修士論文

機械読解における Answer Sentence Selection の再考

佐々木 俊樹

2019 年 3 月 5 日

奈良先端科学技術大学院大学
情報科学研究科 情報科学専攻

本論文は奈良先端科学技術大学院大学情報科学研究科に
修士（工学）授与の要件として提出した修士論文である．

佐々木 俊樹

審査委員：

松本 裕治 教授　（主指導教員）
中村 哲 教授　（副指導教員）
新保 仁 准教授　（副指導教員）
進藤 裕之 助教　（副指導教員）

機械読解における Answer Sentence Selection の再考*

佐々木 俊樹

内容梗概

機械読解力 (Machine Reading Comprehension, MRC) は，機械によるテキスト理解を目的とした自然言語処理分野における挑戦的なタスクであり，与えられる説明文 (コンテキスト) とそれに関連する質問文から回答を導く．この推論時に必要となる文を説明文から選択する処理のことを Answer Sentence Selection (ASS) と呼ぶ．ASS はシステムの回答精度を向上させる手法の一つであり，ASS にて選択された文はシステムの意思決定の要因と考えることができる．ASS で示されるシステムの判断根拠は，ミスの許されない医療システムなどを考える時にとても重要な情報の一つである．しかしながら，ASS を評価するタスクやデータセットがほとんどないことや，これまでの機械読解の研究では回答精度のみが重要と考えられていたことから ASS は軽視されてきた．そこで，我々は既存の MRC のデータセットに対して各質問の回答に必要な文をアノテーションすることで，ASS に取り組むためのデータセットを作成し，どのような能力・情報が ASS に必要かを調べる．実験の結果，ASS は挑戦的なタスクではあるが MRC において重要な処理であることがわかった．

キーワード

機械読解，Answer Sentence Selection, 判断根拠

*奈良先端科学技術大学院大学 情報科学研究科 情報科学専攻 修士論文,
2019 年 3 月 5 日.

Reconsideration of Answer Sentence Selection in Machine Reading*

Toshiki Sasaki

Abstract

Machine reading is a challenging task in the field of natural language processing aimed at understanding texts by machines, and leads answers from given explanatory texts (context) and related question sentences. The process of selecting sentences necessary for this reasoning from explanation is called Answer Sentence Selection (ASS). ASS is one of methods to improve the accuracy of response of the system, and the sentence selected by ASS can be thought of as the factor of decision making of the system. The judgment rationale of the system indicated by ASS is one of very important information when thinking about a medical system etc. which mistake is not allowed. However, ASS has been neglected because there are almost no tasks and data sets to evaluate ASS and only answer accuracy is considered important in research on machine reading. Therefore, we created a data set that required ASS that annotated the sentences necessary for answering each question to the existing machine reading comprehension data set, and examined what kind of ability / information is necessary for ASS . As a result of the experiment, ASS is a challenging task, but it turned out to be an important process in MRC.

Keywords:

Machine Reading, Answer Sentence Selection, evidence of judgement

*Master's Thesis, Department of Information Science, Graduate School of Information Science, Nara Institute of Science and Technology, March 5, 2019.

目次

図目次

第 1 章　序論

1.1　背景

システムに人間と同等の言語理解力を与えることは，自然言語処理や人工知能における中心的な課題である．機械の読解力 (Machine Reading Comprehension, MRC) テストや質問応答 (Question Answering, QA) はその問題に着目した主要なタスクであり，昨今急速な発展を遂げている．例えば，MRC の大規模なデータセットである SQuAD 1.1 [1] では人間のパフォーマンスを超えたシステムが提案されている．現在の MRC の問題点の一つに，システムが回答を導いた判断根拠を示していないことが挙げられる．システムがテキストを理解しているかを判断するためにはシステムの回答に至った要因を示す必要がある．しかしながら現在の多くのシステムは，回答に必要な情報が記述されたテキスト部分を参照して回答するなどのように判断根拠を示すものはない．

MRC における根拠の提示に着目した研究の一つに，推論時に必要となる文を与えられた説明文 (コンテキスト) や多数の文の候補から選択する Answer Sentence Selection (ASS) というタスクが提案されている [2,3]．ASS は MRC や QA の回答精度を向上させるために，回答を 1 つに選択する一段階前の処理としてしばしば用いられている．ASS にて選択された文はシステムの意思決定の要因と考えることができ，ASS で示される根拠はミスの許されない医療システムなどを考える際にはとても重要な情報の一つであると考えられる．しかしながら ASS を評価するタスクやデータセットが少ないことや，これまでの読解問題の研究では回答精度のみが重要視され ASS は軽視されてきた．

そこで，本研究では既存の MRC データセットに対して，質問に回答するために必要となる文をアノテーションすることで ASS をタスクとした新たなデータセットを作成した．この ASS タスク実際にに取り組むことにより，良い ASS システムの開発にはどのような能力・情報が必要であるかを調査し，MRC における ASS の重要性を検討する．

1.2　本論文の構成

本論文の構成は以下の通りである．第 2 章では，読解問題とそのデータセットの分析における関連研究について記述する．第 3 章では，提案手法ついて概説し，本論文で取り組む ASS タスクについて記述する．第 4 章では，ASS タスクでの実験について具体的に記述する．第 5 章では，第 4 章での実験で得られた結果と考察について記述する．第 6 章では，本研究で得た結論について概説する．

第 2 章　関連研究

本章では機械読解に関する既存研究について述べる．

2.1　機械読解力

機械読解力 (Machine Reading Comprehension, MRC) 問題は，機械がコンテキストと呼ばれる説明文とそれに関連する質問文から，その質問の答えを導くタスクのことであり，機械がテキストをどの程度理解しているかを測定するために提案された．機械読解力システムの構築にはパージング (Parsing) や共参照 (Co-Refernece)，推論 (Reasoning)，一般常識 (Common Knowledge) といった複数の自然言語の処理や能力を必要とするため，非常に難しいタスクとなっている．機械読解力のデータセットはコンテキストの種類や回答方式，データセットの作成方法などで異なっており，多種多様なデータセットが存在する．提案されているデータセットを回答方式により大きく分けると，与えれるコンテキスト内の単語またはフレーズを抽出することで質問に回答する抽出式問題と，質問文と共に与えられる複数の回答候補から回答を選択する選択肢式問題の 2 つがある．これまでに提案された MRC データセットのいくつかを 表 2.1 に示す．

表 2.1: MRC データセットの例

データセット	ジャンル	データセット 作成方法	回答方式	ドキュメント数	質問数
SQuAD 1.1	Wikipedia	クラウドソース	単語のスパン	536	100k
NewsQA	ニュース記事	クラウドソース	単語のスパン	10k	100k
RACE	一般	試験	複数選択肢	28k	97k
MCTest	物語	クラウドワーカー	複数選択肢	660	2640

図 2.1 には MRC 問題の一例を示している．2.1(a) は選択肢式問題である MCTest の問題の例を，2.1(b) は抽出式問題である SQuAD の問題の例を示している．実際の正しい答えを赤文字，2.1(a) ではさらに，回答に関連するまたは必要となると考えられる文を青文字で示している．質問文 *"what does molly's cat play*

with?" に対する正しい答えは *"kitty plays with yarn."* の文から分かる通り *"yarn (糸)"* である．このように，質問に回答するために質問文とコンテキストの関係性を明確にすることが *MRC* 問題を解く上で重要だと考えられている．今回の例では，*"kitty plays with yarn."* の文が最も回答に必要な文だと考えられるが，その一文だけでは回答することは不可能である．実際には *kitty* が *molly* の飼っている猫の名前であることや， 文中の *her* が *molly* を示していることも考慮しなければ正しい回答は得られない．すなわち今回の例では，*"molly likes animals."*, *"her cat's name is kitty."*, *"kitty plays with yarn."* の *3* つの文により回答が導けると考えられる．これまでの *MRC* 問題では回答精度のみが重要視されており，このような推論の過程やシステムの判断根拠は軽視される傾向があった．また，定量的にシステムの判断根拠を分析および評価した研究は少ない．

