段階的文脈拡張による多義性解消

黒川 勇輝　　新里 圭司　　黒橋 禎夫
京都大学 大学院情報学研究科
{kurokawa, shinzato, kuro}@nlp.kuee.kyoto-u.ac.jp

1　はじめに

従来の多義性解消手法は，手掛かりとして利用する語を多義語周辺の数語に固定する方法が一般的である [4, 1]．しかしながら，このような語数を限定する手法は，手掛かりが不十分，または不明瞭になるという問題がある．例えば図1(A)において多義語「マック」の語義（マクドナルドまたはマッキントッシュ）を判定する場合,「マック」の近傍に出現する「デパート」や「勉強」を考慮するだけでは，どの語義か判断することは難しい．この場合「マック」から離れた場所に出現している「ハンバーガー」や「食べる」などを考慮することで，マクドナルドの語義と判断できるようになる．一方，(B)に関して多義語「グレイハウンド」の語義（バスまたは犬）を判断する場合,「グレイハウンド」の近傍に出現する「犬」や「走る」を考慮するだけで犬の語義とわかる.「マック」のように，離れた位置に出現する「蒸気」「機関車」などを考慮すると，かえって文脈が不明瞭となり語義判定が難しくなる．

以上より，多義性解消に利用する文脈の幅は文書や多義語によって異なると考えられる．本研究では，十分な確信度をもって語義が判定できるまで，利用する文脈の幅を徐々に拡張する手法を提案する．これにより近傍の単語のみでは文脈が不十分になる問題や，離れた位置にある語を考慮することで文脈が不明瞭となる問題の解決を図る．

2　関連研究

Stokoe ら [5] は，段階的に言語的知識を利用する多義性解消手法を提案している．多義語周辺に，コロケーションや共起語が出現した場合は，コーパスから学習した素性を用い，文脈に手掛かりがない場合や，素性がコーパスに出現しなかった場合は，WordNet に定義された語義確率を用いて曖昧性解消を行う．Semcor1.6 で評価を行った結果，語義の頻度のみを用いるベースラインと比較して，精度が向上することを示した．Stokoe らの手法は，学習時に考慮する文脈はあらかじめ固定されている．本研究では，文脈として考慮する多義語からの距離を，確信度が得られるまで段階的に拡張していく点で異なる．

(A) ... 図書館に行く代わりに，デパート横の**マック**で勉強します．長い間座っていると店員に注意されるので，ハンバーガーを注文して食べながら..
(B) ... 蒸気機関車のようだ」と語った．セントサイモンの疾走はドッグレースに使われる犬「**グレイハウンド**」の走り方そっくりだった．調教師は..

図 1: 多義語を含む文書

Milne ら [3] の研究は，語義確率と類似度を考慮している点で本研究と類似している．Milne らは，語義確率と類似度のバランスを考慮する際，語義確率，類似度，文脈の重要度を素性として機械学習することで98.4%の精度を達成している．しかし，Milne らはウィキペディアの記事を利用して類似度を計算するため，ウィキペディア記事に出現する名詞にしか適用できない．一方，提案手法はウェブコーパス中の内容語との共起を基に作成するベクトルのコサイン類似度で曖昧性解消を行うため，ウィキペディアにない一般名詞や読みの曖昧性解消にも適用できる．

3　ウィキペディアからの多義性解消のための知識獲得

本研究では，多義性解消にとって重要な知識である多義語とその定義文を，図2に示すウィキペディアの**曖昧さ回避ページ**より獲得する．このページには多義語と各語義の定義文が記述されており，タグ等を手掛かりに効率的にそれらを獲得することができる．

また，ウィキペディア記事間に張られている**パイプ付きリンク**に注目することで語義の使われ易さ（**語義確率**）を求める．この語義確率は多義性解消時の有効な手掛かりとして知られている [2, 3]．

本節では多義語とその定義文の獲得方法，及び語義確率の計算方法について述べる．

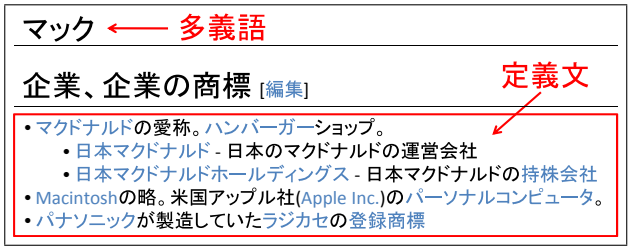

図 2: 曖昧さ回避ページの例

3.1 多義語と定義文の獲得

多義語は<title>タグを，定義文は「*」を手掛かりに曖昧さ回避ページから多義語とその定義文を抽出した．この結果，多義語 3 万語，定義文 18 万 5 千文が得られた．この中には多義語や定義文として不適切なものが含まれているため，以下のフィルタリングを行う．なお，フィルタリングの結果，最終的に多義語 3,871 語，定義文 8,714 文が得られた．

