SYNGRAPHデータ構造における述語項構造の柔軟マッチング

小谷 通隆 † 中澤 敏明 †† 柴田 知秀 †† 黒橋 禎夫 †††

† 京都大学 工学部電気電子工学科 †† 東京大学大学院 情報理工学系研究科
††† 京都大学大学院 情報学研究科
odani@nlp.kuee.kyoto-u.ac.jp, nakazawa@kc.t.u-tokyo.ac.jp,
shibata@nlp.kuee.kyoto-u.ac.jp, kuro@i.kyoto-u.ac.jp

1 序論

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類義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柔軟に吸収するかは自然言語処理における重要な課題である。

この問題を解決するために、大西らはSYNGRAPHデータ構造による柔軟マッチングを提案している [1]。SYNGRAPHデータ構造とは、1個の自立語と0個以上の付属語からなる基本句を単位とした木構造をベースとしており、そこに国語辞典から抽出した類義表現をパックしたものである。文をSYNGRAPHで扱うことで類義表現を効率的に扱うとともに、文同士ではなく、そのSYNGRAPH同士のマッチングを行い、柔軟なマッチングを実現している。しかし、大西らの手法は、付属語を無視して各基本句を自立語のみで近似しており、「褒められる」と「褒める」が完全に一致してしまうなどの問題点があった。

そこで、本研究では基本句に付与される文法素性に注目し、述語項構造の柔軟なマッチングの実現を目指す。まず、文の基本句の文法素性を抽出し、SYNGRAPHにその情報を付与する。文法素性は付属語によって表現されていることが多く、また同じ文法素性でも異なる付属語によって表現されていることがある。そこで、文法素性を抽出することによって同じ文法素性における類義表現を同じように扱い、そのずれを吸収する。また、項（格要素）の述語に対する格を抽出し、SYNGRAPHにその情報を付与する。文法素性と格の情報を利用して、述語項構造単位での柔軟マッチングも可能とする。また、本研究では反義語の否定といった類義表現を扱うために、大西らの利用した語の同義関係と上位下位関係に加えて、語の反義関係も利用する。

2 形態素解析・構文解析

大西らは、形態素解析器 JUMAN[3]、構文解析器 KNP[2] を用いて文の解析を行い、文を SYNGRAPH 化している。本研究では、大西らが用いていない情報を SYNGRAPH にとりいれる。

2.1 基本句への文法素性の付与

本研究では、否定、尊敬、可能、受身、使役の5表現を扱う。本研究で扱っている5つの文法素性がどのように付与されるかを述べる。

否定表現

否定を表す接辞「不」「非」「ない」や機能表現「〜し兼ねる」があれば、KNP は [否定表現][1]という素性を基本句に付与する。例えば以下のような基本句には、[否定表現] という素性が付与される。

「不 信任」、「信任し ない」、「信任し 兼ねる」

尊敬表現

尊敬動詞は、その動詞が尊敬動詞であることおよびその原動詞が JUMAN の辞書に登録されている。例えば、「おっしゃる」は JUMAN によって「言う」の尊敬動詞であることがわかる。この情報によって、KNP は [尊敬表現] という素性を基本句に付与する。また、尊敬の接辞「れる」「られる」や機能表現「お〜になる」があれば、KNP は [尊敬表現] という素性を基本句に付与する。例えば以下のような基本句には、[尊敬表現] という素性が付与される。

「おっしゃる」、「帰ら れる」、「お 帰り になる」

[1] 本論文では文法素性を [ ] で表す。

可能表現

可能を表す接辞「る」「られる」や機能表現「〜することができる」があれば、KNP は [可能表現] という素性を基本句に付与する。例えば以下のような基本句には、[可能表現] という素性が付与される。

「会え る」、「食べ られる」、「食べる ことができる」

受身表現

受身を表す接辞「れる」「られる」があれば、KNP は [受身表現] という素性を基本句に付与する。例えば以下のような基本句には、[受身表現] という素性が付与される。

