E-040

医療事故報告書に対するテキストマイニングシステム
Construction of a Text Mining System for Knowledge Discovery from Medical Documents

松村 崇光　　竹内 孔一
Takamitsu Matsumura　　Koichi Takeuchi

1. はじめに

医療機関では，事故だけでなく事故に至りそうになった状況など含めた報告を記述する事例を収集し分析を実施している．こうしたインシデントレポートには医療現場において，医療器具や薬品を利用した場合の事件や事故などが書かれており，こうした情報を有効利用することで危機の改善や，医療行為における改善につなげられると期待されている．近年このような病院内のデータを有効活用しようという動きが高まってきており，特定の化学物質の発がんリスク評価における文献調査のためのテキストマイニングシステムの研究[1]や投薬のレポートを対象に臨床医の意図と投薬の事実性のアノテーションを付与した MIR コーパスの開発[2]など様々な研究が行われている。

インシデントレポートを分析するには，人が 1 件ずつ読むか，キーワード検索などでテキストを取り出す形で利用することが考えられる．大量に存在しているインシデントレポートに対し，人が 1 件ずつ目を通して確認することは大変コストがかかる上，キーワード検索で探すとしても目的のデータを見つけ，インシデントレポートから有益な情報に気づくのは困難である．これらを改善するためにインシデントレポートに対するテキストマイニングシステムを構築する．テキストマイニングシステムとは単なる検索や分類整理とは異なり，複数の文書データの内容を総合的にとらえることで初めて得られる知見を抽出するための内容分析の技術である．ある単語に対する相関関係や出現頻度，インシデント発生時間などを分析し，医療方法の改善策や注意点などを発見するためのツールとなることが期待される．

2. 方法

実務で利用することを想定し，必要な機能を実装していく．開発対象のデータとして日本医療機能評価機構が一般に公開している医療事故情報収集等事業のインシデントレポート[1]を用いる．理由としては一般に公開されているデータであり，病院などで扱われているデータと似ているので実用する環境と同じような条件で開発できるからである．しかしながら，本研究で作成するシステムは医療事故情報テキストを対象に構築するためシステムの機能を工学的な観点から改善することが可能である．

2.1 テキストマイニングシステムで必要となる機能

使用者がインシデントレポートから重要な情報や関連性を発見するための手助けとなるような機能が必要である．よって必要となる機能としてファイルのアップロードや削除などの機能，文字列で高速に検索できること，時系列で検索した言葉の出現頻度が表示できること，あるキーワードの係り先や係り元が表示できることなどが挙げられる．

2.2 テキストマイニングシステムの構成図

図 1 に本研究で作成するテキストマイニングシステムの概要を示す．

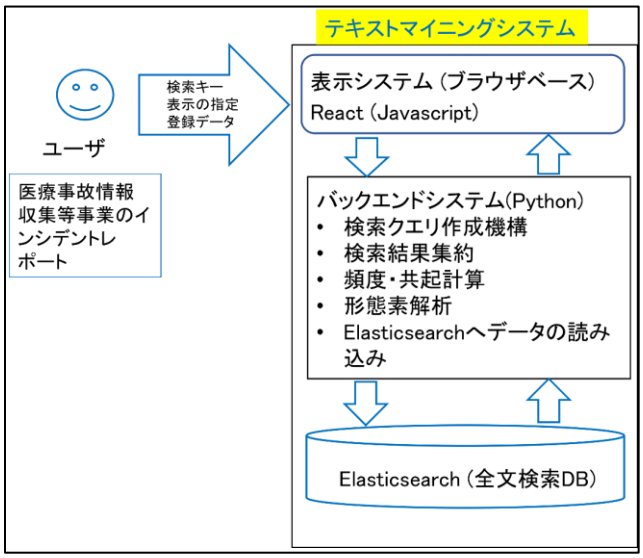

図 1:テキストマイニングシステムの全体像

まず，処理の大まかな流れを説明する．検索ボックスを設けて検索ワードを入力し，入力された検索ワードを用いてバックエンドで処理を行う．その処理結果をフロントエンドに返して，表示する．フロントエンドは React で作成している．フロントエンドとバックエンド間の通信には Axios を用いている．バックエンドシステムは言語処理と Elasticsearch から構成されている．まず，言語処理システムから説明する．言語処理は Python で行っており，単語の頻度計算やグラフ表示のためのデータの整形，係り受け解析の実行，Elasticsearch へのデータの登録を主に行う．Elasticsearch との通信には Python 内の Elasticsearch ライブラリを使用する．Elasticsearch では検索ワードを含むインシデントレポートを検索したり数を数えたりする機能を有する．詳しい説明は次節以降で行う．

