Webフォーラムの構文情報を用いたトラブルシュート文書抽出

栗田 光晴† 柴田 剛志† 田浦 健次朗† 近山 隆†
東京大学工学部‡ 東京大学大学院§ 東京大学大学院§ 東京大学大学院¶
情報理工学系研究科 情報理工学系研究科 新領域創成科学研究科

1 はじめに

インターネットが一般に普及して以来、Web は誰もが発信者たりうる場として、多種多様な情報の交換に用いられている。なかでも開発・配布の大部分をインターネットに依存するオープンソースソフトウェアやフリーソフトウェアにとって、Web 上のフォーラムなどは開発者・利用者双方にとって非常に重要な情報源となっており、ソフトウェアの利用者が何らかの不具合に遭遇した場合、それらの Web サイトが参照されることが多い。

しかし、現状の検索の枠組みでは、検索に際し他のユーザが同じ不具合に対して用いた用語そのものを検索語として選ぶことができなければ目的とする情報が取り出される可能性は低く、そのため考えられる範囲で同じ不具合を表現する別の単語を次々と入力するという事態に陥りがちである。

そこで本研究では、検索時に検索語によって高いマッチを得なかったトラブルシュート文書を拾い上げることを目的として、事前に収集されたトラブルシュート文書をもとに、Web 上に存在するそのほかのトラブルシュート文書を効率的に取得する方法を検討する。

2 既存手法とその問題点

既知のトラブルシュート文書をもとにしてそのほかのトラブルシュート文書を抽出するというタスクは一種の類似文書検索であり、その手法として一般的なものには、文書中の特徴語によって文書をスコアリングするものが挙げられる。

しかし、今回抽出対象とする文書は必ずしも事前に収集された文書と同じ分野のトラブルシュートであるとは限らず、そのため事前に取り出された特徴語に多く含まれる特定分野の特徴語によって、目的とする文書以外のものが高いスコアを付与されるおそれがある。

3 提案手法

上記のような問題に対処するために、既知のトラブルシュート文書の構文情報を用いた手法を提案する。

今回対象とするトラブルシュートはコンピュータの利用に関するものであり、その不具合の表現には具体的なソフトウェア名などによらず高い頻度で出現する言い回しが存在すると考えられる。

そこで、既知のトラブルシュート文に頻出する構造を取り出し、それらが対象とするドキュメントに含まれているか否かをトラブルシュートらしさの指標として用いる。

例として、

- I am having problems connecting to the Internet.
- I've been having weird problems with my sound card.

という 2 文が与えられた場合を考える。語単位の頻度を見ただけでは語同士のつながりを考慮することは不可能であり、また単純に N-gram での抽出を図ると have のような助動詞、weird のような修飾語などによって小さな部分構造しか取得することができない。

それに対し構文解析を行うと、「I am having problems」という構造がこれらの文に共通して出現するという情報が得られる。これらは上記 2 文の特徴語たり得る「Internet」や「sound card」よりもよりトラブルシュートらしさを反映した特徴であり、このような情報を用いることでより目的に合致した文書を取り出すことが可能となると考えられる。

本手法の流れとしては、まず既知のトラブルシュート文書から文中に頻繁に出現する構文を抽出する。しかる後に各頻出構文の非トラブルシュート文書における出現数を調べ、構文のレベルでの TF-IDF 値を算出し、これをもとにして検索対象ドキュメントのスコアリングを行う。以下でこの手順を説明する。

3.1 頻出構文の抽出

ここで言う構文情報とは、述語項構造によって結ばれるグラフのことであり、例えば、ノードが述語の場合、各エッジは、その意味上の主語や目的語を指し示している。本来述語項構造から作られるグラフは、述語および項それぞれの語の活用などの情報とそれらの関係のラベルとによる有向グラフとなるが、今回はこれを語の原形をノードとした、エッジにラベルのない無向グラフとして処理することとする。

既知のトラブルシュート文書に含まれる文を上述の無向グラフに変換し、それらのグラフに共通する部分グラフを得ることで、トラブルシュート文書に頻出する構造を取り出すことができる。

ここで、頻出部分グラフの抽出に必要とされる部分グラ

Troubleshoot Document Extraction Using Sentence Structures of Web Forums
† Mitsuharu Kurita, Takeshi Shibata, Kenjiro Taura, Takashi Chikayama
‡ Faculty of Engineering, The University of Tokyo
§ Graduate School of Information Science and Technology, The University of Tokyo
¶ Graduate School of Frontier Sciences, The University of Tokyo

