

隠喩理解

6T-7

— その検出の実装方式 —

土井 晃一・田中 英彦

東京大学 工学部

1. はじめに

自然言語処理に止まらず、自然言語理解を計算機上で行おうとすると、発話どうりの言内の意味の理解だけではなく、会話の含みと呼ばれる言外の意味の検出、その場の状況の理解、資源利用などの文脈理解、新しい状況に対する話者の認識の仕方の解析が必要である。これらを扱う方法の一つに人間の思考の中心をなす類似概念の新たな創造を行う隠喩を理解する方法がある。

我々は隠喩理解を行う第一段階として隠喩の検出を行う。その実装にあたって、観念的な事を扱うために意味ネットワークを用い、実在的なことを扱うために状況意味論を用いた。本稿ではシステムの全体構成と検出のユニットについて述べる。

2. 隠喩理解

隠喩理解は言外の意味の解析、会話のなされている場の状況の理解、文脈理解、新しい状況に対する話者の認識の仕方の研究が可能になるなどの新しい観点から再び研究が進んでいる。応用面としては、既存単語の新しい意味、新単語の意味の解析も可能となる。単語のニュアンス、言外の意味の問題を扱うため、機械翻訳にも利用できる。

会話における言外の意味の解析の必要性は次のような例で示される。

A: 「これから銀行へ行きます」
B: 「四時半ですよ」

という会話があったとするとBの発話は現在の時刻をAに告げようとしたのではなく、四時半ではもう銀行が閉まっていることを告げようとしている。このように、言内の意味(時刻が四時半である)だけではなく、言外の意味(今から銀行に行っても無駄である)を発見しないと会話の理解はできないことになる。(1)

その場の状況の理解とは、話題と直接関係ない事が発話されたときそれが話題とどう関係するのかを調べることが必要となる。例えば、隣家のことが話題になって、突然、

「隣の花は赤い」

という発話がなされた場合、本当に「隣の花」それ自体

を話題にしようとしているのではなく、人の物は良く見えるものだということを戒めている発話として理解されねばならない。もちろんその場の状況の理解は言外の意味の検出と密接な関係にある。(2)

文脈理解とは、文脈によって文の解釈が変わる場合などに必要となる。例えば、

「人間は人間である」

という発話は普通に発話された場合と、論理学の本で例として挙げられた場合とでは文の解釈が変わってくる。このように文脈理解も会話の理解には不可欠である。

新しい状況に対する話者の認識の仕方の解析とは、現実の世界を話者がどのように理解したかを示すことである。例えば、酒を飲んで酔ったあびくにあばれまわっている人を見て、

「彼は少し酔っている」

という発話がなされたとすると、この話者は酔ってあばれまわっている人間を形容するのに「少し」という言葉がふさわしいと認識したことになる。(3)

隠喩理解を行うことにより、以上のようなことが解決される。

3. システムの全体構成

システムの全体構成は、構文解析の終わったものを入力とし、静的な既存の知識を入れておく意味ネットワーク部、動的な知識を入れておく作業領域、検出ルーチンからなる。意味ネットワーク部は単語の観念的な意味を表現するいわゆる意味ネットワークとその単語の使い方

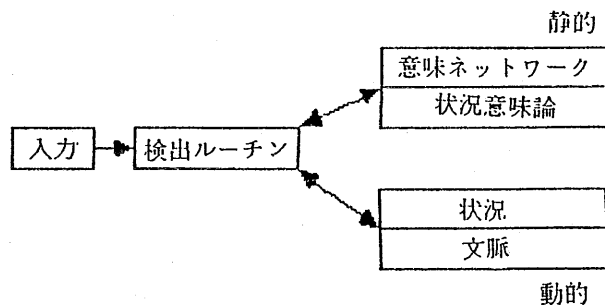


Fig 1 システム構成

を記述した構造部とからなる。作業領域は複数の入力からの隠喩の検出のために設けられている。検出ルーチンは意味ネットワーク部と作業領域を検索することによって検出を行うことになる。この意味ネットワークでは、包含関係、属性関係、対立関係の三つの関係だけをあらわし、そのほかの関係は構造部が表すことになる。

