C-12

Wikipedia における文書トピックを利用した語義曖昧性解消手法

A Word Sense Disambiguation Method by Using Topics of Documents in Wikipedia

道下智之 †　中山浩太郎 ‡　原　隆浩 †　西尾章治郎 †
Tomoyuki Michishita　Kotaro Nakayama　Takahiro Hara　Shojiro Nishio

1　はじめに

近年，Wikipedia が語義曖昧性解消のためのコーパスとして注目されている．これは，Wikipedia では一つの記事で一つの概念を記述しており，各記事は記事タイトルによって一意に識別されるという特徴を持っているためである．曖昧な語の例として「Spring」について考える．この語は，季節の春，機械のバネ，泉・湧水といった語義が存在するが，Wikipedia では「Spring_(season)」，「Spring_(device)」，「Spring_(hydrosphere)」といった別々の記事で各々の概念が記述されている．また，Wikipedia には曖昧な語に対するユーザの情報要求をナビゲートする Disambiguation ページ（曖昧さ回避ページ）が存在する．Disambiguation ページは，{{disambi}} という特別なタグが付与されている記事であり，記事タイトルである曖昧な語の示す概念に対してリンクが張られている．この特徴により，Disambiguation ページから曖昧な語とその語義を抽出することができる．

情報検索（特に Web 検索）の分野では，新しい概念やドメインに特化した概念など多様なクエリが発行される．また，検索システムは人名，地名，組織名，頭字語などの固有名詞や楽曲・映画のタイトルといった曖昧性の高い検索語もクエリとして発行される [5]．これらの語は，一般の辞書や WordNet では網羅されていないため，新しい概念や専門分野，さらに固有名詞や連語を広く網羅した新しいリソースが必要である．筆者らはこの新しいリソースとして即時性や網羅性の観点から Wikipeida が有用であると考えている．

本稿では，Wikipedia から得られた学習データを用いて，従来の語義曖昧性解消手法で用いられる素性に加え，曖昧な語の出現する文とその前後の文，さらに文書それぞれのトピックと語義との関連度を素性として利用することによって語義曖昧性解消の精度が向上することを示す．さらに提案手法を情報検索に適応することで情報検索性能が向上することを示す．

2　関連研究

これまで語義曖昧性解消の研究は数多く行われてきたが，最近は Wikipedia を用いた語義曖昧性解消に関する研究が注目を集めている．

Mihalcea[3] は，Wikipedia の記事中のリンクを語義タグとして扱うことによって，Wikipedia が語義曖昧性解消のための学習コーパスとして利用可能であることを示している．Wikipedia の記事中のリンクは，アンカーテキストが曖昧な語であっても，リンク先はコンテキストに沿った概念が参照されている．例えば，曖昧な語「Spring」がアンカーテキストとなっている「A spring is an elastic object used to store mechanical energy.」という一文を考える．この「Spring」は機械部品の意味であり，Wikipedia の記事では「Spring_(device)」にリンクが張られている．このように，アンカーテキストが曖昧な語であるとき，そのリンクは語義が定義された一種のタグとみなすことができる．この研究では，Wikipedia を語義タグ付き学習コーパスとして利用することの有用性を示すために，学習データから得られるコンテキスト情報を単純ベイズ分類器に学習させ，SENSEVAL-2 と SENSEVAL-3*のテスト・コレクションを用いて評価している．結果として，SENSEVAL のコーパスで学習した分類器よりも，Wikipedia から作成したコーパスで学習した分類器のほうが高い精度が得られている．Mihalcea は，Wikipedia で定義されている概念のほとんどが名詞であることやリンク先の概念が誤っていること，さらに語義の出現頻度分布が歪んでいるといった問題点はあるが，大量の学習データを得られことは有益であると述べている．

