計画：「彷徨」

文法が破壊された文から構成されている文章を認識するプログラムを作っていくことになった。プロジェクト「彷徨（かなた）」である。今回はこれの仕様をまとめてみたい。

（例文）学校から、夕日。楽しかった。歩いた、家。
空、晴れて、夕日、明るい。
・・・・・・・

この例文のような文が集まった文章も解析の対象としていく。

１．処理の構想

辞書に単語とかコーパス、格情報などをフレームとして持ち、入力文書の単語、単語列とマッチングを取っていき、メモリに推定した文構造、意味構造を構築していくという方式を取る。マッチングは基本的に単語、単語列の共起情報を捉えて行う。彷徨の解析処理では文法も共起情報ということで、格フレームとして提供していく方針である。

処理のアウトラインを下図に示す。

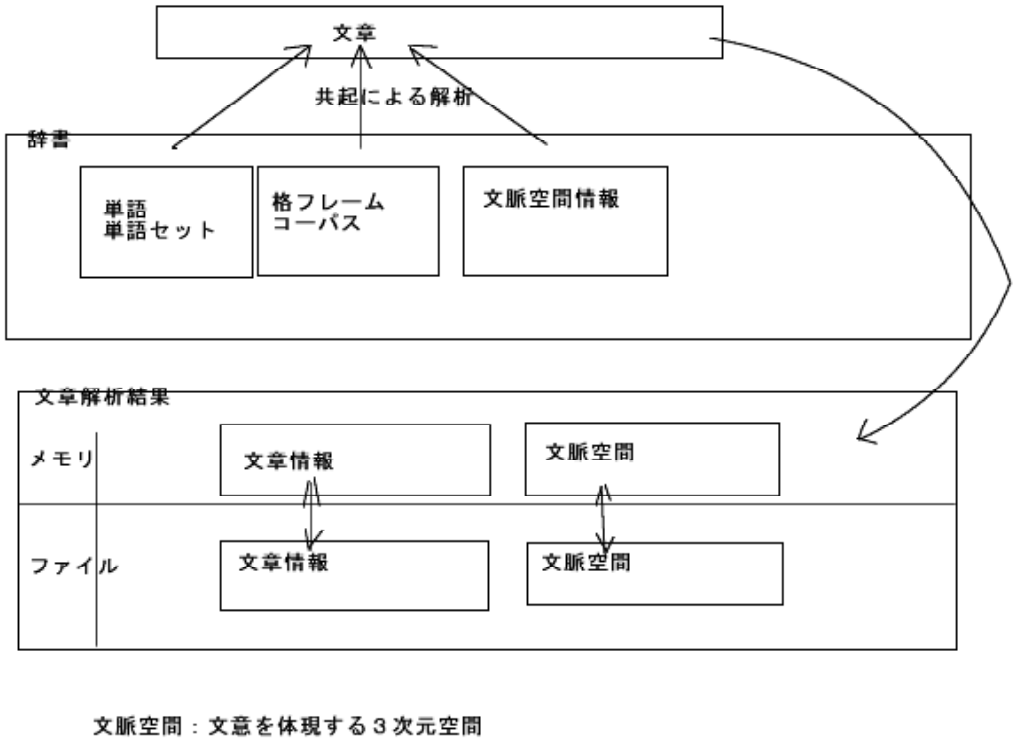

図１　彷徨の処理スケルトン

注意しなければならないのは、文法を破壊された文章の解析には、文脈が重要な位置を占める事である。前の文章で解析したオブジェクトの関係を利用して、後段の文章を解析する。また文法が明確な文章によって確定したオブジェクトの関係を全段で利用するということもしていなくてはならない。それで、オブジェクトの関係を把握するために、文脈空間を導入することになった。これは遠い将来の課題として考えていたものであるが、今回は前倒しで、研究的に実現する事にした。何処まで出来るか分からないが、頑張ってみたい。

ファイル化はファクトリーパターンで実現する。解析結果の情報を保持するオブジェクトは識別子で関係づけることとする。ファイル化データの複雑な処理管理はファクトリーオブジェクトに吸収していく。

２．永続化（ファイル化）

多くの辞書データベースから構成されるが、利用しやすいように連想アクセスとし、呼び出し形式を統一する。大きな連想アクセスシステムとなる。連想は基本的に次の項目からなるとする。

（１）データクラス

連想の範囲を規定する。

（２）対象記号オブジェクト

連想の起点の記号を定義するオブジェクトである。

（３）連想記号オブジェクト

連想を定義するオブジェクトである。

（４）連想記号オブジェクト

対象記号から、規定の連想記号で連想されるオブジェクトである。

（５）確信度

連想の強さを示す数値

連想は、連想元と連想先を入れ替えることもありうるので、「受身形」も持ち、それは連想アクセスシステムで管理していくものとする。

ファイル化される文章解析結果データは次のレベルになる。

（１）要素データエンティティ

単語のレベルの解析結果を保存する。

（２）文データエンティティ

文単位の解析結果を保持する。要素データエンティティをコレクションとして持つ。

（３）文章データエンティティ

文章単位を管理する。文データエンティティをコレクションに持つ。

３．ファクトリー

データの永続化を利用プログラムから隠蔽するためにファクトリー実現する。ファクトリーでは、文単位にファイル化処理をするが、すでにファイル化されているデータエンティティがあればファイル化しないとか、逆に、ファイルからデータエンティティを読み込むときにすでにメモリ上にあればそれをリンクするだけにするとか､家事全般を担当する。メモリ上はポインタで参照しあうが､ファイル化するときには識別子で管理するしかない。この作業が重要である。それら処理をまとめると、

（１）識別子を用いたデータエンティティの管理

（２）メモリ上のデータエンティティとファイル上のエンティティの重複しない一貫性の管理

（３）ポインタと識別子の管理

（４）エンティティの更新管理

（５）新規エンティティの管理、カレントエンティティの管理、メモリ上のエンティティの管理

（６）エンティティの順序管理

（７）コレクションの管理

ファイル中のコレクションはテーブルとして管理し、コレクションへの参照はコレクション識別子で行う。

４．意味・共起管理

意味は、階層的に管理できるようにする。「ある意味カテゴリー中のある意味」を管理したいときが多々あるからである。

曖昧性処理を考慮して､データには全てデフォルトを持つようにする｡解析結果である、データエンティティについても、デフォルトエンティティというものを設ける事を徹底させたい。曖昧性処理はこのデフォルトが何かということを同定していくことを基軸に実現したい。

おわり