Noisy-channel model を用いた略語自動推定

村山 紀文
東京工業大学大学院
総合理工学研究科
murayama@lr.pi.titech.ac.jp

奥村 学
東京工業大学精密工学研究所
oku@pi.titech.ac.jp

1 序論

近年，ウェブページの検索やテキストマイニングなどに代表されるように，大量の文章から特定の商品や人物などの情報や評判を獲得する技術に関心が持たれている．しかし，これらの目的を達成するためには，対象となる商品などの語の同義語の存在に注意しなければならない．例えば,“ プレイステーション ”に関する情報をできる限り多く得るためには,「プレイステーション」と書かれた文書だけではなく,「プレステ」あるいは「Play Station」「PS」などの「プレイステーション」の同義語が出現する文書全てを調査する必要がある．これらの目的を達成するために，自動的に対象の語の同義語を網羅するようなシステムが望まれる．

我々はこれらの同義語のうち，まず「略語関係」にある同義語に注目し，そのような語を自動的に獲得するようなシステムを目指す．本論文では,「略語関係」にある対を,「同義関係にある語の集合の中で，一方がもう一方から表層的に短縮されているような語の対」であると定義する[1]．また，短縮されている語を「略語」，短縮されていない語を「原語」と呼ぶ．

このような略語関係は，一般の人々によって生成され，広められることが多い．そのため，他の同義語と比べると流動的であり，捉えることが難しい．また，新聞などの文字数の制限がある文書や，blog などの一般の人々の書く文書では使用される頻度も多く，同義語の中でも特に重要な関係であると考える．

これまでの略語関係獲得の研究では，若者言葉的な略語は敬遠され特に扱われてこなかったが，我々の研究ではそのような略語も扱えるモデルを目指す．このような略語を幅広く収集するためには，これまでの略語関係獲得の研究で多く採られてきたようなテンプレートを使った手法では難しい．また，そのような略語は原語から生成される際のルールも複雑であり，これまで使われてきたような「要素の先頭から１，２文字取る」といったような単純なルールだけで生成することも難しい．

そこで本研究では，入力された原語から確率モデルにより略語候補を生成し，web 上の情報を用いて検証を行い，対応する略語を出力するシステムの提案を目指す．本論文では特に，第一段階である，原語から表層的な情報のみを利用して確率モデルにより，略語候補を生成する手法の提案を行う．

[1]「プレイステーション」と「PS」は直接の略語関係ではなく，同義語の略語であると考える．そのため，今回扱う略語関係にあるとは考えない．

2 関連研究

2.1 辞書自動生成

原語・略語対の自動獲得は，以前より辞書の自動生成の一環として行われてきた [1][2]．これらの手法は,「A を略して B」「A（以下，B と略す）」のような略語関係が記述されやすいテンプレートをあらかじめ用意し，それを利用して抽出を行う手法が多かった．これらの手法は，文体が一定の文章からの抽出にはよい結果を示しているが，様々な文体が入り混じるコーパスから広く抽出することは難しいと考えられる．

2.2 その他の原語－略語対自動獲得

テンプレートを用いずに原語－略語対の自動獲得を行っている研究として，[3][4][5] が挙げられる．

[3][4] では，新聞コーパスからの原語－略語対の獲得を行っている．酒井らは，まず原語－略語対候補から，表層上の制約ルールを用いて絞り込みを行い，次に両方の間接共起頻度により更なる絞り込みを行っている．[5] では，原語からの省略ルールをいくつか用意して略語候補を生成し，略語候補の web 上での出現数，辞書の情報などを用いて略語を推定している．

いずれの研究も，原語からの省略はルールベースで行われており，あらゆる略語関係に対して十分に柔軟であるとは言い難い．

2.3 英語の原語－略語対自動獲得

英語に対する原語・略語対の自動獲得の研究もいくつか行われている [6][7][8]．多くの場合，英語の自動獲得手法は文章中から略語を発見し，それに対応する原語を発見することで行われる．これは，日本語の略語と異なり，文章中から略語が発見しやすい[2]点をうまく利用したものであると言える．