Cucerzan[1] は，Disambiguation ページから得られる概念を曖昧な語の語義候補とし，それらの概念に関するコンテキスト情報やカテゴリタグ情報を用いて，与えられた文書に出現する固有表現の曖昧性を解消する手法を提案している．さらに，ニュース記事とウィキペディア記事に対して評価実験を行い，提案手法が高い精度で語義曖昧性解消を行うことを示している．

本稿では，Mihalcea が有用性を示した語義タグ付き学習コーパスを利用して，コンテキスト情報以外に曖昧な語の出現する文やその前後の文のトピック，さらに文書のトピックと語義との関連度を素性として利用する．このトピックと語義との関連度計算には，Cucerzan と同様に概念を記述した記事からコンテキスト情報を抽出して算出する．

3　提案手法

3.1　語義の選定

Wikipedia を学習データとして用いるとき，語義の選定を行わずに学習を行うと情報検索においてユーザが要求しない語義が多く含まれてしまう．表 1 は，Wikipedia

† 大阪大学，Osaka University
‡ 東京大学，Tokyo University

*http://www.senseval.org/

表 1: アンカーテキストが「Music」であるリンクの参照先概念（選定前）

番号	概念	出現回数
1	Music	1204
2	Music_of_the_United_States	176
3	Christian_music	150
4	Music_of_Brazil	119
⋮	⋮	⋮
120	1963_in_music	10
121	1964_in_music	10
122	1966_in_music	10
123	1984_in_music	10
124	Music_(Carole_King_album)	10

表 2: アンカーテキストが「Music」であるリンクの参照先概念（選定後）

番号	概念	出現回数
1	Music	1204
2	Music_(Madonna_song)	104
3	Music_(Madonna_album)	58
4	Music_(311_album)	14
5	Music_(Carole_King_album)	10

記事において「Music」がアンカーテキストであるリンクの参照先概念を集計した結果である．ただし，参照回数が 10 回未満の概念は，ほとんど出現しない，または誤ってリンクが張られたと考えて除去した．表 1 からわかるように，参照された概念数は 124 もある．しかし，情報検索において「Music」というクエリから「Music of the United States」,「Music of Brazil」といった国ごとの音楽を検索することはほとんどないと思われる．また，年代も同様である．このことから，Wikipedia を学習データとして用いる場合にはユーザの情報要求を考慮する必要がある．

この問題を解決するために，ユーザの情報要求をナビゲートしている Disambiguation ページを利用して，リンク先の概念を選択することで語義の選定を行う．選定は，Disambiguation ページでリンクされている概念以外を学習データとして使わないことで行う．選定の結果が表 2 である．選定後の概念は，一般の音楽の意味である「Music」に加え，曲名やアルバム名である．これら全ては,「Music」というクエリからユーザが要求する可能性があり，選定は情報検索に適応する上で有益であると考えられる．

3.2 語義曖昧性解消

古くから研究されている語義曖昧性解消は，コンテキストによって意味の変わる曖昧な語の意味を決定するタスクである．このため，多くの手法は曖昧な語の前後の語や品詞，さらに共起回数の高い単語を学習器の素性として利用する．しかし，Web 検索文書では画像のタイトルや表の値など曖昧な語が文として出現しないため，有益な素性を得られないことが多い．さらに，前後の文や文書の全体からでなければその意味を推測できない曖昧な語も多く存在する．

そこで，提案手法では語義と曖昧な語の出現する前後の文や文書で述べられているトピックとの関連度を素性として利用することで，Web 検索文書に対しても高い精度で曖昧性解消できる手法を提案する．関連度は，語義ベクトルと文または文書ベクトルのコサイン類似度で算出する．ベクトルは，テキストを語の集合とみなし，その語を各ベクトルの要素として重み付けすることで算出する．重み付けは，tf-idf をベースとして以下の式で行う．

$$w(t,T) = \sqrt{tf(t,T)}\log(\frac{DocNum+1}{df(t)})$$

$t(\in T)$ は語，T はテキストに含まれる語の集合，$w(t,T)$ はベクトルの要素である語 t の重み，$DocNum$ は Wikipedia の記事数，tf(t, T) は T における語 t の出現回数，$df(t)$ は語 t が出現する Wikipedia の記事数である．

