

人間は、スペルミスだらけの文でも簡単に読み取ることができます。例えば、次のような文です。

(例文1) Tthat isu aa penn.

これはどうして可能になっているのでしょうか。コンピュータには無理なことです。少なくとも、キー、パリュウのV S A Mを知識ベースに持つとすれば不可能なことです。かなり大きなアルゴリズムを用いて実現しなくてはなりません。しかしながら、(例文1)を簡単にアクセプトできることが人間の知識ベースの本質なのではないのでしょうか。この点を追求していくことが、未来の知識ベースの機構はどうあらねばならないかを知る手がかりとなりそうです。

1 . ASAM(Associative Storage Access Method)の提唱

(例文1)から正しい文を連想する手法を考えてみます。それは1文字毎に正しい単語、文を連想させ、もっとも確信度の高い文を答えとして返す知識ベース・・・ということになります。これしか考えられません。1文字ずつ沢山の連想をしていって候補を上げていって、全ての文字が入力を終えたとき、もっとも支持された単語とかその上の文を持って、答えと成すということです。

基本的には連想なのですが、候補同士に互いに作用しあって、・・・例えば共起関係が大きい単語同士は強めあうわけです・・・意味的にあり得ると判断してその連想選択を強めるというような手法(弛緩法)があるでしょう。このような一連の作業をこなす機構をASAMと呼ぶことにします。

ASAM を実現するには連想は単語や文の数ほどありますから、実時間で実現するには超平行処理が必要です。すなわち、ハードウェアから設計していかねばならない問題なのです。ここでは、自然言語処理を考えましたが、画像認識の問題でも、無数の候補パターンと入力画像の一部が一致するかどうかという話になり、超並列の連想機構が必要になります。だから、ASAMは未来の人工知能の肝の技術ということになります。

ASAM はどんな構造になっているべきでしょうか。超平行処理ですから、メニーコアM P Uが必須です。超平行を可能にするには配線を簡略にして、できるだけ多くにコアをプリントできなくてはなりません。簡易命令セットM P Uでなくてはなりません。その仕組みは、パターンマッチングが高速にできればよいということで、正規表現の入力を知識ベースとマッチングさせるということを想定して、M P Uにはオートマトンが命令メモリに設定できるだけで、あとはオートマトンを実現する命令回路があるだけでしょう。その命令は基本的に、compare と jump(オートマトンの次の状態に遷移する)でしょう。ステート記録はステートをポインタするレジスタがあるだけでよいでしょう。レジスタでなくてメモリにポインタを置くべきかも知れません。非決定性オートマトンにすべきかも知れないからです。そうすると store 命令と load 命令も必要ですね。命令が大きな容量を必要と

するからです。終端状態もメモリに覚えることになるでしょうし。ということで、ASAMのMPUはすこぶる簡単なものに設計できることが分かりました。

あとは弛緩法で無数の確信度を調停していく作業になります。これもASAMのMPUで実現します。これは、知識ベースを工夫することで、実現します。知識ベースアクセス手法を高速なハードウェアで実現することになります。それは、プライミング手法です。すなわち、最初のアクセスには別の作業用の知識ベース(テーブル)---テーブルは特別なコアメモリに保存することになるといいます---にヒットした単語とか文のアドレス(URL)を取ることで、2回目以降はヒット数を作業知識ベース上でカウントしていくことです。ここにはヒット数カウンタ項とマスク項があって、他の選択によって促進的に確信度を高めるか、抑制的に確信度をマスクしていくかを記録できるようにしていきます。カウンタアップはASAMのMPUが行いますが、調停は難しい処理になるので、専用の汎用CPUで実現していくことになるでしょう。うまい調停アルゴリズムがあれば、ASAMのMPUでも実現できるかも知れません。

ASAMの存在するストレージはHDDでもフラッシュメモリでもかまいません。大容量であることと、全内容を高速にサイクリックに読み出せたり書き込めたりできるものであることです。ストレージのハードウェアのサイクリックアクセスの速度がASAMの性能になります。MPUの知識ベース側レジスタにストレージの内容を順次いれていく。MPUは無数にありますから、一度に多くの入力パターンとマッチングができるのです。マッチングはMPUの制御メモリに保存してあるオートマトンで制御します。終端記号に一致すれば受理です。

