国語辞典からの類義表現抽出と SYNGRAPHデータ構造による柔軟マッチング

大西 貴士　　黒橋 禎夫
東京大学大学院 情報理工学系研究科

{oonishi,kuro}@kc.t.u-tokyo.ac.jp

1　はじめに

自然言語は自由度が高いため，同じ内容を表現するにしても様々な表現を使用することができる．そのような様々な表現のずれをいかにして吸収するかが自然言語処理における重要な課題である．

例えば，用例ベースの機械翻訳を行うときに，入力文として「ホテルに一番近い駅はどこですか」が与えられたとする．このとき，用例に「旅館の最寄りの駅はどこですか ↔ Where's the nearest station from the hotel?」があったとしても，単純な完全マッチングを行うだけではこの用例を翻訳に使うことはできない．こういった問題は，機械翻訳だけでなく，情報検索や質問応答など様々なタスクでみられる．

このような問題を克服するには，表現のずれを吸収する柔軟なマッチングが必要であり，そのためには，同義・類義表現の知識の獲得と，それらを柔軟に統合して利用する枠組みの 2 つが必要となる．

本論文では，これらの 2 つの問題を次のように解決し，柔軟なマッチングを実現する手法を提案する．

1. 国語辞典から自動で同義関係や上位下位関係の知識を獲得する．
2. 表現のずれをパックした SYNGRAPH というデータ構造を導入することにより，表現のずれの組合わせを効率的に扱う．

1 については，国語辞典を用いることで，常識的，基本的な同義・類義表現を網羅的に自動獲得する．これにより,「夕食」と「食事」のような上位関係,「旅館」と「ホテル」のような単純な同義語に加えて,「一番」と「もっとも」などの副詞の同義語や,「最寄り」と「一番近い」のような語と句の同義関係など，広い範囲の類義関係を扱うことができる．

2 については，類義関係を事前にすべて展開することは組合せ爆発となってしまい，ダイナミックに検索・マッチングを行う手法では計算量が大きすぎる．そこで，同義関係に ID を与え，表現のずれをパックした SYNGRAPH というデータ構造で文を表現する．

2　類義表現データベース

柔軟なマッチングの知識源である類義関係は，国語辞典の見出し語と定義文の関係から，以下のような方法によって自動的に獲得する．

2.1　定義文からの同義・上位下位関係の抽出

定義文から取り出すことができる最も基本的な情報として，定義文の主辞が見出し語の上位語となることがあげられる．日本語の場合，文末の語が文の主辞となるので，次の例のような上位語が抽出できる（以下,「：」の左が見出し語，右が定義文，太線が抽出される語）．

夕食 ： 夕方の**食事**。
重心 ： 重さがつりあって中心となる**点**。

しかし，場合によっては文の主辞以外の語が上位語であったり，同義語，下位語が示される場合もある．これらは次のような文末パターンによって判別することができる（下線部がパターン）．また，定義文が 3 文節以下の長さであれば，定義文全体を同義句として抽出する．

上位語
　土星 ： **わく星** の一つ。
　とび ： **タカ** の一種。
同義語
　アイス ： 「**アイスクリーム**」の略。
　もっとも ： **一番**。（※ 定義文が一語の場合）
下位語
　硬筆 ： 筆に対して、**えんぴつ**や、**ペン・ボールペン** などのこと。
同義句
　夕食 ： **夕方の食事**。
　最寄り ： **一番近い**ところ[1]。

[1]「こと」「ところ」などの抽象度の高い語は削除する．

2.2 類義表現データベースの構築

このように取り出される関係をまとめることにより，類義関係のデータベースを得ることができる．ここでは多義語の扱いに注意する必要がある．国語辞典で複数の項目が与えられている場合，その語は多義語であると考える．

点：1. 小さなしるし。ぽち。
　　2. 場所。位置。

例えば，A = B，B = C という関係がある場合，B が多義語でなければ A = B = C という関係にまとめることができる．しかし，B が多義語の場合には，意味の異なる B を通して不適切なまとまりを作る場合があるので，組合わせ処理は行わない．同様に上位下位関係においても，A → B，B → C や A → B，C → B のときに B が多義語であればこれらの組合わせ処理は行わない．

このように，多義語の場合を除いて関係の組合わせ処理を行うと，まず，同義語，同義句関係をまとめた同義グループができる．各同義グループに ID を与え，以下ではこれを SYNID と呼ぶ．さらに，それらの間の上位下位関係のネットワークが構築される．以降，これを類義表現データベースと呼ぶ．