Paragraph: *Molly likes animals. She has a cat. She has a dog. She has a bird. She has a hamster. She has a bunny. Her cat's name is kitty. Her dog' s name is Spike. Her bird's name is Polly. Her hamster's name is Barry. Her bunny's name is Snowball. Kitty plays with yarn. Spike plays with a ball. Polly plays in her cage. Barry runs on his wheel. Snowball eats carrots.*

Question: *What does molly's cat play with?*
Candidates: *'yarn', 'cage', 'wheel', 'ball'*
Plausible Answer: *yarn*

(a) MCTest の問題の例

Paragraph: *The English name "Normans" comes from the French words Normans/Normanz, plural of Normant, modern French normand, which is itself borrowed from Old Low Franconian Nortmann "Northman" or directly from Old Norse Norðmaðr, Latinized variously as Nortmannus, Normannus, or Nordmannus (recorded in Medieval Latin, 9th century) to mean "Norseman, Viking".*

Question: *When was the Latin version of the word Norman first recorded?*
Plausible Answer: *9th century*

(b) SQuAD の問題の例

図 2.1: MRC 問題の例

2.2 Answer Sentence Selection

システムがどのように回答を選択したかを考えるタスクの一つとして，与えられた質問へ回答するために必要となる文をコンテキストやドキュメントから選択し，関連する順にランク付けする回答文選択 *(Answer Sentence Selection, ASS)* というタスクが提案されている．*Wang* らは *Text REtrieval Conference (TREC) QA track (8-13)* のデータセットから *ASS* のデータセットを作成した [2]．QASent と

呼ばれるこのデータセットは ASS のベンチマークとして使われているが，プログラムによって自動で作成された問題が多く，人手により検証された質の高い QA ペアの数は非常に少ないという欠点がある．また，Yang らは WikiQA という Bing 検索エンジンのユーザログから選択した質問から構成される QASent よりもサイズの大きな ASS データセットを作成した [4]．これまでの ASS の多くはその文が必要かどうかを識別するものではなく，その文を質問文と関係性が強い順番にランク付けするものである．しかしながら，システムが回答や推論を行うために重要であるのは，文をランク付けすることではなくその文が回答に必要であるかを明確にすることである．そこで本論文では，質問に回答するために必要な文であるかどうかを明確に識別する必要があるタスクとして ASS に取り組む．

2.3 データセットの評価と分析

MRC 問題はシステムがどの程度テキストを理解できているかを測定するためのタスクであるが，その評価が正しく行われているかは疑問が残る．例えば，MRC データセットの一つである SQuAD [1] では，人間のパフォーマンスを超えたシステムが提案されている．しかしながら，システムの回答精度が人間に優っているとしても人間に勝るテキスト理解能力をシステムが持っているとは考えづらい．そのような背景から，MRC データセットに対する評価や分析を行った研究がいくつかある．菅原らは複数の MRC データセットを，「回答に必要な能力」と「読みやすさ」の 2 つの指標を用いて評価し，MRC データセットの読みやすさは回答の難しさに直接影響はなく SQuAD のような読みにくいデータセットでも回答は容易であることを示している [5]．また，Kaushik らはいくつかの MRC データセットに対して，質問のみもしくはコンテキストのみをシステムの学習に用いたとしても高い回答精度が得られることを示し，既存の MRC データセットの課題を明らかにすることで今後のデータセットの作成における注意を促した [6]．本論文では ASS の難しさと質問へ回答するために必要となる能力にどのような関係性があるかを明確にし，MRC における ASS の重要性を検証する．

2.4 コンピュータビジョン分野での Attention の解析

コンピュータビジョンの分野では人間が人手で画像にアノテーションした Attentiion を用いて，モデルを評価・分析する研究がなされている．例えば，Liu らは画像キャプション (Image Captioning) 生成のタスクにおいて，Attention が正しくかかっているかを定量的に評価するために各画像に対して人手により疑似的な正解となる Attention を割り当て，提案したモデルの Attention と比較し，さらに回答精度に影響するかを調べた [7]．実験の結果，正しく Attention が割り当てられた場合，そうでない場合に比べてキャプション生成の精度が高くなることを示した．また Das らは画像を用いた質問応答 (Visual Question Answering, VQA) のタスクにおいて， VQA における State-of-the-art のモデルの Attention が人間があらかじめ人手でアノテーションした Attention と同じように画像にかかっているのかを調べ，人間とは異なった Attention を獲得していることを示した [8]．本論文では，コンテキスト中の回答に必要となる文へのスコアを Attention と捉えることにより，モデルが人手でアノテーションした文と同じような Attentilon を獲得できているのかを確認する．

第 3 章　提案手法

これまでの MRC テストでは回答精度のみが重要視されてきたが，医療システムのようなミスの許されないシステムの開発を考える場合にシステムの判断根拠を示すことは非常に重要である．

そこで，我々は回答に必要な文をコンテキスト内から選択する ASS に取り組むために，既存の MRC データセット MCTest に対して回答に必要な文をアノテーションすることで新たなデータセットを作成した．図 3.1 は ASS の簡単な例を示している．Original で示している部分は MCTest に元々アノテーションされている情報であり，質問文，質問のタイプ，回答候補がある．質問のタイプとは，その質問に回答するため複数の文を必要とする場合は Multiple，一つの文から回答が可能な場合は Single がアノテーションされている．図 3.1 の Question1 は Multiple の問題，Question2 は Single の問題を示している．ASS に取り組むために我々が追加でアノテーションした情報が Annotated に記述されている ASS-Answer である．これは，Question-Type のより詳しい情報として，質問文に回答するためにはどの文が必要となるかを示すものである．例えば， Question1: "what was the heardest thing for tom to fix?" に対する Original の回答は window であるが， ASS の回答は window を求めるために必要な文のインデックスであり赤文字で示される 「0, 5, 6」 の系列となる．同様に，Question2: "who was tom's best friend?" の ASS の回答は青文字で示すように jim が best friend であることを示している文のインデックスとなり「4」となる．より具体的には，コンテキストが 5 文で構成され 0 番目 と 1 番目の文が必要となる場合は，[1, 1, 0, 0 ,0] のように回答に必要となる文のインデックスが 1，それ以外が 0 で構成される系列が回答となる．システムの予測した系列と正しい回答の系列の間の完全一致 (Exact Match, EM) スコアと F1 (micro) スコアを用いてシステムを評価する．以下でこの ASS のデータセットの作成について記述する．