ASAMをランダムアクセスすることを可能にするのは、データとインデクスを独立したストレージに保存するにすればよいでしょう。インデクスをランダムにたどって、データアドレスを得て、それでサイクリックアクセスが来たときそのデータをコピーするにすればよいのです。

ASAMの検索対象データは¥{と¥}で囲まれたところと規定すれば、ファイルの制御データ部を無視した検索を高速に行うことができます。

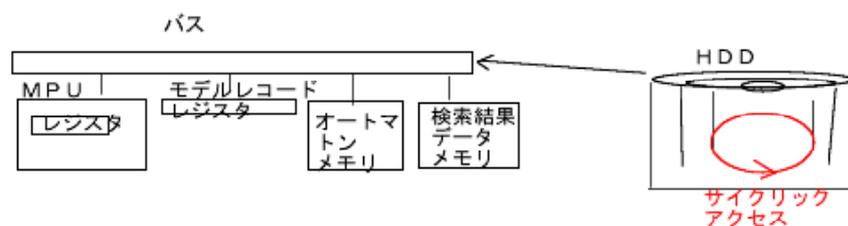


図 ASAMの構造

2. 画像認識への ASAM の応用

画像認識は次のような処理になるはずですが、

(1) 2 値化とオブジェクト切り出し

カラー画像を白黒の 2 値化します。これは画像にある図形を予測してオブジェクト切り出しと同時にされます。予想される図形を残すように黒画素を設定していくのです。

(2) オブジェクトの 2 項関係解析

2 値画素図形のうち、黒画素の固まりを点としてみなしてその相対配置関係から何の図形かを推定していきます。目とか鼻とかですね。無数に候補が連想されるでしょう。ここに ASAM が必要になるのです。

(3) オブジェクトの形解析

点として 2 項関係を解析しましたが、その解析結果の確信度を高めるために、各黒画素の形状を解析します。これも結局は点の 2 項関係の解析に落とし込まれるのです。解析のネスト関係にあるのです。だから、ここでも ASAM が必要になるでしょう。

(4) 大きな枠組みでの図形の意味を同定

各レベルで図形に意味を連想づける。意味が連想されるものだけが正しい解析結果となるのです。これも連想とパターンマッチングです。

(5) (1) から (4) を何度も繰り返す。予測の図形を弛緩法で絞り込みながら、最終結果に至るべく努力する。

画像認識用の知識ベースは、オブジェクトの 2 項関係を記述するものになります。オブジェクトを点と見なしたときの 2 項関係と、オブジェクトを分解した図形を記述するオブジェクトへの参照を記録するものです。

では、ASAM を用いた画像認識の手順を記述してみます。

先ず学習により画像オブジェクトの特徴を記述する知識を獲得していきます。これには雑音のない環境での解析を行い、その結果を知識ベースに記録していくこととなります。雑音のない環境は現実にはあり得ないわけで、そこは基本的に生得的に目とか顔とか体、森みたいなものは知識として持っているべきなの分かります。これらを用いて、雑音を取り除き、画像を 2 値化していきます。

画像の特徴を記述するのは、基本的に 2 項関係であろう。2 項関係は位置関係ですから、オブジェクトを点とみなしてその相対位置関係を記述することになります。画像全体との関係ですから、また状況パラメータも記述することから、2 項関係といってもある文脈のもとの 2 項関係です。文脈はあるモデルの URL で記述します。モデルの中に 2 項関係で様々な記述を持ちますから、モデルも画像の特徴記述データであるわけです。

次に、画像認識をしていきます。画像を特徴点とか線に分断します。領域と判断する部分もあるでしょう。そういう基本的なデータを得ます。そのあとは、線も領域も点とみなして、2 項関係を解析して得ていきます。その 2 項関係を ASAM により高速平行でどのモデルのパターンかを調べて候補をもとめていきます。全点と線分、領域についてこの操作をすると、モデルの確信度の分布ができます。もっとも高い確信度を決め打ちにして、再度画像データを解析していきます。あり得ないところは分離するとか、解釈し直すとか

