The 32nd Annual Conference of the Japanese Society for Artificial Intelligence, 2018

多段 Doc2Vec によるエンティティリンキングの応用

An entity linking method using multiple Doc2Vec

津々見誠[*1] 村上浩司[*2] 梅田卓志[*2]
Makoto Tsutsumi Koji Murakami Takashi Umeda

[*1] 楽天株式会社 Architecture & Core Technology Platform 部
Rakuten, Inc. Architecture & Core Technology Platform Department

[*2] 楽天技術研究所
Rakuten Institute of Technology

Appropriately linking a polysemous word in a text to its corresponding entity in a knowledge base is an essential part in producing structured knowledge. Linking becomes challenging as issues such as name variations, entity ambiguities or absence of entity in knowledge base occurs in the process. We present an entity linking method consisting of multiple Doc2Vec model, achieving high performance and cost effectiveness. In the experiments, our proposed method achieved 83.5% in mapping accuracy, improving 31.0 points higher than the simple string matching.

1. はじめに

近年，データ処理基盤の高性能化に伴ってデータ活用の必要性の高まりが著しい．さらに機械学習や自然言語処理分野における技術革新により，インターネット上に存在する Twitter や Facebook などの CGM (Consumer Generated Media)や，話し言葉の書き起こし，E-Commerce の商品タイトルなど，データを組み合わせて利用する試みが広くみられる．しかしながらこうした非構造化データを扱うことは容易ではなく，何らかの構造化が必要になることが多い．データの構造化は，その目的によって着目する情報の種類が異なる．典型的には大きさや材料などの属性抽出が考えられるが，話題により抽出対象が変化する場合もある．また類義語の抽出や部分全体関係抽出，上位下位関係抽出もデータ構造化の重要なタスクである．こうした構造化においての難しさの一つが，単語の多義性である．例えば「アップル」には「企業名」と「果物名」という2種類の側面があり，必要に応じて適宜構造化する必要がある．

多義性を持つ語と，その実体を正しくリンクさせるための自然言語処理分野の研究技術として，エンティティリンキング(Entity Linking)が挙げられる [1]．これは，テキスト中で何らかの実体（Entity）を指示する記述(Mention)を抽出して，外部の知識ベースに対応付ける技術である．これにより，表記のゆれや語義の曖昧性を解消してエンティティ間の関係性を正確に把握することが可能となる． データ構造化の集大成として整備が進められている外部知識ベースに Wikipedia[*3] がある．これは最も大規模なオンライン百科事典であり，多くのエンティティ間に関係のリンクが設定されており，言語資源としての価値は高い．

我々はデータの構造化の一環として，楽天市場の商品タイトルから必要な情報の抽出を行なっている．ブランドやメーカーは典型的な属性抽出の対象であるが，その名称そのものだけではなく，サブブランド，シリーズと言った上位下位関係の同定も必要となる．本研究では，楽天市場上の商品のメーカー名と，そのメーカーと一致する企業名とのマッピングの自動化に取り組む．本研究でマッピングの対象となる企業は，世間に広く認知されている公開企業とした．本研究における大きな課題は，多義性のあるメーカー名と企業名を高い精度でマッピングすることである．例えば，楽天市場に出現するメーカーに「パイオニア」があるが、これは異なる2企業で使われている．一方は電気機器を主力製品とする上場企業であり，他方は雑貨の企画及び製造販売を行う企業である．マッピングの自動化のためには，こうした企業を正確に区別し適切にマップする枠組みが必要である．

我々は，楽天市場の商品情報から予め抽出した約 70,000 種類のメーカーリストと，約 2,000 の企業リストの間でマッピングを行う．また，企業の連結子会社等もマッピングの対象とした．つまり，メーカー「NEC」や「日本電気通信システム」は，ともに「日本電気」に紐付けられることになる．同様にして，「フジテレビジョン」や「ポニーキャニオン」は，ともに「フジ・メディア・ホールディングス」を正解とする．

