2A4-7

質問応答システムとセンター試験解答フロー: Kachako 対応による標準化・互換化

Question Answering System and Workflows for Solving the National Center Test for University Admissions: Standardization and Compatibility based on the Kachako Platform

狩野 芳伸[*1] Yoshinobu Kano
神門 典子[*2] Noriko Kando

[*1] 科学技術振興機構 さきがけ JST PRESTO
[*2] 国立情報学研究所 National Institute of Informatics

We describe our system that aims to help users developing their solver applications for the National Center Test for University Admissions (Center Shiken). Roughly speaking, we currently have two kinds of components: one is answering, marking and visualization components for the Center Shiken; the other is question answering components. The entire system consists of compatible UIMA compliant components, based on the Kachako system. This design facilitates the users' solver applications to be more reusable, in addition to our own system.

1. はじめに

大学入試問題を解けるような質問応答システムの構築は複雑かつ多様な技術の集大成であり、専門的な知識を有する研究者ですら単独での研究開発はもはや不可能である。また、試行錯誤の過程で蓄積されるソフトウェア群は本プロジェクトで期待される重要な成果であり、誰もが利用しやすい形での社会還元が欠かせない[狩野 12b]。本稿では、再利用性と自動化という点に着目して、ツールやデータの共有・組合せ・実行を容易にするための統合研究基盤の現状を紹介する。

入試問題を解くことは質問に答える作業の一種といえるので、既存の質問応答システムの仕組みを応用できる可能性が高い。ただし、既存の質問応答システムが想定する質問と答えは入試問題とは異なる。そのため入試問題に対応できるよう多かれ少なかれ改良が必要であり、科目によっては質問応答システムのごく一部だけ再利用したいという状況も想定される。そうなると、単に質問応答システム全体をプログラムとして提供しても再利用が難しいため、コンポーネントに分割して整理する必要がある。

また、入試問題を解くプログラムを作成する作業は、実際には様々な試行錯誤が必要となり、プログラムに解答タスクを繰り返し実行させることになる。そのうち、入試問題を入力する部分と、解答を出力し採点する部分は共通化が可能である。プロジェクトにおける当面の目標は、大学入試センター試験を解けるようにすることであるので、まずはセンター試験の問題と解答について共通の研究基盤を提供することを狙う。

質問応答システムにせよ、センター試験問題の解答や採点にせよ、再利用性のためにコンポーネントに分割して共有し、研究者が興味ある部分に集中できるようなるべく作業を自動化・省力化するという目的に変わりはない。共有可能なコンポーネントは研究の進展と共に増加が見込まれるため、コンポーネントの形式を標準化した上で互換性を担保することが重要である。

我々は UIMA (Unstructured Information Management Architecture) [Ferrucci 06]を標準化の枠組みとして用いることにした。UIMA はまさに上記のようなコンポーネントの共有を意図して提供されている国際標準であり、Apache UIMA としてオープンソースで公開されている。UIMA はすでにさまざまな研究開発で用いられている枠組みで、実装が安定している。標準化の利点は、一旦 UIMA に準拠させてしまえば UIMA が担保する範囲での互換性ができ、その利用方法は UIMA 一般のドキュメントで学べるということにある。さらに、UIMA に準拠した実行環境であれば好みの環境で実行できるし、UIMA の API 等を用いて自前の環境に組み込むこともできる。また、さまざまな研究グループから UIMA コンポーネントが公開されているため、それらを同時に利用することも容易である。

コンポーネントの可搬性や意味的な互換性などをしっかり生かせば、UIMA 準拠という以上に開発者の自動化・省力化をサポートできるはずである。一般に UIMA コンポーネントの実行は、好みの UIMA 準拠実行プラットフォームを用いて UIMA ワークフローとして実行できる。実行プラットフォームはグラフィカルユーザインターフェースからプログラマ向けの API アクセスを提供するものまで様々である。中でも Kachako[狩野 12a][Kano 12b]は、テキストや音声などの非構造化データにおける相互運用性と自動化を実現する UIMA 準拠の実行環境であり、最も自動化を考慮して設計されている。そのため、本プロジェクトにおいて必要な相互運用性にかかわる一般的な議論はその中ですでに尽くされているといえる。本稿で述べるコンポーネントは、単に UIMA 準拠というだけでなく、Kachako の要求する自動化に耐えうる実装になっている。また、Kachako の提供する他のコンポーネントとの意味的な互換性も考慮されている。

