Wikification における固有表現クラス情報を用いた候補削減

三好 聖子 †* 仲野 友規 ‡ 乾 孝司 ‡

† 筑波大学情報学群情報科学類 ‡ 筑波大学大学院システム情報工学研究科

1 はじめに

テキスト中の言及を実世界のエンティティに関連づけるエンティティリンキング課題のうち，Wikipedia 記事をエンティティとするものを Wikification[1] と呼ぶ．一般に，Wikification は，入力文書からエンティティを指す言及を抽出する処理（言及抽出）と，抽出された言及をそれを説明する Wikipedia 記事にリンクする処理（エンティティ同定）で構成される．また，後半のエンティティ同定は，言及に対するエンティティの候補となる Wikipedia 記事を列挙（候補生成）し，その中からリンクするべき記事を選び出すことで実現される．

Wikification を含め，エンティティリンキングでは，さまざまな理由からエンティティが存在しない（NIL エンティティ）ことがある．例えば，後述するデータ（表 2）では処理対象となる言及のおよそ 25% が NIL エンティティとなっている．エンティティリンキングでは，この NIL エンティティを適切に処理することが求められる．しかし，標準的な処理手続きでは候補生成によって列挙されたエンティティ候補が 1 つ以上ある場合，その中からリンクするべき記事を選び出す処理に焦点があたるため，記事を選ばない処理となる NIL エンティティを正しく同定することは難しい．

このような背景に対し本研究では，固有表現クラス情報に注目することで従来よりも生成する候補を抑制することで NIL エンティティを同定する手法を提案し，評価実験を通してその有効性を検証する．

2 提案手法

2.1 提案手法の概要

提案手法では既存手法で候補を生成した後，言及，および候補となる Wikipedia 記事の固有表現クラス情報に従って候補の保持／削除を決定し，候補の削減をおこなう．図 1 に提案手法の概要を示す．基本的には「言及と同じ固有表現クラスとなる候補のみを保持し，他の候補を削除する」ことをおこなう．ただし，固有表現クラス情報は自動推定で得る情報であり誤りも含まれるため，保持する条件を緩めることも考える．

固有表現クラスとしては，関根の拡張固有表現階層[*1] の第一層から時間／数値表現を除いた 11 クラスを扱う．

- Color, Disease, Event, Facility, God, Location, Name_Other, Natural_Object, Organization, Person, Product

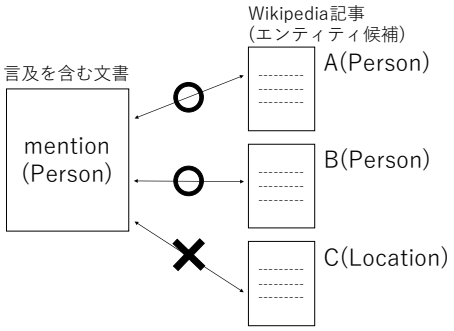

図 1 提案手法の概要図

2.2 候補保持ルール

前節のアイデアを候補保持ルールという形で実現する．候補保持ルールは，言及の固有表現クラス毎に保持する候補の固有表現クラスを規則の条件部として列挙したものであり，この条件にあう候補をそのまま保持し，そうでない候補を削除する．今回は表 1 に示す 2 種類（Tight/Loose）の候補保持ルールを検討する．Tight ルールでは言及と候補の固有表現クラスが厳密に一致する必要がある．一方，Loose ルールは，Tight ルールに加えて，クラス誤りを引き起こしそうな箇所に対して保持クラスを追加することで緩やかな保持条件としている．なお，表中の “NoClass”は，候補の固有表現クラスを自動推定した際にクラス情報が得られなかった場合をあらわす．

表 1 候補保持ルール

言及	候補（Tight）		候補（Loose）
Name_Other	Name_Other	+	Product NoClass
Person	Person	+	NoClass
God	God	+	Product NoClass
Organization	Organization	+	Location Facility Product NoClass
Location	Location	+	Organization Facility Product NoClass
Facility	Facility	+	Organization Location Product NoClass
Product	Product	+	NoClass
Event	Event	+	Product NoClass
Natural_Object	Natural_Object	+	Product NoClass
Disease	Disease	+	Product NoClass
Color	Color	+	Product NoClass

