様々な表現のずれを吸収するテキスト検索システム

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

テキスト検索システムを高精度化する上での最大の問題は，同義語や上位・下位語（これらを類義表現と呼ぶ）が引き起こす様々な表現のずれをいかにして吸収するかという問題である．

表現のずれの吸収は多くの研究で議論されてきた．テキスト検索では，クエリ拡張に代表されるように，様々な手法が試されている．また，Barzilay らのように，類義表現を自動獲得しようという試みもある [1].

本論文では，辞書定義文，シソーラスから得た大量の類義表現を組み合わせて使うことによって，様々な表現のずれを吸収するテキスト検索手法を提案する．大量の類義表現の組み合わせを全展開するといった単純な手法では，組み合わせ爆発を引き起こすため実用的に検索できないおそれがある．そこで，本手法では類義表現間の関係をあらかじめ調べておくことで組み合わせ爆発を起こすことなく効率的なマッチングを行なう手法を提案する．

2　様々な表現のずれ

本論文で扱う表現のずれとは，表記的には異なるが意味的には近い内容を持つ表現で，以下のようなものがある．

1. 同義関係

   かな漢字 地震 ＝ じしん ＝ 地しん

   表記ゆれ コンピュータ ＝ コンピューター

   同義語 発生 ＝ 起こる

   定義文 発災 ＝ 災害が発生する

2. 上位・下位関係
   災難 ← 災害 ← 地震，台風

3. 1 と 2 の複合
   発災 ≃ 地しんが起こる

このような表現のずれを吸収するには，同義関係辞書やシソーラスといった類義表現データベースが必要になる．この類義表現データベースは国語辞典の定義文や大規模コーパスを利用することで大規模なものが獲得することができる [2] が，本論文で問題としているのは，こういった大規模な類義表現データベースをいかに統合し，3 のような複雑な表現のずれの吸収するかということである．実際の文書でも 3 の例のように様々なレベルの類義表現が複合している表現は数多く見られる．

3　提案するテキスト検索システム

類義表現を組み合わせて複雑な表現のずれを吸収する一つのアプローチとして，類義表現データベースのすべての組み合わせを展開して検索を行なう手法が考えられる．しかし，これでは類義表現データベースが大規模になると組み合わせ爆発を起こすため実用的ではない．

また，別のアプローチで，検索時に類義表現データベースをダイナミックに参照し，表現のずれを吸収する手法 [3] もあるが，これも大規模な類義表現データベースに対しては探索範囲が広くなってしまい，マッチング処理に時間がかかりすぎるという問題がある．

そこで，本論文では，類義表現間の関係をあらかじめ調べておくことで組み合わせ爆発を起こすことなく効率的なマッチングを行なう手法を提案する．

まず，類義表現データベースの同義関係や上位・下位関係を ID で代表させる．さらに，類義表現データベース内の関係をあらかじめ調べておく．この作業のことをコンパイルと呼ぶ．以降の処理では，このコンパイルされた類義表現データベースを利用することにより，複雑な表現のずれを効率的に扱うことができる．