また英語の場合，略語は原語から単語の先頭一文字を取ることで生成される場合が大半であり，日本語の生成規則よりもはるかに単純であることも重要な点である．

3 提案手法

3.1 略語の性質

提案手法について述べる前に，日本語の略語の特性について述べ，我々が挑む問題がどのようなものであるかを説明する[3]．

まず，我々人間が原語から略語を考え出すメカニズムを，図 1 に挙げる例と共に説明する．

[2]単語区切りが明確であり，大文字が含まれているなどの表層上の特徴が顕著であるため．

[3]これらの特性を把握する手がかりとして，[9] を参考にした．

原語が単語であった場合は，a)b) のように原語から直接数モーラ[4]が抜粋されて，略語が生成されることが多い．このとき，先頭の３，４モーラが選択されることが多いとされているが，それ以外のモーラ数や後方数モーラが選択されることも少なくない．

原語が複合語であった場合は，c)d) のように，まず原語は語基と呼ばれる構成要素に分割される．その上で各構成要素から数モーラが抜粋され，それらが結合されることで原語の略語が生成される．このような複合語の略語は，２つの要素から２モーラずつ抜粋されて結合された，２＋２モーラ型が多いことが知られている．また d) のように１要素そのままを取って，略語とされることも少なくない．

基本的に，単語や構成要素からの抜粋は１つの部分文字列のみであり,「コンピュータ」から「コンタ」のように抜粋されることはない．しかし，原語が仮名の場合は，e) のように長音（「ー」）や撥音（「っ」）が省略されて抜粋されることが有りうる．

１つの原語から上のような方法で想定されうる略語候補は数多くあるが，それらの中で実際に使用される略語は少ない．実際に使用される略語の選択方法はまだ完全に解明されていないが，原語の使用される頻度，略語の語感，原語と略語のイメージの差，略語に同音異義語が存在するか，または実際に略語を用いるコミュニティの嗜好や特性など，様々な要因がからんでいると考えられる．

3.2 方針

我々のシステムは原語から略語を生成するアプローチを採る．その理由の１つは，先に挙げたように，テンプレートを用いた手法は web 上での適用を考えた場合，適切でないという点である．略語から原語を生成する手法も考えられるが，存在しない情報を補完しなければならないため，非常に困難であると考えられる．また，原語からの略称生成のアプローチは検索エンジンの Query Expansion 機能としてそのまま流用できることも大きな利点であるといえる．

我々の目指すシステムの全体像は以下のようなものを想定している．

1. 入力の原語を適切な要素に分割
2. 原語要素列から，略語候補を出力
3. Web 上の情報を利用して，略語候補の絞り込み

これらのうち最初の２つのステップは，先に示したような人間が略語を考え出すときのモデルを模倣したステップである．

最後のステップでは，以上のように得られた略語候補の中から，Web 上の情報を用いて絞り込みを行い，もっとも適切な候補を選択する．ここでは，各略語候補について,「実際に原語と略語候補が同義語関係にあるか」を検証することによって，略語候補の取捨選択を行う．我々のシステムはこのステップで Web 上の情報を使用する事によって，新聞などには登場しない若者言葉的に使われている略語も獲得出来ると考える．

[4]モーラとは，拍を表わす単位であり，日本語の場合は大抵仮名１文字が１モーラにあたるが,「ファ」などは２文字で１モーラに相当する．

	原語	➡	原語要素列	➡	略語
a)	ヘリコプター	➡	➡	➡	ヘリ
b)	アルバイト	➡	➡	➡	バイト
c)	東京工業大学	➡	東京,工業,大学	➡	東,工,大(東工大)
d)	スーパーマーケット	➡	スーパー,マーケット	➡	スーパー
e)	スーパーコンピュータ	➡	スーパー,コンピュータ	➡	スパ,コン(スパコン)

図 1: 原語－略語対の例

以降では，第二のステップに関して主に取り上げ，説明を行っていく．

ここで考えなければならないのは，第三のステップにかかる検索コストである．単純なルールで略語候補を生成した場合，候補数が非常に大量になる危険性がある．略語候補の情報を検索エンジンに問い合わせるコストを考えると，略語候補は出来る限り少なく，かつその中に正解を含んでいなければならない．そこで，本研究では略語候補の生成に，確率モデルを使用する．これにより，略語候補は原語に対してもっともらしい順に順位を持って出力され，出力のうち上位のものから検証を行うことにより，効率的な検証が行える．