していきます。領域はその内部を更に解析していきます。

このような弛緩法が画像認識にも自然言語処理にも重要なのです。

3 . 自然言語処理への A S A M の応用

文認識は次のようになるはずですが。

(1) 文を語列で把握する。この語列が解析の基盤データになります。

(例文) 私はカモメ。

(語列)(私 は カ モ メ 。)

(2) 単語を同定 (形態素解析、分かち書き化) します。これは辞書の総当たりになりますから、A S A M が必要になります。宴会で他のグループの会話には注意がいかないものですが、好きな人の名前がでたりすると直ぐに注意がいて、聞き耳を立てます。これは、会話を語列の総当たりで、自分が興味ある事項とパターンマッチングしていることを示しています。最長一致法などの複雑な仕組みを利用していないことを示しています。

複数の解釈がありますから、解析結果を語列範囲とその単語との組み合わせ情報をオブジェクトとしてデータの固まりにして、語列基盤データを指し示すネットワークとして持つことになります。利用したデータを全て解析結果として持ちます。組み合わせてはいけません。情報はどんどん追加していただくだけです。

(候補は全件で) 私

私は

私はカ

私はカモ

私はカモメ

は

はカ

はカモ

はカモメ

カ

.....

(3) 係り受け解析をします。

単語列情報を基に単語間の関係を解析します。基本的には可能性全部を修飾・被修飾の形でオブジェクトデータにして単語ネットワーク (木構造ネットワーク) に追加していきます。組み合わせません。2 項関係として表現していきます。

(2)と(3)で、曖昧性が大きいと処理に時間が掛かります。あるモデル知識ベースを基本に、単語の共起関係を得て、もっともらしいもののみをデータとして残す処理がひとつようです。それは解析とは平行に知識ベースの処理として実現すべきでしょう。それによっても、解析結果を全てネットワークで持つことのメリットが生まれます。

(4) 格解析をします。

これも表層表現(動詞と名詞の2項関係)に解析結果の深層表現を付け足します。これも複数あります。曖昧性は逃れられないことです。

この表現から、知識ベースのモデルからもっともらしい解釈を絞り込みます。文脈によって最終決定をすべきことがありますので、ここでは最終結果を得ることは一般には言えないことです。

(5) 文脈処理をします。

ステージ、シーン、カットを見出していき、全体の話の流れをシミュレートや推論していきます。そうして、文の最終解釈を決定します。指示代名詞や欠落語の解釈もしていきます。

4. 思考と身体制御への応用

思考空間は身体制御と密接に関係しています。針に糸を通す作業では、針と糸の位置関係を視覚から得て、未来の状況を思考空間に持ち、その差分でどういうアクションをすべきかを計算して手を動かします。このとき、身体制御空間と思考空間で座標変換が行われるでしょう。

手を動かす制御世界のデータと思考空間のもつデータは独立であるべきでしょう。制御世界のデータは「こうゆう作業のときにはこういう一連の作業を自動で行う」というモデル予測制御で実現すべきでしょう。そうしないと、行動を学習できないからです。行動はセンサとアクチュエータのデータ連鎖なので、思考というものとは独立であるべきなのです。

思考空間からどの制御世界のある動作項目を選ぶのは、連想によるものでしょう。連想のプランニングは思考空間が行いますが、連想結果は自動遂行になります。連想を柔軟に結びつけるためにもASAMが必須になります。

5 . 認知の基本となる知識ベースのモデル群

知識はモデルとして構造を与えられているデータの集まりであるはずですが、それは、「人間」が「目」や「鼻」などを持ち、「行動し」、「食し」、研究したりしていくことを把握すべく知識が構築されているはずだから言えることです。

自然言語処理に象徴されるように、認知は曖昧性のあるデータを基に行われます。何故、曖昧性が許されるかということ、確固とした知識ベースが共有されているからです。知識を伝えあうとき、分かっていることは情報として発信しないという原則は曖昧性を含みます。曖昧であるが、人間同士では明確な曖昧性のない情報として伝わっていく。これは知識ベースがいかに認知に重要であるかを物語っています。画像認識や自然言語処理の技術書を読んでもこの点を全面に出している本がありません。実に残念なことです。表層の文法や意味論を展開しても、十分な議論にはならないと思うのですが、何故か、研究の重点は知識ベースに無いようです。ここを打破する研究開発が待たれるゆえんです。

