4B1-4

大規模コーパスを用いたWord2vecによる比喩の書き換え

Metaphor transcription with Word2vec using a large corpus

冨山翔司 *1　　松尾豊 *2
Jouji Toyama　　Yutaka Matsuo

*1東京大学　松尾研究室
The University of Tokyo, Matsuo Laboratory

Metaphor understanding by computer is the important problem for NLP. For example, when the sentence "My boss is evil." is given, it is better to understand the sentence "My boss is scary." than to understand literally. The effort to understand the metaphor by computer is also important in terms of cognitive linguistics.

In this paper, we transcript sentences like " A is like B" into easy sentences e.g. "My boss is like evil." into "My boss is scary.". Previous research did the transcription by using hand-written knowledge, but we use Word2vec to get meaning of words from a big corpus so that our method does not need any hand-written knowledges.We also use Recurrent Neural Network Language Model to consider the context around the metaphor sentence and aim to improve the accuracy of the transcription.

1. はじめに

計算機による比喩表現の意味理解は、自然言語処理において重要なタスクである。例えば「課長は鬼だ」という文を文字通りに理解するのか、比喩であるとわかったうえで「課長は怖い」と理解するかでは大きな差がある。比喩表現の自動的な理解は自然言語処理の主要な課題である機械翻訳や質問応答、センチメント分析等において不可欠 [1, 2] なものである。

本研究では喩辞と被喩辞が明示された比喩表現 (例：鬼のような心) が与えられた時、それを平易な文 (恐ろしい心) に戻す書き換えを行う。その際、人間の与える知識を一切用いず、大規模コーパス (Wikipedia) を使って単語の分散表現を Word2vec で学習することで、書き換えを行う。更に、Recurrent Neural Network Language Model(RNNLM) を用いることで、文脈を考慮し、書き換え精度の向上を狙う。今まで比喩の書き換えに必要な知識は、人間の手で作られた網羅性に乏しい単語の意味ネットワークを用いていた [3]。しかし本研究は大規模コーパスから単語の分散表現を学ぶことによって知識を獲得するため、全ての単語について書き換えを可能になった。また文脈を考慮した比喩の書き換えモデルも初めての試みである。書き換え可能な単語の網羅性を手に入れたことと文脈を考慮できるようになったのが、本研究の従来研究からの改善点である。結果から、Word2vec と RNNLM の、比喩の書き換えというタスクへの適用可能性を探った。

本研究は、自然言語処理という研究分野に対して以下のような貢献をした。まず、喩辞と被喩辞が明示された比喩表現の書き換えというタスクに対して、Word2vec を初めて用いたことである。本研究では Word2vec を適用することによって多くの単語が書き換え可能になったメリットとともに、Word2vec を適用したことによる問題点もみられた。次に、比喩の書き換えという古くからあるタスクに対して、最新の RNNLM の適用可能性を示したことである。結果はうまくいかなかったが、パラメータの調整や RNNLM モデルの拡張によって精度の向上の可能性が有る。

連絡先: 冨山翔司, 東京大学工学部松尾研究室, 世田谷区等々力 2-12-11,03-5758-2368,toyama.jouji@gmail.com

2. 関連研究

2.1 比喩に関する研究

比喩表現を平易な文に戻そうという比喩の書き換えの研究はあまりなされていない。比喩の書き換えには一般に構造化された意味データと、書き換えのアルゴリズムが必要であると言及されている [4]。比喩の書き換えには Ortony の比較理論 [5] が用いられることが多い。これは比喩が用いられている文章を構成する、Source(喩辞)、Target (被喩辞) の関係に言及したものであり、両者が共有する属性概念を探索し、最も距離が近いものを解とするものである。このアルゴリズムに従い、坂口は連想概念辞書 [3] を用いて比喩の書き換えを行った [6]。連想概念辞書とは人間が知識として保持している一般的な概念とその関係性について記述したデータであり、人との連想実験を通じて得られた刺激概念と連想概念の対、および両者間の距離が定義されている [6]。

2.2 分散表現

分散表現とは、単語をベクトルで表現する方法の一つである。1-of-K ベクトルのような、各単語に 1 つの計算要素（次元、ニューロン）を割り当てる表現方法を、局所表現と呼び、一方各単語が複数の計算要素で表現され、各計算要素は複数の単語の表現に関与するような表現方法を分散表現と呼ぶ [7]。