2.3 Docker を用いたシステムの構築

Docker とは Linux に実装されているコンテナ技術を使ったもので，隔離されたアプリケーションの実行環境を実装することができる．まず一つ目の特徴として，軽量であることがあげられる．コンテナという技術により名前空間やリソースが隔離されているのでコンテナを管理するコストとプロセスを管理するコストがほぼ同じになり，低コストで管理できる．二つ目の特徴は開発環境を別のマシンでも簡単に実装できることである．Docker イメージという機能

[1] https://www.med-safe.jp/mpsearch./SearchReporResult.action

を使うことでこれを実現している．Docker イメージにはコンテナの実行に必要な情報が書かれている．この Docker イメージを共有することによって別のマシンでも容易に環境構築が可能となる．三つ目の特徴として高速ということである．Docker でプロセスを実行するとき， OS 上の一つのプロセスとして実行されるので余計な工程がなく高速で実行できる．このソフトウェアを用いることによって，Docker イメージを共有するだけで作成したシステムを容易に病院のマシンで構築することができ，病院側ではインシデントレポートを登録するだけでテキストマイニングシステムを使用できる．

2.4 Elasticsearch による高速なデータベースの実現

Elasticsearch とは，大量の文書データ向けの高速な全文検索エンジンである．大量の文書データはインデックスという形で登録し，登録される文書は analyzer により解析される．これにより，Elasticsearch 独自の検索が可能となる．またインデックスを分割することで不要なインデックスを削除したり，新しいデータを新しいインデックスとして登録したりすることが容易となり，データ整理が簡単である．

Elasticsearch の検索について説明する．実際に検索したクエリと結果を図 2 に示す．


```
{'query': {'bool': {'must': [{'bool': {'must': [{'multi_match': {'fields':
 ['事故の内容', '事故の背景要因の概要'], 'query': '"プレドニン"', 'type':
'phrase'}}, {'multi_match': {'fields': ['事故の内容', '事故の背景要因の概
要'], 'query': '"配薬カート"', 'type': 'phrase'}}]}}, {'bool': {'must_not'
: {'multi_match': {'fields': ['事故の内容', '事故の背景要因の概要'], 'quer
y': '"朝夕"', 'type': 'phrase'}}}}]}}, '_source': ['事故の内容', '事故の背
景要因の概要', '報告年', '事例ID'], 'highlight': {'number_of_fragments': 0
, 'pre_tags': ['<mark>'], 'post_tags': ['</mark>'], 'fields': {'事故の内容
': {}, '事故の背景要因の概要': {}}}, 'size': 10000}
{'_shards': {'failed': 0, 'skipped': 0, 'successful': 10, 'total': 10},
 'hits': {'hits': [{'_id': 'VrT4aogBqPbPFLknpV4R',
                    '_index': 'medicalreportpubh_2014',
                    '_score': 16.291363,
                    '_source': {'事例ID': 'H365FE9D689CC52C3',
                                '事故の内容': '夕方、配薬カートに定時内服
薬をセットする際にプレドニン錠が残数が４錠であることに気づき、 朝食後のプ
レドニン錠を５錠内服するところを１錠しか内服出来ていなかったことが発覚した
。配薬カートにセットする時点でプレドニン１錠セットされており当事者本人も内
服ダブルチェック時、処方箋にはｍｇの単位で記載があり過少与薬であることに気
づかなかった。',
                                '事故の背景要因の概要': '確認を怠った。',
                                '報告年': '2014年'},
                    '_type': '_doc',
                    'highlight': {'事故の内容': ['夕方、<mark>配</mark><ma
rk>薬</mark><mark>カート</mark>に定時内服薬をセットする際に<mark>プレドニ
ン</mark>錠が残数が４錠であることに気づき、朝食後の<mark>プレドニン</mark>
錠を５錠内服するところを１錠しか内服出来ていなかっ たことが発覚した。<mark
>配</mark><mark>薬</mark><mark>カート</mark>にセットする時点で<mark>プレド
ニン</mark>１錠セットされており当事者本人も内服ダブルチェック時、処方箋に
はｍｇの単位で記載があり過少与薬であることに気づかなかった。']}},
```