知識ベースはどのように作られていて、どのように使われていくべきでしょうか。「人間」とはモデルとして関連情報を蓄えているものでしょう。そこからは、「目」とか「学校に行く」とかの概念を他のモデルとして、そこにポイントする記述があるでしょう。こうして、モデルのネットワークになっていつのが知識ベースです。学習機構を簡単にするには、体系だった整理は不要としていかねばなりません。「ポイント」はアドレスでなくて、モデルの名前であるべきです。そうして、モデルの中にどんどん情報をため込んでいく。コーパスとなっているのがモデルです。抽象化も新しいモデルを作ってそこに情報をため込むことです。「太郎」や「花子」は「人間」という風に抽象化されますが、「人間」のモデルに「太郎」や「花子」の名前が記述され、なにかの拍子に「食事する」というような事象が共通だと、「人間」のモデルにも「食事する」が記述されるようなものです。そんなレベルの学習機構が必要なだけです。

知識ベースの使われ方は、全件内容検索で情報を特定し、そのレコードを解析していくこととなります。インデックスは複雑にするだけです。ASAMを使って実現することとなります。人工知能のモデルが全ての情報処理のピボットになり、認知とはこのピボットを発見することなのです。

6. まとめ

人工知能の基盤は結局、連想に基づく演算によっていると考えることができます。連想をいかに大規模に、平行して処理していけるかに人工知能の性能、品質が掛かっていると云えます。

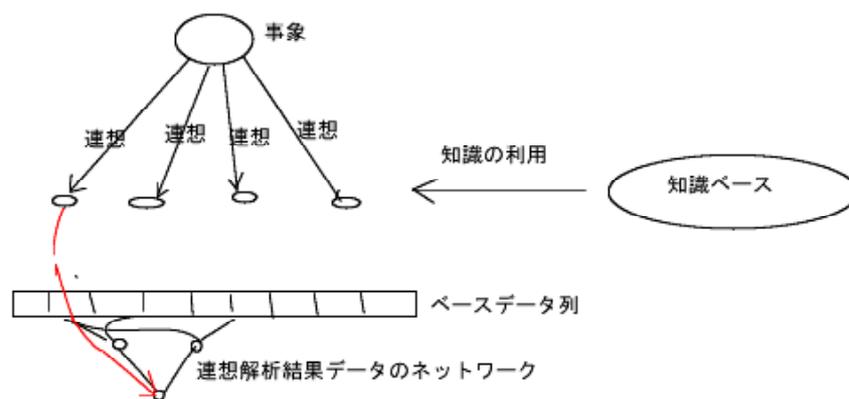


図 認知と構造

画像認識と自然言語処理のプロセッサオブジェクトを考えてみました。以下の通りです。

(1) 学習システム

LearnImageModel 画像モデルを創る

LearnLanguageModel 言語モデルを創る

(2) パターンマッチ/解析

AnalyzeImage 点や線分の画像の2項関係を画像モデルを参考にして解析していく。

AnalyzeLanguage 自然言語の文章を解析していく。言語モデルを参考にしていく。

(3) 文章生成

GenerateLanguage 文章を創っていく。

(4) 連想機能の実現

AssociateImageModel 画像モデルベースの連想

AssociateLanguageModel 自然言語モデルベースの連想

(5) 解析データ管理

- ControlImageAnalyzeInformation . . . 画像解析データの管理
- ControlLanguageAnalyzeInformation . . . 自然言語解析データの管理

(6) 弛緩法の実現

- ImageRelaxProcess 画像解析用弛緩法の実施
- LanguageRelaxProcess 自然言語処理用弛緩法の実施

(7) 評価プロセスの実現

- ImageEvaluation 画像認識の評価の実現
- LanguageEvaluation 自然言語認識の評価の実現

おわり