本稿では、2 節でまず UIMA、Kachako、ACLIA について簡単に紹介する。3 節でセンター試験の解答・採点システムについて、4 節で質問応答システムについて記述する。なお、紙面の都合上詳述できなかった内容もあるため、[狩野 12a] [狩野 12b] [Kano 12b]を併せて参照されたい。

2. 背景と関連研究

2.1 UIMA

UIMA の実行単位はコンポーネントと呼ばれ、コンポーネントを組み合わせることで実行可能な UIMA ワークフローを作成する。UIMA 自体は基本的にコンポーネントそのものの提供はしな

連絡先: 狩野 芳伸 kano@nii.ac.jp http://kachako.org/kano/
神門 典子 kando@nii.ac.jp http://researchmap.jp/kando/

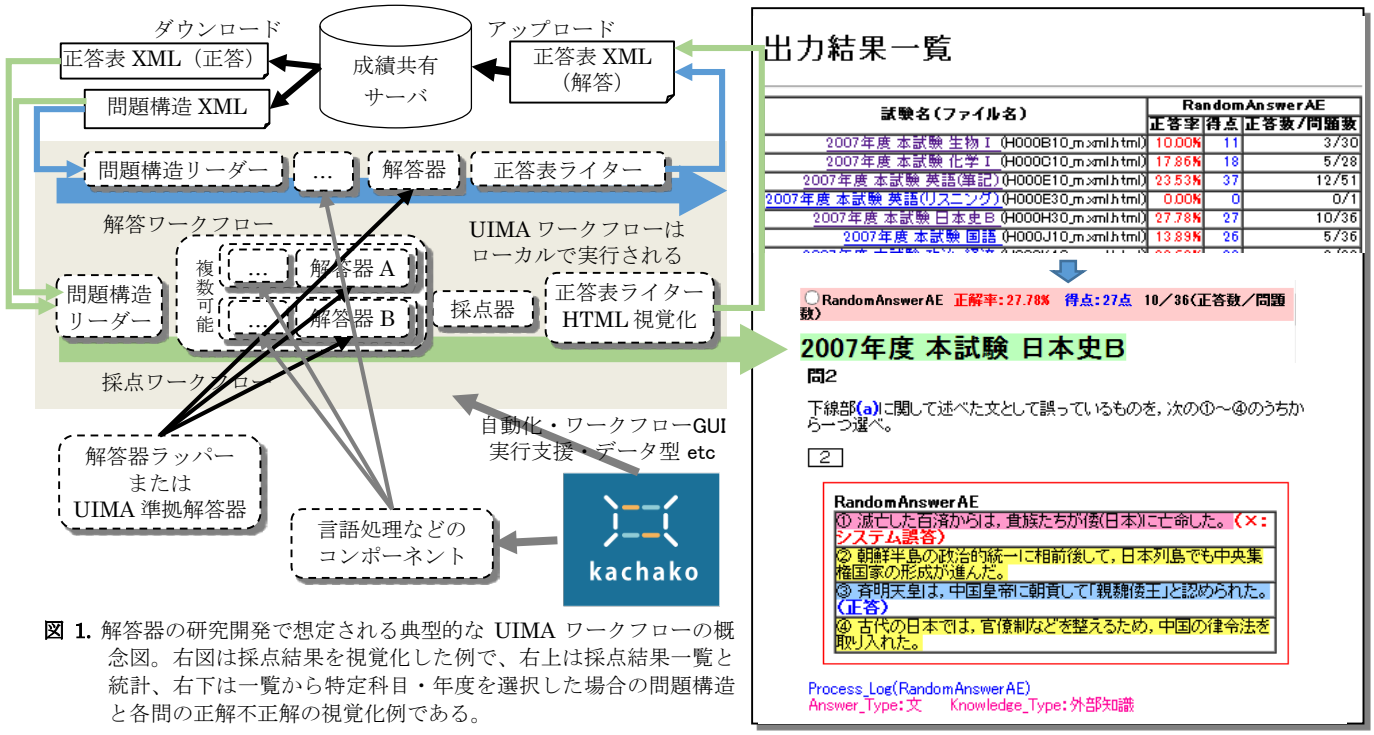

図 1. 解答器の研究開発で想定される典型的な UIMA ワークフローの概念図。右図は採点結果を視覚化した例で、右上は採点結果一覧と統計、右下は一覧から特定科目・年度を選択した場合の問題構造と各問の正解不正解の視覚化例である。

いため、UIMA の利用者は通常、自前のコンポーネントを作成するか、第三者のコンポーネントを使うことになる。UIMA コンポーネントは、設定パラメータなどのメタデータを記述する descriptor XML ファイルと、対応する実行ファイルに分離されている。UIMA API の公式実装には Java と C++があり、さらにそれらへの Perl、Python 等のバインディングも存在する。