また使用する外部知識には，(1) 有償の企業リストなど外部データは使用しない，(2) 人手によるアノテーションはあくまで評価用データに限定し，必要な人手を最小限にする，の2点の制約を設け収集，利用した．我々は実験結果から，文書や単語の分散表現を多段に用いたエンティティリンキング手法により，従来の部分文字列一致等による手法を大きく上回る精度を得た．

2. 関連研究

2.1 エンティティリンキング

エンティティリンキングは一般に，「候補の生成」，「候補の順位づけ」，「リンク有無の予測」に分かれる [2]．候補の生成では，表記ゆれ等を列挙した辞書ベースの使用，Wikipedia のリンク構造等を利用した語の拡張，サーチエンジンの使用等の研究例がある．候補の順位づけにおいては，教師有りの手法，また教師無しの手法が見られる．教師有りにおいてはバイナリ分類器からグラフベースの手法まで様々であるが，教師データの作成に大きなコストがかかる上，これらの多くはデータセットやタス

連絡先：津々見誠，楽天株式会社 Architecture & Core Technology Platform 部，〒158-0094 東京都世田谷区玉川一丁目 14 番 1 号，050-5581-6910，ts-makoto.tsutsumi@rakuten.com

[*1] https://www.rakuten.co.jp/
[*2] https://rit.rakuten.co.jp/
[*3] https://ja.wikipedia.org/wiki/

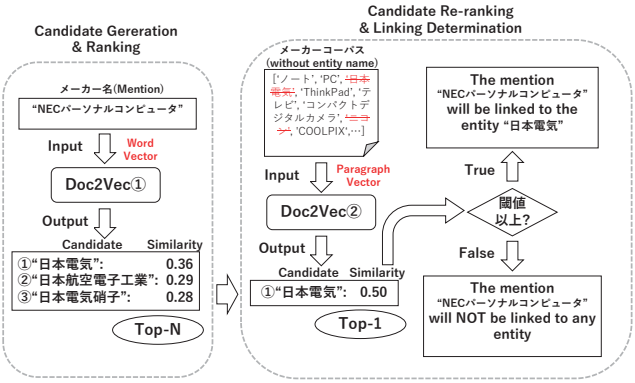

図 1　提案手法によるリンキング

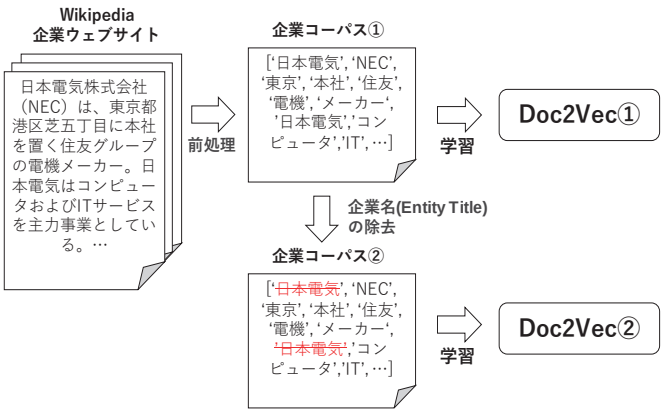

図 2　コーパスの準備と Doc2Vec モデルの学習

クに依存することから，異なるドメインへのモデルの転用が難しく，汎用性に欠ける [3].

