令和 2 年度茨城大学大学院理工学研究科情報工学専攻　修士

学位論文

機械翻訳を利用した BERT による日本語 QA システムの構築

所属　　情報工学専攻

著者　　趙一鳴 (19nm721g)

指導教員　　新納浩幸教授

令和 3 年 2 月 5 日（金)

令和 2 年度茨城大学大学院理工学研究科情報工学専攻　修士学位論文

機械翻訳を利用した BERT による日本語 QA システムの構築

著者

趙一鳴 (19nm721g)

指導教員

新納浩幸教授

論文要旨

現在 BERT を利用し、QA 問題に正確率が高いが、QA 問題に bert を fine-tuning をするために文章と問題と答えをまとめる訓練データがあるが、今のところに英語バージョンしかない。日本語バージョンの訓練データが作れるが、かなりの人力とお金がかかるので。今のところにまだ日本語バージョンがない。

SQuAD(英語バージョンの訓練データの名前) のように日本語の訓練データを用意できれば日本語バージョンのモデルが構築できるが、そのような訓練データを作成することは多大に経費がかかるから現実的には不可能である。そこで本研究では機械翻訳を利用し、日本語の文章とこの文章に対するの問題を英語に翻訳し、学習済みの英語バージョンの SQuAD のモデルに文章と問題を与えて答えを得て、それをまた日本語に訳す。最後にこの時得られた日本語バージョンの答えと最初に自分で用意した日本語バージョンの答えを比較する。

1. 日本語の文章を探す。

2. 各文章の中からいくつかの質問とその答えを探し出す

3. 日本語の文章を英語に訳す（Google 翻訳を利用する）

4. 各文章での質問を Google 翻訳を利用し、英語に訳す。

5 英語の文章と質問を QA モデルに入れて、英語の答えをもらう。

6. 先ほど得た英語の答えを Google 翻訳で、日本語に訳す。

7.6 からもらった日本語の答えと 2 から用意した答えを比較する。

SQuAD には多くの問題がありますが、実用的な記事は少なく、短いため、データセット全体の語彙やトピックの多様性が制限されます。

したがって、SQuAD で適切に機能するモデルをより複雑な問題に使用する場合、スケーラビリティと適用性の両方に問題があります。

ですがしかし、この論文の目的のためだけに、翻訳ソフトウェアの使用は、日本語のデータセットがないという問題を補うことができます。

Master's Thesis in Scholastic 2020, Major in Computer and Information Sciences,
Granduate School of Science and Engineering, Ibaraki University

Construction of Japanese QA system by BERT using machine translation

Auther

YIMING ZHAO (19NM721G)

Adviser

Prof. Hiroyuki Shinnou

Abstract

Currently using BERT, the accuracy rate is high for QA problems, There is training data that summarizes sentences, questions and answers for fine-tuning bert in QA questions, but so far there is only an English version. You can create a Japanese version of the training data, but it costs a lot of manpower and money. There is no Japanese version yet.

If you can prepare Japanese training data like SQuAD (name of training data of English version), you can build a model of Japanese version, but it is realistic because it costs a lot to create such training data. Is impossible. Therefore, in this study, we used machine translation to translate Japanese sentences and questions about this sentence into English, and gave sentences and problems to his learned English version of his SQuAD model to get answers. Is translated into Japanese again. Finally, compare the Japanese version of the answer obtained at this time with the Japanese version of the answer that you prepared yourself first. 1. Search for Japanese sentences. 2. Find some questions and their answers in each sentence 3. Translate Japanese sentences into English (use Google Translate) 4. Translate the questions in each sentence into English using Google Trans- late. 5 Put English sentences and questions in the QA model and get an English answer. 6. Translate the English answer you got earlier into Japanese with Google Translate.

Compare the Japanese answer given in 7.6 with the answer prepared in 2. Although SQuAD has many problems, it has few practical articles and is short, limiting the diversity of vocabulary and topics throughout the dataset. Therefore, there are problems with both scalability and applicability when using a model that works well with SQuAD for more complex problems. However, for the purposes of this paper only, the use of translation software can compensate for the problem of lack of Japanese datasets.