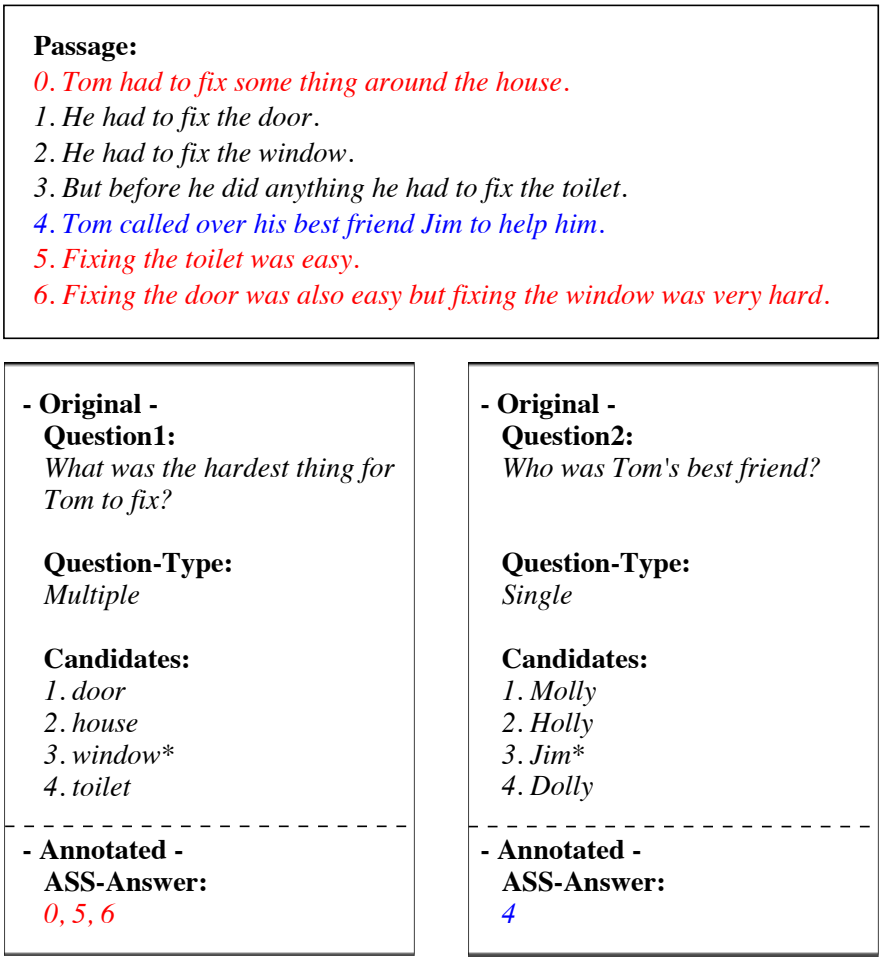


図 3.1: ASS の例

3.1 データセット

MRC タスクのデータセットは大きく分けると抽出式と選択肢式の 2 種類が存在する．抽出式問題は答えのスパンをコンテキストから抽出する問題であり，SQuAD [1] や WikiQA [4], TriviaQA [9] などが該当する．また，選択肢式問題は回答候補が複数個与えられ，その中から正しい答えを選択する問題であり，MCTest [10] や RACE [11] などがある．本論文では代表的な選択肢式問題の一つである MCTest に取り組む．MCTest は小学生を読者として想定した物語から構成されるデータセットである．そのため使用される単語が非常に簡単であると

いう特徴を持っている．各物語に対して 4 つの質問があり，各質問に対して 4 つの回答候補が与えられる．MCTest は物語の数の違いにより MC160 と MC500 の 2 種類があり，MC160 は訓練データ/開発データ/テストデータの数がそれぞれ 280/120/240，MC500 は 1200/200/600 と分配されている．また MC160 は人手により問題の検証がなされているため，問題の曖昧性や回答不可な問題が少ないという特徴がある．MC160 のコンテキスト，質問文，回答の平均単語数はそれぞれ 204, 8.0, 3.4，MC500 ではそれぞれ 212, 7.7, 3.4 である．MCTest はデータサイズが小さいという欠点があるものの，推論を必要とする問題の比率が高いことが示されており，今回の我々の ASS の実験に適していると考えられるため，MC160 をアノテーションを実施する対象に選択し実験を行った．

3.2 アノテーション

次にアノテーションの処理について説明する．まずはじめに，回答を導くために必要な文の定義を明確にする．

回答を導くために必要な文を

(1) 回答を導くために必要であると考えられる文であり，
(2) 質問で問われている主語が明確になっており，
(3) 与えられる複数の回答候補の中から正しい回答 1 つに限定することができる

という条件を満たしているコンテキスト内の文または文の集合であると定義する．

例として，図 3.1 の Question1: "what was the hardest thing for tom to fix?" に対して回答に必要となる文をアノテーションする場合を考える．この質問に回答するために必要な文は，図中での赤文字で示される 0, 5, 6 番目の文である．まずはじめに，0 番目の文は fix する主語が tom を指しており質問の主語を明確にするために必要な文 (2) である．さらに house は fix する対象でないことを示しており，これは与えられる複数の回答候補の中から正しい回答 1 つに限定するために必要な文 (3) であると考えられる．また 5, 6 番目の文も同様に回答を一つに絞るために必要となる文 (3) である．6 番目の文では window を fix することが very hard と記されているが，質問で問われているように hardest であるかどうかはこの文だけ

表 3.1: MCTest の質問タイプの統計値.

	訓練	開発	テスト
オリジナル			
Single	132	53	112
Multiple	148	67	128
アノテーション後			
Single	86	32	66
Multiple	194	88	174 (2)*

では判断ができない．そこで他の候補である toilet と door が window より fix することが easy であることを示している 5, 6 番目の文が必要となる．アノテーションは MCTest に既にアノテーションされている質問タイプ (Single, Multiple) を参考にして行った．

オリジナルの MCTest のデータセットと新たにアノテーションしたデータセットの質問タイプの統計情報を表 3.1 に示す．表が示す通り，アノテーションを行なった結果，訓練・開発・テストデータの全てにおいて質問タイプが Multiple の問題の数がオリジナルでの問題の数よりも増加した．これは，質問タイプに対するアノテーションに曖昧性があることが考えられ，特に共参照が考慮されなかったために本来は Multiple に該当する問題が Single に分類されたケースが多くあることがわかった．

*括弧内の数字はコンテキスト全文を必要とする問題の数を示す.

第 4 章　実験

本章では，作成したデータセットを用いて行った ASS の実験について述べる．大きく分けて 2 つの実験を行なった．1 つめは現在の MRC システムがどの程度 ASS タスクに対応できるのかを確認するために，いくつかのベースラインモデルで我々が人手でアノテーションして作成した ASS のデータセットを用いて ASS に取り組む実験を行なった．また，MCTest のデータセット数が少ないという欠点に対応するため，データ拡張戦略を適用しその効果を確認した．2 つめは ASS の重要性を検討するために，回答に必要である文のみを用いた場合に選択肢式問題の回答精度にどのように影響があるかを確認する実験を行なった．

4.1　前処理

実験について説明する前にデータセットに対する前処理について述べる．まずはじめに，Stanford Core NLP [12] を使用して，コンテキストを単語および文に分割し，質問文と回答候補を単語単位に分割した．また，単語の表記揺れに対応するために，単語単位に分割した後のコンテキスト・質問文・回答候補に対して python を用いて全ての単語を小文字化した．

4.2　Answer Sentence Selection

我々は 3 つのベースラインモデルを用いて実験を行った．一つ目は単語の表層のみを用いた Text-Matching モデル，二つ目はコンテキストと質問文から双方向の Attention を得てコンテキストの表現を獲得する BiDAF モデル，最後が単純なニューラルネット構造を用いた End-to-End の BiLSTM モデルである．これらのモデルについて以下でそれぞれ説明する．

4.2.1 Text-Matching モデル

Text-Matching モデルは他のモデルがニューラルネットを用いるのとは異なり，単語の表層のみから ASS を行う単純なモデルである．図 4.1 に Text-Matching のアルゴリズムを示す．Text-Matching モデルでは，各文に対して回答に必要であるかどうかのスコアを計算しその上位 N 個を最終的な回答とする *topN* モデルと，閾値を設定して閾値以上のスコアの文を回答とする *threshold* モデルの *2* つで実験を行う．

Algorithm 1 Text-Matching Algorithms

Require: コンテキスト内の i 番目の文を S_i，質問文の単語の集合を Q，回答候補の単語の集合を $A_{1...4}$，ストップワードの集合を U とする．

$SA = (Q \cap A_{1...4}) \setminus U$
for $j = 1$ to $|S|$ **do**
    $sc_j = \sum_i \mathbb{I}(SA_i \in S_j)$
end for
return $softmax(sc)$

