修 士 論 文

汎用連想計算エンジン GETA を用いたスパム判別に関する研究

北陸先端科学技術大学院大学
情報科学研究科情報処理学専攻

石黒 雄輔

2008 年 9 月

修 士 論 文

汎用連想計算エンジン GETA を用いたスパム判別に関する研究

指導教官　小川瑞史 教授

審査委員主査　小川瑞史 教授
審査委員　緒方和博 特任准教授
審査委員　青木利晃 特任准教授

北陸先端科学技術大学院大学
情報科学研究科情報処理学専攻

0610006 石黒 雄輔

提出年月: 2008 年 8 月

謝辞

連想計算エンジン GETA の詳細についてご教示いただきました高野明彦氏ならびに西岡真吾氏 (NII) に感謝いたします。また本研究を進めるにあたりご討論ならびにさまざまなコメントを頂きました廣川直氏（JAIST）に感謝いたします。

目 次

第1章　はじめに

電子メールの 90%〜95%をスパムメールが占める [1] 。スパムメールは独特の形式や文字列を含むことが多いため、これらを検出する規則を記述することにより判別することは可能である。この手法の代表例としてヒューリスティックフィルタがある。ヒューリスティックフィルタを用いた手法ではメールヘッダや本文からメッセージを解析し、そこから得られたスパムメールの特徴などをスコア化し、スコアが一定以上の基準値を超える場合にスパムメールとして判断する。しかし、人間がルールを記述していかなければならず、誤認識を避けるためには膨大な時間や労力が浪費され、管理上の負担が重くなる。

これに対し、近年では統計的手法に用いた手法として、学習を行うベイジアンフィルタ [2] が挙げられる。この手法ではベイズの定理に基づき、確率的にスパムメールかそうでないかを判断する。しかし欠点として、問題となりそうな単語を人間の読み取れる程度の誤字で表現するなどして錯乱を引き起こしたり、新出タイプのスパムメールの学習に時間がかかるなどの問題点が指摘されている [3]。現在のスパムルフィルタは、前述した技術要素を組み合わた複合フィルタが主流となっている [4]。

本研究では、大規模な文書データベースの連想検索において有効な連想計算を用いた新たなスパム判別法として、連想スパムフィルタの構成法を提案し、それぞれ数万件のスパムメールからなる二つのデータセットをレファレンススパムメール群として用いた実験によりその可能性を探る。この手法は、既存手法と異なり、経路情報などは一切使用せず、メールの自然言語部分（Subject と本文）のみによるリファレンススパムメール群との類似度を評価し、スパム判別を行う。実験には、Webcat Plus [1]、新書マップ [2]、ほんつな [3] などで用いられている連想検索エンジン GETA [4]を用いる。

現在のスパム判別の閾関数は、スパムメール群に対する単一の類似度（実数値）に対するメールテキストの異なり単語数との線形回帰のみの単純なものに限られるが、日本語と英語に分割すると、スパムメール群と必要なメール群の傾向が顕著に分離し、有効で効率的な一次フィルター（または最終段フィルター）としての可能性を示している。

本稿では 2 章で連想計算、3 章で提案する連想スパムフィルタの構成法、4 章で GETA を含む実験システム構成、5 章で実験による閾関数の推定、6 章で実験結果を述べる。最後に 7 章で考察を行い 8 章でまとめを行う。

[1] http://webcatplus.nii.ac.jp/
[2] http://bookshelf.shinshomap.info/
[3] http://www.hontsuna.com/pages/skensaku/manual/renso
[4] http://geta.ex.nii.ac.jp/

第2章　連想フィルタ

2.1　連想計算

$\mathcal{P}(X)$ は X の空でない部分集合の集合、 $\mathcal{M}(X)$ は X の空でない多重集合 (x から自然数 $\mathbb{N}$ への関数) の集合を表す。

定義 2.1.1. 次の要素からなる 4 組を $A = (ID_1, ID_2, a, f)$ を 連想システム という

- 2 識別子の集合 ID_1、 ID_2
- 相関関数 $a : ID_1 \times ID_2 \to \mathbb{N}$
- 評価関数 $f : ID_2 \times \mathcal{P}(ID_1) \to \mathbb{R}_{\geq 0}$

A に対する連想計算 $\hat{A}$ は次の関数で定義される。$X \subseteq ID_1$ に対し

$$\hat{A}(X, n) = select(n, \{(y, f(y, X)) \mid y \in ID_2\})$$

ここで自然数の対の要素 P に対し、$select(n, P)$ は P の要素 (y, v) のうち v が最大となる p を n 個集めてくる関数である。連想システム $A = (ID_1, ID_2, a, f)$ および評価関数

$$f^t : ID_1 \times \mathcal{P}(ID_2) \to \mathbb{R}_{\geq 0}$$

に対し、$A^t = (ID_2, ID_1, a^t, f^t)$ を A の転置 と呼ぶ。ここで a^t は $a^t(y, x) = a(x, y)$ とする。