3.1.1 多義語のフィルタリング

獲得された多義語には一般には多義語と考えにくい語が含まれている．例えば,「女優」は俳優に加え，ドラマ名の語義がウィキペディアで定義されている．しかしながら,「女優」は俳優の意味で利用される場合が圧倒的に多いと考えられる．このようにウィキペディアより多義語として獲得された一般語はほぼ一義であると考えられる．そこで，JUMAN の辞書に登録されている語を一般語と見なし，得られた多義語から削除した．

また,「桜岡小学校」のように多義性を解消する機会が少ない語も多義語として獲得された．そこでウェブ文書 1 億件での文書頻度が 1,000 未満の語を削除した．

3.1.2 定義文のフィルタリング

獲得した定義文には，語義が類似したものやマイナーな語義のものが含まれており，次の方法でそれらをマージまたは削除する．

類似した語義の定義文のマージ　語義が類似した定義文を以下の 2 段階に分けマージする．2 段階に分けることで，より高い精度での定義文のマージが期待できる．

Step 1: 主辞に基づく定義文のマージ
Step 2: クラスタリングによる定義文のマージ

Step 1 では，各定義文の第 1 文から主辞を抽出し，主辞が一致したもの，もしくは主辞が「駅」「県」「町」となっている定義文同士をマージする．このようなルールベースの手法をとることで，高い精度で類似した語義の定義文をマージすることができる．

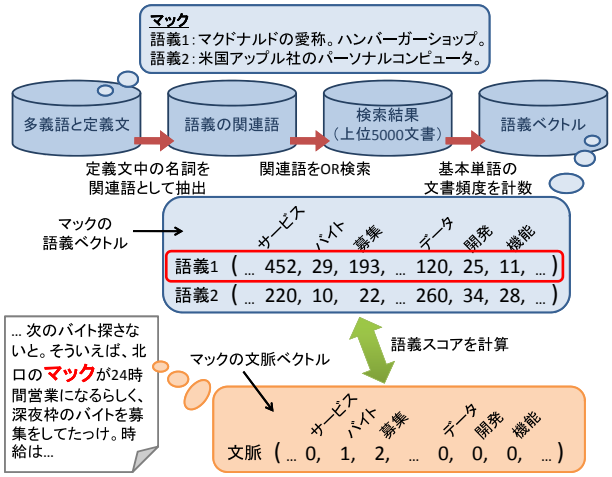

図 3: 提案手法の概要

Step 2 では，Step 1 でマージされなかった定義文をクラスタリングによりマージする．具体的には，4.1 節で述べる語義ベクトルを各定義文について作成し，そのコサイン尺度に従って定義文をマージする．

マイナーな語義に関する定義文の削除　マイナーな語義が文書中で用いられることは少ないと考え，3.2 節で述べる語義確率が 0.1 未満のものは削除した．

3.2 語義確率

多義語の語義の使われやすさには偏りがあると考えられる．例えば，多義語「マック」にはマクドナルド，マッキントッシュの他にラジカセの語義もあるが，ラジカセの意味で「マック」が用いられることは稀である．この語義の偏りが事前に分かれば，多義性解消時の有効な手掛かりになると考えられる．

ウィキペディアの記事には，利用者が閲覧中の記事と関連のある語について効率良く情報を収集できるよう，関連語に対応した記事へリンクが張られている．関連語が多義語の場合は，[[マクドナルド | マック]] のようにパイプ付きリンクを用いてリンクが張られており,「マック」がブラウザ上の表記,「マクドナルド」がリンク先の記事のタイトル（語義）を表す．そこで，パイプ付きリンクを収集し，多義語とそのリンク先の数を計数することで語義確率を求める．

4 提案手法

図 3 に示すように，提案手法は多義性解消に必要なデータを作成するオフライン処理と，多義語を含む文書が与えられた際に行うオンライン処理に大別される．

オフライン処理では，曖昧さ回避ページから獲得した定義文中の名詞（固有表現，複合名詞，一般名詞）

をもとに，各語義の特徴ベクトル (**語義ベクトル**) を作成する．語義ベクトルは，定義文中の名詞を検索エンジン TSUBAKI[1] で OR 検索し，検索結果上位 5,000 件における**基本単語**の文書頻度をもとに作成する．基本単語とはウェブ文書 1 億件の文書頻度上位 10,000 語[2] である．

次にオンライン処理について述べる．多義語を含む文書が入力として与えられると，多義語近傍の基本単語に基づき特徴ベクトル (**文脈ベクトル**) を作成し，各語義ベクトルとの類似度及び語義確率を考慮した**語義スコア**を求める．語義スコアをもとに**確信度**を求め，十分な確信度が得られた場合，最もスコアの高い語義を出力する．確信度には語義スコアの第 1 位と第 2 位の差を用いる．十分な確信度が得られない場合，文脈の幅を拡張し，より多くの基本単語から文脈ベクトルを作成し語義スコアおよび確信度の再計算を行う．この操作を十分な確信度が得られるまで繰り返し行う．