目次

第 1 章

序論

1.1 背景

現在 BERT を利用し、QA 問題に正確率が高いが、QA 問題に bert を fine-tuning をするために文章と問題と答えをまとめる訓練データがあるが、今のところに英語バージョンしかない。日本語バージョンの訓練データが作れるが、かなりの人力とお金がかかるので。今のところにまだ日本語バージョンがない。

1.2 目的

SQuAD(英語バージョンの訓練データの名前) のように日本語の訓練データを用意できれば日本語バージョンのモデルが構築できるが、そのような訓練データを作成することは多大に経費がかかるから現実的には不可能である。そこで本研究では機械翻訳を利用し、日本語の文章とこの文章に対するの問題を英語に翻訳し、学習済みの英語バージョンの SQuAD のモデルに文章と問題を与えて答えを得て、それをまた日本語に訳す。最後にこの時得られた日本語バージョンの答えと最初に自分で用意した日本語バージョンの答えを比較する。

1.3 概要

1. 日本語の文章を探す。
2. 各文章の中からいくつかの質問とその答えを探し出す
3. 日本語の文章を英語に訳す（Google 翻訳を利用する）

4. 各文章での質問を Google 翻訳を利用し、英語に訳す。

5 英語の文章と質問を QA モデルに入れて、英語の答えをもらう。

6. 先ほど得た英語の答えを Google 翻訳で、日本語に訳す。

7.6 からもらった日本語の答えと 2 から用意した答えを比較する。

利用するツール：BERT、Google 翻訳

第 2 章

関連研究

2.1 Bag-of-words

Bag-of-words モデルは、情報検索の分野で一般的に使用されるドキュメント表現方法です。

情報検索では、BOW モデルは、ドキュメントの場合、単語の順序、文法、構文、およびその他の要素を無視し、それをいくつかの語彙のコレクションとして扱うと想定します。（ドキュメント内の各単語の出現は独立しています。 他の単語は出るかどうか関係がありません。順序も関係ありません）

つまり、ドキュメントの任意の位置に表示される単語は、ドキュメントのセマンティクスに影響されることなく、独立しています。 では、それはどういう意味ですか？ 次に、具体的な例を示します。

John likes to watch movies. Mary likes too.

John also likes to watch football games.

上記の 2 つの文に現れる単語に基づいて、dictionary が作成できます。

”John”: 1, ”likes”: 2, ”to”: 3, ”watch”: 4, ”movies”: 5, ”also”: 6, ”football”: 7, ”games”: 8, ”Mary”: 9, ”too”: 10

dictionary には 10 個の単語が含まれており、各単語には一意のインデックスがあります。それらの順序は、文に現れる順序とは関係がないことに注意してください。この dictionary によれば、上記の 2 つの文を次の 2 つのベクトルとして再表現できます。

[1, 2, 1, 1, 1, 0, 0, 0, 1, 1]

[1, 1, 1, 1, 0, 1, 1, 1, 0, 0]

これらの 2 つのベクトルには、合計 10 個の要素が含まれます。ここで、i 番目の要素は、辞書の i 番目の単語が文に現れる回数を表します。したがって、BoW モデルは、histogram と見なすことができます。テキスト検索および処理アプリケーションでは このモデルでは、このモデルを使用して単語の頻度を簡単に計算できます。

しかし、上記から、ドキュメントベクトルを構築する過程で、元の文に単語が現れる順序を表現しなかったことがわかります。前の単語と後ろの単語の関係もわかりません。

2.2　ONE-HOT

機械学習アルゴリズムでは、人の性別が男性と女性であり、母国が中国、米国、フランスなどであるなどの分類機能に遭遇することがよくあります。

これらの固有値は連続的ではなく、離散的で無秩序です。通常、その機能をデジタル化する必要があります。では、機能のデジタル化とは何ですか？例は次のとおりです。

性別の特徴：[”男性”、 ”女性”]

祖国の特徴：[「中国」、「米国」、「フランス」]

スポーツの特徴：[「サッカー」、「バスケットボール」、「バドミントン」、「テーブルテニス」]

例えば、彼の特性は次のようになります [”男”、 ”中国”、 ”ピンポン”]、[0,0,4] を使用して表すことができますが、そのような特徴処理はそうではありませんそれを直接機械学習アルゴリズムに入れます。カテゴリが乱れているからです。

例：性別特性：[”男性”、 ”女性”] 男性= 10、女性= 01、

祖国特性：[”中国”、 ”米国、”フランス ”] 中国= 100、米国= 010 、フランス= 001、

スポーツ機能：[”サッカー”、 ”バスケットボール”、 ”バドミントン”、 ”テーブルテニス”]（ここでは N = 4）：サッカー= 1000、バスケットボール= 0100、バドミントン= 0010

テーブルテニス= 0001。したがって、サンプルが [”male”、 ”China”、 ”テーブルテニス”] の場合、完全な機能のデジタル化の結果は次のようになります。[1、0、1、0、0、0、0、0、1]。