Algorithm topN

$sc_N \leftarrow N$ 番目に大きいスコア
for $i = 1$ to $|S|$ **do**

$$sc_i = \begin{cases} 1 & if sc_i \geq sc_N \\ 0 & otherwise \end{cases}$$

end for
return sc

Algorithm threshold

$T \leftarrow$ 閾値
for $i = 1$ to $|S|$ **do**

$$sc_i = \begin{cases} 1 & if sc_i \geq T \\ 0 & otherwise \end{cases}$$

end for
return sc

図 4.1: ベースラインモデルとして用いる 2 つの Text-Matching モデル

4.2.2 BiDAF モデル

BiDAF は 代表的な *MRC* データセットである *SQuAD* [1] と穴埋め問題のデータセットの CNN/Daily Mail [13] に対して提案されたモデルであり， それぞれのデータセットで State-of-the-art を達成したモデルである [14]．我々はこの BiDAF モデルを基に ASS タスクに適応したモデルとして BiDAF-ASS を提案する．図 4.2 に BiDAF-ASS モデルの外観を示す．BiDAF は階層的に処理を行うモデルであり，以下の 6 層から構成される：

(1) **Character Embedding Layer.** 文字レベル畳み込みニューラルネット (Convolutional Neural Network, CNN) を用いて各単語をベクトル表現に写像する．
(2) **Word Embedding Layer.** 事前学習された単語エンベディングを用いて各単語をベクトル表現に写像する．
(3) **Contextual Embedding Layer.** LSTM を用いて文脈を意識した単語のエンベディング表現を獲得する．これら 3 つのネットワーク層はコンテキストと質問文の両方に適用する．
(4) **Attention Flow Layer.** コンテキストと質問文を用いて，コンテキスト内の単語に対して質問を意識した特徴ベクトルを生成する．
(5) **Modeling Layer.** コンテキストを走査するために LSTM を適用する．
(6) **Output Layer.** 質問文から答えを予測する．

本論文で提案するモデルでは, Character Embedding Layer は用いないが, BiDAF モデルを ASS タスクに適用するために Modeling Layer と Output Layer の間に新たに Sentence Embedding Layer を追加し，スパンを予測するのではなく ASS を予測するように Output Layer を修正する．それぞれのネットワーク層について以下で詳しい説明を述べる．

(1) **Word Embedding Layer.** Word Embedding Layer は各単語を高次元のベクトル空間へ写像する．入力されるコンテキストと質問文の単語をそれぞれ $\{x_1, ..., x_T\}$，$\{q_1, ..., q_J\}$ とする．我々は事前学習した単語ベクトルである GloVe [15] を各単語の Embedding を獲得するために用いる．Word

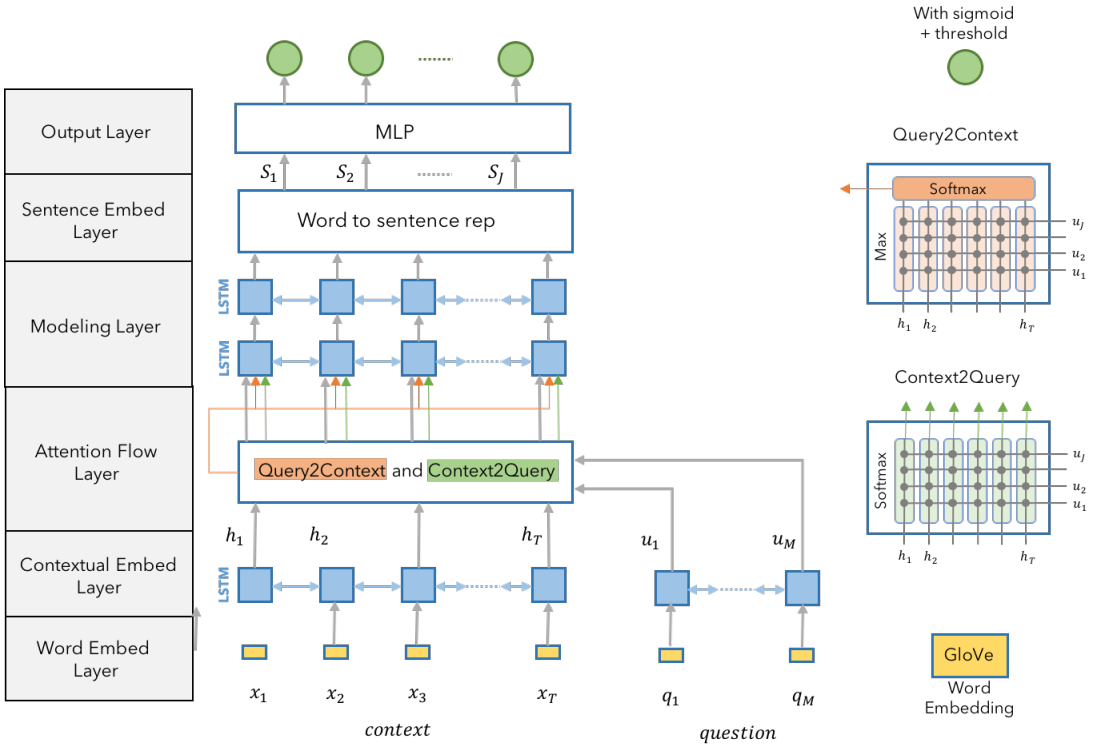


図 4.2: BiDAF モデルの概要

Embedding Layer を通して d 次元の出力を得る．具体的には，コンテキストに対して $\boldsymbol{X} \in \mathbb{R}^{d \times T}$ と質問文に対して $\boldsymbol{Q} \in \mathbb{R}^{d \times J}$ の 2 つの行列を得る．

(2) **Contextual Embedding Layer.** 単語間の文脈をモデル化するために Word Embedding Layer で獲得した表現に対して LSTM を適用する．双方向の LSTM を用いて，その 2 つの出力を結合することで文脈を考慮した単語の表現を獲得する．すなわち，コンテキストの単語ベクトル $\boldsymbol{X}$ から $\boldsymbol{H} \in \mathbb{R}^{2d \times T}$，質問文の単語 $\boldsymbol{Q}$ から $\boldsymbol{U} \in \mathbb{R}^{2d \times J}$ を獲得する．順方向の LSTM と逆方向の LSTM のそれぞれの出力を結合しているので，$\boldsymbol{H}$ と $\boldsymbol{U}$ のそれぞれの行ベクトルは 2d 次元となっている．

(3) **Attention Flow Layer**. Attention Flow Layer はコンテキストと質問文の単語を結びつけ，それらの関係をモデル化する．この Attention Flow Layer はこれまでの Attention Mechanism とは異なり，コンテキストと質問文を一つの特徴ベクトルに要約するのではなく，各タイムステップの Attention ベク

トルは一つ前の層からの Embedding とともに後続の Modelling Layer へ入力することによりシンプルな要約による情報の欠損を防いでいる．

コンテキストの表現 $\boldsymbol{H}$ と質問文の表現 $\boldsymbol{U}$ が入力され，質問を意識したコンテキストのベクトル表現 $\boldsymbol{G}$ が Contextual Embedding とともに出力される．

この層では， Attention をコンテキストから質問文へと質問文からコンテキストへの 2 つの方向から計算する．これらの Attention はコンテキストの Contextual Embedding $\boldsymbol{H}$ と質問文の Contextual Embedding $\boldsymbol{U}$ の間の類似度行列 $\boldsymbol{S} \in \mathbb{R}^{T \times J}$ から導かれる．ここで，S_{tj} は t 番目のコンテキストの単語と j 番目の質問文の単語の類似度を示しており，以下のように計算される，

$$\boldsymbol{S}_{tj} = \alpha(\boldsymbol{H}_{:t}, \boldsymbol{U}_{:j}) \in \mathbb{R} \tag{4.2.1}$$

α は学習可能な類似度を求める関数であり，$\boldsymbol{H}_{:t}$ は $\boldsymbol{H}$ の t 番目の列ベクトル，$\boldsymbol{U}_{:j}$ は $\boldsymbol{H}$ の j 番目の列ベクトルを示している．また $\alpha(\boldsymbol{h}, \boldsymbol{u}) = \boldsymbol{w}^{\mathrm{T}}_{((S))}[\boldsymbol{h}; \boldsymbol{u}; \boldsymbol{h} \circ \boldsymbol{u}]$ であり，$\boldsymbol{w}_{(S)} \in \mathbb{R}^{6d}$ は学習可能な重みベクトル，$\circ$ は要素積, [;] はベクトルの行方向での結合を示している．次に，$\boldsymbol{S}$ を用いて Attention と双方向のベクトルを獲得する．