本システムでは図 1 のように，

1. 類義表現データベースのコンパイル
2. 検索対象テキストのインデキシング

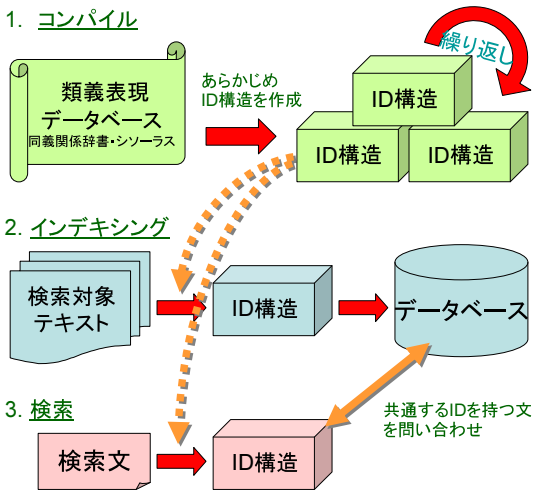

図 1: システムの概略図

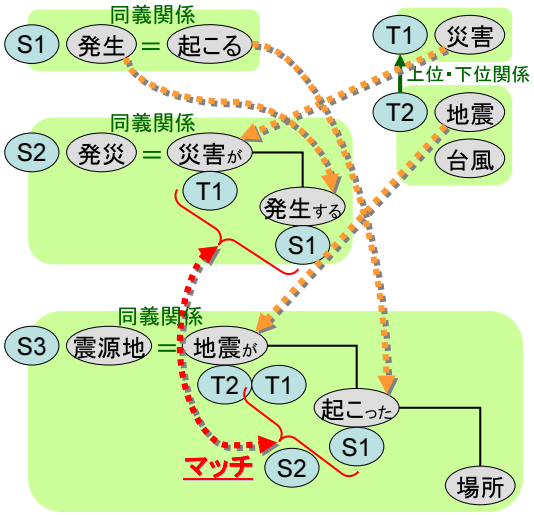

図 2: 類義表現データベースのコンパイル

3. 検索

の三つのステップで処理を行なっている．その中で最も重要な部分が 1 の類義表現データベースのコンパイルである．

3.1 類義表現データベースのコンパイル

類義表現データベースをあらかじめコンパイルしておき，それを利用することで，以降の検索対象テキストや検索文の処理を効率的に進めることができる．

このコンパイルの手順は以下のとおりである．

1. 同義関係への ID 付与

   図 2 のように「発生＝起こる」や「発災＝災害が発生する」といった同義関係のグループごとに S1，S2 といった ID をつける．

2. 上位・下位関係への ID 付与

   これも図 2 のように，上位語（災害）や下位語（地震，台風）ごとに T1，T2 といった ID をつける．

3. 形態素・構文解析

   類義表現データベースの各表現に対して形態素・構文解析を行ない，助詞や助動詞，その他ストップワードを除いたキーワード列を木構造にする．ここで，ひらがなの「あった」を含む表現があり，形態素解析器が「会った」や「合った」といった曖昧性の候補を挙げた場合は,「あった」に加えて「会った」と「合った」もキーワードとして扱うことでどちらの表現にもマッチできるようにする．

4. 上位・下位関係との関連付け

   キーワードが上位・下位関係で定義されていれば，キーワードにその ID を付与する．さらに上位語があれば上位語の ID も付与する．図 2 の例では，S2 の「災害」には T1 が，S3 の「地震」には T2 と T1 の 2 つの ID が付与される．この処理は全てのキーワードについて行なわれる．その結果，上下 1 段階の語がマッチの対象となる．

   このように，各表現をキーワードとそれに付随する ID を木構造で表したものを，ID 構造と呼ぶことにする．

5. 類義表現データベース内の関連付け

   類義表現データベース内の各 ID 構造同士でマッチングを行ない，ある同義関係の ID 構造とマッチする部分があれば，そこにマッチした同義関係の ID を付与する．

   例えば，S2 の「災害が発生する」の「発生」は，S1 の「発生」とマッチするので S1 の ID が付与される．同様に，S3 の「地震が起こった場所」の「起こった」の部分にも S1 が付与される．すると，「地震が－起こった」の部分が S2 の「災害が－発生する」と T1 － S1 でマッチし，この部分に S2 の ID が付与される．このように,「地震が起こった場所」という表現を ID 構造で表すことによって,「災害が起こった場所」や「発災した場所」など様々な表現とマッチできるようになる．

   このようにして類義表現データベース内の ID 構造同士のマッチングを繰り返し，これ以上新しい ID が付与できなくなるまで続ける．

3.2 検索対象テキストのインデキシング

検索対象テキストの各文に対して，類義表現データベースをコンパイルして得られた ID 構造とのマッチングを行ない，各文の ID 構造を作成する．

検索対象テキストの先頭のキーワードから順に，それをヘッドとする類義表現とのマッチングを考える．ID 構造は木構造で表現されており，木構造同士のマッチングは複数の子ノードがある場合もあるので，ヘッドのノードから子ノード方向に向かって探索が進む．