図 2:Elasticsearch における検索クエリと結果

図 2 では，検索ワードを“プレドニン”と“配薬カート”の OR 検索，除外キーワードを“朝夕”としてクエリを生成している．Elasticsearch の bool query を使用することで AND 検索，OR 検索を自由に組み合わせ，自由度の高いクエリを作成できるようにした．検索する対象を“事故の内容”と“事故の背景要因の概要”の 2 つとしているので multi-match query を使用し，複数のフィールドに対し同時に検索することができる．また，“type”に“phrase”を指定することにより検索方法を Match-phrase 検索とすることができる．Match-phrase 検索とは分割されたキーワードがクエリと同じ順序で含まれていなければならず，キーワード間に余計な文字が入ると検索できないというものである．また記号（・や！）を除いてマッチしたり，大文字小文字を正規化してからマッチしたりするので，データの取りこぼしなく検索できる．その他には highlight 機能を用いて，検索ワードにハイライトタグを入れている．クエリの下側にそのクエリで検索した結果を表示している．図 2 では“hits.hits”の中に今回検索した結果の配列が格納されている．“source”はヒットした文書の情報が格納されており，“highlight”内にはヒットした文書に指定したハイライトタグを追加した文書が格納されている．検索ワードである“配薬カート”が Elasticsearch により“配”，“薬”，“カート”の 3 つに分割されており，この 3 つを match-phrase で検索している．このような検索方法のため分割されたキーワードそれぞれにハイライトタグが適応されている．

2.5 React を利用したブラウザベース

本研究ではブラウザベースでシステムを作成するため React を使用している．React とは JavaScript ライブラリである．UI を再利用可能なコンポーネントに分割し，それらを組み合わせることで複雑な UI を構築することができる．この React をベースに Material-UI や React-chartjs-2 などを使用し，意図した機能を実装することができる．

Material-UI は React ベースの UI コンポーネントライブラリであり，ボタンやテキストフィールド，ダイアログなどをカスタマイズ可能なコンポーネントとして提供している．これらのコンポーネントを用いることで意図したとおりの機能を実装できる．

React-chartjs-2 は折れ線グラフや棒グラフ，ドーナツチャートなど様々なグラフを描画できるライブラリである．本研究は使用者が有益な情報を発見するためのアプリの開発なので，検索方法にあった結果を表示しなければならない．また，クリックイベントでのデータの詳細な情報の表示が実装しやすいので今回 React-chartjs-2 を使用する．

2.6 Python ベースの言語処理システム

近年，係り受け解析は社会問題[3]や教育[4]など様々な分野で活用されている．そして，一般的に CaboCha[2]や MeCab[3]を用いて文書の解析[5]やテキストマイニングシステムの構築[6]を行っている．本研究では商標利用も考慮して Ginza[4]を用いて係り受け解析を行う．対象となる医療事故情報データに対し，形態素解析を実行し，文節ごとの係り関係を抽出する．係り元，係り先，事例 ID の 3 つをセットとし，インデックスとして登録しておく．そして，係り元を対象として検索し，ヒットする係り先を抽出，カウントする．

係り受けの頻度表示により，検索ワードに対しての述語を取ることができる．よって，検索ワードがどのようにインシデントレポートと関連しているかが分かる．例えば，「三方活栓」というキーワードを検索した結果として「忘れる」12 件，「外した」8 件，「変更し」3 件という情報を得られたとする．この結果を見ると「三方活栓」に対して「忘れる」や「外した」などの述語が係り先となっている場合が多いことが分かる．よって，「三方活栓」とは忘れる時や外す時に医療事故につながる出来事が起こってい

[2] https://taku910.github.io/cabocha/

[3] http://taku910.github.io/mecab/

[4] https://megagonlabs.github.io/ginza/

ると分かる．このように，検索ワードとインシデントレポートの関連性を表示し，検索ワードに対する対策や改善策などを気づくための手助けとなる．

3. 評価実験

本章では 2 章で構築したシステムの評価実験を行う．3.1 節では実験の設定について，3.2 節では実験の評価方法について，3.3 節では実験結果について説明する．