Context-to-query Attention. Conext-to-Query (C2Q) Attention はどの質問の単語が最もコンテキスト内の単語に関連しているかを示すものである．$\boldsymbol{a}_t \in \mathbb{R}^J$ をコンテキストの t 番目の単語の Attention の重みとし，全ての t に対して $\sum \boldsymbol{a}_{tj} = 1$ とする．attention の重みは $\boldsymbol{a}_t = softmax(\boldsymbol{S}_{t:}) \in \mathbb{R}^J$ により計算する．これにより，質問文の単語ベクトルは $\tilde{\boldsymbol{U}}_{:t} = \sum_j \boldsymbol{a}_{tj} \boldsymbol{U}_{:j}$ となり，$\tilde{\boldsymbol{U}}$ は質問ベクトルを考慮した $2d \times T$ のコンテキスト全体を表す行列となる．

Query-to-context Attention. Query-to-context (Q2C) Attention はどのコンテキストの単語が最も質問文の単語に類似しているかを示すものであり，それゆえ質問に回答するのに重要な attention である．我々はコンテキストの単語の Attention の重みを $\boldsymbol{b} = softmax(max(_{col}(\boldsymbol{S}))) \in \mathbb{R}^T$ により獲得する． max_{col} は各列に対して最大値を求める関数である．これによって得るコンテキストのベクトル表現は $\tilde{\boldsymbol{h}} = \sum_t \boldsymbol{b}_t \boldsymbol{H}_{:t} \in \mathbb{R}^2 d$ で示される．$\tilde{\boldsymbol{h}}$ は列方向に T 倍され，$\tilde{\boldsymbol{H}} \in \mathbb{R}^{2d \times T}$ を獲得する．

最後に Contextual Embedding と Attention ベクトルは G を得るために融

合される．また，この G の各列ベクトルは質問文を意識したコンテキストの単語の表現として考えられる．G は以下で定義される．

$$\boldsymbol{G}_{:t} = \beta(\boldsymbol{H}_{:t}, \tilde{\boldsymbol{U}}_{:t}, \tilde{\boldsymbol{H}}_{:t}) \in \mathbb{R}^{d_G} \tag{4.2.2}$$

$G_{:t}$ は t 番目の列ベクトルを示しており，t 番目のコンテキストの単語に対応するものである．β は 3 つの入力を結合する学習可能な関数であり，d_G は β 関数の出力次元数を示している．β 関数は任意の学習可能なニューラルネットワークであるが，以下の単純なベクトルの結合を本実験では用いる: $\beta(\boldsymbol{h}, \tilde{\boldsymbol{u}}, \tilde{\boldsymbol{h}}) = [\boldsymbol{h}; \tilde{\boldsymbol{u}}; \boldsymbol{h} \circ \tilde{\boldsymbol{u}}; \boldsymbol{h} \circ \tilde{\boldsymbol{h}}] \in \mathbb{R}^{8d \times T} (i.e., d_G = 8d)$.

(4) **Modeling Layer.** Modelling Layer は質問文を意識したコンテキストの単語の表現 $\boldsymbol{G}$ を入力として受け取り，コンテキストの単語間の相互作用を捉えて出力する役割がある．我々は出力の次元数が d の双方向 LSTM を 2 層用いる．これらによって，行列 $\boldsymbol{M} \in \mathbb{R}^{2d \times T}$ を獲得する．$\boldsymbol{M}$ の各列ベクトルはコンテキストと質問文全体のそれぞれの単語についての文脈情報を含んだベクトルである．この層の最終的な出力は Attention Flow Layer の出力 $\boldsymbol{G}$ を用いて，$\tilde{\boldsymbol{M}} = \boldsymbol{w}^{\mathrm{T}}[\boldsymbol{G}; \boldsymbol{M}]$ で表す．$\boldsymbol{w}^{\mathrm{T}} \in \mathbb{R}^{10d}$ は学習可能な重みベクトルである．

(5) **Sentence Embedding Layer.** Sentence Embedding Layer は ASS に対応するためにコンテキスト中の単語の表現を用いて文のベクトル表現を獲得する層である．我々は文の単語の平均を求めることで文の表現として以下の式で表す．

$$\boldsymbol{SS}_k = \frac{1}{|S_k|} \sum_m S_{k_m} \tag{4.2.3}$$

ここでの S_k はコンテキスト中の k 番目の文，S_{k_m} は各文中の m 番目の単語，SS_k は S_k に対するスコアを表している．各 S_{k_m} は対応する G の要素から取得する．

(6) **Output Layer.** Output Layer では，2 層の多層パーセプトロン (Multi Layer Perceptron, MLP) を用いて回答を予測する．$\tilde{\boldsymbol{M}}$ を入力として以下の式で求める．

$$\boldsymbol{p} = MLP2(MLP1(\tilde{\boldsymbol{M}})) \tag{4.2.4}$$

MLP1，MLP2 の重みベクトルの次元数はそれぞれ 10d, d であり，$\boldsymbol{p} \in \mathbb{R}^{|P|}$ である．|P| はコンテキスト中の文の数を示している．

学習． 我々はロス関数として以下の Joint Binary Cross Entropy を最小化することでモデルを学習する．

$$L = -\sum_{j}^{C}\Big(y_j log p_j + (1-y_j) log(1-p_j)\Big) + \lambda\|\theta\|^2, \tag{4.2.5}$$

y_j はコンテキスト中の j 番目の文が回答に必要であるかを示す binary の正解ラベル，p_j はコンテキスト中の j 番目の文が必要であるかを示す binary のシステムの予測，λ は L2 正則化項の重み，θ は全てのパラメータの集合を示している．

4.2.3 End-to-End BiLSTM モデル

End-to-End BiLSTM モデルは BiLSTM を用いた単純なニューラルモデルである．End-to-End BiLSTM モデルの外観を図 4.3 に示す．BiLSTM モデルは BiDAF と共通のネットワーク層を持つが，質問文の表現の獲得方法と Attention Layer に違いがある．BiLSTM モデル特有の処理について以下で述べる．

(1) **質問文の獲得方法．** まず BiLSTM モデルでは質問文の表現を獲得する．Contextual Embedding Layer が BiLSTM を通して出力した質問文の各単語ベクトル $\boldsymbol{U}$ は，具体的には，

$$\overrightarrow{\boldsymbol{u}}_i = LSTM(\overrightarrow{\boldsymbol{u}}_{i-1}, \boldsymbol{q}_i), i = 1, ..., J$$
$$\overleftarrow{\boldsymbol{u}}_i = LSTM(\overrightarrow{\boldsymbol{u}}_{i+1}, \boldsymbol{q}_i), i = J, ..., 1$$

で表される．順方向 LSTM の最後のタイムステップの隠れ状態と逆方向 LSTM の最後のタイムステップの隠れ状態を結合することで，質問文のベクトル表現 $\boldsymbol{Q} = concat(\overrightarrow{\boldsymbol{u}_J}, \overleftarrow{\boldsymbol{u}}_1)$ を獲得する．ここで LSTM の隠れ状態を d とすると，質問文のベクトル表現は $\boldsymbol{Q} \in \mathbb{R}^{2d}$ である．

(2) **Attention Layer．** Attention Layer はコンテキストの単語ベクトル $\boldsymbol{H}$ と質問文ベクトル $\boldsymbol{Q}$ の間の類似度を学習可能な双線形関数によって求め，コ

