B4IM2502

修士論文

文章理解のための談話内文脈の表現学習

小林 颯介

2016 年 8 月 22 日

東北大学 大学院
情報科学研究科 システム情報科学専攻

本論文は東北大学 大学院情報科学研究科 システム情報科学専攻に
修士 (情報科学) 授与の要件として提出した修士論文である.

小林 颯介

審査委員：

乾 健太郎 教授　　（主指導教員）
木下 賢吾 教授
伊藤 彰則 教授
岡崎 直観 准教授　（副指導教員）

文章理解のための談話内文脈の表現学習*

小林 颯介

内容梗概

深い言語理解のためには，文章中の各文を独立的に解釈するのではなく，前後の一貫した文脈を踏まえた大局的な意味解釈が重要である．例えば，ある文で登場人物が犯罪を犯した事象が記述された場合，後の文でその人物が逮捕されたり，さらに犯罪を重ねたりする事象の自然さは高まる．これは常識的推論による選択選好性に基づいている．このような解釈の自動処理が実現すると，言語理解の中でも照応解析（後続文の「“彼” が逮捕された.」における “彼” の特定）や事象間の因果推論（「彼が逮捕された.」と「彼が祝福された.」のどちらが自然か）などに役に立つと考えられる．本研究では，文章内の各登場人物やエンティティに個々の分散表現を割り当て，対象が文に登場する度に，その時点での表現を単語分散表現の代わりに用い，また，その文を読んだ後に対象の周辺文脈によって分散表現を更新する手法を提案する．そして，言語モデルと文章読解による要約穴埋め問題の 2 種類のタスクでの評価実験により提案手法の有用性を示す．

キーワード

自然言語処理，表現学習，文章理解，談話処理, 事象間関係, 分散表現

*東北大学 大学院情報科学研究科 システム情報科学専攻 修士論文, B4IM2502, 2016 年 8 月 22 日.

目 次

図 目 次

表 目 次

1　はじめに

1.1　背景

自然言語処理の究極の目標のうちの一つは，コンピュータによる自動的な文章の読解である．コンピュータが大量の文章を人間同様に正しく理解し，それらの情報を整理することができれば，日々生み出される Web ページ，学術論文，医療カルテ，判例などを処理させることで，例えば，人間の活動を補助するエキスパートシステムとなることが期待できる．ところで，文章を意味的に正しく理解することの定義として唯一無二の確立されたものはない．しかし，文章読解の能力を測る方法として，ある文章を与えられた後にその文章内で述べられた事象について質問することが提案されている．例えば，単純な例として，次のような 1 文からなる文章 (1t) と穴埋め形式の質問文 (1q) を考える．

(1) t. *John is the president of the U.S.*

q. *[X] is the president.*

このような質問応答では，まず質問文 (1q) のプレースホルダ *[X]* に関する局所文脈（この例では *is the president*）を把握し，それと意味的に類似した局所文脈を持つ名詞句（解答候補）を文章中から探すのが基本的なアプローチの 1 つとして考えられる．

このような読解問題は，難易度の差異はあれど大学入試センター試験を始めとした数々の試験においても文章読解能力のテストに用いられており，その問題をコンピュータが解くことを目指す東ロボ（ロボットは東大に入れるか）プロジェクトも行われている．大学入試問題を含めて，数々の文章読解用のデータセットはこれまで提案されてきたものの，それらの規模は非常に小さいため，性能比較の指標として分散が大きく頑健でなく，また，十分な訓練用データが無く機械学習アプローチを行いづらいという問題点もあった．しかし，近年大規模な文章読解用の質問応答のデータセット CNN QA Dataset が Hermann らによって公開された [2]．また，Hermann らはニューラルネットワークを用いた質問応答モデルを提案し，単語ベクトルを含めたモデルパラメータをランダムな初期値にして学習を始めても，質問応答の解答の誤差を逆伝搬するだけで高い性能が出るまで学

習が行えることを示した．

このように大量のデータからニューラルネットワークを一から学習するアプローチは，画像・音声処理やその他の分野のみならず，自然言語処理分野においても大きな存在感を示している．機械翻訳，極性分類，文書分類，文書要約，質問応答，画像説明文生成などの応用的なタスクだけでなく，文構造解析，品詞タグ付与，単語分割，言語モデル [3] といった基礎的なタスクについても研究が数多く行われている．タスクにも応じて様々なモデルが提案なされているが，各単語を一つの分散表現（単語ベクトル）として表現した後に，それらをフィードフォーワードニューラルネットワーク, リカレントニューラルネットワーク, 畳み込みニューラルネットワークなどに入力して予測を得るという共通点を持つものが大半である．単語の分散表現はタスクの学習時に同時に誤差逆伝播法で学習することもあれば，別の手法により事前訓練（pretrain）したものを用いることもある．

これまで多くの研究では (i) 各文ごとの独立的に処理を行い，かつ，(ii) 入力される単語ベクトルは単語の表層毎に定義された静的なものである．(i) については，そもそも文単位でのタスクが多かったことが原因の一つとしてあげられる．しかし，文章読解という複数の文からなる談話構造を踏まえなければならないタスクにおいて高度な理解を実現するためには，これまでとは異なる大局的な処理を行うことが適切だと考えられる．例として，文章中の複数の情報を組み合わせて初めて解答できるような質問を以下に示す．

(2) t. *John is the president of the U.S.*
Jacqueline is the wife of John.
q. *[X] is the wife of the president.*

(2) のような質問に答えるには，(2t) の 1 文目の *John* の局所文脈 *is the president* と 2 文目の *Jacqueline* の局所文脈 *is the wife of John* を組み合わせて *is the wife of John, who is the president* のような情報を解答候補 *Jacqueline* に関する文脈情報として把握する必要がある．こうした情報の組み合わせ（集約）が上手くモデル化できれば単なる質問応答を越えて談話の理解に一歩近づくと考えられるが，これまでの分散表現に基づく質問応答の研究ではこうした現象を扱えていない．(ii) については，一つの単語は大なり小なり複数の意味をもつという「多義性」の

考慮が欠けている問題がある．その対策として，各単語ごとではなく各意味毎にベクトルを学習して定義しておき，活用時にも語義曖昧性解消を行った上でベクトルを分けて用いるアプローチに関する研究などがあった [4]．しかし，例えば “John” という単語を含むようなタイトルに含むような Wikipedia の記事は 5 万弱 (2016 年 6 月 6 日時点) 存在する．それほど多くの “John” が存在する中で，全 “John” に対応するような汎用的なベクトルを学習することや語義毎にベクトルを獲得し分別して使用するアプローチについても困難だと考えられる．加えて，多義性のみならず，言語には「新語」や「新たな語義」が常に生み出されていくという本質的な特性がある．したがって，あらゆる全ての単語・語義に対して，事前に単語ベクトルを学習や定義しておくことは根本的に不可能であるといえる．一方で，当然人間も文章を読むときに「未知語」に出会うことがしばしばある．しかし，人間は未知語の意味を文章内で推測し，同時にその推測した意味や属性の情報を用いながら文章全体を理解することが出来る．これに関する一つの仮説として，文章に初めて “John” が現れた時点では，意味や素性についてはほぼ空であったものの，文章を読み進めるにつれて “John” の周辺の文脈から情報を集め，後続文などの理解に生かすような処理を行っているのではないかと考えられる．

そこで本稿では，このように文章を読み進めながら，エンティティの情報を集め同時にその情報を談話内での意味理解に生かす人間の処理モデルをニューラルネットワークによって実現する方法を提案する．また，上で述べた大規模な質問応答データセット CNN QA[2] における要約的読解での評価実験，及び新たに提案する共参照付き言語モデルタスクでの評価実験の結果から，提案手法が正答率及び予測誤差の改善に貢献することを示す．

以下に本研究の貢献をまとめる．

- エンティティの分散表現を文章中の文脈情報から動的に構築し活用する手法（動的分散表現; Dynamic Entity Representation; DER）の提案
- 動的分散表現を用いた文章読解モデルの提案
- 動的分散表現を用いた言語モデルの提案
- エンティティごとの新たな言語モデルタスクの提案

- 言語モデルタスクと要約的な文章読解質問応答タスクによる動的分散表現の評価実験

1.2 本論文の構成

本論文の構成は以下の通りである．2 章で本研究に関連のある，言語の分散表現の学習および深層学習を用いた意味合成に関する研究を概観する．3 章では，文章読解に関するデータセット CNNQA および，新たに作成した言語モデル用のデータセットについて説明する．そして，文章を読み進めながら文脈意味表現を構築し活用する提案手法，動的分散表現について 4 章で説明する．5 章にて，文章に関する要約的な質問応答に適用できる読解モデルと，提案手法によるモデルの拡張について説明を行う．6 章では，提案手法による言語モデルの拡張について説明する．そして，7 章では，提案手法の動的分散表現についての評価実験を CNN QA データセットを用いた要約的な質問応答と言語モデルの 2 つのタスクについて行い，提案手法の効果を示すとともに結果を考察する．最後に 8 章で本研究の総括を行う．