しかし、上記から、特徴が多すぎると、特徴ベクトルが非常に大きくなり、非常にまばらになることもわかります。

2.3　WORD2VEC

2.3.1 単語ベクトルを使用する理由

自然言語処理では通常、単語を個別の単一シンボルとして扱います。たとえば、「cat」という単語は Id537 として表され、「dog」という単語は Id143 として表される場合があります。 これらの記号のコーディングは不規則であり、異なる単語間の可能な関連についての情報を提供することはできません。 つまり、「犬」という単語に関する情報を処理する場合、モデルは「猫」に関する既知の情報を使用できなくなります（たとえば、すべて動物であり、4 本の足を持ち、ペットとして使用できるなど)。 語彙を上記の独立した離散記号として表現すると、データがさらにまばらになるため、統計モデルをトレーニングするときに、より多くのデータを探す必要があることがわかります。 語彙のベクトル表現は、上記の問題を克服します。

NLP で最も直感的で現在最も一般的に使用されている単語表現方法は One-hot Representation です。これは、単語を記号化する方法です。つまり、各単語のベクトルサイズは dictionary のサイズであり、ベクトルの位置は 1 です。 もちろん、他の位置は 0 です。このようなベクトルには、前の単語と後ろの単語は関係がありません。

2.3.2 one hot を使わない理由

one hot では、文字の数と同じ数の次元ベクトルがあります。10,000 文字の場合、各文字ベクトルは 10,000 次元になります（一般的に使用される文字は多くない場合があり、数千程度ですが、単語の概念によると 、一般的に使用される数十万の単語が存在する可能性があります)、これは計算上耐えられない次元の爆発を引き起こす可能性があるため、連続的なベクトル表現が生成されます。たとえば、100 次元の実数ベクトルを使用して単語を表すと、大幅に削減されます。 もちろん、より深い理由は、wors2vec の単語と単語ベクトルが意味情報を含むことができ、ベクトルの角度コサインが単語と単語の類似性をある程度表現できることです。

2.3.3 word embedding とは

単語の意味を単語のベクトルに入れる方法は？ 1954 年に Harris によって提案された distributional hypothesis は、このアイデアの理論的基礎を提供しました。類似した意味を持つ単語は、類似したベクトルを持っています。 distributional hypothesis に基づく単語表現方法は、異なるモデリングに従って、マトリックスベースの分布表現、クラス

ターベースの分布表現、およびニューラルネットワークベースの分布表現の 3 つのタイプに分けることができます。 単語の埋め込みは、通常、ニューラルネットワークに基づく分散表現です。

たとえば、ドキュメントが与えられた場合、そのドキュメントは「A B A C B F G」などの一連の単語であり、ドキュメント内の異なる単語ごとに対応するベクトル（通常は低次元のベクトル）表現を取得したいと考えています。 たとえば、「ABACBFG」のシーケンスの場合、最終的に次のようになります。対応するベクトルは [0.1 0.6 -0.5]、B の対応するベクトルは [-0.2 0.9 0.7]、このベクトルは単語埋め込みと呼ばれます。

2.3.4 ベクトル空間モデルとは

ベクトル空間モデル（VSM）は、連続したベクトル空間で単語を表現し、意味的に類似した単語が隣接するデータポイントにマップされます。 ベクトル空間モデルは、自然言語処理の分野で長く豊富な歴史がありますが、このモデルを使用するほとんどすべての方法は、分散された仮定に依存しています。コアアイデアは、コンテキストに表示される単語が同様のセマンティクスを持っているということです。 この仮説を用いた研究手法は、大きく分けて、カウント方式と予測手法の 2 つに分類されます。

要するに、カウントベースの方法は、大きなコーパスやその他の統計で語彙とそれに隣接する語彙の頻度を計算し、これらの統計を小さくて密なベクトルにマッピングします。予測方法は、隣接する単語から直接単語を予測しようとし、その過程で学習された小さくて密なネストされたベクトルを使用します。

2.3.5Word2Vec とは

word2vec は、完全に接続されたニューラルネットワークであり、隠れ層が 1 つだけあります。これは、特定の単語との関連性が高い単語を予測するために使用されます。 言い換えれば、それは言語モデルです。

全体的な手順：

1. 入力レイヤーで、単語がワンホットベクトルに変換されます。

2. 最初の隠れ層で、入力は W*x + b（x は入力ワードベクトル、W、b はパラメーター）であり、線形モデルを作成します。これは単なるマッピングであり、問題はありません。 もちろん、線形活性化関数はニューロンが線形である可能性があり、線形回帰関数と同等です。

3.Softmax 回帰を使用すると、第 3 層は単純に分類子と見なすことができ、最終的な出

力は各単語の確率です。

CBOW と Skip-Gram

Word2vec は、非常に効率的なワードネスティング学習を実行できる予測モデルです。

現在、より一般的に使用されている 2 つのバリアントがあります。つまり、CBOW と Skip-Gram です。

アルゴリズムの観点からは、これら 2 つの方法は非常に似ています。違いは、CBOW がソースワードコンテキストボキャブラリ（the cat sits on the）に基づいてターゲットボキャブラリ（mat）を予測するのに対し、Skip-Gram はその逆を行うことです。 ターゲットボキャブラリーを通じてソースボキャブラリーを予測します。

