日本語 wikification ツールキット: jawikify

松田 耕史 † 岡崎 直観 † 乾 健太郎 †
† 東北大学
{matsuda,okazaki,inui}@ecei.tohoku.ac.jp

1 はじめに

エンティティリンキングとは，テキスト中の固有表現を，知識ベースのレコード (エンティティ) に対応付けるタスクである．知識ベースに Wikipedia を用い，レコードとして個別の Wikipedia ページを用いるときには，特に wikification と呼ばれる．エンティティリンキングは，近年の Freebase や DBpedia 等 のエンティティに基づいた知識ベースの発展に伴って重要性が増しているタスクである．情報検索システムの性能向上など，自然言語処理の応用分野からの要請のみならず，自然言語処理の基礎的なタスクで活用する知識を大規模テキストから自動獲得するタスクにおいても，基盤技術として重要である．

本論文では，我々が構築を進めている日本語テキストに対する エンティティリンキング / wikification のためのソフトウェアツールキットについて紹介する．英語に対応したソフトウェアはいくつか公開されているものの，日本語に対しては自由に使えるソフトウェアが存在していない[1]．唯一，Google Natural Language API を用いてエンティティ解析が可能であるが，現状ではあまりカスタマイズの余地がない．また，従量課金制の有償であるため，大規模なテキストデータに適用して知識獲得を行うような用途には現在のところ適用が難しい．

カスタマイズ性もエンティティリンキングにおいて重要な要素である．Wikipedia は現実世界のエンティティをバランス良く含んではいるものの，テキストのドメインによっては，Wikipedia において単一の記事になっていないエンティティに対してもリンクが必要になることがある．たとえば，音楽関係の記事を解析する際には，曲名やバンドのメンバーなど，Wikipedia の記事にはなりにくい固有名詞をデータベースレコードにリンクすることが重要であることは想像に難くない．

そこで，我々は別の選択肢として，オープンソースの wikification ツールキット `jawikify` を開発している．本論文ではその現状について述べる．`jawikify` には，以下のような特徴がある．

図 1: wikification のタスク概要

- エンティティ辞書のカスタマイズが可能である．明示的に辞書を持っているため，辞書に新しいエントリを追加することで，モデルの再訓練を伴わずに新しいエンティティを抽出することが可能である．加えてタグ付きコーパスを用意すれば，モデルの再訓練も可能である．
- Wikipedia 日本語版をベースにしたエンティティ辞書，解析モデルを同梱しており，すぐに使いはじめることが可能である．

本ソフトウェアは，日本語 wikification におけるエンティティ曖昧性解消について論じた Zhou らの報告 [4] を参考に，その中で効果が高いと報告されている手法をまとめてパッケージ化し，言及抽出モジュールを加えたものである．

2 エンティティリンキングの流れ

我々は，エンティティリンキングを大きく以下の二つのサブタスクに分けてパイプライン的に解くことにした．

言及抽出 このステップにおいては，テキスト中のどのスパンをエンティティにリンクするか決定する．このテキストスパンを **言及** と呼ぶ．図 1 におけ

[1] 正確に言えば， Babelfy (`http://babelfy.org/`) は日本語の文書を処理できるものの，単語という区切りが存在せず，文字単位の処理になっているため，分かち書きされていないテキストに対してその正確性については検証が行われていない．

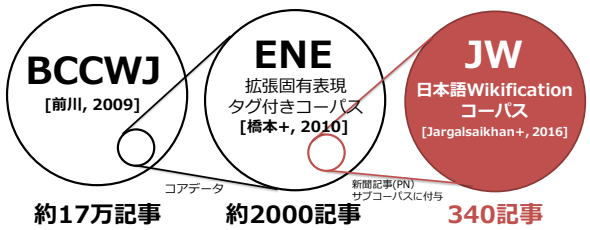

図 2: コーパス間の包含関係

る，例文の下線部分が言及である．言及を抽出する問題は固有表現抽出問題とみなすことが可能であるが， wikification タスクの場合は一般的な固有表現認識問題で扱われるカテゴリよりも，自然物の名前や現象の名前など，より広いカテゴリの言及を扱う必要がある．

