博士論文
単語間意味関係知識の獲得と応用

鷲尾光樹

目次

第 1 章

はじめに

単語間意味関係とは，単語・概念間にある関係のことであり，同義関係（*car-auto*），上位下位関係（*animal-cat*）などが含まれる [52]．これらの知識は自然言語理解において重要な要素であり，含意関係認識 [14] や質問応答 [28] などの高度な意味処理を含むタスクで用いられる．

自然言語処理における単語間意味関係知識は主に，WordNet[51] に代表される語彙知識ベースに蓄えられ，様々なタスクで利用される．WordNet は専門家が人手で作成した語彙知識ベースであり，上位下位関係，部分全体関係などの多様な意味関係が記述されている．

しかし，人手で作成される語彙知識ベースはカバーされているドメインも限られており，拡張にも大きなコストがかかる．その結果，語彙知識ベースを用いた自然言語処理システムが，未知語や新語に対応できないといった問題が生じてしまう．この問題を解決するために，コーパスから単語ペアの意味関係知識を自動的に獲得する技術が研究されている．また，近年ではニューラルネットワークモデルが様々な自然言語処理タスクで性能を向上させており，ニューラルネットワークに知識を挿入する場合は，低次元で密なベクトルの形式で知識を表現するのが有力なアプローチである．よって，コーパスから獲得された知識は，適切に単語間意味関係を表現しており，さらに知識の表現は後続のタスクで扱われる埋め込みの形式で獲得されることが望ましい．しかし，**既存の獲得法は，意味関係表現としての適切さと表現できる単語ペアの範囲の広さが両立していないという問題**がある．また，単語間意味関係知識は高度な意味処理に必要であるが，このような知識をどのようにシステムに組み込むかは自明ではなく，**獲得された意味関係知識を新たなタスクにどのように応用し，単語間意味関係知識の応用の幅を広げるか**，それ自体も問題である．

本研究では，これらの問題を解決するために，二つの研究に取り組んだ．一つ目の研究は，コーパスからの意味関係知識獲得の研究としての，関係パタンを用いた単語ペア埋め込みの教師なし学習である．**単語ペア埋め込み**は単語間意味関係知識を低次元で密なベクトルとして表現したものである．**関係パタン**とは文中で共起した二語を結びつける単語系列，あるいは依存構造パスであり，単語間の意味関係を表現する上で重要な特徴である．埋め込みの形でコーパ

スから得られる代表的な表現としては，二語の単語埋め込みの引き算によって単語間の関係を表現する方法があるが，この手法では，従来より意味関係に関する情報を持つとされてきた関係パタンの情報を扱えない．適切な単語間意味関係知識を表現するためには，関係パタンの情報を用いるのが望ましい．一方で，関係パタンを用いて単語ペアの表現を得る方法では，実際にコーパス上で共起した単語ペアについてしか表現を獲得できず，表現が得られる単語ペアが大幅に制限されてしまう．本研究では，コーパスから自動的に意味関係知識を獲得するために，関係パタンとの共起から，ニューラルネットワークを教師なし学習し，単語ペアの意味関係を適切に表現する埋め込み表現を獲得する方法を提案する．なるべく多くの単語ペアについて関係パタンの情報を扱って意味関係知識を適切に表現する埋め込みを獲得するために，ニューラルネットワークの汎化能力を利用する．先行研究 [76] では，関係パタンを用いた場合，コーパス上である程度共起した単語ペアのみにしか埋め込みを得ることができなかったが，ニューラルネットワークを用いることで，単語埋め込みが割り当てられている任意の単語ペアについて，関係パタンの情報を捉えた単語ペア埋め込みを獲得することができる．コーパスからの教師なし学習とニューラルネットワークによる汎化を組み合わせることで，関係パタンを用いた単語間意味関係知識のカバレッジが向上する．単語ペア埋め込みの学習法の評価は，単語ペアの意味関係類似度の評価ベンチマーク [40] と，意味関係識別の性能 [71, 70] で行い，有効性を確認した．

二つ目の研究は，単語間意味関係知識の新たな応用である．本研究では，単語ペア埋め込みを定義文処理 [56, 75, 8] に応用する手法を提案する．定義文処理は，定義文からの良質な単語埋め込みの獲得や，単語埋め込みからの定義文の生成を含む，定義される見出し語と定義文間のマッピングをモデリングするタスクである．情報検索や含意関係認識，質問応答などで単語間意味関係知識を用いる研究は存在するが，定義文処理に適用する研究は存在しない．定義文には見出し語と意味関係を有する単語が存在し [1]，これらをモデリングに組み込むことで，より適切な定義文処理が期待できる．先行研究 [56, 8] では，リカレントニューラルネットワーク（RNN）を用いてモデリングを行っているが，本研究では見出し語と定義文内の各語の意味関係に着眼し，RNN の学習時に単語ペア埋め込みを用いることで，定義文からの単語埋め込みの獲得と定義文生成の性能が向上することを示す．

以上の二つの研究を通して，本博士論文では，コーパスから単語間意味関係知識を自動獲得し，それを自然言語処理に活かすことの重要性を示す．

第 2 章

背景

この章では，本博士論文における研究の背景知識や関連研究について述べる．

2.1　分布仮説に基づく単語の表現

自然言語処理では，文章を処理する上での基本単位である単語について，それらをどう表現すればいいのかを模索してきた．本節では現在，単語を表現する上で主流の方法である，分布仮説に基づく単語のベクトル表現について述べる．

分布仮説とは，似た意味を持つ単語は似た出現文脈に現れる傾向があるという経験則である [30]．分布仮説に基づき，コーパス上で表現を得たい単語の周辺に出現した単語を用いて，ベクトルの形で単語を表現することで，似た意味を持つ単語がベクトル空間上の似た位置に配置されるように，単語の意味を表現することができる [80]．以降では単語のベクトル表現を単語ベクトルと呼ぶ．

出現文脈との共起頻度を集計し，数万次元のスパースな共起頻度ベクトルを単語に割り当てるのは古典的な手法であり，コサイン尺度等の指標により，単語間の類似度を適切に捉えることができる．

近年では，ニューラルネットワークなどの機械学習モデルで用いることを考慮して，単語に低次元で密な埋め込み表現を割り当てるのが主流である [50, 58, 42]．この方法は，共起頻度ベクトルのように，出現文脈との共起をそのまま単語ベクトルにするのではなく，各単語にパラメータを割り当てておき，コーパス上の出現文脈との共起を訓練データとして，出現文脈との共起を予測するようにパラメータを学習する．学習の結果得られる，数百次元のベクトルが，そのまま単語ベクトルとなる．低次元で密な単語埋め込みとして，単語を表現することで，単語の情報をそのままニューラルネットに順伝播させることができる．

単語埋め込みは以下のようにコーパスからの教師なし学習[*1]によって学習される．単語 $w \in W$ と文脈 $c \in C$ があるとき，教師なし学習のデータ $D = ((w_1, c_1), \ldots (w_n, c_n))$ は，コーパス上での単語と文脈の共起を集めたものである．単語埋め込みは共起データ D から，基本的には以下のようなプロセスを経て学習される．

1. 各単語 w と各文脈 c にそれぞれ，何らかの方法で d 次元のパラメータベクトル $\boldsymbol{w}, \boldsymbol{c}$ を割り当てる．このとき，$\boldsymbol{w}$ が w の単語埋め込みとなる．
2. 分布仮説に基づく目的関数を設計する．
3. 設計した目的関数と D に基づき，各単語・文脈に割り当てられたパラメータベクトルを最尤推定する．

分布仮説に基づく目的関数 L としては，文脈 c に単語 w が出現する確率をモデル化した以下のようなものが考えられる．

$$L = \sum_{(w,c)\in D} \log P(w|c) \tag{2.1}$$

$$= \sum_{(w,c)\in D} \log \frac{e^{\boldsymbol{w}\cdot\boldsymbol{c}}}{\sum_{w_k \in W} e^{\boldsymbol{w}_k \cdot \boldsymbol{c}}} \tag{2.2}$$

ただし，e はネイピア数である．式 (2.2) の分母は数万から数百万の単語数分の内積計算と指数計算を含み，計算が重たいため，以下のような負例サンプリング目的関数 [50] が提案されている．

$$L_{Neg} = \sum_{(w,c)\in D} \left\{ \log P((w,c) \in D) + \sum_{(w,c')\in D'_{(w,c)}} \log P((w,c') \notin D) \right\} \tag{2.3}$$

$$= \sum_{(w,c)\in D} \left\{ \log \sigma(\boldsymbol{w}\cdot\boldsymbol{c}) + \sum_{(w,c')\in D'_{(w,c)}} \log \sigma(-\boldsymbol{w}\cdot\boldsymbol{c'}) \right\} \tag{2.4}$$

ただし，$D'_{(w,c)}$ は各 $(w,c) \in D$ に対してランダムに生成された負例サンプルであり，σ はシグモイド関数である．

単語埋め込みの代表的な獲得法である Skipgram モデル [50] では，文脈とは，単語の前後数語の範囲に現れた単語である．よって c は単語を表す．たとえば，コーパスに *The quick brown fox jumps over the lazy dog ...* という単語列があり，*jumps* という単語に注目するとする．このとき，前後 2 語を文脈としてみなすとき，$(jumps, brown), (jumps, fox), (jumps, over), (jumps, the)$ が D に追加される．このように D を構築したのち，前述の学習法によって単語埋め込みを学習する．

[*1] ここでいう教師なし学習とは，人手による注釈を介さずに得られるデータを用いた機械学習のことである．

単語埋め込みは文脈の取り方や学習法によって様々なバリエーションが存在する．CBOW[49] は文脈を周辺に現れた文脈単語の埋め込みの平均として表現する．Levy らは文中で依存関係にある語を文脈とする手法を提案した [42]．GloVe[58] は単語埋め込みと文脈埋め込みの内積から共起頻度を直接予測することで，コーパスの全体の統計量を用いて単語埋め込みの学習を行う．

2.2 単語間意味関係とは

単語間意味関係とは，二つの単語（あるいは概念）の間に成り立つ意味的な関係である．単語間の関係には，意味関係だけでなく，屈折や派生を含む統語的・形態的な関係（e.g. *drink-drank*）等の語彙的な関係や，*Paris-France* のような固有名詞間に成り立つ首都-国の関係等の百科事典的な世界知識 [6] も存在するが，本研究では質問応答や含意関係認識等の高度な意味処理を含む自然言語処理において重要な，一般的な言語知識としての単語間の意味関係に焦点をあてる．

語彙意味論における代表的な意味関係として，以下のものがあげられる [52]．

- 同義関係 (synonymy, e.g. *couch-sofa*)
- 上位下位関係 (hypernymy/hyponymy, e.g. *animal-cat*)
- 兄弟関係 (co-hyponymy, e.g. *dog-cat*)
- 対義関係 (antonymy, e.g. *cold-hot*)

同義関係とは，(ほぼ) 同じ意味を持つ二つの単語間に成り立つ関係である．上位下位関係は，二つの単語の外延に包含関係がある場合に成り立つ関係であり，上位語の外延が，下位語の外延を包含する．兄弟関係は，共通の上位語を持つ二語に成り立つ関係である．対義関係は兄弟関係の一種であるが，意味がある側面において反対の二語の間に成り立つ関係である．この他にも部分全体関係，因果関係などの様々な意味関係がある．

2.3 人手構築された言語知識における単語間意味関係

2.2 節で説明したような単語間意味関係知識は，機械可読な形式の語彙知識ベース（シソーラス）に蓄えられ，自然言語処理に用いることができる．語彙知識ベースにおいては，ノードが単語・概念を表し，ノードを結びつける意味関係ラベル付きのエッジによって意味関係が表現される．

WordNet[51] は人手で構築された英語の語彙知識ベースとしてもっとも普及しているものであり，意味関係の種類も豊富である．WordNet においては，ノードは同義語集合を表現しており，一つのノードに同じ意味を持つ複数の単語が割り当てられている．WordNet におい

ては，基本的に同義語集合間において意味関係が注釈されているが，対義関係などの一部の関係は単語間に注釈されている．

2.4 単語間意味関係の自動獲得

WordNet のような人手で構築される語彙知識ベースは，作成・拡張に大きなコストがかかるため，語彙のカバレッジに限界がある．そのため，コーパスから単語間意味関係知識を自動的に獲得する一連の研究がある．単語間意味関係知識の自動獲得の研究には二つの流れがある．一つは，WordNet などの既存の語彙知識ベースの拡張を目的として，単語ペアの意味関係の識別を行おうというものである．この研究では，単語ペアをあらかじめ定義された意味関係カテゴリに分類するという問題を解くことになる．もう一つは，そのような特定の目的から離れて，ベクトルの形で単語ペアの意味関係を表現するものである．単語ペアのベクトル表現を，似た意味関係を持つ単語ペアがベクトル空間上で近くなるように獲得すること自体がその目的となる．

本節は関係パタンと単語間意味関係知識の関係（2.4.1 節）について述べたのち，語彙知識ベース拡張のための意味関係識別（2.4.2 節）と単語間意味関係のベクトル表現（2.4.3 節）の先行研究について紹介する．

2.4.1 関係パタンと単語間意味関係

コーパスからの単語間意味関係知識の自動獲得では，コーパス上での単語ペアの共起文脈である関係パタンが有用な手がかりであるとされている．本研究における関係パタンとは，単語ペアが文中で共起したときに，二語の間に出現した単語系列や，あるいは二語を結びつける依存構造パスであると定義する．

このような関係パタンと単語間意味関係を結びつけた最初期の研究として，Hearst が上位下位関係を持つ単語ペアをコーパスから自動的に抽出するために関係パタンを用いている [31]．Hearst は上位下位関係知識を示唆すると考えられるいくつかのパタン（*Y such as X* や *X and the other Y*）を考え，ブートストラッピング法を用いてパタンを拡張しつつ，上位下位関係を持つ単語ペアを抽出している．このように，コーパスからの単語間意味関係知識の研究の初期は，上位下位関係や部分全体関係を示唆すると考えられる関係パタンを求める研究が盛んであった [66, 2, 25]．

Turney は，分布仮説と Hearst の知見を結びつけ，似た関係パタンで共起する単語ペアは似た意味関係を持つという**潜在関係仮説** [78] を提唱している．潜在関係仮説のもとでは，人間にとって解釈の難しい関係パタンも，何らかの形で単語ペアの意味関係を反映をしていると考えることができる．2.4.2 節や 2.4.3 節で，単語ペアを共起した関係パタンで表現する手法は，潜

在関係仮説を下敷きにしているとみなすことができる．

前述の通り，関係パタンは，文中で単語ペアが共起したときに二語の間に出現した単語系列か，二語を結びつける依存構造パスである．しかし，単語系列や依存構造パスがそれぞれどのような単語間意味関係を捉えているか，すなわち，関係パタンとして選択された文脈のそれぞれの長所・短所などを網羅的に比較した研究はなく，現時点ではそれらは明確ではない．類似タスクである情報抽出分野における関係抽出の研究では，固有名詞間の共起関係パタンを用いるが，依存構造パスを用いることで文中での長距離依存関係が捉えられるとされており [21]，これは関係パタンを用いた単語間意味関係知識の獲得においても当てはまると思われる．

2.4.2 語彙知識ベースの拡張のための意味関係識別

WordNet などの語彙知識ベースを拡張するために，コーパスから得られる情報を用いて，未知の単語ペアがあらかじめ定義された意味関係カテゴリ（上位下位関係など）に属するかを識別する一連の研究がある．訓練データを用いないアプローチと，語彙知識ベース内の既知の単語ペアを訓練データとして用いる教師あり学習のアプローチがあり，近年では教師あり学習の研究が多い．

訓練データを用いないアプローチでは，関係パタンによるルールベースの抽出や，二語の出現文脈の包含性を利用する研究がある．Hearst は X *is a* Y や Y *such as* X のような関係パタンで共起する単語ペアを上位下位関係として識別する手法を提案し，関係パタンの検索によって語彙知識ベースの拡張を試みた [31]．Weeds らは，上位語の出現文脈は下位語の出現文脈を包含するという分布包含仮説 [23] に基づき，二語の共起頻度ベクトルの各次元の値を比べ，上位下位関係と密接に関係する意味の包含性を測る手法を提案した [85].

教師あり学習のアプローチでは，単語ペアを何らかの方法で特徴ベクトルとして表現し，訓練データを用いて分類器を訓練する．単語ペアの特徴ベクトル化には，二語の単語ベクトルを用いるものと，コーパス上で共起した関係パタンを特徴として用いるものがある．

単語ベクトルを用いる手法では，二語の単語ベクトルの結合 [3, 62] や，ベクトルの差分 [63, 84, 82] によって単語ペアの特徴ベクトルを作る．しかし，これらの手法は二語の関係性を学習しているのではなく， 個々の単語がそれぞれどれくらい対象の関係を持ちやすいかを覚えているだけであるという報告がある [45].

関係パタンを用いる手法 [72, 71, 70] は，コーパス上で対象の二語と共起した，二語を結びつける単語系列や依存構造パスである関係パタンを特徴として，単語ペアを表現する．Hearst[31] が示すように，関係パタンは二語の意味関係を反映することが多く，分類器が二語の意味関係を学習できる [72]．しかし，単語系列・依存構造パスなどの系列を特徴に用いた場合，要素の組合わせの膨大さによって特徴空間の次元数が大きくなり，大抵の系列は一度しか出現しないため，特徴空間がスパースになってしまうという問題がある．Shwartz らは，

RNN を用いて，関係パタンを低次元で密な埋め込みに変換することで，この問題を緩和した [71, 70]．Shwartz らの手法については，3.6.1 節で詳細に述べる．

2.4.3 単語間意味関係のベクトル表現

もう一つの流れとして，単語ペアの意味関係を表現するベクトル表現をコーパスから学習する研究がある．前節で述べた語彙知識ベースの拡張のための意味関係識別においても，単語ペアを特徴ベクトルとして表現しそれを識別に用いるものがあったが，こちらの流れでの単語ペアの表現は，その目的にとどまらず，単語ペアをベクトルとして表現することを通じて，ニューラルネットワークなどの機械学習モデルに単語間意味関係知識を組み込むことを可能とし，様々なタスクに貢献することや，そもそも，意味関係が類似している単語ペアの表現をベクトル空間上で近づけること自体を目的としている．

このような単語ペアのベクトル表現としては，前節の教師あり意味関係識別と同様に，二語の単語ベクトルを用いるものと，関係パタンの情報を用いるものがある．前者の手法としては，コーパスから学習した二語の単語埋め込みの差分を用いる方法が，単純かつ強力な手法とされている [50, 89, 43, 82]．この手法では，単語ペア (a, b) を以下のように，埋め込みの差分で表現する．

$$\boldsymbol{v}_{(a,b)}^{vecoff} = \boldsymbol{v}_b - \boldsymbol{v}_a \tag{2.5}$$

ここで，$\boldsymbol{v}_a, \boldsymbol{v}_b$ は単語 a, b の単語埋め込みである．

単語埋め込みの差分を用いる手法は，二語の共起を必要とせずに単語ペアを表現できるが，関係パタンの情報を用いることができない．先行研究においては，単語埋め込み内の情報と関係パタンの情報は相補的であり，関係パタンで捉えれられる情報は単語埋め込みの空間では表現されていないことが示されている [45, 71]．これは，Levy らの分析によると，単語埋め込みは対象となる単語と出現文脈との共起から単語の表現を学習するが，このように学習された二つの単語の表現からは，二語が共起するときの出現文脈である関係パタンを復元できないためである [45]．よって、より良く意味関係を捉えるために、関係パタンを用いる動機づけがある．

関係パタンを用いる手法では，単語ペアと関係パタンの共起から単語ペアのベクトルを獲得する．Turney らは単語ペアの意味関係を表現するためのベクトル空間モデルとして，重要と思われる 64 個の関係パタンを特徴とした単語ペアベクトルを獲得した [79]．さらに Turney は，より多様な関係パタンを用いつつ，関係パタンに単語系列を用いる際のスパースネスの問題を緩和するために，潜在関係解析（Latent Relational Analysis, LRA）を提案した [76, 77]．これは単語ペアと関係パタンの共起頻度行列に特異値分解を適用して次元削減を行い，共起頻度行列の背景にある潜在的な構造を捉えた低次元で密な単語ペア埋め込みを得る手法である．LRA は，似た関係パタンで共起する単語ペアは似た意味関係を持つという潜在関係仮説に基づいている [80]．