2　関連研究

本章では本研究の目的である文・句・文脈の分散的意味表現の学習手法及び文章読解に関する既存研究について述べる．

2.1　ニューラルネットワーク

ニューラルネットワークの最も基本な形は，ある固定次元の入力ベクトル x に対して，非線形関数 f を用いた関数 $f(Wx+b)$ を複数回適用したものによる，ベクトル x からスカラー y を求める関数であり（複数のスカラー値を求めることで，ベクトルを出力する形に自然に拡張できる），フィードフォーワードニューラルネットワークと呼ばれる．その内部のパラメータとなる行列 W 及びバイアスベクトル b は，最終的な出力 y と解 t との誤差を逆伝播することによって得られる勾配によって誤差を減らす方向へ最適化が行われる．十分なパラメータを持つニューラルネットワークは任意の関数を近似できる表現能力があることが知られている．また，複数のベクトルが結合されたもの，あるいは画像のような二次元的な構造をもったデータに対して，適用範囲をずらしながら同じパラメータでフィードフォーワードニューラルネットワークを適用する畳み込みニューラルネットワークは，画像分野で大きな成果を挙げている．また，複数のベクトルが系列となっている場合には，リカレントニューラルネットワークによる処理も適している．リカレントニューラルネットワークでは，ある時刻におけるベクトルの出力を，その時刻に対応するベクトルデータと，一時刻前に出力されたベクトルを用いて決定する再帰的な構造を持っている．処理時に参照するデータは一時刻前のベクトルまでであるが，そのベクトルもまた再帰的に初期のベクトルから計算されてきたものであるため，ベクトル系列の性質を長期依存を含めて関数で表現することができる表現力がある．文が単語の系列として表される自然言語処理や時系列音響データを扱う音声処理の分野では特に盛んに用いられている．

2.2　言語の分散表現の学習

自然言語処理の分野では，単語をベクトルによって表現するアプローチは古くから存在していた．最もよく用いられていたものは，分布意味論から着想を得た単語共起頻度を用いたベクトルである．分布意味論とは，単語の意味はその単語が現れる文脈（周辺の情報）によって決められる（予測できる）という考えである．そこで，言語のコーパスを用いて各単語が出現したときにどのような単語と共起したかを計算し，それをベクトルとして表現するアプローチが生まれた．例えば，ある単語について周辺 2 単語以内に出現する単語を数えあげて，ベクトルの 1 次元目を「“red” が周辺 2 単語以内に出現した回数」，ベクトルの 2 次元目を「“fruit” が周辺 2 単語以内に出現した回数」，のように，それぞれのベクトルに共起頻度を割り当てることが考えられる．すると，この例で言えば，“apple” のような単語は 1 次元目も 2 次元目も高いようなベクトルになることが想像できる．この対象となる単語が十分に多ければ，各単語の意味を判別することに十分な特徴が得られる可能性がある．単純な共起頻度ではなく，tf-idf などを用いて，より適切に相対的な特徴量を得ようとする試みも盛んに行われた．しかし，単語の多くは十分な回数だけコーパス中に出現するとは限らない．また，本来用いられてもおかしくはない事例についても，有限のデータの中ではそのような事例をすべて網羅することはできない．このようなデータのスパース性により，単に共起頻度を単純に数えるだけのベクトルは多くの値がゼロになったり，使用したコーパスの偏りやノイズによる影響が強く出たものになってしまう．また，その特徴量を用いてなんらかの分類器を学習しても，過学習が起きやすくなってしまう．そこで，その問題の解消のために行列分解などにより元のベクトル（を総じた際の行列）を低次元に再構成するアプローチが行われる．

上のような数え上げベースの伝統的なアプローチに加えて，近年では新たなアプローチが登場し，盛んに研究が行われ始めた．その代表的なものとして，Mikolov らの Skip-gram[5] を説明する．これはまず初めに各単語に固定次元の実数値ベクトルを割り当て，その後に，その単語がコーパス中に出現する度に，周辺の共起している単語のベクトルとその単語のベクトルが近づくように最適化を行う．非常に単純なモデルながら，学習を終えた後のベクトルは良好な特徴量を獲得でき

ていることが示されており，数え上げによるベクトルを行列分解したものよりも単語関連度算出タスクなどで良い性能を示すことが報告されている．また，複数の単語を足し合わせたり引いたりすることにより意味の演算が行われているかのような現象も確認されている．

複数の単語の意味表現から，そのフレーズや文の意味表現を構築する手法についても研究が盛んである．最もシンプルな形としては，各単語のベクトルを単に足しあわせて総和や平均をとるものであり，単純ながら強力なベースライン手法として用いられる．また，リカレントニューラルネットワーク，リカーシブニューラルネットワーク，畳み込みニューラルネットワークを用いた構成アプローチも多く提案され，中でもリカレントニューラルネットワーク（RNN）を用いた手法が多くのタスクで高性能をあげている．リカレントニューラルネットワークはベクトル系列の中のベクトルを 1 つずつ受け取り，各時刻で 1 つずつベクトルを出力する．そのため，文を単語ベクトルの系列としてみなし，その系列を最後まで処理した後の出力を文のベクトルとして扱うことで文のベクトル化（エンコード）モデルとすることが一般的である．文ベクトルを分類器（e.g., ニューラルネットワーク，SVM）にかけることで極性分類などの文ごとにラベルがついた分類タスクを行ったり，また，文ベクトルを入力として RNN 言語モデル [3] を用いることで機械翻訳などの文生成を行う場合もある．それらもまた，誤差逆伝播法を用いて単語ベクトルまで含めて学習を行うことが可能である．

ニューラルネットワークによるエンコードの研究は数多い．その一方で，その入力となるベクトルに焦点をおいた研究は数少ない．ほぼ全ての研究が，各単語に応じて 1 つ割り当てられた静的なベクトルを用いている．文章読解にニューラルネットワークで取り組んだ先行研究でなる Hermann ら [2] の Attentive Reader や Hill ら [6] の Memory Networks も同様に静的なベクトルを用いている．文章中に単語表層が未知である変数表現 (e.g., *@entity7*) が出てきたときも同様に，*police, loves, at* などのような一般単語と同様に結び付けられた唯一のベクトルを用いている．しかし，本研究では，変数表現（やエンティティごとの表層単語）は，文章中に現れるエンティティ同士の共参照を束縛するための単なる記号だと見なす．各単語自体に静的な意味が割り当てられているわけではなく，その実際的な意味はそ

の単語の結びついた実際のエンティティの意味を動的に反映するものである．各エンティティの談話中での意味共有を考慮することは，自然言語理解において重要であり，これまでにも，event inference [7, 8], semantic roles [9], discourse relations [10], coherence [11] and coreference resolution [12] など幅広いタスクで活用されてきた．例えばRothとLapata[9]は，談話内で各エンティティの意味役割が共起する傾向を意味役割付与に活用した．しかし，分散表現においてエンティティの共有関係を考慮し，さらに，分散表現自体を文脈から動的に構築する手法は，本研究が初である．本研究で提案する**動的分散表現**（*dynamic entity representation*）は特に文章読解に特化した手法ではなく，上で述べたような様々なタスクに活用することも考えられる一般的なものである．また，エンティティの表層を記号的なリンクと見なす分散表現アプローチは，neural-symbolic integration[13]の観点からも新しいものである．

単語の分散表現を表層と一対に結び付けない研究は幾つか存在する．LiとJurafsky [4]は，単語の分散表現を多義性を推論しながら選択的に学習し，活用時にも語義曖昧性解消と組み合わせて分散表現を用いるアプローチが，幾つかのタスクで性能向上に寄与することを示した．Chengと Kartsaklis [14]は，単語の分散表現の曖昧性解消をディープニューラルネットワークの構造内部で行うことを目指した．加えて，積層のRNN[15]も本研究が期待する効果を実現しうると考えられる．一層目のRNNは静的な単語分散表現を受け取るものの，それより深い層では過去の文章内で得た情報を，静的な単語表現にマージする効果を学習で獲得する可能性があるが，学習の難易度は非常に高いと考えられる．また，そのような効果を解析した研究はこれまでにない．

2.3　アテンションメカニズム

近年，Bahdanauら [1]によってアテンションメカニズムが提案された．広義には，あるクエリとなるベクトルと，対象となる複数のベクトルがあった場合に，クエリの内容に応じて対象ベクトルごとの重みを算出する手法となっている．Bahdanauらは，RNNによる機械翻訳モデルについて，入力文を全てエンコード（一つの固定次元のベクトルに圧縮）するそれまでのアプローチについて学習・情

