スプログの調査と実システムにおける判別手法

著者
高橋哲朗(　ニフティ株式会社　)
野田雄也(　〃　)
岩倉友哉(株式会社富士通研究所)

発表者：三上健太

はじめに

- ブログの情報
  - マーケティングに有用な情報
  - いくつものサービスが提供されている(kizasi.jpなど)

- ブログの中にはスプログ(スパムブログ)が大量に混入
  → ブログの分析結果に悪影響を与える

↓

- スプログの種類と割合についての調査結果の報告
- 引用型スプログの判定手法の提案と評価実験の報告

スプログの現状調査

- 日本語で書かれたほぼ全ブログからランダムに抽出したブログ記事3719件を人手で分析

- 引用型：ニュース記事などをコピーし生成した記事
- アフィリエイト：アフィリエイトリンクのみで構成され，オリジナル記述をほとんど含まない
- アダルト：卑猥な文書が書かれている記事
- ワードサラダ：複数の情報源から単語を抽出し，それらを並び替え生成した記事
- ギャンブル・金融：ギャンブルやFXで儲ける方法などの話題に閉じている記事
- ショッピング：商品が列挙されたショッピングサイト

スプログ種類	件数	割り合い (%)
引用型	449	12.1
アフィリエイト	249	6.7
アダルト	191	5.1
ワードサラダ	148	4.0
ギャンブル・金融	107	2.9
ショッピング	39	1.0
スプログ記事	909	24.4
非スプログ記事	2810	75.6
計	3719	100.0

※1つの記事に対して複数のスプログ種類を割り当てることもあるので，各スプログ種類件数の合計値とスプログ記事数は一致しない

スプログの現状調査(2)

- 本稿では引用型スプログの検出を行う手法を提案
- 引用型スプログ
  - 主にニュース記事から引用された記事を元としたブログ
  - 引用部分の前後に，生成された文章が置かれている

  ex.

  ・引用部分とは関係ない話題

  「ではそろそろ気持ちを切り替えて，ちょっと外出してきます」

  ・引用部分への感想

  「最近{記事タイトル}が気になります」

提案手法

- 引用型スプログ
  - 単独の記事そのものの特徴のみからの判定は困難
  - ある記事集合を対象とし，その中から引用型スプログの集合を見つける必要がある
  - 引用型スプログ検出を類似文書集合を探す問題ととらえる
- 記事数：Mとしたとき，単純に類似度計算をすると・・・

  → $\mathcal{O}(M^2)$ の計算量が必要

  → 効率的な計算手法が必要
- 本稿では，転置インデックス形式を応用した，高速に類似記事を判定する手法を提案

提案手法(2)

- $\{x_1, \cdots, x_N\}$: 類似度計算を行うN文
- 各文　$x_i (1 \le i \le N)$: 単語の集合 $\{x_{i,1}, \cdots, x_{i,|x_i|}\}$
- $|x_i|$: 文サイズ
- 目的 → 類似度　$SIM(x_i, x_j)(1 \le i, j \le N, i \ne j)$　を閾値s以上
  となる文のペア集合を発見
- $$SIM(x_i, x_j) = \frac{\sum_{k=1}^{|x_i|} x_{i,k} \subseteq x_j}{|x_j|}$$
- $\omega \in x_i$　: x_i が単語ωを含んでいる場合1,それ以外は0を返す
  $(|x_i| \le |x_j|)$
- 類似度は $0 \le s \le 1$　となる

提案手法(3)

- x_i との類似度が閾値s以上となる文 x_j を見つける

$$\longrightarrow s \leq \frac{|x_i|}{|x_j|} \longrightarrow |x_j| \leq \frac{|x_i|}{s}$$ を満たす文 x_j だけを計算する

- 共通して文中に出てくる単語数を調べる
- 比較する文の単語のうち共通して出てくる単語の割合が閾値以上ならば類似文とみなす

評価実験～問題設定～

- 記事から本文テキスト部分を抽出
- この入力に対し類似している記事集合を発見する
- 類似している文の割合が0.3以上 → 類似記事
- 閾値は0.8に設定
- 単語：形態素解析結果から品詞により内容語のみを選択し，
  それらの表層文字列
- 実験結果：スプログに含まれる割合に依存
  → 2種類のデータを用いる
  - window:ある時間間隔に投稿された記事集合
  - query:特定のクエリを含む記事集合
- 1000件の記事を抽出，3つのアルゴリズムにより実験

