電子文書参照支援システム BENLI における内容検索機能
—— 意味構造の一致度の判定手法 ——

3F−9

和田 浩一，石橋 由希夫，小川 洋一
セコム(株)IS 研究所

1 はじめに

我々は，社内向けの技術文書を対象として，その参照を容易にするための，BENLI というシステムの開発を行なっている [1]．そのシステムにおいて，実現を目指している機能の1つが内容検索機能である．この機能では，検索キーとして日本語の文を使う．我々はこれをキーセンテンスと呼んでいる．BENLI の内容検索部は検索対象の文書から，ユーザーが入力したキーセンテンスと意味的に類似した部分を検索する．本稿では，キーセンテンスと検索対象の文との類似度をどのようにして判定しているかについて述べる．

2 部分一致による類似度の判定

キーセンテンスと検索対象の文は，それぞれ構文解析されて，我々が意味構造と呼んでいるデータに変換された上で比較される．意味構造は，文中の自立語の修飾関係を表した有向グラフである．“音声でガイダンスするコントローラ”という文の意味構造の例を図1に示す．図1の意味構造

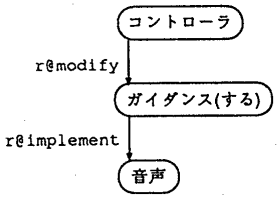

図 1: 意味構造の例

のグラフを構成するノードはさらに，内部構造を持っており，それも同様にグラフの形で表したものを比較に用いる．このようにして詳細化した意味構造のうち，表記の情報だけを表した例を図2に示す．類似度の判定は，この詳細化した

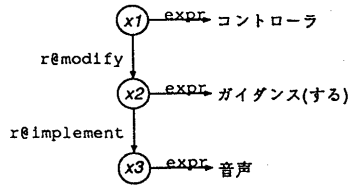

図 2: 詳細化した意味構造の例

た意味構造どうしを比較した時に，どのくらいの部分的な一致 [2] が得られるかを指標として判定している．これには，意味構造を構成するノードどうし，アークどうしに，1対1の対応をつけることによって共通の部分構造を得て，その部分構造を構成するアークの評価値の総和を類似度の評価値としている (図3)．

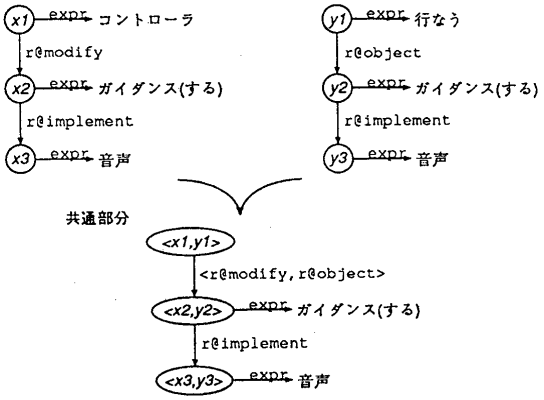

図 3: 部分一致

一致する部分構造は複数個存在するので，探索を行なって得られた解のうち，最も評価値の大きいものを採用する．

解の探索は次のように行なっている．まず，対応可能なアークの対の組合せをすべて作る (図4)．図中では2つのアークの対を p_1, p_2 と表している．

こうしてできたアーク対の集合 $P = \{p_1, p_2, p_3 \cdots, p_n\}$ の部分集合で，アークの対応および，ノードの対応が矛盾なく1対1になっているものを探索する．例えば，図4中の p_1 では $<x_1,y_1>$ という対応付けがあるのに対して，p_2 では $<x_1,y_2>$ という対応付けがなされている．この場合，p_1 と p_2 はノードの対応が1対1でなく両立しない．いいかえると，p_1 と p_2 を同時に含む部分集合は対応に矛盾があり，解とならない．したがって，探索の際に p_1 が選択されたとすると，p_2 は選択されない．以下，探索アルゴリズムを示す．

1. S=P とする．

$$S = \{p_1, p_2, p_3, \ldots, p_n\}$$
$$P = \{p_1, p_2, p_3, \ldots, p_n\}$$

A method to determine the similarity of semantic structures for content search in the BENLI system
Koichi WADA, Yukio ISHIBASHI, Youichi OGAWA
SECOM IS Laboratory