LRA は対象となる単語ペアの集合 $W = \{(a_1, b_1), \ldots, (a_n, b_n)\}$ と対象となる関係パタンの集合 $C = \{p_1, \ldots, p_m\}$ について共起を集計する．関係パタンについては，一部の単語あるいはすべての単語をワイルドカードで置換することで，汎化を試みている．共起頻度行列 M の各行は単語ペア (a_i, b_i) あるいは (b_i, a_i) に対応し，各列はパタン Xp_iY あるいは Yp_iX に対応するため，行列のサイズは $2n \times 2m$ となる．ここで，X, Y は対象の単語ペアの各語に対応するスロットである．行列 M は正の自己相互情報量（Positive Pointwise Mutual Information, PPMI）で重み付けされたのち，特異値分解により次元削減され，各単語ペアに単語ペア埋め込み $\boldsymbol{v}_{(a,b)}^{LRA}$ が割り当てられる．

2.5 獲得された単語間意味関係知識の評価

獲得された単語ペアのベクトルの良さ，つまりそのベクトルが単語間意味関係を適切に表現しているかを測る方法として，主に以下の二つの枠組みがある．

- 意味関係類似性タスク
- 意味関係識別タスク

2.5.1 意味関係類似性タスク

意味関係類似性タスクでは，対象の意味関係カテゴリにある単語ペアがどれぐらい適合するか，システムの判断と人間の判断がどれぐらい強く相関しているかによって，システムが持つ意味関係知識を評価する [76, 40]．代表的なベンチマークとして，SemEval2012 Task2 データセットがある [40]．このデータセットでは，10 の大分類と 79 の小分類の詳細な各意味関係カテゴリについて，その意味関係を持ついくつかの典型的な単語ペアと，数十の候補単語ペアが割り当てられている．意味関係の小分類には，*Taxonomic* 関係（分類学的な上位下位関係）や *Sign-Significant* 関係（記号とその意味の関係），*Agent-Object* 関係（主体とその主体によって作られるモノやよく使われる道具の関係）などがある．たとえば，*Agent-Object* 関係では，*tailor:suit*，*oracle:prophesy*，*baker:flour* が典型ペアであり，意味関係カテゴリによく適合するものである．候補ペアには *Agent-Object* 関係に適合したり，しなかったりする様々な単語ペアが含まれている．適合の度合いの注釈は以下のように行われている．各意味関係カテゴリに対して，MaxDiff 質問法に基づいて，クラウドワーカーに 4 つか 5 つの候補ペアを提示する．クラウドワーカーはもっとも適合するペアともっとも非適合であるペアを選択してもらう．各質問には 5 人のクラウドワーカーが回答する．ある候補ペアの適合スコアは，適合するペアと判断されたパーセンテージと非適合であると判断されたパーセンテージの差であり，適合スコアの範囲は -100 から 100 の間である．

タスクとしては，それぞれのカテゴリについて，典型単語ペアと候補単語ペアの類似度計算に基づいて候補単語ペアをランキングし，人間のランキングとのスピアマンの順位相関を測ることで，システムの評価を行う．システムが単語ペアをベクトルとして表現している場合，単語ペア間の類似度計算は，基本的に，二つのベクトルのコサイン尺度が用いられる．

2.5.2 意味関係識別タスク

意味関係識別タスク [4, 3, 71] は，2.4.2 節で述べたように，単語ペアが持つ上位下位関係や兄弟関係などの適切な意味関係カテゴリを識別するタスクである．このタスクは語彙知識ベースの拡張という研究目的への貢献を直接評価するものである．アプローチとしては訓練データを用いた教師あり学習が多く，単語ペアを何らかの方法で特徴ベクトルとして表現し，訓練データによって分類器を獲得して，その性能によって単語ペアのベクトルの良さを評価する．教師あり意味関係識別のデータセットにおける事例は，単語ペアとその単語ペアが属する意味関係カテゴリのセットである．たとえば，以下で述べる K&H+N データセットには，それぞれ一例を挙げると，上位下位関係カテゴリに属する単語ペアとして *(pecan, plant)*，兄弟関係に属する単語ペアとして *(pineapple, strawberry)*，ランダムな関係（関係を持たない場合）に属する単語ペアとして *(tautog, projection)* が含まれている．標準的には以下の四つのデータセットが用いられる．

K&H+N[53] WordNet から上位下位関係・兄弟関係・部分全体関係にある語を抽出して構築したデータセットである．単語ペアの片方の単語をランダムに置き換えたランダムペアも負例として含まれている．

BLESS[4] 17 個のドメイン（爬虫類，衣類，道具など）の曖昧性の低い 200 個の名詞について，上位下位関係・兄弟関係・部分全体などの単語を集めたデータセットである．

EVALution[68] 語彙知識ベースである WordNet と，一般知識・常識の知識ベースである ConceptNet[73] から，単語間意味関係知識を抽出したデータセットである．意味関係カテゴリについては，クラウドソーシングによって人手で分類を行い，カテゴリに一貫性をもたせている．

ROOT09[67] BLESS や EVALution を含む複数のデータセットから，意味関係知識を集めて構築されたデータセットである．各意味関係カテゴリの事例数が等しくなるように調整されている．

いずれのデータセットにおいても，事例を訓練用・開発用・テスト用に分割し，訓練用と開発用で機械学習モデルの学習とハイパーパラメータの調節を行って，テスト用データセットで評価を行う．評価指標には多クラス分類性能を測る F 値等が用いられる．

2.6 単語間意味関係知識の利用

単語間意味関係知識は自然言語処理において意味が関わる様々な場面で必要となる．代表的なものとしては，以下のようなタスクがある．

- 情報検索
- 語彙的言い換え・語彙平易化
- 含意関係認識
- 質問応答・機械読解

情報検索においては，文書をデータベースから検索する際に，検索クエリを拡張するのに単語間意味関係知識を用いることができる [81]．クエリ内の単語の同義語や下位語をクエリに追加することで，より頑健な情報検索が可能となる．たとえば，*golf swing* のようなクエリで検索を行う際に，*swing* の同義語である *shot* や，下位語である *slice*，*drive* などを追加することで，元のクエリとの文字列マッチだけでは得られない文書を抽出できる．

語彙的言い換えや語彙平易化は，文脈を考慮しつつ文内の語句を（語彙平易化の場合は平易な）同義語や上位語に置き換えることで達成することができる [10, 5]．たとえば，Biran らは，分布仮説に基づき類似している単語ペア (a, b) を集め，b が a の WordNet における上位語か同義語である場合を，a から b に置換する語彙平易化規則として収集している [5]．平易化規則 (a, b) をある文の単語 a に適用する場合は，文内の単語と a，b が大規模コーパスにおいて共起した単語の一致度合いを計算し，それがしきい値を超えた場合に平易化規則を適用している．

含意関係認識と質問応答・機械読解は，文の関係性を捉えるタスクであり，いずれのタスクでも二文の間の単語ペアが持つ同義関係，上位下位関係などの意味関係知識が重要になる．含意関係認識では，前提文 *Tom has a jeep.* と仮説文 *Tom has a car.* の間の含意関係を捉えるには，*jeep* と *car* が上位下位関係にあることを知識として持っていなければならない [11]．質問応答においても，クエリである疑問文内の単語と，答えを含む文内の単語の意味関係知識が重要である．たとえば，*When were the "Game of Thrones" broadcasted?* という質問の答えが，*"Game of Thrones" were aired in 2011.* という文にあるとき，この文を特定する際は，*broadcast* と *air* の同義関係が大きな手がかりになる [29]．

以上のように，文字列マッチを超えた意味・概念に関わる情報を扱う場合は，単語間意味関係知識が役に立つ場面は多い．特に単語間意味関係知識は含意関係認識や質問応答・機械読解などの言語理解タスクにとって重要である．一般にこれらの言語理解タスクをモデリングするためのニューラルネットワークに対して，意味関係知識を挿入する際の知識の表現形式には以下の三つがある．

1. 語彙知識ベース内の関係の離散的表現（知識ベース記号表現）
2. 語彙知識ベースから得られる単語ペアの表現（知識ベース埋め込み表現）
3. コーパスから得られる単語ペアの表現（コーパス埋め込み表現）

一つ目の知識ベース記号表現の形式では，表現したい単語ペアについて語彙知識ベースを参照し，単語ペアが意味関係を持つか否かを離散的にニューラルネットワークに入力する．この形式における単語ペアの表現は，語彙知識ベース内の意味関係の種類分の次元を持ち，単語ペアが持つ意味関係の次元のみが 1 となり，その他の次元は 0 となる表現である．語彙知識ベース内の意味関係の種類数はたかだが数百であると思われ，数百次元のベクトルはニューラルネットワークの入力として扱うことができる．

二つ目の知識ベース埋め込み表現は，語彙知識ベースから何らかの方法で対象の単語ペアの埋め込みを獲得し，ニューラルネットワークに入力する方法である．語彙知識ベースからの埋め込み獲得は，グラフで表現される知識ベースから埋め込みを獲得する知識グラフ埋め込み [7, 55, 83] の手法を用いることができる．語彙知識ベースは頂点を単語，辺を意味関係カテゴリとするグラフとみなすことができ，一部の知識グラフ埋め込みの手法では，単語ペアの表現を獲得することが可能であり，これもニューラルネットワークに入力することができる．

三つ目のコーパス埋め込み表現は，2.4.3 節で述べたような単語埋め込みの差分や LRA，また，3 章で提案する手法など，コーパスから獲得される単語ペア埋め込みを用いる方法である．

Chen らは含意関係認識のニューラルネットワークモデルに，WordNet から得られる知識ベース記号表現を入力し，性能を向上させている [11]．Joshi らは，3 章で提案した枠組みで獲得したコーパス埋め込み表現を含意関係認識と機械読解のニューラルネットワークモデルに適用している [39]．

第 3 章

関係パタンとの共起を汎化する単語ペア埋め込みの教師なし学習

3.1　はじめに

2.6 節で述べたように，文字列マッチを超えた高度な意味処理を含む自然言語処理タスクにおいては，単語間意味関係知識は重要な要素である．近年では，含意関係認識や質問応答などのタスクにおいて，ニューラルネットワークを用いた機械学習モデルが高い性能を達成しており [12, 47]，機械学習モデルで扱いやすい低次元で密な埋め込みの形式で適切に単語間意味関係知識を表現する動機がある．また，既存の語彙知識ベースを拡張する教師あり意味関係識別（2.4.2 節）でも，意味関係を適切に表現する単語ペア埋め込みは，良い特徴ベクトルとして識別性能に貢献しうる．

2.4.3 節で説明したように，単語ペアの意味関係のベクトル表現としては二語の単語ベクトル，あるいは単語埋め込みの差分によって意味関係を表現する方法 [50] と，単語ペアと共起した関係パタンを用いて単語ペアをベクトル化する方法 [79, 76] が主流である．単語ベクトルの差分を用いる方法は，単純かつ強力な手法であるが，単語ペアの意味関係について重要な情報を持つ関係パタンの情報を用いることができない．一方で，関係パタンを用いた場合，単語ペアを表現するにはコーパス上での二語の共起が必要となる．しかし，この要請は，たとえ単語ペアが何らかの意味関係を持っていたとしても必ずしも満たされるわけではない．もし単語ペアの共起が得られなかった場合，そのペアに対しては埋め込みを割り当てることができなくなってしまう．この問題は 2.4.2 節で述べた意味関係識別における関係パタンを用いる手法でも同様であり，コーパス上で共起しなかった単語ペアについては，関係パタンの特徴を分類に用いることができない．この問題をパタン欠落問題と呼ぶことにする．

本研究では，この問題を解決するために，ニューラルネットワークを用いて，単語ペアと関係パタンの共起の汎化を行う単語ペア埋め込みの教師なし学習を提案する．ニューラルネット

ワークによって共起を汎化することで，コーパス上で共起が得られなかった単語ペアについても，関係パタンの情報を捉えた単語ペア埋め込み表現が得られる．

2.5 節で述べた意味関係類似性タスクと教師あり意味関係識別タスクのデータセットを用いた実験により，提案手法によって得られた単語ペア埋め込みを用いた場合，それぞれのタスクにおいて性能が向上することがわかった．さらに，分析によって，コーパス上で共起が得られなかった単語ペアについても，提案手法は適切に意味関係を表現し，パタン欠落問題を緩和していることがわかった．

3.2 関連研究

3.2.1 単語埋め込みによる単語ペア埋め込み表現

2.4.3 節で述べたようにコーパスから学習した二語の単語埋め込みの差分を用いる方法は，単純かつ強力な単語間意味関係知識の表現手法である [50, 89, 43, 82]．この手法では，式 2.5 で述べたように単語ペア (a, b) を以下のように，ベクトルの差分で表現する．

$$\boldsymbol{v}_{(a,b)}^{vecoff} = \boldsymbol{v}_b - \boldsymbol{v}_a \tag{3.1}$$

ここで，$\boldsymbol{v}_a, \boldsymbol{v}_b$ は単語 a, b の単語埋め込みである．

3.2.2 関係パタンを用いた単語ペア埋め込み表現

関係パタンを用いる単語ペア埋め込み表現として，Turney は潜在関係解析（LRA）を提案した [76]．2.4.3 節で述べたように，この手法は似た意味関係を持つ単語ペアは似た関係パタンと共起するという潜在関係仮説 [78] に基づき，コーパスから単語ペアと関係パタン（単語系列）の共起頻度行列を作り，特異値分解により次元削減することで，単語ペア埋め込みを得る．

LRA は，単語ペアの集合 $W = \{(a_1, b_1), \ldots, (a_n, b_n)\}$ と関係パタンの集合 $C = \{p_1, \ldots, p_m\}$ について共起を集計して得られる共起頻度行列 M について，PPMI で重み付けを行ったのち，特異値分解を施すことで得られる単語ペア埋め込み $\boldsymbol{v}_{(a,b)}^{LRA}$ を，各単語ペアに割り当てる．

2.4.3 節で述べたように，このような関係パタンを用いた単語ペアの表現は，前節で述べた単語埋め込みを用いた単語ペアの表現とは相補的な情報を捉えているという報告がある [45, 71]．

3.3 パタン欠落問題

LRA のような関係パタンを用いて埋め込みを獲得する手法は，対象の単語ペアの共起を必要とする．しかし，たとえ大規模コーパスを用いたとしても，意味関係を持つ二語が共起する

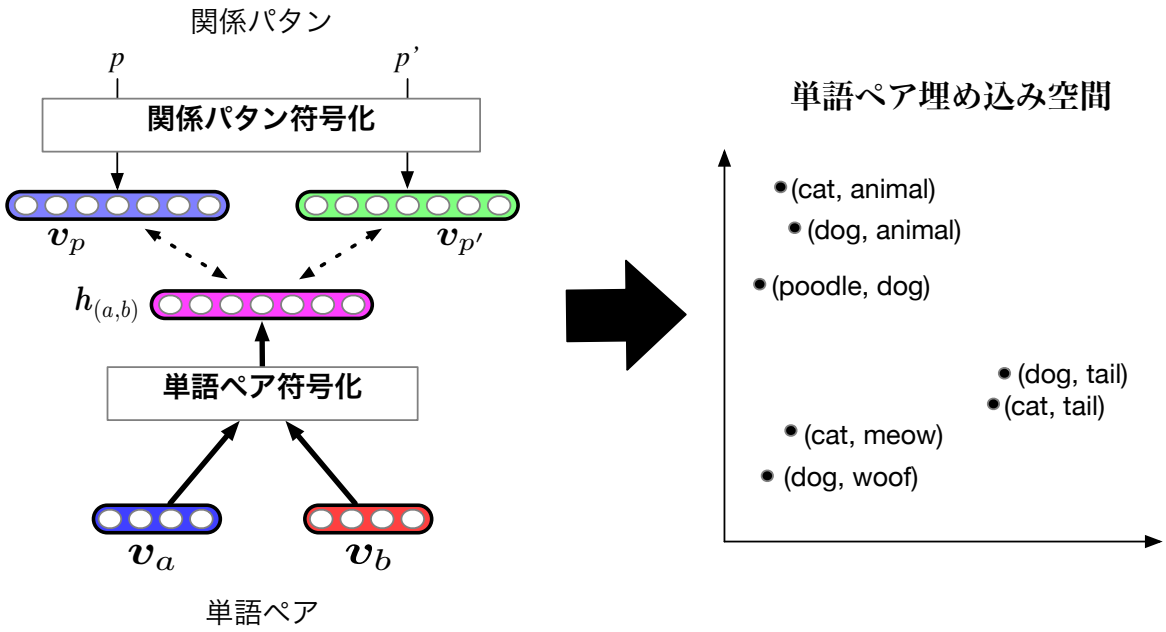

図 3.1　ニューラル潜在関係解析（NLRA）

とは限らない．なぜならば，単語の出現頻度はジフの法則 [27] に従うことが知られており，興味がある内容語の大部分は極めて低頻度だからである．このパタン欠落問題により，LRA のような単語ペアと関係パタンの共起頻度行列に基づく手法は，実際に共起を得ることができた一部の単語ペアにしか埋め込みを割り当てることができなくなってしまう．これは，できるだけ多くの単語ペアについて意味関係の情報を獲得したいという目的において，非常に大きな問題である．この問題は，教師あり意味関係識別において，2.4.2 節で述べたような関係パタンによって特徴ベクトルを構成する一連の手法 [72, 71, 70] にも同様に影響する．

3.4　提案手法

本研究では，パタン欠落問題を解決するために，単語ペアと関係パタンの共起をニューラルネットワークによって汎化して単語ペア埋め込みを獲得するニューラル潜在関係解析（Neural Latent Relational Analysis, NLRA）を提案する．ニューラルネットワークを用いて汎化することで，コーパス上で共起しなかった単語ペアについても，意味関係知識表現に重要な特徴を持つ関係パタンとの共起情報を捉えた単語ペア埋め込みが得られる．以下で図 3.1 に概要を示す NLRA の枠組みについて述べていく．

3.4.1　単語ペアと関係パタンの符号化

本手法では，単語ペアと関係パタンをそれぞれ埋め込みとして符号化し，それらの内積から共起する程度を予測することで学習を行う．単語ペア (a, b) と関係パタン p の共起トリプル (a, b, p) の集合を D とする．単語ペア (a, b) は微分可能な関数を用いて符号化を行う．

$$\boldsymbol{h}_{(a,b)} = f(\boldsymbol{v}_a, \boldsymbol{v}_b) \tag{3.2}$$

ただし，関数 f は，二つの単語埋め込み $\boldsymbol{v}_a, \boldsymbol{v}_b$ を入力とし，ベクトル $\boldsymbol{h}_{(a,b)}$ を返すパラメータを持った微分可能な関数である．関数 f は典型的には多層パーセプトロンである．多層パーセプトロンへの 2 つの単語埋め込みの入力の仕方には様々な選択肢がある．たとえば，3.5 節では，$[\boldsymbol{v}_a; \boldsymbol{v}_b; \boldsymbol{v}_b - \boldsymbol{v}_a]$ を，3.6 節では，$[\boldsymbol{v}_a; \boldsymbol{v}_b]$ を用いている．ただし，; はベクトルの結合を表す．基本的に，$[\boldsymbol{v}_a; \boldsymbol{v}_b]$ が入力に含まれていれば，入力の複雑な相互作用を学習する多層パーセプトロンの入力であるため，大きな差はないと考えられる．

関係パタン p の符号化にはいくつかの方法がある．もっとも単純な方法は，ランダムな数値で初期化した埋め込みを $\boldsymbol{v}_p$ として，各パタンに割り当てるものである．この方法では関係パタンの符号化の汎化は困難になるが，符号化は埋め込みを呼び出すだけで良いため，学習が高速になる．より洗練された方法は，RNN などのニューラルネットワークで関係パタンを符号化して $\boldsymbol{v}_p$ を得る手法である．RNN で関係パタンを符号化することで，学習時間は増加するが，関係パタンの構成性を考慮することができる．関係パタンは個々の要素の意味がまとめあげられることで全体としての意味が成り立つことが多いと思われ，このような構成性を考慮することで，各関係パタンに直接埋め込みを割り当てる方法では捉えられないような，関係パタン間の類似性を考慮することができる．RNN を用いた符号化では，関係パタン内の各要素は埋め込みとして表現され，それらが順次 RNN に入力されていき，最終的に一つの埋め込みが出力されるというように，個々の表現をまとめあげることで関係パタン全体を表現する表現が構成する．目的関数の最適化を通して，目的関数の最適化に有効な関係パタンの構成を RNN は学習する．これによって，RNN を用いた場合は関係パタン間の類似性を考慮した学習ができると考えられる．