我々は，ここでの確率モデルに，Noisy-channel model を利用する．

3.3 Noisy-channel model

本研究では，略語候補生成のモデルとして，Noisy-channel model を利用する．Noisy-channel model は，翻訳 [10] や要約 [11] などに使用されてきた確率モデルである．以下で，我々のタスクにおける Noisy-channel model の考え方を簡単に説明する．

原語要素列 O が与えられたとき，我々が求めるべき略語要素列 A は式 (1) で示される．これを変形していくと，式 (3) のようになる．

$$A = \arg\max_{A^*} P(A^*|O) \quad (1)$$

$$= \arg\max_{A^*} \frac{P(A^*)P(O|A^*)}{P(O)} \quad (2)$$

$$= \arg\max_{A^*} P(A^*)P(O|A^*) \quad (3)$$

式 (3) において，$P(O|A^*)$ は A^* から O への擬似的な「変換モデル」であると考えることができ，$P(A^*)$ は A^* の略語らしさ，すなわち「言語モデル」であると考えることが出来る．

特に，$P(A^*)$ は我々のタスクにおいて重要な意味を持っており，この確率を正確に求めることが出来れば，このモデルから高い確率で出力される略語候補は，より略語らしいものであることが保障される．

3.4 変換モデル

$P(O|A)$ は「略語要素列 A から原語要素列 O が生成される確率」である．本研究では，この確率を４つのルールの生起確率の積によって考える．これらのルールは，図 2 のような変化過程に基づいている．

我々のモデルでは，略語から原語への変換は大きく２つのステップに分かれる．

- Expand Step: 各略語要素が対応する原語要素へ拡張される
- Insert Step: 対応する略語要素がない原語要素が挿入される

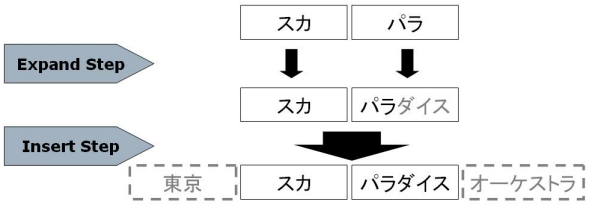

図 2: $P(O|A)$ のモデル化

それぞれステップは，2つのルールの適用で表わされる．

3.4.1 Expand Step

Expand Rule 1:E1 この E1 は，略語要素にどのように文字が足されて対応する原語要素になるかを示すルールである．以下の要素から構成される．

- *type*: 略語要素の文字タイプ[5]
- *abbnum*: 略語要素の文字数[6]
- *headnum*: 略語の前方に付けられる文字数
- *char*: 略語の直後に付けられる文字

E1 は「*type* の *abbnum* 文字からなる略語が，略語の前方に *headnum* 文字足し，後方に *char* から始まる文字列を足す」というルールである．このルールは，3.1 節に挙げた原語要素から略語要素が抜粋される際の特徴を考慮したうえで，その逆変換を考えたものである．

この生起確率は，以下の式で求められる．

$$P(E1) = P(headnum, char|type, abbnum)$$

例えば，図 1 の a) の「ヘリ」→「ヘリコプター」の変換は以下の式になる．

$$P(E1) = P(0, コ \mid 仮名, 2)$$

Expand Rule 2:E2 この E2 は，仮名略語における撥音，長音の省略を例外的に扱うためのルールである．以下の要素から構成される．

- *type*: 略語要素の文字タイプ
- *abbch*: 略語要素の中に含まれる長音記号（「ー」），撥音文字（「っ」）
- *wordch*: 原語要素の中に含まれる長音記号，撥音文字

