文書内における単語の共起を利用した上位下位概念の推定

平野 正徳[1] 坂地 泰紀[2] 木村 笙子[3] 和泉 潔[2] 松島 裕康[2] 長尾 慎太郎[3] 加藤 惇雄[4]
[1]東京大学 工学部 [2]東京大学 工学系研究科
[3]大和証券投資信託委託株式会社 [4]株式会社大和総研
[1]hirano@g.ecc.u-tokyo.ac.jp
[2]{sakaji, izumi, matsushima}@sys.t.u-tokyo.ac.jp
[3]{shoko.kimura, nagao.s}@daiwa-am.co.jp [4]atsuo.kato@dir.co.jp

1 はじめに

上位下位概念の抽出は自然言語処理において重要なタスクとなっている．大きく分けて二つのアプローチがある．一つは X such as Y などといった，文中で共起している単語の包摂関係を示すようなパターンを使用した研究 [2] から始まり，そこからブートストラップ的にパターンを見つけていく研究 [9] などがある．Word2vec[6] の登場以降は，もう一つのアプローチとして，単語の特徴ベクトルを組み込んだ研究が行われている [5]．さらに，それらを融合し，機械学習を取り入れた方法として，HypeNET[8] などがある．

上位下位概念の抽出を含む，オントロジー構築に関する研究は様々な形で行われている．日本語に関しては,「森羅：Wikipedia 構造化プロジェクト 2018」[7] などがあげられる．英語などを対象とした例として，YAGO[10] などの手法がある．しかし，YAGO は Wikipedia のカテゴリー構造に依存しており，汎用性が低い．無論，Wikipedia はオープンなリソースで，様々な言語のデータがあるため，非常に便利ではあるが，特定のドメインに特化したオントロジーを作成しようとすると障害がある．

上記の手法の多くはテキストを学習させるなどして行われるため，学習量の少ない場合や新語の足りない場合，特定のドメインに絞ったドキュメントを使用する場合には精度に限界があると考えられる．我々が提案したスキーム [3] の一部を拡張すると，上位下位概念の抽出において，学習量に依存が少なく，かつ，新語や特定のドメインに絞った場合などにも適用可能であると可能性が高い．そこで，本論文内では，新たな上位下位概念抽出の手法を提案し，その有用性について議論する．

本論文の内容や意見は執筆者個人に属し，いかなる組織の公式見解を示すものではありません．連絡先:〒 113-8656 文京区本郷 7-3-1 東京大学工学部システム創成学科 和泉・坂地研究室 平野正徳 HP: https://mhirano.jp/

2 提案手法

任意の語、又は、テーマに関して記述された文書を用意する．Wikipedia であれば，特定の言葉や事象に関する記事それぞれを一つの文書として利用可能である．企業情報を利用するのであれば，一つの企業に関する説明や資料をひとまとめにしたものを一つの文書として利用することが可能である．ここで，これらの文書を $doc_1, doc_2, \cdots, doc_N$ とラベリングする．

次に，上位下位概念の関係にあるかどうかを知りたい単語ペアを用意する．これを $word_A, word_B$ とする．

さらに，$word_A, word_B$ を含む文書を取り出し，$word_A$ を含む文書群を set_A，$word_B$ を含む文書群を set_B とする．この時，次式のように定義する．

$$P_{AB} = \frac{|set_A \cap set_B|}{|set_A|}, R_{AB} = \frac{|set_A \cap set_B|}{|set_B|} \quad (1)$$

$word_A$ が $word_B$ の下位概念である場合，多くの文書において，$word_A$ の説明として，$word_B$ が文書内に同時に登場する可能性が高いという仮定をおけば，$P_{AB} > R_{AB}$ が成立すると予想される．逆の場合も然りである．そこで，閾値を設定して，以下のように定義し，分類する．閾値を thr と表記する．

1. $P_{AB} < thr$ かつ $R_{AB} < thr$ の時：$word_A, word_B$ 間には上位下位概念がないとする．
2. $P_{AB} \geq thr$ かつ $R_{AB} < thr$ の時：$word_A$ が $word_B$ の下位概念とする．
3. $P_{AB} < thr$ かつ $R_{AB} \geq thr$ の時：$word_A$ が $word_B$ の上位概念とする．