例えば，図 3 の多義語「マック」を含む文書が入力として与えられると，はじめは多義語近傍に出現する基本単語「北口」「時間」を基に文脈ベクトルを作成し，語義スコアを計算する．この時点で十分な確信度が得られた場合，スコアが最大の語義が出力されるが，この例の場合「北口」「時間」は「マック」のどの語義にも特徴的でないため，十分な確信度を得ることができない．そこで，文脈を拡張して「営業」「募集」「時給」などの基本単語も考慮した上で再度文脈ベクトルを作成する．すると今度は，マクドナルドの語義に特徴的な基本単語が文脈ベクトルに考慮されたため，十分な確信度で「マック」の語義がマクドナルドであると推定できる．

本節では，語義ベクトル，文脈ベクトルの作成方法，語義スコアの計算方法について述べる．

4.1 語義ベクトルの作成

語義ベクトルの各要素は基本単語の文書頻度を元に計算する．各要素の値は以下の式で求める．

$$W(s,w) = DF(w,d_s) \cdot IDF(w)$$

ここで s は語義，w は基本単語，d_s は語義 s に対応する定義文に含まれる全名詞を検索エンジン TSUBAKI で OR 検索[3] した際の上位 5,000 文書，$DF(w,D)$ は文書集合 D における w の文書頻度，$IDF(w)$ は TSUBAKI が検索対象とするウェブ文書 1 億件から求めた w の IDF 値である．$W(s,w)$ は，d_s 中に頻出し，かつウェブ文書 1 億件で文書頻度が低い語に対して高い値を与える．

4.2 文脈ベクトルの作成

文脈ベクトルの作成方法について，以下に示す多義語「マック」を含む文書を例に述べる（下線は基本単語）．

..営業 店舗 で 夜 を 明かす 人々 を 指す「**マック** 難民」なる造語も 生まれ た コーヒー 一 杯 で 宿泊，**マック** 難民 が 急増，J-CASニュース2007 年 3 月 30 日付 配信．
ネット カフェ と同様..

まず多義語の係り元，係り先の基本単語「難民」から文脈ベクトルを作成する．次節で述べる語義スコアを計算し確信度が得られない場合は，文脈を拡張して考慮する基本単語を増やす．具体的には，多義語を中心に window 幅を設定し，幅を 5，10，20 の順に大きくして window 幅内の基本単語を文脈ベクトルに追加する．window 幅は基本単語のみを考慮する．例えば window 幅 5 の場合，基本単語「店舗」「ネット」など含む，多義語前後に出現する 10 語から文脈ベクトルを作成する．

文脈ベクトルの各要素の値は $Freq(w) \cdot IDF(w)$ で求める．ここで w は基本単語，$Freq(w)$ は window における w の出現頻度である．文書中に同じ多義語が複数回出現する場合は同じ語義を持つと考え，個々の文脈ベクトルの和を求める．

4.3 語義スコア

語義ベクトルと文脈ベクトルの類似度，および語義確率を用いて語義スコアを求める．語義 s の語義ベクトルを $\mathbf{v_s}$，文書 d の文脈ベクトルを $\mathbf{v_d}$，s の語義確率を $P(s)$ としたとき，語義スコア $Score(s,d)$ を以下の式で求める．

$$Score(s,d) = \alpha \cdot \cos(\mathbf{v_s}, \mathbf{v_d}) + (1 \quad \alpha) \cdot P(s)$$

ここで $\cos(\mathbf{x},\mathbf{y})$ は $\mathbf{x}$ と $\mathbf{y}$ のコサイン尺度，α は類似度と語義確率の混合比を表す．

5 評価実験

5.1 データセットの作成

ウィキペディア記事間のリンクを利用しデータセット（パラメータ推定用セット，評価セット）を構築した．前述したように，ウィキペディアの記事は，関連のある語について説明されている記事へのリンクが張られている．関連語が多義語の場合はパイプ付きリンク（[[マクドナルド | マック]]）が用いられており，リンク

[1] http://tsubaki.ixnlp.nii.ac.jp/
[2] 「こと」「もの」等を含む 571 語はストップワードとして除く．
[3] 各名詞はフレーズとして扱う．

