

パターン発見は人工知能の基本中の基本の機能です。基本的に、2度以上同じ記号（オントロジー）の構造を持つ知識（コンセプト）が現れれば、それをパターンとする・・・というようなこと。特に、その回数が多ければなおよいわけで、記号列の発生する回数をカウントしていくこととなります。

コンセプトはオブジェクトですが、これをRDBなどに保存するため、テキストでの非構造化データとします。パターン発見ではこの非構造化データを利用していくこととなります。記号の関係がオブジェクト内での関係を示しているからです。

つまり、なにかオブジェクトデータ（コンセプト）が発生したとすると、そのテキスト化非構造化データを見ます。その一部でも既存のデータの非構造化データの一部とマッチングすると、そのマッチングしていた部分を切り出してパターンとします。そのパターンがすでにあれば、カウントアップします。このパターンマッチングは全件検索になりますから、高速でしかも並行処理ができないといけません。基本的に、テキストの中のオントロジー記号の記号列が一致するかどうかというようなことで、テキストマッチングをして、全オントロジー記号が一致したら、オントロジーの実際の配置をオブジェクトで観て、構造としてもパターンとして一致していることを確認するという手段で処理を進めることになるでしょう。そうして、パターンは特別な領野に蓄えていけばいいのです。

ということで、パターン発見は力技で実現できることが分かりました。

パターンの形式：(パターンオブジェクト、パターンのテキスト非構造化データ形式、カウント)

特別なメモリ領域にパターン記述を蓄積していくこと。

おわり