3.1 実験設定

本実験はテキストマイニングシステムに必要な機能が実装できているかを確認する．今回確認する機能は高速に検索できること，全文検索機能，グラフ表示機能，係り受け解析による頻度表示機能である．

次に用いているデータについて説明する．使用したデータは一般に公開されている日本医療機能評価機構の医療事故情報データを 2010 年から 2019 年分である．合計約 7 万事例である．図 3 にデータの例を示す．

```
"事業区分","事例ID","報告年","発生曜日","曜日区分","発生時間帯","発生時間帯_不明（記述）","医
"ヒヤリ","H000D1096BB837FDA","2010年","土曜日","","12:00～13:59","","実施あり","治療なし"
"ヒヤリ","H00145579E55593F5","2010年","金曜日","","22:00～23:59","","実施なし","","軽微な
"ヒヤリ","H001E3617C393F32F","2010年","金曜日","","18:00～19:59","","実施なし","","軽微な
"ヒヤリ","H00231843C7F1E844","2010年","水曜日","","22:00～23:59","","実施あり","治療なし"
"ヒヤリ","H003060811B6E7AF0","2010年","土曜日","","8:00～9:59","","実施あり","治療なし","
```

図 3:医療事故情報データのインシデントレポートの例

図 3 のデータではインシデント例が 1 行ごとに記載されており，インシデントの年日時や曜日，時間などが記録されていることが分かる．図 3 には表示されていないがインシデントの内容はテキストで詳細に報告されている．本システムでは，報告年，事故の内容，事故の背景要因の概要事例 ID の 4 つの項目を利用し，検索対象とするのは事故の内容，事故の背景要因の概要の 2 つとする．全文検索機能では検索対象に文字列検索を実装する．グラフ表示機能では指定した文字列が検索対象の中に含まれているデータの数を報告年ごとに表示する．係り受け解析による頻度表示では検索対象に形態素解析を行い，指定した文字列の係り先の頻度を表示する．

3.2 評価項目

本稿で評価する項目はデータ操作機能，キーワードによるテキスト検索機能，キーワードのグラフ表示機能，係り受け解析による頻度表示機能，指定したデータの詳細を表示する機能である．すべての機能に対して意図した動作が行えているかと結果を表示するまでの処理時間を評価する．テキストマイニングシステムでは使用者の意図に応じて様々な検索要求に対してテキストデータを提示する必要がある．使用者の発想を止めないように処理速度は短いほうが優れている．データ操作機能ではデータの登録を 1 つも取りこぼすことなく登録できているか，必要となるデータの操作が可能であるかなどを評価する． そのほかの機能ではデータの表示やグラフの表示が正確に行えているか，グラフのクリックイベントが正確な順序で処理を行えているかなどを評価する．

3.3 実験結果

本節では主要となる機能の説明と実験結果について説明する．本実験で評価する機能はデータの操作機能，全文検索機能，頻度表示機能，指定したデータの詳細な情報を表示する機能，係り受け解析による頻度表示機能の 5 つの機能を評価する．

3.3.1 データの操作機能

図 4 にデータ操作機能のページを示す．

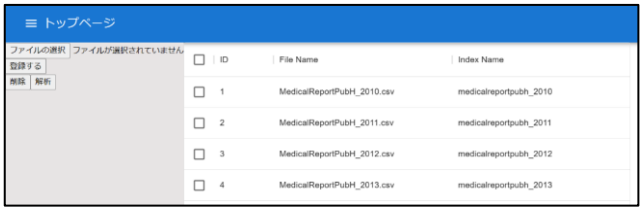

図 4:データ操作のページ

このページはシステムを立ち上げた際のトップページとなっており，右側の表で Elasticsearch 内に登録されているファイル名とインデックス名を表示している．左上のハンバーガーボタンで全文検索のページや頻度表示のページへと遷移することができる．左のサイドメニューではデータの登録，データの削除，データの解析ができる．

データの登録では“ファイルの選択”ボタンからファイルを指定することで Elasticsearch に指定されたファイルをインデックスとして登録することができる．ファイルは一度に複数指定することができる．登録にはバルクインサートを使用しており，5000 件ずつ一括で登録している．今回使用している 2010 年から 2019 年分の医療機能評価機構の医療事故情報データ，合計約 7 万事例分を一括で登録にかかった時間は約 51.585 秒である．また，データの更新にかかった時間は約 0.034 秒だった．一度に登録するデータが多くなると処理時間が長くなるが一度登録をすると Docker コンテナを削除しない限り登録したデータは消えないのでそこまで重要な問題ではないと考える．