Skip-Gram モデルが CBOW の逆プロセスを採用する動機は、CBOW アルゴリズムが多くの分散情報を平滑化することです（たとえば、コンテキスト情報全体を単一の観測として扱います)。 多くの場合、このプロセスは小さなデータセットに役立ちます。 対照的に、Skip-Gram モデルは、各「コンテキストとターゲットの語彙」の組み合わせを新しい観測として扱います。これは、大規模なデータセットでより効果的です。

2.4　DOC2VEC

2.4.1 Doc2ve の原理

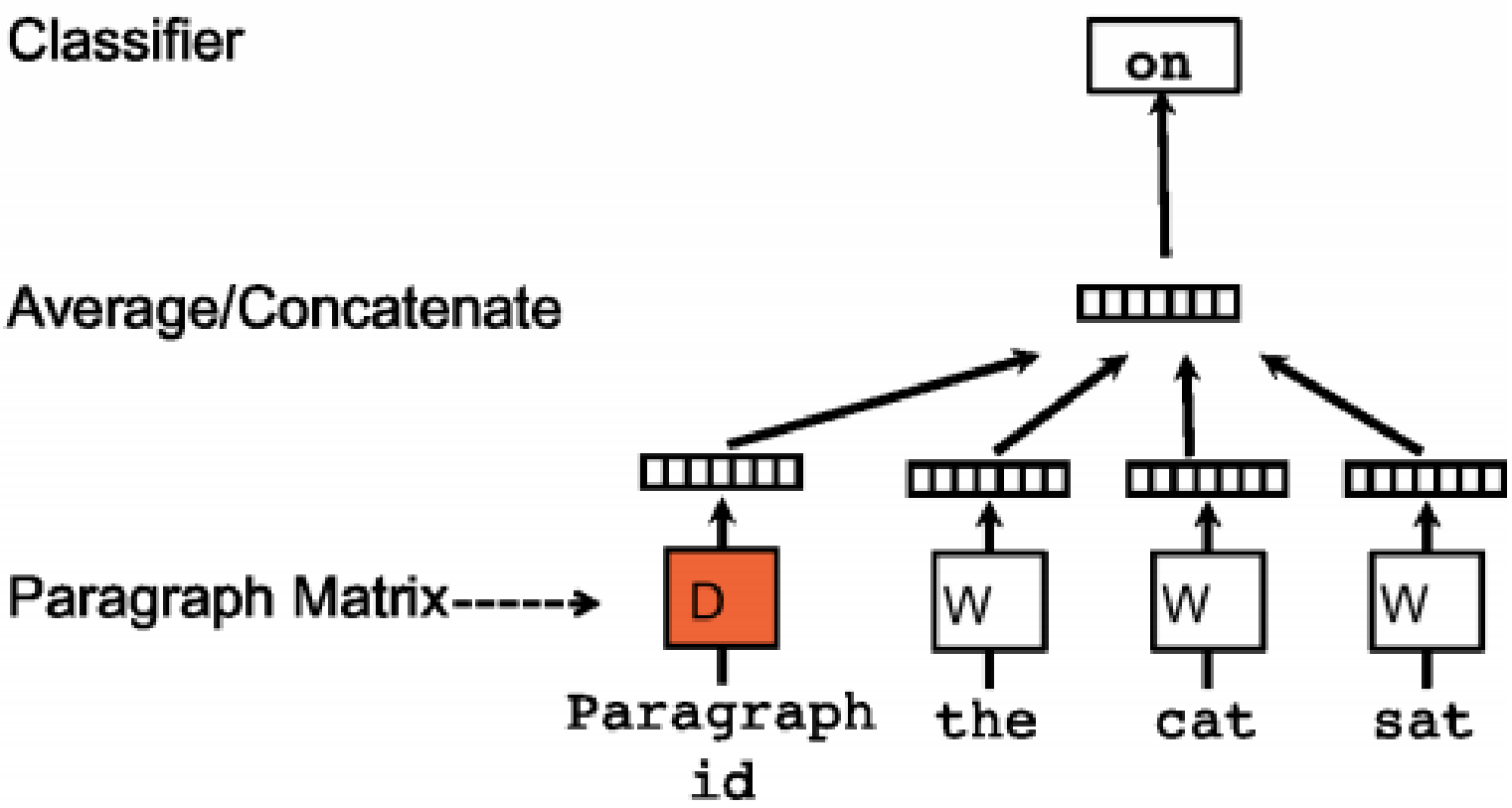
Doc2vec メソッドは、可変長のテキスト（文、段落、ドキュメントなど）から固定長の特徴表現を学習できる無教師モデルです。 Doc2vec は、ParagraphVector または SentenceEmbeddings とも呼ばれ、文、段落、ドキュメントのベクトル表現を取得できます。Word2Vec の拡張であり、文の長さが固定されていない、トレーニングサンプルとして異なる長さの文を受け入れるなどの利点があります。 Doc2vec アルゴリズムは、さまざまなドキュメントを表すベクトルを予測するために使用されます。モデルの構造は、bag-of-words モデルの欠点を克服する可能性があります。

