# LE-7　柔軟なテキストマイニングミドルウエアの構築

## Developing a Flexible Text Mining Middleware

浦本 直彦†
Naohiko Uramoto

## 1．はじめに

従来から開発されてきたテキストマイニングツールなどの文書処理システムは、単体のアプリケーションとして開発され、巨大でかつ複雑なものが多い。このようなアプリケーションを異なる分野、対象文書に適用しようとする場合、いくつかの問題が生じる。たとえば、アプリケーションを構成するコンポーネント群が密に結合しているため、一部を変更したり、一部だけを使用したりすることが困難である。また、アプリ独自のデータデータフォーマット、アクセスメソッドが使用されており、他のシステムと統合することが困難である。

コンポーネントの再利用やアプリケーション統合は、システムの再利用性や拡張性を高めるために必要不可欠であり、Web サービス[2]のようなさ標準技術が提案されている。本論文では、テキスト処理を行うツールの再利用性を高め、他のシステムとの統合を促進するために開発されているテキストマイニングミドルウエアの構築手法について述べる。

## 2. ミドルウエアアーキテクチャ

図 1 にミドルウエアのアーキテクチャを示す。本論文で提案するミドルウエアは Java で記述されており、以下に示す抽象クラスあるいはインターフェイスからなる。

- Database: 特定のデータベース実装を抽象化。
- Dictionary: 単語やイディオムの情報を格納。
- Taxonomy:上位下位関係などの単語間関係を格納。
- Annotator: XML 文書を処理する。
- Analyzer: Annotator の解析結果を元に、テキストマイニングなどのサービスを行う
- Transport: ミドルウエアが提供するクラスをローカルにあるいは分散環境で呼び出すためのインターフェイス

文書に対する様々なレベルの注釈付け(例. 構文解析)を行うコンポーネントを、Annotator(アノテータ)と呼ぶ。アノテータは、XML 文書を受け取り、何らかの処理を行い、XML 文書を返すサービスである。各サービスが入力として受け取る XML 文書および出力する XML 文書の型(スキーマ)はあらかじめ規定されている。

本論文で提案するアーキテクチャでは、単機能のサービスを提供するアノテータを組み合わせることで全体のサービスを構成する。もっとも単純な場合は、複数のアノテータを直列に配置する場合である。各アノテータは、XML 文書を受け取り、処理を行い、XML 文書を返す。

このような考え方は、現在、ビジネスアプリケーションの構築のためのモデルとして提案されている Web サービス[2]に基づいている(後述するように、必ずしも Web サービスアーキテクチャを用いる必要はない)。Web サービスでは、分散環境にある Web アプリケーション間で XML ベースのメッセージング仕様である SOAP[1]を用いる。

そもそも、この考え方はそれほど新しいものではない。従来の自然言語処理システムにおいても、perl などで書かれたツールを UNIX のパイプを使って組み合わせることでアノテータを結合する手法が一般的に行われてきた。

主に比較的小さな XML 文書がやりとりされるビジネスアプリケーションの分野に最適化されている Web サービスを、テキストマイニングのようなテキスト処理におけるコンポーネント技術にそのまま適用するのは以下に示す理由から、危険である。

(1)テキスト処理においては、扱う文書サイズが非常に大きい場合がある。そのため、結合されるサービスは、できるかぎり同じプロセス空間で動作することが望ましい。しかし、現在の Web サービスにおけるサービス結合においては、サービス同士を直接メソッド呼び出しでコールする統一的な仕組みがない。そのため、http を使った分散環境におけるサービス結合と直接メソッドコールでの呼び出しでは異なる実装をする必要がある。

(2)データを XML 文書で持つことは、データサイズの観点からは効率的ではない。Apache SOAP に代表される SOAP エンジンでは、XML メッセージを、ユーザが定義した XML 文書として送る方法と、RMI(遠隔メソッド呼び出し)のように、メソッドの型と値をシリアライズして送る手法の２種類があるが、どちらも、ネットワーク上を XML 文書として流れることには違いがない。1K バイトの XML 文書を送る場合には、SOAP を使っても問題ないが、100M バイトの文書を送りたい場合にはどうすればいいだろうか？SOAP 以外のトランスポート(圧縮したり、バイナリで送る)が必要な場合がありえる。

(3)現在、様々なテキスト処理サービスが Web 上で公開されている。類似するサービスを複数の機関が提供していることも多く、いかに最も適したサービスを検出し組み合わせるかが重要である 。しかし、現状では、どのようなサービスを組み合わせれば利用者にとって最適な処理が行われるかを自動的に判別することができない。結合可能なサービスの集合が与えられた時に、同じプロセス空間で動くサービスの組み合わせを優先する、あるいはネットワーク的に近いサービス群を優先するという規則を与えることで、(半)自動的にサービスを構成する仕組みが現時点では存在しないためである。本論文では、Web サービスの考え方を取り入れながら、ここであげた問題点を解決するための手法について述べる。