3.4.2　目的関数

目的関数には単語埋め込みの学習に用いられる負例サンプリング目的関数 [50] を拡張したものを用いる．単語ペアを表現する $\boldsymbol{h}_{(a,b)}$ と関係パタンを表現する $\boldsymbol{v}_p$ が計算できるとき，実際に共起したトリプル (a, b, p) を共起しなかったトリプル (a, b, p') と区別するように，以下の

ような目的関数 L で学習を行う．

$$L = \sum_{(a,b,p)\in D)} \left\{ \log P\left((a,b,p) \in D\right) + \sum_{(a,b,p')\in D'_{(a,b,p)}} \log P((a,b,p) \notin D) \right\} \tag{3.3}$$

$$= \sum_{(a,b,p)\in D} \left\{ \log \sigma\left(\boldsymbol{v}_p \cdot \boldsymbol{h}_{(a,b)}\right) + \sum_{(a,b,p')\in D'_{(a,b,p)}} \log \sigma\left(-\boldsymbol{v}_{p'} \cdot \boldsymbol{h}_{(a,b)}\right) \right\} \tag{3.4}$$

ただし，σ はシグモイド関数，$\cdot$ はベクトルの内積，$D'_{(a,b,p)}$ は $(a,b,p) \in D$ に対してランダムに関係パタンをサンプリングすることで生成された負例サンプルの集合である．負例サンプル (a,b,p') は，各 $(a,b,p) \in D$ に対してランダムに k 個生成される．この負例サンプル数 k はハイパーパラメータである．この目的関数 L を確率的勾配降下法によって最大化することで学習が行われる．

学習の結果として，$\boldsymbol{h}_{(a,b)}$ には各関係パタンとどれぐらい共起しやすいかという情報が埋め込まれることになり，潜在関係仮説に基づいた単語間意味関係知識を表現することができる．

3.4.3 単語ペア埋め込みの計算

ニューラルネットワークの学習後，単語埋め込みを持つ任意の二語について，式 3.2 に基づいて，関係パタンの情報を捉えた表現 $\boldsymbol{h}_{(a,b)}$ を得ることができる．ここから以下のように単語ペア (a,b) の埋め込み $\boldsymbol{v}_{(a,b)}$ を計算する．

$$\boldsymbol{v}_{(a,b)} = \left[\boldsymbol{h}_{(a,b)}; \boldsymbol{h}_{(b,a)}\right] \tag{3.5}$$

$\boldsymbol{h}_{(a,b)}$ と $\boldsymbol{h}_{(b,a)}$ は正規化した後に結合してもよい．単語ペアの関係の方向性は $\boldsymbol{h}_{(a,b)}$ と $\boldsymbol{h}_{(b,a)}$ の結合の順序によって表現されている．また，このように単語ペアを表現することで，コサイン尺度で二つの単語ペアの類似度を計算する際に，単語ペアの二語を反対にした場合でも一貫性が保たれる．

$$\cos(\boldsymbol{v}_{(a,b)}, \boldsymbol{v}_{(a',b')}) = \cos(\boldsymbol{v}_{(b,a)}, \boldsymbol{v}_{(b',a')}) \tag{3.6}$$

ただし，cos はコサイン尺度を計算する関数である．

3.5 実験：意味関係類似性ベンチマークでの評価

本節では，代表的な意味関係類似性ベンチマークである SemEval2012 Task2 データセットでの実験について述べる．

表 3.1 単語ペア埋め込みの教師なし学習のハイパーパラメータ（意味関係類似性）

ハイパーパラメータ	値
次元数	
- 単語埋め込み	300
- 多層パーセプトロンの隠れ状態	300
負例サンプリング数 k	10
ミニバッチサイズ	100
最適化手法	AdaGrad
学習率	0.01
エポック数	50

3.5.1 SemEval2012 Task2

SemEval2012 Task2 データセット [40] は，2.5.1 節で述べたように，単語ペア間の意味関係類似性に基づいて，システムがどれほど適切に意味関係を捉えられているかを測るベンチマークである．タスクとしては，数十個の意味関係カテゴリそれぞれについて，システムが計算した典型単語ペアと候補単語ペアの類似度に基づいて候補単語ペアをランキングし，人間が注釈した候補ペアの適合度とのスピアマンの順位相関を測ることで，システムの評価を行う．

本実験では，典型単語ペアが複数ある場合は，各典型単語ペアと候補単語ペアの類似度の平均を取って類似度を計算した．なお，先行研究 [61, 89] に従い，テストセットに指定されている 69 個の意味関係カテゴリで評価を行った．

3.5.2 比較手法

提案手法の有効性を示すために，以下の手法を比較した．

VecOff[89] 式 3.1 に従い，教師なし学習された単語埋め込みの差分で単語ペアを表現するモデル．300 次元の訓練済み GloVe[58][*1]を用いた．この単語埋め込みは英語版 Wikipedia と Gigaword コーパス[*2]から獲得されており，これらのコーパスは以下で述べる LRA や NLRA で用いたコーパスを含んでいる．

LRA[76] 2.4.3 節，3.2.2 節に基づいて LRA を実装した．対象となる単語の集合 W は SemEval Task2 データセット内の単語ペアに設定した．英語版 Wikipedia をコーパスと

[*1] http://nlp.stanford.edu/data/glove.6B.zip
[*2] https://catalog.ldc.upenn.edu/LDC2011T07

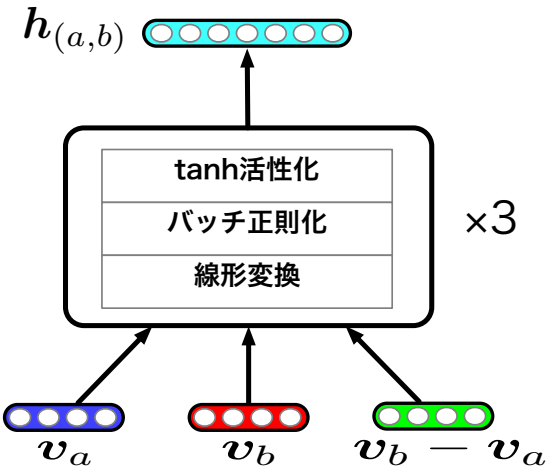

図 3.2　意味関係類似性ベンチマークの実験での関数 $f(a,b)$

して用いて，各単語ペアについて，単語ペアの間に共起した 1 語から 3 語の単語系列を関係パタンとして抽出した．関係パタンの各単語にはレンマ化を施した．たとえば，*... dogs such as animals ...* という単語系列があるとき，*dogs* と *animals* に注目すると，*(dog, animal)*, *X such as Y* という単語ペアと関係パタンの共起が抽出される．Turney の先行研究 [78] に従い，関係パタンをワイルドカードで一般化した．たとえば，*X such as Y* という関係パタンは，*X * as Y*，*X such * Y*，*X * * Y* というように，関係パタンを構成する各単語がそれぞれワイルドカード（*）に置換され，それぞれについて単語ペア *(dog, animal)* と共起したとみなす．結果として得られる関係パタンの頻度を集計し，頻度の高い上位 $20|W|$ 個のパタンを対象の関係パタンの集合 C として，共起頻度行列 M を作った．各値を PPMI で重み付けし，特異値分解によって 300 次元に次元削減を行い，W の各単語ペアに埋め込みを割り当てた．

NLRA（提案手法） 3.4 節に従い NLRA を実装した．LRA の実装時に抽出された単語ペアと関係パタン（1 語から 3 語の単語系列）の共起を D とした，NLRA においては，ワイルドカードによる関係パタンの一般化は行わない．単語埋め込みには VecOff で用いたものと同じ 300 次元 GloVe を割り当てた．式 3.2 の関数 f には図 3.2 に示すように，入力に線形変換，バッチ正則化 [35]，*tanh* 活性化関数による非線形変換を施す処理を一つの計算単位として，それらを三回施す多層パーセプトロンを用いた．多層パーセプトロンの入力は，$[\boldsymbol{v}_a; \boldsymbol{v}_b; \boldsymbol{v}_b - \boldsymbol{v}_a]$ とし，隠れ状態の次元数は 300 とした[*3]．関係パタン p の符号化には，RNN の一種である長短期記憶ネットワーク（LSTM）[33] を用いた．LSTM は関係パタン内の各単語の埋め込みを 1 ステップの入力としてとり，す

[*3] NLRA の入力に $\boldsymbol{v}_b - \boldsymbol{v}_a$ が結合されているが，本手法で類似度計算に用いられるのは式 3.5 で計算される関係パタンの情報を捉えた $\boldsymbol{v}_{(a,b)}$ であり，VecOff で $\boldsymbol{v}_b - \boldsymbol{v}_a$ が直接類似度計算に使われるのとは異なる．

表 3.2　各意味関係カテゴリ（大分類）における各手法のスピアマン順位相関係数

意味関係カテゴリ	VecOff	LRA	NLRA	NLRA+VecOff
Class-Inclusion	0.487	0.427	**0.622**	0.611
Part-Whole	0.304	0.282	0.38	**0.395**
Similar	0.267	0.123	0.271	**0.315**
Contrast	0.108	0.065	0.092	**0.124**
Attribute	0.406	0.299	0.367	**0.456**
Non-Attribute	**0.217**	0.16	0.125	0.174
Case Relations	0.391	0.291	**0.553**	0.544
Cause Purpose	0.345	0.387	0.397	**0.454**
Space-Time	0.424	0.31	0.489	**0.493**
Reference	0.297	0.346	**0.404**	0.378
平均	0.321	0.246	0.36	**0.391**

べての単語が入力された後の最後の隠れ状態を関係パタンの表現 $\boldsymbol{v}_p$ として出力する．学習は，式 3.3 を AdaGrad[16] で最大化することで行った．学習では，単語埋め込み，単語ペア符号化関数 f，関係パタン符号化用の LSTM を含むすべてのパラメータを最適化した．学習後，各単語ペアに式 3.5 に従って埋め込みを割り当てた．その他のハイパーパラメータを表 3.1 に示す．

NLRA+VecOff（提案手法） NLRA と VecOff を組み合わせた手法であり，NLRA で計算した類似度と VecOff で計算した類似度の平均を単語ペア間の類似度とするモデルである．

3.5.3　結果

結果を表 3.2 に示す．

NLRA vs. LRA

まず，結果として，NLRA は LRA を相関係数の平均で上回っていた．これは，NLRA が共起しなかったペアに対しても，良い単語ペア埋め込みを割り当てることができるためだと考えられる．例を表 3.3 に示す．この表では *Referece-Express* 関係を例に，人間が高適合・中適合・低適合とみなした単語ペアについて，LRA と NLRA のスコアを示している．2.5.1 節で述べたように，人間のスコアはクラウドワーカーに提示された候補ペアの選択タスクにおいて，もっとも適合すると判断された回数の割合ともっとも非適合と判断された回数の割合の差

表 3.3 人間，LRA，NLRA が *Reference* − *Express* 関係の候補単語ペアに割り当てたスコア

	単語ペア	人間	LRA	NLRA
高適合	laugh:happiness	50	0.217	0.578
	nod:agreement	46	0.245	0.347
	tears:sadness	44	0.381	0.483
	...		...	
中適合	scream:terror	26	0.396	0.417
	handshake:cordiality	24	**0 (関係パタンなし)**	0.34
	lie:dishonesty	16	0.206	0.394
	...		...	
低適合	discourse:relationship	-60	0.331	0.275
	friendliness:wink	-68	**0 (関係パタンなし)**	0.26

である．中適合ペア *handshake:cordiality* と低適合ペア *friendliness:wink* は，コーパス上での共起が得られず，LRA では埋め込みが得られなかった．そのため，スコアを正しく割り当てることができていない．一方，NLRA は単語ペアと関係パタンの共起を汎化しているため，共起が得られなかったペアについても，適切にスコアを割り当てられてることができており，パタン欠落問題が緩和されていることがわかる．

NLRA+VecOff vs. 他のモデル

NLRA+VecOff は他の手法を平均において上回っていた．この結果は，先行研究 [45, 71] と同様に，関係パタンで捉えられる意味関係と単語埋め込みで捉えられる意味関係が相補的であり，二つを組み合わせると様々な意味関係カテゴリにおいて頑健になることを示している．

3.5.4 議論：捉えきれない意味関係

NLRA は共起を汎化することで，意味関係類似性において全体的に LRA を上回り，さらに相補的な単語埋め込みで捉えられる意味関係知識と組み合わせることで，多くの意味関係カテゴリにおいて相関係数が向上することがわかった．

一方で，*Contrast* と *Non-Attribute* の意味関係カテゴリにおいては，NLRA + VecOff において相関係数が 0.3 を下回っており，関係パタンに基づく方法と分布仮説に基づく方法を組み合わせても，これらの意味関係を適切に捉えられているとは解釈しづらい．

Contrast 関係は対義関係のことであり，分布仮説に基づくだけでは捉えることが難しいことが知られている [46]．これは対義関係にある二語の出現文脈が似ているからである．関係パ

タンを用いた対義関係知識抽出の研究 [46] では，*from X to Y* や *either X or Y* のような関係パタンを用いており，関係パタンを用いる NLRA は有効な手法かと思われるが，NLRA の Contrast 関係における相関係数は 0.1 に満たない．これには以下のような理由が考えられる．

第一に，今回の実験で用いた関係パタンは，X と Y の間にある単語系列であり，*from X to Y* の *from* や *either X or Y* の *either* などは，関係パタンから除外されている．このことが単語ペア埋め込み空間における対義関係知識の獲得に影響を及ぼしている可能性がある．第二に，対義関係を示唆する関係パタンは兄弟関係を示唆する関係パタンと区別が難しい．実際に対義関係は兄弟関係の一部であり，*either X or Y* のような関係パタンでは対義関係だけでなく兄弟関係の単語ペアも共起すると考えられる．よって，関係パタンを用いた教師なし学習のみで対義関係を正確に識別するのは難しいかもしれない．

Non-Attribute 関係は，*bulwark:flimsy* のような，X が Y という性質を持ちにくいという意味関係である．このような関係を持つ単語ペアは，その典型例であっても文内で共起しづらいと思われ，関係パタンに基づく学習では捉えるのが難しいと思われる．

これらの意味関係を捉えるには，以下のような方法が考えられる．第一に，語彙知識ベースなどの言語資源を用いることが考えられる．5.5 節でも述べるように，語彙知識ベースとコーパスの両方から単語ペア埋め込みを学習することで効率的な学習が可能となるとともに，語彙知識ベース内にある対義関係などを反映し，汎化することができる可能性がある．第二に，学習に用いる文脈を拡張することが考えられる．現在は単語ペア埋め込みの学習に，節内に収まるような共起した短い関係パタンのみを用いている．しかし，節や文をまたいだ共起も意味関係知識の獲得によって重要である．Roth らは，*but* や *although* などの語用論における談話標識を用いた意味関係知識の獲得を行っており，同じ文内でこれらの談話標識の左右で共起したか否かを特徴に加えることが，対義関係の識別に効果的であることを報告している [64]．また，語用論的な要素を用いられるように文脈を拡張することで，*Non-Attribute* 関係のような意味関係も捉えられるようになるかもしれない．このような語用論的な特徴を十分に捉えるためには，場合によっては文をまたいだ長距離の共起を捉えられるように，文脈を拡張する必要があると思われる．

3.6　実験：意味関係識別タスクでの評価

提案した教師なし学習による単語ペア埋め込みを評価するための，意味関係識別での実験について述べる．本実験では，意味関係識別の最先端のモデル [71] である LexNET（3.6.1 節）に，提案した単語ペア埋め込みを適用した場合の貢献を検証することで，単語ペア埋め込みの有効性を示す．

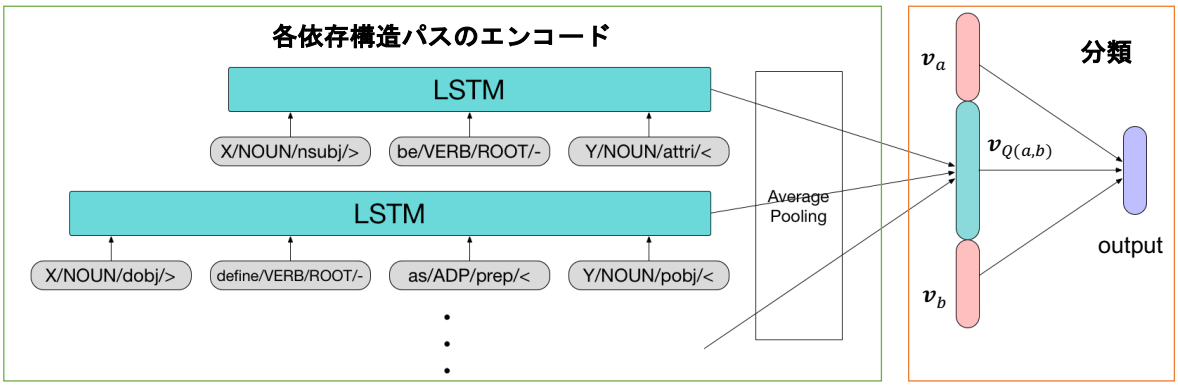

図 3.3　LexNET

3.6.1　LexNET

まず，実験で比較のために用いる教師あり意味関係識別の最先端の手法 LexNET について述べる．図 3.3 に LexNET の概要を示す．

LexNET は，RNN を用いて関係パタン（依存構造パス）を低次元で密なベクトルに符号化することで，特徴空間のスパース性を回避する手法である [71, 70]．LexNET では依存構造パスが関係パタンとして用いられる．たとえば，*A dog is a mammal.* のような文があり，対象の単語ペアを *dog* と *mammal* とするとき，依存構造パスは（`X/NOUN/nsubj/>`, `be/VERB/ROOT/-`, `Y/NOUN/attr/<`）となる．ここで，`X` は *dog*，`Y` は *mammal* である．エッジ `X/NOUN/nsubj/>`は，*dog* が名詞で，動詞 *is*（レンマ *be*）に `nsubj`（主語）の関係で依存（`>`）していることを示している．依存構造パスの各エッジは，レンマ，品詞，依存関係，依存の方向で構成される．このときエッジは，各要素の埋め込みの結合として，以下のように表現される．

$$\boldsymbol{e} = [\boldsymbol{v}_l; \boldsymbol{v}_{pos}; \boldsymbol{v}_{dep}; \boldsymbol{v}_{dir}] \tag{3.7}$$

ここで，$\boldsymbol{v}_l$，$\boldsymbol{v}_{pos}$，$\boldsymbol{v}_{dep}$，$\boldsymbol{v}_{dir}$ は，それぞれレンマ，品詞，依存関係，依存方向の埋め込みを表す．エッジ埋め込み $\boldsymbol{e}$ は，LSTM の各時点の入力となる．LSTM の時点 t の隠れ状態 $\boldsymbol{h}_t$ は以下のように計算される．

$$\boldsymbol{h}_t = LSTM\left(\boldsymbol{h}_{t-1}, \boldsymbol{e}_t\right) \tag{3.8}$$

ここで $LSTM$ は LSTM の計算式に沿って，前の隠れ状態 $\boldsymbol{h}_{t-1}$ と現在の入力 $\boldsymbol{e}_t$ から，現在の隠れ状態を計算する関数である．依存構造パスのすべてのエッジを入力した後の LSTM の最後の隠れ状態 $\boldsymbol{o}_p$ が，依存構造パス p の表現となる．単語ペアが共起した依存構造パスの表現をまとめるために，以下のような平均プーリングが適用される．

$$\boldsymbol{v}_{Q_{(a,b)}} = \frac{\sum_{p \in Q_{(a,b)}} F_{(a,b,p)} \cdot \boldsymbol{o}_p}{\sum_{p \in Q_{(a,b)}} F_{(a,b,p)}} \tag{3.9}$$

表 3.4 各データセットの訓練・開発・テストデータの事例数

	訓練	開発	テスト
K&H+N	40256	2876	14377
BLESS	10215	700	3643
ROOT09	5988	427	2187
EVALution	2298	148	794

ただし，$Q_{(a,b)}$ は単語ペア (a,b) がコーパス上で共起した依存構造パスの集合，$F_{(a,b,p)}$ は，依存構造パス p が (a,b) と共起した頻度である．

