1B-02

開発文書リコメンド技術に関する評価と考察

齋藤鎮成† 須永聡† 原田山人† 宮尾浩†
日本電信電話株式会社 ネットワークサービスシステム研究所†

1. はじめに

ウォータフォール型のソフトウェア開発においては、バグ対応コストは下流工程になるにつれて増大する傾向にあり、過去の調査によると納品以降のバグ修正コストは要求仕様時の修正コストの約 200 倍にもなる、との試算が出ている[1]。また、納品後にバグが発生すると、ユーザからの信用失墜にもつながる為、開発担当者は上流工程で定義する要件定義書、設計書、方式検討資料等の文書を漏れなく確認し、仕様に盛り込んでいくことが求められる。

著者らは、上記に関する対策の 1 手法として、開発文書をリコメンドする手法（開発文書リコメンド技術）[2]を研究している。同技術は、検索対象の文書群に含まれる語の特徴を予め分析することで、開発者による検索時に、その検索語に対する共起語を追加付与し、適切な関連文書を表示させるものである。

本稿では、同技術を盛り込んだ「開発文書リコメンドシステム（以下、本システム）」を実際のソフトウェア開発事例に適用し、その有効性について評価を行った。

2. 本システムの概要

本システムは、Web ベースの検索システムであり、Google のような既存の検索エンジンと同様、ユーザが検索ワードを入力した際に、関連文書をスコア順に表示するシステムである。ユーザが頻繁に投入するであろう「検索語」とその語と関連性の深い語である「関連語」を紐づけるテーブル（以下、「関連語テーブル」）を事前作成しておくことにより、ユーザが投入した「検索語」に「関連語」を自動付与し、より適切な開発文書を上位表示させるものである。Google のようなインターネットで利用される検索エンジンは、全世界からの膨大な検索ワードを分析することでサジェスト機能等の提供が可能であるが、一方、本システムは社内の各部署に閉じたものを想定しており膨大な検索数は望めない。そのため、検索対象文書群を事前に分析し静的な関連語テーブルを作成しておくことで、即座に本システムの利用が可能となる。

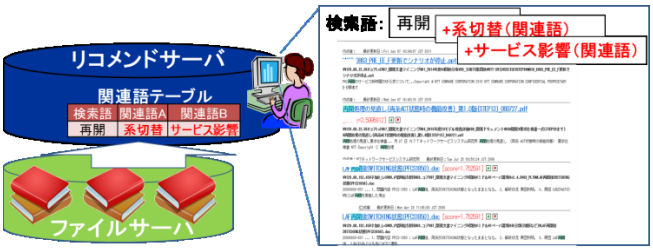


図 1 「開発文書リコメンドシステム」の概略図

3. 「関連語テーブル」の作成手法

開発文書の調査により、ユーザが投入する「検索語」と関連性の強い語は同一小節内に多く含まれることが想定される。そこで、「関連語」は「検索語」との同一小節内の共起頻度上位 20 語とした。ただし、単に共起頻度だけで集計するだけでは、「機能」や「手順」といった、どの「検索語」の周辺にも共通して共起する関連性の弱い語（「共通ノイズ語」）が上位となってしまうため、それらを除去した上で関連語テーブルを作成した。

また、「検索語」の候補は、検索対象とする文書群の出現数で決定したが、「共通ノイズ語」を除去すると「関連語」の候補である共起頻度上位 20 語の 80%以上が除去されてしまうような場合には、関連性の弱い語としか結びついていない語として「検索語」の対象から除外した。
（下記図 2 参照）

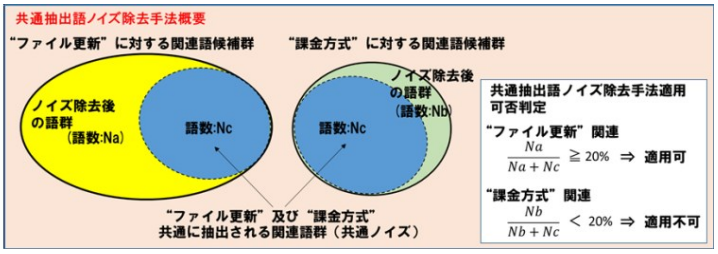


図 2 「共通ノイズ語」の割合による「検索語」への適用条件

4. 本システムの評価

(Ⅰ)本システムを利用することにより、文書確認漏れによるバグ発生の抑止効果
(Ⅱ)本システムを利用することによる、文書検索時間短縮効果
という 2 点に関して評価を行った。

An evaluation of the technology to recommend appropriate development documentations
†SAITOU Tsunenari ,SUNAGA Satoshi ,HARADA Yamato, MIYAO Hiroshi
NIPPON TELEGRAPH AND TELEPHONE CORPORATION
Network Service Systems Laboratories

Copyright ©2018 Information Processing Society of Japan.
All Rights Reserved.