エンティティ同定 言及抽出で抽出した言及それぞれに対して，どの **エンティティ** が対応するかを決定する．図 1 における枠で囲まれた Wikipedia の記事が， wikification タスクにおけるエンティティである．言及には曖昧性があるため，これは語義曖昧性解消問題の一種のバリエーションとみなすことが可能である．典型的な語義曖昧性解消問題とのあいだには，語義目録が WordNet や国語辞典などの機械可読辞書の見出し語であるか，それとも知識ベースのエントリであるか，という違いが存在する． wikification において，語義目録はすべての Wikipedia ページの集合である．このサブタスクの中には，候補集合の中に適切なエンティティが存在しない，ということを検知する問題（NIL 検知）も含まれる．たとえば，図 1 における “キリン・カップ第 2 戦” という言及は， Wikipedia 内に対応するエンティティが存在しないため NIL という特別なエンティティを付与する．

エンティティリンキングの統一されたタスク定義は，英語においてもいまだに存在せず，若干の混乱があることが指摘されている [2]．たとえば，どのようなクラスの表現をリンク対象とするかといったタスクの根幹をなす設定において，統一された見解は存在せず，いくつかの研究グループが，それぞれ異なった正解基準の元に研究を行っている．

日本語においても，長らく広く使われるような正解データは存在しなかったが，近年 Jargalsaikhan らは，関根の拡張固有表現に定義された固有表現クラスがアノテートされた拡張固有表現タグ付きコーパス [8]（以下，**ENE コーパス** と表記する）をベースとして，その一部 (およそ 340 記事の新聞記事) に対してエンティティ情報が付与されたコーパス（日本語 wikification コーパス:以下，**JW コーパス**）を作成した [1]．ENE コーパスは BCCWJ コーパス [7] のコアデータをもとに構築されており，BCCWJ コーパス，ENE コーパス，JW コーパスの包含関係は概ね 図 2 のようになっている．

我々は JW コーパスをリファレンスコーパスとして用い，エンティティ情報を付与することを目指す．ただし，JW コーパスは比較的小規模であり，言及抽出モデルの学習に充分ではないと考えたため，言及抽出モデルは ENE コーパスから学習を行った．

2.1 言及抽出

リンク対象とするテキストスパンを抽出するために，単語単位で CRF に基づく系列ラベリングを行った．ENE コーパスには，およそ 200 クラスにおよぶ固有表現のクラスとその境界情報が付与されている．しかしながら，200 クラスの細かいレベルの固有表現クラスを直接当てるのは，データスパースネスと計算時間の問題から現実的ではない（CRF のデコードに用いられる動的計画法に基づく推論アルゴリズムである Viterbi アルゴリズムは，クラス数の 2 乗に比例する時間計算量を要する）．そのため我々は，ENE が持つ階層構造を利用し，そのトップグループ 11 クラスにラベルを抽象化し，そのラベル集合の上で系列ラベリングを行った．内部的には，チャンクの表現方法として IOB2 コーディングを用い， CRFSuite をバックエンドの実装として用いている．モデルの素性として，単語の表層形，単語に含まれる文字の集合，単語の最初の文字とその文字種，単語の最後の文字とその文字種を，対象単語および，その前後二単語から抽出している．

2.2 エンティティの同定

このステップにおいては，言及が指すエンティティを候補の中から選択する．具体的には，言及抽出で検出された全ての言及に対して，その言及に結びつく可能性のあるエンティティを辞書から検索し，候補エンティティの集合を生成する．その後，候補集合の中から，言及が出現した文脈において最も尤もらしいエンティティをランキングすることによって選択する．

候補集合の生成においては，Wikipedia 内のリンクアンカーの情報から自動で生成 [3] した辞書を用いた．具体的には，“拡張固有表現+ wikipedia データ” [5] から作成したデータを用いている．これは 2015 年 11 月の Wikipedia 日本語版のダンプから得られたメタデータを JSON 形式でまとめたものであり，候補生成のための辞書やエンティティの事前確率の計算に必要なアンカーに関する情報が含まれている．アンカーから生

