論文：「未夢拡張計画」

自然言語処理の研究にと始めた未夢の開発も、基本的な機能は実現できて、一段落ということになった。第一フェーズとしてはこんなものかなという思いである。無論最終目標は、ずっと先のこと、これまで１年掛けてきたが、最低後１年は掛かるであろう。

そこで、今回は最終形を明確にするべく、今後開発するプログラムのアウトラインを議論したい。

１．はじめに

今作りかけている機能、きっちり作り込みたい目標機能は次の通りである。

（１）文章内容問い合わせ応答（セマンティック Web）

文章の意味を把握して、内容の問い合わせに柔軟に答えるシステムである。基本は Prolog の推論機能を利用して、視点の変更にも耐え得るようなシステムを目指している。今は、未夢の意味解析結果から、Prolog 文を生成する機能を作った所である。

（２）文章要約

文章の意味を理解して、内容の要約を目標に応じて作って、自然言語文章で出力するシステムである。今は、未夢の解析結果を単に文章化するだけを実現している。未夢のデバックツールに使っている。

（３）文章構造推定

文章の構造を推定していき、文法が破壊されている場合でも、ある程度意味を推定して、自然な文章にして出力するシステムである。機械翻訳と組み合わせると、単語が少ししか辞書に定義されてない状況でも、文章のおおざっぱな内容は把握できるものとすることを目標にできる。今は、未夢の耐久テストで利用していこうと開発中である。引用文とか、コメントとかを持った文章でも未夢が適用できるようにする。

（４）テキストマイニング

自然言語処理で利用する辞書など、知識ベースをコーパスから作っていく事を目指している。今は、未定義語の抽出や、is_a 関係の知識ベースの構築などを行っている。is_a 関係の辞書はロケーションの推定に利用している段階である。

これについては世の中では技術が大いに進んでいるようなので、しっかり勉強して、未夢で実際に使えるレベルを目指したい。

（５）機械翻訳

未夢を利用した意味処理ベースの機械翻訳を目指す。現段階はパイロットシステムとして、トランスファー方式を実現した段階である。不備があって、大幅な修正が必要・・・この方式での開発継続は研究にならないと断念した。日本語文の分かち書き化プログラムを未夢で利用している。未夢の単語定義辞書はこの機械翻訳システムのものを使っている。

（６）知識ベース

オートノミックフレームワークでの辞書などの知識情報を統合的に管理するシステムを目指す。基本は RDF 型の形式を持った連想指向の問い合わせ応答システムとする計画である。物理的には、個性有るデータベースの混合システムとなるはず。今は、基本仕様を設計中である。オートノミックフレームワークにするところが野心の現れである。

以下に、現在考えている仕様を記述していく。

２．仕様説明

２．１　文章内容問い合わせ応答（プロジェクト名：香澄）

このシステムのために、未夢に保存されている文章解析結果を Prolog ライクな文章表現にして、文章知識ベースとして所定のデータベースに保存する。本システムは、全て Prolog ライクな命令セットを基盤に処理を行う。

文章内容の問い合わせは自然言語で行い、これを Prolog ライクな香澄命令セットにして、問い合わせファイルに書き込む。

問いは、文章知識ベースの項から特別な情報を出力だすように処理命令セットを追加することで実現する。特別な追加命令を付加された状況で、文章知識ベースを推論機構にかませて、頭から読み上げをしていくと、答えがでてくるというものである。

知識ベースの Prolog ライクな項は、DataElement クラスで統一的に表現する。１命令は、SentenceElement　クラスで管理する。文章全体は DocumentElement クラスで管理する。DataElement レベルを処理するのが ProcessElement クラスで、SentenceElement レベルを処理するのが SentenceProcess になる。

Prolog ライクであるから、処理制御とか、処理結果はメモリ上に固有のデータベースを置いて処理を管理していくことになる。そのデータベースクラスを、ProcessDataBase クラスとする。背後に RDB を持たせたいと思っている。作業用データベースだから、実行開始時に初期化する。特に、DataElement は実行時に一意に特定することによって、項への情報の追加ということを行うために、生成と管理は統一して行う必要があるので、ファクトリーパターンを使ってそれを行っていく。そのファクトリークラスは、ItemFactory とする。DataElement 本体はデータベースに保存して、項の書き換えに備える。