4.1 評価実施条件

実際にシステム開発を行っている担当部署が閲覧している開発文書、方式検討資料等、約 3 万 4 千ファイルを検索対象文書群としてリコメンドシステムに登録した。関連語テーブルは上記検索対象文書に登録されている要件定義書、機能仕様書から抽出した約 1000 語の「検索語」とそれぞれの「検索語」に紐づく 20 語の「関連語」を前述の手法により抽出し「関連語テーブル」を作成した。

4.2 評価方法

上記（Ⅰ）バグ抑止効果に関して以下の方法で評価した。

(1) 過去のシステム開発に関するソフトウェアバグ約 270 件を調査し、文書の確認漏れにより発生したバグ、且つ、確認漏れをした該当の文書（正解文書）が明確なバグ件数を抽出

(2) （1）の中で、正解文書を本システムで検索した際に、「関連語」付与による順位上昇で 50 位以内にランクインしたバグ件数を抽出（「関連語」無しで 50 位内だったものが「関連語」付与により更に順位上昇したものも含む）

ここでは（2）で抽出されたバグを、「関連語」付与による正解文書の順位上昇により抑止可能であったバグ、として集計する。

上記（Ⅱ）検索時間短縮に関して以下の方法で評価した。

(3) 通常、所在不明のファイル検索を行う場合どの程度の時間を要するのかを集計するため、5 名に対して計 50 件のファイル検索作業を実施してもらい、平均時間を算出

(4) 本システムの利用ログから、本システム利用時の目的文書検索に要する時間を集計

5. 評価結果

（Ⅰ）バグ発生抑止効果に関して以下のような結果を得た。(1)の件数は 11 件、(2)の件数は 4 件となり、調査した全バグ件数を母数とした場合のバグ抑止率は 1.5%、文書確認漏れが原因のバグ件数を母数とした場合のバグ抑止率は 36%となった。（下記図 3 参照）

全調査バグ数(A)	文書確認漏れが原因且つ、該当の文書が見つかったバグ数(B)	本システム利用により、確認漏れが防げたバグ(C)	バグ抑止率	
			(C)/(A)	(C)/(B)
268件	11件	4件	1.5%	36%

図 3　バグ抑止効果に関する評価結果まとめ

（Ⅱ）検索時間短縮効果については、(3)の平均時間は約 18 分、(4)の平均時間は約 2 分となり、ファイル検索 1 件につき、16 分程度の短縮効果を確認した。

6. 考察

（Ⅰ）のバグ抑止に関しては、本システムを利用することにより、文書確認漏れが原因のバグに対しての抑止の可能性が示せた。11 件の内、うまくいかなかった 7 件に関しては、付与した「関連語」の一部が正解文書以外の文書のスコアを上昇させていることで相対的に正解文書の順位が下がる、という結果になっている。正解文書が適切に上位に表示されるよう、今後は各文書の特徴となる語を抽出し、「関連語」テーブルに反映するといったことを行うことで、更なるバグ抑止率の向上が図れると考えている。

（Ⅱ）の検索時間短縮効果については、想定通りとなった。

7. まとめ

過去のソフウェア開発で発生した、開発文書の確認漏れで発生したバグに関して、本システムを適用した場合にバグ抑止が可能であったか、という観点で評価した。

その結果、本システムの関連語付与効果により一部のバグ抑止が可能であったと判断でき、一定の効果を見ることができた。また検索時間短縮の効果も確認できた。

今後は「関連語」抽出方法の改良を図ることでバグ抑止率の向上を図ると共に、調査バグ数を広げていくことで本システムの信頼性の向上を図っていく。

参考文献

[1] “先進的な設計・検証技術の適用事例報告書 2015 年度版 「コストモデル」を使った 開発品質・生産性向上の取組み”. https://www.ipa.go.jp/files/000049404.pdf

[2]柴田 弘,山田佳彦”ソフトウェア開発における重要な関連情報を提示するリコメンド方法” 2016 年 信学会ソ大, B-7-10, (2016).

Copyright ©2018 Information Processing Society of Japan.
All Rights Reserved.