各語の単語埋め込みの情報も考慮するために，単語ペア (a,b) の関係パタン表現 $\boldsymbol{v}_{Q_{(a,b)}}$ に，それぞれの単語埋め込み $\boldsymbol{v}_a$，$\boldsymbol{v}_b$ が結合され，最終的な単語ペアの表現となる．

$$\boldsymbol{q}_{(a,b)} = \left[\boldsymbol{v}_a; \boldsymbol{v}_{Q_{(a,b)}}; \boldsymbol{v}_b\right] \tag{3.10}$$

この単語ペアの特徴表現 $\boldsymbol{q}_{(a,b)}$ をもとに，最終的な出力として各意味関係カテゴリに割り当てられる確率を表現するベクトル $\boldsymbol{y}$ が以下のように計算される．

$$\boldsymbol{y} = \mathrm{softmax}\left(\boldsymbol{W}\boldsymbol{q}_{(a,b)} + \boldsymbol{b}\right) \tag{3.11}$$

ただし，$\boldsymbol{W}$ は線形変換のパラメータ行列，$\boldsymbol{b}$ はバイアス項を表すベクトルである．softmax は以下のようなソフトマックス関数である．

$$\mathrm{softmax}(\boldsymbol{x})_i = \frac{e^{\boldsymbol{x}_i}}{\sum_k e^{\boldsymbol{x}_k}} \tag{3.12}$$

学習は交差エントロピー損失関数を最小化することで行う．

各語の単語埋め込みの情報のみでなく，LSTM によりスパース性を回避して構成性を捉えた関係パタンの表現を用いることで，LexNET は高い汎化性能を持つことが報告されている [71, 70]．

しかし，単語ペアの共起が得られなかった場合，LexNET は関係パタンに関わる有効な手がかりを得ることができないため，パタン欠落問題の影響を受ける．共起が得られなかったペアについて，LexNET は `UNK-lemma/UNK-POS/UNK-dep/UNK-dir` のようなダミーパタンでパディングを行う．しかしこの処理では，モデルは単語ペアの間に意味関係がなかったので共起しなかったのか，意味関係があったがたまたま共起しなかったのかを区別できない．

3.6.2 意味関係識別データセット

本実験では，2.5.2 節で述べた代表的なデータセットである，K&H+N[53]，BLESS[4]，EVALution[68]，ROOT09[67] の 4 つのデータセットの名詞ペアを用いて実験を行った．表

表 3.5 各データセットで扱う名詞の単語間意味関係

データセット	意味関係
K&H+N	hypernym, meronym, co-hyponym, random
BLESS	hypernym, meronym, co-hyponym, random
ROOT09	hypernym, co-hyponym, random
EVALution	hypernym, meronym, attribute, synonym, antonym, holonym, substance meronym

表 3.6 共起依存構造パスが得られた事例数の割合

データセット	事例数	共起依存構造パスを持つ事例数	割合
K&H+N	57509	8866	15.4%
BLESS	14558	8775	60.3%
ROOT09	8602	6582	76.5%
EVALution	3240	3199	98.7%

3.5 は各データセットで扱っている名詞の意味関係である．Shwartz らの先行研究 [70] に従い，EVALution からは事例数の少ない *Entails* と *MemberOf* は取り除いた．訓練・開発・テストデータの分割は，Shwartz らが用意したものを用いた．表 3.4 は各分割の事例数を示す．

3.6.3 コーパスと依存構造解析

LexNET のような関係パタンを用いた意味関係識別の手法では，単語ペアと関係パタンの共起を集める必要があるため，英語版 Wikipedia を spaCy[*4]で依存構造解析を行い，名詞ペアと依存構造パスの (a, b, p) トリプルを抽出した．このとき a，b にはレンマ化を施した．Shwartz らの実装[*5]に従い，出現頻度が 5 以下の依存構造パスを含むトリプルは扱わないことにした．

表 3.6 に，各データセットの共起依存構造パスが得られた単語ペアの事例数の割合を示す．すべてのデータセットで，必ずしもすべての名詞ペアが共起するわけではないことがわかる．

3.6.4 単語ペア埋め込みの教師なし学習と適用

単語ペア埋め込みを獲得するために，以下のような処理を行った．関係パタンには，最も頻度の高い 3 万の名詞を結びつける依存構造パスのみを教師なし学習に用いた．単語埋め込みは配布されている 50 次元の GloVe[58][*6]で初期化し，教師なし学習による更新は行わなかった．教師なし学習のために，単語 a，b が GloVe の語彙に含まれている (a, b, p) トリプルを抽

[*4] https://spacy.io
[*5] https://github.com/vered1986/LexNET
[*6] http://nlp.stanford.edu/data/glove.6B.zip

表 3.7　単語ペア埋め込みの教師なし学習のハイパーパラメータ（意味関係識別）

ハイパーパラメータ	値
次元数	
- 単語埋め込み	50
- 多層パーセプトロンの隠れ状態	100
負例サンプリング数 k	5
ミニバッチサイズ	100
最適化手法	Adam
学習率	0.001
エポック数	5

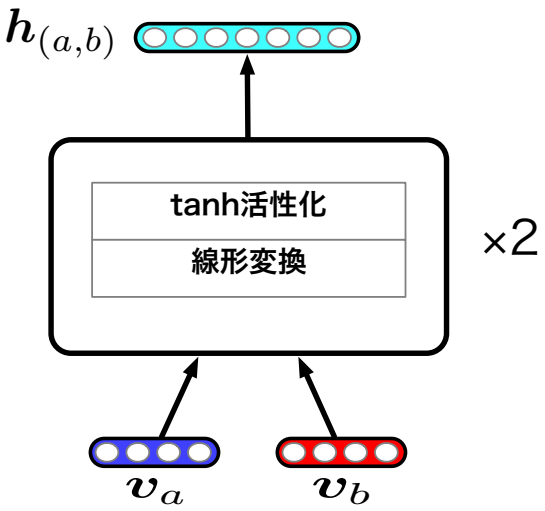

図 3.4　意味関係識別の実験での関数 $f(a,b)$

出し，教師なし学習用の訓練データ D を構築した．

式 3.2 の関数 f には，図 3.4 で示すように入力に線形変換，$tanh$ 活性化関数による非線形変換を一つの計算単位として，それらを 2 回施す多層パーセプトロンを用いた．入力は，$[\boldsymbol{v}_a; \boldsymbol{v}_b]$ とした．関係パタン p の符号化には，学習を高速化するために，ランダムに初期化した埋め込みを $\boldsymbol{v}_p$ として，各パタンに割り当てた．その他のハイパーパラメータを表 3.7 に示す．教師なし学習の後，単語ペア埋め込みは式 3.5 に従って計算する．

以上のように得られる単語ペア埋め込みを既存の意味関係識別モデル LexNET に適用するために，本研究では，図 3.5 に示すように，LexNET の単語ペアの特徴表現（式 3.10）に，式 3.5 で計算される $\boldsymbol{v}_{(a,b)}$ を結合する．

$$\boldsymbol{q}_{(a,b)} = \left[\boldsymbol{v}_a; \boldsymbol{v}_{Q_{(a,b)}}; \boldsymbol{v}_b; \boldsymbol{v}_{(a,b)}\right] \tag{3.13}$$

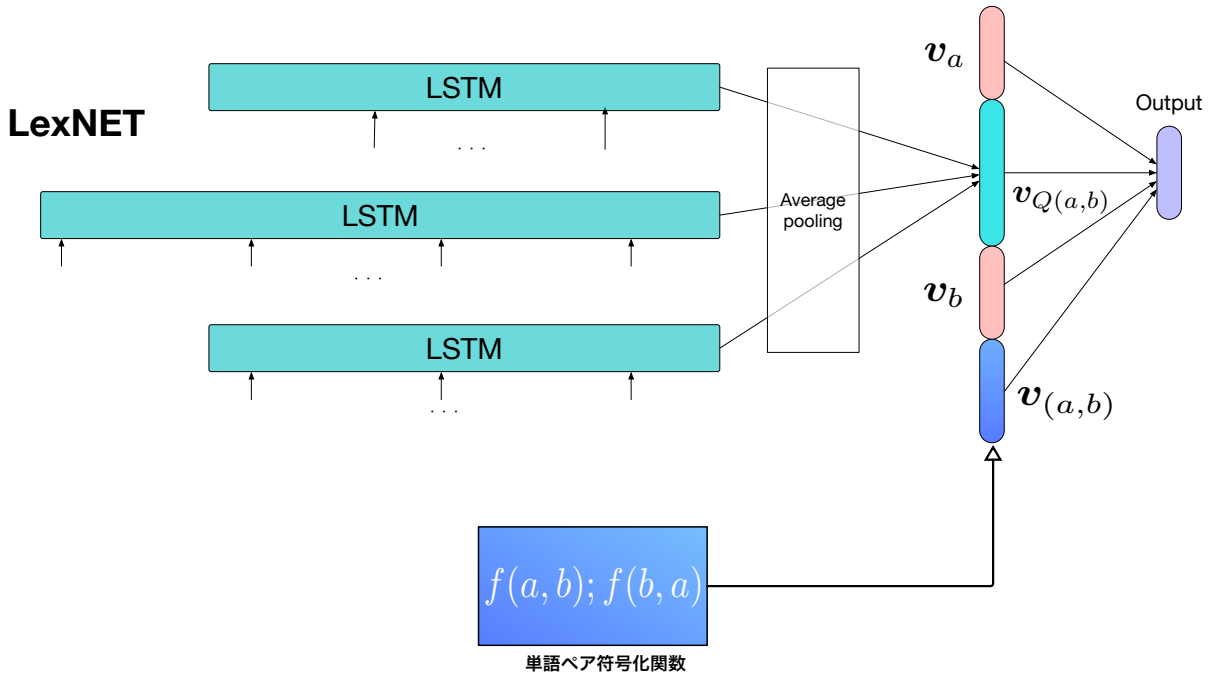

図 3.5　LexNET への単語ペア埋め込みの適用

これにより，パタン欠落問題により，関係パタン表現 $\boldsymbol{v}_{Q_{(a,b)}}$ が分類に有用な情報を有してない場合でも，共起しうる関係パタンの情報を捉えた単語ペア埋め込み $\boldsymbol{v}_{(a,b)}$ により，関係パタンの情報を用いた意味関係識別が可能となる．

3.6.5　比較手法

提案した教師なし学習の有効性を示すために，以下のモデルを比較した．なお，各モデルの訓練は，開発セットでの性能が 7 エポック向上しなかったらストップし，もっとも開発セットでの性能が高いモデルを，テストセットで評価した．評価指標は，Shwartz らの先行研究 [71] にならい，各クラスの事例数を考慮する scikit-learn[*7]の重み付き平均 F1 スコアを用いた．

LexNET [70] 3.6.1 節に従って，ベースラインとなる LexNET を実装した．ハイパーパラメータを表 3.8 に示す．レンマ埋め込み $\boldsymbol{v}_l$ は 3.6.4 節で用いた GloVe と同じもので初期化を行った．正則化は，エッジ埋め込み $\boldsymbol{e}$ の各要素をドロップアウトして行った [37, 41]．ドロップアウト率は開発セットの性能で調整した．

LexNET_h [70] LexNET の拡張で，各語の単語埋め込み $\boldsymbol{v}_a$，$\boldsymbol{v}_b$ と関係パタン表現 $\boldsymbol{v}_{Q_{(a,b)}}$ の相互作用を捉えるために，出力層との間に追加で隠れ層を追加したモデルである．隠れ層の次元数は 60 とした．その他は LexNET と同じである．

[*7] https://scikit-learn.org/stable/

表 3.8 LexNET のハイパーパラメータ

ハイパーパラメータ	値
次元数	
- $\boldsymbol{v}_l$	50
- $\boldsymbol{v}_{pos}$	4
- $\boldsymbol{v}_{dep}$	5
- $\boldsymbol{v}_{dir}$	1
- LSTM の隠れ状態	60
LSTM のレイヤー数	2
ドロップアウト率	{0.0, 0.2, 0.4}
ミニバッチサイズ	100
最適化手法	Adam
学習率	0.001

表 3.9 各データセットにおける分類性能（平均 F1 スコア）

モデル	**K&H+N**	**BLESS**	**ROOT09**	**EVALution**
LexNET	0.969	0.922	0.776	0.539
LexNET_h	0.968	0.927	0.810	0.546
LexNET_Pair	**0.970**	**0.941**	**0.846**	**0.576**

LexNET_Pair（提案手法） 3.6.4 節で述べた，教師なし学習で得られる単語ペア埋め込みを用いて LexNET を式 3.13 のように拡張したモデルである．教師なし学習の貢献を厳密に評価するために，付加された単語ペア埋め込みは，意味関係識別の訓練では更新しない．その他は LexNET と同じである．

3.6.6 結果

各データセットのテストセットでの平均 F1 スコアを表 3.9 に示す．結果として，4 つのデータセットすべてで，提案手法 LexNET_Pair が二つのベースラインを上回った．単語ペア埋め込みを用いた場合，隠れ層を追加する LexNET_h よりも性能が向上しているため，単語ペア埋め込みは，LexNET 内の各語の単語埋め込みと関係パタン表現以上の情報を符号化していると考えられる．さらに，ほとんどすべての単語ペアが関係パタンを持つ EVALution においても性能が向上していることから，LexNET のように訓練データ内の単語ペアと，それらと共起した関係パタンのみを学習に用いるのではなく，教師なし学習によってコーパス全体を

表 3.10 意味関係を持つが共起しなかった単語ペアに対する性能（平均 F1 スコア）

モデル	K&H+N	BLESS	ROOT09	EVALution
LexNET	**0.975**	0.930	0.812	1.0
LexNET_Pair	0.972	**0.940**	**0.942**	1.0

学習に用いたことが性能に貢献していると考えられる．

K&H+N での性能の向上が他のデータセットと比べてわずかであるが，これはこのデータセットにおいて単語埋め込みの情報の貢献が強く，2.4.2 節で述べたような，単語ペアの各語が各意味関係をどれほど持ちやすいか，という学習のみでほとんどの事例に正解でき，関係パタンの情報があまり性能に貢献しないのが原因だと思われる．Shwartz らの先行研究においても，このデータセットにおいては，関係パタンを用いる LexNET が単語埋め込みのみを用いた手法をごくわずかしか上回っていない [70]．

3.6.7 分析

パタン欠落問題の緩和

提案手法がパタン欠落問題をどれほど緩和できているかどうか調べるために，各データセットにおける開発セットにおいて，共起が得られなかった意味関係を持つ単語ペア（パタン欠落ペア）についての重み付き平均 F1 スコアを，ベースラインである LexNET と提案手法の LexNET_Pair とで比較した．結果を表 3.10 に示す．

表から BLESS と ROOT09 においては単語ペア埋め込みにより，共起が得られなかったパタン欠落ペアについての分類性能が適切に向上していることがわかる．EVALution においては，開発セットにパタン欠落ペアがひとつしか含まれておらず，それについては両モデルとも正解していたため，評価が困難であった．K&H+N においては，ベースラインが提案手法を上回っていた．これは 3.6.6 節で述べたように，単語埋め込みの情報のみで大半の事例に正解できるデータセットであり，関係パタンの情報を捉えた単語ペア埋め込みがあまり貢献しないデータセットであるからと考えられる．

関係パタンの共起予測

学習された単語ペア埋め込みが，どのような情報を符号化しているかを見るために，教師なし学習で得た $\boldsymbol{h}_{(a,b)}$（式 3.2）と $\boldsymbol{v}_p$ を用いて，BLESS の訓練セット内のコーパス上で共起しなかった単語ペアについて，どのような依存構造パスが共起しやすいかの予測を行った．

式 3.3 に示した負例サンプリング目的関数 [50] の性質により，内積 $\boldsymbol{v}_p \cdot \boldsymbol{h}_{(a,b)}$ は単語ペア (a, b) と依存構造パス p の共起のしやすさを表すため，共起しなかった単語ペアについて，教

表 3.11　BLESS 内の共起しなかった単語ペアについて予測された依存構造パス

単語ペア	意味関係	予測された依存構造パス
X = *jacket*, Y = *commodity*	上位下位関係	X/NOUN/nsubj/> be/VERB/ROOT/- shooter/NOUN/attr/< Y/NOUN/compound/<
		X/NOUN/nsubj/> be/VERB/ROOT/- Y/NOUN/attr/< manufacture/VERB/acl/<
		red/ADJ/amod/< X/NOUN/nsubj/> be/VERB/ROOT/- Y/NOUN/attr/<
X = *goose*, Y = *creature*	上位下位関係	X/NOUN/nsubj/> be/VERB/ROOT/- species/NOUN/attr/< of/ADP/prep/< Y/NOUN/pobj/< of/ADP/prep/>
		X/NOUN/nsubj/> be/VERB/ROOT/- specie/NOUN/attr/< of/ADP/prep/< Y/NOUN/pobj/< in/ADP/prep/>
		X/NOUN/pobj/> of/ADP/ROOT/- bird/NOUN/pobj/< Y/NOUN/conj/<
X = *owl*, Y = *rump*	部分全体関係	X/NOUN/ROOT/- represent/VERB/relcl/< Y/NOUN/nsubj/<
		X/NOUN/nsubj/> have/VERB/ROOT/- Y/NOUN/dobj/< be/VERB/relcl/>
		all/DET/det/< X/NOUN/nsubj/> have/VERB/ROOT/- Y/NOUN/dobj/<
X = *mug*, Y = *plastic*	部分全体関係	X/NOUN/pobj/> of/ADP/ROOT/- arm/NOUN/pobj/< Y/NOUN/conj/<
		the/DET/det/< X/NOUN/nsubjpass/> make/VERB/ROOT/- from/ADP/prep/< Y/NOUN/pobj/<
		X/NOUN/compound/> gun/NOUN/ROOT/- Y/NOUN/appos/<
X = *carrot*, Y = *beans*	兄弟関係	X/NOUN/compound/> leaf/NOUN/ROOT/- Y/NOUN/conj/<
		X/NOUN/compound/> specie/NOUN/ROOT/- Y/NOUN/conj/<
		X/NOUN/dobj/> use/VERB/ROOT/- in/ADP/prep/< Y/NOUN/pobj/< of/ADP/prep/>
X = *cello*, Y = *kazoo*	兄弟関係	X/NOUN/dobj/> play/VERB/ROOT/- guitar/NOUN/dobj/< Y/NOUN/conj/<
		X/NOUN/pobj/> for/ADP/ROOT/- piano/NOUN/pobj/< Y/NOUN/conj<
		X/NOUN/pobj/> on/ADP/ROOT/- drum/NOUN/pobj/< Y/NOUN/conj/<

師なし学習に用いた 3 万の依存構造パスを内積でランキングすることで，どのような依存構造パスと共起しうるかを見ることができる．

各単語ペアについて上位三件までの依存構造パスを予測した例を表 3.11 に示す．各単語ペアの意味関係を示唆すると思われる依存構造パスを赤字にしている．予測された依存構造パスについては，解釈が難しいものを含みつつも，各単語ペアの意味関係を表すと明らかに解釈されるものも予測されている．たとえば，(*jacket*, *commodity*) や (*goose*, *creature*) などの上位下位関係のペアにおいては，*X is Y manufactured* や，*X is a species of Y* のような，包含関係を表す is-a の関係パタンの依存構造パスが予測されている．また，部分全体関係を持つ (*owl*, *rump*) では *X has Y* のような所有の関係を表すパスが予測されている．同じく部分全体関係を持つ (*mug*, *plastic*) に対しては，材料の関係を表す *X made from Y* が予測されている．いずれも部分全体関係を示唆する関係パタンである．兄弟関係を表すペアについては単語