項に問い合わ情報を付加するプログラムは ItemAttachmentProcessor クラス、文章を解析するプログラムは PrologAnalizer クラスである。

全体を管理するプログラムは KasumiProcessor クラスとする。

２．２　文章要約システム（プロジェクト名：Wonnya）

文の生成においては、次のことを決定して行いかなくてはならない。

（１）テーマ
（２）文章構造
（３）文構造、文表現
（４）使用する単語
（５）使用する文法

テーマは要約のポイントをユーザから指定される事を通して決定することとしたい。当面は、「意志」とか、「感情」といったものは扱わないからだ。そのポイントであるが、次の表にまとめてみた。

【基本テーマ】

場所の遷移トレース	アクターの場所移動をトレースしていく。詳細化のレベルも指定する。
時間の経過トレース	アクターの行動を時間を追ってトレースしていく。詳細化のレベルも指定する。
アクターの行動追跡	ある場所、ある時間のアクターの行動を記述する。詳細化のレベルも指定する。

【拡張テーマ】

原因結果	ある時間、ある場所でのアクターの行動を評価する。詳細化のレベルも指定する。
目的、手段、結果	ある時間、ある場所でのアクターの行動を評価する。詳細化のレベルも指定する。
５Ｗ１Ｈ	ある場所、ある時間のアクターの行動を記述する。詳細化のレベルも指定する。

文章構造には規則性がある。起承転結とか、序破急とか、前書き・本文・結語とか。これはテンプレートを持っていて、部分テーマにブレークダウンすることで、テーマの処理で実現する。後は、文章の配置を制御するだけである。

文構造、文表現であるが、「だ」「である」か、「です」「ます」調にするかとか、倒置分にして強調するか、フォーカスはどの単語に掛けるか、単語の省略、指示代名詞の使用とか、細々として処理がある。文の構造とか、単語の選択には大きな自由度があるので、その制御は複雑になる。なんらかのポリシーをもって、文を滑らかに出力することが肝要である。

あとは、特定の表記パターンがあるから、その組み合わせを制御する機能を実現していかなくてはならない。

使用する単語は基本的に共起情報を元に決定していく。共起は、次に３つを考慮しなくてはならない。

（１）単語間共起

（２）意味記号間共起

（３）単語・意味記号間共起

また、意味の相同性を管理する情報（単語グループ）を設置して、文脈に応じて的確な単語を選択していくようにする必要がある。

最後に文法を決定して、文を生成することになる。文法は、単語に付属して定義されるもので、特に動詞は格をもつので、複雑になる。後は、活用語尾とか、付属語の接続とか、かなり機械的な話題もある。

文法ベースを決める辞書群を作り込み、精確な表現をしていくようにしたい。

２．３　文章構造推定（プロジェクト名：彷徨（かなた））

元々は、文法が破壊された文章でも意味処理を実現しようとして立ち上げたプロジェクトである。その中で、文章の枠組みというものの特定も意味処理として重要な部分を作るプロセスであるとの認識から、未夢の入力補助機構として、作り込む事へと拡張したものである。

単語や単語の配置から意味の推定、文法の推定を行っていくのは、共起情報（連想情報）を手がかりにしていくことになる。共起知識ベースというものを構築して、対応していくつもりである。

なお、未夢は文法がしっかりした文章を扱うので、未夢用にはもっと簡単なつくりで、分かち書き化データを並べたファイル作って、そこにパターン（コーパスや、共起情報から得られるパターン）を適用して加工していくことで、追加の意味処理を行っていく。

ここで決定された意味は、未夢本体でも利用できるようにする。意味記号の管理は単語ノードに次の表のような形式で保存する。

【単語意味記号】

第一意味記号	未夢の単語辞書（dict01）で定義している意味記号列
第二意味記号	彷徨からの意味記号列 [大分類記号@意味記号.値@意味記号.値@・・・@@]
第三意味記号	その他、拡張意味記号列 [大分類記号@意味記号.値@意味記号.値@・・・@@]