連想計算は ID_1 の集合から連想される ID_2 の要素の順位付のリスト（重み付集合）を得る関数である。本来、連想計算は文書集合から類似文書を検索するため導入された [5]。ID_2 を文書集合、ID_1 を文書の集合に出現する単語の集合とする連想計算において、文書集合 X に対し

$$\hat{A}(\{y \mid (y, v) \in \hat{A}^t(X, n_1)\}, n)$$

により、文書集合から文書集合への連想計算が定義される。ただし n_1 は定数であり、通常 $n_1 = 200$ 程度とすることが有効であることが実験的に知られている。このとき連想計算は、文書中の語順などは無視し、単語頻度に基づく適切な尺度のみに基づき、類似度を計算している。

2.2 連想フィルタ

連想フィルタ は連想計算に閾値判定を設けることによって実現する。$A = (ID_1, ID_2, a, f)$ を $ID_2 \subseteq \mathcal{P}(ID_1)$ を満たす連想システムとする。$\mathcal{P}(ID_1) \to \mathbb{R}_{\geq 0}$ の関数を A の閾関数と呼び F と記す。連想システム A と閾関数 F の対を連想フィルタと呼ぶ。

$$\{Y \in \mathcal{P}(ID_1) \mid \hat{A}(Y, 1) \geq F(Y)\}$$

第3章　連想スパムフィルタ

連想スパムフィルタを構成する上で

- メールの中の自然言語部分 (文書) のみに注目し、文書に対し、機械的な前処理と連想フィルタを用いてスパム判別を行う。
- 連想計算エンジンとして GETA を用いる。
- 閾関数を統計的実験により定める。

を前提とする。さらに機械的な前処理は可能な限り既存ツールを利用可能とするように、前処理の機能分割を行う。

本構成法における連想フィルタでは、

- ID_2 を文書集合（メールから抽出された文書の集合)、
- ID_1 を ID_2 に現れる単語の集合、
- **相関関数** $a(x, y)$ は文書 $y(\in ID_2)$ に現れる単語 $x(\in ID_1)$ の出現頻度

とする。ただし、**評価関数** f については、連想計算エンジン GETA のライブラリを用いているため 4 章において詳述する。

前処理部の構成

連想スパムフィルターの前処理は

1. 各メールから自然言語部分（ヘッダー中の Subject 部とメール本文）の抽出
2. 文字コードを EUC に変換
3. メール集合を類別
4. （日本語メールの場合）各メールの形態素解析器を用い単語分割し、文書（単語の多重集合）へ変換
5. 各メールの抽象化

からなる。ここでメール集合の類別とはメール集合の分割をさし、メールの抽象化とは、メール単体の構成要素のうち、どの部分を抽出するか、また抽出された部分に対する処理をいう。

閾関数の構成

閾値推定は、もっとも単純な場合の有効性を評価するため、メール内の単語数（重複は数えない）と類似度を入力とする線形回帰を用いる。その際、スパムメールのデータセットと必要なメールのデータセットに対し、最小二乗法によりそれぞれの中心線を計算し、その中間線を閾関数とする。

第4章　実験システム構成

連想スパムフィルタは、前処理機構と連想フィルタ、閾値判別機構の3つからなる。閾値判別機構は現状では未実装であり、連想スパムフィルターの可能性を評価するための実験システムを以下に示す。前処理部においては、形態素解析器を除くと perl ならびに shell script で記述し作成した。

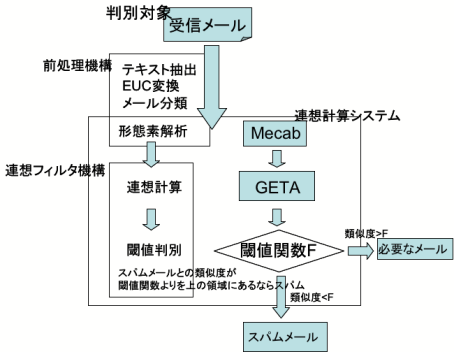

図 4.1: システム構成

形態素解析

形態素解析器は Chasen[1]、KAKASHI[2]、Mecab[3] が有名であるが本システムではもっとも処理速度の速い Mecab を採用した。Mecab の特徴としては、Chasen や KAKASHI よりも 3-4 倍程度高速であることが報告されている [7]。

[1] http://chasen.naist.jp/hiki/ChaSen/
[2] http://kakasi.namazu.org/index.html.ja
[3] http://mecab.sourceforge.net/

連想計算エンジン GETA

GETA は文書間の連想計算を行うことを想定して設計されており、ID_2 を文書（単語の多重集合）の集合、ID_1 を ID_2 内の文書に現れる単語すべての集合として用いることが一般的である。このとき相関関数 $a(x, y)$ は、文書 y 内の単語 x の出現頻度となり、行成分を文書内の単語頻度ベクトル、列成分を単語の文書での出現頻度ベクトルとする行列で表現される。これを WAM (Word Artcle Matrix) とよぶ。一般に文書数が膨大な集合の WAM は巨大疎行列のため、文書からの検索 $(a^t(y, x))$、単語からの検索 $(a(x, y))$ に対応し、それぞれを圧縮して別個のデータ構造としている。これにより WAM をメモリ上に保持することを可能となり、GETA の高速化のキーとなっている。

GETA における連想計算の実現の効率化のための制限として

- 単語集合を ID_1 文書集合を ID_2 とすると単語集合 ID_1 の要素である単語を含む文書が ID_2 の中に必ず存在する。
- 文書集合 ID_2 の要素である文書には必ず ID_1 の要素である単語が存在する。
- 評価関数はある単語集合 X 内のすべての単語 x が文書 y に現れなければスコアが 0 である。