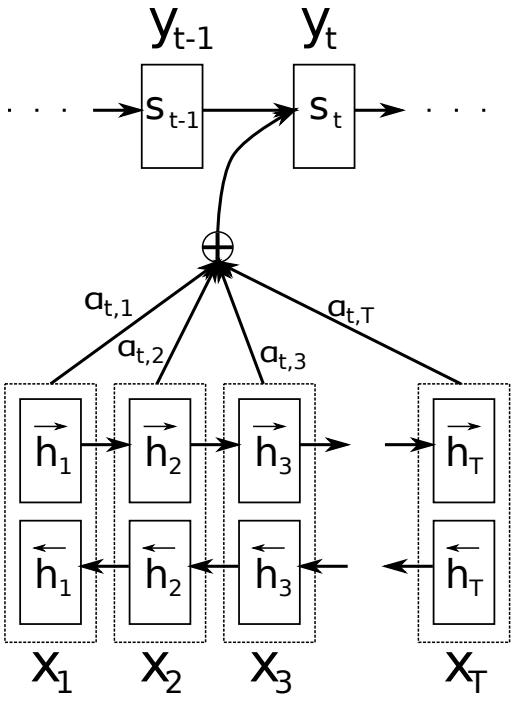
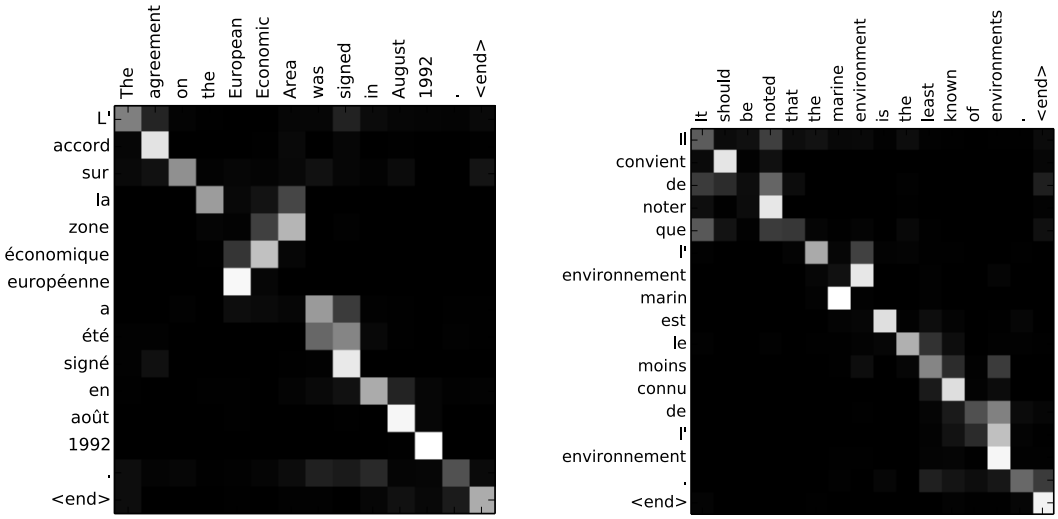


図 1: Bahdanau ら [1] の提案モデル図．入力文の単語が x_t，処理した RNN の隠れ層が $\overrightarrow{h}_t, \overleftarrow{h}_t$，アテンションの重みが $\alpha_{t,t'}$，出力時の RNN の隠れ層が s_t，出力単語が y_t を表している．論文より引用．

報の伝搬が難しくなっている点を指摘し，新たにアテンションメカニズムを考案した．RNN 機械翻訳モデルは，sequence to sequence モデルと呼ばれるアプローチをベースとしており，入力文の単語を一時刻につき一単語ずつ入力し最終的な固定次元のベクトルが生まれ，その後にそれを入力として，出力文を一時刻につき一単語ずつ出力するという形式をとっていた．Bahdanau らは，この出力の各時刻において，その時点での隠れ状態をクエリベクトルとして入力文処理時の各時刻の隠れ状態について重みを算出して足し合わせたベクトルを，その都度新たに入力として受け取る拡張を行った．その提案モデルを図 1 に示す．また，アテンションメカニズムを適用した際の重みの計算結果の具体例を図 2 に示す．翻訳先言語であるフランス語の文を出力していく過程で，各単語を出力する際にどの元文の



(a) (b)

図 2: Bahdanau ら [1] の提案モデルのアテンションメカニズムにより算出された重みを表した行列．英語（上軸）からフランス語（左軸）へ翻訳した際の実験結果．黒いほど重みが 0 に近く，白いほど重みが強い（1 に近い）ことを表す．論文より引用．

英単語に注目しているのかが分かる．(a) の行列を見ると，L'->The（“定冠詞”）, accord->agreement（一致）, économique->Economic（経済の）, August->août（8 月） など，注目した単語に応じた単語を出力していることが分かる．この際のアテンションメカニズムのクエリベクトルは前時刻の隠れ層のため，例えば，前時刻に août（8 月）という単語を出力した情報を手がかりに次に年号が来やすいという選好性を踏まえて，単語 1992 に注目が行われているという解釈が考えられる．

重み付けの計算方法は様々な方法が考えられるが [16]，重み計算の全てがニューラルネットワークによる連続的な計算で与えられるため，その重み計算のための行列パラメータも誤差逆伝播法による学習が可能である．アテンションメカニズムはその拡張性の高さから，機械翻訳 [1, 16], 画像キャプション生成 [17], 文書要約 [18], 含意関係認識 [19], 質問応答 [20, 21, 2, 6] など，幅広いタスクに用いられ

て高い性能を示している．質問応答に関しては，問題文を RNN などでエンコードしたベクトルをクエリとし，それを用いて解答のための知識や証拠を集めるためにアテンションメカニズムを用いるのが一般的な設定となっている．

2.4 文章読解

本章では，CNN QA データセットや他のタスクに適用された具体的な関連研究を紹介する．

まず，CNN QA データセットを公開した Hermann ら [2] は，アテンションメカニズムを用いた Attentive Reader 及び Impatient Reader を提案した．両者の性能は拮抗しているため，ここでは比較的簡易なモデルである Attentive Reader について述べる．Attentive Reader は，まず文章中の全単語を（文も全てつなげた系列として）bi-directional LSTM によって処理を行い，それら全ての時点での出力についてアテンションメカニズムを適用する．そして，その重みに応じて出力を足しあわせて，その結果がどの変数ベクトルと似ているかを元にして解を予測する．しかし，アテンションメカニズムの適用範囲を事前に狭めることによってモデルの性能が上がることが近年 Luong ら [16] や Xu ら [17] によって示されている．一方で本研究では，各文ごとの各エンティティごとのベクトルへエンコードした上で，さらにエンティティごとにアテンションメカニズムを適用するため，アテンションメカニズムの範囲を狭めることに成功している．また，最終的な予測についても，静的な変数ベクトルとマッチングを行うのではなく，動的に構築したエンティティベクトルとのマッチングを行う．

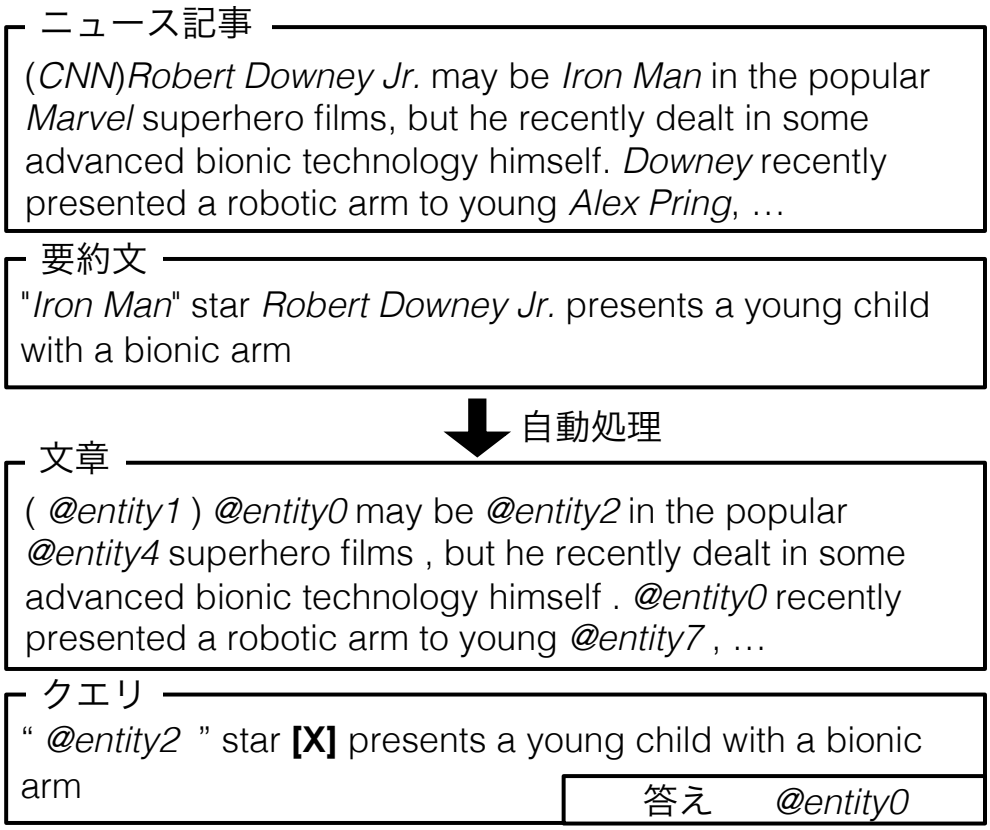


図 3: CNN QA の問題例．下部が実際の QA データ．

3 CNN QA Dataset

本研究で使用するデータセットについて述べる．

本研究では評価実験のために，Hermann らが近年公開した CNN QA データセット[1][2] を用いる．図 3 に示すように，〈文章, 穴埋め質問文, 答え〉の三つ組を 1 問のデータとしており，問題はすべてニュースサイト[2]の記事文章とその要約文を用いて自動で構築したものである．なお，特徴的な点として全ての固有表現（例. “*Robert Downey Jr.*”, “*Downey*”）は共参照関係を自動解析され，その関係リンクを保ったままランダムな変数表現（例. *@entity0*）へと置換されている．また，