## 3. トランスポート層の抽象化

前章で述べた問題点の 1 と 2 を解決するために、直接 Web サービス(SOAP)をトランスポートとして用いるので

† 日本 IBM (株)東京基礎研究所

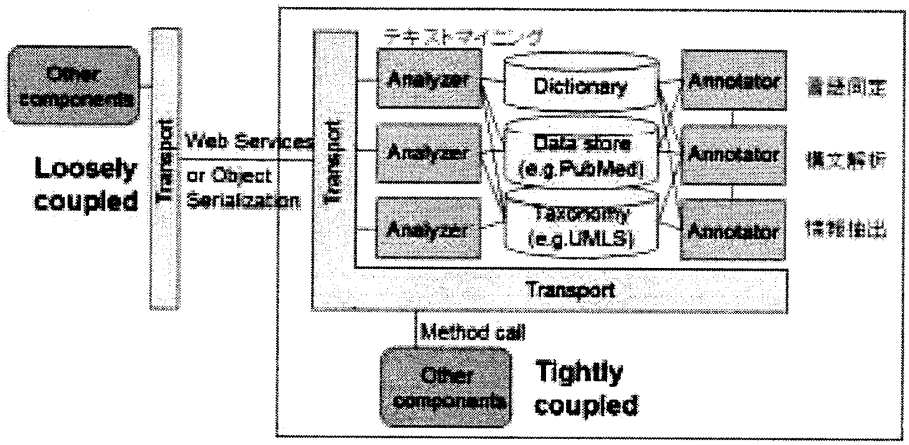


図1 ミドルウエアアーキテクチャ

はなく、トランスポート層を抽象化するインターフェイスを定義し、その実装クラスを複数用意することでアノテータ、アナライザ、コンポーネント間の通信を行う。現在、以下の実装クラスを開発中である。

- Web サービス/SOAP
- Java オブジェクトシリアライゼーション
- 直接メソッド呼び出し

これらの実装は、同一のインターフェイスを提供している。つまり、分散環境においても、同じプロセス空間においても、サービスを組み合わせるためのプログラムは同一である。トランスポートを変更することで簡単に結合の方法を変更することができる。どのトランスポートを選択すべきかは利用環境に依存するが後述する自動化されたサービス結合によって決定することもできる。

## 5. サービスグラフによるサービスの半自動結合

様々な条件に基づいて柔軟に最適なサービスの組み合わせを行うために、本論文では、与えられたサービス(アノテータ)集合を有効グラフとして表現する。複数のサービスの組み合わせはグラフ上の２点間の経路として表現されるので、最適なサービス結合を最適経路探索問題として解くことができる。

各サービスは、入力と出力を持っており、その型(スキーマ)は XML スキーマ言語によって規定されている。たとえば、サービス a の出力スキーマと、別のサービス b の入力スキーマが同一である場合、a と b をノードとし、a から b へのエッジによって２つのノードが連結されていると考えることができる。与えられたサービス集合をこのように組み合わせると、一つないし複数のグラフが構成できる。これをサービスグラフと呼ぶ(図 2 参照)。図 2 では、構文解析を行うサービスの後に、要約を行うサービスが結合している。エッジ部分には、サービス a とサービス b の結合に関する重みが付与されている。たとえば、a と b が同一のプロセス空間で動作する場合には小さな値を、分散環境で動作する場合は大きな値を付与する。

サービスを結合する場合には、想定される処理の入力と出力と一致するスキーマを持つ２つノードを選択し、入力ノードから出力ノードへの最短経路を計算する。最短経路の計算については、効率の良いアルゴリズムが知られており、本論文では詳細にふれない。エッジに付与した重みを考慮することで、最適なサービスの組み合わせを得ることができる。

もちろん、入力スキーマと出力スキーマが一致するからといって、２つのサービスを結合できるとは限らない。結合の結果は人間がチェックする必要がある。

## 6. 終わりに

本論文では、柔軟で再利用可能なテキストマイニングミドルウエアについて述べた。現在、ライフサイエンス分野の文書 (Medline[3]アブストラクト 1100 万件)を対象に、ミドルウエアの実装を行っている。アノテータとして、構文解析器や情報抽出器、アナライザとして、筆者らが開発しているテキストマイニングツール MedTAKMI を用いており、現在、実装とその評価を行っている。

## 参考文献

[1] SOAP, http://www.w3.org/TR/SOAP/
[2] Web Services, http://www.webservices.org/
[3] PubMed, http://www.ncbi.nlm.nih.gov/entrez/
[4] Knowledge Management - Text analysis and knowledge mining system -, IBM System Journal, Vol. 40, No.4, 2001.

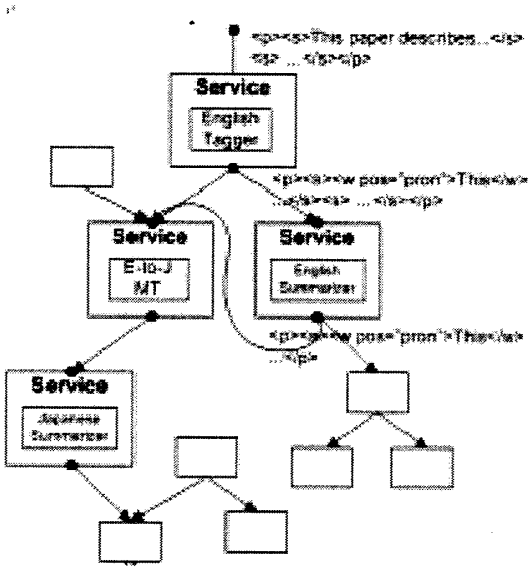


図 2 サービスグラフ