ンテキストと質問文の相互関係をモデル化する．

$$\alpha_i = softmax_i(\boldsymbol{q}^T W_s \boldsymbol{h}_i) \tag{4.2.6}$$

$$\tilde{\boldsymbol{h}}_i = \alpha_i \boldsymbol{h}_i \tag{4.2.7}$$

ここで，$\boldsymbol{W}_s \in \mathbb{R}^{d \times d}$ は質問文ベクトル $\boldsymbol{Q}$ とコンテキストの単語ベクトル $\boldsymbol{h}_i$ の類似度を求める双線形関数であり，単純な内積よりも柔軟に学習できる利点がある．

Attention Layer で得られたコンテキストの単語ベクトル $\tilde{\boldsymbol{H}}$ は BiDAF モデルと同様に Sentence Embedding Layer に入力され回答を導く．

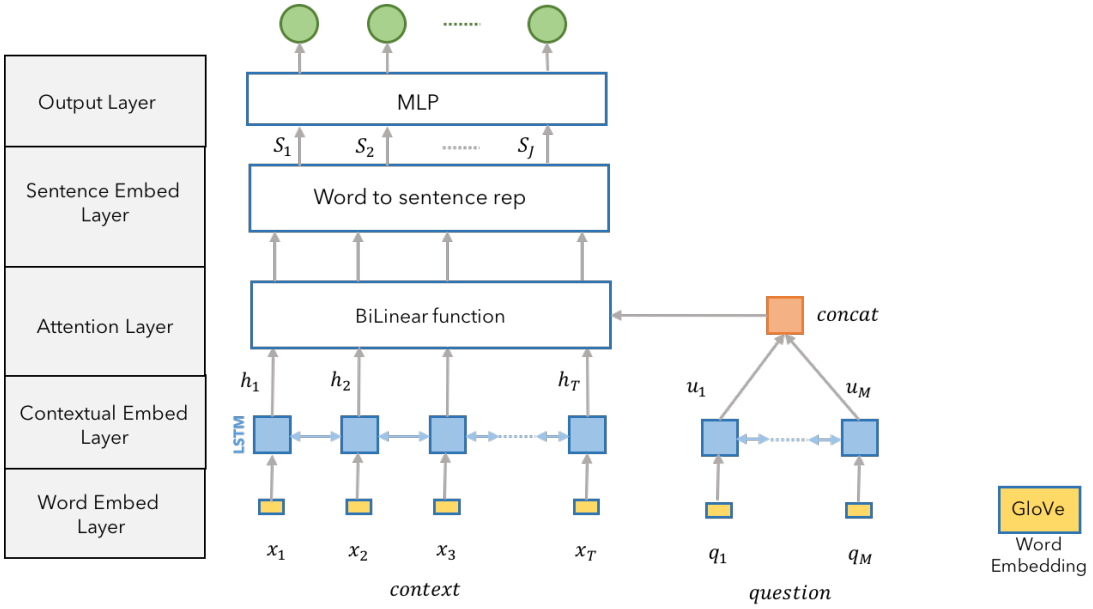


図 4.3: BiLSTM モデルの概要

4.3 データ拡張戦略

MCTest はデータサイズが非常に小さいという欠点があるため，我々はデータ拡張戦略を追加で適用し実験を行った．データ拡張戦略は，既存のデータからデータを拡張することでシステムの学習に用いることのできるデータの数を増やし，シス

テムの改善を目指す機械学習手法の一つであり，コンピュータビジョン分野での研究などでよく用いられている．これまでの MRC の研究でも様々なデータ拡張手法が提案されてきた．例えば Jia らはコンテキスト中の単語を変化させ質問生成を行うデータ拡張戦略手法として AddSent と AddAny という手法を提案している [16] が，我々はより単純な方法として，データセット内からランダムに複数の文を抽出しそれらをコンテキスト内のランダムな位置に挿入するデータ拡張戦略手法を用いる．ランダムに挿入する文の数は 5, 10, 20 と決め打ちで変化させ，データ拡張戦略が ASS システムにもたらす効果を確かめる．

4.4 回答に必要な文のみを用いた場合の選択肢式問題への影響

最後に我々が手動でアノテーションした，コンテキスト中の回答に必要な文のみを用いた場合に，元々のタスクである選択肢式問題にどのように影響を及ぼすかを調査する．Richardson らの提案する 2 つのモデルを用いて選択肢式問題に取り組む [10]．一つが Sliding Window モデル，もう一つが Distance Based モデルである．図 4.4 にそれぞれのモデルのアルゴリズムを示す．Sliding Window モデルでは質問と回答候補の単語の集合がテキストとマッチするかを特徴とし，それぞれの回答候補のスコアを計算する．この Sliding Window では捉えられない単語間の距離を考慮したモデルが Distance Based モデルである．Sliding Window モデルで用いる単語をカウントしたものの逆数は TF-IDF を意識したものである．

Algorithm 4 Baseline Algorithms

Require: コンテキストを P，コンテキストの単語の集合を PW，コンテキスト内の i 番目の単語を P_i，質問の単語の集合を Q，回答候補の単語の集合を $A_{1..4}$，ストップワードの集合を U とする．

Define: $C(w) := \sum_i \mathbb{I}(P_i = w)$

Define: $IC(w) := log(1 + \frac{1}{C(w)})$

Algorithm 5 Sliding Window

1: **for** $i = 1$ to 4 **do**
2: $S = A_i \cup Q$
3: $sw_i = \max_{j=1..|P|} \sum_{w=1..|S|} \begin{cases} IC(P_{j+w}) & if P_{j+w} \in S \\ 0 & otherwise \end{cases}$
4: **end for**
5: **return** $sw_{1..4}$

Algorithm 6 Distance Based

1: **for** $i = 1$ to 4 **do**
2: $S_Q = (Q \cap PW) \setminus U$
3: $S_{A_i} = ((A_i \cap PW) \setminus Q) \setminus U$
4: **if** $|S_Q| = 0 or |S_{A_i}| = 0$ **then**
5: $d_i = 1$
6: **else**
7: $d_i = \frac{1}{|P|-1} \min_{q \in S_Q, a \in S_{A_i}} d_p(q, a)$
8: **end if**
9: **end for**
10: **return** $d_{1..4}$

Algorithm 7 Algorithm SW

1: **return** $\arg\max_i sw_{1..4}$

Algorithm 8 Algorithm SW+D

1: **return** $\arg\max_i sw_{1..4} - d_{1..4}$

図 4.4: 選択肢式問題に取り組むモデル

また，比較モデルとして MCTest の現在の State-of-the-art モデルである Reading Strategies Net [17] と Finetuned QACNN [18] を用いる．以下に簡単にそれぞれのモデルの概要を述べる．

Reading Strategies Net

Reading Strategies Net は現在の MCTest の State-of-the-art のモデルである．Reading Strategies Net は大規模な MRC データセットの一つである RACE を用いて transformer 言語モデルを fine-tuning し，3 つの戦略: (1) 入力される系列の順番をオリジナルで読むか逆から読むかを考慮する BACK AND FORTH READING，(2) 品詞情報を用いて重要な単語に学習可能な新たなエンベディングを加える HIGHLIGHTING，(3) 質問と回答を生成する SELF-ASSESSMENT を適用したモデルである．このモデルは 6 つの非抽出型の MRC タスクで State-of-the-art を達成した．

Finetuned QACNN

Finetuned QACNN は大規模な MRC データセットの一つである MovieQA [19] をソースとし，既存の QA モデルである QACNN [20] によって MCTest へ転移学習を行ったモデルであり，Reading Strategies Net の一つ前の State-of-the-art のモデルである．

どちらのモデルも非常に大きなサイズの他の MRC データセットを用いて複雑なシステムを学習している．それに対して我々はニューラルネットワークではなく，単純な表層系を考慮するシステムでも，回答に必要な文を選択することで回答精度が向上することを示す．

第 5 章　結果

本章では、回答に必要となる文を選択する ASS に対して提案した 3 つのモデルの実験結果について述べる．また，提案するデータ拡張戦略の効果を示す．最後に，回答に必要となる文のみを用いた場合に選択肢式問題の回答にどのように効果がでるかを確認する実験の結果について述べる．