評価実験～実験結果～

- 提案手法による処理速度向上は明らか
- 記事数1000のとき約47(query)～85倍(window)の速度差
- 検出されたスプログの記事数(入力記事数1000のとき)
  → window：163件，query：300件
- データ種類によってスプログ記事数が異なる(処理時間も5倍)
- スプログ記事数が及ぼす処理時間への影響は，提案手法のほうが大きい
- 枝刈り効果も確認された
  → 入力記事数1000のとき約5倍程度の速度向上

類似記事発見の処理時間(sec)

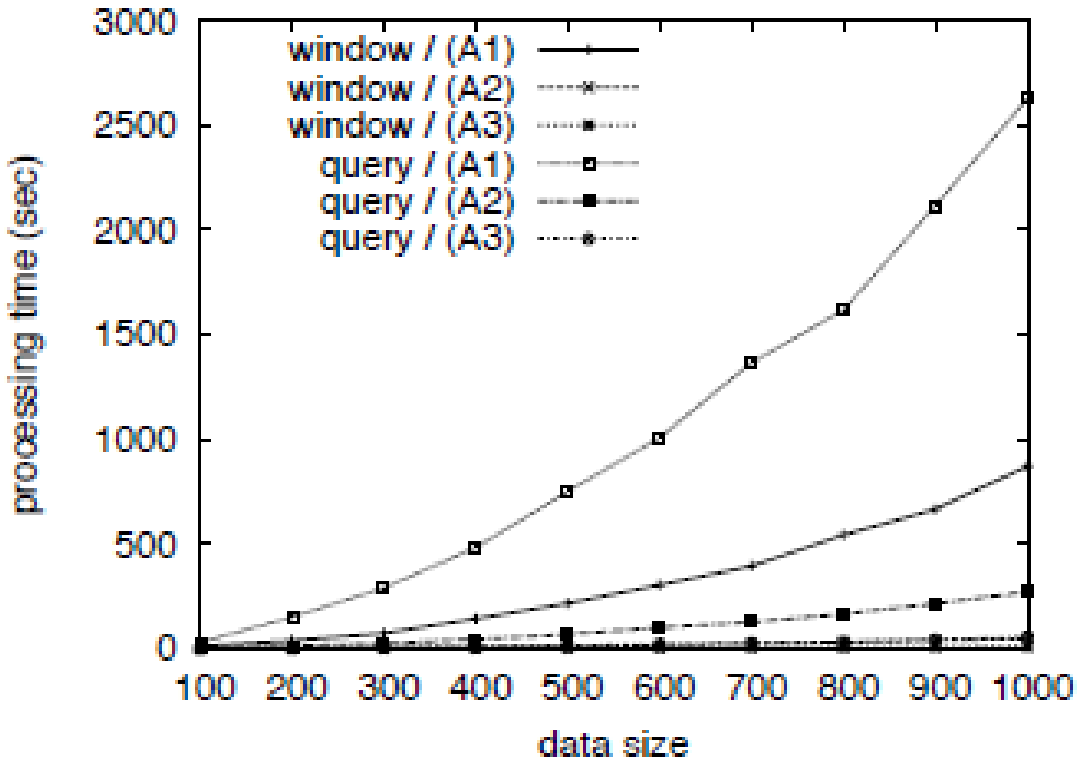
データ	手法	入力記事数 100	200	500	1000
window	(A1)	11.728	39.356	217.45	872.88
	(A2)	0.841	2.336	11.35	49.21
	(A3)	0.737	0.781	2.83	10.22
query	(A1)	30.267	151.658	751.65	2627.25
	(A2)	2.904	12.513	68.16	271.56
	(A3)	0.758	2.763	14.77	55.65

約85倍

約47倍

A1:単純な類似度判定
A2:提案手法(枝刈りなし)
A3:提案手法(枝刈りあり)