E2 は略語候補の文字間に長音，撥音が挿入されることを示すルールとなっている．この生起確率は，以下の式で求められる．

$$P(E2) = P(wordch|type, abbch)$$

例えば，図 1 の e) の「スパ」→「スーパー」の変換は以下の式になる．

$$P(E2) = P(長音 \mid 仮名, なし)$$

3.4.2 Insert Step

Insert Rule 1:I1 Insert Step では，対応する略語要素がない原語要素を，略語要素列に挿入する．まずルール I1 では，原語要素がどの位置にどれだけ挿入されるかを決定する．

- *type*: 略語要素列の文字タイプ
- *abbnum*: 略語要素数
- *top*: 略語要素列の前に挿入される原語要素数
- *middle*: 略語要素の間に挿入される原語要素数
- *bottom*: 略語要素列の後に挿入される原語要素数

この生起確率は，以下の式で求められる．

$$P(I1) = P(top, middle, bottom|type, abbnum)$$

図 2 の e) の場合は以下の式になる．

$$P(I1) = P(1, 0, 1| \text{漢字} + \text{仮名}, 2)$$

Insert Rule 2:I2 次に，ルール I1 で決定された位置に，どのような原語要素を挿入するかを決定する．

- *type*: 略語要素列の文字タイプ
- *place*: 挿入場所（top,middle,bottom）
- *wordtype*: 挿入する原語要素の文字タイプ
- *wordnum*: 挿入する原語要素の文字数

この生起確率は，以下の式で求められる．

$$P(I2) = P(wordtype, wordnum|type, place)$$

図 2 の e) の原語要素「東京」の挿入は以下の式になる．

$$P(I2) = P(\text{漢字}, 2| \text{漢字} + \text{仮名}, top)$$

3.5 言語モデル

$P(A)$ は A の略語らしさを示す．本研究では，以下の情報を用いて略語らしさを定義する．

- wordcharnum: 略語の文字数
- elementnum: 略語の要素数
- charnum: 略語の要素ごとの文字数

これらの生起確率は，略語の文字タイプを条件とした，条件付き確率として求められる．

3.6 確率計算

$E1, E2, I1, I2$ と $P(A)$ のためのルールが全て互いに独立であると仮定すると，求めるべき確率は，以下のように定義できる．

$$P(A)P(O|A) \quad = \quad P(E1)P(E2)P(I1)P(I2)P(A)$$

それぞれの実際の確率は，訓練データでの出現数を元に計算される．

ここで，全ての略語候補について結果を計算していると膨大な計算量になるため，実際にはビーム探索法を用いて動的に計算を行っている．

4 実験

以上のモデルを用いて，実験を行った．

実験には，Web 上の百科事典 Wikipedia[7]に略語，略称関係にあると明記してあるものを抜粋したものを用いた．これらには，１つの原語に対して複数の略語が存在するものも含まれている．結果，748 語の原語とそれらに対応した略語 851 語を得られた．

これらの原語は，事前に全て人手で要素に分割を行った．

[5]文字タイプは対象の文字種により,「仮名」「漢字」「アルファベット」「数字」「以上の組み合わせ」「記号のみ」の値を取る．

[6]本研究では，基本として文字で数えるが，仮名においてはモーラで数えている．

[7]http://ja.wikipedia.org/wiki/

表 1: 実験結果

	再現率	精度
上位 1 位	0.135	0.153
上位 2 位	0.214	0.121
上位 3 位	0.253	0.096
上位 5 位	0.307	0.079
上位 10 位	0.518	0.060
上位 20 位	0.614	0.038
上位 30 位	0.684	0.030
baseline	0.226	0.060

我々のモデルの目的は，次の検証の段階のために，正解を含みつつ出来るだけ少ない候補数で略語候補群を生成することである．そこで評価においては，モデルの出力から上位 n 位を取り，それらに対して以下に定義する再現率と精度を計算する．

$$\text{再現率}_n = \frac{\text{(上位 } n \text{ 位の出力のうち，正解に含まれる数)}}{\text{(原語に対する正解の略語数)}}$$

$$\text{精度}_n = \frac{\text{(上位 } n \text{ 位の出力のうち，正解に含まれる数)}}{\text{(上位 } n \text{ 位の全出力数)}}$$

ここで，再現率は「出力の，全略語に対するカバー率」であると言える．精度は次のステップにおいて，100 個の出力を検証して得られる正解の略語数，すなわち「次のステップにかかるコスト」を示している．