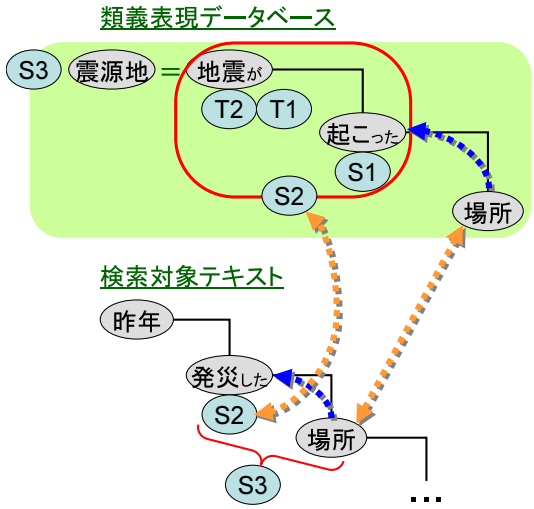

図 3: 検索対象テキストへの ID の付与

図 3 を例にして検索対象テキストに ID が付与される手順を説明する．ID の付与は,「昨年」「発災」「場所」･･･ の順で進む．この例では「昨年」には ID が何も付かず,「発災」には S2 が付く．そして,「場所」が S3 の右辺の ID 構造のヘッドとマッチすると，次に双方の係り元のキーワードについてマッチする ID があるかチェックする.「発災」の部分には既に S2 付与されており，S3 の右辺の ID 構造にも S2 があるため，これらがマッチする．S3 側の S2 にはこれ以上係り元がないため，これで ID 構造のマッチングが完了する．その結果,「発災した場所」の部分に S3 の ID が付与される．

このように，いま考えている部分より前の ID 構造を使ってマッチするかの判定をするため，先頭のキーワードから順に ID が確定できる．そのため，後戻りすることなく ID 構造を作成することができる．

また，あらかじめ類義表現データベースをコンパイルしているため，類義表現データベースの複数の表現を組み合わせて参照する必要がない．このため ID 付与の処理を効率的に行なうことができる．できあがった検索対象テキストの ID 構造はデータベースに保存される．

3.3 検索

検索文に対しても検索対象テキストの時と同様に ID 構造を作る．そして，検索対象テキストのデータベースから検索文の ID 構造と共通する ID を持つ検索対象テキストの ID 構造を抽出する．一般に，こうして得られる ID 構造は非常に多くなってしまう．そこで，データベースへの問い合わせの際に，マッチした ID 数をキーワード数で正規化した粗い類似度を求め，上位 1000 文程度に絞り込みを行なう．その後，正確な類似度計算を行なってランキングする．

類似度は，検索対象テキストに対して検索文のキーワードや係り受け関係がどのくらい一致するかの割合として定義し，以下の式で計算した．

$$\text{類似度} = \frac{\text{マッチしたキーワード数} + \text{マッチした係り受け数}}{\text{検索文の全キーワード数} + \text{検索文の全係り受け数}}$$

4 リソース

類義表現データベース

類義表現データベースは，同義関係辞書とシソーラスからなる．同義関係辞書は，国語辞典の定義文が 3 文節以内の短いものであれば言い換え可能な表現であると考え，それらを自動で収集した．また，国語辞典で類義語として与えられているものも同義関係辞書に加えた．シソーラスは国語辞典の定義文から自動構築したものを利用した．類似度の計算では，上位・下位関係の ID によってマッチしたキーワードには 0.9 倍することでペナルティを与えた．