が前提となっている。

GETA においては、ユーザが連想計算における文書 d と単語集合 X の間の評価関数 $f(d, X)$ を定義することが可能である。WAM においては、

$$\begin{aligned} \text{文書} &= \text{単語の多重集合（単語の頻度ベクトル）} \\ \text{単語} &= \text{文書の多重集合（文書の頻度ベクトル）} \end{aligned}$$

とみなす双対な関係があるため、単語 w と文書集合 Y の間の評価関数 $f^t(w, Y)$ は同じ評価関数 $f(w, Y)$ を用いることが可能である。

GETA ライブラリでは TF,SMARTmeasure が提供されているが、これらの評価関数は本来、文書と文書の類似度を計算するものであり、単語の集合を仮想的に文書（単語の多重集合）とみなすことで実装されている。たとえば、SMARTmeasure では、単語の重みをユーザが定義すること[4]で、単語集合をすべて多重度 1 の単語の多重集合（文書）とみなしている。

[4]通常は各単語の重みは 1 と設定することが多い

第5章　閾関数推定のための統計的実験

5.1　実験における WAM の構成

実験において試みた *WAM* の構成の 3 つの比較軸について述べる。

メールの類別

スパムメールの判定は、本来、多岐にわたる条件で定まるものであり、単一の類似度だけに基づく連想スパムフィルタで十分な精度が得られるかどうかは定かではない。そのため、連想スパムフィルタの多次元化と対象集合の制限にもどづく精度向上をめざし、メール集合の機械的操作による類別を、統計的実験の一つの軸とした。具体的に類別は、

1. 日本語メールと英語メールの類別
2. メールの重複の保存と重複の除去

とした。

メールの抽象化

対象とするメールは、ヘッダー、本文、添付ファイルなどから構成される。ここではメールの自然言語部分（文書）のみを対象にし、添付ファイルは除外する。さらにヘッダー内の経路情報なども除外し、人間が最初に読むであろう Subject 部と本文文書のみに注目する。具体的に抽象化は

1. （日本語メールの場合）全品詞の単語抽出と名詞のみの抽出
2. （英語メールの場合）\ # { } _ - などの区切り記号を保存と区切り記号の除去
3. メールの本文と `Subject` に分類し比較

とした。

データセット:	受信時期	総件数	総単語数	重複無日本語	重複無英語
α :	2006.01〜2007.12	81,299	1,058,976	945	12,963
β :	2005.10〜2008.04	152,640	2,378,976	21,033	49,477
γ:	2006.10〜2008.03	904		789	115

表 5.1: データセット

評価関数

GETA のライブラリで提供されている

1. TF (Term Frequency), 単語の文書集合中の出現頻度
2. SMART measure [8]

に基づく評価関数を比較した。これらの類似度の定義は補遺を参照されたい。

5.2 実験環境

実験で用いた機器は以下の通りである。

OS: Mac OS X ver.10.3.9
プロセッサ: 1.33GHz PowerPC G4
メモリ: 768MB DDR SDRAM
GETA: (第３版：複数ＣＰＵ用)

実験で用いたデータセットとして、SPAM メール群を二つ（データセット α および β)、必要なメール群（データセット γ) を一つ用いる（表１)。なお、この３つのデータセットの受け取り手は異なる。

以下では、データセット α, β をリファレンススパムメール群として、それぞれから連想スパムフィルタの WAM を作成する。指定された WAM による類似度の実験結果は散布図で表示されるが、

- 散布図内の水色の点がデータセット α、
- 緑色の点がデータセット β、
- 桃色の点がデータセット γ、
- 散布図の x 軸は単語数、y 軸は類似度

とする。また、各選択における WAM の表記として、$WAM\,(d, l, t, r, s)$ を用いる。ただし、

- d データセットの種類 { α, β, γ } と表す
- l 言語の種別 { 日 (本語), 英 (語) } を表す
- t テキストの種類 { `B(ody)`, `S(ubject)` } を表す
- r 重複の { 有, 無 } を表す
- s 付加情報は、日本語メールについては { 名 (詞), 全 (品詞)}、英語メールについては { (記号) 保 (存)、(記号) 除 (去) } を表す

とする。なお、節 5.1 の軸に従いデータセットを処理した場合の諸データをデータセット α については表 5.2, データセット β については表 5.3 に示す。

WAM：	JB 無全-	JB 有全-	EB 無-除	EB 有-除	EB 無-保	-S 有全有保
文書数:	945	26,737	12,960	48,935	12,960	81,299
単語数:	122,420	122,420	641,560	807,556	642,812	29,057
WAM 作成時間:	6s	24s	15s	82s	20s	11s

表 5.2: データセットに基づく α の各軸

WAM：	JB 無全-	JB 有全-	EB 無-除	EB 有-除	EB 無-保	-S 有全有保
文書数:	21,033	38102	49,477	67,463	49,477	152,640
単語数:	139,660	139,660	1,41,253	1,474,960	924,960	61,531
WAM 作成時間:	14s	16s	77s	83s	61s	32s

