システムログの意味抽出のための自然言語処理的アプローチによる解析手法の検討

小林論 [1]　福田健介 [2]　江崎浩 [1]
Satoru Kobayashi　Kensuke Fukuda　Hiroshi Esaki

東京大学大学院情理工学系研究科 [1]
The University of Tokyo

国立情報学研究所 [2]
National Institute of Informatics

概 要

システムログから機械的にトラブルの原因究明を行うには、ログ中の文脈についての意味抽出を行う必要がある。解析にはシステムログを出力形式パターンから分類することが必要となるが、ログ文字列を信号処理的に扱うことで高速化を試みている既存手法ではトラブルの原因究明という目的を満たすだけの精度を確保することが難しい。そこで本研究ではシステムログの記述の特徴を元に自然言語処理的な手法を適用し、それにより精度の問題を解決することを目指す。

1　背景

サーバやスイッチ、ルータ等のネットワークノードの管理運用において重要視されるものの１つとして、システムログが挙げられる。システムログはネットワーク上にアクセスや変更、トラブルが発生した際に記録され、一般には管理用のサーバ等に送られ集中管理される。これらのログは主にトラブルの発見とそのデバッグ、原因究明を行う為に用いられる。

しかし、システムログから発生したトラブルの状況を把握しその原因を見つけ出すにはオペレータの知識と経験が必要となり、また時間的なコストも大きい。同時に、これらのログから機械的な処理により原因の手がかりを得るのは簡単ではない。ログのそれぞれの行は発生した事実を示すのみであり、原因の究明にはそれらのログの文脈、つまりログ行同士の前後関係やそのログが出力された背景となる機器の設定などの情報が不可欠である。

本研究ではシステムログから機械的にトラブルの原因究明を行うのに必要な情報のうち、ログ行同士の関連性の情報を得ることを目的に、システムログの大規模集積データの解析を行う。特にトラブルを始めとした低頻度かつ突発的なログ出力についても十分な精度を確保する為、システムログの特性を踏まえた自然言語処理的なアプローチによる手法を検討する。

2　要件

本研究では、国立情報学研究所で運用されている学術情報ネットワーク (SINET) において、ネットワークを形成するスイッチ及びルータから出力された 2010 年 4 月から 2013 年 3 月までの 3 年分の syslog 形式のシステムログ (11GByte) を対象に解析を行う。

ログ行同士の関連性の情報を得る為に、以下の手続きが必要となる。

1. ログ行を出力形式を元に機械的に分類し、ログパターンとして定義する
2. ログ行ごとにその日時と該当するパターン ID からなる時系列データとして再構成する
3. 時系列解析を行い、各ログパターン同士の関連性を評価する

このうち本稿では特に 1 について効率的かつ高精度に行う為の手法を検討する。

3　既存手法

システムログを解析する手法は複数存在する。Vaarandi[1] は、システムログを構成する各単語はログ出力形式に共通する「記述部」が、具体的な値を示す「変数部」よりもログ全体における出現回数が多いという推定に基づくアルゴリズムを提案した。各ログ行を形成する文字列を単語に区分し、それぞれの単語のログ全体における出現回数をカウントする。そして、予めユーザ側で指定した閾値を超える出現回数の単語を記述部、閾値に満たないものを変数部とみなす。最後に記述部のみを組み合わせることで、各ログパターンの出力形式の共通部分が得られる。この手法の利点は、高速にログを処理することが可能なことであるが、一方で閾値の設定がログ全体を通して一定であるため精度に欠けるという問題点が存在する。

また Xu[2] は、システムログの出力がオープンソースのプログラムにより行われている場合にそのソースコードを解析することでログ出力形式の共通部分を得るという手法を提案した。出力されるログ自体を解析するよりも効率的にログパターンを得ることが可能であるが、本研究で対象としているようなネットワーク機器は一般的にオープンソースではないため、この手法を適用できる環境は大きく限られてしまう。

4　提案手法

4.1　解析データの考察

システムログから出力形式に基づくログパターンを抽出することは、ログ行を構成する各単語が「記述部」か「変数部」かを判別する問題に帰結する。そこでまず、変数部として考えられる単語の書式を正規表現として予め整理し、それにより解析データにおける記述部と変数部の分布の状況の把握を試みる。この方法は誤入力やログ出力のバグ等への対応性に欠け、また自由出力によるシ

ステムログの変数部表記に対してこの書式を効率的に用意するのは難しく、汎用的な手法としては実用性に欠ける。しかし解析データ中の記述部と変数部の分布状況を知るには有用である。