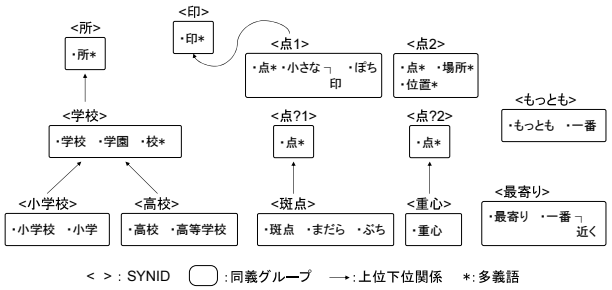

図 1: 類義表現データベースの例

図 1 に構築される類義表現データベースの例を示す．この中で，「点」が多義であるので，＜斑点＞→＜点＞と＜重心＞→＜点＞の関係をまとめることは行わない．マッチング対象文中に「点」という表現があれば，各同義グループが別々に与えられることになる．国語辞典の定義文中の語の多義性解消や，マッチングを行う文中の語の多義性解消は今後の課題である．

3 SYNGRAPH

3.1 SYNGRAPH のデータ構造

ある文について，前節で抽出した類義表現を組合わせ的に使えば，様々な，類義の文を作り出すことができる．

ホテルに一番近い駅はどこですか
＝ ホテルにもっとも近い駅はどこですか
＝ ホテルの最寄りの駅はどこですか
＝ 旅館に一番近い駅はどこですか
　 ホテルに近い駅はどこですか
　 ...

柔軟なマッチングで必要なことはこれらの類義関係を認識することであるが，これを事前に展開しておく方法は組合わせ爆発を起こしてしまい，ダイナミックに検索する方法では計算量が爆発する．そこで，文のあらゆる類義表現をパックした SYNGRAPH というデータ構造を考え，これを用いてマッチングを行うことによって柔軟なマッチングを実現する．上記の例を SYNGRAPH で表現したものを図 2 に示す．

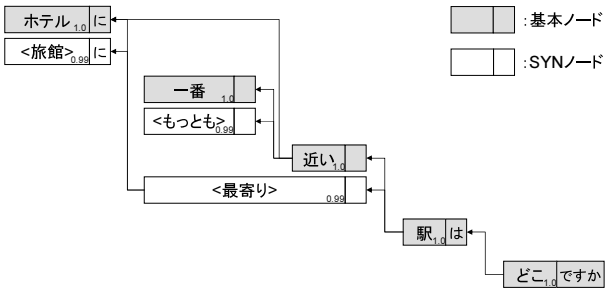

図 2: SYNGRAPH の例

SYNGRAPH のベースとなるのは，もとの文の依存構造木であり，そのノードは 1 つの自立語と 0 個以上の付属語からなるもので，以下基本ノードと呼ぶ．そして，基本ノードの自立語に同義グループがあれば，その SYNID を別のノードとして与える（これを SYN ノードと呼ぶ）．

さらに，複数のノードに対応する表現に同義グループがあれば，その SYNID のノードも加える．図 2 の＜最寄り＞がこのような SYN ノードである．そしてさらに，各 SYN ノードに対して，類義表現データベースにおいて上位の同義グループがあれば，その SYN ノードを加える[2]．

各ノードには，もとの文の表現からのずれに応じたスコア，NS(Node Score) を与える．NS の計算方法は 4.1 節で説明する．

3.2 SYNGRAPH マッチング

2 つの SYNGRAPH は，もとの文を過不足なくカバーする同一のノード群が，同一の係り受け関係をもつ場合にマッチすると考える（図 3）．これを SYNGRAPH 全体マッチと呼ぶ．そして，$SYNGRAPH_1$ の部分木と別の $SYNGRAPH_2$ が SYNGRAPH 全体マッ