比較となる base-line は,「原語の各要素」と「原語の１、２番目の要素から先頭 1,2 文字ずつ獲得して結合したもの」を用いた．

実験は，上記のデータを用いた５分割交差検定で行った．実験結果を表 1 に示す．

4.1 考察

表 1 に示した結果では，上位１～３位の再現率が低いが，このモデルの出力が次のステップでの検証のためであることを考えると，大きな問題ではないと考える．

また，上位５位～１０位の結果で base-line，すなわち単純なルールベースの結果を上回ることができた．また，上位 30 位の結果では 68%の略語をカバーすることができた．

今回示した結果は，まだ改良の余地はあるものの，現段階で十分な結果を残せたものと考える．

5 結論

本論文では，原語からの略語自動生成のシステムの第一段階として，Noisy-channel model を用いた略語候補生成について述べた．本モデルは，様々な略語生成ルールに柔軟に対応でき，生成した略語候補をより確からしい順に出力することが出来る．

本モデルは，我々のシステムの一部として考えたものではあるが，原語と略語の表層的な対応関係の確からしさを計れる点で他のシステムへの流用が可能である．たとえば，テンプレート方式で得られた原語－略語対が本当に正しいかどうかの検証にも使うことが出来る．

我々の今後の予定としては，まず今回のモデルの性能を高めることを第一に考えている．このためには，原語の長さなどの特徴により異なる統計量を使用する，$P(A)$ の計算に略語の音韻的な情報を利用するなどが考えられる．

また，$P(A)$ を実際に A が出現する確率で計算してみることも考えている．これは，web の検索エンジンの Hit 数を基に求めることを想定しているが，そこには検索コストがかかるため，現行のモデルの上位の結果のみを計算し，リランキングを行うことを想定している．

同時に，原語の要素への分解，web 情報での検証についても研究を進めていきたい．

謝辞

本研究は文部科学省科学研究費 (21 世紀 COE プログラム「大規模知識資源の体系化と活用基盤構築」) の補助のもとに行われた。

参考文献

[1] 桜井裕, 佐藤理史. ワールドワイドウェブを利用した用語説明の自動生成. 情報処理学会論文誌, Vol. 43, No.5, pp. 1470–1480, 2002.

[2] 土田正明, 松井藤五郎, 大和田勇人. World wide web を用いた辞典システムの構築. 第 18 回 人工知能学会全国大会, 1A3-04, 2004.

[3] 酒井浩之, 増山繁. 名詞とその略語の対応関係のコーパスからの自動獲得. 電子情報通信学会論文誌 D-II, vol.J85-D-2, no.10, pp. 1624–1628, 2002.

[4] 酒井浩之, 増山繁. 略語とその原型語との対応関係のコーパスからの自動獲得手法の改良. 自然言語処理, Vol.12, No.4, pp. 207–231, 2005.

[5] 桝井文人, 松田良一, 野呂康洋, 河合敦夫, 井須尚紀. World wide web を知識源としたカタカナ語省略形の自動生成. 2004 年度電子情報通信学会ソサイエティ大会講演論文集, A-13-1, 2004.

[6] James Pustejovsky, Jose Castano, Brent Cochran, Maciej Kotecki, Michael Morrell, and Anna Rumshisky. Linguistic knowledge extraction from medline: Automatic construction of an acronym database. In *10th World Congress on Health and Medical Informatics (Medinfo 2001)*, 2001.

[7] Manuel Zahariev. An efficient methodology for acronym-expansion matching. In *Proceedings of the International Conference on Information and Knowledge Engineering, IKE 03, volume 1*, 2003.

[8] Youngja Park and Roy J. Byrd. Hybrid text mining for finding terms and their abbreviations. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2001.

[9] 窪薗晴夫. 新語はこうして作られる もっと知りたい!日本語. 岩波書店, 2002.

[10] Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. A statistical approach to machine translation. In *Computational Linguistics, 16(2):7985*, 1990.

[11] Hal Daume III and Daniel Marcu. A noisy-channel model for document compression. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-2002)*, 2002.