基づく要約結果との比較を行うために、表 1 と表 2 を元にして両手法の精度、再現率、F 値を図として表す。

図 3 に要約結果の精度を示す。50 文字付近の点が TF・IDF に基づく要約 (L=50) を示している。また、100 文字付近の点は TF・IDF に基づく要約 (L=100) の精度である。この二点を結んだ直線よりも SVM に基づく要約結果が高ければ、SVM に基づく要約結果が TF・IDF に基づく要約よりも、高い精度を持つと予測される。この結果、精度は 4 ジャンル全てにおいて SVM に基づく要約の精度が高いことが分かる。

また、図 4 の再現率、図 5 の F 値においても同様のことが言える。

携帯記事の名詞類との一致率において、要約精度では SVM に基づく要約に優位差が見られた。しかし、SVM に基づく要約では、文末の用言が削除される場合があった。今後は、文末の用言が削除された場合の言い換え処理の必要性がある。

表 1: SVM に基づく要約結果

	精度 (%)	再現率 (%)	F 値 (%)	文字数		記事数
				要約後	要約前	
政治	47.0	68.3	54.4	84	151	84
経済	39.3	61.6	47.2	87	151	248
国際	47.0	68.3	53.9	94	170	191
社会	43.8	58.0	48.7	76	180	238
統合	42.8	63.6	50.2	85	165	761

表 2: TF・IDF に基づく要約結果

	精度 (%)		再現率 (%)		F 値 (%)		文字数		記事数
	100 字程度	50 字程度	100 字程度	50 字程度	100 字程度	50 字程度	100 字程度	50 字程度	
政治	37.8	50.2	65.6	39.4	47.6	40.2	104	55	84
経済	36.3	42.1	63.8	40.8	46.0	40.8	107	56	248
国際	41.4	46.5	71.1	44.8	52.0	44.6	106	56	191
社会	36.2	41.5	64.2	39.7	45.9	39.5	106	56	238
統合	37.7	43.9	66.0	41.3	47.6	41.3	106	56	761

5 まとめ

本稿では SVM により、携帯端末向けに Web 新聞記事の要約を行なう手法を提案した。SVM に基づく要約は要約結果として残す文節を SVM により名詞類の TF・IDF 値、係り受けの深さなどの素性に基づき判定し、非重要文節を削除する手法である。

また、SVM に基づく要約結果を従来の TF・IDF に基づく要約結果と比較し、評価を行った。今後は、SVM で学習する学習データの量を増やすことで要約精度がどのように変化するのかを調査する必要がある。また、人が要約結果を読んだ場合の読みやすさの評価を行う予定である。

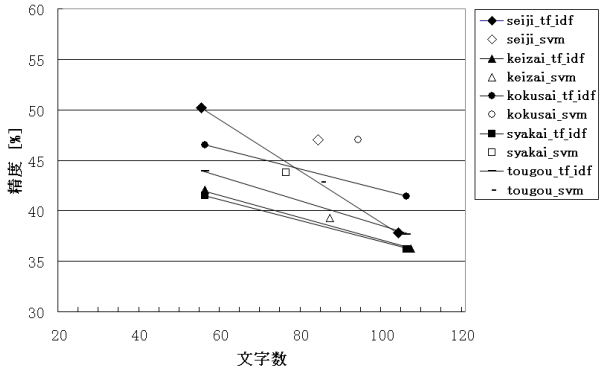

図 3: 精度

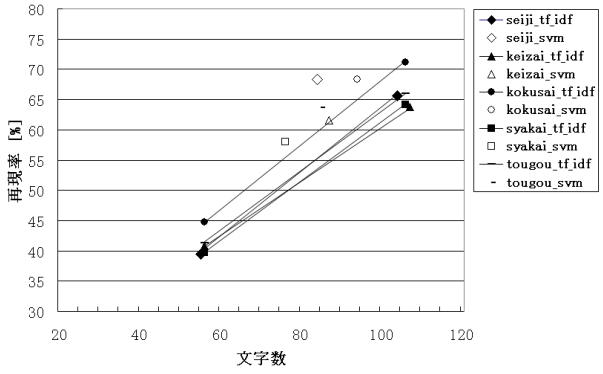

図 4: 再現率

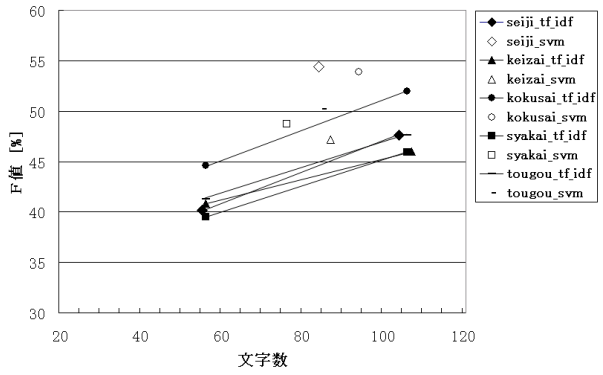

図 5: F 値

参考文献

[1] 大森岳史, 増田英孝, 中川裕志：Web 新聞記事の自動要約と i モード記事による評価, 言語処理学会 第 9 回年次大会 A3-2, pp. 202–205 (2003).

[2] 大森岳史, 金田崇宏, 増田英孝, 中川裕志：携帯端末向け記事とインターネット新聞記事の対応付け, 情報処理学会第 64 回全国大会, Vol. 3, pp. 147–148 (2002).

[3] 奈良先端科学技術大学院大学自然言語処理学講座：日本語形態素解析システム「茶筌」, http://chasen.aist-nara.ac.jp/.

[4] 奈良先端科学技術大学院大学自然言語処理学講座：日本語係り受け解析器「南瓜」, http://cactus.aist-nara.ac.jp/~taku-ku/software/cabocha/.

[5] Kudoh, T.: TinySVM: Support Vector Machines, http://cl.aist-nara.ac.jp/~taku-ku/software/TinySVM/.