[2]これによって，上位下位 1 段階の表現のずれがマッチする．

チしていることを，SYNGRAPH$_1$ に SYNGRAPH$_2$ が SYNGRAPH マッチしていると呼ぶ．

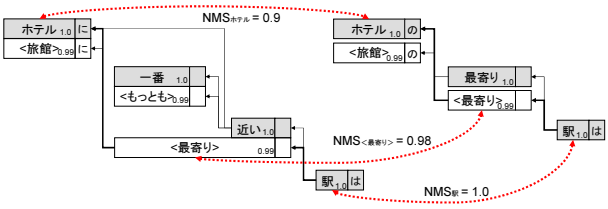

図 3: SYNGRAPH 全体マッチ

マッチする SYNGRAPH について，マッチのスコアを定義する．まず，マッチしているノード間のスコア，NMS(Node Match Score) を定義する．これは，マッチする SYN ノードのスコア，NS$_1$，NS$_2$ をもとに次のように計算する．

$$NMS = NS_1 \times NS_2 \times \text{付属語不一致ペナルティ}$$

ここで，付属語不一致ペナルティは付属語に不一致があれば 0.9，そうでなければ 1 とする．

次に，SYNGRAPH マッチのスコア，SMS(SYNGRAPH Match Score) を定義する．SMS は，それぞれの NMS を部分木側の基本ノード数で重みづけ平均したもので，SYNGRAPH$_1$ に SYNGRAPH$_2$ が SYNGRAPH マッチしているときのスコア，SMS$_{12}$ は以下のように定義する．

$$SMS_{12} = \frac{\sum(NMS \times \text{基本ノード数}_1)}{\sum \text{基本ノード数}_1}$$

図 3 の例では，＜ホテル＞ノードの NMS は付属語の不一致があるので 0.9，＜最寄り＞ノードの NMS は 0.98,「駅」ノードの NMS は 1.0 となり．SMS は，$SMS_{\text{左右}} = \frac{0.9\times1+0.98\times2+1.0\times1}{1+2+1} = 0.965$，$SMS_{\text{右左}} = \frac{0.9\times1+0.98\times1+1.0\times1}{1+1+1} = 0.96$ となる．

2 つの SYNGRAPH がマッチするかどうかは，それぞれの SYNGRAPH の末尾から，マッチするノードの係り受け関係をそれぞれたどっていくことで調べられる．また，2 つの SYNGRAPH のマッチには，複数のノードの対応付けが考えられる場合があるが（図 3 で上位の＜旅館＞ノードを対応付ける場合など），その中から最も SMS が大きい対応付けを選択する．

4 SYNGRAPH による柔軟マッチング

SYNGRAPH を使った柔軟マッチングによる用例検索や情報検索は以下の手順で行う．

1. 類義表現データベースを SYNGRAPH に変換し，類義表現データベース内の相互関係を明確化する．
2. SYNGRAPH 化した類義表現データベースと検索対象との SYNGRAPH マッチングによって検索対象を SYNGRAPH に変換する．
3. 入力を SYNGRAPH に変換する．そして，機械翻訳の用例検索においては，さらに入力と検索対象（用例集合）の SYNGRAPH マッチングを行うことによって入力と部分一致する用例を検出する．情報検索においては，入力と検索対象（文書集合）の SYNGRAPH の類似度計算に基づいて文書のランキングを行う．

4.1 類義表現データベースの SYNGRAPH 化

まず，類義表現データベース中の各同義グループの各表現を依存構造木にする．これがもっとも単純な SYNGRAPH であり，各基本ノードのスコア (NS) は 1.0 とする．

次に，各表現の一部分と，他の表現の全体がマッチするかどうかを SYNGRAPH マッチングによって調べる．マッチする場合には，部分的にマッチした方のノード集合の部分に，全体的にマッチした表現の SYNID のノードを付与する．また，そこに上位の同義グループがあればその SYN ノードも付与する．これを繰り返し，どの同義表現にもそれ以上新たに SYN ノードを付与できなくなるまで繰り返す．

新しく付与される SYN ノードの NS は，SYNGRAPH マッチのスコア，SMS に，関係に応じたペナルティ（同義関係は 0.99，上位関係は 0.7）をかけたものとする．先の例では＜水中＞ノードの NS は 0.99，＜中＞ノードの NS は 0.7 となる．

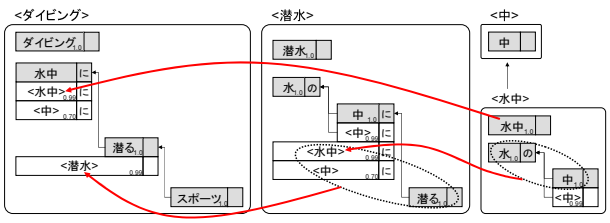

図 4: 類義表現データベースの SYNGRAPH 化

図 4 の例では，例えば＜潜水＞の「水の中に潜る」の一部分「水の中」が，＜水中＞の「水の中」にマッチするので，ここに＜水中＞ノード，さらにその上位の＜中＞ノードが与えられる．そして，2 回目の繰り返しで，＜ダイビング＞の「水中に潜る」の部分に＜潜水＞ノードが与えられる．