成した辞書に基づく候補生成は英語におけるエンティティリンキングのベースライン手法として有用であることが報告されており，日本語においてもカバレッジは充分に高い（90%以上）ことが報告されている [4]. しかし，この方法にも，候補エンティティが増えすぎることがあるという問題が存在する．たとえば，“日本” という言及は 256 種類，“オリンピック” という言及は 155 種類もの候補エンティティがあり，そのほとんどは殆ど言及されないエンティティである．今回は計算量を削減するために，辞書でマッチしたエンティティの中から比較的メジャーなエンティティ（2.3 で述べる事前確率が 0.05 以上のエンティティ）のみを候補にした．

エンティティの同定においては，言及が出現した文脈において尤もらしいエンティティを候補集合をランキングすることで選択するモデルを採用した．ランキングのためのスコアは以下の式で計算される:

$$score(e, m, c) = \sum_i w_i f_i(e, m, c)$$

ここで，$f_i(e, m, c)$ は言及 m と エンティティ e，言及が出現した文脈 c を引数にとる素性関数であり，w_i は重みである．ある文脈と言及が与えられた際に，正しいエンティティとの組み合わせに対して，そうでないエンティティより相対的に高いスコアが出力されるように重みを訓練する．ランキング関数の学習には線形カーネルのランキング SVM（SVM^{rank}）を用いた．

2.3 ランキングモデルの素性

Zhou らは，日本語におけるエンティティ同定において有効な素性についての分析を行っている [4]．我々もこれに沿い，効果が高いと報告されている以下の 3 種類の素性を用いてランキングモデルを訓練した．

文字列類似度 エンティティ e を指す Wikipedia 記事のタイトルと，言及 m の間のレーベンシュタイン距離を 1 から引き，類似度として用いた．

大域文脈 エンティティ e を指す Wikipedia 記事のアブストラクトと，言及が含まれる文書との間の TF-IDF で重み付けされた cos 類似度．

事前確率 $p(e|m)$ で計算される，言及のもとでのエンティティの条件付き確率．アンカー辞書から計算可能である．

2.4 NIL 検知

言及抽出において，エンティティに対する言及であると認識されたとしても，実際には知識ベースに収録されていないエンティティにたいする言及である場合もある．今回は，候補生成のステップにおいて候補が見つからなかった場合に NIL と判定した．

表 1: 言及抽出モデルの性能 (上位階層 11 クラスのマイクロ平均)

訓練コーパス	評価コーパス	精度	再現率	$F_{\beta=1}$
ENE-	ENE-	67.19	43.58	52.87
ENE-	JW	66.28	47.79	55.54
ENE- + JW	JW	70.85	55.84	62.45

3 実験

我々が構築した wikification ソフトウェアの性能のおおよその見積もりをつけるために，コンポーネント別に以下のような評価を行った．

言及抽出モデルの学習に用いる ENE コーパスと，エンティティ同定モデルの学習に用いる JW コーパスには包含関係があるため，ENE コーパスから，JW コーパスに含まれる文書 (新聞記事サブコーパス) を除いたコーパス（ENE-）を作成し，このコーパスを 8:2 に分割して，それぞれ言及抽出モデルの訓練とテストに利用した．また，言及抽出モデルの学習元である ENE コーパスと JW コーパスのドメインの差がどのように影響を及ぼすかを調べるため，同様に，ENE-コーパス全体を訓練コーパスとして用い，JW コーパスのうちの 140 記事をテストコーパスとした場合の性能を測定した．

最後に，対象ドメインの訓練データを追加することでの性能向上を確かめるために，ENE-コーパス全体に加えて，JW コーパスのうちの 200 記事を訓練データとして用い，JW コーパスのテストセット 140 記事で評価を行った．

エンティティ同定におけるランキングモデルの評価においては，JW コーパスのうちの 200 記事で訓練を行ったモデルを残りの 140 記事で評価した．

3.1 結果と考察

言及抽出の性能を 表 1 に示す．いずれの設定においても，それほど性能が高いとはいえない．また，JW コーパスの訓練セットを訓練データに加えることで，幾分の性能向上がみられる．既存研究 [9] 比較すると，全体的に低い性能にとどまっているが，抽出が容易で比較的高い F 値を達成できる時間表現・数量表現などは wikification タスクにおいては抽出する必要がないた