表 1: 従来手法と提案手法の精度比較

手法	語義確率 無		語義確率 有		
	精度 (%)	閾値	精度 (%)	α	閾値
係り受け関係	72.44	–	86.81	0.58	–
window5	80.59	–	90.81	0.61	–
window10	80.81	–	91.41	0.62	–
window20	79.56	–	91.56	0.63	–
係 ▷w5	80.81	0.7	90.89	–	0.3
係 ▷w5▷w10	**81.70**	0.6,0.2	91.26	–	0.3,0.5
係 ▷w5▷w10▷w20	**81.26**	0.6,0.2,0.2	91.41	–	0.3,0.5,0.5

先の記事のタイトル（マクドナルド）を多義語 (マック) の正解語義と見なすことができる．

自動獲得した多義語 3,871 語からランダムに 90 語選択し，半分の 45 語をパラメータ（確信度のしきい値および語義スコア計算時の α）推定用，残りを評価用の多義語とした．90 語に関して，各多義語が出現している記事を 30 件ずつをウィキペディアよりランダムに抽出し，1,350 記事からなるパラメータ推定用セットと評価セットを作成した．

5.2 実験結果

従来の文脈幅を固定する手法と提案手法を比較した結果を表 1 に示す．表より文脈を固定する手法と比べ，語義確率を用いない場合において提案手法の精度が高いことがわかる．

提案手法により改善/改悪した例を図 4 に示す．図中 (A) の多義語「リンク」を含む文書は，文脈を拡張することで正解となった例である.「リンク」にはウェブリンクに加えゲームキャラクターの語義がある．この文書の場合，window 幅 5 で考慮される「ゲーム」「サイト」では十分な確信度が得られなかったため文脈が拡張されていった．その結果，window 幅 20 の時に「メンテナンス」や「更新」などの語を文脈ベクトルに考慮できるようになり，高い確信度をもって正解語義（ウェブリンク）が出力された．

(B) は広い文脈を考慮することで文脈が不明瞭となり誤る例である．この文書に含まれる多義語「アルゴス」には，ギリシア神話に登場する魔神とギリシャの都市名の語義がある．window 幅 20 では「口」「巣」などの語が考慮され文脈が不明瞭となり不正解となる．しかしながら，提案手法は「アルゴス」の係り元「住む」を考慮した時点で十分な確信度が得られたため正解語義（都市）を出力できた．

一方，(C) は提案手法で改悪した例である．多義語「ヤシ」には植物に加え都市名の語義があり，その語義確率は植物が 0.79, 都市名が 0.21 である．この文書の場合，多義語の係り先である「市」を文脈として考慮することで都市名の語義ベクトルと文脈ベクトルの類似度は大きな値 (0.743) をとる．しかしながら，語義スコアとしてみると，語義確率の影響を受けてしまい小さな値になってしまう．このため，十分な確信度を得ることができず文脈が拡張され,「集団」「軍」などが文脈ベクトルに考慮されて文脈が不明瞭となり，誤った語義を出力していた．そのため，語義確率の利用方法について検討する必要がある．

正解となった文書:
(A) ... 正しくゲームをするページで，本サイトはそこから**リンク**している．サイト中央には同名のゲーム作成ツールが設置されている．最後に更新は行われていないが，不定期にメンテナンスが ... (正解語義：ウェブリンク)

(B) ... ヘーラクレースは口と鼻を布で覆いながらヒュドラーの住む**アルゴス**近くのレルネーの沼地へとやってきた．そのヒュドラーの巣に火矢を打ち込み ... (正解語義：都市)

不正解となった文書:
(C) ... 記載されている．東ルーマニア (ベッサラピア，**ヤシ**市など) にいたユダヤ人は，集団的な ... 鉄衛軍が扇動してクーデターを起こしたが ... (正解語義：都市)

図 4: 文脈の段階的拡張により正解/不正解となる例

6 おわりに

本稿では文脈を段階的に拡張する多義性解消手法を提案し，ウィキペディアから構築した評価セットを用いて，その有効性を示した．今後は語義確率の利用方法を検討する予定である．

参考文献

[1] Ping Chen, Wei Ding, Chris Bowes, and David Brown. A fully unsupervised word sense disambiguation method using dependency knowledge. *The 2009 Annual Conference of the North American Chapter of the ACL (NAACL 2009)*, pp. 28–36, 2009.

[2] Olena Medelyan, Milen, and Ian H. Witten. Topic indexing with wikipedia. In *Proceedings of the AAAI 2008 Workshop on Wikipedia and Artificial Intelligence (WIKIAI 2008)*, 2008.

[3] David Milne and Ian H. Witten. Learning to link with wikipedia. In *Proceedings of the 16th ACM Conference on Information and Knowledge management (CIKM 2008)*, 2008.

[4] Hwee Tou Ng and Hian Beng Lee. Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In *Proceedings of the 34th annual meeting of the Association for Computational Linguistics*, pp. 40–47, 1996.

[5] Christopher Stokoe, Michael P. Oakes, and John Tait. Word sense disambiguation in information retrieval revisited. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pp. 40–47, 2003.