UIMA ワークフロー実行時のデータ構造は CAS と呼ばれる汎用構造に統一されている。一般的な UIMA コンポーネントは CAS をひとつ受け取り処理結果を加えてから同じ CAS を返す。CAS は生テキストを保持する部分と、テキストへの付加情報を保持する部分に分かれている。テキストへの付加情報についてはデータの型付けが必須であり、開発者は type system と呼ばれるデータ型階層を定義する。UIMA 標準のデータ型は数値や文字列、配列など基本的な型のみのため、開発者間の type system 互換性を考慮する必要がある。実行には、コンポーネントの実行順序などを記述した UIMA ワークフロー記述 XML を作成する。

2.2 Kachako

Kachako[1]は UIMA 準拠の統合プラットフォームと互換 UIMA コンポーネント群を提供している。Kachako の目的は自然言語処理におけるユーザタスクを徹底した自動化によりサポートすることであり、プラットフォームおよびコンポーネントのインストール、ワークフロー生成、ウェブサービス展開、大規模処理、結果の視覚化、汎用比較評価などを全自動でサポートする機能を提供している。日本語・英語を中心に型階層とコンポーネント群も提供している。自動化を実現するためにはコンポーネントの可搬性と互換性が必須である。Kachako 提供のものや本稿で述べるコンポーネントは自動化に対応できる実装になっている。

2.3 ACLIA

ACLIA (Advanced Cross-Lingual Information Access) は NTCIR-7 および NTCIR-8 の質問応答タスク[Mitamura 10]である。ACLIA では、参加者がタスク中の興味ある部分のみでも参加できるよう、中間結果を定義してタスクを切り分けている。

3. センター試験解答・採点ワークフローと成績共有サーバ

本プロジェクトの最初の課題は、大学入試センター試験の解答器を作成することにある。試験問題の解答器は、解答する科目によって仕組みが大きく異なることが予想される。また同じ科目であっても、問題によってその性質が異なるため、一人の研究者で一科目分の問題をすべて解答できるようなシステムを構築するのは難しいと思われる。そのため、別個に作られた解答器を総合して実行できるような仕組みが理想的である。

3.1 センター試験の問題構造 XML と正答表 XML

本プロジェクトでは、大学入試センター試験問題アノテーション済みデータ（問題構造・問題分類）を作成した。国立情報学研究所との間でデータ利用許諾に関する覚書を締結すれば研究目的で利用可能であり、後述の成績共有サーバにアカウントを作成することでダウンロードできる。問題構造 XML は「問題用紙」に相当し、各設問の情報や下線部などの位置がマークアップされている。正答表 XML は「マークシート」に相当し、正解が記載されているが、以下で述べる解答器の解答を出力するのにも同じフォーマットを用いる。我々の提供するセンター試験の解答・採点ワークフローでは、最初と最後の入出力や非 UIMA ツールとの入出力にはこれらの XML 形式を用いるが、ワークフロー内では基本的に UIMA 準拠のデータ形式を用いる。

3.2 解答・採点のためのコンポーネント

センター試験の解答と採点を行うための UIMA コンポーネントとして、以下のものを提供する。

(1) 問題構造 XML リーダー

問題構造 XML を読み込み、UIMA 形式に変換する。ここで

[1] Kachako 公式ウェブサイト http://kachako.org/ を参照。

図 2. 成績共有サーバのスクリーンショット。ユーザの過去の成績の統計やグラフがカテゴリごとに閲覧できる。

読み込んだ問題はすべてワークフローの後段(解答器など)に渡されるため、必要に応じて科目など対象ファイルを指定する。

(2) 解答器ラッパー

非 UIMA 実装の解答器と接続するために、UIMA 形式から問題構造 XML に変換したものを渡し、解答として正答表 XML を受けとって再度 UIMA 形式に戻す。ラッパーではなく最初から UIMA 準拠で作られた解答器コンポーネントを使うこともできる。詳細は後述の「解答器の典型的な開発方法」の節を参照。

(3) 正答表 XML ライター

UIMA 形式の解答結果を正答表 XML 形式で書きだす。

(4) 採点・HTML 視覚化コンポーネント

次節で述べる特殊な採点ワークフローの結果を受け取り、採点を行ったうえで HTML 形式で採点結果を視覚化する。採点結果は各設問の正解不正解および解答器のログ出力に加え、センター試験としての配点による総得点と、配点を考慮しない正答率が算出できる。