表 5.3: データセット β の各軸

第6章　実験結果

6.1　本文と Subject の比較

図 6.1,6.2 に各メールの本文から作成した WAM、図 6.3,6.4 に各メールの Subject から作成した WAM を、それぞれデータセット α と γ、データセット β と γ に対し連想スパムフィルタを適応した散布図を示す。

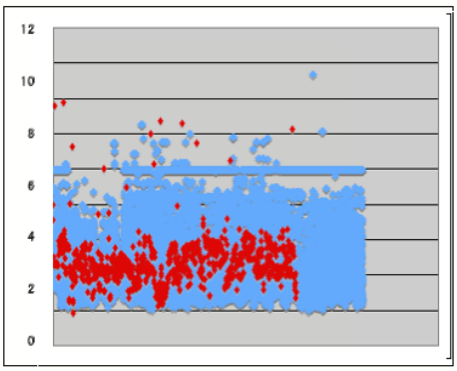

図 6.1: WAM(α, 英日, B, 有, 名) に対する α,γ の散布図

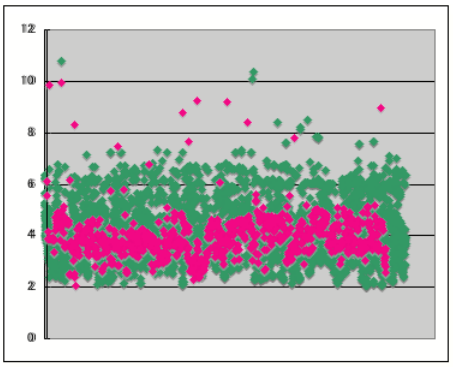

図 6.2: WAM(β, 英日, B, 有, 名) に対する β,γ の散布図

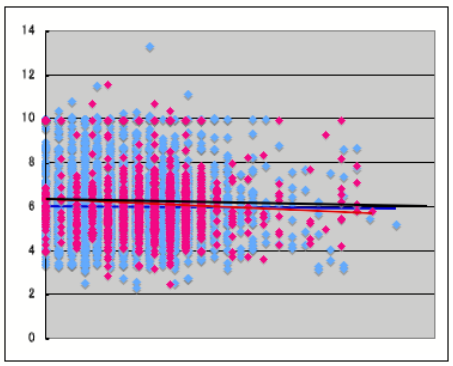

図 6.3: WAM(α, 日英, S, 有, 全) に対する α,γ の散布図

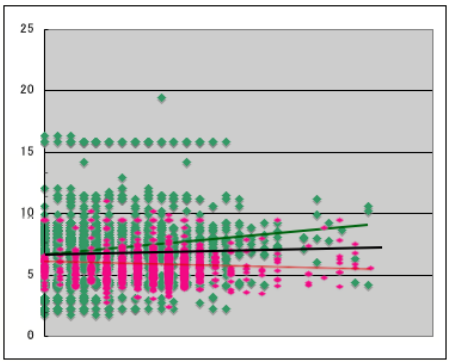

図 6.4: WAM(β, 日英, S, 有, 全) に対する β,γ の散布図

いずれの場合も十分な分離がなされていないが、いずれのデータセットにおいても本文の方が Subject より明確な傾向を示している。これは Subject は 30 語数以下であり、

本文の単語数が、より多いためと考えられる。このため、以下では本文を対象に実験を行うが、日本語メールは出会い系、英語メールは Rolex, Viagra, 各種ソフトウェアの海賊版（？）とおぼしき商用メール、と傾向が分かれている。そのため、英語と日本語にメールを類別することを最初の方針とする。

6.2 日本語メールに対する実験

メール集合の日本語と英語の判定の方法として文書を 1 行ずつ読み取り、$ASCII$ コード以外の文字コードある場合、日本語と判断した。$ASCII$ コードでなければ日本語を含むとし分離した。なお、本節での各散布図の α 横軸は 0 語数から 500 語数である。β の横軸は 0 語数から 1500 語数である。γ 横軸は 0 語数から 1500 語数である。

名詞のみと全品詞の場合の比較

単語分割の際に、名詞は重要性が高いと考え, 名詞のみと全品詞の 2 つの WAM を作成し比較した。図 6.5 ,6.6 に名詞のみの WAM、図 6.7 ,6.8 に全品詞の WAM を、それぞれデータセット α と γ、データセット β と γ に対し連想スパムフィルタを適応した散布図を示す。

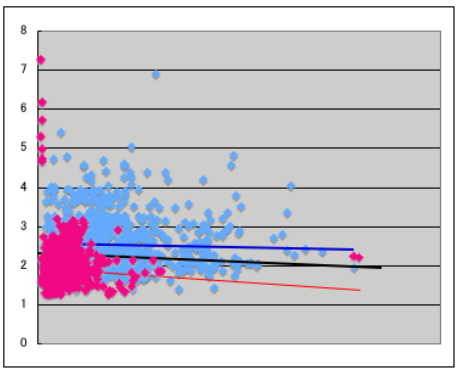

図 6.5: WAM(α, 日, B, 有, 名) に対する α,γ の散布図

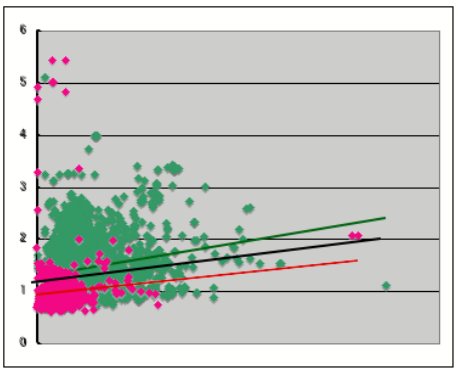

図 6.6: WAM(β, 日, B, 有, 名) に対する β,γ の散布図

散布図に顕著な差は見られないが、データセット β と γ においては若干全品詞の方が分離できている。

重複有-重複無

図 6.9,6.10 に日本語の重複を取り除いた WAM を、それぞれデータセット α と γ、データセット β と γ に対し連想スパムフィルタを適応した散布図を示す。