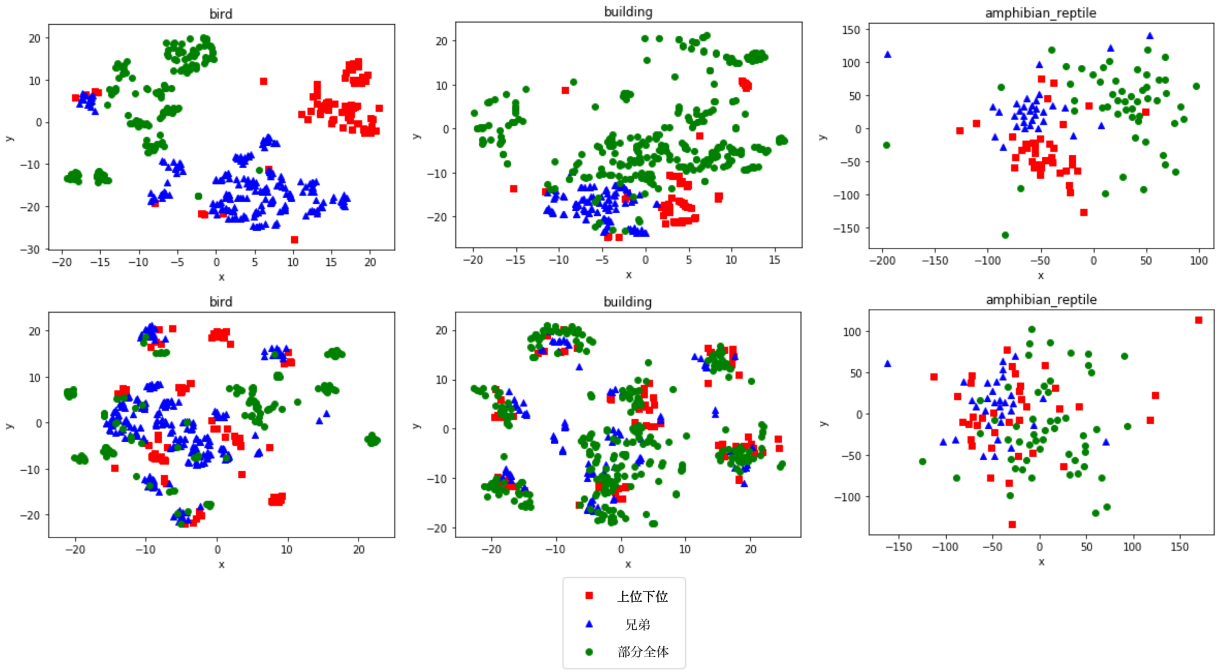

図 3.6　t-SNE による単語ペア表現の可視化（上段は $\boldsymbol{v}_{(a,b)}$，下段は $[\boldsymbol{v}_a; \boldsymbol{v}_b]$）

ペアのドメインに固有な，二つの名詞が同じカテゴリに属すことを示唆する依存構造パスが予測されている．(*carrot*, *beans*) のような野菜の単語ペアでは，*X leaf and Y*，(*cello*, *kazoo*) のような楽器の単語ペアでは，*play X, guitar, and Y* のような，等位接続の関係パタンが予測されており，兄弟関係を示唆している．

これらの例から，本研究で提案された単語ペア埋め込みが，適切に関係パタンの情報を符号化していることがわかる．これにより，単語ペア埋め込みが適切にパタン欠落問題を緩和していると考えられる．

単語ペア埋め込みの可視化

単語ペア埋め込み $\boldsymbol{v}_{(a,b)}$ の性質を調べるために，BLESS では，単語ペアに意味的分類としてドメインが注釈されているので，そのドメインごとに，t-SNE[48] によるデータ点の可視化を行った．各ドメインについて t-SNE による次元削減を行い上位下位関係，兄弟関係，部分全体関係のペアをプロットした．比較として，LexNET で用いられる二語の単語埋め込みの結合に対しても，同様に可視化を行った．結果として，いくつかのドメインに関して，単語ペア埋め込み $\boldsymbol{v}_{(a,b)}$ の空間では，各意味関係のクラスタが形成されていることがわかった．鳥（`bird`），建物（`building`），爬虫類（`amphibian_reptile`）のドメインの可視化を例として図 3.6 に示す．図においては，上位下位関係，兄弟関係，部分全体関係のデータ点がプロットされている．単語埋め込みの結合の散布図は，各意味関係のデータ点が散らばったり，混ざり

合ったりしているが，$\boldsymbol{v}_{(a,b)}$ の散布図は，各意味関係ごとにクラスタを形成しているのがわかる．これは $\boldsymbol{v}_{(a,b)}$ の空間が意味関係の類似性を捉えており，意味関係識別に有用な性質を有していることを示している．

3.7　結論

本研究では，多くの単語ペアについてより適切に意味関係を表現する単語ペア埋め込みを得るために，関係パタンとの共起を汎化する単語ペア埋め込みの教師なし学習法を提案した．単語間意味関係知識を表現する上で重要な特徴である関係パタンを用いた手法には，パタン欠落問題により，表現できる単語ペアが実際にコーパス上で共起したものに限られるという重大な問題があったが，本研究で提案したニューラルネットワークを用いた共起の汎化を伴う教師なし学習法によって，コーパス上で共起しなかった単語ペアに対しても，関係パタンの情報を考慮した単語ペア埋め込みを割り当てることができるようになった．

意味関係類似性ベンチマーク，教師あり意味関係識別での実験から，提案した教師なし学習法で学習された単語ペア埋め込みが意味関係の情報を従来法よりも良く捉えていることと，パタン欠落問題を適切に緩和できていることがわかった．

第 4 章

単語間意味関係知識の定義文処理への応用

4.1　はじめに

本章では，単語間意味関係知識の新たな応用として，定義文処理に単語ペア埋め込みを適用する．定義文処理は，定義文からの良質な単語埋め込みの獲得や，単語埋め込みから定義文の生成を行うタスクを含む，辞書内の見出し語と定義文のマッピングを機械学習モデルに学習させるタスクである．従来手法では RNN を用いて単純なモデリングを行っているが，本研究では図 4.1 に示したような，定義文内の各語と見出し語の間に存在する様々な単語間意味関係に着眼した．この図は見出し語 *knife* とその定義文 *edge tool used as a cutting instrument* の間にある単語間意味関係を示している．*knife* は，定義文内の *tool* や *instrument* と上位下位関係（Is-a 関係）を，*edge* と部分全体関係（Has-a 関係）を，*cutting* と道具-用途関係（Used-for 関係）を持つ．

本研究では，このような見出し語と定義文の間にある意味関係をモデルの学習や予測に用いることで，定義文処理の性能が向上することを示す．機械学習モデルにこのような単語間意味関係知識を組み込むためには本来，定義文に意味関係の注釈が必要であるが，前章で提案した

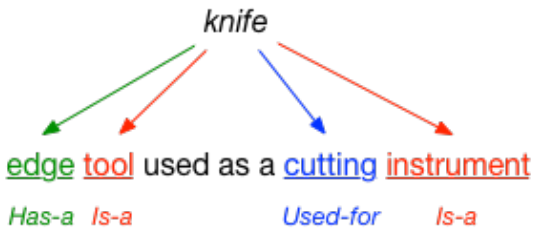

図 4.1　*knife* の定義文（WordNet より）と定義文内の語との意味関係

単語ペア埋め込みを用いることでそのコストを回避できる．

定義文からの単語埋め込み獲得と定義文生成タスクでの実験によって，単語ペア埋め込みによる意味関係知識を用いたモデルは，ベースラインを性能で上回ることがわかった．

4.2　関連研究

4.2.1　単語ペア埋め込みの応用

2.6 節では，単語間意味関係知識の様々な自然言語処理タスクへの応用について述べた．これらの多くは WordNet などの語彙知識ベース内の意味関係知識を利用しており，関係パタンに基づく単語ペア埋め込みを利用したものは少ない．これは，関係パタンが意味関係を表現する上で有益であるにも関わらず，3.3 節で述べたパタン欠落問題によって埋め込みを割り当てることができる単語ペアに制限があったためと思われる．

単語ペア埋め込みを自然言語処理タスクに適用した先行研究の一つは，Joshi らによる含意関係認識と質問応答（機械読解）タスクへの適用である [39]．Joshi らは 3 章で提案した単語ペア埋め込み獲得法の枠組みに沿って教師なし学習を行い，獲得した単語ペア埋め込みを含意関係認識と質問応答のニューラルネットワークモデルに適用した．Joshi らの研究によって，ニューラルネットワーク内の隠れ状態に単語ペア埋め込みを単純に結合するだけで，各タスクの性能が大きく向上することがわかった．さらに，単語ペア埋め込みによる性能の向上は，ニューラルネットワークに WordNet の意味関係知識を組み込む手法 [11] よりも大きかった．このように，単語間意味関係知識を用いるタスクにおいて，ニューラルネットワークモデルに単語ペア埋め込みを組み込むことで，さらなる性能向上が期待できる．

4.2.2　定義文処理

本研究で焦点を当てる定義文処理ついて述べる．定義文処理は見出し語と定義文のマッピングを行うタスクであるが，タスクとしては主に，定義文からの単語埋め込み獲得と，見出し語の単語埋め込みからの定義文生成に分けられる．両タスクとも，RNN を用いた手法が従来法の中では一番良い結果を残している．以下では各タスクの概要と RNN を用いた手法について説明する．なお，本章では定義文の単語系列を $D = (w_1, \ldots, w_T)$，見出し語を w_{trg} とする．

4.2.3　定義文からの埋め込み獲得

定義文からの単語埋め込みの獲得は，定義文をベクトルの形で符号化することで，より良い単語埋め込みを獲得することを目的とするタスクである．

定義文から単語埋め込みを獲得する動機として，単語間の類似度の意味的類似性と関連性の

観点からの分析がある [32]．たとえば，*coffee* は一般的に *cup* に注がれるので，*coffee* と *cup* は強い関連性を持つ．一方で，*coffee* は飲み物で，*cup* は容器なので，意味的な類似性は低い．*coffee* と意味的な類似性が高い単語としては，同じ飲み物である *tea* があげられる．

定義文から得られる見出し語の埋め込み（定義文埋め込み）はこのような，関連性と対比される意味的類似性を捉えるのに適しているとされている [8]．2.1 節で述べた分布仮説に基づく単語埋め込みは，出現文脈が似ている単語がベクトル空間上で近くに位置するように学習を行うため，関連性の高い単語の類似度が高くなりやすいが，定義文に基づいて単語埋め込みを学習することで，意味的な類似性が高い単語がベクトル空間上で近くなるように学習を行うことができる．

Bosc らが提案した RNN を用いた定義文埋め込みの獲得法 [8] では，LSTM を用いて以下のように定義文の符号化を行う．

$$\boldsymbol{h}_t = LSTM\left(\boldsymbol{h}_{t-1}, \boldsymbol{w}_t\right) \tag{4.1}$$

$$\boldsymbol{h}_{def} = \boldsymbol{W}_e \boldsymbol{h}_T + \boldsymbol{b}_e \tag{4.2}$$

ただし，$\boldsymbol{w}_t$ は D 内の t 番目の単語 w_t の単語埋め込み，$\boldsymbol{h}_t$ は時点 t の隠れ状態である．また，$\boldsymbol{W}_*$，$\boldsymbol{b}_*$ は，それぞれパラメータ行列とバイアス項ベクトルである．LSTM から出力された最終時点の隠れ状態 $\boldsymbol{h}_T$ を線形写像した結果得られる $\boldsymbol{h}_{def}$ が定義文埋め込みとなる．

一連のパラメータは，定義文埋め込み $\boldsymbol{h}_{def}$ からの定義文の単語バッグの復元と，$\boldsymbol{h}_{def}$ と見出し語の単語埋め込みである $\boldsymbol{w}_{trg}$ を近づけるような損失関数によって学習される．前者の定義文の単語バッグの復元は，以下のような損失関数 J_{BOW} で学習される．

$$\begin{aligned} J_{BOW} &= -\sum_t \log P\left(w_t \in D | \boldsymbol{h}_{def}\right) \\ &= -\sum_t \log \frac{exp(\boldsymbol{w'}_t \cdot \boldsymbol{h}_{def} + b_{w'_t})}{\sum_k exp(\boldsymbol{w'}_k \cdot \boldsymbol{h}_{def} + b_{w'_k})} \end{aligned} \tag{4.3}$$

ただし，$\boldsymbol{w'}_t$，$b_{w'_t}$ は，単語 w_t に割り当てられた，単語バッグ復元用のパラメータである．

後者の $\boldsymbol{h}_{def}$ を $\boldsymbol{w}_{trg}$ に近づける損失関数 J_d は以下のように定義される．

$$J_d = \|\boldsymbol{h}_{def} - \boldsymbol{w}_{trg}\|^2 \tag{4.4}$$

最終的な損失関数は以下のように二つの損失関数を組み合わせたものである．

$$J = \alpha J_{BOW} + \lambda J_d \tag{4.5}$$

ただし，α と λ は二つの損失関数のバランスを取るためのハイパーパラメータである．これらは開発セットによって決定する．

学習後，各定義文を符号化した $\boldsymbol{h}_{def}$ を見出し語の単語埋め込みとして用いることができる．Bosc らはこの手法を，Consistency Penalized AutoEncoder (CPAE) と名付けている．

対象の語が多義語の場合，一つの単語に定義文が複数割り当てられる．その場合，Bosc らはすべての定義文を結合して一つの定義文としている．Bosc らは，単一の埋め込みは多義性をうまく捉えられていると分析する先行研究 [87] をもとに，一つの単語に複数の表現を割り当てることは必ずしも必須ではないということを前提としている．このことによる帰結や影響は定かではないが，単語間意味関係知識についての研究である本論文では，関連する問題ではあるものの，語の多義性をどのように表現するかという問題については踏み込まない．

4.2.4　定義文生成

定義文生成は対象の語の単語埋め込みから定義文を生成するタスクである [56]．単語埋め込みから定義文を生成することで，辞書にないが埋め込みがある単語の意味の説明が可能になる [54, 36]．

Noraset らが提案した RNN を用いた定義文生成では，見出し語の単語埋め込みを入力として定義文の生成を行う，LSTM を用いた条件付き言語モデルを学習する [56]．この条件付き言語モデルでは，以下のように見出し語 w_{trg} を条件として定義文 D が生じる確率を計算する．

$$P\left(D|w_{trg}\right) = \prod_{t=1}^{T} P\left(w_t|w_{i<t}, w_{trg}\right) \tag{4.6}$$

見出し語 w_{trg} で言語モデルを条件付ける方法としては，以下の 4 つの手法を用いている．

- Seed(S)
- Gate(G)
- CHaracter(CH)
- Hypernym Embeddings(HE)

Seed 法は単純に，LSTM の最初の入力として，見出し語の単語埋め込み $\boldsymbol{w}_{trg}$ を追加する手法である．

$$\boldsymbol{h}_1 = LSTM\left(\boldsymbol{h}_0, \boldsymbol{w}_{trg}\right) \tag{4.7}$$

その後の $P\left(w_t|w_{i<t}, w_{trg}\right)$ の計算は以下のように行われる．

$$\boldsymbol{h}_t = LSTM\left(\boldsymbol{h}_{t-1}, \boldsymbol{w}_{t-1}\right) \tag{4.8}$$

$$P\left(w_t|w_{i<t}, w_{trg}\right) = \mathrm{softmax}\left(\boldsymbol{W}_d\boldsymbol{h}_t + \boldsymbol{b}_d\right) \tag{4.9}$$

条件付き言語モデルにおける復号化においては，式 4.8 のように，一つ前の隠れ状態 $\boldsymbol{h}_{t-1}$ と直前に予測された単語の単語埋め込み $\boldsymbol{w}_{t-1}$ から，現在の隠れ状態 $\boldsymbol{h}_t$ が計算される．なお，本タスクでは，機械翻訳などの復号化を含むタスクと同様に，末尾（$t = T + 1$）に生成の終わ

りを示す特殊トークンを追加する．Seed 法により，見出し語の情報を LSTM 内に順伝播させることができる．

しかし，最初の時点の入力に見出し語の情報を追加するだけでは，t が進むにつれて，見出し語の影響が弱くなってしまう．Gate 法は，この問題を解決するためにゲート機構 [13] を用いて，各時点で見出し語の情報を考慮しつつ LSTM の隠れ状態を更新するものである．Gate 法では，隠れ状態の更新は以下のように行われる．

$$\boldsymbol{z}_t = \sigma\left(\boldsymbol{W}_z\left[\boldsymbol{w}_{trg}; \boldsymbol{h}_t\right] + \boldsymbol{b}_z\right) \tag{4.10}$$

$$\boldsymbol{r}_t = \sigma\left(\boldsymbol{W}_r\left[\boldsymbol{w}_{trg}; \boldsymbol{h}_t\right] + \boldsymbol{b}_r\right) \tag{4.11}$$

$$\tilde{\boldsymbol{h}}_t = \tanh\left(\boldsymbol{W}_h\left[\left(\boldsymbol{r}_t \odot \boldsymbol{w}_{trg}\right); \boldsymbol{h}_t\right] + \boldsymbol{b}_h\right) \tag{4.12}$$

$$\boldsymbol{h'}_t = \left(1 - \boldsymbol{z}_t\right) \odot \boldsymbol{h}_t + \boldsymbol{z}_t \odot \tilde{\boldsymbol{h}}_t \tag{4.13}$$

ただし，σ はシグモイド関数，$\odot$ は要素積である，Gate 法は $\boldsymbol{z}_t$ と $\boldsymbol{r}_t$ によって，見出し語の情報を考慮しつつ，隠れ状態を更新する．更新された隠れ状態 $\boldsymbol{h'}_t$ が，以下のように LSTM の次の入力となると同時に，時点 t の単語の確率分布の計算に用いられる．

$$\boldsymbol{h}_t = LSTM\left(\boldsymbol{h'}_{t-1}, \boldsymbol{w}_{t-1}\right) \tag{4.14}$$

$$P\left(w_t | w_{i<t}, w_{trg}\right) = \mathrm{softmax}\left(\boldsymbol{W}_d \boldsymbol{h'}_t + \boldsymbol{b}_d\right) \tag{4.15}$$

これにより，各時点で見出し語の単語埋め込みの影響を弱めずに，定義文を復号化することができる．

CHaracter 法（CH 法）と Hypernym Embedding 法（HE 法）はそれぞれ，ゲート機構への入力時に見出し語の表現を拡張する手法である．CH 法は畳み込みニューラルネットワーク（CNN）で見出し語の文字列を符号化し，見出し語の埋め込み $\boldsymbol{w}_{trg}$ に結合する方法である．これによって，見出し語の形態論的な手がかりを捉えながら定義文を生成することができる．HE 法は上位語の単語埋め込みを見出し語の単語埋め込みに結合し，見出し語のカテゴリの情報を定義文生成で考慮する手法である．見出し語の上位語は，Hearst のパターンを用いた上位下位関係識別 [31] を Web ページ群に適用することで構築された WebIsA データベース [69] をサーチすることで抽出する．WebIsA データベースの上位語には，見出し語と上位下位関係パタンで共起した頻度が付与されている．それらを用いて頻度上位 5 語の埋め込みを重み付けし，足し合わせたものを上位語の埋め込みとして，見出し語の埋め込み $\boldsymbol{w}_{trg}$ に結合する．

各パラメータの学習は，以下の交差エントロピー損失関数を最小化することで行われる．

$$L_e \quad = -\sum_{t=1}^{T} \log P\left(w_t | w_{i<t}, w_{trg}\right) \tag{4.16}$$

4.2.5　文脈付き定義文生成

Noraset らが提案した定義文生成タスクは，見出し語のみから定義文を生成するタスクであり，語の多義性を考慮していない．語の多義性を考慮しつつ定義文を生成するために，Gadetsky らは文脈付き定義文生成タスクを導入した [22]．たとえば，*base* という単語は様々な意味を持つが，*the base of the mountain* という句においては，*base* は *the bottom or the lowest part* という定義文が表すような意味を持ち，*a place that the runner must touch before scoring* というような意味は持たない．よって生成されるべき定義文は前者の定義文である．このように，文脈付き定義文生成では，見出し語 w_{trg} と見出し語が出現した文脈の単語系列 $C = (c_1, \ldots, c_m)$ から，文脈 C に適合する定義文 D の確率値を以下のように計算するモデルを学習する．

$$P\left(D|w_{trg}\right) = \prod_{t=1}^{T} P\left(w_t|w_{i<t}, w_{trg}, C\right) \tag{4.17}$$

Gadetsky らは，文脈に応じて，見出し語の埋め込み $\boldsymbol{w}_{trg}$ の着目する部分を変更するために，以下のように LSTM の入力部分を拡張している．

$$\boldsymbol{m} = \sigma\left(\boldsymbol{W}_c \frac{\sum_{i=1}^{m} FNN\left(\boldsymbol{c}_i\right)}{m} + \boldsymbol{b}_c\right) \tag{4.18}$$

$$\boldsymbol{w'}_{trg} = \boldsymbol{w}_{trg} \odot \boldsymbol{m} \tag{4.19}$$

$$\boldsymbol{h}_t = LSTM\left(\boldsymbol{h}_{t-1}, [\boldsymbol{w}_{t-1}; \boldsymbol{w'}_{trg}]\right) \tag{4.20}$$

$$P\left(w_t|w_{i<t}, w_{trg}, C\right) = \mathrm{softmax}\left(\boldsymbol{W}_d \boldsymbol{h}_t + \boldsymbol{b}_d\right) \tag{4.21}$$