表 1 のように，ドメインに依存しない大規模な類義表現データベースを得ることができた．

表 1: 自動構築した類義表現データベース

同義関係辞書のグループ数	6520
同義関係辞書の全表現数	16654
シソーラスの分類数	4425
シソーラスの単語数	18906

検索対象テキスト

本システムの検索対象テキストとして，阪神・

淡路大震災教訓情報資料集 [4] からの約 2000 文と，IREX[5] の情報検索課題の検索対象テキストに指定されている毎日新聞 94 年・95 年の全記事，約 270 万文の 2 つのテキスト集合を用いた．

5 考察

本システムで複雑な表現のずれを吸収し，マッチできた例を示す．

例 1 検索文：「地震 が 起きた すぐ あと」

シソーラス↓　‖同義語

災害 が 発生 する ‖ 同義語

定義文 ‖

検索テキスト：「発災 直後」

$$\text{類似度} = \frac{3.9 + 3}{4 + 3} = 0.99$$

例 2 検索文：「癌 の 告知」

かな漢字 ‖　↓シソーラス

検索テキスト：「がん を 知らされ て … 」

$$\text{類似度} = \frac{1.9 + 1}{2 + 1} = 0.97$$

検索対象テキストで用いた地震の教訓集も新聞記事もある程度統一された表現で記述されている．しかし，一般的に検索文は様々な表現のバリエーションがあるため，従来の単純なマッチングしか行なわない検索システムでは，うまく検索できない検索文もあった．例えば，地震の教訓集には「火事」という表現は一切使われておらず，すべて「火災」に統一されている．このため，「火事」では何も検索できなかった．

本システムでは，コンパイルされた類義表現データベースを参照することで，様々な検索文に対して柔軟に検索することができた．

しかし，類義表現データベースを利用したために，検索されて欲しくないものまで検索されてしまう例もあった．その代表的なものとして「夏の天気」・「冬の天気」といった時間を含む表現がある．これらは，「夏」と「冬」共に上位語である「四季」の ID が付くため，非常に近い関係であると判断されてしまう．今後，どのような上位・下位関係があるときにこういった不都合が起こるのかを調査し，上位・下位関係を利用する場面を制限するなどして対処したい．

また，類義表現データベースをさらに整備し，より柔軟なマッチングを可能にする枠組みを考える必要がある．例えば，「延焼」の定義文は，「火事が次々に燃え広がる」であるが，「次々に」の部分は必須の要素ではない．このような場合，類義表現データベースに「次々に」の部分が必須ではないという情報も与えておく．そして，類義表現データベースのコンパイルの際に，必須でないキーワードは飛ばしてもよいというルールを新たに付け加えることで「火事が燃え広がる」ともマッチさせることが可能になる．

6 おわりに

本論文では，日本語表現における様々な表現のずれを大規模な類義表現データベースと ID を用いた意味表現によって吸収し，柔軟にマッチングを行なう手法を提案した．そして，類義表現データベースをあらかじめコンパイルしておき，それを利用することでインデキシングや検索の処理を効率的に行なうことができた．

今後は，テキスト検索システムとしてもさらに完成度を高めていくつもりである．本システムでのスコア計算は検索文との類似度として計算しているが，それぞれのキーワードの重みは考慮していない．そこで，キーワードの重みとして tf・idf などを用いることで，柔軟かつ強力なテキスト検索システムにしていきたい．

参考文献

[1] Regina Barzilay and Lillian Lee. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *HLT-NAACL 2003*, pp. 16–23.

[2] 大福泰樹, 河原大輔, 黒橋禎夫. 大規模コーパスと国語辞典の統合的利用によるシソーラスの自動構築. 言語処理学会 第 10 回年次大会 発表論文集, pp. 341–344, 2004.

[3] 黒橋禎夫, 酒井康行. 日本語表現の柔軟な照合. 言語処理学会 第 7 回年次大会 発表論文集, pp. 343–346, 2001.

[4] 阪神・淡路大震災教訓情報資料集. http://www.hanshin-awaji.or.jp/kyoukun/index.html.

[5] IREX. http://www.csl.sony.co.jp/person/sekine/IREX/.