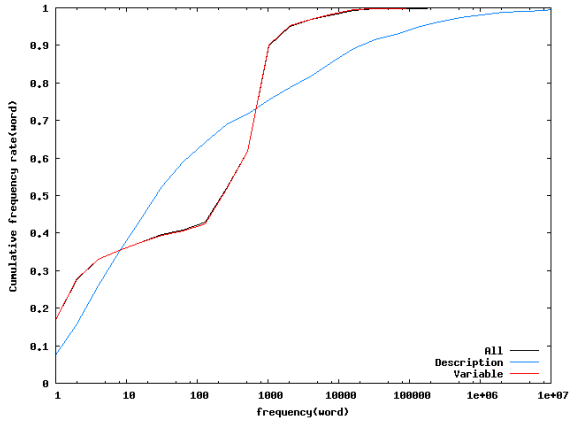

図 1　単語の出現回数分布

この手法で得られた記述部と変数部の単語の出現回数分布を累積相対度数で表すと図 1 のようになる。記述部と変数部には出現回数の傾向に差があり、特に変数部はデータの範囲が約 1000 日分であることから 1000 回前後の出現回数の単語が多くなっている傾向が見られる。しかし一方で Vaarandi のアルゴリズムの前提となる「記述部は変数部よりも出現回数が多い」という推定は、記述部の単語の出現回数に大きな幅があることから常に成り立つものとは言えない。

例えばこの問題の簡易な対策として、Vaarandi のアルゴリズムを各ログ行を構成する単語のうち全体における出現回数が多いものの何割を記述部とみなすというように変更を加える方法がある。しかし、ログパターンごとにどれだけの単語の割合が記述部であるかは差があり、一概に何割を記述部とみなす、という方法で全てを満たすような閾値を決めることはできない。またこの場合でも「0」やアカウント名などの一部の極めて出現頻度の高い変数部については固定部と判断されてしまう。

本研究の最終的な目的はトラブルの原因究明にあり、そのためにはトラブルの際にのみ発生するような全体に対して出現頻度の少ないログ行のログパターンについてもある程度正確に検出できることが求められる。しかしこれらの Vaarandi のアルゴリズムの改善のみでは、出現頻度の低いログ行ほど相対的にそれらのログにしか含まれない記述部の単語も出現回数が低くなることとなり、精度が大きく低下してしまう。

4.2　提案手法

本研究ではシステムログ中の単語が記述部か変数部かをより高精度に判別する為、システムログの表記の特徴を用いた自然言語処理的なアプローチによる推定を試みる。システムログはヘッダ部分を除いて自由記述による出力が行われており、自然言語のシンプルなモデルとして自然言語処理の手法を応用することが可能である。

例として、以下のような推定が可能である。

1. 単語の前後関係による推定

システムログでは、特定の単語の直後には変数が来る等といった決まった組み合わせがよく見られる。例として、user という単語の直後には極めて高確率でアカウント名と見られる変数部の文字列が現れる。このようなシステムログの性質は、ログ行中の変数部の推定を行う上で有用である。

2. 単語間の記号による推定

Vaarandi のアルゴリズム等では簡単のためログは単なる単語の集合とみなしているが、実際のシステムログでは単語は多彩な記号列により区切られている。これらの記号列の種類によって、前後の単語の関連性がある程度推定可能である。例として、“=” で区切られた 2 つの単語がある場合、後者の単語は変数部である可能性が高い。逆に、単語が “(” などで区切られている場合はそれらの単語に前後関係があるとは考えにくい。

これらの推定を元に、教師あり学習による解析を試みる。一定範囲のログデータについて単語が記述部か変数部かを判別する正解データを与え、自然言語処理における条件付確率場 (CRF) のモデルに適用して学習を行う。これにより各単語や単語間の記号の組み合わせについての記述部または変数部である周辺確率が求まり、判別の上で定量的な評価基準とすることが可能になる。

これらの手法を Vaarandi のアルゴリズム等と複合的に用いることで、最終的により高精度なログパターンを得ることを目指す。4.1 で述べている正規表現により記述部、変数部を区別したデータを正解データとみなし、これらの手法を用いて記述部、変数部の判別を行った際の正解データに対する正答率の傾向の変化を評価することで、最適な組み合わせを模索する。

5　まとめ

本研究ではシステムログから機械的にログ行同士の関連性を得ることを目的に、システムログの出力形式を自然言語処理的な手法により取得する手法について検討した。提案手法で述べたそれぞれの単語推定手法について評価を行い、その最適な組み合わせを求めることで精度の問題の緩和を試みる。今後はこの手法により得られる時系列データを元に、要件 3 の時系列解析によりトラブルの原因究明という目的における各ログ行の重要性の評価や、ログ行同士の関連性を解析していく。

参考文献

[1] R.Vaarandi. A data clustering algorithm for mining patterns from event logs. In *IEEE IPOM*, pp. 119–126, 2003.

[2] W.Xu et al. Detecting large-scale system problems by mining console logs. In *ACM SOSP*, pp. 117–132, 2009.

[3] A.Medem et al. Troubleminer: Mining network trouble tickets. In *IFIP/IEEE IM*, pp. 113–119, 2009.