分散表現をえる代表的な手法としては、単語文脈行列や、単語文脈行列を特異値分解することによってより密で低次元なベクトルに圧縮した Latent Semantic Analysis(LSA)[8] などが挙げられる。2013 年には Mikolov が [9] で Skip-gram を用いたニューラルネットワーク言語モデルを提唱した。Skip-gram はある単語から前後の文脈にある単語を予測するようなモデルであり、隠れ層に単語を分散表現するベクトルが現れる。この手法は Word2vec として知られ、LSA と比べて様々なタスクで精度が高く、また計算量も少ない [10]。分散表現はこのように数多くの研究がなされているが、句や文を表現するときには、分散表現では構成性に欠けているとの指摘も受けている [11]。

3. 提案手法

3.1 対象とするタスク

本研究では「名詞 (喩辞) のような名詞 (被喩辞)」の形で表される比喩表現 (e.g.「鬼のような心」) について、喩辞を形容詞に変えるような書き換えを行う。喩辞を Source, 被喩辞を Target と定義する。この問題を解決することは、喩辞と被喩辞が明示された文章 (e.g. 「課長は鬼だ」) の比喩の書き換え問題の解決と同じである。一般に、比喩の書き換えには構造化された意味データと、書き換えのアルゴリズムの二つが必要であると言われている [4]。

3.2 背景知識の説明

本節では、本研究の提案手法に用いられている背景知識について説明する。

3.2.1 Word2vec

Word2vec は何万語と存在する単語を、任意の次元のベクトルで分散表現する手法であり、Mikolov によって提案された [10, 9]、ニューラルネットワーク言語モデルの一つである。Word2vec によって単語を分散表現することによって、単語同士の類似度をコサイン距離を計算することによって測ることができる。以下、Word2vec でよく用いられる Skip-gram ネットワーク構造について説明する。

Skip-gram は、ある単語 $w(t)$ を入力とし、その周辺の単語 $w(t-k), w(t-k+1), \cdots, w(t-1), w(t+1), ..., w(t+k)$ を出力（予測）するニューラルネットワークである。入力と出力はそれぞれ単語の 1-of-K ベクトルとなっている。隠れ層は任意の次元 m を持ち、入力層と隠れ層をつなぐ重み行列 W は K × m である。このニューラルネットワークを、式 1 を最大化するように、誤差逆伝搬法 [12] で学習する。

$$L = \frac{1}{T} \sum_{t=1}^{T} \sum_{-c \leq k \leq c, k \neq 0} \log p\left(w_{t+k} \middle| w_t\right) \tag{1}$$

3.2.2 RNNLM

Recurrent Neural Network(RNN) は、固定長のベクトルしか入力, 出力できなかった従来のニューラルネットワークの欠点を修正するため、可変長のベクトルに対しても入力、出力できるようになったニューラルネットワークである。主に音声認識や機械翻訳など、時系列データを扱うときに用いられ [13, 14]、様々なタスクにおいて高い精度をマークしている。

RNNLM は、RNN を用いて、単語もしくは文字を入力とし、出力が次に出る単語、もしくは文字になるように学習していくモデルである。例えば「吾輩は猫である」という文を RNNLM で学習する。すると t=1 の入力は”吾”となり、出力の正解は「輩」となる。そして t=2 の入力は「輩」となり、出力の正解は「は」となる。このようにして学習していき、重み W を誤差逆伝搬法によって学習していく。ある時刻 t の隠れ層が入力 (t) だけでなく、前の時刻隠れ層 (t-1) からも影響を受けているため、出力 (t) は t より前の時系列データの影響を受けることになる。これによって RNNLM ではある入力の次の単語もしくは文字を、その前の文脈を考慮して予測することが可能になる。

さらに今回は、長期依存を捉えるために Long-Short Term Memory(LSTM) を用いている。RNN は誤差逆伝搬で長期の系列をたどる際、重みを複数かけることによる勾配の爆発及び消滅が発生してしまう。これを防ぐために、誤差が余計な重みをかけずに伝搬する仕組みが LSTM である [15]。

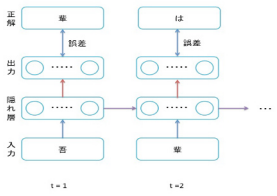

図 1: RNNLM

3.3 従来研究による問題の解決方法

従来研究 [6] では、連想概念辞書を用いて問題の解決を試みた。連想概念辞書は人間が知識として保持している一般的な概念とその関係性について記述したデータであり、連想実験を通じて得られた刺激概念と連想概念の対, および両者間の距離が定義されている。従来手法は、連想概念辞書を構造化された意味データとして用いた。