データの削除では指定したデータを削除することができる．現在の状態を表す表の中にチェックボックスがあり，削除したいインデックスにチェックを入れ削除ボタンを押すと削除される．今回登録していた 10 個のファイルの削除にかかった時間は約 0.871 秒だった．

データの解析では指定したインデックスに対し，係り受け解析を行い，行った結果を新たなインデックスとして登録している．本稿では 2010 年のファイルを解析した．処理にかかった時間は約 3 時間だった． 1 万件弱のデータ内の 2 つの項目に対し解析を行っているので約 2 万件弱のデータを解析していることになる．解析する項目が増えればより多くの時間を必要とする．一度解析すれば半永久的に使用できるが今後の為にも改善が必要な部分であると考える．

3.3.2 全文検索機能

図 5 に全文検索のページを示す．

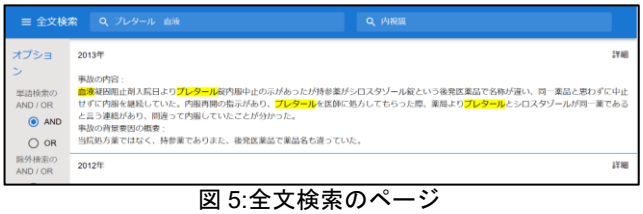

図 5:全文検索のページ

今回単語検索に“プレタール”と“血液”を AND 検索，除外検索に“内視鏡”とし検索を行った．結果は 2 件ヒットし，条件通りの結果となった．検索単語である“プレタ

ール”と“血液”のどちらにもハイライトが入っていた．検索時間は約 0.024 秒だった．

図 6 に単語検索を“プレタール”と“血液”の OR 検索，除外検索を“内視鏡”と“医師”の AND 検索としたものの結果の一部を示す．

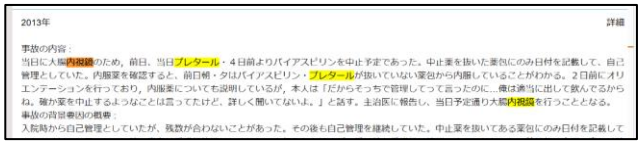

図 6:単語検索を“プレタール”と“血液”の OR 検索，除外検索を“内視鏡”と“医師”の AND 検索とした結果の一部

ヒットした件数は 1254 件で検索時間は約 0.212 秒だった．図 5 と図 6 を比較しても分かる通り図 6 では“プレタール”のみの場合でもヒットしており，“内視鏡”と“医師”の両方が入っていないので検索結果としてヒットしている．ヒットした件数が多くなると検索時間も長くなっているが問題ない長さであると考える．

3.3.3 頻度表示機能

図 7 に頻度表示のページを示す．

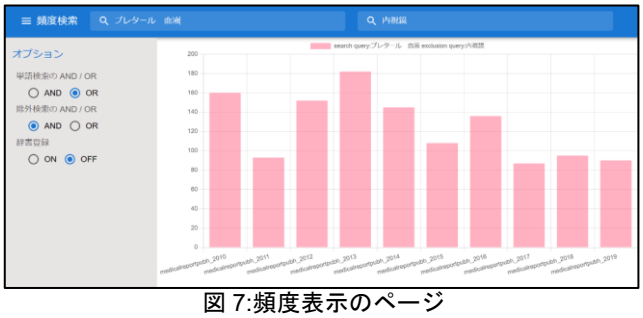

図 7:頻度表示のページ

今回単語検索に“プレタール”と“血液”を OR 検索，除外検索に“内視鏡”とし検索を行った．それぞれ 2010 年:160 件，2011 年:93 件，2012 年:152 件，2013 年:182 件，2014 年:145 件，2015 年:108 件，2016 年:136 件，2017 年:87 件，2018 年:95 件，2019 年:90 件の合計 1248 件となった．検索時間は約 0.209 秒だった．クリックイベントも実装しており，その機能は次節で説明，評価する．この機能の検索時間も問題ない長さであると考える．