* 代表連絡先：miyoshi@mibel.cs.tsukuba.ac.jp

*1 https://sites.google.com/site/extendednamedentity711/

表 2　日本語 Wikification コーパスの固有表現クラス内訳表

	言及数	NIL エンティティの割合（%）
Color	77	10.4
Disease	213	13.6
Event	958	49.8
Facility	1,106	38.9
God	6	16.7
Location	3,879	4.8
Name_Other	7	85.7
Natural_Object	963	11.8
Organization	3,087	23.7
Person	3,127	36.7
Product	6,557	29.1
全体	19,980	25.2

3　評価実験

3.1　各種の設定

データセットには，Jargalsaikhan ら [2] が構築した日本語 Wikification コーパスを用いる．このコーパスのうち，言及に対する固有表現クラスの推定結果が先述の 11 クラスであり，かつ，正解エンティティとなる Wikipedia 記事が更新作業等によって削除されていないものを評価実験の対象とする．本研究では NIL エンティティの同定に注目しているため，評価対象を NIL エンティティをもつ言及とそれ以外の言及に分割（NIL セットと通常セット）し，それぞれの性能を確認する．表 2 に日本語 Wikification コーパスの固有表現クラスごとの言及数と，そのうち NIL エンティティであるものの割合を示す．

提案手法は事前に生成した候補を絞り込むために使用する．評価実験では以下の手順で事前に候補を生成した．まず，Wikipedia のアンカーテキストに基づいて候補を生成する．Wikipedia のアンカーテキストは [[リンク先の Wikipedia 記事タイトル | アンカーテキスト]] という形式で記述されている．そこで，アンカーテキストを言及文字列，リンク先を候補とすることで候補生成をおこなう．この状態では明らかに不要な候補も多く含まれるため，この結果にさらにリンク頻度に応じた絞込みを適用する．これは，ある言及 m に対して m のリンク先がエンティティ e になる確率 $p(e|m)$ を求め，その値が閾値未満の場合に候補を削除する方法である．閾値は先行研究 [3] に従って 0.05 とする．

言及の固有表現クラスは CRF に基づくチャンキングによって推定した．設定は南ら [7] に従っており，推定性能は F 値で 0.893 である．

- 分類アルゴリズム：CRF
- 素性：文字と前後窓枠 2 の範囲の文字，文字種，品詞
- 学習データ：拡張固有表現タグ付きコーパス [8] から日本語 Wikification コーパスとの重なり箇所を除外したデータ

クラス推定の誤りの中には過抽出となる事例も含まれるため，評価実験で使用するデータ中の NIL エンティティの割合は表 2 よりも増加している．

候補となる Wikipedia 記事の固有表現クラスは SVM に基づく記事分類を実行することで推定した．設定を以下に示す．One vs Rest 方式で固有表現クラスの数だけ分類器を構築し，Rest と出力判定されなかった固有表現クラスすべてをその候補のクラスとする．学習データの交差検定で性能を測定したところ，分類正解率で 0.974 であった．

- 分類アルゴリズム：SVM（線形カーネル）
- 素性：Wikipedia 記事本文の形態素の基本形
- 素性値：TF
- 学習データ：NAIST_JENE データ [9] の Wikipedia 記事のうち，日本語 Wikification コーパスが参照する箇所を除外したデータ

次に，評価指標について述べる．候補生成のうち，NIL セットについては平均候補数とシャットアウト率で評価する．シャットアウト率とは評価対象の言及のうち，出力候補数が 0 であるものの割合である．通常セットについては平均候補数と正解含有率で評価する．正解含有率とは評価対象の言及のうち，正解エンティティが候補に含まれているものの割合である．また，Wikification については Wikification 正解率を用いて評価する．Wikification 正解率とは評価対象の言及のうち，正解エンティティを正しく出力できたものの割合である．

以下ではまず，候補生成のみを実行する実験（実験 1 ）の結果を示し，その後，実験 1 の結果に続けて Wikification を実行した実験（実験 2 ）の結果を示す．

3.2　（実験 1 ）候補生成

図 2 に，NIL セットに対して提案手法を適用する前後での候補生成の結果を示す．横軸が平均候補数，縦軸がシャットアウト率である．図 3 に，通常セットに対しての結果を示す．横軸が平均候補数，縦軸が正解含有率である．

また，NIL セットのうち，候補をすべて削除できた言及の例を以下に示す．括弧内は言及に対して自動推定した固有表現クラスの値である．例えば，“二階”は「地下鉄駅前のビルの 二階 という...」という文脈に現れる言及であるが，人物や小説である候補を正しく削除できている．

言及	削減前の候補
二階（Facility）	二階俊博，二階 (松本清張)
社会福祉法人（Product）	社会福祉法人
事務局長（Product）	国際連合児童基金, 事務総長
中村 (Person)	中村駅, 中村 (相撲), 中町 (岡崎市)
小早川 (Location)	小早川氏, 小早川毅彦

