

隠喩理解

— その検出システムとデータ構造 —

土井 晃一 · 田中 英彦

東京大学 工学部

5T-6

1. 序論

自然言語処理に止らず、自然言語理解を行うためには、象徴と指示関係の理解の他にも、単語の使い方、言外の意味の解析、状況理解、文脈理解、話者の認識の仕方の解析が必要となる。これらを扱うアプローチの一つとして、隠喩理解がある。

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。その際観念的な事を扱うために意味ネットワークを用い、実在的な事を扱うために状況意味論を用いた。本稿では、隠喩理解の意義を述べ、さらに隠喩検出システムの構成とそのデータ構造について述べる。

2. 隠喩検出の意義

本論文では、隠喩理解を隠喩検出と隠喩理解に分け、隠喩検出について述べる。その際隠喩だけを検出の対象とするよりも換喩、提喩も含めた、コノテーション(判示的意味作用)を検出の対象とした方が、応用上望ましいので、それらを検出の対象とする。

3. 隠喩理解の応用

隠喩理解を行うことにより、既存単語の新しい意味、新単語の意味の解析が可能になる。つまり隠喩理解は新知識獲得の一手法である。

機械翻訳に応用すれば単語のニュアンスの問題を扱うことができる。すなわち一般の翻訳を考えると、各国語の文化的背景を持った隠喩は頻繁に出現するものであって、隠喩理解は不可欠の要素になる。

4. 隠喩検出システムの全体構成

既存の単語についての知識を保持する静的な意味ネットワーク部と、状況、文脈を保存する動的な作業領域、実際に検出を行う検出ルーチンとからなる。

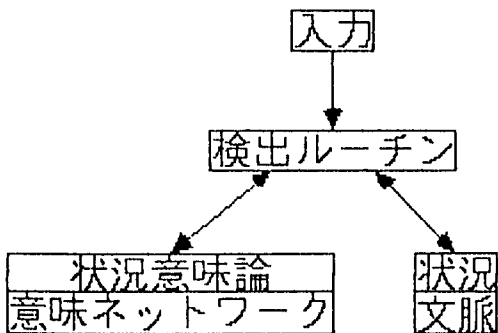


図1 システム構成

状況の移り変わりはcourse of eventとして順次入力されることになる。

全体はプロログで書かれており、意味ネットワークのノードはアトムでリンクはクローズで記述してある。状況意味論の部分もプロログで実装した。

5. 隠喩の検出方法

検出の原理はグライスの会話の作法を改良したスベルベル・ウイルソンの「呼び起こし」の原理に基づき、意味論、統辞論を含む修辞論の立場から実行される[1]

[2][3]。

スベルベル・ウイルソンによると、聞き手は話し手の発話明らかに偽である時、不条理なとき、有意性を欠いた発言をしているとき「呼び起こし」が生じ、再解釈を始めると分析している。我々は従来のグライスの会話の作法を用いずこの「呼び起こし」原理に基づき、計算機上で隠喩の検出を行う。

全体の流れとしては上の三つの場合を順次調べることになる。上のどれかに当てはまると隠喩の可能性のあるものと言うことになり、どれにも当てはまらない場合には、このシステムにとっては新知識と言うことになる。

6. 検出のユニットその一(明らかに偽の場合)

6. 1. 包含関係を用いるもの

例として、

「人間は狼である」

という隠喩を検出する手順を説明する。まず入力として以下のようなstate of affairを仮定する。

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

これを意味ネットワーク部のネットワーク部でmanというノードがwolfというノードの下(ノード間に別のノードがいくつかあってもよい)にあるかどうかで隠喩あるいは虚偽の発話のどちらかであることが検出される。具体的にはman, wolfがプロログのアトムとして、包含関係をクローズとして、

be (man, animal).

be (wolf, animal).

という形で保持されていてこれらのクローズに対して

?- be (man, wolf).

という質問をすることによって隠喩的な発話かどうかを判定する。この発話が虚偽の発話かどうかは、隠喩解釈を待たなければならない。この例および他のいくつかの例は計算機上で実装済みである。

6. 2. 属性関係を用いるもの

例として

「貿易摩擦」

の例を挙げる。これは次のように構文解析された入力形をとって入ってくる。

in s: at l: rub, trade; yes

これをtradeというノードの属性にrubがあるかどうかを見に行き、そういう属性がなければtradeの上位のノードの属性を見に行く、こうして一番上のノードまで見に行き、そのような属性があれば普通の発話、そうでなければ隠喩あるいは虚偽の発話として検出される。

7. 検出のユニットその二(不条理な場合)