ただし，FNN は適当なフィードフォワードニューラルネットワークであるが，本研究では Gadetsky らにならい，線形変換に *tanh* 活性化関数を施したものを用いている．Gadetsky らは，文脈に応じて，$\boldsymbol{w}_{trg}$ の情報を制御する手法を注意機構とみなし，この手法を Input Attention(I-Attention) と名付けている．

文脈付き定義文生成においても，前節と同様に交差エントロピー損失関数を最小化することで学習が行われる．

4.3　提案手法

本研究では，定義文処理において，定義される単語である見出し語と定義文内の各語の意味関係を，定義文からの埋め込み獲得と定義文生成に適用する手法を提案する．定義文に注釈が存在しない潜在的な意味関係を利用するために，3 章で提案した，コーパスから教師なし学習される単語ペア埋め込みを用いる．

以下では，定義文からの埋め込み獲得と定義文生成それぞれに，どのように単語ペア埋め込みを適用するかについて述べる．

4.3.1 定義文からの埋め込み獲得への適用

定義文からの単語埋め込み獲得では，LSTM により定義文 D の符号化を行う．このとき，定義される見出し語と定義文内の各語の意味関係を考慮するためには，以下のように，各時点 t において意味関係を表現する単語ペア埋め込み $\boldsymbol{v}_{(w_{trg},w_t)}$ を入力に結合すればよい．

$$\boldsymbol{h}_t = LSTM\left(\boldsymbol{h}_{t-1}, \left[\boldsymbol{w}_t; \boldsymbol{v}_{(w_{trg},w_t)}\right]\right) \tag{4.22}$$

これは式 4.1 の拡張である．本手法では，機能語や特殊トークン等と見出し語の関係など，無意味な意味関係を取り除くために，ストップワードや特殊トークンの埋め込みが LSTM に入力される際は，$\boldsymbol{v}_{(w_{trg},w_t)}$ をゼロベクトルとする．

見出し語と定義文の各語の意味関係を表現する単語ペア埋め込みを，定義文の符号化の際に考慮することで，定義文の各語がどのような情報を表すのかをモデルが識別することができるようになる．たとえば，見出し語の上位語を表すような単語が入力された際は，それが見出し語のカテゴリを表し，見出し語が道具を表す単語の際に，見出し語と道具-用途関係を表す語が入力された際は，入力が道具が使われる目的を示していると，符号化の際に識別することができる．これによって定義文に関する言語理解が促進され，より意味的類似性を考慮した定義文埋め込みの計算が可能になると思われる．

4.3.2 定義文生成への適用

定義文を復号化する条件付き言語モデルに意味関係の情報を活かすために，言語モデルの学習で用いられる交差エントロピー損失関数に加えて，以下の損失関数 L_{rel} をあわせて用いる．

$$L_{rel} = \frac{1}{|K|} \sum_{w_t \in K} \left\| \boldsymbol{v}_{(w_{trg},w_t)} - (\boldsymbol{W}_r \boldsymbol{h}_t + \boldsymbol{b}_r) \right\|^2 \tag{4.23}$$

$$K = \{w_t | w_t \in D \wedge w_t \notin S\} \tag{4.24}$$

ただし，S はストップワード，特殊トークンの集合である．K は D の中でストップワードや特殊トークンではないものの集合である．この損失関数は，LSTM の隠れ状態 $\boldsymbol{h}_t$ を線形写像し，時点 t に出力されるべき単語と見出し語の意味関係を表現した埋め込み $\boldsymbol{v}_{(w_{trg},w_t)}$ に近づけるものである．これによって，学習時に意味関係の情報をニューラルネットワークに逆伝播することができ，モデルが意味関係の生起パタンを学習しやすくなる．たとえば，道具を表す見出し語の定義文を生成するときは，w_{trg} のカテゴリにあたる上位語を出力した後に，見出し語の用途を表す単語が出力されるべきである，というようなパタンの学習が考えられる．

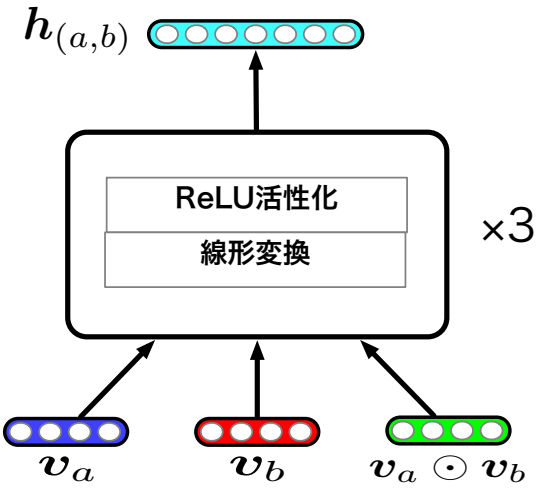

図 4.2　定義文処理の実験での関数 $f(a,b)$

4.4　実験

提案手法の評価を行うために，定義文からの埋め込み獲得と定義文生成の実験を行った．本研究で用いる単語ペア埋め込みの獲得と，それぞれのタスクの実験について以下で述べる．

4.4.1　単語ペア埋め込みの獲得

単語ペア埋め込みを獲得するために英語版 Wikipedia から単語ペア (a,b) と関係パタン p のトリプル (a,b,p) を抽出した．ここで，単語 a,b は名詞・形容詞・動詞かつ，配布されている GloVe[*1]のボキャブラリーの中で頻度が高い 10 万語であるものに制限した．関係パタン p は単語ペアを文中で結びつける依存構造パスであり，1〜3 語をパス内に含むものに限定した．コーパスの依存構造解析には spaCy を用いた．トリプルのフィルタリングとして，出現回数が 5 回未満の p を含むトリプルを捨てた．さらに，Joshi らが提案した単語ペアのサブサンプリングを用いて，高頻度なペアと低頻度なペアのバランスを取った [39]．サブサンプリングでトリプル (a,b,p) を捨てる確率 p_d は次のように計算する．

$$p_d(a,b) = p_u(a) \cdot p_u(b) \tag{4.25}$$

$$p_u(w) = 1 - \sqrt{\frac{5 \cdot 10^{-7}}{\mathrm{freq}(w)}} \tag{4.26}$$

ただし，freq は頻度を返す関数である．このサブサンプリング手法は，簡単のために，a，b の個々の頻度に基づいて，トリプルを捨てる確率を算出している．

[*1] http://nlp.stanford.edu/data/glove.6B.zip

獲得されたトリプルに対して，以下のようなニューラルネットワークの学習を行う．式 3.2 の関数 f は図 4.2 に示すように，Joshi ら [39] にならい，入力に線形変換，*ReLU* 活性化関数による非線形変換の処理を一つの計算単位として，それらを 3 回施す多層パーセプトロンを用いた．多層パーセプトロンの入力は，$[\boldsymbol{v}_a; \boldsymbol{v}_b; \boldsymbol{v}_a \odot \boldsymbol{v}_b]$ とした．ただし，$\boldsymbol{v}_a$，$\boldsymbol{v}_b$ は単語 a, b の単語埋め込みである．これらの単語埋め込みは配布されている訓練済みの GloVe[*2]で初期化した．

依存構造パス p の符号化は以下のようにして行う．抽出された p はレンマと依存方向付き依存関係ラベルで構成される系列 $p = (e_1, \ldots, e_l)$ である．e はレンマか依存方向付き依存関係ラベルを表す．たとえば，*Anarchism is a political philosophy.* という文における，*anarchism* と *philosophy* の間にある依存構造パスは $p = (\texttt{nsubj>}, \texttt{be}, \texttt{attr<})$ である．$(e_1, \ldots, e_l)$ に対応するベクトルの系列 $(\boldsymbol{e}_1, \ldots, \boldsymbol{e}_l)$ に対して，双方向 LSTM を用いて以下のように符号化を行った．

$$\boldsymbol{v}_p = \boldsymbol{W}_p [\boldsymbol{h}_f; \boldsymbol{h}_b] + \boldsymbol{b}_p \tag{4.27}$$

ただし，$\boldsymbol{h}_f$，$\boldsymbol{h}_b$ は，それぞれ系列 $(\boldsymbol{e}_1, \ldots, \boldsymbol{e}_l)$ を入力とする前向き LSTM と後向き LSTM の最終時点の隠れ状態である．レンマのベクトルは同じく GloVe の埋め込みで初期化した．

ニューラルネットワークの学習は，3 章で提案された枠組みをもとに，Joshi らが単語ペア埋め込みの学習を改良するために提案した，多項負例サンプリング目的関数 L_{mv}[39] を用いて行った．これは，式 3.3 のように関係パタンのみを負例サンプリングするのではなく，各単語 a，b についても負例サンプリングを行う目的関数である．

$$\begin{aligned} L_{mv} = \sum_{(a,b,p) \in D} \Bigg\{ & \log \sigma \left(\boldsymbol{v}_p \cdot \boldsymbol{h}_{(a,b)} \right) + \sum_{(a,b,p') \in D'_{(a,b,p)}} \log \sigma \left(-\boldsymbol{v}_{p'} \cdot \boldsymbol{h}_{(a,b)} \right) \\ + & \sum_{(a,b',p) \in D'_{(a,b,p)}} \log \sigma \left(-\boldsymbol{v}_p \cdot \boldsymbol{h}_{(a,b')} \right) + \sum_{(a',b,p) \in D'_{(a,b,p)}} \log \sigma \left(-\boldsymbol{v}_p \cdot \boldsymbol{h}_{(a',b)} \right) \Bigg\} \end{aligned} \tag{4.28}$$

ただし，$(a, b, p) \in D$ に対する負例サンプルの集合 $D'_{(a,b,p)}$ は，a, b, p それぞれに対してランダムに k 個の単語 a', b' や関係パタン p' をサンプリングすることによって生成される．その他のハイパーパラメータを表 4.1 に示す．

ニューラルネットワークの教師なし学習の後，Joshi らのヒューリスティックス [39] に従い，単語ペア埋め込み $\boldsymbol{v}_{(a,b)}$ は以下のように計算した．

$$\boldsymbol{v}_{(a,b)} = \left[\frac{\boldsymbol{h}_{(a,b)}}{\|\boldsymbol{h}_{(a,b)}\|}; \frac{\boldsymbol{h}_{(b,a)}}{\|\boldsymbol{h}_{(b,a)}\|} \right] \tag{4.29}$$

[*2] http://nlp.stanford.edu/data/glove.6B.zip

表 4.1　単語ペア埋め込みの教師なし学習のハイパーパラメータ（定義文処理）

ハイパーパラメータ	値
次元数	
- 単語埋め込み	300
- 多層パーセプトロンの隠れ状態	300
- LSTM の隠れ状態	300
a, b, p の各負例サンプリング数 k	10
ミニバッチサイズ	100
最適化手法	AdaGrad
学習率	0.01
エポック数	10

4.4.2　定義文からの単語埋め込み獲得

データセット

提案手法を評価するために, Bosc らの先行研究 [8] にならい, Word Embedding Benchmarks プロジェクト[*3]のベンチマークを評価に用いた.

このベンチマークは単語埋め込みを評価するために，以下のデータセットがまとめられている.

- SimLex999(SL999) [32]
- SimLex333(SL333) [32]
- SimVerb(SV) [24]
- MEN [9]
- RG [65]
- WS353 [20]
- SCWS [34]
- MTurk(MT) [60, 26]

いずれのデータセットにおいても，システムは与えられた二語の類似度を計算し，システムが与えた類似度と人間が与えた類似度とのスピアマン順位相関係数によって評価される．これらのベンチマークの中で SL999，SL333，SV は，4.2.3 節で述べたような，関連性と対比される意味的類似性に焦点をあてたデータセットである．一方で，他のデータセットでは関連性に焦

[*3] https://github.com/tombosc/cpae

表 4.2 CPAE のハイパーパラメータ

ハイパーパラメータ	値
単語埋め込み	Google Word2Vec（300 次元）
LSTM	
- レイヤー数	1
- 隠れ状態の次元数	300
ミニバッチサイズ	32
最適化手法	Adam
学習率	3e-4
エポック数	50
勾配クリップのしきい値	5.0
α	1
λ	{1, 2, 4, 8, 16, 32, 64}

点をあてている．

定義文を符号化するニューラルネットワークの訓練には WordNet 内の見出し語と定義文のペアを用いた．WordNet では多義語に対しては複数の定義文を割り当てているため，そのようなときは，Bosc らにならい，すべての定義文を結合して一つの定義文とした．

比較手法

提案手法の評価のために，以下の手法を比較した．なお，単語間の類似度にはいずれの手法においてもコサイン尺度を用いた．

- GloVe: Wikipedia と Gigaword コーパスで訓練された 300 次元の単語埋め込み[*4]
- Google Word2Vec: Google ニュースコーパスで訓練された 300 次元の単語埋め込み[*5]
- CPAE[8]
- CPAE w/ Pair（提案手法）

Glove と Google Word2Vec は配布されている訓練済みの単語埋め込みを用いて類似度を計算する．CPAE は 4.2.3 節で述べた Bosc らのモデルで得られる単語埋め込みを用いる．提案手法として，4.4.1 節に則って獲得した単語ペア埋め込みを用いて，CPAE に 4.3.1 節で述べた手法を適用したものを，上記のベースラインと比較する．なお，CPAE で用いる単語埋め込みは Google Word2Vec 埋め込みで初期化を行った．CPAE のその他のハイパーパラメータ

[*4] http://nlp.stanford.edu/data/glove.6B.zip
[*5] https://code.google.com/archive/p/word2vec/

表 4.3　各データセットにおけるスピアマン順位相関係数（×100）

	意味的類似性ベンチマーク			関連性ベンチマーク				
	SL999	**SL333**	**SV-test**	**RG**	**SCWS**	**MEN-test**	**MT**	**WS353**
GloVe	37.1	20.7	22.0	77.0	55.9	74.2	65.0	47.8
Google	44.2	29.7	35.8	76.1	66.0	**75.6**	67.1	63.5
CPAE	48.1	33.3	42.4	**82.7**	63.4	70.1	60.2	**66.8**
CPAE w/ Pair	**48.5**	**33.8**	**44.4**	81.2	**66.7**	74.3	**67.6**	65.4

表 4.4　人間，CPAE，CPAE w/ Pair が MEN の開発セットの単語ペアに割り当てた類似度スコア（人間のスコアに沿って降順に上位 10 件を表示）．人間のスコアの範囲は 0 から 50 の間である．

単語ペア	人間	**CPAE**	**CPAE w/ Pair**
sun-sunlight	50	0.581	0.721
automobile-car	50	0.662	0.824
morning-sunrise	49	0.52	0.585
rain-storm	49	0.537	0.574
camera-photography	49	0.560	0.527
cat-kitten	49	0.664	0.678
stair-staircase	49	0.689	0.727
dance-dancer	49	0.602	0.683
sunny-sunshine	48	0.582	0.655
beach-sand	48	0.403	0.412

を表 4.2 に示す．式 4.5 のハイパーパラメータの調整は，Bosc らにならい，α を 1 に固定し，λ を開発セットでチューニングした．開発セットには，SV と MEN の開発セットを用いて，開発セットで最も相関係数の高かったモデルを評価した．なお，Bosc らにならい，モデル選択の際には，SV の相関係数に二倍の重みをつけた．チューニングの結果，CPAE は $\lambda = 8$，CPAE w/ Pair は $\lambda = 4$ となった．

結果

各データセットにおける結果を表 4.3 に示す．

提案手法である単語ペア埋め込みを用いた CPAE w/ Pair は，類似性ベンチマークにおいてはすべてのデータセットでベースラインの相関を上回っていた．この結果は単語ペア埋め込みによってもたらされる単語間意味関係知識が，定義文の理解を促進し，定義文の符号化の結果である単語埋め込み空間における意味的類似性が向上したことを示している．CPAE が Google Word2Vec で埋め込みを初期化し，WordNet で訓練されている一方，CPAE w/ Pair

は単語ペア埋め込みにより Wikipedia から得られる情報も用いているが，同じく Wikipedia の情報を用いた GloVe が意味的類似性ベンチマークで Google Word2Vec を大きく下回っていることから，CPAE w/ Pair における性能向上は，単語ペア埋め込みから得られる意味関係知識の情報が適切に定義文の理解を促進した結果であると解釈できる．

一方で，提案手法は，関連性ベンチマーク内の 3 つのデータセットでベースラインを下回っていた．CPAE と比較すると，5 つのデータセットの内，SCWS，MEN，MT の 3 つにおいて，性能を大きく上回っている．RG と WS353 においては，CPAE を下回っている．このような性能の上下は，意味的類似性と関連性は相補的でも独立ではなく [32]，関連性ベンチマークに意味的類似性が関わるものが含まれているのが原因であると考えられる．表 4.4 に MEN の開発セットにおける，人間が与えた類似度に沿った上位 10 個の単語ペアと，CPAE，CPAE w/ Pair が与えた類似度スコアを示す．この表において，*sun-sunlight*，*automobile-car*，*cat-kitten*，*stair-staircase* などの単語ペアは，関連性が高いだけでなく，意味的類似性も高いペアだと考えられる．このように MEN には，意味的類似性が高いペアも高類似度のペアとしてデータセット内に入っている．一方で，CPAE w/ Pair が捉える範囲で意味的類似性が低いにもかかわらず，関連性が高いゆえにベンチマーク上で人間によって高い類似度が割り当てられているような単語ペアもデータセットに含まれていると思われる．たとえば，表における *camera-photography* は関連性が高く意味的類似性が低い単語ペアであるため，意味的類似性をより良く捉えられていることが理由で，CPAE w/ Pair は CPAE よりも低い類似度を割り当ててしまっていると考えられる．このような傾向は，他の関連性ベンチマークにおいても，割合は異なるかもしれないが同様に現れていると考えられる．

関連性ベンチマークにおける CPAE と CPAE w/ Pair の差は，CPAE よりも CPAE w/ Pair の方が意味的類似性を捉えていることが，関連性ベンチマークの一部の単語ペアについて影響しており，あとは各テストセット内のデータの分布によって，優劣が決まっているのだと解釈できる．

LSTM の隠れ状態の類似度

単語ペア埋め込みのモデルへの影響を確認するために，定義文を符号化する LSTM の隠れ状態の類似度を分析した．分析では，以下の *kettle* と *knife* の定義文をベースラインの CPAE と CPAE w/ Pair で符号化した．

- *kettle*: *a metal pot for stewing or boiling*
- *knife*: *edge tool used as a cutting instrument*

これら二つの定義は，二つとも道具を表す単語の定義であり，上位語（*pot*, *tool*）でカテゴリを示した後に，道具-用途関係（*stewing*, *boiling*, *cutting*）の語を配置する似たスタイルを持つ．

図 4.3 は，*kettle* の定義文を符号化した LSTM の最後の隠れ状態と，*knife* の定義文を符号

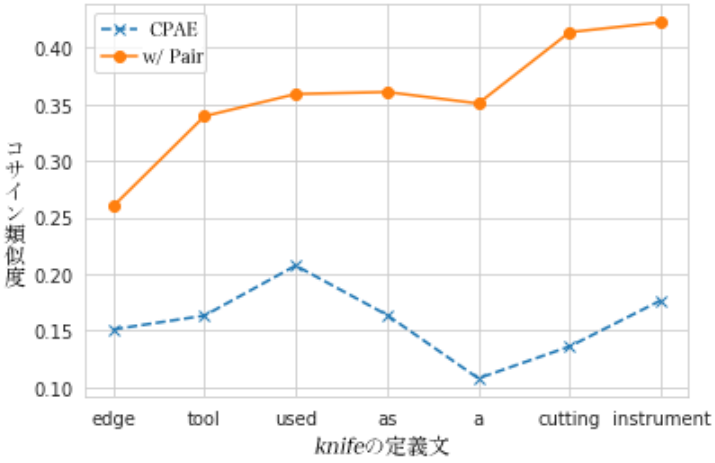

図 4.3　*kettle* の定義文を符号化した LSTM の最終隠れ状態と，*knife* の定義文を符号化した LSTM の各隠れ状態のコサイン尺度

化した LSTM の各隠れ状態のコサイン尺度による類似度を示している．この図からは，*knife* の上位語である *tool* と，*knife* の用途を表す *cutting* が LSTM に入力されたときの類似度の上昇が，ベースラインの CPAE よりも単語間意味関係知識を用いた提案手法において大きいことがわかる．これは提案手法のモデルが意味関係を考慮した学習を行っていることを示している．このような学習の結果によって，定義文の理解が促進されており，定義文埋め込みにおける意味的類似性がより良く捉えられていると考えられる．