4.2 検索対象の SYNGRAPH 化

SYNGRAPH 化した類義表現データベースを参照して，検索対象の各文の SYNGRAPH 化を行う．

この処理は前節の類義表現データベースの SYNGRAPH 化とほぼ同じである．すなわち，まず検索対象文を基本的な SYNGRAPH に変換する．次にこの SYNGRAPH のあらゆる部分と，類義表現データベースの各表現の SYNGRAPH の全体とのマッチングを行い，マッチすればその SYN ノード（さらに上位の同義グループがあればその SYN ノード）を付与していく．

4.3 機械翻訳における用例検索

用例ベースの機械翻訳では，与えられた入力文とマッチする用例を組合わせることで翻訳を行う．このとき，最初の例でも示したように，同義・類義表現の用例を用いるために柔軟マッチングが必要となる．

これは，まず入力文を SYNGRAPH 化し，次に入力文の SYNGRAPH と検索対象（用例集合）の SYNGRAPH でマッチするものを探し出す処理となる．つまり，入力文と用例で SYNGRAPH マッチングすればよい．これは，前節で示した検索対象の SYNGRAPH 化は同一の処理である．

5 実験

SYNGRAPH を用いたマッチングの有効性を示すために機械翻訳と情報検索の 2 つのタスクで実験を行った．

実験に使用した類義表現データベースは，例解小学国語辞典（見出し語，約 3 万語）から自動構築したもので，表 1 に示すように大規模な類義表現データベースとなっている．

表 1: 自動構築した類義表現データベース

同義グループ数	17,393
1 同義グループ内の平均表現数	1.8
同義グループ間の上位下位関係数	11,446

機械翻訳の評価型ワークショップである IWSLT'05 の日英翻訳タスクを用いて，著者らによる用例翻訳システムをベースとして実験を行った [1, 2].

表 2: IWSLT 結果

類義表現 DB	NIST	BLEU
同義×上下×	8.630	0.422
同義○上下×	8.734	0.428
同義○上下○	8.689	0.425

実験結果を表 2 に示す．同義×上下×は類義表現データベースを使用しないベースラインの結果である．これに対して同義○上下×は同義関係のみを使用する場合で，この場合が最もスコアが高く，相対スコアで NIST で 1.2%，BLEU で 1.4%の，統計的に有意な向上がみられた ($p < 0.05$)．

同義○上下○，すなわち上位下位関係も利用した場合は，同義○上下×に比べてスコアが下がった．これは，翻訳においては上位下位関係の用例の訳を利用することは必ずしも妥当ではないことを示している．例えば,「デパート」に対してその上位語「店」の翻訳 “store” が使われるなどして，スコア低下の原因となっていた．

情報検索の評価型ワークショップである IREX[3] のデータセットを用いて実験を行ったところ（表 3），同義関係に加えて上位下位関係を利用する場合が最もスコアが高く，ベースラインと比べて相対スコアで 8.4% の統計的有意な向上がみられた ($p < 0.05$)．

表 3: IREX 結果

手法	類義表現 DB	R-precision
BM25	同義×上下×	0.474
提案手法	同義○上下×	0.509
	同義○上下○	0.514

6 おわりに

本論文では，自然言語の重要な特徴である表現のずれを取り扱うために，柔軟なマッチングを実現する方法を提案した．柔軟マッチングのためには，知識源（網羅性のある類義表現）と枠組みの問題がある．前者については国語辞典からの自動抽出を行い，後者については SYNGRAPH というデータ構造の導入を行った．機械翻訳における用例検索と，情報検索における類似文書ランキングにこの手法を適用し，有効性を確認した．

今後の課題として，辞書からの関係抽出への語の曖昧性解消の導入や，構文的なずれのある類義表現の取り扱いなどがある．

参考文献

[1] Sadao Kurohashi, Toshiaki Nakazawa, Kauffmann Alexis, and Daisuke Kawahara. Example-based machine translation pursuing fully structural nlp. In *International Workshop on Spoken Language Translation (IWSLT'05)*, pp. 207–212, 2005.

[2] IWSLT. http://www.is.cs.cmu.edu/iwslt2005/, 2005.

[3] IREX. http://nlp.cs.nyu.edu/irex/, 1999.