書き換えのアルゴリズムは、Source と Target と共通して関係するような形容詞を正しい書き換えとするという Ortony の比較理論を根拠としている。まず、ニューラルネットワークを設定して、ニューロンに語を割り当てて相互結合した。つぎに、連想概念辞書に定義された語間の距離を元に、ニューロン同士の結合強度を計算した。形容詞のニューロンと Source のニューロンとの結合強度、及び形容詞のニューロンと Target のニューロンとの結合強度から、形容詞のニューロンの活性値を計算し、活性値が高いものを書き換えの解とした。

従来手法は連想概念辞書に登録されている単語しか書き換えができない。

3.4 本研究による問題の解決方法

本研究では、構造化された意味データによって定義される語間の関係性を、大規模コーパスから抽出する語間の類似度によって代用する。語間の類似度を抽出するためのモデルとして、Word2vec を用いる。Word2vec で構造化された意味データを代用することによって、意味データの作成を自動的に行うことが可能となる。

本研究の書き換えのアルゴリズムも、従来研究と同じく Ortony の比較理論を元としている。全体のプロセスは図 2 のようである。まず、Source と Target と類似度が高い形容詞をいくつか抽出する。つぎに、形容詞と Target をクエリとして Web 上でアンド検索をし、検索件数を元に計算した値を計算する。最後に、Source を形容詞に置き換えた時、形容詞が前後の文脈と適合するかどうかを計算する。この三つの値をかけあわせたものが、形容詞が最終的に持つ値であり、一番値が大きい形容詞を書き換えの解とした。アルゴリズム全体としての入力は Source,Target, 文脈となり、そこから形容詞と値のペアを出力する。

以下の 4 節では、上記 3 つのステップ及びにそれらを合算して最終的な形容した選択するまでのプロセスを詳細に説明する。

3.4.1 Source と Target に類似する形容詞の抽出

Source と Target に類似する形容詞の抽出を、Word2vec を用いた Source と形容詞の類似度の計算及び Target と形容詞の類似度の計算によって行う。本節は図 2 の①に該当する。Word2vec を使った理由は前述の通り、高い分散表現能力を持ち、語の類似関係をうまくモデリングできるからである。

Word2vec に用いるコーパスは日本語版 Wikipedia の全文

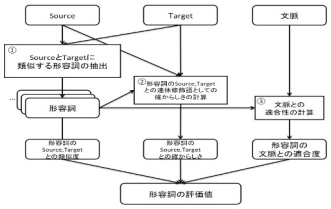

図 2: 本研究の比喩の書き換えのプロセス

(jawiki-latest-pages-articles.xml.bz2) を用いた。コーパスは元データを形態素解析によって、名詞、動詞、形容詞のリストにすることによって作成した。

つぎに、Word2vec のネットワーク構造について説明する。Word2vec のニューラルネットワーク構造は Skip-gram を用いた。出力は入力単語の前後 5 単語にし、隠れ層は 200 次元にした。

最後に Source と Target に類似する形容詞の抽出方法を説明する。まず、Source と他単語の類似度を、分散表現した Source と単語のコサイン類似度によって計算し、上位 30000 単語を抽出した。このうち、形容詞だけを抽出して、これらを形容詞群とした。それぞれの形容詞は Source との類似度 ($Similarity\_S$) を持つ。つぎに、形容詞と Target の類似度 ($Similarity\_T$) をコサイン類似度によって計算する。形容詞は $Similarity\_S$ と $Similarity\_T$ を掛け合わせた値 (図 2 の「形容詞の Source,Target との類似度」に該当) を持ち、上位 1 5個の形容詞を抽出した。これら形容詞を $adj_i\,(1 \leq i \leq 15)$ とする。

3.4.2 形容詞の連体修飾語としての確からしさの計算

本節では形容詞の、Source 及び Target との連体修飾語としての確からしさの計算について説明する。本節は図 2 の②に該当する。「形容詞　Source(Target)」の確からしさの計算を行うために、本研究は Web 検索をする。Web 検索をする目的は、抽出した形容詞と Source 及び Target との共起頻度を Web 上のすべての文書に対して測ることで、明らかにおかしい形容詞を排除することができるということにあるため、式 2,3 には上限値を設定している。

