

# 要求獲得におけるステークホルダモデルの自動生成手法の提案

北澤 玲央<sup>†</sup> 長岡 武志<sup>‡</sup> 北川 貴之<sup>‡</sup> 位野木 万里<sup>†</sup>

工学院大学<sup>†</sup>

東芝デジタルソリューションズ株式会社<sup>†</sup>

## 1. はじめに

Digital Transformation の実現などのデジタルサービスの開発が大規模化・複雑化し、関係するステークホルダも多様化している[1]. ステークホルダの洗い出しが不十分であることに起因して、開発の途中やリリース後に要求変更や要求追加が発生するケースが報告されている[2]. 情報システムの要求定義において、ステークホルダ識別が必要である[3]. 企画や提案の初期の段階で、関係するステークホルダを漏れなく洗い出すことは困難である.

ところで、自然言語処理技術が進化し、キーワード抽出、文の分類、文書要約に加えて、文の生成に関する技術の精度も向上している。また、要求定義関連のタスクについても自動化の取り組みが行われている。例えば、*elicitation, analysis, modeling, verification and validation* などのタスクに対して自然言語処理技術の活用が取り組まれている[4]. しかし、生成技術については、テストケースや UML モデル変換などにとどまっておき、前述した関連するステークホルダをスムーズに自動生成する状況には至っていない。

著者らの研究グループでは、要求仕様書の検証や要約などの自動化の研究に取り組んできた[5]. 本研究では、これらの知見に基づき、ステークホルダの候補を自動生成する手法の考案を試みる。具体的には、自然言語処理技術の事前学習モデルと質問応答技術を活用して、対象ドメインに関連するステークホルダ候補を自動抽出し、ステークホルダモデルとして可視化する手法を提案する。

## 2. 課題と解決策へのアプローチ

要求獲得の出発点となる、提案書やシナリオの断片から、自然言語処理技術を用いて、出現するステークホルダに該当する用語を自動抽出することはステークホルダ識別の効率化や均質化に有効であると考えられる。

しかし、ステークホルダ識別では、利用者、発注者、開発者に加えて、利用者に関連する組織や環境、発注者側とやりとりのあるクライアントなど、幅広く洗い出すことが求められている[6]. 与えられた提案書や仕様の断片に出現する用語のみを抽出するだけでは不十分である。

そこで、BERT[7]の質問応答技術を用いて、関連するステークホルダを自動抽出し、ステークホルダ図を自動生成する手法を提案する。提案手法では、提案書や仕様の断片などの対象ドメインの文脈がわかるコンテキスト情報を入力として、質問応答によって自動抽出するための質問文を自動生成し、段階的に関連するステークホルダを自動抽出することを試みる。

## 3. 提案手法

本研究で提案する手法のプロセスと支援ツールを図 1 に示す。提案手法は、対象ビジネスのコンテキスト情報を入力として、関連するステークホルダリストを自動生成する。図 1 では、本手法の出力はステークホルダ図としている。本稿では、ステークホルダリストを自動生成し、生成されたステークホルダリストを用いて、手動にてステークホルダ図を作成することになる。

提案手法は、入力されたコンテキスト情報とリソースモデルを用いて、S01 で関連するステークホルダを問う質問文を自動生成する。その自動生成された質問文を用いて、S02 では BERT の質問応答技術により、関連するステークホルダの候補を導出する。S03 では導出されたステークホルダ候補を用いてさらに関連するステークホルダを問う質問を繰り返し適用し、関係するステークホルダ候補を段階的に導出する。なお、導出されたステークホルダ名には、異音同義語が含まれる可能性があるため、必要に応じて、類似度を用いて、ステークホルダ名の整理統合を行う。

本研究では、形態素解析器として Juman++[8]、自然言語処理モデルとしては BERT、日本語 Wikipedia をソースとした事前学習モデル (<https://huggingface.co/cl-tohoku>) を用いる。

Proposal of method for automatic generation of stakeholder model in requirements elicitation -