[1] https://github.com/deepmind/rc-data
[2] http://www.cnn.com/

モデルの訓練時にはデータ中の変数を毎回シャッフルして訓練を行う．そのため，エンティティの表層は分からず，事前の背景知識の影響を抑えた，より純粋な文章読解力のテストを行えるようになっている．この問題設定は，まさに 1.1 で述べた，各エンティティ（の単語・その意味）を知らないという状態である．よって，読み進めていく中で意味を把握しながら，後続文や質問の理解に生かすという提案手法をテストするのにも適している．訓練用データには約 38 万問（約 9 万記事），開発用データ（Valid）及びテスト用データ（Test）には約 3 千問（約 1 千記事）を含む．平均すると，記事内には約 25 種類のエンティティの変数表現を含み，記事の長さは約 700 語である．

3.1 言語モデル用データセット

本研究では質問応答タスクに加えて，言語モデルによっても提案手法の評価を行う．そのため，固有名詞の名詞句を共有変数表現へと匿名化を施し，さらに共参照関係にある代名詞についても同じ変数表現へと変換した文章データセットを，CoNLL-2012 の共参照解析の Shared Task で使われた OntoNotes [22] から構築した．また，著者への外界照応などの代名詞（e.g., I, you）も同様に変数表現へと置換した．元のデータセットから，訓練用の 2725 データ，サンプルした開発用とテスト用それぞれ 100 記事を用いた．変数の異なり数が 50 以下のみ出現する記事のみに絞った．また，訓練記事内での出現頻度上位 9947 個の単語，文頭記号，文末記号，未知語記号，50 個の変数からなる合計 10000 個の記号を語彙集合とし，各データ内の語彙外の単語は全て未知語記号へと予め置換した．

4　談話内における動的分散表現

動的分散表現は各エンティティに割り当てられ，初期値は対応する変数単語ベクトル x_e とする．エンティティ e が文 c（文長 T_s）の τ 番目に出現した際の文脈の分散表現 $\boldsymbol{d}_{e,c}$ は，以下のように獲得される．

$$\boldsymbol{d}_{e,c} = \tanh(W_{hd}[\vec{\boldsymbol{h}}_{c,\tau}, \overleftarrow{\boldsymbol{h}}_{c,\tau}] + \boldsymbol{b}_d) \tag{1}$$

$$\boldsymbol{d}_{e,c} = \tanh(W_{hd}[\vec{\boldsymbol{h}}_{c,\tau-1}, \overleftarrow{\boldsymbol{h}}_{c,\tau+1}] + \boldsymbol{b}_d) \tag{2}$$

$$\vec{\boldsymbol{h}}_{c,t} = \overrightarrow{RNN}(\boldsymbol{x}_{c,t}, \vec{\boldsymbol{h}}_{c,t-1}) \quad \text{(順方向)} \tag{3}$$

$$\overleftarrow{\boldsymbol{h}}_{c,t} = \overleftarrow{RNN}(\boldsymbol{x}_{c,t}, \overleftarrow{\boldsymbol{h}}_{c,t+1}) \quad \text{(逆方向)} \tag{4}$$

文中の各トークンに対応する分散表現 $\boldsymbol{x}_{c,t}$ を bidirectional RNN によって処理し，対象トークンの位置 τ における出力，あるいは，その位置を挟み込むような出力を結合したものをフィードフォーワードニューラルネットワークにより合成することで文脈表現 $\boldsymbol{d}_{e,c}$ を得る．

文章中にエンティティ e が複数出現した場合，文 C まで時点での文 $c^1, c^2, ... \prec C$ について，文脈表現は $\boldsymbol{d}_{e,c^1}, \boldsymbol{d}_{e,c^2}, ...$ のように複数個得られる．それら $N_{e,C}$ 個を統合した文脈の分散表現 $\boldsymbol{d}_{e,\prec C}$ は，RNN，マックスプーリング，平均プーリング，総和などによって実現できる．

$$\boldsymbol{d}_{e,\prec C} = \overrightarrow{RNN'}_{c' \prec \mathrm{C}}(\boldsymbol{d}_{e,c'}) \tag{5}$$

$$\boldsymbol{d}_{e,\prec C} = \underset{c' \prec C}{\text{max-pooling}}(\boldsymbol{d}_{e,c'}) \tag{6}$$

$$\boldsymbol{d}_{e,\prec C} = \sum_{c' \prec C} \boldsymbol{d}_{e,c'} \tag{7}$$

$$\boldsymbol{d}_{e,\prec C} = \frac{1}{N_{e,C}} \sum_{c' \prec C} \boldsymbol{d}_{e,c'} \tag{8}$$

統合された後の分散表現をもとにしたものを，エンティティの単語ベクトルや言語モデルの予測行列の列ベクトルに代用することで，談話内の文脈情報を考慮した処理を行う．この処理は，一度変数が登場した後にのみ行う．

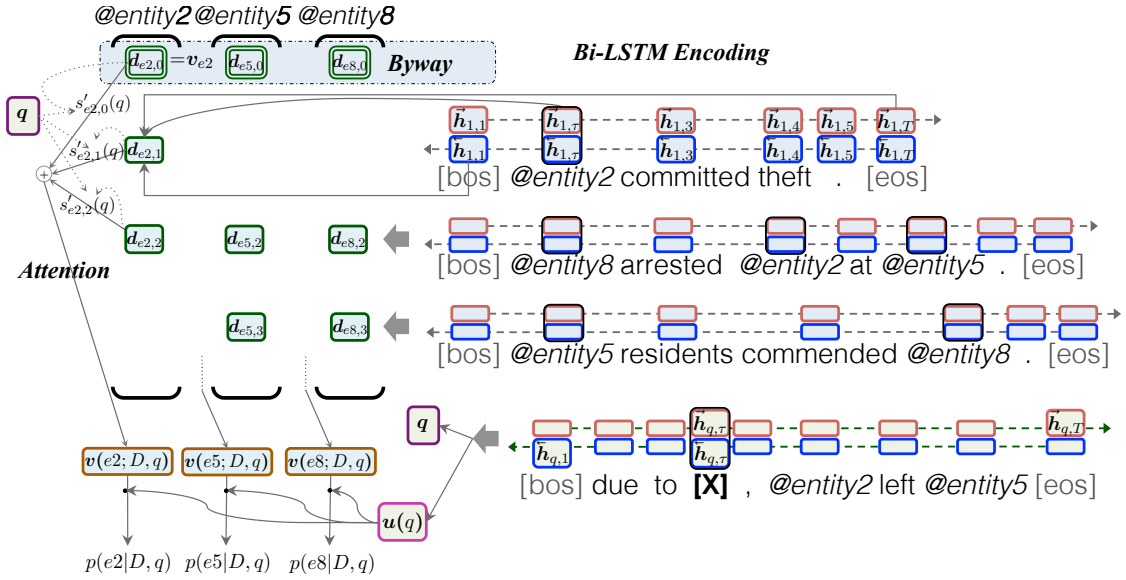


図 4: DER-Network による質問応答モデル.

5 エンティティごとの文脈情報に注目した読解モデル

全体のフレームワークを図 4 に示す.