「怒ら れる」、「食べ られる」

使役表現

使役を表す接辞「せる」「させる」があれば、KNP は [使役表現] という素性を基本句に付与する。例えば以下のような基本句には、[使役表現] という素性が付与される。

「怒ら せる」、「食べ させる」

2.2 基本句の格情報の付与

KNP は、格フレームを用いて用言の格解析を行っており、基本句に格情報を付与する [4]。以下の例では、どちらも「彼」に「ガ格」が付与される。

「彼が 行く」、「彼は 行く」

3 SYNGRAPH

3.1 類義表現データベース

SYNGRAPH の説明をする前に、文を SYNGRAPH 化するのに必要な類義表現データベースについて説明する。類義表現データベースは国語辞典から獲得した以下の 3 つの情報からなる。

同義グループ

同義関係にある語や句を獲得し、同義グループとしてまとめる。同義グループには ID を与え、これを **SYNID** と呼ぶ。本論文では、SYNID を同義グループの一表現を＜ ＞で囲ったもので表現する。

＜異国＞ {「異国」、「海外」、「外国」…}
＜水中＞ {「水中」、「水の中」}
＜最寄り＞ {「最寄り」、「一番近い」}

同義グループ間の上位下位関係

語の上位下位関係を獲得し、同義グループ間に上位下位関係を付与する。

＜地震＞ → ＜災害＞ → ＜災難＞

同義グループ間の反義関係

本研究では、語の反義関係も獲得し、同義グループ間に反義関係も付与する。

＜高い＞ ↔ ＜低い＞

3.2 SYNGRAPH データ構造

次に、SYNGRAPH データ構造を説明する。SYNGRAPH のベースとなるのは、1 個の自立語と 0 個以上の付属語からなる基本句を単位とした木構造である。SYNGRAPH はこの基本句を基本のノードとし、これを**基本ノード**と呼ぶ。基本ノードは、基本句の自立語（本論文ではこれを**基本 ID** とよぶ）と文法素性を表す**文法フラグ**を持つ。また、各ノードは自分の子供のノードへのポインタをもち、それによってノード間の依存構造を表す。さらに基本句に付与された格情報をノードに付与し、親のノードとの格関係を表す。

類義表現データベースにおいて、基本 ID がある同義グループに属していれば、その SYNID を新しいノードとして付加する（図 1 の＜過去＞）。これを **SYN ノード**と呼ぶ。さらに、各 SYNID の上位、反義関係の SYNID があれば、その SYNID を新しい SYN ノードとして付与していき、様々な表現をパックしていく（図 1 の＜時＞）。上位、反義関係から作られるノードにはそれを示す**関係フラグ**がたててあり、同義関係で作られるノードと区別する。例えば、「負ける」には＜勝つ＞の反義語であることを示すノードが付与される。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 を新しい SYN ノードとして付与する（図 1 の＜親友＞）。SYN ノードも文法フラグ、子供のノードへのポインタ、格情報の要素をもち、それらは基本ノードのものを受け継ぐ。