日本語のエンティティリンキングの研究は英語圏と比較して少ない．古川らは学術分野の知識ベースが英語中心であることに着目し，日本語の用語を英語に対訳する手法を提案している [4]．また Zhou らの手法では Wikipedia のアンカーテキストを用いたエイリアス辞書により語を拡張し，Entity タイトルとの文字列間距離をとって候補を生成する．順位づけでは文字列距離，エンティティの被リンク数，周辺語の BoW(Bag-of-Words)，分散表現等を素性に用いた上で，SVM による教師有り学習を行うことで良好な結果が得られている [5]．これらの研究は候補生成と候補の順位づけにおいて，異なる手法またはモデルを採用している共通点がある．候補生成においては辞書の活用や語の拡張に留まり，分散表現を利用した研究は多くない．

我々の研究は，以下の 2 点で Zhou らの研究と大きく異なる．(1) 候補生成と候補順位づけで使用するモデル基盤を共通化する．すなわち，Doc2Vec を多段に使用する．この際，語の拡張やエイリアス辞書の作成を必要としない．(2) 候補生成，候補順位づけともに教師なし学習である．このため，教師データの作成を必要としない．これらの理由から，提案手法は従来手法と比較して大幅な省力化が可能である．

2.2 Doc2Vec

Doc2Vec[1]とは，単語を数百次元程度の実数ベクトルで表現する手法である Word2Vec を，文書単位へと拡張した手法である [6] [7]．Word2Vec はニューラルネットワークモデルであり，中心の単語から周辺の単語を予測するモデルである Skip-gram と，周辺の単語から中心の単語を予測するモデルである CBoW（Continuous Bag-of-Words）に分かれる．Doc2Vec においては，Skip-gram を拡張したものは PV-DBOW(Paragraph Vector with Distributed Bag-of-Words)，CBoW を拡張したものは PV-DM(Paragraph Vector with Distributed Memory)と呼ばれる．

Doc2Vec は分散表現として単語ベクトル(Word Vector)とパラグラフベクトル(Paragraph Vector)を保持し，これらはニューラルネットワークモデルの中間層の重みで表現される．パラグラフベクトルは局所的な文脈の中で失われた情報を代表するものであり，パラグラフのトピック等抽象度の高い情報を記憶する機能を持つ [7]．離散的で高次元，かつスパースな BoW による表現と比較して，分散表現は低次元で密な表現であり，単語や文書間の距離をより高い精度で計算可能である．

3. 提案手法

3.1 基本的な考え方

メーカー名と企業名とのリンキングを考えた場合，基本的には着目するメーカー名に近い名を持つ企業が候補として考えられる．しかし，1節の「フジテレビジョン」や「ポニーキャニオン」のように，複数のメーカーが単一の企業に結びつく場合や，「パイオニア」のように，同一のメーカー名が異なる企業を指す場合を考慮する必要がある．つまり語やエンティティの多義性が大きな問題となる．我々はこの問題に対して，(1)任意の文書には，用語・文脈・文書トピック等，抽象度の異なる情報が存在する，(2)語やエンティティの多義性の解消には，これら抽象度の異なる情報を組み合わせることが有効である，の2つの仮説を立てた．人間は多義性のある用語の解決を試みるとき，語そのものからは一旦距離をおき，より広い文脈から俯瞰して用語の意味を検討する．上記の仮説はこの流れに沿うものである．

上記の仮説に基づき，提案手法では Doc2Vec モデルを多段に組み合わせ，抽象度の異なるコーパスと分散表現を使用することで語とエンティティの多義性解消を試みる．1段目ではエンティティ名を含むコーパスと Word Vector を使用することで，局所的で具体性の高い用語レベルの情報を重視し，候補漏れを最小限に抑えた候補生成と順位づけを行う．2段目ではエンティティ名を除いたコーパスと Paragraph Vector を使用することで，大域的で抽象度の高い情報を重視し，語の多義性を解消しつつ，候補の再評価とリンキング有無の決定を行う．

3.2 手法の流れ

提案手法は大きく(1)コーパスの準備，(2)リンキング候補の生成と順位づけ (3)候補の再評価とリンキング有無の決定，の 3 つから構成される．全体の処理を図 1 に示す．

(1) コーパスの準備

ナレッジベースとなる企業側の情報として，Wikipedia および企業のウェブサイトを使用した．

最初に Wikipedia 上に存在する企業のページを取得した．Wikipedia はエンティティ間に多数のリンクを内包し，さらにページ内の重要な用語には対応するエンティティへのリンクが付加されている．一方で，知名度の低い企業等はページやリンクが存在しないケースが多く，これらの情報を利用できない．こうした情