4. $P_{AB} \geq thr$ かつ $R_{AB} \geq thr$ の時：

   (a) $P_{AB} > R_{AB}$ の時 $word_A$ が $word_B$ の下位概念とする．

   (b) $P_{AB} < R_{AB}$ の時 $word_A$ が $word_B$ の上位概念とする．

   (c) $P_{AB} = R_{AB}$ の時 $word_A$ と $word_B$ は同等の概念とする．

3 実験

提案手法の精度を検証するために，実験を行なった．

今回は，文書として，日本語版 Wikipedia の記事[1]を利用した．バージョンとしては，02-Oct-2018 13:53 を利用した．Wikipedia のそれぞれの記事を $doc_1, doc_2, \cdots, doc_N$ とし，記事タイトルも文書の一部とした．なお，$N = 1,122,013$ であった．

さらに，上位下位概念の正解データ，評価データを作成する必要があるため，その目的のために，日本語版 Wordnet[4][2]を利用した．日本語版 Wordnet の仕様は基本的に Wordnet[1] と同様である．この日本語版 Wordnet には 57,238 概念 (synset 数)，93,834 語，158058 語義 (synset と単語のペア)，135,692 定義文，48,276 例文が含まれている．

この日本語版 Wordnet の中から，概念 (synset) 同士の関連のうち，上位下位概念のセットを取り出し，その概念に対応する単語 (日本語) を取り出してきてペアにすることで，Wordnet に基づいた上位下位概念のある日本語の単語ペア 32,085 ペアを抽出した．(実際に単語ペアは 37,113 ペア存在したが，Wikipedia の文書に一度も含まれない単語などは除外した．)

さらに，Wordnet に含まれている日本語の名詞 65,788 個のうち，ランダムで 2 つ語を選び，擬似的な上位下位概念のない日本語の単語ペア 32,085 ペアを生成した．(厳密にはランダムで生成しているので上位下位概念のあるペアが含まれている可能性は若干あるが，以下では上位下位概念のないペアと呼ぶ．また，Wikipedia の文書に一度も含まれない単語などは同様にこちらでも削除して，代わりのペアを生成した．) こうしてできた，上位下位概念のある単語ペア 32,085 ペアと上位下位概念のない単語ペア 32,085 ペアのうち，上位下位概念のある単語ペア 1,000 ペアと上位下位概念のない単語ペア 1,000 ペアの合計 2,000 ペアをパラメータ thr のチューニング用に使用し，残り 31,085 ペアずつ，合計 62,170 ペアを評価データとして使用した．

結果の評価，またはパラメータ thr のチューニングには精度評価を利用した．精度評価の方法は以下の通りである．まず，上位下位概念のある単語のうち，$word_A$ が上位，$word_B$ が下位になるように整理した．以下では，$word_A$ からみて $word_B$ が下位概念である時に，“down”，上位概念である時に “up”，同等の概念である時に “even”，上位下位概念の関係にない時を “not” と表記する．つまり，上位下位概念のある単語ペアにおいては，“down” と判定されるのが正解であり，上位下位概念のない単語ペアにおいては，“not” と判定されるのが正解である．これらの条件下で，結果は表 1 のように分類されるはずである．表 1 において，正解であるのは C と H である．これに基づいて精度計算を行う．表 2 は結果の正誤に基づく分類である．TP, FN, FP, TN はそれぞれ True Positive, False Negative, False Positive, True Negative である．これに基づき，それぞれのケースに対して Precision, Recall, F1 を計算した．さらに，その 2 つの F1 に対して，算術平均を取ることで，Macro-f1 を計算した．この Macro-f1 を最終的な評価値として利用した．

表 1: 結果の分類

	Pred.			
	up	even	down	not
上位下位概念あり	A	B	C	D
上位下位概念なし	E	F	G	H

表 2: 結果の正誤に基づいた分類

上位下位	TP	FN	FP	TN
あり	C	$A+B+D$	G	$E+F+H$
なし	H	$E+F+G$	D	$A+B+C$

4 結果

4.1 テストデータにおける結果とパラメータチューニング

まず，thr を 0 から 1 で 0.001 ずつ変化させた結果を提示する．図 1 は thr を変化させていった場合の，上位下位概念あり単語ペア 1,000 セットの評価結果で

[1] Available at https://dumps.wikimedia.org/jawiki/latest/.

[2] Available at http://compling.hss.ntu.edu.sg/wnja/.