はじめに個々の解答候補の個々の出現（mention）について，その局所文脈を分散表現（ベクトル）にエンコードすることを考える．これには双方向 LSTM[23] を用いる．双方向 LSTM は単語列の情報を分散表現にエンコードするのにしばしば用いられる方法で，次の漸化式 (9)(10) で与えられる．

$$\vec{\boldsymbol{h}}_{c,t} = \overrightarrow{LSTM}(\boldsymbol{x}_{c,t}, \vec{\boldsymbol{h}}_{c,t-1}) \quad \text{(順方向)} \tag{9}$$

$$\overleftarrow{\boldsymbol{h}}_{c,t} = \overleftarrow{LSTM}(\boldsymbol{x}_{c,t}, \overleftarrow{\boldsymbol{h}}_{c,t+1}) \quad \text{(逆方向)} \tag{10}$$

(9) の $\vec{\boldsymbol{h}}_{c,t}$ は，文 c の 1 番目の単語（文頭記号）から t 番目の単語までの局所文脈をエンコードした分散表現である．同様に (10) の $\overleftarrow{\boldsymbol{h}}_{c,t}$ は，文 c の文末の単語（文末記号）から t 番目の単語までの局所文脈をエンコードした分散表現である．

これらを使って文 c における解答候補 e の出現（τ 番目の単語とする）に対する局所文脈の分散表現（ベクトル）$\boldsymbol{d}_{e,c}$ を図 5 のように計算する．まず，解答候補の出現の左右文脈 $\vec{\boldsymbol{h}}_{c,\tau}$，$\overleftarrow{\boldsymbol{h}}_{c,\tau}$ を計算する．次に，文頭から文末までの単語列の

ベクトル $\vec{\boldsymbol{h}}_{c,T}$ と文末から文頭までのベクトル $\overleftarrow{\boldsymbol{h}}_{c,1}$ を計算する．最後に，これら 4 つのベクトルを結合し，次式の変換を施して $\boldsymbol{d}_{e,c}$ を得る．

$$\boldsymbol{d}_{e,c} = \tanh(W_{hd}[\vec{\boldsymbol{h}}_{c,T}, \overleftarrow{\boldsymbol{h}}_{c,1}, \vec{\boldsymbol{h}}_{c,\tau}, \overleftarrow{\boldsymbol{h}}_{c,\tau}] + \boldsymbol{b}_d) \tag{11}$$

$\boldsymbol{d}_{e,c}$ は，対象エンティティを囲むような左右の文脈に加えて文全体の情報を捉えたような意味表現になっている．なお，W_{hd} は行列，$\boldsymbol{b}_d$ はバイアスベクトルであり，いずれも学習で調整する[3]．

また，質問文 q についても同様に，プレースホルダの位置を τ とするとき，その局所文脈を次式で計算する．

$$\boldsymbol{u}(q) = W_{hq}[\vec{\boldsymbol{h}}_{q,T}, \overleftarrow{\boldsymbol{h}}_{q,1}, \vec{\boldsymbol{h}}_{q,\tau}, \overleftarrow{\boldsymbol{h}}_{q,\tau}] + \boldsymbol{b}_q \tag{12}$$

質問応答では基本的には，$\boldsymbol{u}(q)$ に最も近い局所文脈を持つ解答候補を探せばよい．

5.1 局所文脈の集約

次に，同じ解答候補 e が談話内で複数回出現する状況を考える．それぞれの出現が局所文脈 $\boldsymbol{d}_{e,c}$ を持つので，それらを重み付き平均で集約する．このとき，直感的には，質問文に近い局所文脈により大きな重みを与えるようにすればよいと考えられる．そこで，そうした重み付き平均の制御を最適化する方法として，近年統計的機械翻訳やキャプション生成などに適用され始めたアテンションメカニズム（attention mechanism）[1, 17] を使う．具体的には，まず，質問文 $\boldsymbol{q}$[4] と個々の出現文脈 $\boldsymbol{d}_{e,c'}$ の関連度 $s'_{e,c'}(q)$ を式 (5) で計算し，式 (6) で正規化する[5]．

$$s'_{e,c'}(q) = \boldsymbol{m}^{\mathrm{T}} \tanh(W_{dm}\boldsymbol{d}_{e,c'} + \boldsymbol{q}) + b_s \tag{13}$$

$$s_{e,c}(q) = \frac{\exp(s'_{e,c}(q))}{\sum_{c'} \exp(s'_{e,c'}(q))} \tag{14}$$

[3] 添字 hd は，W_{hd} が層 $\boldsymbol{h}$ のベクトルを層 $\boldsymbol{d}$ のベクトルに写像する行列であることを表す．添字 d は，$\boldsymbol{b}_d$ が層 $\boldsymbol{d}$ のベクトルと同じ次元数であることを表す．本稿では以下でもこの表記法を用いる．

[4] ベクトル $\boldsymbol{q}$ は式 (12) のパラメータを変えた同様の計算で求める．

[5] ベクトル $\boldsymbol{m}$，行列 W_{dm}，スカラー値 b_s は，アテンションメカニズムのための学習パラメータである．

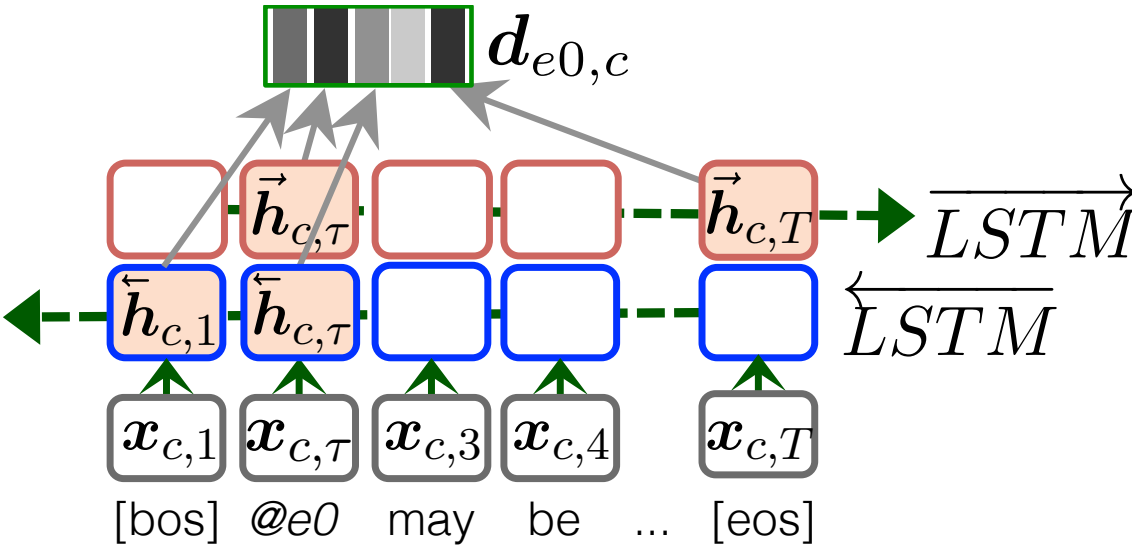


図 5: 文 c における $@e0$ の文脈情報 $\boldsymbol{d}_{e0,c}$ のエンコード.

ここで得られる $s_{e,c}(q)$ は，質問文 q が与えられたときに，解答候補 e の出現のうちどの出現に注目すべきかを数値化したものと解釈できる．アテンションメカニズムではこうして計算される注目の大きさを「アテンション（注目度）」と呼んでおり，この注目度を使って重み付き平均を次式のように計算する．

$$\boldsymbol{v}(e;D,q) = W_{dv}\Big[\sum_{c\in D} s_{e,c}(q)\boldsymbol{d}_{e,c}\Big] + \boldsymbol{b}_v \tag{15}$$

こうして得られるベクトル $\boldsymbol{v}(e;D,q)$ は，質問文 q と文章 D が与えられたときの解答候補 e の局所文脈を集約したものと解釈することができる[6]．したがって，質問応答は，質問文のベクトル $\boldsymbol{u}(q)$ に最も近い局所文脈ベクトル $\boldsymbol{v}(e;D,q)$ を持つ解答候補 e を探す問題として定式化できる．すなわち，q および D に対して解答候補 e が答えとなる条件付き確率 $p(e|D,q)$ を次式で推定することができる．

$$p(e|D,q) \propto \exp(\boldsymbol{v}(e;D,q)^{\mathrm{T}}\boldsymbol{u}(q)) \tag{16}$$

以上を提案モデルの基本形（Basic）とする．

[6] 実際には，式 (15) 内ではバイアスベクトル $\boldsymbol{b}_v$ の他に，「エンティティが質問文内に既に現れている」場合に足し合わせるヒューリスティック用のベクトル $\boldsymbol{b}_{v'}$ も存在するが，ここでは説明の簡略化のために省略した．

Byway ベクトルによる拡張　上述の基本形モデルでは，各文で局所文脈をまとめ，さらに各解答候補ごとにアテンションメカニズムを適用する．ただし，このままでは，アテンションメカニズムへの誤差伝搬において学習が的確に行われない可能性がある．不正解の解答候補（負例）からアテンションメカニズムに誤差が伝搬するプロセスを考えると，解答候補の推定確率を下げるには，質問文から遠い局所文脈に対する注目度を上げればよいので，何も工夫をしなければ，質問文と似ていない局所文脈に注目を集めるようにアテンションメカニズムが学習されてしまう．アテンションメカニズムは本来質問文と似ている局所文脈に注目を集めるように学習されるべきなので，上のような方向の学習は避けなければならない．この問題は，解答候補のどの出現（mention）にも対応しない空の mention を仮想的に追加し，それに対する仮想的な局所文脈ベクトルを用意することによって解決することができる．ここでは，この仮想的な局所文脈ベクトルを “Byway”（裏道）ベクトルと呼ぶ．“Byway” ベクトルを導入することによって，負例からの誤差伝搬では “Byway” ベクトルに注目が集まるように学習される．しかも，正例の学習の障害にならない．