[1] https://radimrehurek.com/gensim/models/doc2vec.html

報の欠損に対処するため，Wikipedia に加えて各企業のウェブサイトから，会社情報や事業紹介等を中心に 3 ページほど取得した．さらに Wikipedia は id=content タグ内を，企業ウェブサイトは body タグ内を対象とし，HTML タグや javascript コード等を除去した．次に各文書に対して以下の前処理を行ったのち MeCab [8]＋ipadic-utf8（ユーザー辞書に企業名を追加）の組み合わせで形態素解析し，名詞を抽出してコーパス（企業コーパス①）とした．

- 英数字は，全角から半角に変換
- 頻度の高い用語のうち，「項目」「関連」等，具体性が低くノイズとなりそうな用語をストップワードとして除去

上記の前処理に加えて，2 段目の Doc2Vec モデル用に企業コーパス①から企業名を除去したコーパス（企業コーパス②）も用意する．

またモデルへの入力とするリンキング元の情報として，予め抽出したメーカー名リストに加えて，各メーカーを表現する文書を準備する．作成にあたり，「商品価格ナビ」のウェブページを利用した[1]．本ページでは任意のメーカーの主要製品をジャンル別の売上順に把握できる．このページ及び上位 4 製品の商品情報をメーカーの文書とし，企業コーパス②と同様に前処理，名詞抽出，企業名削除を行いメーカーコーパスとした．

(2) リンキング候補の生成と順位づけ(Doc2Vec 1 段目)

候補となるエンティティの生成と順位づけを行う．本段階での目標は，約 2,000 ある企業のうちリンキングすべき企業を TOP-N の範囲に含めることである．ここで TOP-N に含まれない場合，後述する段階での挽回が不可能となるため，高い精度で TOP-N に含める仕組みが求められる．さらにメーカー「ポニーキャニオン」と企業「フジ・メディア・ホールディングス」のように名称に類似性がないリンキングも少なくない．この問題の解決のため，表層的な文字列間距離ではなく単語の分散表現を使用した．具体的にはメーカー名の Word Vector と企業名の Word Vector との間のコサイン類似度を計算し，上位 N 件の企業を TOP-N として選び出す．

Doc2Vec モデルの学習に用いたパラメータは以下の通りである．これらは予備実験を通して最良な結果が得られた組み合わせである．図 2 にモデル学習の概要を示す．

- モデルとして，PV-DBOW を使用する．
- Word Vector の再学習を行う．
- 周辺単語の窓幅=5，中間層サイズ=300，エポック数=10

(3) 候補の再評価とリンキング有無の決定(Doc2Vec 2 段目)

生成された候補を TOP-N の範囲で再度順位づけし，TOP-1 を選び出してリンキング有無の決定を行う．本段階での目標は，候補を別の観点から再評価し，リンキングの誤りをできるだけ減らすことである．例えば，「パイオニア」のように企業名が同一で実体が異なる企業の判別は，メーカー名を入力とする 1 段目のモデルだけでは原理上不可能である．2 段目では，これら多義性を含む語の語義曖昧性の解消が求められる．

そのため 2 段目の Doc2Vec では，モデルとパラメータは1段目と共通であるが，コーパスとベクトルの種類を変更して学習を行う．具体的には企業コーパス②を用いて学習させた Doc2Vec モデルに対して，メーカーコーパスを入力とし，その Paragraph Vector と企業の Paragraph Vector との間のコサイン類似度を計算する．このとき類似度が最大となる候補を TOP-1 として決定する．最終的に，この類似度が閾値を上回ればリンキングし，下回る場合はリンキングなしとする．