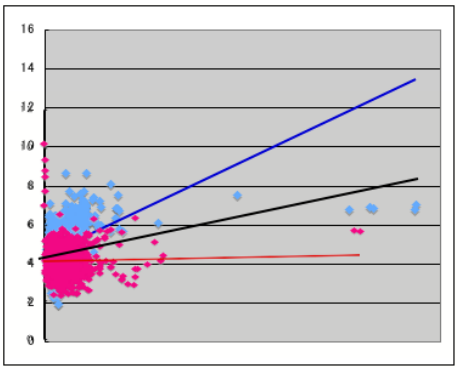

図 6.7: WAM(α, 日, B, 有, 全) に対する α,γ の散布図

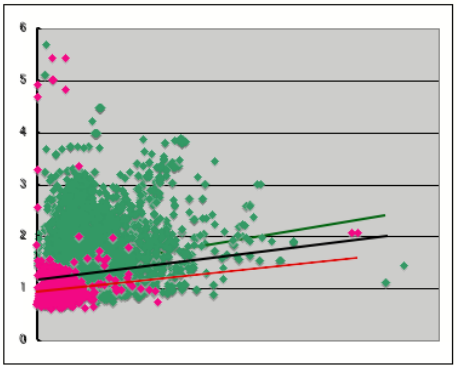

図 6.8: WAM(β, 日, B, 有, 全) に対する β,γ の散布図

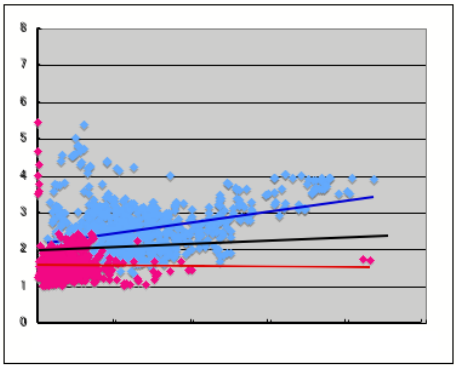

図 6.9: WAM(α, 日, B, 無, 全) に対する α,γ の散布図

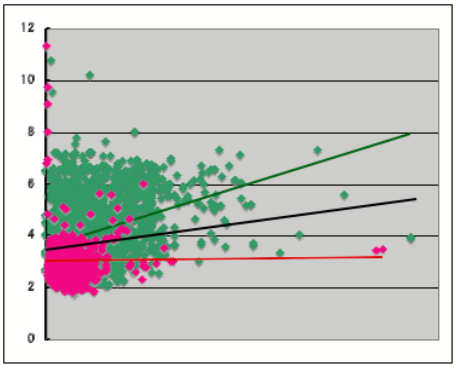

図 6.10: WAM(β, 日, B, 無, 全) に対する β,γ の散布図

データセット α と γ は日本語はパターンが一緒のメールが多く件数が激減したが、顕著な改善がみられる。これから日本語メールの場合、重複除去は有効と思われる。

6.3 英語メールに対する実験

本節での各散布図の横軸は α は 0 語数から 12000 語数である。β は 0 語数から 5000 語数である。γ は 0 語数から 800 語数である。

記号の保存-記号の除去

図 6.11,6.12 に記号を保存した WAM、図 6.13,6.14 に記号を除去した WAM を、それぞれデータセット α と γ、データセット β と γ に対し連想スパムフィルタを適応した散布図を示す。