5.2　局所文脈の動的分散表現の蓄積

3.2 節で導入したアテンションメカニズムで複数の局所文脈を足し合わせられるようになった．しかし，冒頭で述べた (2) の例は局所文脈の足し合わせだけでなく，ときには異なるエンティティ（解答候補）の局所文脈をつなぎ合わせる必要があることを示唆していた．そこで，本稿の 2 つめの提案として，各解答候補の局所文脈を談話の進行に従って動的に計算することによって，局所文脈のつなぎ合わせを実現する方法を考える．

冒頭の例 (2) では，*Jacqueline* の局所文脈 *is the wife of John* に *John* の局所文脈 *is the president* をつなぎ合わせて，*is the wife of John, who is the president* という局所文脈を作ることができれば解答できる．このような局所文脈のつなぎ合わせは，図 3 のように先行文脈の局所文脈ベクトルを現在文の LSTM に入力することによって実現できる．図 3 では，3 文目の局所文脈を計算するときに *John* の入力ベクトルとして，過去 2 回の *John* の局所文脈を用いている．これによっ

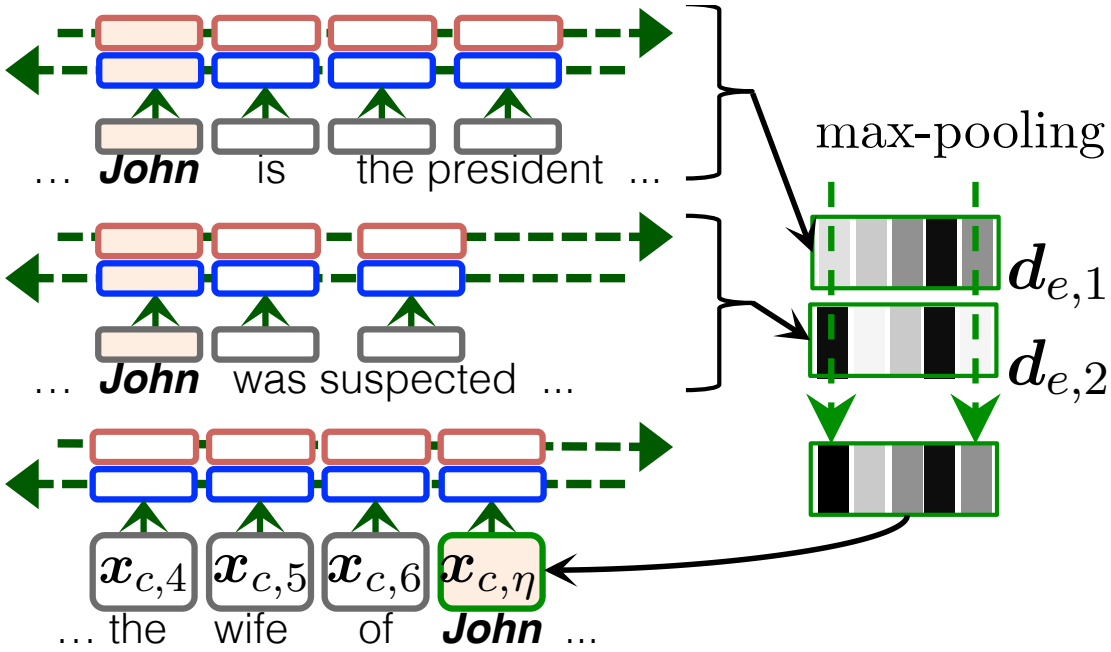


図 6: 複数の文脈情報をマックスプーリングで蓄積し，LSTM への入力 $\boldsymbol{x}_{c,\eta}$ として用いる．

て 3 文目の局所文脈 *the wife of John* と *John* の先行局所文脈をつなぎ合わせることができる．

上の例のように先行局所文脈が複数ある場合は，それらをどのように重ね合わせるかがもう一つの問題となるが，ここではマックスプーリング [24] によってベクトルの重ね合わせを行う．マックスプーリングは各要素の時刻や順序の変化に対する頑健性が高いと言われており，我々の目的に適う道具立てだと期待できる．具体的には，解答候補 e が出現したそれまでの文 c' の局所文脈 $\boldsymbol{d}_{e,c'}$ すべてにわたって，各次元について最大値をとる．以上を総合すると，解答候補 e が後続文 c の位置 η に現れる時，それに対応する LSTM への入力を次式で与える．

$$\boldsymbol{x}_{c,\eta} = W_{dx} \underset{c' \prec c}{\text{max-pooling}}(\boldsymbol{d}_{e,c'}) + \boldsymbol{b}_x$$

6　動的分散表現による言語モデルの拡張

RNN 言語モデル [3] は，文章中の t 番目の単語 w_t を予測する語彙集合に対する確率分布 $\boldsymbol{p}_t$ を次のように計算する．

$$\boldsymbol{p}_t = \mathrm{softmax}(W_{hp}\boldsymbol{h}_t + \boldsymbol{b}_{hp}) \tag{17}$$

$$\boldsymbol{h}_t = \overrightarrow{RNN}_{\mathrm{t}'\prec\mathrm{t}}(\boldsymbol{x}_{w_{t'}}) \tag{18}$$

$$\mathrm{softmax}(\boldsymbol{s})_i = \frac{\exp(s_i)}{\sum_{s_j \in \boldsymbol{s}} \exp(s_j)} \tag{19}$$

$$\tag{20}$$

関数 $\mathrm{softmax}(\boldsymbol{s})_i$ は，N 次元のベクトル $\boldsymbol{s}$ に対して適用される関数で，i 次元目のスカラー値を自然指数関数 exp にかけたものを，各次元のスカラー値を自然指数関数 exp にかけたものの総和で割って正規化した値を求める．

ベクトル $\boldsymbol{p}_t$ のうち単語 w の出力確率は，出力行列 $W_{hp} \in \mathbb{R}^{V\times N}$（語彙数 V，ベクトル次元数 N）のうちの一列 $\boldsymbol{W_{hp}(w)}$ と $b_{hp}(w)$ を用いて以下のように表せる（softmax 関数の正規化項 Z の計算のためには W_{hp} や $\boldsymbol{b}_{hp}$ 全体が用いられる）．

$$p_t(w) = \frac{\boldsymbol{W_{hp}(w)}^{\mathrm{T}}\boldsymbol{h}_t + b_{hp}(w)}{Z} \tag{21}$$

$$Z = \sum_{w'} \boldsymbol{W_{hp}(w')}^{\mathrm{T}}\boldsymbol{h}_t + b_{hp}(w') \tag{22}$$

通常 W_{hp} は学習後の静的なものを用いるが，これを動的分散表現によって拡張する．具体的には，各エンティティに対応する列ベクトルを動的分散表現から生成した動的なベクトル $\boldsymbol{d}'_e$ によって代用する．

$$\boldsymbol{d}'_e = W_{dy}\boldsymbol{d}_e + \boldsymbol{b}_{dy} \tag{23}$$

バイアス値 $b_{hp}(w)$ については，エンティティである場合には共通してスカラー値 b_{hpe} を用いる．

また，各単語トークンに対応するベクトル $\boldsymbol{x}_{w_{t'}}$ もエンティティについては $\boldsymbol{x}_{w_{t'}}$ によって代用する．

$$\boldsymbol{d}''_e = W_{dx}\boldsymbol{d}_e + \boldsymbol{b}_{dx} \tag{24}$$

また，さらなる拡張として，純粋に動的に構築したベクトルのみを使うのではなく，そこに，元のデフォルトのベクトルを足したものを使うモデル（add-default），さらに，重み付けで両者を足し合わせるモデル（add-scale）についても提案する．これらにより，変数としての静的な意味表現も共通して保持することを期待する．例えば，文系列 C を読み込んだ後のエンティティ e の入力単語ベクトル $\boldsymbol{x}_{e(C)}$ については以下のように計算する．

$$\boldsymbol{x}_{e(C)} = \tanh(W_{dx}\boldsymbol{d}_{e,\prec C} + \boldsymbol{b}_{dx}) \tag{25}$$

$$\boldsymbol{x}_{e(C)} = \tanh(W_{dx}\boldsymbol{d}_{e,\prec C} + \boldsymbol{b}_{dx}) + \boldsymbol{w}_e \quad \text{(add-default)} \tag{26}$$

$$\boldsymbol{x}_{e(C)} = \boldsymbol{\alpha} \odot \tanh(W_{dx}\boldsymbol{d}_{e,\prec C} + \boldsymbol{b}_{dx}) + (\mathbf{1} - \boldsymbol{\alpha}) \odot \boldsymbol{w}_e \quad \text{(add-scale)} \tag{27}$$