5.1　ASS タスク

Text-Matching モデル

Text-Matching モデルの topN と threshold モデルの結果をそれぞれ 表 5.1 と表 5.2 に示す．まずはじめに topN モデルでの結果について述べる．top1 モデルは質問タイプが Single の問題に対して高い EM スコアとなりそれぞれ 39.02 %, 43.33 % となった．F1 スコアは N の値ではあまり変わらないが，Test データでは N=2 の時に最も F11 スコアが高く 31.02 となった．また，この時の Precision と Recall の値はそれぞれ 30.63, 36.62 である．一方で N=3 以上の値では EM スコアは 0 となり，F1 値はどの N を取っても同等の値となった．

次に threshold モデルでは，どの threshold の値でも topN に比べて EM スコアは非常に低くなった．全体の F1 スコアは topN とほぼ同等だが，Test データで threshold=0.1 の時，topN よりも良い値となり 33.26 となった．この時の Precision と Recall の値はそれぞれ 30.73, 47.1 となった．

表 5.1: Text-Matching topN の回答精度 (%) と F1 値

	1		2		3		5	
	EM	F1	EM	F1	EM	F1	EM	F1
開発								
Single	*39.02*	*39.02*	*0.00*	*40.65*	*0.00*	*35.37*	*0.00*	*27.64*
Multiple	*0.00*	*11.33*	*2.53*	*18.48*	*0.00*	*23.31*	*0.00*	*25.96*
All	*13.33*	*20.79*	*1.67*	*26.05*	*0.00*	*27.43*	*0.00*	*26.54*
テスト								
Single	*43.33*	*43.33*	*0.00*	*37.04*	*0.00*	*30.56*	*0.00*	*27.04*
Multiple	*0.00*	*21.75*	*6.00*	*27.40*	*0.00*	*31.11*	*0.00*	*31.03*
All	***16.25***	*29.84*	*3.75*	***31.02***	*0.00*	*30.90*	*0.00*	*29.53*

表 5.2: Text-Matching threshold の回答精度 (%) と F1 値

	0.05		0.07		0.1		0.2	
	EM	F1	EM	F1	EM	F1	EM	F1
開発								
Single	0.00	29.79	*0.00*	*31.93*	2.44	34.97	*24.39*	*40.65*
Multiple	0.00	25.63	*0.00*	*26.11*	0.00	21.62	*1.27*	*10.11*
All	0.00	27.05	*0.00*	*28.1*	0.83	26.18	*9.17*	*20.55*
テスト								
Single	1.11	26.73	*1.11*	*29.50*	7.78	35.48	*25.56*	*35.00*
Multiple	0.00	31.76	*0.00*	*31.80*	1.33	31.93	*1.33*	*19.89*
All	0.42	29.88	*0.42*	*30.94*	3.75	**33.26**	*10.42*	*25.56*

BiDAF モデル次に BiDAF モデルでの実験の結果を表 5.3 に示す．表の h はモデル内で用いる BiLSTM の隠れ層の次元数を示している．Text-Matching モデルに比べて，EM スコア，F1 値のどちらも低い結果となった．

表 5.3: BiDAF モデルの回答精度 (%) と F1 値

h	EM	F1
100	*2.20*	*14.5*
200	*1.20*	*11.5*
300	*1.20*	*9.80*

BiLSTM モデル

次に BiLSTM モデルでの実験の結果を表 5.4 に示す．表の h はモデル内で用いる BiLSTM の隠れ層の次元数を示している．BiLSTM モデルも BiDAF モデルと同様に Text-Matching モデルに比べて，EM スコア，F1 値のどちらも低い結果となった．

表 5.4: BiLSTM モデルの回答精度 (%) と F1 値

h	EM	F1
100	*0.20*	*11.40*
200	*0.70*	*10.90*
300	*1.20*	*11.90*

データ拡張戦略の適用

表 5.5 は，$h = 100$ の時の BiLSTM モデルにデータ拡張戦略を適用した場合の実験の結果を示している．表の行は開発データとテストデータから取得したデータ拡張に用いる文の数，列は拡張後の訓練データのサイズを示している．表が示す通り EM スコアに対しては僅かだがスコアが上昇した．しかしながら，F1 値についてはよくなる場合もあれば，データ拡張により悪くなる場合があった．これは挿入する文が多い場合，学習を妨げるノイズとなっている可能性が考えられる．

表 5.5: データ拡張戦略を適用した場合の回答精度 (%) と F1 値

挿入する文の数	拡張後のデータサイズ			
	オリジナル *(280)*		*1000*	
	EM	F1	EM	F1
5			***1.70***	***12.90***
10	*0.20*	*11.40*	*1.60*	*8.30*
20			*1.00*	*7.60*

5.2 回答の正誤と MRC 能力の関係

最後にオリジナルのデータセット (コンテキスト全文を使用した場合) と回答に必要な文のみを用いて MCTest の選択肢式問題に取り組んだ場合の回答精度の比較を行った．表 5.6 に選択肢式問題に取り組んだ場合の結果を示す．コンテキスト全文を用いた場合，提案手法は他のモデルより回答精度が低いが，アノテーションした回答に必要な文のみを用いた場合，Finetuned QACNN の 76.4 % を超え，現在の State-of-the-art のモデルである Reading Strategies Net に迫る回答精度を達成した． 回答に必要な文のみを用いることによって，オリジナルデータでの回答精度から SW モデルは 5 ポイント，SW+D モデルでは 10 ポイントの改善を示した．表 5.7 には提案手法のより具体的なスコアを示している．表が示す通り，訓練，開発，テストデータの全てのデータにおいて，回答に必要な文のみを用いた場合に回答精度が大幅に向上している．

また図 5.1 には，提案モデルがオリジナルデータを用いた場合と回答に必要な文のみを用いた場合のそれぞれでどのような MRC の能力を必要とする問題が回答可能だったかを示しており，正解だった問題を赤，不正解だった問題を青で示している．図 5.1(a) はオリジナルデータを用いた場合，図 5.1(b) は回答に必要な文のみを用いた場合を示している．またそれぞれの MRC の能力を表 5.8 に示している．図 5.1(a) と図 5.1(b) を比較すると分かる通り，オリジナルデータを用いた場合に解けなかった問題および MRC 能力を必要とする問題が，回答に必要な文のみを用いることで回答可能になっていることがわかる．特に 2 (共参照の解決), 8 (橋渡し), 11 (概略句の関係) の能力を必要とする問題がかなり解決できている．0 (オブジェクトの追跡) の問題だけが唯一，回答に必要な文のみを用いた場合にオリジナルでは正解していた問題が不正解になった．

今回の実験により大規模なニューラルネットの構造を持つ複雑なモデルを必要としなくても，答えを導くために必要な文のみを抽出することにより，テキストの表層を用いる単純なモデルであってもそれらと同等の回答精度が得られることがわかった．これは MRC システムの開発にとって ASS が非常に重要な役割を担うと考えられる．

表 5.6: MC160 の回答精度 (%)

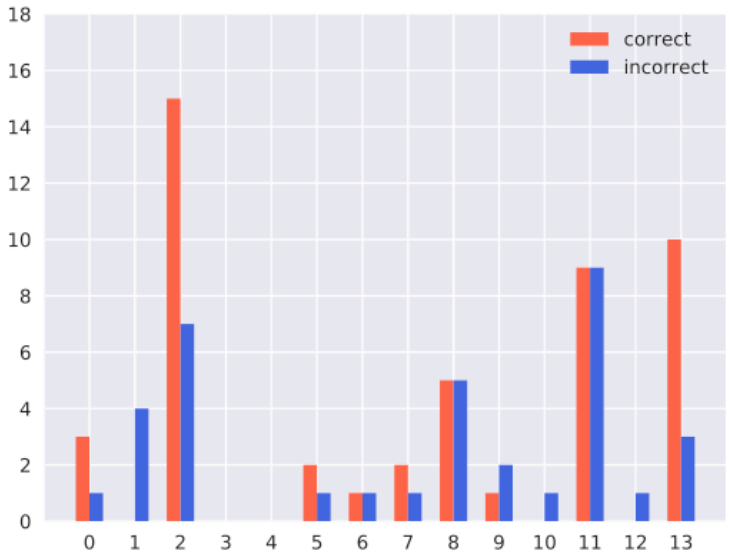
使用するデータ	モデル	EM
オリジナル	Finetuned QACNN [18]	*76.4*
	Reading Strategies Net (Single) [17]	*80.00*
	Reading Strategies Net (Ensemble) [17]	***81.7***
	提案手法 SW	*62.08*
	提案手法 SW+D	*68.33*
アノテーション後	提案手法 SW	*67.08*
	提案手法 SW+D	*78.33*