単一科目であっても全種類の問題を解ける解答器は少ないと想定されるため、解答のない問題は不正解とはせずカウントしない正答率算出モードも実装した。複数解答器を同時に採点し並列して視覚化表示できるため、開発過程で作成された様々なバージョンの解答器の改善点などの比較が容易に行える。

3.3 解答および採点ワークフロー

前節のコンポーネントとユーザ自身の作成する解答器の組み合わせにより、解答または採点を行う UIMA ワークフローを構成することができる(図 1)。我々の提供する典型的なワークフロー記述を編集すれば、ユーザの作業は最小限で済む。

(1) 解答ワークフロー

問題構造リーダー、解答器、正答表ライターを順に実行するだけの単純なワークフローである。結果を後述の成績サーバにアップロードすれば採点を行える。また、「模試」などで正解が入手できない場合はこのワークフローを用いることになる。

(2) 採点ワークフロー

正解正答表が手元にある場合に、ローカル環境で採点を行うワークフローである。内部的には Kachako の比較評価機能を用いて正解正答表と解答を比較評価し採点するため、特殊な構成のワークフローになっているが、ユーザは提供するワークフローのうち解答器の指定を自身のものに置き換えるだけでよい。最後に HTML 視覚化コンポーネントを実行すると想定している。

3.4 センター試験解答器の成績共有サーバ

本節では、国立情報学研究所社会共有知研究センターの舛川竜治氏らが作成した成績共有サーバ [2]について述べる。成績共有サーバでは、ユーザが自らの解答結果を正答表 XML 形式でアップロードして保存すると、ウェブサーバ上で前述の採点システムが実行される。採点結果は他のユーザと比較共有することができ、科目や年度ごとの統計表示や過去の採点結果との比較などができる(図 2)。サーバ上で前述の採点結果視覚化の実行・表示もできる。ユーザ操作はすべてウェブブラウザで完結する。このサーバはもう一つ、ユーザ ID 管理下で問題構造等のリソースや実行ファイルを配布するという役割を持っている。

3.5 解答器の典型的な開発方法

本稿で紹介するさまざまなツールを用いて、各個のユーザが解答器を開発する典型的な方法がいくつか想定できる。いずれの場合も、採点・視覚化には成績共有サーバまたは採点ワークフローのどちらも利用できる。

(1) XML 互換

問題構造 XML を直接受け取り、解答として正答表 XML を出力するネイティブ解答器を実装する。ライセンスやシステムの複雑さ等の制約で解答器の共有が難しい場合、かつ自前の実装のみで完結させたい場合に向いている。問題構造・正答表 XML の処理以外の知識を必要としない。

(2) XML 互換+スタンドオフ入力

(1)に加え、解答ワークフローの前段に UIMA コンポーネントをいくつか配置し前処理として用いる。Kachako 提供のコンポーネントであれば、Kachako の定義する型階層の知識が必要。コンポーネントの処理結果は Kachako 提供のスタンドオフ形式(テキストフォーマット)で渡されるため、スタンドオフの解釈が必要。

(3) Kachako-UIMA 互換

入出力をすべて UIMA 形式で行う。コンポーネントとして、特に解答器の一部を分割したうえで他ユーザと共有できる場合に向いている。ポータブルな実装形式であれば Kachako 経由で自動インストール配信ができる。初歩的な UIMA の知識が必要。

4. 質問応答システムの互換コンポーネント化と結果の視覚化

コンポーネント作成に当たっては、1 節で述べたように、Kachako プラットフォームの要求するレベルの互換性・可搬性をもち、入出力条件の明示的な UIMA コンポーネントを実装するのが理想である。本研究基盤の課題は、そのような条件を満たしつつ、入試問題を解くのに利用可能な実際の処理を行うコンポーネントを提供することにある。作成したコンポーネントはオープンソースで順次公開の予定である。また、自動実行可能なものを中心に Kachako からも配信する。そのようなコンポーネントとして、まず日本語の質問応答システムを対象とした。

4.1 コンポーネント化の設計

質問応答システムのコンポーネント化にあたり、既存システム

[2] 東ロボ公式ウェブサイト http://21robot.org/ からアクセス可能。

図 3. 質問応答システムのコンポーネント構成例（左）と質問応答システム処理結果の HTML 視覚化例（右）。

として横浜国立大学で開発された MinerVA Factoid [Mori 05]とカーネギーメロン大学で開発された Javelin IV [Shima 08]をご提供いただき、MinverVA については Java で再実装のうえ、これらを互換 UIMA コンポーネント化した。その際、NTCIR の ACLIA[Mitamura 10]によるタスク分割をコンポーネント分割の基準とした。具体的には、質問解析・文書検索・回答抽出・回答選択の各コンポーネントである。また、いずれの質問応答システムも内部的に形態素解析や検索エンジンといった外部ツールを呼び出して利用している。こうした外部ツールも極力 UIMA コンポーネント化して切り出すようにし、細かな粒度でコンポーネントの置換や共有が行えるように設計した(図 3 左)。