語義のテキストは，その概念を記述している記事から抽出する．このとき，出現する Wikipedia の記事数が 10 回未満の語（$df(t) < 10$）はベクトルの要素として含めないようにした．

提案手法では，分類器として SVM(Support Vector Machine) を用いて，既存の語義曖昧性解消で用いられる素性に加えて語義とトピックとの関連度を素性として学習を行った．

既存の素性としては，コンテキスト情報の局所的な素性として曖昧な語とその品詞，曖昧な語の前後 1 語，曖昧な語の前後 3 語の品詞，曖昧な語の前後に出現する 2 または 3 語の連語（bi-gram，tri-gram），曖昧な語の前後に出現する名詞・動詞を利用した．さらに大域的素性として曖昧な語の各語義における共起回数が最も多い 5 語（3 回以上共起）を用いた．これらの素性は，語義曖昧性解消の既存研究において有効性が確認されているものである [2, 4].

語義とトピックの関連度の素性は，曖昧な語の各語義と曖昧な語が出現した文およびその前後の 3 文それぞれとの関連度（素性の数は語義数× 7），および曖昧な語の各語義と文書との関連度（素性の数は語義数）とする．

提案手法では，曖昧な語の大文字・小文字はすべて区別するようにした．これは，曖昧な語の大文字・小文字といった情報によって語義曖昧性解消が行われてしまうためである．例えば,「apple」は小文字である場合はほとんどがリンゴの意味である．このように，大文字・小文字の情報だけで曖昧性解消の精度が向上しないように大文字・小文字は区別して曖昧性解消器を生成した．

表 3: Wikipedia の曖昧な語の語義の頻度分布エントロピー

曖昧な語の数	9326
$E(w) > 0.5$	7868
$E(w) > 1.0$	3000
$E(w) > 1.5$	1426
$E(w) > 2.0$	340
$E(w) > 2.5$	70
$E(w) > 3.0$	6

4 実験・結果

提案手法の評価のために三つの実験を行った．

一つ目の実験は，学習コーパスの曖昧性に関する評価である．これは，Wikipedia の曖昧な語が語義の選定を行っても高い曖昧性があるかを調べることを目的とする．このとき，学習データの語義頻度が 10 未満の語義は除外した．評価指標としては，語義の頻度分布エントロピー $E(w)$ を用いる．

$$E(w) = -\sum_i p(s_i|w) \log p(s_i|w)$$

$p(s_i|w)$ は，単語 w が語義 s_i として出現する確率であり，$E(w)$ が大きいほど語義の頻度分布が一様であり，曖昧性が高いと考えられる．逆に，$E(w)$ が小さいほど，語義の頻度分布に偏りがあるため，曖昧性が低いと考えられる．

表 3 が，語義頻度分布のエントロピーの結果である．ここで，エントロピーが 1.5 以上の語は一般的に多義語として思われている曖昧性の高い語が多く含まれていた．このことから，Wikipedia から得られた曖昧な語は曖昧性の高いものが多く含まれていると考えられる．

二つ目の実験は，提案した語義曖昧性解消の精度に関する実験である．語義頻度分布のエントロピーが 1.5 以上の曖昧な語 20 語に対して提案手法の精度を評価した．語義曖昧性解消の精度は，Wikipedia から得られた学習データに対して 10-fold cross validation*を用いて算出する．最も出現回数が多い語義を選んだときの精度である MFS(Most Frequent Sense)，提案手法を用いずに学習した手法 Default，Default の素性に加えて曖昧な語が出現した文およびその前後 3 文と語義との関連度を素性に学習した手法 Sentences，Default の素性に加えて文書のトピックと語義との関連度を素性に学習した手法 Document，そしてすべての素性を学習に用いた提案手法 All を比較した．