4. 隠喩検出

隠喩検出はスペルベル・ウイルソンの呼び起こしの原理に基づいて実行される。スペルベル・ウイルソンによると、

- (1)明らかに偽である場合
- (2)不条理である場合
- (3)有意性を欠く場合

呼び起こしが生じ、発話に対する再解釈が行われることになる。我々は各々に対して次のような検索を行うものとする。

- (1)意味ネットワーク内の検索
- (2)構造部の検索
- (3)作業領域内の検索

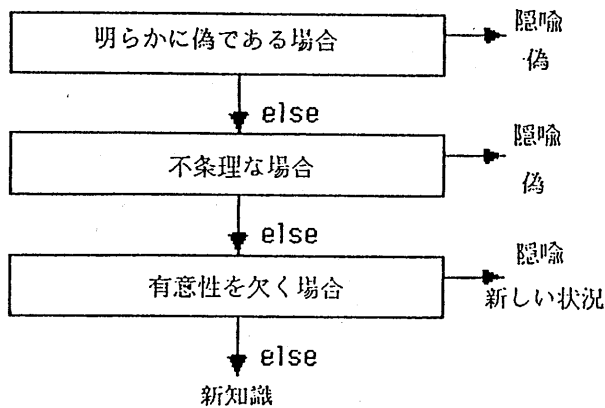


Fig 2 システムの流れ

実際の検出の仕方を以下に述べる。

5. 隠喩検出ユニットその一

ここでは意味ネットワーク内の検索を行う。検索の内容は、包含関係のチェック、対立リンクのチェックである。例として、

「人間は狼である」

という発話から隠喩を検出してみる。実際の入力は

in s : at l : is, man, wolf ; yes

という形である。これを意味ネットワーク内を検索して包含関係の矛盾を見つけだすことになる。

6. 隠喩検出ユニットその二

ここでは構造部内での検索を行う。検索の内容は、構造部内に記述してある状況に入力の状況があてはまるか

どうかを判定することである。例として前にあげた

「彼は少し酔っている」

という発話を検出してみると、実際の入力は、

in s1 : at 11 : drinking, A, much ; yes

in s2 : at 12 : going wild, A ; yes

in s3 : at 11 : recognizes, B, s1 ; yes

in s4 : at 12 : recognizes, B, s2 ; yes

in s5 : at 12 : saying, B, Z ; yes

in s6 : at 12 : gets drunk, A, a little ; yes

(11 < 12)

DU : = at 12 : speaking, B ; yes

addressing, B, A ; yes

saying, B, Z ; yes

という形をとる。gets drunk (酔っている) というノードは構造部を持ち、

in s1 : at 11 : drinking, a, much ; yes

at 12 : going wild, a ; yes

at 12 : gets, drunk, a, much ; yes

という記述があるから、a littleとmuchを意味ネットワーク上で調べると対立リンクが張ってあることから隠喩であることが検出されることになる。

7. 隠喩検出ユニットその三

ここでは状況、文脈に照らし合わせて発話での隠喩の検出を行う。例えば、

「隣の花は赤い」

という発話は単独では解釈できず、文脈によって解釈を決定しなければならない。これは、動的な作業領域を検索し、隣家の花の事が話題になっていたかどうかを調べることによって諺として使われたのかどうかを判定することになる。

8. おわりに

このように隠喩理解を行うことにより、いろいろなことが解析できる。今後は詳細なアルゴリズムの決定と試作を行うことを予定している。

参考文献

- (1) 安井 稔 (1978), 言外の意味, 研究社
- (2) 菅野 盾樹 (1985), メタファーの記号論, 勁草書房
- (3) 佐藤 信夫 (1978), レトリック感覚, 講談社