Doc2vec モデルは、Word2Vec モデルに触発されています。 Word2Vec で単語ベクトルを予測する場合、予測された単語には単語の意味が含まれます。同じ構造が Doc2vec に組み込まれているため、Doc2vec はバッグオブワードモデルのセマンティクスの欠如を克服します。現在トレーニングサンプルがあると仮定すると、各文はトレーニングサンプルです。Word2Vec と同様に、Doc2vec にも 2 つのトレーニング方法があります。1 つは、Word2Vec の CBOW と同様に、段落ベクトルの分散メモリモデル（PV-DM）で

す。モデル、もう 1 つは、Word2Vec の Skip-gram モデルに類似した Paragraph Vector（PV-DBOW）の Distributed Bag ofWords バージョンです。

2.4.2　Distributed Memory Model of Paragraph Vectors

文ベクトルを訓練する方法は、単語ベクトルの方法と非常に似ています。 単語ベクトルをトレーニングすることの中心的なアイデアは、各単語のコンテキストに従って予測することです。つまり、コンテキスト内の単語のペアが影響を及ぼします。 同様に、Doc2vec も同じ方法でトレーニングできます。 たとえば、「水を飲みたい」という文の場合、「欲しい」という単語を予測したい場合は、他の単語に基づいて特徴を生成するだけでなく、他の単語や文に基づいて特徴を生成して予測することもできます。 したがって、Doc2vec のフレームワークを以下に示します。



Doc2vec では、各文は、行列 D の特定の列で表される一意のベクトルで表されます。各単語は、行列 W の特定の列で表される一意のベクトルでも表されます。文からスライドして固定長の単語をサンプリングするたびに、一方の単語が予測単語として使用され、もう一方の単語が入力単語として使用されます。入力ワードに対応するワードベクトルとこのセンテンスに対応するセンテンスベクトルパラグラフベクトルを入力レイヤーの入力として使用します。このセンテンスのベクトルと今回サンプリングしたワードベク

トルを加算して平均化または累積し、新しいベクトル X を形成します。次に、このベクトル X を使用して、このウィンドウで予測される単語を予測します。

Doc2vec と Word2vec の違いは、新しい文ベクトル Paragraph ベクトルが入力レイヤーに追加されることです。Paragraph ベクトルは、メモリの役割を果たす別の単語ベクトルと見なすことができます。平均的な単語ベクトルでは、Word2Vec を使用して単語ベクトルをトレーニングします。これは、各トレーニングが文中の単語トレーニングのごく一部のみを傍受し、このトレーニング単語以外の文内の他の単語を無視するため、各単語のトレーニングのみが行われるためです。ベクトル表現、文は、各単語のベクトルが合計され、平均化された単なる表現です。上記のように、平均単語ベクトルの欠点は、テキストの単語順序を無視します。 Doc2vec のパラグラフベクトルはこの欠陥を補います。トレーニングするたびに、スライドしてセンテンスのごく一部をインターセプトしてトレーニングします。パラグラフベクトルは同じセンテンスの複数のトレーニングで共有されるため、同じセンテンスは複数のトレーニングがあり、各トレーニングの入力には段落ベクトルが含まれています。それは文の主題と見なすことができ、それによって、文の主題は毎回入力の一部として訓練されます。このようにして、各トレーニングプロセスで、単語がトレーニングされるだけでなく、単語ベクトルが取得されます。同時に、トレーニングプロセス中に文をスワイプして一度に複数の単語を取得すると、各トレーニングの入力レイヤーの一部である共有パラグラフベクトルが、ベクトルのテーマをますます正確に表現します。 Doc2vec の PV-DM モデルの特定のトレーニングプロセスは、Word2Vec の CBOW モデルのトレーニングプロセスと同じです。