「Source 形容詞」と「Target 形容詞」をそれぞれクエリとして bing で検索し、検索件数を取得した。adj_i に対する $Count_S_{adj_i}$及び $Count\_T_{adj_i}$ を以下のように定義した。これを、形容詞の Source,Target との確からしさとする。関数 $bing(query1, query2)$ は、「$query1$ $query2$」に対する bing による検索件数を返す。

$$Count\_S_{adj_i} = \begin{cases} \frac{bing(Source, adj_i)}{\sum_i bing(Source, adj_i)} & (Count\_S_{adj_i} < 0.05) \\ 0.05 & (otherwise) \end{cases} \quad (2)$$

$$Count\_T_{adj_i} = \begin{cases} \frac{bing(Target, adj_i)}{\sum_i bing(Target, adj_i)} & (Count\_T_{adj_i} < 0.05) \\ 0.05 & (otherwise) \end{cases} \quad (3)$$

3.4.3 文脈との適合性の計算

Source を形容詞に置き換えた時、その形容詞と文脈との適合性を RNNLM によって計算する。本節は図 2 の③に該当する。

文脈の適合性を、RNNLM による Source の場所での形容詞の出現確率によって計算する。まず、Source より前の部分の文脈を形態素解析によって名詞、動詞、形容詞、助詞のリストにする。そして、順次 Forward RNNLM に読み込ませる。リストの最後の語を入力した時の出力によって、3.4.1 で抽出した形容詞がどれくらいの確率であるかをみる。この確率を $p_{for}\,(adj_i)$ とする。これと同様にして、Source より後の部分の文脈を、文の末尾から形態素解析によって名詞、動詞、形容詞、助詞のリストにする。そして、順次 Backward RNNLM に読み込ませる。そして前と同様にして Source の場所の形容詞の出現確率を $p_{back}\,(adj_i)$ とする。形容詞の文脈との適合度を式 4 のように定義する。

$$p\,(adj_i) = p_{for}\,(adj_i) + p_{back}\,(adj_i) \quad (4)$$

3.4.4 評価値の計算及び形容詞の選定

今まで説明した形容詞の Source と Target との類似度、形容詞の Source,Target との確からしさ、形容詞の文脈との適合度から、図 2 の一番下のように形容詞が式 5,6 のような評価値を持つようにした。なお、文脈との適合度を含めた (RNNLM 有り) 評価値と含めない評価値を二つ用意し、4 章でこの二つの評価値による書き換え結果を比べる。文脈との適合度を含めない評価値では、Web 検索は行っている。

$$V_{adj_i} = Similarity\_S_{adj_i} \times Similarity\_T_{adj_i} \times \left(Count\_S_{adj_i} + Count\_T_{adj_i}\right) \quad (5)$$

$$V^R_{adj_i} = Similarity\_S_{adj_i} \times Similarity\_T_{adj_i} \times \left(Count\_S_{adj_i} + Count\_T_{adj_i}\right) \times p\,(adj_i) \quad (6)$$

この評価値によって選ばれた上位 3 つの形容詞を最終的な書き換え結果とする。

4. 実験

本章では、実験に用いたテストデータとその正解、実際の比喩表現の書き換えを行なった結果、そして提案手法の精度を説明する。

4.1 テストデータ

テストデータとして、青空文庫に収録されている小説の中から「名詞のような名詞」にあてはまるものを 20 個選んだ。表 1 が小説の一覧及び今回用いたテストデータの一部である。実際のテストデータはこの比喩表現が用いられた文脈も含む。正解データは表 2 のように定義した。語には多くの同義語が存在するため、同じような意味の言葉であれば正解とする。また、正解が複数あるときは、すべて正解とする。

表 1: テストデータ概要

番号	比喩表現	小説	作者
1	油のような海	青鬼の褌を洗う女	坂口安吾
2	筆のような脚	平馬と鶯	林不忘
3	鬼のような心	煩悩秘文書	林不忘
4	鬼のような顔	斜陽	太宰治

表 2: 正解データ

番号	比喩表現	比喩表現の書き換えの正解	文脈を考慮した正解
1	油のような海	穏やかな	穏やかな
2	筆のような脚	細い	細い
3	鬼のような心	恐ろしい	恐ろしい
4	鬼のような顔	恐い、赤い	赤い

4.2 比喩の書き換え結果

比喩の書き換えの結果を、表 3 に示す。正解データと一致したものを太字で表している。形容詞は評価値の高い順に並べている。20 のテストデータのうち、書き換えの第一候補結果から読み取れたことは以下の通りである。

同一属性の形容詞が出ることが多い…これは Word2vec によって同一属性のものが同じようなベクトル空間に配置されることに起因する。例えば色に関する形容詞はすべて同じような位置に存在してしまう。それゆえ書き換え結果でもその影響が出てしまった。