表 2: JW コーパスのテストデータ 140 文書における wikification の性能

言及データ：	Gold standard			言及抽出モデル		
素性セット	精度	再現率	$F_{\beta=1}$	精度	再現率	$F_{\beta=1}$
(A) 文字列類似度	0.766	0.756	0.761	0.781	0.685	0.730
(B) 大域文脈	0.721	0.711	0.716	0.735	0.645	0.687
(C) 事前確率	0.782	0.772	0.777	0.796	0.698	0.744
(A) + (B) + (C)	0.785	0.775	0.779	0.799	0.701	0.747

め，今回の評価対象には含めておらず，直接比較することはできない，という点に注意が必要である．

JW コーパスのテスト文書 140 文書における wikification （言及抽出およびエンティティ同定）の性能を表 2 に示す．すべての素性を用いた場合，強いベースラインになりえる事前確率ベースラインに対して僅かに性能が向上した．また，言及抽出の性能がそれほど高くないにもかかわらず，最終的な wikification 結果は大きく悪化していない．再現率に影響を与えているものの，精度は逆に向上しており，F 値の低下は 3〜4 ポイント程度にとどまっている．これは，エンティティ同定が比較的容易に行えるメジャーなエンティティに対する言及は言及抽出においても同様に容易に抽出できるため，全体への影響は少ないこと，今回はエンティティ同定において言及の意味クラスを考慮していないため，エラーの大部分を占める意味クラス同士の誤りの影響を殆ど受けなかったことが理由になると考えている．

4 エンティティ辞書のカスタマイズ

`jawikify` の Google Natual Language API に対する利点として，エンティティ辞書のカスタマイズが容易である点がある．特定のドメインのテキストを解析する際に，その分野でよく使われるエンティティを，モデルの再学習を経ずに新しく抽出対象に含めることが可能である．エンティティを追加するためには，エンティティの名前，説明文などの情報を JSON 形式で保存し，辞書の追加コンパイルを行えばよい．詳しくは，ソフトウェアのドキュメントを参照されたい．

5 おわりに

本稿では，我々が構築している wikification ツールキットについて紹介した．現状はまだシンプルな構成であるが，新聞記事に出現する言及のうち，およそ 7 割前後の言及に対して正しいエンティティを割り当てる事が可能である．今後，エンティティに対して付与された関根の拡張固有表現階層に基づくクラスラベルの情報 [6] や ENE コーパスで提供されているような細粒度のカテゴリ情報が，エンティティの曖昧性解消にどのように影響するか調査する予定である．また，Google Natural Language API 等の既存ツールとの性能比較を行っていく予定である．

本ソフトウェアは，実行に必要な知識ベースを含めて， github [2] にてオープンソースライセンスのもとで公開しているので，フィードバックを頂ければ幸いである．

謝辞 この研究は，文部科学省受託研究「実社会ビッグデータ利活用のためのデータ統合・解析技術の研究開発」の一環として行われた．

参考文献

[1] 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, 2016.

[2] Xiao Ling, Sameer Singh, and Daniel Weld. Design challenges for entity linking. *TACL*, Vol. 3, pp. 315–328, 2015.

[3] Valentin I. Spitkovsky and Angel X. Chang. A cross-lingual dictionary for english wikipedia concepts. In *Proc. of LREC*, pp. 3168–3175, 2012.

[4] Shuangshuang Zhou, Koji Matsuda, Ran Tian, Naoaki Okazaki, and Kentaro Inui. A pipeline japanese entity linking system with embedding features. In *Proceedings of the 30th Pacific Asia Conference on Language, Information and Computation (PACLIC 30)*, pp. 267–276, 2016.

[5] 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.

[6] 鈴木正敏, 松田耕史, 関根聡, 岡崎直観, 乾健太郎. Wikipedia 記事に対する拡張固有表現ラベルの多重付与. 言語処理学会第 22 回年次大会, 2016.

[7] 前川喜久雄. 代表性を有する大規模日本語書き言葉コーパスの構築 (<特集>日本語コーパス). 人工知能学会誌, Vol. 24, No. 5, pp. 616–622, sep 2009.

[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 研究報告 自然言語処理 (2008-NL-188), pp. 113–120, 2008.

[9] 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, 2011.

[2] `https://github.com/conditional/jawikify`