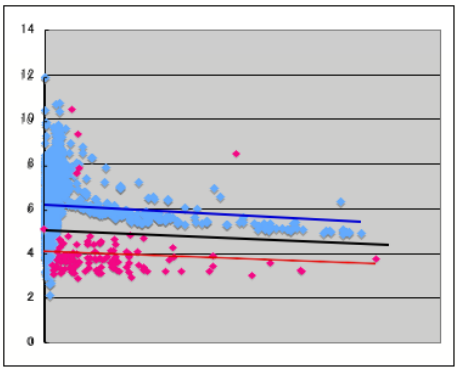

図 6.11: WAM(α, 英, B, 有, 保) に対する α,γ の散布図

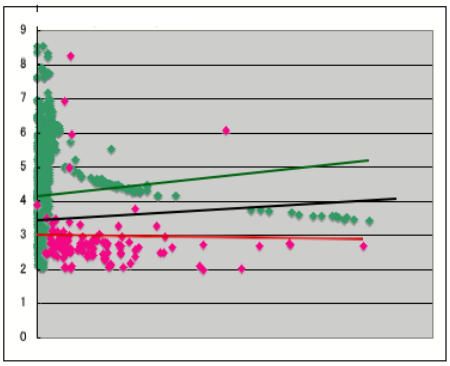

図 6.12: WAM(β, 英, B, 有, 保) に対する β,γ の散布図

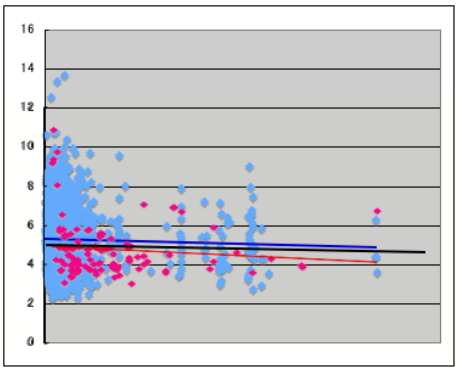

図 6.13: WAM(α, 英, B, 有, 除) に対する α,γ の散布図

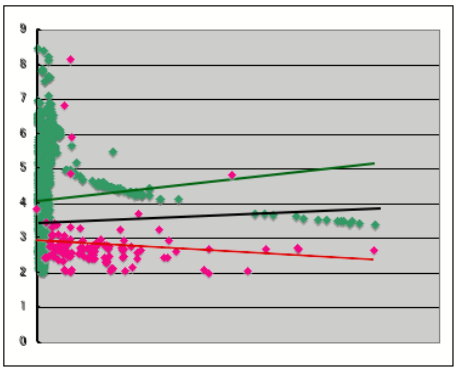

図 6.14: WAM(β, 英, B, 有, 除) に対する β,γ の散布図

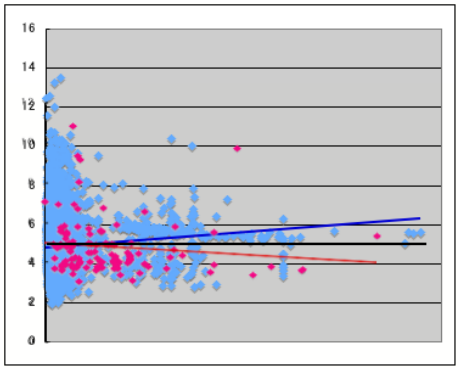

図 6.15: WAM(α, 英, B, 無, 除) に対する α,γ の散布図

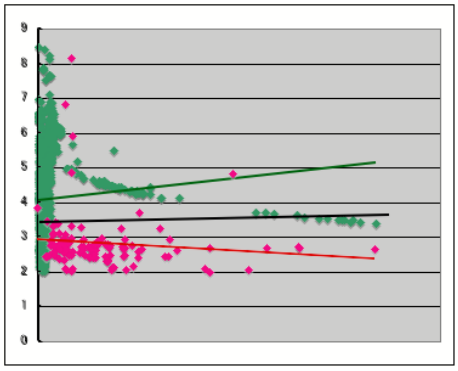

図 6.16: WAM(β, 英, B, 無, 除) に対する β,γ の散布図

英語メールの場合、重複の有無で顕著な差は見られないが、一つにはもともと重複が多くなかったためと考えられる。

6.4 類似度関数による違い

データセット α と γ に対し、日本語メールの全品詞・重複除去を行った場合の類似度の評価関数の比較を行った。評価関数は TF、Singhal の SMART measure の 2 つである。図 6.17 は `TF`、図 6.18 は `SMART measure` を用いた連想スパムフィルタを適応した散布図を示す。

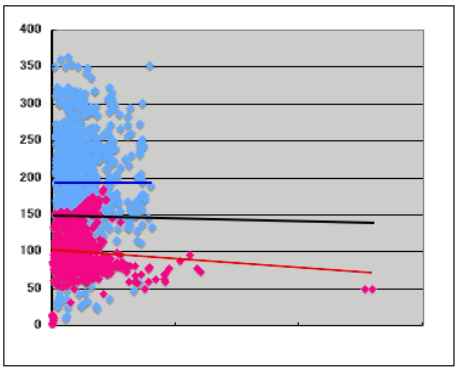

図 6.17: WAM(α, 日, B, 無, 全) における TF に対する α,γ の散布図

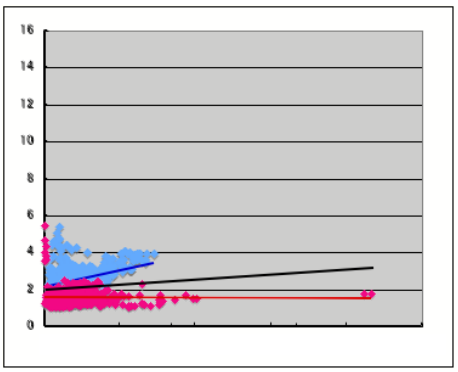

図 6.18: WAM(α, 日, B, 無, 全) における smart measure に対する α,γ の散布図

若干傾向が異なる結果が出たが、全般的には Smart measure の方が分離は良い。ただし、長さが短いメールに対しては TF の方がデータセット γ 内のメールに対し適切な類似度の切り分けを示している。