フ同形判定は NP 完全であり、グラフの構造によっては非常に大きなコストを要する場合があるが、一文に含まれる単語数は高々数十程度でラベルが重複するノードが存在する可能性も高くなく、gSpan アルゴリズムによって現実的な時間で処理することができる。
さらに、得られた頻出部分グラフがトラブルシュート文以外の文書に出現する回数を調べ、これらの値を用いて頻出構文の TF-IDF 値を算出する。

3.2 ドキュメントのスコア

検索対象ドキュメントのスコアとしては、対象ドキュメントに出現した頻出部分グラフの TF-IDF 値の総和を用いる。

4 実験

構文解析器としては東京大学辻井研究室による Enju[1]を用いた。Enju は主辞駆動句構造文法 (HPSG) による構文解析器であり、それにともなって述語項構造を出力することができる。[2]

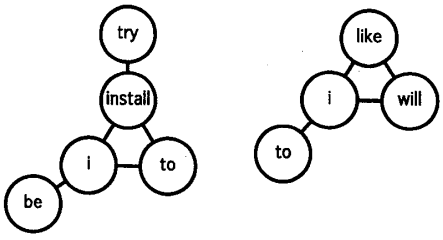
gSpan の実装としては、Katharina Jahn による Optimized gSpan[3] を一部改変して用いた。

トラブルシュート文書のサンプルとしては「Open Source and Linux Forum」[2]の Linux Forums に投稿されたトピック 48416 個から、各トピックの最初の発言の中で、タイトルに含まれる語の割合が最も多い一文を利用した。これは、トラブルシュート文書の主題として書かれた文には、トラブルに直面し困っている様子を記述する文が多く含まれ、共通した表現が現れやすいと考えられるためである。

IDF を算出するための非トラブルシュート文書としては、無差別なクロールで収集された 20964 ドキュメントを用いた。

5 実験結果

まず、上述の方法で得られた頻出部分グラフの一部を示す。



上記のグラフはそれぞれ「I am trying to install」、「I would like to」を表しており、トラブル報告に出現しやすい構造であるように思われる。これに対し語単位で TF-IDF をとった場合、「linux」、「suse」、「fedora」といった必ずしもトラブルシュート文書の特徴であるとは言えない語も高い TF-IDF 値を与えられている。

前述の通り、検索対象ドキュメントのスコアはその中に出現した頻出構造が持つ TF-IDF 値の和とする。ドキュメントの中には非常に長大なものもあるため、先頭 30 文のみを対象とした。評価に際してはトラブルシュート文書とそうでない文書が混在した文書集合を得る必要があるため、適当な検索語で Google に問い合わせた際の検索結果の上位 50 件を取り出し、それらのページに対するスコアを構文情報を用いた場合と単語の出現頻度を用いた場合とで比較する。

今回は、検索語として実際のトラブル事例をもとに、「vmware hp」という語の組を用いた。この際、Google による検索結果上位 50 件のうち、英語で書かれたトラブルシュート文書は 3 件であり、それぞれ 17、18、28 番目にリストアップされている。

提案手法でスコアを付与した場合、これら 3 つのドキュメントが最上位 3 件にソートされた。これに対し、語単位の TF-IDF で、同様に各ドキュメント内の語の TF-IDF 値の合計を用いた場合、1 位、3 位、11 位にそれぞれソートされる結果となった。単語でのスコアリングでは、全体的に「server」や「linux」といった語がスコアに大きく寄与しており、11 位に位置したドキュメントにはこれらがほとんど含まれていないため、低いスコアを付与される結果となっていた。一方で提案手法を用いた場合、このドキュメント中の「I'm having major problems」というフレーズがスコアに寄与し、順位を上昇させていることが確認できた。

6 おわりに

この実験においては、トラブルシュート文書に出現するトラブル報告表現に頻出する構文情報を用いた同様の文書の効率的な抽出の可能性について示した。

特に今回はトピック内の先頭の発言から機械的に例文を抽出したためトラブルシュート表現の一部の特徴を利用するにとどまっているが、トピック全体から適当な形で選別されたデータを用意することにより、より精度の高い処理が可能となると考えられる。

参考文献

[1] Satoshi Sekine. On-Demand Information Extraction. the International Committee on Computational Linguistics and the Association for Computational Linguistics, 2006.

[2] Yusuke Miyao and Jun'ichi Tsujii. Probabilistic modeling of argument structures including non-local dependencies. In Proceedings of the Conference on Recent Advances in Natural Language Processing (RANLP) 2003, pp. 285-291

[3] Katharina Jahn and Stefan Kramer. Optimizing gSpan for Molecular Datasets http://wwwkramer.in.tum.de/kramer/jahn_mgts05.pdf

[1] http://www-tsujii.is.s.u-tokyo.ac.jp/enju/
[2] http://www.linuxforums.org/