この段落ベクトルまたは文ベクトルも単語と見なすことができ、その機能はこの段落のコンテキストまたはトピックのメモリユニットと同等であるため、一般にこのトレーニングメソッドを Distributed Memory Model of Paragraph Vectors（PV-DM）と呼びます。トレーニング中に、コンテキストの長さを固定し、スライディングウィンドウ方式を使用してトレーニングセットを生成します。段落ベクトルまたは文ベクトルは、このコンテキストで共有されます。

トレーニング後、トレーニングサンプル内のすべての単語ベクトルと各文に対応する文ベクトルを取得します。Doc2vec はどのようにして新しい文の段落ベクトルを予測しますか？実際、新しい文を予測する場合、段落ベクトルはランダムに初期化され、モデルに入れられ、ランダムな勾配降下に基づいて繰り返し、最終的な安定した文ベクトルが取

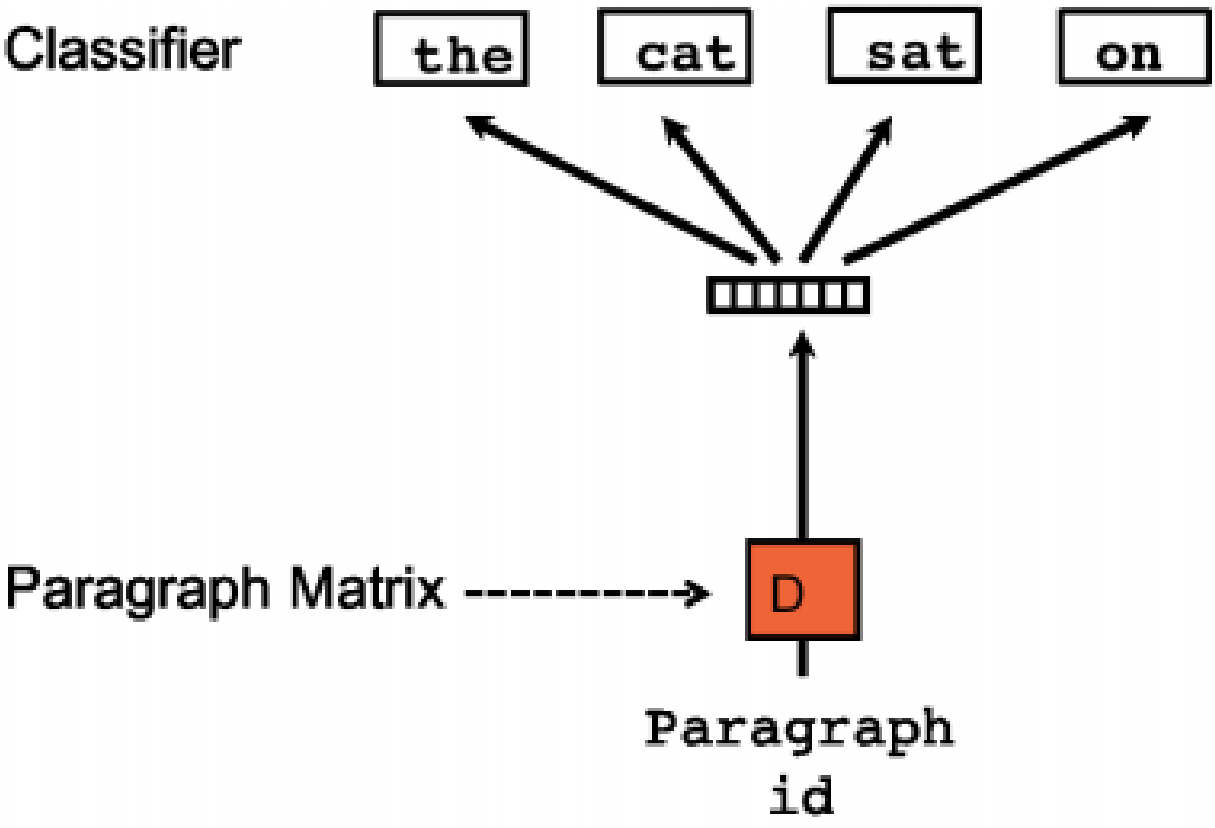
得されます。ただし、予測プロセスでは、モデル内のワードベクトル、投影レイヤーから出力レイヤーへの softmax の重みパラメーターは変更されないため、段落ベクトルのみが一定の反復で更新され、他のパラメーターは修正されています。これにはほとんど時間がかかりません。予測するパラグラフベクトルを計算できます。PV-DM のプロセスを要約すると、2 つの主要なステップがあります。

1. モデルをトレーニングし、既知のトレーニングデータから単語ベクトル W、softmax のパラメーター U と b、および段落ベクトルまたは文ベクトルを取得します。