4. 評価実験

4.1 データセット

提案手法の有効性を確認するため，表 1 に示すような評価用データを手作業で 200 件作成した．企業名が「-」の部分は，メーカー名に対応する公開企業データが存在しないことを示す．例えば 1 行目のメーカー「パイオニア」は上場企業である「パイオニア」が対応するが，2 行目の「パイオニア」はワッペンを主力製品とする非公開企業であり，対応する公開企業データは存在しない．同様にメーカー「アサヒビール」は企業「アサヒグループホールディングス」と対応するが，「アサヒシューズ」は靴メーカーであり企業「アサヒグループホールディングス」とは無関係である．テストデータ 200 件中，対応企業有りのメーカー数は 142 件，対応企業なしのメーカー数は 58 件であった．また，メーカー名と企業名が完全一致するペアは 113 件，部分一致は 63 件，不一致は 24 件であった．

4.2 実験方法

実験では，(1)「テキストマッチング」，(2)「Doc2Vec1 段」，(3)「Doc2Vec2 段」により構築したシステムのマッピング精度を比較する．

表 1 評価用データセット例

メーカー名	正解企業名	文字列一致度	関係性
パイオニア	パイオニア	完全一致	同一企業
パイオニア	－	(完全一致)	関係性無し
アサヒビール	アサヒグループホールディングス	部分一致	子会社
アサヒシューズ	－	(部分一致)	関係性無し
麒麟麦酒	キリンホールディングス	不一致	孫会社
ポニーキャニオン	フジ・メディア・ホールディングス	不一致	連結子会社

表 2 各手法による精度の比較

リンキング手法		精度(%)
テキストマッチング		52.5
Doc2Vec1 段	Paragraph Vector	46.0
	Word Vector	71.5
Doc2Vec2 段	Paragraph Vector（1 段目） Word Vector（2 段目）	61.5
	Word Vector（1 段目） Paragraph Vector（2 段目）	78.5
	Word Vector（1 段目） Paragraph Vector（2 段目） 2 段目企業名除去	**83.5**

[1] https://product.rakuten.co.jp/

表 3　誤り例の分析

メーカー名	正解企業名	予測企業名	類似度①	類似度②	備考
エスピーケイ	SPK	- (小林製薬)	0.41	0.16	1段目で小林製薬が候補に選ばれた. 2段目では類似度②が閾値 0.18 以下より, 「対応企業無し」と予測
ソニー・ ミュージックダイレクト	ソニー	- (ソニー)	0.91	0.09	1段目でソニーが候補に選ばれた. 2段目では類似度②が閾値 0.18 以下より, 「対応企業無し」と予測

ここで, 先頭 5 文字一致によるテキストマッチングをベースラインとした. すなわち, メーカー名の先頭 5 文字が, 企業名に含まれる場合にメーカーと企業をリンキングする. リンキング候補となる企業が複数存在する場合は, 最初に出現した候補をリンキングする. 閾値は各アルゴリズムで最良の精度となる閾値を探索的に求めた. 予備実験から, 2 段構成のシステムにおける 1 段目における最適な TOP-N 数は 1 とした.

4.3 実験結果

実験結果を表 2 に示す. まず Doc2Vec1 段では, Paragraph Vector による手法が 46.0%とベースラインの 52.5%を下回ったのに対して, Word Vector による手法は 71.5%であり, ベースラインより 19.0 ポイント高い精度を示した. また Doc2Vec2 段では, 1 段目に Word Vector, 2 段目に Paragraph Vector を用いた手法は 78.5%であり, Doc2Vec1 段と比較して更に 7.0 ポイント向上したことから, Doc2Vec を多段に使用することの有効性が確認された. 最も精度が高かったのは, Doc2Vec2 段, かつ2段目のモデルを企業コーパス②で学習させた場合の 83.5%であり, ベースラインと比較して精度が 31.0 ポイント向上した. この結果から, 仮説で示した抽象度の異なる情報を組み合わせることの有効性が確認された.