<sup>†</sup>Reo Kitazawa, Mari Inoki, Kogakuin University

<sup>‡</sup>Takeshi Nagaoka, Takayuki Kitagawa, Toshiba Digital

Solutions Corporation



図1 ステークホルダモデルの自動生成ツール

4. 適用評価

4.1 方法

「駅周辺経路誘導」をとりあげて、本手法の適用評価を行った。渋谷駅を対象として、コンテキスト情報（329文字）を入力情報とした。本研究では、あらかじめ、有識者によりステークホルダ図を手動で作成し、本手法によって生成されたステークホルダ一覧を比較し、再現率、適合率、F値の観点で分析した。本研究では、提案手法であるステークホルダ図の自動生成ツールを用いてステークホルダ図を作成した。

4.2 結果

本ツールによって、生成した質問文として、例えば「[MASK]が渋谷駅周辺経路誘導システムをオリンピック開催時に渋谷駅で利用する」を生成した。また、本ツールにより自動生成したステークホルダ一覧では、20件のステークホルダを生成した。実際に自動生成ツールを用いて、手動でステークホルダ図にマッピングした例を図2に示す。生成されたステークホルダの中には、「Google」や「ホテル」などの手動では定義されなかったステークホルダが生成された。また、その他の「鉄道」や「通信会社」などの、本コンテキストには明確には関連しないステークホルダが生成された。

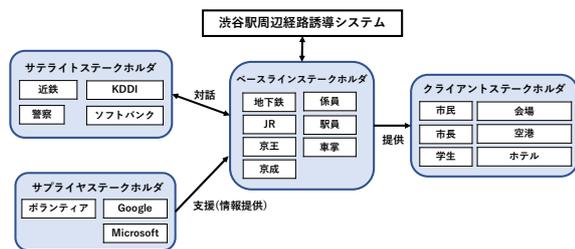


図2 渋谷駅周辺経路誘導システムのステークホルダ図

4.3 評価分析

作成したステークホルダ図に対して正確性の評価を行った。正確性の評価では、Precision, Recall, F1の3項目に対して評価を行った。結果は、Precisionが0.66, Recallが0.68, F1が0.67となった。また、評価において、異音同義語を正解とみなした場合、Precisionが0.72, Recallが0.73, F1が0.73となった。

5. 考察

適用評価結果に基づけば、本ツールで自動抽出したステークホルダ一覧は、手動の定義結果と比較し、一定程度の一致度を備えていること、また手動では抽出できていなかった新たなステークホルダの特定が可能であることから、初級の技術者のステークホルダ識別の支援に有効であると考えられる。

しかし、異音同義語をどう扱うか、4つのステークホルダのどこにも分類されないステークホルダをどう扱うかなどの改善点がある。また、今回手動で行った部分を自動化することで、さらに効率化できることが考えられる。

6. まとめ

本稿では、初級技術者が簡単にステークホルダを識別できる状態を目指し、自然言語処理技術を用いたステークホルダモデルの自動生成手法を提案した。今後、更にステークホルダの洗い出しによる分析を継続し、より良い自動生成と可視化の統合手法の提案に努めていく。

謝辞

本研究はJSPS 科研費 JP19K11907の助成を受けた。

参考文献

- [1] 位野木万里, et al., エンジニアリング部会 要求工学グループ 活動報告, 情報サービス産業協会, Quarterly, No.141, Spring, 2021.
- [2] REBOK 企画 WG, 要求工学実践ガイド, 情報サービス産業協会, 2014.
- [3] REBOK 企画 WG, 要求工学知識体系, 情報サービス産業協会, 2011.
- [4] Liping Zhao, et al., Natural Language Processing for Requirements Engineering: A Systematic Mapping Study. ACM Comput. Surv. 54, 3, Article 55, 2022.
- [5] 高橋宏季, et al., 要求仕様書中のアクター名の定義漏れパターンと組織変更がもたらす影響—実案件分析と得られた教訓—, 情報処理学会 デジタルプラクティス, 11巻/2号, pp.367-388, 2020.
- [6] Sharp, H., Finkelstein, A. and Galal, G. : Stakeholder identification in the requirements engineering process, Proceedings of 10th International Workshop on Database & Expert Systems Applications (DEXA), pp. 387-391, 1999.
- [7] BERT 日本語 Pretrained モデル, [https://nlp.ist.i.kyoto-u.ac.jp/index.php?ku\\_bert\\_japanese](https://nlp.ist.i.kyoto-u.ac.jp/index.php?ku_bert_japanese) (参照 2022-08-02)
- [8] Juman++, <https://nlp.ist.i.kyoto-u.ac.jp/?JUMAN%2B%2B>, (参照 2022-08-03)