$\boldsymbol{\alpha}$ は学習パラメータとして新たに追加する．

本研究では，文脈エンコードによる動的分散表現の更新は，文を読み終えるごと（文区切り記号が入力されるごと）に行う．

7　評価実験

7.1　CNN QA の実験設定

提案手法の効果を測るため，CNN QA データセットを用いて性能評価実験を行った．配布されている CNN QA の記事文章データは文境界がない単語列となっている．提案モデルでは文ごとに処理を行うため，句読点を用いた単純なヒューリスティックで文を独自に分割し，文頭と文末にシンボル単語を追加した．また，モデルのハイパーパラメータは開発用データで簡単にチューニングした[7]．学習時は推定確率の交差エントロピー誤差を最小化した．なお，我々の提案モデルはすべて `Chainer`[8][26] によって実装した．

[7] ベクトルの次元数: 300, Dropout 率: 0.3, バッチサイズ: 50, 最適化手法: RMSProp with momentum [25, 15] (momentum: 0.9, decay: 0.95), 学習率: 0.0001 からスタートさせ，データセット 1 周毎に半減, 勾配クリッピングの上限ノルム: 10. 単語ベクトルは [-0.05, 0.05] の一様分布で初期化し, その他の学習を行う行列は平均 0，分散 2/(列数 + 行数) でのガウス分布で初期化した．

[8] http://chainer.org/

モデル	開発	テスト
Basic Proposed Model (Basic)	0.614	0.623
Basic + Max-pooling	0.712	0.707
Basic + Byway	0.691	0.706
Basic + Byway, Max-pooling (Full)	0.708	0.720
Full + w2v-initialization	**0.713**	**0.729**
Deep LSTMs*	0.550	0.570
Attentive Reader*	0.616	0.630
Impatient Reader*	0.618	0.638
Memory Networks**	0.635	0.684
+ Ensemble (11 models)**	0.662	0.694

表 1: CNN QA における正解率．∗ の結果は Hermann ら [2]，∗∗ は Hill ら [6] からの引用である．

7.2 CNN QA の実験結果

表 1 に各モデルの正解率を示す．まず，“Byway” ベクトルの追加によって大きな性能向上が見られる．そして，マックスプーリングによる動的分散表現モデルの場合にもモデルの性能が劇的に向上しており，提案手法の有用性を示している．それら 2 つを組み合わせて使用した場合（Full）にはさらに性能が向上した．さらに，訓練済みの word2vec[9] [27] を用いて単語ベクトルを初期化したところ[10]，CNN QA における既報の最高正解率を上回った．

表 1 の下段には CNN QA に対する既存の state-of-the-art 手法の性能を掲載した．このうち，Attentive Reader と Impatient Reader[2] は，我々同様に双方向 LSTM とアテンションメカニズムを用いているが，これらのモデルは，全ての解答候補の全ての出現から注目すべき出現を選択するモデルと解釈できる．一方，我々の基本モデルは，解答候補ごとに注目すべき出現を選択するモデルになっている点で異なる．アテンションメカニズムは，アテンションの選択範囲が過度に

[9] http://code.google.com/p/word2vec/ 上の GoogleNews-vectors-negative300.bin.gz を用いた．

[10] なお，外部の単語ベクトルを用いても固有表現の単語ベクトルは依然使えないため，タスクの枠を外れて背景知識を使ったことにはならない．

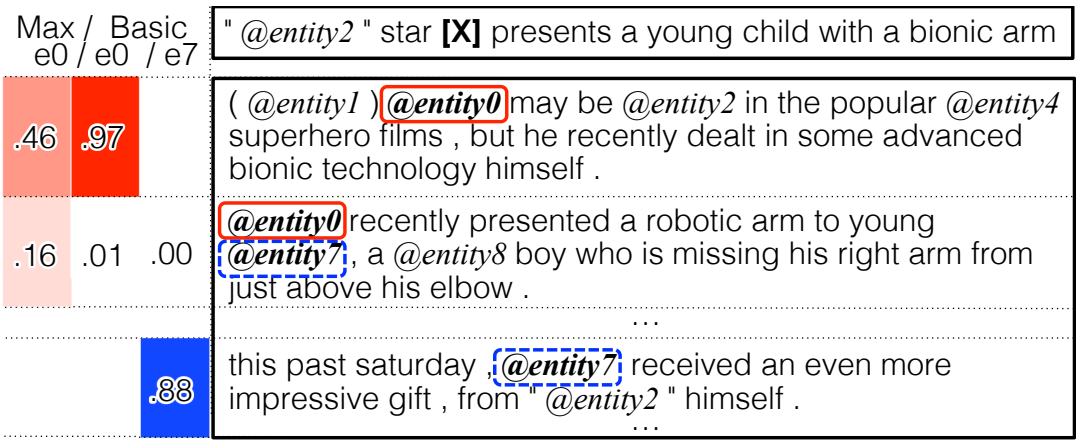


図 7: アテンションメカニズムの挙動の例．各文の注目度重みを文の左に示す．

広いと上手く働かないことが報告されており [16, 17]，この点で我々のモデルの方が有利であると期待できる．また，我々のモデルは解答候補ごとに局所文脈を集約するため，式 (8) のように解答候補の選択を質問文との局所文脈の比較として自然に実現することができる．実際，表 1 に示すように，“Byway” ベクトルを補った我々の基本モデル（Basic+Byway）は，Attentive Reader と Impatient Reader の性能を上回っている．なお，Memory Networks[6] は我々のモデルとは大きく異なっており定性的な比較は難しい．

図 7 に示した問題では正答は *@entity0* であり，1 文目と 2 文目の情報が複合的に含まれたような質問文になっている．Basic モデルでは，この問題に *@entity7* と誤答している．一方で動的分散表現モデルでは 2 文目にも注目度が割り振られ，複合的に情報を用いて正しく *@entity0* と解答できている．

さらに，定量的にも分析を行った．開発用データにおいて，動的分散表現モデルのみで正解した 583 問における正解エンティティの本文中の出現回数の平均値（7.4）は，Basic モデルでも併せて正解した 2064 問における値（6.6）よりも大きかった．これは動的分散表現モデルが，エンティティが多く出現した場合での文脈情報をより適切に統合できるようになったことを示唆している．

7.3 言語モデルの実験設定

3 で述べた方法で作成したデータセットにおいて比較実験を行った．RNN 言語モデル [3] としては，単純な $\boldsymbol{h}_t = \tanh(W_x\boldsymbol{x}_t + W_h\boldsymbol{h}_{t-1} + \boldsymbol{b})$ という RNN を用いるのではなく GRU（Gated Recurrent Unit）[28] を用いた．

$$\boldsymbol{z}_t = \sigma(W_{xz}\boldsymbol{x}_t + W_{hz}\boldsymbol{h}_{t-1} + \boldsymbol{b}_z) \tag{28}$$

$$\boldsymbol{r}_t = \sigma(W_{xr}\boldsymbol{x}_t + W_{hr}\boldsymbol{h}_{t-1} + \boldsymbol{b}_r) \tag{29}$$

$$\boldsymbol{h}'_t = \tanh(W_x\boldsymbol{x}_t + W_h(\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1}) + \boldsymbol{b}) \tag{30}$$

$$\boldsymbol{h}_t = (1 - \boldsymbol{z}_t) \odot \boldsymbol{h}_{t-1} + \boldsymbol{z}_t \odot \boldsymbol{h}'_t \tag{31}$$

$\odot$ は 2 ベクトルの要素積を表す．GRU は LSTM 同様に単純な RNN よりも様々なタスクで高性能を示しやすいことが知られている [29]．また，言語モデル部分の RNN だけでなく，今回は文脈をエンコードする部分についても GRU を同様に用いた．

単語埋め込みベクトルの次元数と言語モデル部分の GRU のユニット数は共に 256，文脈エンコーダ部分の (bi-directional) GRU のユニット数は 128 とし，動的分散表現の次元数は 256 とした．文脈エンコーダ（による動的分散表現）を用いる場合は，加えて行列 $W_{dx} \in \mathbb{R}^{d\times d}, W_{dy} \in \mathbb{R}^{d\times d}, W_{hd} \in \mathbb{R}^{d\times 2d}$ のパラメータもモデルは保持する．モデルパラメータは動的分散表現を使う場合の方が多くなっているため，さらなる比較のためにほぼ同数のパラメータを持つように単語埋め込みベクトルの次元数と言語モデル部分の GRU のユニット数を共に 275 とした場合の単なる言語モデルも比較対象に加えた．動的分散表現の合成にはベクトルの平均を用いた．言語モデルの逆伝播は 2 文ごとに区切って行った．

最適化には，目的関数を予測単語ごとの交差エントロピー誤差とし，ミニバッチサイズを 32 として，RMSProp with momentum [25, 15] (momentum: 0.9, decay: 0.95) を学習率を初期値 0.0005 としてデータセット 1 周分の記事を処理するごとに学習率を半減させた．また，過学習の抑制のために，言語モデルの GRU の入力と，最終出力層への入力に確率 0.2 の Dropout をかけた．また，文脈エンコーダの事前学習として，訓練コーパスを利用して文穴埋め（sentence completion）単語予測をデータセット 10 周分行った場合も検証した．