表 5.7: オリジナルと回答に必要な文のみを用いた場合の選択肢式問題での回答精度 (%) の比較

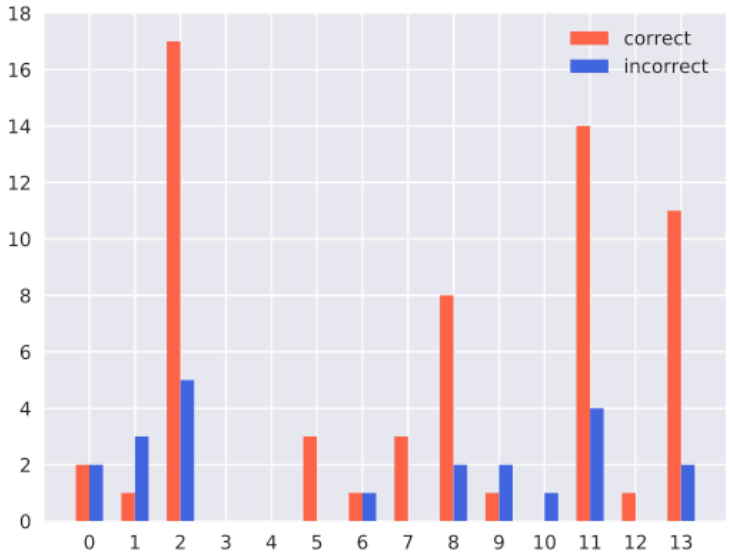
	訓練 280 Q's		開発 120 Q's		テスト 240 Q's	
	SW	SW+D	SW	SW+D	SW	SW+D
オリジナル						
Single	*69.7*	*75.76*	*64.15*	*67.92*	*69.64*	*77.68*
Multiple	*63.51*	*70.27*	*50.75*	*55.22*	*55.47*	*60.16*
All	*66.43*	*72.86*	*56.67*	*60.83*	*62.08*	*68.33*
アノテーション後						
Single	*80.3*	*86.36*	*77.37*	*84.91*	*70.54*	*83.04*
Multiple	*72.97*	*79.73*	*62.69*	*73.13*	*64.06*	*74.22*
All	*76.43*	*82.86*	*69.17*	*78.33*	*67.08*	*78.33*

表 5.8: MRC の能力の一覧

番号	スキル名	スキルの概要
0	オブジェクトの追跡	複数のオブジェクトの追跡や把握.
1	数学的推論	四則演算や幾何学的理解の能力，統計的推論や定量的推論.
2	共参照の解決	共参照の検出および解決.
3	論理的推論	条件式、数量詞、否定、および推移性など述語論理の理解.
4	類推	比喩の理解.
5	因果関係	"why", "because", "the reason for" などの明示的な表現で表される因果関係の理解.
6	時空間関係	複数の状態やイベントなどの空間的または時間的関係の理解.
7	省略記号	省略された情報の認識.
8	橋渡し	文法的および語彙的知識（例えば、同義語、上位語など）による推論.
9	精巧さ	既知の事実、一般的な知識などを使用した推論.
10	メタ知識	メタ的な視点からの読者，作家，テキストのジャンルなどの知識.
11	概略句の関係	等位接続や従位接続を含む複雑な文の理解.
12	句読点	句読点の理解（例：括弧、ダッシュ、引用符、コロン、セミコロン）.
13	スキルなし	上位のスキルに該当しないもの.



(a) オリジナルデータを使用した場合



(b) 回答に必要な文のみを使用した場合

図 5.1: 回答の正誤と菅原らの MRC 能力との関係

第 6 章　結論

本論文では ASS が MRC システムにどのように貢献するか，また良い ASS システムの開発にはどのようなスキルが必要かについて調査した．具体的には既存の MRC データセットに対して，回答に必要となる文をアノテーションすることにより ASS に取り組むためのデータセットを作成し，モデルがコンテキストと質問文からコンテキスト内の必要な文を選択できるかを調べ，さらに回答が難しい問題にはどのような特徴があるかを示した．単語の表層を用いた単純な Text-Matching モデルと 2 種類の文脈を考慮したニューラルネットモデルを用いた実験では，どのモデルでも F1 値は大きくは変わらないことを示したが，ASS によって選択した文のみを用いて選択肢式問題に取り組む実験では，単純な Text-Matching モデルでも現在の State-of-the-art の大規模なデータセットを用いて言語モデルを fine-tuning した複雑なモデルに迫る回答精度を示した．

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松本裕治教授に心より感謝いたします．同じく，適切なご指導ご助言をいただいた指導教員の新保仁准教授と進藤裕之助教にも心より感謝いたします．また，本論文の審査をお受けしていただきました本学の中村哲教授に深く感謝いたします．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に感謝いたします．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北川祐子秘書に感謝いたします．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] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[2] Mengqiu Wang, Noah A Smith, and Teruko Mitamura. What is the jeopardy model? a quasi-synchronous grammar for qa. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007.

[3] Tomasz Jurczyk, Michael Zhai, and Jinho D Choi. Selqa: A new benchmark for selection-based question answering. In *Tools with Artificial Intelligence (ICTAI), 2016 IEEE 28th International Conference on*, pp. 820–827. IEEE, 2016.

[4] Yi Yang, Wen-tau Yih, and Christopher Meek. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 2013–2018, 2015.

[5] Saku Sugawara, Yusuke Kido, Hikaru Yokono, and Akiko Aizawa. Evaluation metrics for machine reading comprehension: Prerequisite skills and readability. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1, pp. 806–817, 2017.

[6] Divyansh Kaushik and Zachary C Lipton. How much reading does reading comprehension require? a critical investigation of popular benchmarks. *arXiv preprint arXiv:1808.04926*, 2018.

[7] Chenxi Liu, Junhua Mao, Fei Sha, and Alan L Yuille. Attention correctness in neural image captioning. In *AAAI*, pp. 4176–4182, 2017.

[8] Abhishek Das, Harsh Agrawal, Larry Zitnick, Devi Parikh, and Dhruv Batra. Human attention in visual question answering: Do humans and

deep networks look at the same regions? *Computer Vision and Image Understanding*, Vol. 163, pp. 90–100, 2017.

[9] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

[10] Matthew Richardson, Christopher JC Burges, and Erin Renshaw. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 193–203, 2013.

[11] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*, 2017.

[12] Stanford core nlp. `https://stanfordnlp.github.io/CoreNLP/`.

[13] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pp. 1693–1701, 2015.

[14] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*, 2016.

[15] 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, 2014.

[16] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*, 2017.

[17] Kai Sun, Dian Yu, Dong Yu, and Claire Cardie. Improving machine reading comprehension with general reading strategies. *arXiv preprint arXiv:1810.13441*, 2018.

[18] Yu-An Chung, Hung-Yi Lee, and James Glass. Supervised and un-

supervised transfer learning for question answering. *arXiv preprint arXiv:1711.05345*, 2017.

[19] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Movieqa: Understanding stories in movies through question-answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4631–4640, 2016.

[20] Tzu-Chien Liu, Yu-Hsueh Wu, and Hung-Yi Lee. Query-based attention cnn for text similarity map. In *ICCV Workshop*, 2017.