ここでは

「彼は少し酔っている」

という例を挙げて抽出の手順を説明する。

現実の発話は次のような一連の文の後に述べられたものとする。

juwaは酒を飲み過ぎた。  
juwaは今暴れている。  
それを見ていたshiroは、  
「juwaはずこし酔っている」  
と言った。

これらの一連の文から次のような(1) - (4)の日本語文になることを仮定する。文の時制と場所の副詞から時空位置を設定し、発話の主語や指示対象を設定することが必要とされる。

「juwaはたくさん酒を飲んだ」(1)  
「そしてjuwaは今暴れている」(2)  
「shiroは(1)と(2)を認識した」(3)  
「shiroは”juwaは少し酔っている”と言った」(4)

これらの文は実際の入力として次のような状況意味論の実在的状況になることが要求される。

```
in s1: at l1: drinking, juwa, much; yes
in s2: at l2: going_wild, juwa; yes
in s3: at l1: recognize, shiro, s1; yes
in s4: at l2: recognize, shiro, s2; yes
in s5: at l2: saying, shiro, Z; yes
in s6: at l2: gets_drunk, juwa, a_little; yes
(11<12)
```

のような形をとる(“<”はl1よりもl2のほうが時間的に後であることを示す)。計算機内のデータ構造としては、

```
?- do(1, drink, juwa, much).
?- do(2, going_wild, juwa).
?- do(1, recognize, shiro, s1).
?- do(2, recognize, shiro, s2).
?- do(2, saying, shiro, z).
?- do(2, gets_drunk, juwa, a_little, M).
```

という形を持っている。ここで最初のatom 1, 2は相対時刻を示し、1<2なので1が時間的に前であることが表されている。また最後のクローズのMは隠喩的なものかどうかを返すための返り値のはいる変数である。

人間であるjuwa, shiroは「酒を飲む」「酔う」「認識する」「発話する」などの属性をとることが可能であるから範疇誤りはおかしていない。これは次のように計算機上で表されている。

```
attr(man, drink, liquor).
attr(man, gets_drunk).
attr(man, recognize, X).
attr(man, say, X).
```

よってこの一連の入力はユニット一を通過し、このユニットにはいってくる。

構造部の側は

```
in s1: at l1: drinking, A, much; yes
in s2: at l2: going_wild, A; yes
in s3: at l1: recognize, B, s1; yes
in s4: at l2: recognize, B, s2; yes
in s5: at l2: saying, B, Z; yes
in s6: at l2: gets_drunk, A, much; yes
(11<12)
```

と言う抽象的状況をクローズの形で持っており、入力の実在的状況が入ってくるたびに抽象的状況の指数が一つ進む。最後のs6と言う状況に入ってくるとa\_littleと

muchが一致していないことが分かり。そこで改めてa\_littleとmuchというノードを見てみるとこの二つのノードの間には対立リンクが張ってあるので隠喩あるいは虚偽の発話であることが検出される。これらの状況のうち最初の二つが構造部の中に、

```
do(L1, drink, A, much).
do(L2, going_wild, A).
```

のように実際に書かれており、後の四つは自動的に生成される。

8. 抽出のユニットその三(有意性を欠く場合)

8. 1. 常識のレベルを定めるもの

ここでは常識のレベルを定めるものとして専門用語だけを取り上げる。専門用語のノードには専門用語であることが、

```
com(man, natural).
com(animal, naturals).
```

のように記述してあり、隠喩的な表現ではないことが判定される。

ここでは普通の隠喩の例として、

「人間は動物である」

という例を挙げる。入力形は、

```
in s: at l: is, man, animal; yes
```

という形をとる。ユニットその一では範疇誤りをおかしていないので通過し、ユニットその二ではこの例のような抽象的状況が存在しないので通過する。最後にこのユニットに入ってきてman, animalが専門用語でないことを確認して隠喩的な発話と判定される。

8. 2. 諺に関するもの

焦点を各単語に定め、各単語によって呼び起こされる文脈、状況に適した諺を検索する。

例として、

「隣の花は赤い」

という発話を挙げる。実際の入力は、

```
in s: at l: having, neighbor, flower; yes
: is, flower, red; yes
```

が入ってくる。ユニット一、二は通過する。「隣」「花」「赤い」という語は専門用語ではないので常識のレベルにかからず、このユニットに入ってくる。ここで「隣」「花」「赤い」によって呼び起こされる諺を検出し、隠喩理解ルーチンに引き渡す。

9. 結論

これらの隠喩抽出ユニットでは隠喩である可能性のあるものをすべて検出する。隠喩でないものも検出されてしまうが、これは隠喩理解ユニットにまかされることになる。今後の研究は、明らかに偽である場合で虚偽の発話かどうかの判定をどうするか、不条理な場合で抽象的状況の記述をより抽象化する方法をどうするか、諺の検出でもう少し絞り込む方法はないか、などを研究していく予定である。

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] スペルベル, 菅野 盾樹訳 (1979), 象徴表現とは何か, 勁草書房