4.4 考察

実験結果において最も精度が高い手法について, 33 の誤り例を分析した. まず, 対応する企業が存在するにもかかわらず 1 段目で TOP-N 圏外となった誤り例が 10 例存在した. 残りの 23 例は 2 段目での順位づけまたは閾値による対応づけで失敗していた. それぞれの誤り例を表 3 に示す. ここで類似度①は Word Vector による類似度を, 類似度②は Paragraph Vector による類似度を示す.

まず表 3 の 1 段目の誤り例では, 正解は企業「SPK」であるのに対し, 1 段目では誤って「小林製薬」と予測した. Wikipedia で小林製薬のページを確認すると, 小林製薬には関連企業として「エスピー・プランニング株式会社」があり, 「エスピー」が重要な特徴として予測に影響を与えた可能性が考えられる. 実際, 「エスピーケイ」を本研究で用いた形態素辞書を用いて改めて分かち書きすると, 「エスピー」と「ケイ」に分かれることを確認した. 表記揺れに対しては逐次ユーザー辞書に追加することで回避可能だが, 辞書の構築と維持にはコストがかかる. 効率の良い辞書構築法や固有表現の抽出法の検討は今後の課題である.

2 段目の誤り例では, 正解は企業「ソニー」であり, 1段目では正しく TOP-1 に選出されたものの, 2 段目では Paragraph Vector の類似度が 0.08 と非常に低くなり, 閾値以下となったことから対応企業無しと誤判定された. これは, (1)ソニーは企業規模が大きく, 事業が多角化されていること, (2) ソニー・ミュージックダイレクトの商品は主に CD 等音楽関連商品に限定されること, (3)ドキュメントのソースがメーカー側と企業側で異なる(メーカー側は商品情報が主であり, 企業側は企業情報が主である), の大きく3点の理由から, 企業「ソニー」を代表するコーパスと, メーカー「ソニー・ミュージックダイレクト」のコーパスとの間の語彙分布に大きな差異が生じ, 結果として Paragraph Vector の類似度が低い値になったと推測される. このようにコーパスの性質の違い等に起因する語彙分布の差異をどう吸収するかも, 今後の課題である.

5. 終わりに

本研究では, 楽天市場の商品のメーカーとその企業との自動マッピングに取り組んだ. 役割の異なる Doc2Vec モデルを多段に重ねて複数の観点からリンキングの正当性を検証する提案手法は, 用語の多義性の解消に対して有効であることを示した. さらに, この手法は辞書による語の拡張, 教師データの作成といった人手を要する工程を必要とせず, コスト面の優位性が大きい. 今後は, 一般的なデータセットを使用した本手法の有効性の確認や, より適切な固有表現抽出, コーパスの差異にロバストな手法の開発に取り組む予定である.

参考文献

[1] R. Mihalcea , A. Csomai, “Wikify!: linking documents to encyclopedic knowledge,” Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, 2007.

[2] W. Shen, J. Wang , J. Han, “Entity linking with a knowledge base: Issues, techniques, and solutions,” IEEE Transactions on Knowledge and Data Engineering, 2015.

[3] X. Ling, S. Singh , D. S. Weld, “Design Challenges for Entity Linking,” Transactions of the Association for Computational Linguistics, 2015.

[4] 古川竜也, 相良毅 , 相澤彰子, “言語横断エンティティリンキングのための語義曖昧性解消,” 情報知識学会誌 24.2, 2014.

[5] S. Zhou, N. Okazaki, K. Matsuda, R. Tian , K. Inui, “Supervised Approaches for Japanese Wikification,” Journal of Information Processing 25, 2017.

[6] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado , J. Dean, “Distributed Representations of Words and Phrases and their Compositionality,” Advances in neural information processing systems, 2013.

[7] Q. Le , T. Mikolov, “Distributed Representations of Sentences and Documents,” International Conference on Machine Learning, 2014.

[8] T. Kudo, “MeCab : Yet Another Part-of-speech and Morphological Analyzer,” IEICE technical report 110(85), 2006.