表 4 が，語義曖昧性解消の実験結果である．エントロピーが 1.5 以上という曖昧性の高い語を選択したため，語義数が多く，MFS が小さいという傾向であった．既存の語義曖昧性解消手法である Default と提案手法である All をみると，すべての曖昧な語において精度向上がみられ，平均として 14.4821%精度が向上した．特に，学習データが少ない「Cookie」,「Offspring」,「Typhoon」,「Zip」の 4 語において精度が平均 23.7311%向上したことから，提案手法は既存手法よりも少ない学習データで語義曖昧性解消の精度を出すことができることを示している．また，比較的精度が低かった「development」を調べると，語義となる概念「Developmental_biology」や「Child_development」,「International_development」,「Economic_development」といった概念に対する分類が失敗していた．これは，語義となる概念から抽出したテキスト中の語が似ているため，トピックとの関連度の値が似通ってしまったことが原因ではないかと考えられる．

表 5: 情報検索の平均精度

	amazon	apple	jaguar
提案手法なし	17.1429	20.0000	14.2857
提案手法あり	35.7143	24.0000	25.7143

また，Sentences と Document を比較すると Sentences のほうが平均 3.7552%高い精度であったことから，曖昧な語の前後の文のトピックのほうが曖昧性解消に有益であることがわかる．All と Sentences を比較すると，文書との関連度の素性を加えることで 2.4306%精度が向上している．このことから文書の関連度も有益な情報であるといえる．

三つ目の実験として，提案手法の Web 検索での有効性を検証するため，提案手法を用いて Web 検索結果の分類を行った．Web 検索結果は，Yahoo! Search BOSS†を用いた．クエリとして利用した曖昧な語は，二つ目の実験で用いた「amazon」,「apple」,「jaguar」の 3 語である．クエリを発行し，検索された文書中の曖昧な語に対して語義曖昧性解消を行い，使われていた語義にその文書を分類した．このとき，検索は大文字・小文字が無視されるので，大文字・小文字を無視して文字列にマッチするすべての曖昧な語に対して語義曖昧性解消を行った．

評価指標として，検索結果の上位 10 件を対象に各語義で検索したときの精度の平均を用いた．例えば,「Apple」を検索語としたときの検索結果が「Apple_Inc.」に関する文書が 7 件，リンゴの意味の「Apple」が 3 件,「Apple_Records」,「Apple_(album)」,「Apple_Corps」が 0 件であったとき，この平均精度は，$(0.7 + 0.3 + 0.0 + 0.0 + 0.0)/5 * 100 = 20(\%)$ となる．

表 5 が，実験結果である．各曖昧な語とも，提案手法により平均精度が向上していることがわかる．特に，「amazon」では，語義である「Amazon.com」,「Amazon River」,「Amazon Basin」,「Amazon Rainforest」の分類精

*k-fold cross validation では，学習データを k 分割する．そのうち一つを評価データとし，残りを学習データとして語義曖昧性解消の精度を算出する．k 個に分割したデータそれぞれを評価データとして精度を求めることによって，語義曖昧性解消の平均精度を算出する．