2. 推論段階（推論段階）は、新しい段落の場合、そのベクトル式を取得します。具体的には、行列 D に列を追加し、W、U、b が固定されている場合は、上記の方法でトレーニングを行い、勾配降下法を使用して新しい列を取得し、新しい段落のベクトル式を取得します。

2.4.3 Distributed Bag of Words version of Paragraph Vector

別のトレーニング方法は、入力コンテキストを無視し、モデルに段落内のランダムな単語を予測させることです。 つまり、各反復で、ウィンドウがテキストからサンプリングされ、次に予測タスクとしてこのウィンドウから単語がランダムにサンプリングされ、モデルが予測に使用されます。入力は段落ベクトルです。このモデルを Paragraph Vector（PV-DBOW）の Distributed Bag ofWords バージョンと呼びます、次の図に示すように



上記の 2 つの方法では、PV-DM または PV-DBOW を使用して、段落ベクトルまたは文ベクトルを取得できます。 ほとんどのタスクでは、PV-DM 方式が適切に機能しますが、論文の著者は 2 つの方式の組み合わせも強く推奨しています。(Le Q, Mikolov T. Distributed representations of sentences and documents[C]// International conference on machine learning. 2014: 1188-1196.)

2.5　ELMO

word2vec、glove などの過去の単語ベクトルモデルでは、特定の単語に対して生成された単語ベクトルは固定されており、単語の多面的な現象を解決できません。たとえば、「apple」はコンテキストによって意味が異なります。 果物の一種、携帯電話、コンピューター、タブレットなどの会社名やブランド名を表しますが、単語ベクトルモデルを使用して単語ベクトルを生成する場合、単語は単語ベクトルにしか対応できず、明らかに私たちのニーズを満たすことはできません 。 2018 NAACL では、この問題を解決できる単語ベクトルモデル-ELMO（言語モデルからの埋め込み）がついに登場しました。

1. ELMO のメリット：

コンテキストを考慮し、コンテキストごとに異なる単語ベクトルを生成できます。

さまざまな文法的または意味的な情報を表現できます。 たとえば、「アクティビティ」という単語は、名詞または動詞のいずれかであり、件名または述語として使用できます。この状況に対応して、elmo は、さまざまな文法情報または意味情報に基づいて、さまざまな単語ベクトルを生成できます。

2. ELMO のデメリット：

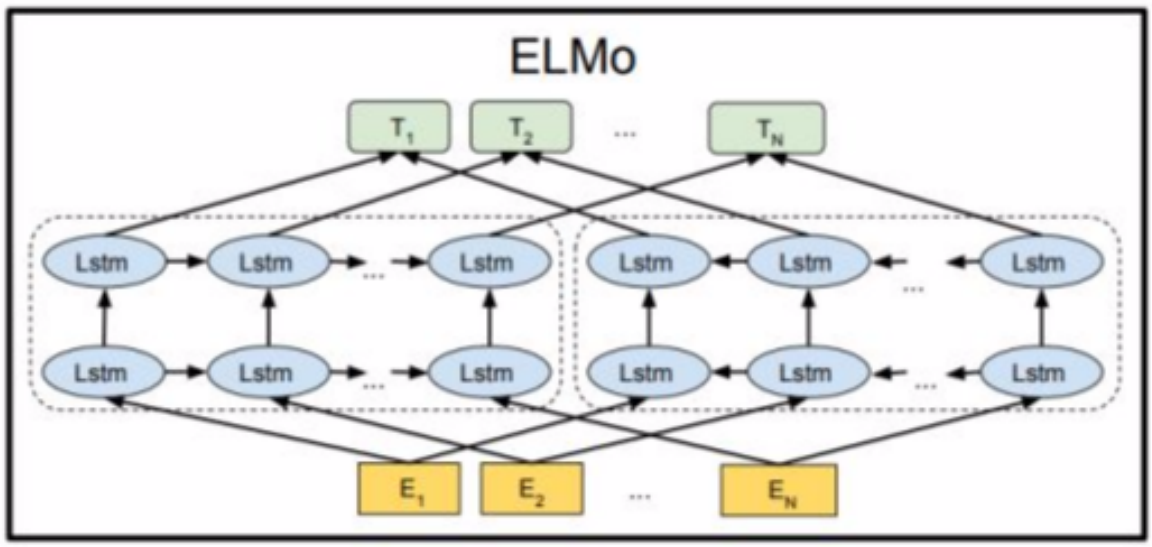
LSTM を使用して特徴を抽出すると、特徴を抽出する LSTM の機能は Transformer よりも弱くなります。