3.3.4 指定したデータの詳細な情報を表示する機能

図 8 に図 7 の 2011 年に該当するバーをクリックした時に移動するページを示す．

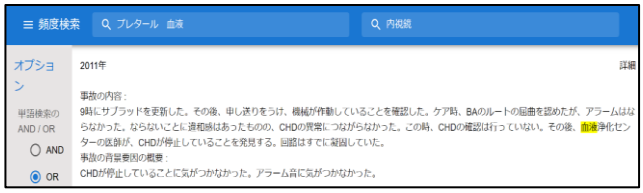

図 8:グラフのクリックした際にグラフに対応したテキストデータを表示

このページの最初の検索は“事例 ID”を用いて検索し，ハイライトは図 7 と同じクエリを生成し，そのクエリを highlight query とすることでハイライトを入れている．またこのページでは 2011 年の 93 件のみのインデックスを作成し，その 93 件のみを対象とした検索ができる．このページに遷移する処理時間は約 0.565 秒だった．普段のページ遷移と比べると検索やインデックス作成などの処理があるので長くなっているが問題ない長さであると考える．

3.3.5 係り受け解析による頻度表示機能

図 9 に係り受け解析による頻度表示のページを示す．

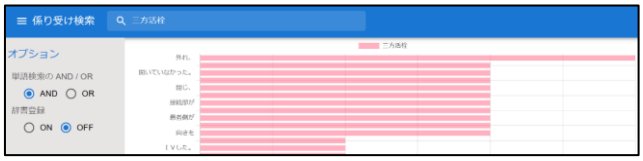

図 9 : 係り受け解析による頻度表示のページ

今回単語検索に“三方活栓”とし 2010 年のデータを解析し，検索を行った．結果から“外れる”や“開閉”のような単語が多く存在し，このような動作によってインシデントが発生することが分かる．検索時間は約 0.016 秒だった．前処理として解析を行っていたので検索自体は高速で行えることができた．

4. まとめ

本研究ではテキストマイニングシステムの構築とそれに対する評価を行った．本研究における実験では実際のインシデントレポートとして日本医療機能評価機構が作成している医療事故情報データを用いてシステムの実装をした．その結果，データの操作機能，全文検索機能，頻度表示機能，指定したデータの詳細を表示する機能の実装に成功した．いずれの機能も速度面では十分の結果が出ていた．また，AND 検索，OR 検索，除外検索を組み合わせることにより，ユーザの望むデータを見つけることを可能とする柔軟な検索を実装した．今後の課題として，今回期待する結果を得られなかったデータの解析時間の短縮や共起検索の実装，類義語辞書の登録などの新しい機能の拡張が必要であると考える．

参考文献

[1] Hongkuan Zhang, Ryohei Sasano, Koichi Takeda, Zoie Shui-Yee Wong,Development of a Medical Incident Report Corpus with Intention and Factuality Annotation, Proceedings of the Twelfth Language Resources and Evaluation Conference, 4578–4584, (2020)

[2] Yufan Guo, Diarmuid O Seaghdha, IlonaSilins, Lin Sun, Johan Hogberg, Ulla Stenius, Anna Korhonen, CRAB 2.0: A text mining tool for supporting literature review in chemical cancer risk assessment, Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations, 76–80, (2014)

[3] 佐藤岳文, 堀田昌英, Web マイニングを用いた因果ネットワークの自動構築手法の開発, 社会技術研究論文集, Vol.4, 66-74,(2006)

[4] 松河秀哉, 大山牧子, 根岸千悠, 新居佳子, 岩崎千晶, 堀田博史, 串本剛, 川面きよ, 杉本和弘, トピックモデルによるテキスト分析を支援するソフトウェアの開発, 日本教育工学会論文誌, Vol.42, 37-40, (2018)

[5] Hiroshi Masuichi, Tomoko Ohkuma, Construction of Practical Japanese Parsing System Based on Lexical Functional Grammar, Journal of Natural Language Processing, Vol.21, No.2, 659-677, (2014)

[6] Siti Norbaya YAHAYA, Nusaibah MANNSOR, Kazuhiro OKAZAKI, Analyzing Trend of Capital Adequacy and Basel Standard using Text Mining Technique, International Journal of Japan Association for Management Systems,Voi.8,No.1,9-16, (2016)