モデル	パープレキシティ（テスト）
動的分散表現 (add-scale)	89.3
動的分散表現 (add-scale, pretrain 無し)	91.3
動的分散表現 (add-default)	95.94
不使用 (275 次元)	92.2

表 2: 各モデルのパープレキシティ．

7.4 言語モデルの実験結果

学習後の言語モデルの各モデルのパープレキシティを表 2 に示す．パープレキシティは値が小さいほど言語モデルの性能が高いことを表す．動的分散表現を用いた場合のパープレキシティが減少しており，提案手法による性能向上が言語モデルにおいても確認できた．しかし，add-scale の場合のみ性能が向上し，単なる add-default の場合には向上がみられなかった．

8　おわりに

本論文では文章読解に向けた談話内の文脈情報の動的な分散表現生成に取り組んだ．具体的には，動的分散表現構築手法として (1) 文中のターゲットエンティティに対する局所文脈のエンコード手法の提案，(2) マックスプーリングなどによる複数の文脈ベクトルの合成，(3) ニューラルネットワークによるエンコーダへの文脈ベクトルの使用を新たに提案した．また，(4) エンティティごとの文脈ベクトルへのアテンションメカニズムを用いて，CNN QA のような文章読解型の質問応答タスクに対する QA アーキテクチャを提案した．そして，CNN QA による評価実験によって，動的分散表現の有用性を示すとともに State-of-the-art の性能を示した．また，(5) 言語モデルに対しても応用できることを提案し，比較実験によって選択選好性に関する性能向上を QA よりも具体的に示した．

謝辞

終始，研究に関し熱心なご指導ご助言をいただいた指導教員の乾健太郎教授に心より感謝致します．同じく，適切なご指導ご助言をいただいた指導教員の岡崎直観准教授にも心より感謝致します．

また，本論文の審査をお受けしていただきました本学の木下賢吾教授及び伊藤彰則教授に深く感謝致します．

本研究を進めるにあたり，執筆に関しても多くのご助言をいただいた乾・岡崎研究室の田然研究特任助教に深く感謝致します．また，研究会や様々な機会に議論やご助言のお時間をいただいた同研究室の皆様及び株式会社 Preferred Networks の皆様に感謝致します．また，研究室や課外における研究活動に際し，事務処理を始めとして多大なご援助をいただいた八巻智子秘書，成田順子技術補佐員，菅原真由美秘書に感謝致します．

末筆ながら，これまで多大に支えていただきました家族と友人に感謝致します．

参考文献

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR*, 2015. iii, 8, 9, 10, 16

[2] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *NIPS 28*, pp. 1684–1692. 2015. 1, 3, 7, 10, 11, 12, 23

[3] Tomas Mikolov, Martin Karafit, Luks Burget, Jan Cernock, and Sanjeev Khudanpur. Recurrent neural network based language model. In *INTERSPEECH*, pp. 1045–1048. ISCA, 2010. 2, 7, 20, 25

[4] Jiwei Li and Dan Jurafsky. Do multi-sense embeddings improve natural language understanding? In *Proceedings of EMNLP 2015*, pp. 1722–1732, 2015. 3, 8

[5] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *ICLR Workshop*, 2013. 6

[6] Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. The goldilocks principle: Reading children's books with explicit memory representations. *CoRR*, Vol. abs/1511.02301, , 2015. 7, 10, 23, 24

[7] Nathanael Chambers and Dan Jurafsky. Unsupervised learning of narrative schemas and their participants. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pp. 602–610, 2009. 8

[8] Karl Pichotta and Raymond Mooney. Statistical script learning with multi-argument events. In *Proceedings of the 14th Conference of the European*

Chapter of the Association for Computational Linguistics, pp. 220–229, Gothenburg, Sweden, 2014. Association for Computational Linguistics. 8

[9] Michael Roth and Mirella Lapata. Context-aware frame-semantic role labeling. *Transactions of the Association for Computational Linguistics*, Vol. 3, pp. 449–460, 2015. 8

[10] Yangfeng Ji and Jacob Eisenstein. One vector is not enough: Entity-augmented distributed semantics for discourse relations. *Transactions of the Association for Computational Linguistics*, Vol. 3, pp. 329–344, 2015. 8

[11] Regina Barzilay and Mirella Lapata. Modeling local coherence: An entity-based approach. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pp. 141–148, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. 8

[12] David Bean and Ellen Riloff. Unsupervised learning of contextual role knowledge for coreference resolution. In *HLT-NAACL 2004: Main Proceedings*, Boston, Massachusetts, USA, 2004. 8

[13] Artur d'Avila Garcez, Tarek R Besold, Luc de Raedt, Peter Földiak, Pascal Hitzler, Thomas Icard, Kai-Uwe Kühnberger, Luis C Lamb, Risto Miikkulainen, and Daniel L Silver. Neural-symbolic learning and reasoning: Contributions and challenges. In *2015 AAAI Spring Symposium Series, Knowledge Representation and Reasoning: Integrating Symbolic and Neural Approaches*, 2015. 8

[14] Jianpeng Cheng and Dimitri Kartsaklis. Syntax-aware multi-sense word embeddings for deep compositional models of meaning. In *Proceedings of EMNLP*, pp. 1531–1542, 2015. 8

[15] Alex Graves. Generating sequences with recurrent neural networks. *CoRR*, Vol. abs/1308.0850, , 2013. 8, 22, 25

[16] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of EMNLP 2015*, pp. 1412–1421, 2015. 10, 11, 24

[17] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 2048–2057, 2015. 10, 11, 16, 24

[18] Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 379–389, 2015. 10

[19] Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. Reasoning about entailment with neural attention. *CoRR*, Vol. abs/1509.06664, , 2015. 10

[20] Sainbayar Sukhbaatar, arthur szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In *NIPS 28*, pp. 2431–2439. 2015. 10

[21] Ankit Kumar, Ozan Irsoy, Jonathan Su, James Bradbury, Robert English, Brian Pierce, Peter Ondruska, Ishaan Gulrajani, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. *CoRR*, Vol. abs/1506.07285, , 2015. 10

[22] Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Proceedings of the Sixteenth Conference on Computational Natural Language Learning (CoNLL 2012)*, Jeju, Korea, 2012. 13

[23] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, Vol. 9, No. 8, pp. 1735–1780, 1997. 15

[24] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, Vol. 86, No. 11, pp. 2278–2324, 1998. 19

[25] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5 - msprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning.*, 2012. 22, 25

[26] Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. Chainer: a next-generation open source framework for deep learning. In *Proceedings of Workshop on LearningSys in NIPS 28*, 2015. 22

[27] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *NIPS 26*, pp. 3111–3119, 2013. 23

[28] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, 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*, 2014. 25

[29] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. An empirical exploration of recurrent network architectures. In David Blei and Francis Bach, editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pp. 2342–2350. JMLR Workshop and Conference Proceedings, 2015. 25

発表文献一覧

受賞一覧

- 2016 年 3 月 言語処理学会第 22 回年次大会 優秀賞
- 2015 年 10 月 The 22nd ITS World Congress, Best of the Rest

国際会議論文

1. Yuki Igarashi, Hiroya Komatsu, Sosuke Kobayashi, Naoaki Okazaki, Kentaro Inui. Tohoku at SemEval-2016 Task 6: Feature-based Model versus Convolutional Neural Network for Stance Detection. In proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016), Jun. 2016.
2. Sosuke Kobayashi, Ran Tian, Naoaki Okazaki, Kentaro Inui. Dynamic Entity Representation with Max-pooling Improves Machine Reading. In proceedings of the NAACL HLT 2016, Jun. 2016.
3. Naoya Inoue, Yasutaka Kuriya, Sosuke Kobayashi, Kentaro Inui. Recognizing Potential Traffic Risks through Logic-based Deep Scene Understanding. In proceedings of the 22nd ITS World Congress, Oct. 2015.

国内会議・研究会論文

1. 小林颯介, 田然, 岡崎直観, 乾健太郎. 談話内における局所文脈の動的分散表現. 言語処理学会第 22 回年次大会, Mar. 2016. http://www.anlp.jp/proceedings/annual_meeting/2016/pdf_dir/A7-5.pdf
2. 小林颯介, 海野裕也, 福田昌昭. 再帰型ニューラルネットワークを用いた対話破綻検出と言語モデルのマルチタスク学習. 第 75 回 人工知能学会 言語・音声理解と対話処理研究会, B5-02, pp.41-46, Oct. 2015.

3. 小林颯介, 井之上直也, 栗谷康隆, 近藤敏之, 安部克則, 奥野英一, 乾健太郎. 物理モデルと論理推論の統合による運転シーンの潜在的危険の予測. 自動車技術会 2015 年春季大会学術講演会講演予稿集, pp.1076-1081, May 2015.