実験結果から以下のことが分かる．まず，どちらのデータセットでもルール適用によって候補が削減できている．Tight ルールの方が削減数が大きい．NIL セットでは，ルール適用によってシャットアウト率が向上している．しかし，通常セットでは，正解含有率の低下を招いている．

続いて，固有表現クラス毎の内訳を見て実験結果の詳細を確認する．表 3 に，NIL セットに対して Tight ルールを用いて候補削減をした場合の削減前後の結果を言及の固有表

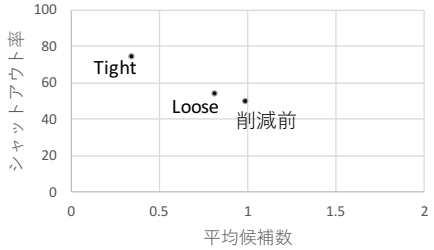

図 2　候補生成の結果（NIL セット）

図 3　候補生成の結果（通常セット）

現クラス毎に示す．各クラスの上段が平均候補数，下段がシャットアウト率である．また，表 4 に，通常セットに対して Tight ルールを用いて候補削減をした場合の削減前後の結果を言及の固有表現クラス毎に示す．各クラスの上段が平均候補数，下段が正解含有率である．

表を見ると，NIL セットについてはすべての固有表現クラスにおいて削減前後でシャットアウト率が向上している．一方で，通常セットにおいて正解含有率が落ち込んでいる．以上から，提案手法は NIL エンティティの同定には有効であるが，通常のエンティティに対しては候補を過剰に削減していることがわかる．

3.3　（実験 2）Wikification

Tight ルールで候補生成を実行した後，Wikification 処理を行なった．Wikification 処理には Jargalsaikhan ら [2] と同じ手法を用いる．すなわち，言及 m に対して m のリンク先がエンティティ e になる確率 $p(e|m)$ を求め，その値が最大となるエンティティ候補を出力する．

実験結果を表 5 に示す．これは，3.2 節 とは違い，NIL セットおよび通常セットの言及をまとめた結果である．表の上段は Wikification 正解率であり，下段はその計算元になる各言及数である．

提案手法では候補生成の時点で正解含有率の低下を招いているため，多くのクラスで Wikification 正解率も低下していた．ただし，正解含有率の低下を少なく抑えられていたクラス “Person” では，Wikification 正解率が向上しており，候補生成におけるシャットアウト率の正解含有率のバランスによっては Wikification 正解率が向上することが確認できる．

固有表現クラスが “Person”と推定された言及について，候補削減前は正解できなかったが提案手法で候補削減をお

表 3　候補生成の結果（NIL セット，固有表現クラス別）

	削減前	削減後
Color	1.0 0.167	0.0 1.0
Disease	0.34 0.710	0.05 0.995
Event	0.43 0.817	0.10 0.921
Facility	0.41 0.750	0.16 0.874
God	1.0 0.0	0.0 1.0
Location	1.02 0.456	0.47 0.632
Name_Other	1.0 0.50	0.0 1.0
Natural_Object	1.01 0.391	0.41 0.609
Organization	0.73 0.542	0.20 0.812
Person	1.36 0.593	0.53 0.734
Product	1.02 0.352	0.30 0.705

表 4　候補生成の結果（通常セット，固有表現クラス別）

	削減前	削減後
Color	1.57 0.973	0.0 0.0
Disease	1.08 0.938	0.36 0.344
Event	1.89 0.816	0.67 0.453
Facility	1.17 0.839	0.48 0.387
God	- -	- -
Location	1.68 0.961	0.93 0.719
Name_Other	1.0 1.0	0.0 0.0
Natural_Object	1.50 0.931	0.76 0.692
Organization	1.22 0.838	0.55 0.472
Person	2.11 0.738	1.17 0.659
Product	1.75 0.888	0.23 0.166

表 6　提案手法によって Wikification が正解できるようになった言及例（Person クラス）

言及	正解エンティティ	削減前の候補（提案手法で削減された候補に下線）
パウンド	NIL	グラウンドパンチ
蓮	NIL	ハス, 蓮駅, 蓮 (駆逐艦)
イエス	イエス・キリスト	イエス・キリスト, イエス (バンド)
菅	菅直人	菅直人, 神奈川県立菅高等学校, 菅義偉, 菅, 菅 (川崎市), 菅内閣