また、各ノードには、基本ノードとの類似度に応じたスコアを与える。ただし、基本ノードのスコアは 1.0 とする。

4 述語項構造の柔軟マッチング

最後に、SYNGRAPH によって述語項構造の柔軟なマッチングを実現する手順を述べる。2 文を SYNGRAPH 化して、SYNGRAPH マッチングを行うこ

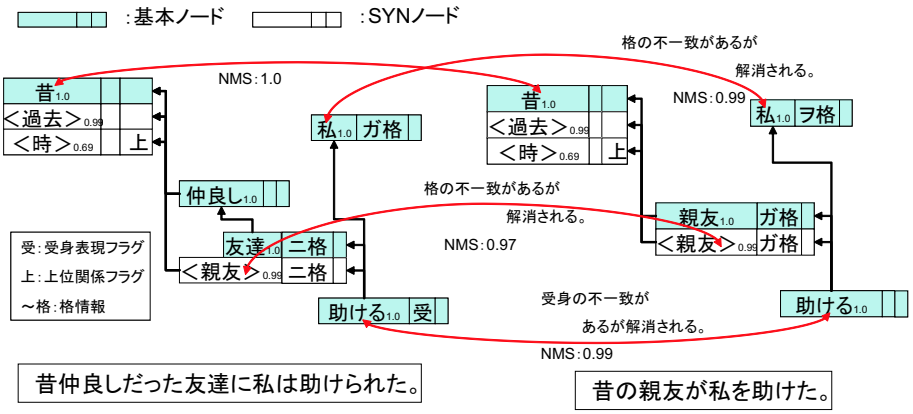

図 1: SYNGRAPH マッチングの例

とで、類似度を計算する。図 1 は「昔仲良しだった友達に私は助けられた」と「昔の親友が私を助けた」の SYNGRAPH マッチングである。

SYNGRAPH マッチングは以下の手順で行なう。

ステップ 1 SYNGRAPH の近似マッチングを行う。
ステップ 2 述語項構造単位で表現のずれの解消する。
ステップ 3 SYNGRAPH の類似度の計算を行う。

4.1 SYNGRAPH の近似マッチング

近似マッチングでは、ノードが同じ基本 ID または SYNID をもっていればをマッチしていると考える。すなわち、近似マッチングでは文法フラグなどの要素の違いは無視してマッチングを行う。

マッチしたノードの対応にはスコアが与えられ、マッチしたノードのスコアのかけ算によって定義される。また、それぞれの対応において文法フラグや、関係フラグの不一致があればそれを記録しておく。ここで、否定表現の文法フラグと反義の関係フラグについては両者は同義であると考え、対応においてその総数が奇数であったときに、否定表現の不一致があったと記録する。SYNGRAPH がマッチするかどうかは、それぞれの SYNGRAPH のヘッドから、マッチしたノードの係り受け関係を順に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づけが考えられる場合（図 1 で＜過去＞や＜時＞を対応づける場合など）があるが、その場合には対応のスコアが最も高いものを選択する。

4.2 述語項構造単位での表現のずれの解消

近似マッチしたことが分かっても、その対応するノード間には何らかの不一致が存在することがある。しかし、それらの中には述語項構造単位で比べたときには同じ内容を指しているようなものがある。例えば、「叩かれた」と「叩いた」のマッチングを考えると、どちらも「叩く」に近似されてマッチするが、受身表現の不一致がある。しかし、「彼に叩かれた」と「彼が叩いた」のマッチングを考えると、「叩く」の対応には受身の不一致があり、「彼」の対応にも格の不一致があるが、述語項構造単位で考えると同じ内容であることがわかる。本研究では、以下に述べる「受身表現による表現のずれ」「使役表現による表現のずれ」「偶数個の否定表現の不一致をもつ表現のずれ」「述語に否定表現の不一致をもつような表現のずれ」を扱っている。これらをチェックして、当てはまるときは対応の違いを解消する。

受身表現による表現のずれ

述語の対応に受身表現の不一致があり、その子供の対応に格の不一致があったとき、次の 3 条件のいずれかに該当するかを調べる。

- 受身表現の側で「ガ格」であるものがもう一方では「ヲ格」であれば、格の不一致と受身表現の不一致を解消する。以下に例を示す。

  「私が怒られた」＝「私を怒った」

- 受身表現の側で「ニ格」であるものがもう一方では「ガ格」であれば、格の不一致と受身表現の不一致を解消する。以下に例を示す。

  「彼に怒られた」＝「彼が怒った」

- 上記のような格の違いがどちらもあれば、双方の格の不一致と受身表現の不一致を解消する。以下に例を示す。

  「彼を彼女が叩いた」＝「彼が彼女に叩かれた」

使役表現による表現のずれ

述語の対応に使役表現の不一致があるとき、その子供の対応に格の不一致があり、使役表現の側で「ニ格」であるものがもう一方では「ガ格」であるならば、使役表現による表現のずれと認識し、それらの不一致を解消する。以下に例を示す。