文章は、引用文や注釈などの構造を持っているし、文章の形跡結果は次々に改版していくので、データベースの１つのテーブルとして管理していく。レコードが部分文章で、そのレコードキーに文章が引用文なのか、注釈なのか、どのレベルの意味解釈なのかを、そしてその解釈の改版を表す記号を置く。レコードのデータ部は意味解析結果を置く。そして、テーブル全体がこのように構造化された文書ということになる。

２．４　テキストマイニング（プロジェクト名：夏樹）

共起情報を収集していく。また、単語定義もできる限りサポートして、辞書作り作業の負担を軽減していきたい。これについては多くの論文があるので、参考にして、いいものを作れればと願っている。

未夢プロジェクト全体で一番重要な機能だとの認識である。ここで得られる情報が未夢の精度を左右するからである。ここの情報がなければ未夢は作れない。

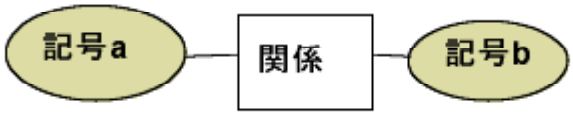
２．５　機械翻訳システム（プロジェクト名：瑞樹）

各国語で、未夢を作れば、あとは、意味記号のすりあわせをして、未夢本体の解析結果を他国語のものに置き換えられる。あとは、文章生成系を走らせるだけである。要約出力ならば、全ての単語を知る事もなく、早く実用化されるはずである。

ここでの研究は、意味記号の変換（文単位で意味を把握した上での単語選択）と、知識をどう訳語に活かしていくかという技術の開発が主目的になる。「前向きに検討したいと大臣が語った」は、対訳で「努力する気持ちはあるが、実現の確約はできない」という意味があるというのをどう表現するかである。注釈として表示するか、訳文本体に組み入れ、訳文を工夫していくか・・・。

２．６　知識ベース（プロジェクト名：Rhu）

辞書の構築ツール、辞書アクセスツールであるが、インテリジェントを持たせていくつもりである。連想アクセスにしてしまって、辞書の構造をユーザが知る必要がなくなるようにしたい。連想は、次のような３項関係として表現されるもの。それ以上ではない。



オートノミック・フレームワークのプロットタイプ作りという側面も強調して、作っていく。

例えば、関係と記号 a を提示されたら、記号 b を貰えるような汎用的 API を用意し、利用プログラムから辞書の構成、構造を隠蔽する。RDB の SQL みたいなものにするといえば分かりやすいと思う。

３．考察

これらのシステムは独立したものではなく、互いに力を出し合っていくことで、効率的な開発ができ、高度な技術を作り上げる事ができるものとなっている。互いに他を必要とするものである。文章解析プログラム未夢も何時か改版して、オートノミックフレームワークで実現するつもりでいる。今回の Rhu でのオートノミックフレームワークへの挑戦はその試行版でもある。上手くいったら、全てこのフレームワークに統一するつもりである。

４．まとめ

論文や製品説明記事を読むにつけ、現在すごい勢いで研究が進んでいるのを感じる。論文読んで自らを磨き、更に上の技術を発表していければ良いなと思うこの頃である。

昔から、選択と集中というものが大事であると言われているが、私は手広く研究をしていくつもり。分野がはずれていなければ、おおくのシナジー効果が期待できるから。考察でも述べた通り、どれが欠けても自然言語処理は完成しないのである。自然言語処理の最終目標は自律して言語を獲得して、利用していくシステムを完成することに有るから、どんなことも関係してくるのである。そして、重要でないことなどない。自律系がロボって出る事も、究極では必然であろう。それは、知能というものが身体性なものであるからである。フレーム問題や接地問題の解決には知能が身体を持っている事が重要であるという指摘はも１９９０年代より前から言われている事だ。

ということで、シナジー効果を最大限に生み出すという戦略で頑張りたいと思う。つま

り、あれもこれもだけれども、視点はぶれないでプログラムを開発していく。考察は物理的な、経済的な縛りはないから大きく、広く、自由に発想して行くつもりである。

おわり