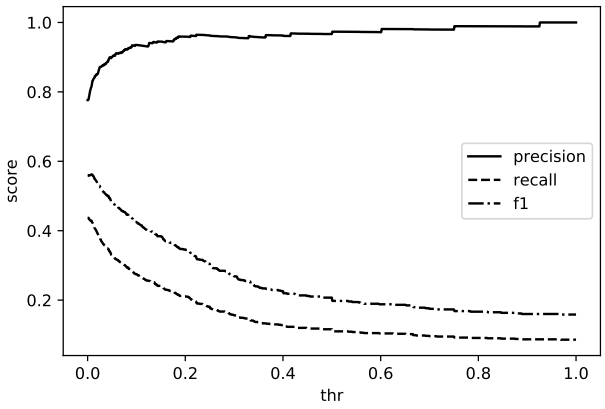

図 1: *thr* を変化させた場合の上位下位概念ありのペアでの Presicion, Recall, F1

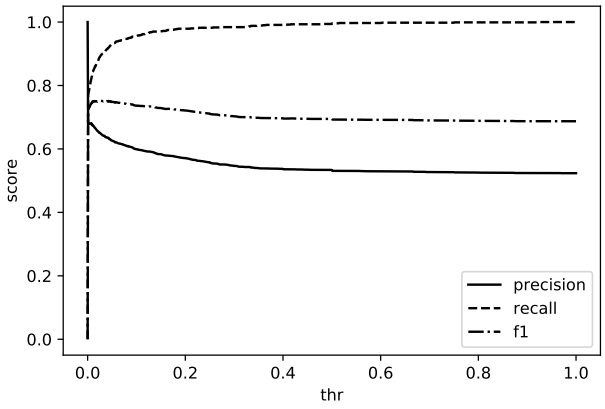

図 2: *thr* を変化させた場合の上位下位概念のない単語ペアでの Presicion, Recall, F1

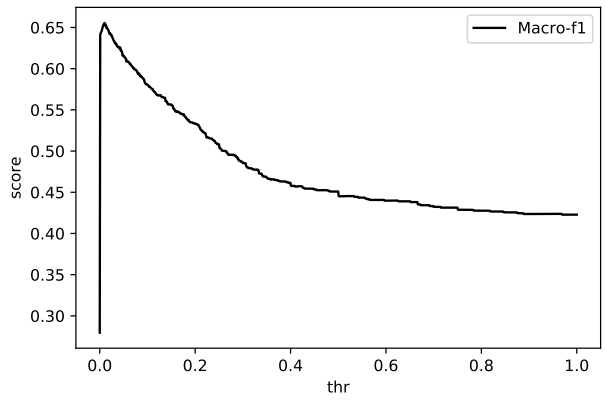

図 3: *thr* を変化させた場合の Macro-f1 の変化

ある．*thr* を引き上げていくにつれて Precision が向上し，Recall が減少する傾向にある．図 2 は *thr* を変化させていった場合の，上位下位概念のない単語ペア 1,000 セットの評価結果である．こちらは，*thr* を引き上げていくと，Recall が上昇し，Precision が減少する傾向にある．最後に図 3 だが，これは *thr* を変化させた時の Macro-f1，つまり，図 1 と図 2 における F1 の平均の変化である．この結果によると，Macro-f1 は *thr* の変化に対して，極大点を持つことがわかる．そこで，さらに，*thr* の変化の刻みを 0.00001 として，極大点を探索したところ，極大点における *thr* は 0.00991 であり，その時の Macro-f1 は 0.65556 であるとわかった．この極大点における *thr* を *thr* の最適値として採用した．

4.2 評価

次に，上記で定めた $thr = 0.00991$ を利用して，評価用データ 62,170 ペア (上位下位概念あり 31,085 ペア，上位下位概念なし 31,085 ペア) に対して，実験を行なった．表 3 は評価データに対する分類結果を示している．上位下位概念ありペアにおいて，正しく “down” に分類されたのは 13,722 ペアで，また，11,533 ペアが “not”，つまり上位下位概念なしと判定されている．一方，上位下位概念のない単語ペアについては，大半が “not”，つまり上位下位概念なしと判定されている．これらの結果に基づき，各種精度を計算したところ，表 4 の通りになった．表 4 にある通り，上位下位概念ありと判定された場合 (今回の場合，$word_A$ が上位概念で $word_B$ が下位概念と判定された場合，つまり “down” と判定された場合)，その適合率は 84%ほどとなっている．一方で，上位下位概念ありのペアのうち，正しく上位下位概念ありと判定されたのは 44%であ

表 3: 評価用データに対する分類結果

上位下位概念	Pred.			
	up	even	down	not
あり	5,810	20	13,722	11,533
なし	2,509	0	2,593	25,983