第7章　考察

7.1　妥当性の検証

妥当性の検証は、異なるデータセットにより作成されたWAMを適応し行った。

- データセットαから作成したWAMをデータセットβ, γに適用した結果（妥当性検証1）が図7.1
- データセットβから作成したWAMをデータセットα, γに適用した結果（妥当性検証2）が図7.2

である。（どちらも、重複を除去した日本語メールの本文の全品詞を対象とした。）

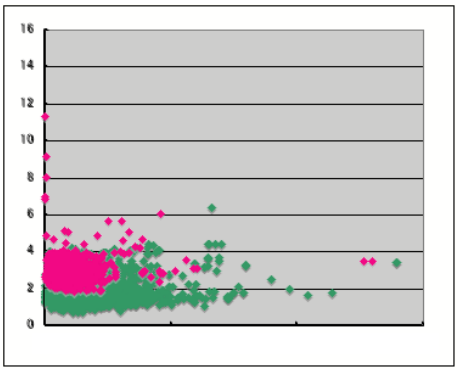

図 7.1: α の連想フィルタに対しての β,γ の散布図

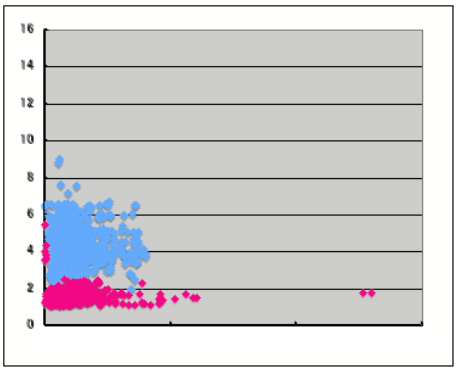

図 7.2: β の連想フィルタに対しての α,γ の散布図

この結果、妥当性検証 1（図 7.1）では大きく精度が低下し、妥当性検証 2（図 7.2）では良好な分離を示している。一つの可能性として、同じデータセットで作成された WAM を用いると入力となるスパムメールと同じメールが最上位の類似度となることが考えられるが、実際にはその影響は限定的である。実際、リファレンススパムメール群に含まれる入力スパムメールに対し、同じメールが最上位の類似度となる比率は

- データセット α で作成した WAM でデータセット α を判定した場合、17.50%,
- データセット β で作成した WAM でデータセット β を判定した場合、31.03%,

に限られた。妥当性検証 1（図 7.1）における精度低下の原因はいまだ不明であり、受け取り手により受け取るスパムメールの傾向が異なることなどの理由が推察されるが、さらなる実験による検証が必要である。

7.2 有効性

日本語メールに関しては

- 名詞のみと全品詞では（データセット α, β ともに）顕著な差はみられない。
- 全品詞においてメールの重複の有無は、データセット α において改善が見られ、データセット β においては大きな傾向の差はみられない。

となっている。日本語メールでもっともよく分離された場合は、重複除去を行い本文の全品詞を対象とした場合である。これは日本語スパムメールでは出会い系のメールが多く、同じカテゴリや様式のメールが多かったためと思われる。なお、このときにスパムメールと誤認される極端に高い類似度を示したデータセット γ（必要なメール群）のメールは、

多くが極端に短いものであり、単語数が極端に短いものは、Smart measure のかわりに TF を用いるなど別処理を行う必要があると考えられる。

英語メールに関しては

- 記号を保存した場合と除去した場合は、データセット α において記号を保存した方が顕著に分離されている。データセット β においてはいずれの場合も良好な分離が観察される。
- 記号を除去した場合、データセット α, β ともに重複の有無に顕著な差は見られない。

となっている。英語メールでもっともよく分離された場合は、記号除去・重複除去ともに行わなかった場合（記号除去を行わず重複除去を行った場合は未評価）であり、特に記号除去を行わないことの効果が（予想に反し）顕著である。これは英語メールにおいては特有の記号の使い方が大きな影響を与えていると推測される。

7.3 今後の改善の可能性

スパムメールの多くは題名は普通のメールを装い、本文は出会い系や商用のメールというパターンが多い。

約 2 週間程度にわたり頻繁（100 通程度）に商用スパムフィルターを通過してきた同一送り元のスパムメール（出会い系）は、通常のスパムメールと Subject（1 行程度）と本文（1〜2 行程度で非常に短い）が逆転したようなメールであったが、これはデータセット α の本文から作成した WAM で Subject に対し適用したところ、半数以上が高いスパム類似度を示した。

このことから、複数の WAM を組みあわせたスパム判別が有効であることが予想される。たとえば日本語メールについては、類似度が最上位となるスパムメールが特定のメールに偏る傾向がみられる。（英語メールの場合は未調査。）これを利用し、類似度が頻繁に最上位となるスパムメールを指標スパムメールとして、その特定のメールとの類似度が高いスパムメールで WAM を複数作成するなどの手法が考えられる。

第8章　まとめと今後の課題

本稿では、連想計算にもとづく連想スパムフィルタを提案し、汎用連想計算エンジンGETAを用いて、それぞれ数万件からなる二つのスパムメールのデータセットを用いた実験を行った。その結果、日本語メールについては全品詞・重複無の場合に英語メールでは記号保存・重複有の場合にスパムメール群と必要なメール群の傾向が分離する傾向が明らかであり、有効で効率的な一次フィルター（または最終段フィルター）としての可能性が示された。