4.4.3　定義文生成

データセット

定義文生成において提案手法を評価するために，Noraset らの文脈なし定義文生成データセット [56] と，Gadetsky らの文脈付き定義文生成データセット [22] を用いた．Noraset らのデータセットは，WordNet と GCIDE[*6]の定義文から作られている．このデータセットは一つの見出し語に対して，正解となる定義が複数割り当てられているデータセットである．一方で，Gadetsky らのデータセットは電子版の Oxford Dictionary から作成されており，各事例は見出し語と文脈となる例文，そして文脈に沿った一つの定義文から構成されている．

いずれの設定でも，定義文生成モデルの評価は，定義文を生成した際のパープレキシティ (PPL) の低さと機械翻訳の自動評価指標である BLEU スコア [57] の高さによって行われる．

比較手法

文脈なし定義文生成においては以下の手法を比較した．

[*6] http://gcide.gnu.org.ua/

表 4.5 文脈なし定義文生成のハイパーパラメータ

ハイパーパラメータ	値
単語埋め込み	{Google Word2Vec, GloVe}（300 次元）
Character Embedding の次元数	20
Character CNN の次元数	20
LSTM	
- 隠れ状態の次元数	300
- レイヤー数	2
ミニバッチサイズ	64
ドロップアウト率	0.5
最適化手法	Adam
学習率	1.2e-6
勾配クリップのしきい値	5.0

- S+G+CH+HE[56]
- S+G+CH+HE w/ L_{rel}（提案手法）

S+G+CH+HE は 4.2.4 節で述べた手法を組み合わせたものであり，Noraset らの先行研究では最も良い性能を出しているモデルである．これをベースラインとして，4.3.2 節で述べた L_{rel} を用いたものと比較して，提案手法の評価を行う．各モデルの訓練は，エポック毎に計算される開発セットでの PPL が 5 エポック向上しなかったときストップし，開発セットでの PPL がもっとも低かったモデルをテストデータで評価する．S+G+CH+HE のハイパーパラメータについて，表 4.5 に示す．

文脈付き定義文生成においては，以下の二つの手法を比較する．

- S+I-Attention[22]
- S+I-Attention w/ L_{rel}（提案手法）

S+I-Attention は 4.2.4 節で述べた Seed と 4.2.5 節で述べた I-Attention を組み合わせたものである．これに L_{rel} を用いたもの提案手法として比較する．モデル選択は文脈なし定義文生成と同様にして行う．S+I-Attention のハイパーパラメータについて，表 4.6 に示す．

結果

各データセット・設定での結果を表 4.7 に示す．表内の Google と GloVe は，それぞれ単語埋め込みに Google Word2Vec と GloVe を用いた場合を表す．各設定において，提案手法を

表 4.6　文脈付き定義文生成のハイパーパラメータ

ハイパーパラメータ	値
単語埋め込み	{Google Word2Vec, GloVe}（300 次元）
LSTM	
- 隠れ状態の次元数	300
- レイヤー数	2
FNN の次元数	300
ミニバッチサイズ	64
ドロップアウト率	0.5
最適化手法	Adam
学習率	1.2e-6
勾配クリップのしきい値	5.0

表 4.7　文脈なし・文脈付き定義文生成における PPL と BLEU スコア

文脈なし定義文生成 [56]		
モデル	**PPL**	**BLEU**
S+G+CH+HE (GloVe)	50.7	31.6
w/ L_{rel}	**44.9**	**34.9**
S+G+CH+HE (Google)	46.8	35.4
w/ L_{rel}	**39.5**	**37.9**
文脈付き定義文生成 [22]		
モデル	**PPL**	**BLEU**
S+I-Attention(GloVe)	56.8	11.8
w/ L_{rel}	**54.7**	**12.0**
S+I-Attention(Google)	59.6	12.0
w/ L_{rel}	**43.8**	**12.3**

用いると PPL と BLEU スコアが向上している．Wikipedia と Gigaword コーパスで訓練された GloVe で単語埋め込みを初期化した場合でも，Wikipedia で訓練された単語ペア埋め込みを適用すると PPL と BLEU が向上していることから，単語ペア埋め込みによってもたらされる単語間意味関係が定義文生成において有効であることがわかる．文脈なし定義文生成と文脈付き定義文生成でスコアで BLEU スコアが大きく異なるが，これは文脈なし定義文生成では参照定義文が複数あるにも関わらず，文脈付き定義文生成では参照文が一つしかないことによると思われる．

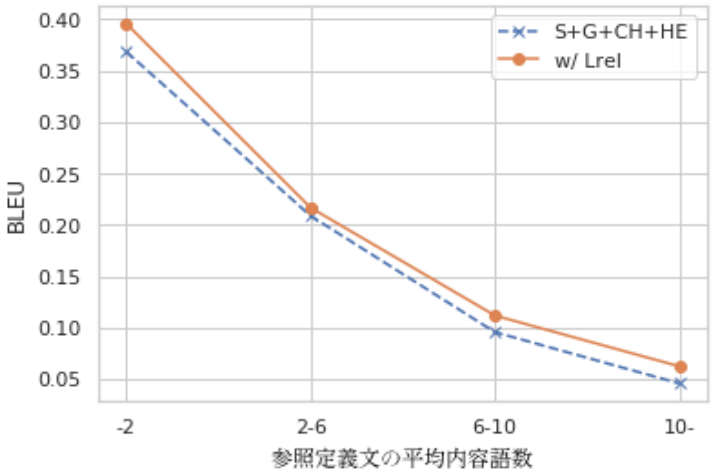

図 4.4 S+G+CH+HE(Google) と S+G+CH+HE(Google) w/ L_{rel} の BLEU スコアと参照定義文の平均内容語数の関係

表 4.8 S+G+CH+HE(Google) と S+G+CH+HE(Google) w/ L_{rel} の生成例

	S+G+CH+HE(Google)	w/ L_{rel}
academician	a person who specializes in a particular profession	one who is versed in a scholarly or scientific field
artist	one who is a person who is a person or thing is made	one who creates a picture or representation of a creative work
adolescence	the state of being pregnant	the state of being mature

定義文生成への効果

単語ペア埋め込みによる意味関係の効果を調べるために，文脈なし定義文生成の開発セットにおける BLEU スコアと参照定義文の内容語数の関係を調べた．図 4.4 にそれを示す．この図を見ると，参照定義文の内容語数が極端に少ない場合（-2）と内容語数が多い場合（6-10, 10-）は，L_{rel} を適用したほうが，適切に定義文を生成できていることがわかる．前者の性能向上により，見出し語のカテゴリとなる語の生成が向上していることがわかる．また，後者の性能向上は，見出し語の定義により詳細な説明が必要なとき，提案手法を用いるとより正確な単語を選んで生成できていることを示している．

生成例

表 4.8 に Noraset らのデータセットの開発セットにおける，ベースライン（S+G+CH+HE）と提案手法（S+G+CH+HE w/ L_{rel}）の生成例を示す．

academician と *artist* について，ベースラインは定義文を正しく生成できていないが，提案手法では適切に生成できていることがわかる．ベースラインでは，見出し語のカテゴリ（*person*, *one*）は正しく出力できているが，その後の説明で間違えている．一方で提案手法は，カテゴリの後に来る見出し語の詳細な説明の部分でも正しく単語を選べている．

adolescence については，両方のモデルとも誤った定義文を生成してしまっている．単語ペア埋め込みによって意味関係知識を考慮しているにも関わらず，提案手法は *adolescense* と反対の意味の定義文を生成している．このような反対の意味の定義文を生成してしまう問題は，単語埋め込みからの定義文生成では困難な問題であることが Norset らにより報告されている [56]．提案手法は，詳細な説明の正しい生成を助けるが，生成例を見ると対義関係については不十分であり，この問題の解決のためには，入力となる単語埋め込み空間か，学習時に用いられる単語ペア埋め込み空間のどちらか，あるいは両方において，対義関係をより適切に捉えて生成を行う必要があると考えられる．単語ペア埋め込み空間において対義関係を適切に捉える場合は，単語間意味関係知識の学習時に，対義関係知識を持つ WordNet などの語彙知識ベースを同時に用いるなどの方法が考えられる（5.5 節）．

4.5　結論

本研究では，単語間意味関係知識の新たな応用として，3 章で提案した単語ペア埋め込みを定義文処理に適用した．単語ペア埋め込みを用いて，見出し語と定義文の間にある潜在的な意味関係を考慮することで，定義文からの単語埋め込みの獲得（定義文の符号化）と定義文生成（定義文の復号化）において性能が向上することを示した．定義文処理に対する今後の方向性としては，単語だけでなく句に対する定義の生成 [36] にも，単語ペア埋め込みを適用していくことが考えられる．さらに，2.6 節で述べたような，単語間意味関係知識が有用であるとされるタスクに対して，単語ペア埋め込みを適用していきたい．

第 5 章

考察と今後の方向性

本博士論文の研究では，関係パタンとの共起を汎化する単語ペア埋め込みの教師なし学習の提案と，単語ペア埋め込みの定義文処理への適用を行った．

本章では，本博士論文の各研究に関して考察するとともに，一連の研究で明らかになっていないことや，今後の研究の方向性について述べる．

5.1　意味関係知識の獲得に関する考察

本博士論文における意味関係知識の獲得の研究の目的は，意味関係知識表現としての適切さと表現可能な単語ペアの範囲を両立させる単語ペア埋め込みである．3 章で提案した手法により，単語埋め込みを持つ任意の二語について，従来より意味関係知識の重要な特徴である関係パタンの情報を考慮した単語ペア埋め込みを割り当てることができるようになった．

先行研究では単語ペアを表現する際に，それぞれの単語埋め込みの引き算により単語ペアを表現する手法（VecOff），単語ペアと関係パタンの共起頻度行列を特異値分解して単語ペアの表現を獲得する手法（LRA）が代表的な手法であった．前者の手法は単語埋め込みが割り当てられた単語同士であれば，任意の単語ペアに埋め込みを割り当てることができるが，従来より意味関係知識を示唆すると考えられてきた，単語ペアが共起した関係パタンの情報を活用することができない．単語埋め込みの学習は，単語と文脈の共起に基づいており，単語ペアと文脈の共起は学習しないためである．一方で，後者の手法である LRA は，関係パタンの情報を単語ペアの表現に反映できるものの，実際にコーパス上で共起した単語ペアについてのみしか表現を獲得できない．

本研究で提案した NLRA は，LRA の問題をニューラルネットワークの汎化能力によって解決し，VecOff と同じく，それぞれが単語埋め込みを持つ任意の二語について，関係パタンの情報を反映した単語ペア埋め込みを割り当てることができる．この優位性に関しては，3.5 節の表 3.3 において，共起しなかった単語ペアについても意味関係カテゴリへの適合スコアを

適切に計算できていることや，3.6 節の表 3.10 で，共起しなかった単語ペアに関する分類性能が向上することなどで確認した．また，NLRA で得られる埋め込みは VecOff とは相補的な情報を捉えており，合わせて用いるとより意味関係知識が捉えられることも 3.5 節の実験で確認した．これにより，単語間意味関係知識が関わるタスクを解くための後続のニューラルネットワークの入力として，単語埋め込みだけでなく単語ペア埋め込みを追加することで，性能の向上が期待できる．

一方で，3.5 節の実験結果から，捉えれない意味関係知識として，関係パタンのみでは兄弟関係と判別しづらい対義関係や，二語がそもそも共起しづらい「X が Y という性質を持ちにくい」という関係があることがわかった．後者のような共起しづらい関係は，コーパスからは学習しづらいかもしれない．一つの解決方法は，文脈の拡張である．本研究では文内で単語ペアと共起した関係パタンのみに基づいて埋め込みを獲得したが，Roth ら [64] が談話標識を用いて対義関係知識を獲得したように，言語学的直観に基づいて有効な文脈を設計することで，捉えられる意味関係知識の幅が広がるかもしれない．もう一つの方法として，5.5 節で述べるような，対義関係など，コーパスから獲得しづらい意味関係知識を含む語彙知識ベースを，コーパスと同時に学習に用いる手法も考えられる．

5.2　意味関係知識の応用に関する考察

本博士論文における意味関係知識の応用に関する研究の目的は，高度な意味処理において取り扱いが自明ではない単語間意味関係知識の応用の幅を広げることである．4 章では，実際に，単語ペア埋め込みによる意味関係知識の新たな応用として，定義文処理に単語ペア埋め込みを適用し，その有効性を示した．定義文処理では見出し語と定義文の間に存在する単語間意味関係知識は注目されていなかったが，定義文には見出し語と上位下位関係を持つ語や部分全体関係を持つ語などが含まれており [1]，モデリングにこれらの単語間意味関係知識を組み込むことで，定義文処理の性能が向上することを示した．本研究によって，単語間意味関係知識を活かせるタスクが一つ広がったといえる．

4 章では，単語間意味関係知識の表現として単語ペア埋め込みを用いたが，2.6 節で述べたように，語彙知識ベースなどから得られる知識を用いることも可能である．本節では，ニューラルネットワークに対する意味関係知識の挿入形式による影響について考察する．

2.6 節で，ニューラルネットワークに対する単語間意味関係知識の挿入する際の単語ペアの表現は，大きく分けて以下の三つの形式があることを述べた．

1. 語彙知識ベース内の関係の離散的表現（知識ベース記号表現）
2. 語彙知識ベースから得られる単語ペアの表現（知識ベース埋め込み表現）
3. コーパスから得られる単語ペアの表現（コーパス埋め込み表現）

3 章で提案した NLRA で得られる単語ペア埋め込みは，コーパスから学習されるので，コーパス埋め込み表現とみなせる．

ニューラルネットワークへの入力形式は後続のタスクにどのように影響するのだろうか．知識ベース記号表現と知識ベース埋め込み表現を比較すると，表現できる単語ペアの数に違いがある．知識ベース記号表現では実際に語彙知識ベース内で意味関係を持つとされる単語ペアに対してのみ表現が得られ，語彙知識ベース内に収録されていない単語ペアに対しては，すべて意味関係を持たない単語であるとみなされてしまうが，知識ベース埋め込み表現では，知識グラフ埋め込みの手法によって汎化を行うことにより，語彙知識ベース内で関係を持たないとされている単語ペアに対しても，汎化の結果，語彙知識ベース内の意味関係を持ちそうな単語ペアであれば，その意味関係を反映した埋め込み表現を得ることができる．

これらの表現を比較すると，コーパス埋め込み表現には，いくつかの利点があるように思われる．第一に，コーパス埋め込み表現は，語彙知識ベース内にはないがコーパスに出現する未知語や新語などについて意味関係知識を獲得できる．知識ベース記号表現や知識ベース埋め込み表現は，語彙知識ベースに存在しない単語については表現が得られない．第二に，コーパス埋め込み表現は，語彙知識ベース内の意味関係カテゴリでは捉えきれない意味関係を捉えられる可能性がある．たとえば，WordNet では *coffee bean-coffee* と *caffeine-coffee* は両方とも部分全体関係とされている．これは誤りではないが，前者を「材料-モノ」関係，後者を「成分-モノ」関係として区別することもできる．語彙知識ベースに基づいた表現の場合，意味関係カテゴリで区別されていない関係は同じものとみなされるが，潜在関係仮説に基づくコーパス埋め込み表現ならば，*X made from Y* のような関係パタンと *X contain Y* のような関係パタンとの共起のしやすさなどから，これらの区別について学習できるかもしれない．

代表的な言語理解タスクである含意関係認識では，単語間意味関係知識の挿入として，これらの表現がニューラルネットワークの入力として用いられている．Chen らは含意関係認識のニューラルネットワークモデルに単語間意味関係知識を挿入するために，WordNet から得られる知識ベース記号表現を入力している [11]．Chen らは知識ベース埋め込み表現を入力とする手法も試しているが，含意関係認識タスクにおいては知識ベース記号表現との差は確認できなかったことを報告している．Joshi らは，コーパス埋め込み表現を含意関係認識のニューラルネットワークモデルに入力している [39]．本研究で提案した NLRA の枠組みで得られる単語ペア埋め込みを，語彙知識ベースから得られる表現の代わりに用いることで，Chen らの手法と比較して性能が大きく向上したことを Joshi らは報告している．性能向上の明確な要因は定かではないが，仮説としては先に挙げたような，未知語・新語などに関する知識や，語彙知識ベース内の意味関係カテゴリでは捉えきれないような意味関係知識がコーパスから学習され，それらが挿入されることで，含意関係認識が向上していると考えることができる．

一方で，前節でも述べたように，現状のコーパス埋め込み表現の手法では捉えきれない意味関係が存在することも確かであり，そのような意味関係に関しては，5.5 節で述べるように，

語彙知識ベースを参照することで改善される可能性がある．

5.3　単語ペア埋め込みの獲得法の優位性とハイパーパラメータ

本博士論文では 3 章で関係パタンとの共起を汎化する単語ペア埋め込みの獲得法（NLRA）を提案した．コーパス上で実際に共起した単語ペアの表現しか得られない LRA と比較を行ったが，NLRA はそれぞれが単語埋め込みを持つ任意の単語ペアに対して，関係パタンの情報を反映した単語ペア埋め込みが得られるという明確な優位性を持つ．しかし，評価としては 3.5 節において，ある一つの設定（ハイパーパラメータ）で相関係数を上回ったことを確認したのみにとどまる．NLRA は先に述べたように明確な優位性を持つものの，LRA・NLRA がもつ様々なハイパーパラメータがどのように性能に影響するかは明らかではなく，本来ならば網羅的な比較が必要である．このような網羅的な検証は，どのような設定が一般的に良いのかという知見を発見することにもつながるため，大きな価値を持つ．

NLRA のハイパーパラメータは大きく以下のような要素に分けられる．

- 関係パタンの選択（単語系列，依存構造パスなど）
- 単語ペアの符号化関数 f の選択
- 埋め込みの次元数
- 関係パタンの符号化法の選択（埋め込みを直接割り当てる，RNN で符号化など）
- 目的関数の選択（負例サンプリング目的関数，多項負例サンプリング目的関数など）
- 負例サンプルの数
- サブサンプリングの有無
- 最適化手法の選択

これらの各要素がより細かいハイパーパラメータを持つ．たとえば，f に多層パーセプトロンを用いた場合は隠れ層の数や次元数がハイパーパラメータとなる．LRA も同様に，関係パタンの選択や特徴選択の方法，次元数などがハイパーパラメータである．検証のためには，これらの膨大な組み合わせを探索しなければならない．

単語埋め込みに関しては，Levy らが古典的な単語ベクトルと単語埋め込みについて，ハイパーパラメータの様々な組み合わせを試し，網羅的に単語類似ベンチマークなどにおける性能を調査している [44]．Levy らは合計 672 個の組み合わせを試し，古典的な単語ベクトルと単語埋め込みの間で，明確な優劣は確認できなかったことを報告している．一方で，負例サンプリングを用いた Skipgram モデルが様々なベンチマークに対して頑健であり，経験則として Skipgram モデルが良い選択肢であることも述べている．NLRA は LRA に対し，表現が獲得できる単語ペアの数において優位性を持つが，単語ペア埋め込みにおいても，正確な比較検証とともに良い学習の設定を見つけるために，このような調査研究が必要と思われる．

5.4　単語ペア埋め込みで捉えられている意味関係の分析

本博士論文では，関係パタンに基いた単語ペア埋め込みの獲得法を提案したが，どのような意味関係がどこまで埋め込みとして表現されているのかは明らかになっていない．

単語埋め込みがどのような情報を捉えているのかに関してはいくつかの分析がある．Levy らは，単語埋め込みを獲得する際に，何を文脈に用いるかによって，捉えられる類似性が変化すると分析している [42]．単純に周辺に出現した語を文脈にした場合は，似たトピックの語がベクトル空間内で近くに位置するように学習されるが，文の依存構造木の中で依存関係にある語を文脈とすると，機能的な類似性が捉えられると報告している．また，4.2.3 節で述べたように，単語埋め込み空間内の距離についての意味的類似性と関連性といった分析軸 [32] も提供されている．