コンポーネントの互換化設計における一般的な指針は、機械的に読み取り可能で明示的な入出力記述を行える形に実装することである[狩野 12a]。すなわち、コンポーネントの処理はなるべく自己完結するようにし、外部からみてコンポーネントがどのような役割を果たすのかを明確にする必要がある。言い換えると、コンポーネントのソースコードを解読しなくとも、入出力記述で役割が判断できるよう、適切な入出力を設定する必要がある。そうすれば、タスクに応じてプログラムを修正することなくより多くの場合で再利用できるはずである。さらに、入出力情報からワークフローの自動生成等も可能で、理想的な相互運用の形といえる。

そこで、UIMA の sofa という仕組みを用いて汎用性をなるべく保持するようにした。sofa (Subject OF Analysis)とは多面的なデータを保持するための仕組みで、単一の CAS の中に sofa とよばれる仮想的な CAS を複数保持できる。必要に応じて仮想的な CAS を別の CAS にコピーすれば、特定の用途を前提に作られたコンポーネントと形態素解析のような一般的なコンポーネントを混在させて利用できる[Kano 12a]。質問応答システムの場合は、新たなテキストが増える毎に sofa を生成するようにした。

4.2 実行結果の視覚化表示

質問応答システムを用いた開発を加速させるため、質問応答システムの結果表示に特化した HTML 視覚化ツールを実装した(図 3 右)。各 sofa の存在を前提にしているが、それ以外は基本的に任意の UIMA 標準のデータ形式を読みこめる柔軟な実装になっている。表示対象とするデータ型を指定でき、表示の最適化・高速化に対応している。また、各 sofa の代表データ型を指定すると、例えば解候補文字列をヘッダ一覧に表示できる。

謝辞

横浜国立大学の森辰則教授・国立情報学研究所の石下円香氏およびカーネギーメロン大学の三田村照子教授・嶋秀樹氏、ご協力いただいた東ロボプロジェクトメンバーの各氏に深謝申し上げたい。大学入試センター試験の問題および解答データについては、株式会社ジェイシー教育研究所が販売する「大学入試センター試験問題データベース　センターTen 2011　通常版全教科セット」を利用した。

参考文献

[Ferrucci 06] Ferrucci, D., Lally, A., Gruhl, D., Epstein, E., Schor, M., Murdock, J. W., Frenkiel, A., Brown, E. W., Hampp, T., et al. (2006) Towards an Interoperability Standard for Text and Multi-Modal Analytics. IBM Research Report.

[Kano 12a] Kano, Y. (2012a) Towards automation in using multi-modal language resources: compatibility and interoperability for multi-modal features in Kachako. *The Eighth edition of the International Conference on Language Resources and Evaluation (LREC 2012)*. Istanbul, Turkey.

[Kano 12b] Kano, Y. (2012b) Kachako: a Hybrid-Cloud Unstructured Information Platform for Full Automation of Service Composition, Scalable Deployment and Evaluation. *In the 1st International Workshop on Analytics Services on the Cloud (ASC), the 10th International Conference on Services Oriented Computing (ICSOC 2012)*.

[Mitamura 10] Mitamura, T., Shima, H., Sakai, T., Kando, N., Mori, T., Takeda, K., Lin, C.-Y., Song, R., Lin, C.-J., et al. (2010) Overview of the NTCIR-8 ACLIA Tasks: Advanced Cross-Lingual Information Access. *NTCIR-8 Workshop*.

[Mori 05] Mori, T. (2005) Japanese question-answering system using A* search and its improvement, *4*(3), 280-304. New York, NY, USA: ACM.

[Shima 08] Shima, H., Lao, N., Nyberg, E. and Mitamura, T. (2008) Complex Cross-lingual Question Answering as Sequential Classification and Multi-Document Summarization Task. *NTCIR-7 Workshop*.

[狩野 12a] 狩野芳伸. (2012) Kachako: 誰でも使える全自動自然言語処理プラットホーム. *2012 年度人工知能学会全国大会（第 26 回）*.

[狩野 12b] 狩野 芳伸. (2012) 統合研究基盤：質問応答システムの互換コンポーネント化による再利用性向上と開発自動化支援. *人工知能学会誌,27(5) 特集号「ロボットは東大に入れるか」*.