window:記事内容はランダム
query:記事内容に偏りあり



入力データサイズと類似記事発見の処理時間

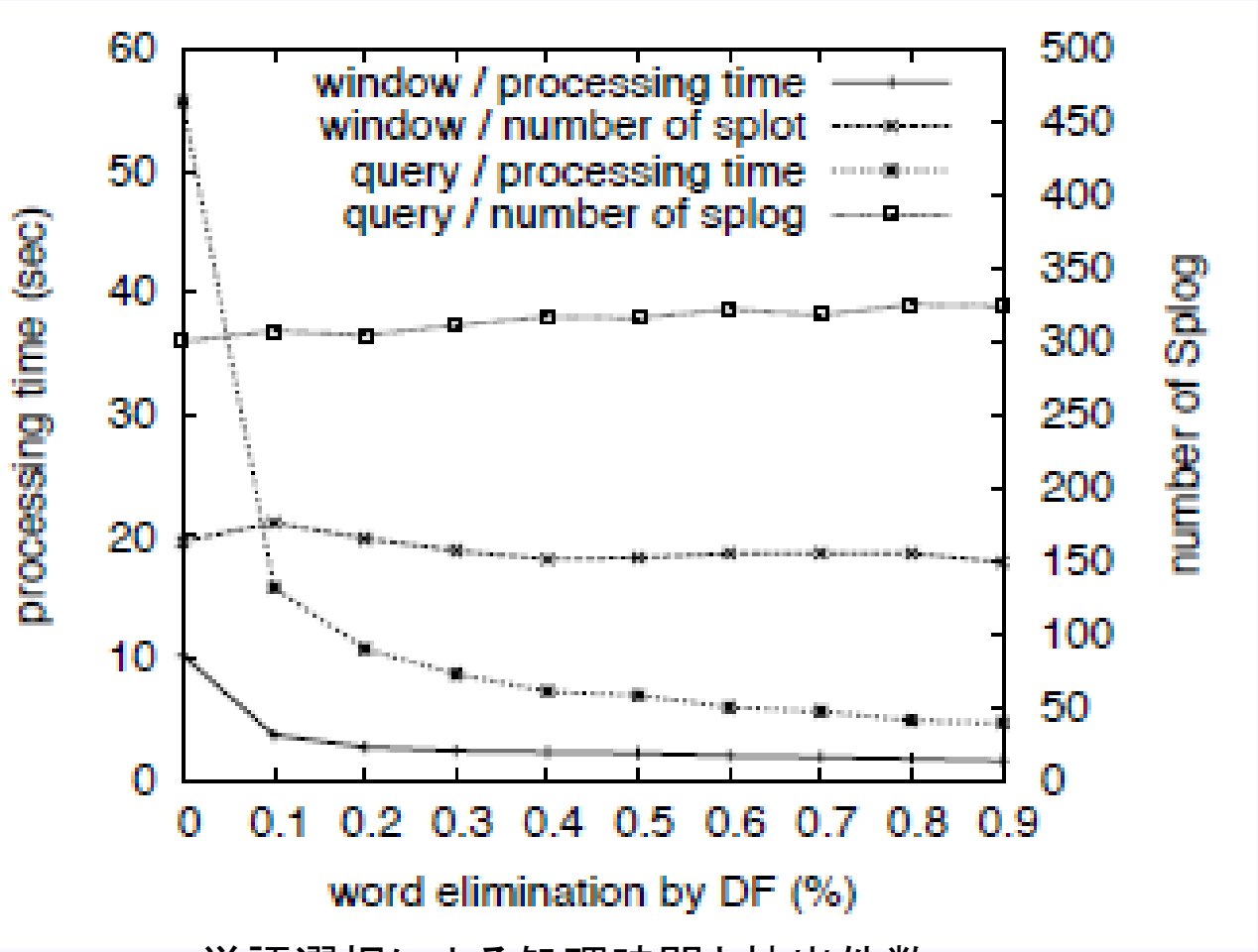
評価実験～実験結果(2)～

- 使用する単語のうちDF(Document Frequency：文書頻度)の高い方から一定の割合で単語を削除(上位0.2%)
  → window：3.8倍，query：5.2倍の速度向上

- 単語を削除しても検出されるスプログ数に大きな変化なし
  → 多くの文に共通に含まれる単語だから(判定に影響少)
  (する，いる，今日など)

単語選択による処理時間(sec)

データ	DF に基づく使用単語の削除 (%)					
	0.0	0.1	0.2	0.3	0.5	0.9
window	10.22	3.68	2.70	2.40	2.20	1.56
query	55.65	15.77	10.75	8.70	6.94	4.69



単語選択による処理時間と抽出件数

関連研究

- 教師あり機械学習による手法
  - アダルト系やギャンブル系に有効
  - ワードサラダ型や引用型スプログには無効
- 単語の出現分布による手法
  - ワードサラダ型には効果的
  - 引用型スプログには無効
- 類似度による手法
  - 引用型スプログの判別は困難，(または効率が悪い)

まとめ

- 日本語で書かれたスプログの分類を行った
- 引用型スプログに対する検出方法の提案と評価を行った
- 提案手法を用いた結果，処理速度向上を確認(約100倍)

- 今後の課題
  - どのくらいの時間範囲の記事を対象とするかを決める
    → 実システムで運用するため
- 今後拡張
  - 類似記事に関する情報を蓄積しておき，ストリームとして入力されるブログ記事を特定できるようにする