単語埋め込みの分析が行われている一方で，単語ペア埋め込みで捉えられる意味関係についての分析は少ない．Vylomova らは，単語埋め込みの差分による単語ペアの表現が，多様な意味関係を表現できていると分析している [82]．しかし，先行研究 [45, 71] や 3.5 節，3.6 節の実験結果でわかるように，関係パタンが捉える意味関係知識と単語埋め込みが捉える意味関係知識は相補的であると考えられる．それぞれがどのような意味関係を捉えているかはまだ明らかではなく，さらなる分析が必要である．それぞれの手法で捉えれる意味関係の範囲を明らかにすることで，捉えきれない意味関係については語彙知識ベースを参照して集中的に補完するというシステムの構築が可能になる．

5.5　語彙知識ベースとコーパスから獲得される意味関係知識の統合

本博士論文で提案した手法を含む意味関係知識を表現するベクトルの獲得は，コーパスからの情報のみに基づいている．コーパスから意味関係知識を獲得することで，語彙知識ベース上にない語の意味関係を得ることができ，さらに，含意関係認識や質問応答など，ニューラルネットワークを用いて複雑な特徴間の相互作用を学習するような手法でなければ高性能を出すのが難しいタスクにおいても，低次元で密な埋め込み表現として意味関係知識を表現することで，意味関係知識をニューラルネットワークに組み込むことができる [39]．一方で，語彙知識ベース内に蓄えられている意味関係知識も単語ペア埋め込みを獲得する上で重要な知識源となるはずである．たとえば，3.5.4 節では，単語埋め込みと単語ペア埋め込みでは，対義関係などは捉えることが難しいことに触れた．このような問題を解決するために，コーパスのみではなく，語彙知識ベース内の大量の意味関係知識を扱うことで，さらに適切に意味関係知識を捉えた埋め込みが期待できる．

単語埋め込みにおいては，語彙知識ベース等の外部知識を用いて，埋め込みの質を向上させる研究が盛んであり [88, 86, 18]，意味関係を表現する単語ペア埋め込みにおいても，語彙知識ベースを参照しつつコーパスから知識を獲得するような手法が考えられる．

5.6　ニューラル言語モデル内の意味関係知識

近年，大規模コーパスを用いて巨大なモデルサイズのニューラル言語モデルを訓練し，そこで得られる表現を後続のタスクに活かすことで，様々なタスクにおいて性能が大幅に向上することが明らかになった [59, 15]．性能の向上は，言語モデルが文脈を考慮した単語の表現を提供すると同時に，モデルが様々な言語能力を，言語モデルの目的関数から学習できていることを示している．言語モデルが統語的な言語知識等を教師なし学習によって捉えているかについては，現在，盛んに研究がなされている [38, 74]．

このような文脈において，言語モデルがどのような意味関係知識を学習しているかは，5.4 節の問題と同様に興味深い問題である．言語モデルが意味関係知識を適切に学習しているのであれば，言語モデルからそれらの知識を抽出することで，語彙知識ベースの拡張も可能となる．実際に，言語モデル内から，訓練データを用いずに，知識ベース内に存在するような百科事典的な知識や，常識的知識を抽出する研究もなされている [17, 19]．

さらに，5.5 節と同じように，WordNet などの語彙知識ベースを，ニューラル言語モデルの学習に組み込み，意味関係知識の学習を促進させる方向性の研究も興味深い．

第 6 章

結論

本博士論文では，高度な意味処理を含む自然言語処理において重要な単語間意味関係知識の獲得と応用の研究について述べた．

意味関係知識の獲得の研究においては，コーパスから得られる単語ペアと関係パタンの共起を汎化する単語ペア埋め込みの学習法を提案した．ニューラルネットワークを用いて共起の汎化を行うことで，従来法において大きな問題であった，パタン欠落問題の緩和を行い，単語埋め込みを持つ任意の二語について，関係パタンの情報を捉えた単語ペア埋め込みの獲得を可能にした．意味関係類似性ベンチマークと意味関係識別のデータセットにおける評価によって，提案手法で得られる単語ペア埋め込みが，従来法よりも様々な意味関係を良く捉えていること，パタン欠落問題を適切に緩和できていることを示した．この研究により，多くの単語ペアにおいて，単語埋め込みとは相補的な関係パタンから得られる意味関係知識を用いることができるようになり，後続の多数に用いられるニューラルネットワークで扱いやすい埋め込み形式の，適切な単語間意味関係知識の表現へと近づいた．

意味関係知識の応用の研究においては，単語間意味関知識の定義文処理への新たな応用を提案した．単語ペア埋め込みを用いることで，定義文内に潜在的に存在する見出し語との意味関係を捉えることができ，見出し語と定義文のマッピングが向上することを示した．

これらの研究によって，コーパスからの意味関係知識の獲得と，それを自然言語処理タスクに応用することの重要性を示した．

最後に，それぞれの研究に関する考察と，今後の方向性として四つの研究の方向性について議論した．今後の方向性としては，第一に，提案した単語ペア埋め込みの学習における網羅的な比較検証と，一般的に良いとされるハイパーパラメータの探索の必要性について述べた．第二に，単語ペア埋め込みが有用であることは本研究で示されたものの，埋め込みがどのような意味関係を捉えているのかは，明らかではない．関係パタンを用いた単語ペア埋め込みや，単語埋め込みの差分等が，どのような意味関係をどれくらい捉えられるのかを分析するのは重要である．第三に，語彙知識ベースとコーパスを組み合わせて意味関係知識の表現を獲得するこ

との重要性について述べた．現状，単語ペア埋め込みはコーパスのみから学習されているが，単語埋め込みの研究で行われているように，語彙知識ベースを学習のリソースとして追加で用いることで，効果的な学習が期待できる．本博士論文の研究においては，コーパスからの学習では対義関係を捉えることが難しいことを分析で述べたが（3.5.4 節，4.4.3 節），語彙知識ベース内の対義関係知識によって，そのような知識も単語ペア埋め込み空間で捉えられるかもしれない．最後に，近年，自然言語処理の水準を押し上げたニューラル言語モデル内に，どのような意味関係が含まれているかの分析と，言語モデルからの語彙知識の抽出について述べた．

謝辞

本研究をまとめるにあたり，粘り強くご指導いただいた指導教官の加藤恒昭先生，本論文に関して重要なご指摘を頂いた関根聡先生，林良彦先生，山口和紀先生，川崎義史先生に深く感謝する．特に，関根先生には，理化学研究所革新知能統合センターのパートタイムリサーチャーとして雇用していただき，様々なサポートをしていただいた．

そして，遊んだり議論したり助言をいただいたりした大学内外の方々にも感謝したい．彩りのある学生生活となったのは，みなさんのおかげである．

最後に，様々な面から支えてくれた父の鷲尾隆，母の鷲尾ひろみ，猫のシェルダンに感謝する．

参考文献

[1] Robert A. Amsler. A taxonomy for english nouns and verbs. In *Proceedings of the 19th Annual Meeting of the Association for Computational Linguistics*, pp. 133–138, Stanford, California, USA, June 1981. Association for Computational Linguistics.

[2] Maya Ando, Satoshi Sekine, and Shun Ishizaki. Automatic extraction of hyponyms from Japanese newspapers. using lexico-syntactic patterns. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May 2004. European Language Resources Association (ELRA).

[3] Marco Baroni, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-chieh Shan. Entailment above the word level in distributional semantics. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 23–32. Association for Computational Linguistics, 2012.

[4] Marco Baroni and Alessandro Lenci. How we blessed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on Geometrical Models of Natural Language Semantics*, pp. 1–10. Association for Computational Linguistics, 2011.

[5] Or Biran, Samuel Brody, and Noémie Elhadad. Putting it simply: a context-aware approach to lexical simplification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 496–501, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[6] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 1247–1250. AcM, 2008.

[7] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pp. 2787–2795. Curran

Associates, Inc., 2013.

[8] Tom Bosc and Pascal Vincent. Auto-encoding dictionary definitions into consistent word embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1522–1532, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[9] Elia Bruni, Nam-Khanh Tran, and Marco Baroni. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, Vol. 49, pp. 1–47, 2014.

[10] Jill Burstein, Jane Shore, John Sabatini, Yong-Won Lee, and Matthew Ventura. The automated text adaptation tool. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pp. 3–4, Rochester, New York, USA, April 2007. Association for Computational Linguistics.

[11] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, and Si Wei. Neural natural language inference models enhanced with external knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2406–2417. Association for Computational Linguistics, 2018.

[12] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1657–1668. Association for Computational Linguistics, 2017.

[13] 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. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics.

[14] Ido Dagan, Bill Dolan, Bernardo Magnini, and Dan Roth. Recognizing textual entailment: Rational, evaluation and approaches–erratum. *Natural Language Engineering*, Vol. 16, No. 1, pp. 105–105, 2010.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[16] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online

learning and stochastic optimization. *Journal of Machine Learning Research*, Vol. 12, No. Jul, pp. 2121–2159, 2011.

[17] A. H. Miller P. Lewis A. Bakhtin Y. Wu F. Petroni, T. Rocktäschel and S. Riedel. Language models as knowledge bases? In *To Appear in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2019*, 2019.

[18] Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1606–1615, Denver, Colorado, May–June 2015. Association for Computational Linguistics.

[19] Joshua Feldman, Joe Davison, and Alexander M. Rush. Commonsense knowledge mining from pretrained models. In *To Appear in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2019*, 2019.

[20] Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. Placing search in context: The concept revisited. *ACM Transactions on information systems*, Vol. 20, No. 1, pp. 116–131, 2002.

[21] Katrin Fundel, Robert Kffner, and Ralf Zimmer. RelEx—Relation extraction using dependency parse trees. *Bioinformatics*, Vol. 23, No. 3, pp. 365–371, 12 2006.

[22] Artyom Gadetsky, Ilya Yakubovskiy, and Dmitry Vetrov. Conditional generators of words definitions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 266–271, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[23] Maayan Geffet and Ido Dagan. The distributional inclusion hypotheses and lexical entailment. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pp. 107–114, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[24] Daniela Gerz, Ivan Vulić, Felix Hill, Roi Reichart, and Anna Korhonen. SimVerb-3500: A large-scale evaluation set of verb similarity. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2173–2182, Austin, Texas, November 2016. Association for Computational Linguistics.

[25] Roxana Girju, Adriana Badulescu, and Dan Moldovan. Automatic discovery of part-whole relations. *Computational Linguistics*, Vol. 32, No. 1, pp. 83–135, 2006.

[26] Guy Halawi, Gideon Dror, Evgeniy Gabrilovich, and Yehuda Koren. Large-scale learning of word relatedness with constraints. In *Proceedings of the 18th ACM*

SIGKDD international conference on Knowledge discovery and data mining, pp. 1406–1414. ACM, 2012.

[27] Patrick Hanks. The impact of corpora on dictionaries. In Paul Baker, editor, *Contemporary Corpus Linguistics*, pp. 214–236. Continuum, London, Great Britain, 2009.

[28] Sanda Harabagiu and Andrew Hickl. Methods for using textual entailment in open-domain question answering. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pp. 905–912. Association for Computational Linguistics, 2006.

[29] Sanda Harabagiu, Dan Moldovan, Marius Pasca, Rada Mihalcea, Mihai Surdeanu, Razvan Bunsecu, Roxana Girju, Vasile Rus, and Paul Morarescu. The role of lexico-semantic feedback in open-domain textual question-answering. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pp. 282–289, Toulouse, France, July 2001. Association for Computational Linguistics.

[30] Zellig S Harris. Distributional structure. *Word*, Vol. 10, No. 2-3, pp. 146–162, 1954.

[31] Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *COLING 1992 Volume 2: The 15th International Conference on Computational Linguistics*, 1992.

[32] Felix Hill, Roi Reichart, and Anna Korhonen. SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, Vol. 41, No. 4, pp. 665–695, December 2015.

[33] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.

[34] Eric Huang, Richard Socher, Christopher Manning, and Andrew Ng. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 873–882, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[35] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*, pp. 448–456, 2015.

[36] Shonosuke Ishiwatari, Hiroaki Hayashi, Naoki Yoshinaga, Graham Neubig, Shoetsu Sato, Masashi Toyoda, and Masaru Kitsuregawa. Learning to describe phrases with local and global contexts. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics,

2019.

[37] Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1681–1691, Beijing, China, July 2015. Association for Computational Linguistics.

[38] Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3651–3657, Florence, Italy, July 2019. Association for Computational Linguistics.

[39] Mandar Joshi, Eunsol Choi, Omer Levy, Daniel Weld, and Luke Zettlemoyer. pair2vec: Compositional word-pair embeddings for cross-sentence inference. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3597–3608, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[40] David Jurgens, Saif Mohammad, Peter Turney, and Keith Holyoak. Semeval-2012 task 2: Measuring degrees of relational similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pp. 356–364. Association for Computational Linguistics, 2012.

[41] Eliyahu Kiperwasser and Yoav Goldberg. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, Vol. 4, pp. 313–327, 2016.

[42] Omer Levy and Yoav Goldberg. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 302–308. Association for Computational Linguistics, 2014.

[43] Omer Levy and Yoav Goldberg. Linguistic regularities in sparse and explicit word representations. In *Proceedings of the eighteenth conference on computational natural language learning*, pp. 171–180, 2014.

[44] Omer Levy, Yoav Goldberg, and Ido Dagan. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, Vol. 3, pp. 211–225, 2015.

[45] Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. Do supervised distributional methods really learn lexical inference relations? In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 970–976. Association for Computational Linguistics, 2015.

[46] Dekang Lin, Shaojun Zhao, Lijuan Qin, and Ming Zhou. Identifying synonyms among distributionally similar words. In *IJCAI*, Vol. 3, pp. 1492–1493, 2003.

[47] Xiaodong Liu, Yelong Shen, Kevin Duh, and Jianfeng Gao. Stochastic answer networks for machine reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1694–1704, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[48] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, Vol. 9, No. Nov, pp. 2579–2605, 2008.

[49] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.

[50] 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.

[51] George Miller. *WordNet: An electronic lexical database*. MIT press, 1998.

[52] M Lynne Murphy. *Lexical meaning*. Cambridge University Press, 2010.

[53] Silvia Necsulescu, Sara Mendes, David Jurgens, Núria Bel, and Roberto Navigli. Reading between the lines: Overcoming data sparsity for accurate classification of lexical relationships. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pp. 182–192. Association for Computational Linguistics, 2015.

[54] Ke Ni and William Yang Wang. Learning to explain non-standard english words and phrases. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 413–417, Taipei, Taiwan, November 2017. Asian Federation of Natural Language Processing.

[55] Maximilian Nickel, Lorenzo Rosasco, Tomaso A Poggio, et al. Holographic embeddings of knowledge graphs. In *AAAI*, Vol. 2, pp. 3–2, 2016.

[56] Thanapon Noraset, Chen Liang, Larry Birnbaum, and Doug Downey. Definition modeling: Learning to define word embeddings in natural language. In *The Proceedings*

of the Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[57] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[58] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543. Association for Computational Linguistics, 2014.

[59] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[60] Kira Radinsky, Eugene Agichtein, Evgeniy Gabrilovich, and Shaul Markovitch. A word at a time: computing word relatedness using temporal semantic analysis. In *Proceedings of the 20th international conference on World wide web*, pp. 337–346. ACM, 2011.

[61] Bryan Rink and Sanda Harabagiu. Utd: Determining relational similarity using lexical patterns. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pp. 413–418. Association for Computational Linguistics, 2012.

[62] Stephen Roller and Katrin Erk. Relations such as hypernymy: Identifying and exploiting hearst patterns in distributional vectors for lexical entailment. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2163–2172. Association for Computational Linguistics, 2016.

[63] Stephen Roller, Katrin Erk, and Gemma Boleda. Inclusive yet selective: Supervised distributional hypernymy detection. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pp. 1025–1036. Dublin City University and Association for Computational Linguistics, 2014.

[64] Michael Roth and Sabine Schulte im Walde. Combining word patterns and discourse markers for paradigmatic relation classification. In *Proceedings of the 52nd Annual*

Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 524–530, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[65] Herbert Rubenstein and John B Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, Vol. 8, No. 10, pp. 627–633, 1965.

[66] Sara Rydin. Building a hyponymy lexicon with hierarchical structure. In *Proceedings of the ACL-02 Workshop on Unsupervised Lexical Acquisition*, pp. 26–33, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[67] Enrico Santus, Alessandro Lenci, Tin-Shing Chiu, Qin Lu, and Chu-Ren Huang. Nine features in a random forest to learn taxonomical semantic relations. In *LREC*, Portorož, Slovenia, 2016.

[68] Enrico Santus, Frances Yung, Alessandro Lenci, and Chu-Ren Huang. Evalution 1.0: an evolving semantic dataset for training and evaluation of distributional semantic models. In *Proceedings of The 4th Workshop on Linked Data in Linguistics (LDL-2015)*, pp. 64–69. Association for Computational Linguistics, 2015.

[69] Julian Seitner, Christian Bizer, Kai Eckert, Stefano Faralli, Robert Meusel, Heiko Paulheim, and Simone Paolo Ponzetto. A large database of hypernymy relations extracted from the web. In *LREC*, 2016.

[70] Vered Shwartz and Ido Dagan. Path-based vs. distributional information in recognizing lexical semantic relations. In *Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon (CogALex - V)*, pp. 24–29. The COLING 2016 Organizing Committee, 2016.

[71] 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 (Volume 1: Long Papers)*, pp. 2389–2398. Association for Computational Linguistics, 2016.

[72] Rion Snow, Daniel Jurafsky, and Andrew Y Ng. Learning syntactic patterns for automatic hypernym discovery. In *Advances in neural information processing systems*, pp. 1297–1304, 2005.

[73] Robyn Speer and Catherine Havasi. Representing general relational knowledge in ConceptNet 5. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pp. 3679–3686, Istanbul, Turkey, May 2012. European Languages Resources Association (ELRA).

[74] Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Compu-*

tational Linguistics, pp. 4593–4601, Florence, Italy, July 2019. Association for Computational Linguistics.

[75] Julien Tissier, Christophe Gravier, and Amaury Habrard. Dict2vec : Learning word embeddings using lexical dictionaries. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 254–263, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[76] Peter D. Turney. Measuring semantic similarity by latent relational analysis. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pp. 1136–1141, 2005.

[77] Peter D. Turney. Similarity of semantic relations. *Computational Linguistics*, Vol. 32, No. 3, pp. 379–416, 2006.

[78] Peter D Turney. The latent relation mapping engine: Algorithm and experiments. *Journal of Artificial Intelligence Research*, Vol. 33, pp. 615–655, 2008.

[79] Peter D Turney and Michael L Littman. Corpus-based learning of analogies and semantic relations. *Machine Learning*, Vol. 60, No. 1-3, pp. 251–278, 2005.

[80] Peter D Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, Vol. 37, pp. 141–188, 2010.

[81] Ellen M Voorhees. Query expansion using lexical-semantic relations. SIGIR' 94, pp. 61–69. Springer, 1994.

[82] Ekaterina Vylomova, Laura Rimell, Trevor Cohn, and Timothy Baldwin. Take and took, gaggle and goose, book and read: Evaluating the utility of vector differences for lexical relation learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1671–1682. Association for Computational Linguistics, 2016.

[83] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 29, No. 12, pp. 2724–2743, 2017.

[84] Julie Weeds, Daoud Clarke, Jeremy Reffin, David Weir, and Bill Keller. Learning to distinguish hypernyms and co-hyponyms. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pp. 2249–2259. Dublin City University and Association for Computational Linguistics, 2014.

[85] Julie Weeds, David Weir, and Diana McCarthy. Characterising measures of lexical distributional similarity. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, 2004.

[86] Chang Xu, Yalong Bai, Jiang Bian, Bin Gao, Gang Wang, Xiaoguang Liu, and Tie-Yan Liu. Rc-net: A general framework for incorporating knowledge into word representations. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, pp. 1219–1228. ACM, 2014.

[87] Yadollah Yaghoobzadeh and Hinrich Schütze. Intrinsic subspace evaluation of word embedding representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 236–246, Berlin, Germany, August 2016. Association for Computational Linguistics.

[88] Mo Yu and Mark Dredze. Improving lexical embeddings with semantic knowledge. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 545–550, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[89] Alisa Zhila, Wen-tau Yih, Christopher Meek, Geoffrey Zweig, and Tomas Mikolov. Combining heterogeneous models for measuring relational similarity. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1000–1009, Atlanta, Georgia, June 2013. Association for Computational Linguistics.