表 7　提案手法によって Wikification が誤るようになった言及例（Person クラス）

言及	正解エンティティ	削減前の候補（提案手法で削減された候補に下線）
小泉純一郎	小泉純一郎	小泉純一郎
渡辺茂	渡辺茂	渡辺茂, 渡辺茂 (作曲家), 渡辺茂 (システム工学者)
清水	清水エスパルス	清水エスパルス, 清水 (杉並区), 清水 (名古屋市)
野口裕之	野口裕之	野口裕之, 野口裕美

表 5　Wikification 結果（NIL セット，通常セットの合算）

	削減前	削減後
Color	0.837 (36/43)	0.140 (6/43)
Disease	0.866 (116/134)	0.515 (69/134)
Event	0.759 (381/502)	0.715 (359/502)
Facility	0.757 (809/1069)	0.653 (698/1069)
God	0.0 (0/1)	1.0 (1/1)
Location	0.759 (3336/4396)	0.679 (2983/4396)
Name_Other	0.75 (3/4)	0.5 (2/4)
Natural_Object	0.773 (408/528)	0.669 (353/528)
Organization	0.690 (1382/2003)	0.596 (1194/2003)
Person	0.634 (2203/3476)	0.676 (2350/3476)
Product	0.583 (3206/5495)	0.443 (2433/5495)

こなうことで正解できるようになった言及の例を表 6 に示す．例からわかるように，NIL エンティティに対して候補をすべて削除することに成功している事例もあれば，NIL エンティティではないが，不要な候補を削減することで Wikification が正しく行えるようになる事例もあることがわかる．また，逆に，提案手法で候補削減をおこなうことで誤るようになった言及の例を表 7 に示す．例から，提案手法によって正解エンティティを誤って削減してしまっていることがよくわかる．

4　おわりに

本稿では，Wikification の候補生成の処理において，言及およびエンティティ候補となる Wikipedia 記事の固有表現クラス情報を用いることで候補を削減する手法を提案した．評価実験の結果，提案手法を適用することで，NIL エンティティの候補をうまく削減できる一方で NIL でない通常エンティティの候補も多く削減してしまうことがわかった．ただし，一部の固有表現クラスによっては Wikification 正解率の向上に繋がることも確認できた．今後は，候補保持ルールの追加検討に加え，誤り事例の中には固有表現クラス情報の推定誤りに起因する事例も見られたため，これらの推定精度の向上についても検討していきたい．

謝辞

本研究の一部は科研費（15K20884）の助成を受けて実施されました．

参考文献

[1] Rada Mihalcea and Andras Csomai. Wikify!: Linking documents to encyclopedic knowledge. *CIKM*, pp. 233-242, 2007.

[2] Davaajav Jargalsaikhan, 岡崎 直観, 松田 耕史, 乾 健太郎. 日本語 Wikification コーパスの構築に向けて. **言語処理学会 第 *22* 回年次大会 発表論文集** *(2016* 年 *3* 月*)* , pp. 793-796.

[3] 松田 耕史, 岡崎 直観, 乾 健太郎. 日本語 wikification ツールキット: jawikify. **言語処理学会 第 *23* 回年次大会 発表論文集** *(2017* 年 *3* 月*)* , pp. 250-253.

[4] Shuangshuang Zhou, Koji Matsuda, Ran Tian Naoaki, Okazaki Kentaro Inui. A Pipeline Japanese Entity Linking System with Embedding Features. *30th Pacific Asia Conference on Language, Information and Computation (PACLIC 30) Seoul, Republic of Korea, October* 28-30, 2016.

[5] 藤井裕也, 飯田 龍, 徳永健伸. Wikipedia 記事を利用した曖昧性のある表現の固有表現クラス分類. **言語処理学会 第 *16* 回年次大会 発表論文集** *(2010 / 3)*, pp.15-18.

[6] 関根聡, 安藤まや, 松田耕史, 鈴木正敏, 乾健太郎. 「拡張固有表現＋ wikipedia」データ. **言語処理学会第 *22* 回年次大会**, *2016.*

[7] 南和江, 藤井康寿, 土屋雅稔, 中川聖一. 大規模コーパスを用いた固有表現抽出手法の検討. **言語処理学会 第 *17* 回年次大会 発表論文集**, pp. 328-331, 2011.

[8] 橋本泰一, 乾孝司, 村上浩司. 拡張固有表現タグ付きコーパスの構築. **情報処理学会自然言語処理研究会** *(2008-NL-188)*, 2008.

[9] NAIST_JENE データ. https://github.com/masayu-a/NAIST-JENE