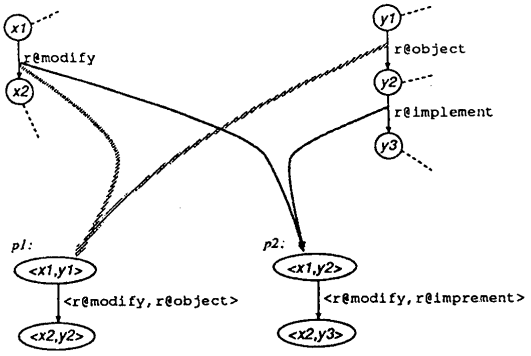

図 4: アークの対

2. まず，S から任意の要素 (例として p_1) を選び，それを S 中では削除，P 中では ○ による印付けを行なう．また，選んだ要素と両立できないものを，P から削除する．(例として，p_1 と p_2 は両立しないとする．)

$$S = \{\textcircled{1}, p_2, p_3, ..., p_n\}$$
$$P = \{\textcircled{1}, \not{p}_2, p_3, ..., p_n\}$$

3. P から印のついてない要素を選んで，○ による印付けを行ない，S 中から削除する．同様に，選んだ要素と両立できないものを，P から削除する．(例として，p_3 と p_n は両立しないとする．)

$$S = \{\textcircled{1}, p_2, \not{p}_3, ..., p_n\}$$
$$P = \{\textcircled{1}, \not{p}_2, \textcircled{3}, ..., \not{p}_n\}$$

4. P に，無印のものがなくなるまで 3. を繰り返す.
5. P で，○ のついたものを 1 つの解として出力する.
6. P を元に戻す.

$$S = \{\not{p}_1, p_2, \not{p}_3, ..., p_n\}$$
$$P = \{p_1, p_2, p_3, ..., p_n\}$$

7. S のうち，削除されてないものから要素を選び (例では p_2)，2.〜6. を行なう．.
8. S が空になるまで 7. を繰り返す.

このアルゴリズムは，新たな解の探索を始める時に，以前に得られたどの解の中にも入ってない要素から探索を開始することによって，探索空間を減らそうというものである.

3 書き換え規則による制御

ある述語とある句が同じ意味を持っていても，語の構成が違うため，それらの意味構造を直接比較するだけでは，満足な結果は期待できない．そこで，類似表現に関する知識を，意味構造の書き換え規則という形で用意して，類似度判定の制御を行なう．これは，特定の意味構造の間の類似度の評価値を大きくすることを狙ったものである.

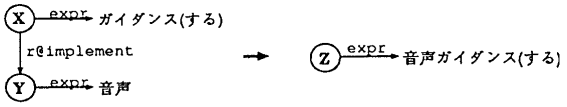

図 5: 書き換え規則

書き換え規則の例を図 5に示す．また，書き換えの対象となった部分と，対象とならなかった部分の接続関係を記述するために，記号を付け換える規則を同時に用意する (図 6).

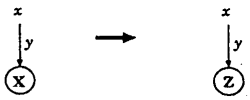

図 6: 接続関係の規則

これらの規則は，実際は左辺を条件とするプロダクションルールとして解釈され，生成された右辺の構造が単に付け加えられる．右辺の構造中のノードの記号 (図 5の Z) には，それまで使われていない新しい記号を使うこととする.

図 1の意味構造にこの規則が適用されるようすを，図 7に示す．このように，結果は書き換える前と書き換えた後の

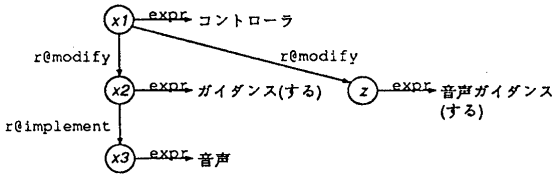

図 7: 規則の適用

構造が混在した形で保持されるため，両立しない記号に関する情報を別に持ち，解釈するときに用いる．この場合，$\{x_2, x_3\}$ と $\{z\}$ は，両立しない．書き換え規則によって生成されるノードの記号 (例では z) は，常に新しい記号なので，混乱は生じない．また，連続する書き換えの過程は，停止性が保証されないので，書き換えの段数で制御を行なう.

4 おわりに

BENLI の内容検索機能における意味的類似度の判定手法について大まかに述べた．今後は検索システムの評価のために，書き換え規則の拡充を進めていく.

参考文献

[1] 小川他．“電子文書参照支援システム BENLI − システムの構成と機能の概要 −”，第 45 回情処全大 3F-07(1992-10)

[2] 原口．“類推の機械化について” 知識の学習メカニズム,4 章,pp 125-154. 共立出版株式会社 (1987).