「彼が向かった」＝「彼を向かわせた」

偶数個の否定表現の不一致をもつ表現のずれ

否定表現の不一致が存在する対応が述語項構造全体で偶数個あれば、全ての対応における否定表現の不一致を解消する。

以下の例では、「正しい」に＜正しい＞というSYNノードが付与され、「誤った」に＜正しい＞の反義であるというSYNノードが付与される。＜正しい＞と「言う」の対応において否定表現の不一致があり、述語項構造全体で偶数個の否定表現の不一致がある。よって、双方の否定表現の不一致を解消する。

「彼は正しいことを言う」
＝「彼は誤ったことを言わない」

述語に否定表現の不一致をもつような表現のずれ

述語の対応に否定表現の不一致があり、その子供の対応に格の不一致が２つあれば、それらの不一致をすべて解消する。以下に例を示す。

「彼は彼女に負けた」＝「彼に彼女は勝った」

「車はバイクより高い」＝「車よりバイクは高くない」

4.3 類似度計算

ステップ１、２の結果、マッチしたペアのマッチングスコア（NMS:Node Match Score）を計算する。ステップ１での対応のスコアを基本とし、その対応に何らかの不一致が存在するとき、そのタイプに応じたペナルティを掛けたものをNMSとする。例えば、「会う」と「会える」はマッチするが、可能表現のフラグの不一致があるために、そのスコアには可能表現の不一致に応じたペナルティをかける。また、ステップ２の中で不一致が解消されたときは、完全なマッチと区別するため、0.99のペナルティをかける。

SYNGRAPHマッチのスコア（SMS:Syngraph Match Score）は、NMSの平均とする。

$$SMS = \frac{\sum(NMS)}{\text{対応の数}} \quad (1)$$

図1の例では、2つのSYNGRAPHは近似マッチしており、さらに述語項構造単位でみたときに受身表現による表現のずれが存在することがわかる。このときのNMSは

$$NMS_{\text{昔}} = 1.0 \times 1.0 = 1.0$$
$$NMS_{\text{<親友>}} = 0.99 \times 0.99 \times 0.99 = 0.97$$
$$NMS_{\text{私}} = 1.0 \times 1.0 \times 0.99 = 0.99$$
$$NMS_{\text{助ける}} = 1.0 \times 1.0 \times 0.99 = 0.99$$

となり、このときのSMSは、

$$SMS = \frac{1.0 + 0.97 + 0.99 + 0.99}{4} = 0.99 \quad (2)$$

となる。

5 結論

本研究では、基本句の文法素性と用言の格解析結果を利用して、述語項構造における様々な表現のずれを吸収できる可能性を示した。今後の課題としては、まず本研究で提案した手法を情報検索や用例ベースの機械翻訳で大規模に実験し、本研究の有効性を示すことが挙げられる。また、「彼と彼女が行く」と「彼女と彼が行く」のような並列構造における表現のずれや、「太郎は怒ったので帰った」と「怒った太郎は帰った」のような連体修飾を用いた表現によるずれの柔軟なマッチングをいかに実現するかも今後の課題とする。

参考文献

[1] 大西貴士, 黒橋禎夫. 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. 言語処理学会 第 12 回年次大会 発表論文集, pp. 1127–1130, 2006.

[2] 黒橋禎夫. 日本語構文解析システム KNP version 2.0 b6 使用説明書. 京都大学大学院情報学研究科, 1998.

[3] 黒橋禎夫, 河原大輔. 日本語形態素解析システム JUMAN version 5.1 使用説明書. 京都大学大学院情報学研究科, 2005.

[4] 河原大輔, 黒橋禎夫. Web から獲得した大規模格フレームに基づく構文・格解析の統合的確率モデル. 言語処理学会 第 12 回年次大会 発表論文集, pp. 1111–1114, 2006.