表 4: 評価用データに対する分類結果

上位下位概念	Precision	Recall	F1
あり	0.84107	0.44143	0.57899
なし	0.69258	0.83587	0.75751
		Macro-f1:	0.66825

り，あまり良くない結果となっている．また，擬似的に上位下位概念の関係がないものとして作成したペアにおいて，正しく上位下位概念なしと判定されたのが同様に 84%程度であった．しかし，上位下位概念なしと判定されている中で本当に上位下位概念なしで合ったものは 69%ほどであった．

5 考察

今回の実験において，thr を変化させると明確な極大点を持っており，パラメータのチューニング方法は充分であったと考えられる．

しかし，実験結果の精度については議論の余地がある．直接比較はできないが，HypeNET[8] の lexical split における F1 は 0.700 であり，それと比べると，精度は低いと考えられる．直接比較する必要があるため，これに関しては，今後の検討課題であると考える．

今回提案しているモデルはチューニングすべきパラメータは 1 つであるという特徴があり，今回は，テストデータ 2,000 に対して，評価データ 62,170 としたが，評価データとテストデータでは精度に大きな差は存在しなかった．そのため，非常に少ない訓練データでチューニング可能であるということも特徴であると考えられる．また，P_{AB} などのみを使用する場合などには教師なしで使用することができることも非常に大きな特徴であると考える．

さらに，今回は手軽に手に入る Wikipedia の記事を使用したが，実際には他のテキストでも代用可能である．我々の以前の研究 [3] では，$doc_1, doc_2, \cdots, doc_N$ に各企業の決算短信などの企業情報を使用しており，それぞれの分野に合わせて特有の文書を使うことが可能になっており，そうすることで，適用先の分野にあった結果になることが予想される．

今後の展望としては，上で述べた評価における直接比較のほか，第 2 節の提案手法における分類の 4.(a)-(c) のアップデートが必要であると考える．同等の概念だった場合でも，$P_{AB} = R_{AB}$ となるわけではなく，多少の誤差が発生すると考えられるからである．

6 おわりに

本研究において，文書内における単語の共起を利用した上位下位概念の推定の手法を提案した．本手法に基づき，Wikipedia の記事を文書として使い，Wordnet に含まれる上位下位概念をデータとして実験を行った．その結果，精度は低いものの，一定の有効性を確認することができた．本手法は非常に少ない訓練データで必要なパラメータのチューニングが可能であることもわかった．

参考文献

[1] Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. The MIT Press, 1998.

[2] Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics -*, Vol. 2, p. 539, Morristown, NJ, USA, 1992. Association for Computational Linguistics.

[3] Masanori Hirano, Hiroki Sakaji, Shoko Kimura, Kiyoshi Izumi, Hiroyasu Matsushima, Shintaro Nagao, and Atsuo Kato. Selection of Related Stocks using Financial Text Mining. *18th IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 191–198, 2018.

[4] Hitoshi Isahara, Francis Bond, Kiyotaka Uchimoto, Masao Utiyama, and Kyoko Kanzaki. Development of the Japanese WordNet. *Proceedings of the 6th International Conference on Language Resources and Evaluation*, pp. 2420–2423, 2008.

[5] Maria Karyaeva, Pavel Braslavski, and Yury Kiselev. Extraction of Hypernyms from Dictionaries with a Little Help from Word Embeddings. In *Analysis of Images, Social Networks and Texts*, pp. 76–87. 2018.

[6] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *Proceedings of the International Conference on Learning Representations (ICLR)*, pp. 1–12, 2013.

[7] Sekine Satoshi, Kobayashi Akio, Ando Maya, Baba Yukino, and Inui Kentaro. Wikipedia 構造化データ「森羅」構築に向けて. 言語処理学会 第 24 回年次大会発表論文集, 2018.

[8] Vered Shwartz, Yoav Goldberg, and Ido Dagan. Improving Hypernymy Detection with an Integrated Path-based and Distributional Method. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 2389–2398, 2016.

[9] Rion Snow, Daniel Jurafsky, and Andrew Y.NG. Learning syntactic patterns for automatic hypernym discovery. In *Advances in Neural Information Processing Systems 17 (NIPS 2004)*, pp. 1297–1304, 2004.

[10] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. YAGO: A Core of Semantic Knowledge Unifying WordNet and Wikipedia. In *Proceedings of the 16th international conference on World Wide Web - WWW '07*, p. 697, New York, New York, USA, 2007. ACM Press.