類似度では抽出できない形容詞がある…1 の「油のような海」などから「穏やかな」を抽出するためには、油のドロドロした感じといった認知的な知識に基づく連想がないと抽出が難しい。

4.3 手法の精度評価方法の説明と文脈適合度の有無の比較

RNNLM の有無による精度の比較が表 4 である。ここで、$Accuracy_S$ は表 2 の正解との精度評価であり、以下のように評価する。書き換え結果として三つの形容詞があるが、評価値が最も高いものが正解であれば 3 点を与え、二番目に高いものであれば 2 点、三番目に高いものであれば 1 点を与える。正解とは違うが妥当性があるもにには半分の点数を与える。 正解とは違うが妥当性があるものには半分の点数を与える。合計の点数を全てが正解であった時の点数（今回であれば 120) で割った値を $Accuracy_S$ とする。一方、$Accuracy_C$ は、表 2 の文脈を考慮した正解との精度評価であり、$Accuracy_S$ と同じように評価する。両者の精度評価においても、文脈適合度を含まないほうが結果が良い。これの原因は、RNNLM のハイパーパラメータの調整の難しさや、学習データとテストデータが違うことがあげられる。

表 3: 比喩の書き換え結果

番号	比喩表現	書き換え結果	RNNLM 有り
1	油のような海	温い, 少ない, 安い	少ない, 速い, 薄い
2	筆のような脚	**細い**, **ほそながい**, 弛い	**細い**, 弛い, 拙い
3	鬼のような心	**恐ろしい**, **悪い**, 憂い	**恐ろしい**, 古い, **悪い**
4	鬼のような顔	黒い, **気味が悪い**, **恐い**	黒い, 怪しい, **恐い**

表 4: 精度比較

書き換え手法	$Accuracy_S$	$Accuracy_C$
RNNLM 無し	0.417	0.242
RNNLM 有り	0.320	0.183

5. 考察

比喩の書き換えの精度をよりあげるには、Source と類似する形容詞を属性ごとにうまく抽出する必要がある。例えば、鬼と類似する形容詞を“色”という属性、“心理的印象”という属性、“身体的印象”という属性から各々抽出するということである。これは [3] や [16] などでは属性別に単語の特徴的な属性を記述していたため実現できていたが、Word2vec では語の属性情報や階層性をコーパスから抽出することができない。二章で述べた、分散表現の構成性がないという欠点がまさにここに出ている。

本研究では「名詞のような名詞」に限定して提案手法を適用したが、これは汎用性にかける手法である。人は喩辞が出現しない比喩表現も理解が可能であり (e.g. 「卒論」を「離婚届」と理解する)、そのような書き換えを可能とするのはこれからの課題である。

今後の課題として、単語の意味を周囲の文との関係性から抽出するだけでなく、画像などといった認知的な情報と

参考文献

[1] 内海彰. 比喩理解への計算論的アプローチ. 認知科学, Vol. 20, No. 2, pp. 249–266, 2013.

[2] Ekaterina Shutova, Simone Teufel, and Anna Korhonen. Statistical metaphor processing. *Computational Linguistics*, Vol. 39, No. 2, pp. 301–353, 2013.

[3] 石崎俊, 岡本潤, 寺岡丈博. 自然言語処理と常識の使用–人間の連想に基づく常識の抽出 (産業日本語関連). *Japio year book*, Vol. 2009, pp. 122–127, 2009.

[4] 内海彰. 比喩の認知/計算モデル. *Computer Today*, Vol. 96, No. 3, pp. 34–39, 2000.

[5] Andrew Ortony. Beyond literal similarity. *Psychological review*, Vol. 86, No. 3, p. 161, 1979.

[6] 坂口琢哉. 連想概念辞書のニューラルネットワークへの符号化と比喩理解システムへの応用. 安田女子大学紀要, Vol. 38, pp. 169–179, 2010.

[7] Geoffrey E Hinton. Distributed representations. 1984.

[8] Scott C. Deerwester, Susan T Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *JAsIs*, Vol. 41, No. 6, pp. 391–407, 1990.

[9] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119, 2013.

[10] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[11] Jerry A Fodor and Zenon W Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, Vol. 28, No. 1, pp. 3–71, 1988.

[12] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Cognitive modeling*, Vol. 5, p. 3, 1988.

[13] Alan Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 6645–6649. IEEE, 2013.

[14] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[15] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. *Neural computation*, Vol. 12, No. 10, pp. 2451–2471, 2000.

[16] 徳永健伸, 寺井あすか. 比喩理解のための言語処理. 月刊 「言語」, Vol. 37, No. 8, pp. 46–53, 2008.