連想スパムフィルターのセットアップ（WAMの作成）かなりの長時間がかかっているが、これには二段階あり、WAM作成の前処理としてのメール群の単語頻度情報作成、ならびにWAMの生成がある。前者は現状の実装では重い処理だが、一度行えば新たなリファレンススパムメールの追加処理は漸進的であり、大きなオーヴァーヘッドにはならない。また、WAMの作成は漸進的にはできないが10万件程度のスパムメールであれば、数十秒程度で終了する。これらのセットアップができていれば、入力メールに対する連想スパムフィルターの処理は0.3秒以下であり、効率的に行うことができる。

現在の閾関数は、スパムメール群に対する単一の類似度（実数値）に対するメールテキストの異なり単語数との線形回帰のみの単純なものに限られており、より適切な閾関数の設計や複数のWAMを組みあわせたスパム判別などによる改善が期待される。たとえば日本語メールについては、類似度が最上位となるスパムメールが特定のメールに偏る傾向がみられる。これを利用し、類似度が頻繁に最上位となるスパムメールを指標スパムメールとして、その特定のメールとの類似度が高いスパムメールでWAMを複数作成するなどの手法が考えられる。また英語メールについては、図6.11,6.12などのように良好な分離を示している場合でも、閾関数として二次関数や二つの線形回帰の組み合わせなどがが必要であり、これらが今後の課題である。

参考文献

[1] 勝村幸博 , 「メールの 95 %は『迷惑メール』だった」、2007 年のスパム動向 *ITpro.* 日経 *BP* 社. (2007-12-14) http://itpro.nikkeibp.co.jp/article/NEWS/20071214/289521/

[2] Paul Graham, A Plan for Spam,2002.

[3] 田端利宏, SPAM メールフィルタリングベイジアンフィルタの解説 Trees. 情報の科学と技術, Vol. 56, No. 10 (2006), pp. 464–468.

[4] 安藤一憲, spam メールの現状と対策の動向 -Special Feature on Anti-Spam Activities - *IPSJ Magazine*, Vol. 46 No. 7 July (2005), pp. 739–791.

[5] 西岡真吾 , 汎用連想計算エンジン GETA の実装 http://brandenburg.cs.nii.ac.jp/ nis/geta32.pdf

[6] 高野明彦, 西岡真吾, 今一修, 岩山真, 丹羽芳樹, 久光徹, 藤尾正和, 徳永健伸, 奥村学, 望月源, 野本忠司, 汎用連想計算エンジンの開発と大規模文書分析への応用 http://geta.ex.nii.ac.jp/pdf/itx2002.pdf

[7] 宮崎 真, 廣安 知之, 三木 光範 , MeCab の基礎 http://mikilab.doshisha.ac.jp/dia/research/report/2005/0813/011/report20050813011.html

[8] A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *Proceedings of SIGIR'96*, (1996), pp. 21–29.

本研究に関する論文投稿

石黒雄輔, 小川瑞史, 「汎用連想計算エンジン GETA を用いたスパム判別」日本ソフトウェア科学会第 25 回大会,Sep. 2008(投稿中)

文書と文書の類似度の評価関数

評価関数は TF、TF-IDF、Singhal の SMART measure の 3 つである。

- D は文書集合
- q, d は文書
- t は単語
- $TF(t|d)$ を単語 t の文書 d における出現頻度
- $DF(t)$ は単語 t の出現する D 内の文書数
- N は D 内の文書総数
- $ave(TF(\cdot \mid q))$ は q における $TF(t \mid q)$ の平均
- $len(d)$ は d 内の異り単語数

とする。暗黙に、q は検索クエリ、d は文書群 D 内の文書であることを想定する。

TF

$TF(d|q)$ は単語 t の文書 d,q における出現頻度のみに依存する。計算式は以下の通りである。

$$TF(d|q) = \frac{1}{len(d)} \Sigma_t \, TF(t|d) \cdot TF(t|q) \tag{1}$$

Singhal の SMART measure

Singhal の方法は、文書の長さによらず適切な類似度を求めることを目的として設計されている。基本的なアイデアは単語ごとに検索語の TF、及び検索対象の TF-IDF の積を計算し足し合わせ、文書の長さにより正規化を行う。TF-IDF は、文章中の特徴的な単語（重要とみなされる単語）の抽出を目的として設計されており、主に情報検索や文章要約などの分野で利用される。TF-IDF は、TF（単語の出現頻度）と IDF（逆出現頻度）の二つの指標で計算される。計算式は以下の通りである。

$$IDF(t) = \log \frac{N}{DF(t)} \tag{2}$$

文書 q と文書 d の類似度 $sim(q|d)$ の計算式は以下の通りである。

$$wq(t|q) = \frac{1 + log(TF(t|q))}{1 + log(ave(TF(q)))} \times IDF(t) \tag{3}$$

$$wd(t|d) = 1 + log(TF(t|d)) \tag{4}$$

$$norm(d) = (ave(len(d)) + 0.2 \times (DF(\cdot|d) - ave(len(d)))) \times (1 + log\frac{TF(\cdot|d)}{DF(\cdot|d)}) \tag{5}$$

$$sim(q|d) = \frac{1}{norm(d)} \times \sum_{t \in q} \{wq(t|q) \times wd(t|d)\} \tag{6}$$