†http://developer.yahoo.com/search/boss/

表 4: 提案手法による語義曖昧性解消の精度比較

曖昧な語	語義数	学習データ数	エントロピー	MFS	Default	Sentences	Document	All
Amazon	7	843	2.3983	28.9442	68.8019	73.6655	72.0047	**75.9193**
Apple	5	866	1.506	63.0485	78.5219	86.3741	82.9099	**86.9515**
Cookie	3	49	1.5669	40.8163	57.1429	71.4286	59.1837	**79.5918**
development	8	302	2.8164	26.1589	49.3377	63.245	58.6093	**63.9073**
Jaguar	7	1283	1.93	38.5035	81.9174	89.7896	88.3087	**90.569**
King	11	662	2.5478	50	80.5136	89.2749	87.1601	**90.1813**
Magic	10	289	3.0052	30.1038	59.5156	77.5087	74.3945	**80.6228**
Mercury	11	2671	2.5387	36.9899	78.6222	90.2658	85.0618	**92.1003**
Offspring	3	43	1.5702	39.5349	76.7442	88.3721	79.0698	**90.6977**
Phoenix	21	1285	3.7227	17.5875	61.6342	77.7432	81.7899	**85.1362**
Ruby	5	162	2.0796	43.8272	60.4938	77.1605	75.3086	**82.716**
Space	9	601	2.2282	49.4176	74.7088	80.5324	81.8636	**86.0233**
Spirit	8	504	2.3863	39.0873	84.5238	91.8651	87.8968	**93.254**
State	9	8229	1.9634	49.1797	58.6463	98.3109	85.0407	**98.2622**
state	10	8001	1.954	54.1057	91.2886	**92.8509**	91.751	92.9759
string	5	367	1.6982	43.0518	78.2016	84.4687	85.5586	**87.4659**
Taxi	6	365	1.9297	55.8904	75.3425	80.274	78.3562	**82.4658**
TNT	5	1154	1.5642	53.0329	86.7418	94.1941	92.2877	**95.6672**
Typhoon	3	70	1.5439	41.4286	64.2857	80	74.2857	**81.4286**
Zip	4	58	1.9783	31.0345	65.5172	86.2069	77.5862	**86.2069**
AVERAGE	7.5	1390.2	2.1464	41.5872	71.6251	83.6766	79.9214	**86.1072**

度が高く，それぞれ上位 10 件に適合文書が増えたため精度が大きく向上した．逆に,「apple」は,「Apple Inc.」の意味で使われている「Apple」をリンゴの意味で使われていると分類する誤りが多かった．また，上位の検索文書にリンゴの意味で使われている文書が出てこないことも精度が上がらなかった理由である．また,「jaguar」では，車の意味とネコ科の動物の意味が高精度に分類されていたことに加え，ゲーム機の「Atari Jaguar」（従来の検索結果では 245 位），ギターの「Fender Jaguar」（従来の検索結果では 195 位）といった従来の検索ランクが低い語義の文書も分類することができた．Wikipedia は，このようなゲーム機やギターなどあらゆる分野の概念を網羅していることから，従来の検索システムでは検索することが困難な概念を検索するために有益なリソースであると考えられる．

5 まとめ・今後の課題

本稿では，新しい概念，ドメイン特化した概念および固有名詞や連語を網羅している新しいリソースである Wikipedia を学習コーパスとして用いて語義とトピックとの関連度を学習することにより，語義曖昧性解消の精度が向上することを示した．さらに，提案手法を情報検索に適応することによって検索の平均精度を向上させるのに加え，従来は検索することが困難な概念を検索することができた．

今後の課題は,「development」のように語義間が類似した場合においても明確な値の差異がでるような関連度計算手法の提案，および提案手法を情報検索へ適応したときの評価内容の充実である．

謝辞 本研究の一部は，科学研究費補助金基盤研究 B(21300032)，C(20500093)，およびマイクロソフト産学連携研究機構 CORE 連携研究プロジェクトの助成によるものである．ここに記して謝意を表す．

参考文献

[1] Cucerzan, S.: Large-scale named entity disambiguation based on Wikipedia data, *Proceedings of EMNLP-CoNLL*, pp. 708–716 (2007).

[2] Lee, Y. and Ng, H.: An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation, *Proceedings of ACL-02*, pp. 41–48 (2002).

[3] Mihalcea, R.: Using wikipedia for automatic word sense disambiguation, *Proceedings of NAACL HLT*, pp. 196–203 (2007).

[4] Ng, H. and Lee, H.: Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach, *Proceedings of ACL-96*, pp. 40–47 (1996).

[5] Sanderson, M.: Ambiguous queries: test collections need more sense, *Proceedings of ACM SIGIR*, pp. 499–506 (2008).