ベクトルスプライシングを使用してコンテキスト機能を融合します。この方法で取得されたコンテキスト情報は、期待したほど良くありません。

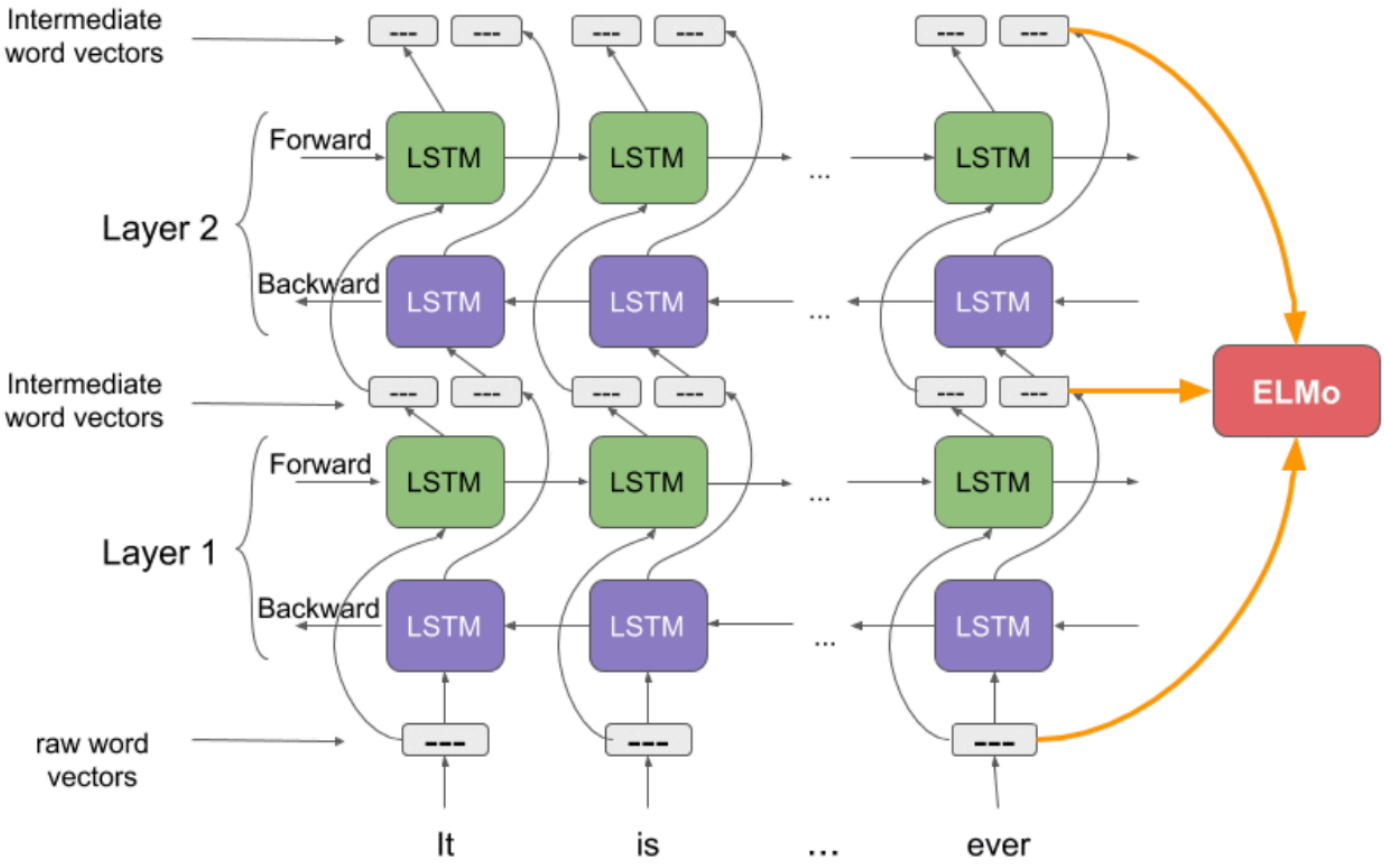
2.5.1　ELMO モデル構造

ELMO は、典型的な 2 段階のプロセスを使用します。最初の段階は事前トレーニングされた言語モデルでトレーニングし、2 番目のステージは特定のダウンストリームタスクを実行するときにダウンストリームタスクの入力を補足する機能として、事前トレーニングモデルから対応する単語の単語埋め込みを抽出することです。 入ってください。

最初の段階：次の図に示すように、2 層の双方向 LSTM を使用して入力をトレーニングする事前トレーニングプロセスでは、単語機能は静的な単語埋め込み（word2vec、glove など）を使用し、各 lstm 層は単語機能を前の単語ベクトルと結合します 以下の単語ベクトルとのスプライシングが現在の入力ベクトルとして使用され、第 1 層の lstm が構文上の特徴を取得し、第 2 層の lstm が意味上の特徴を取得します。



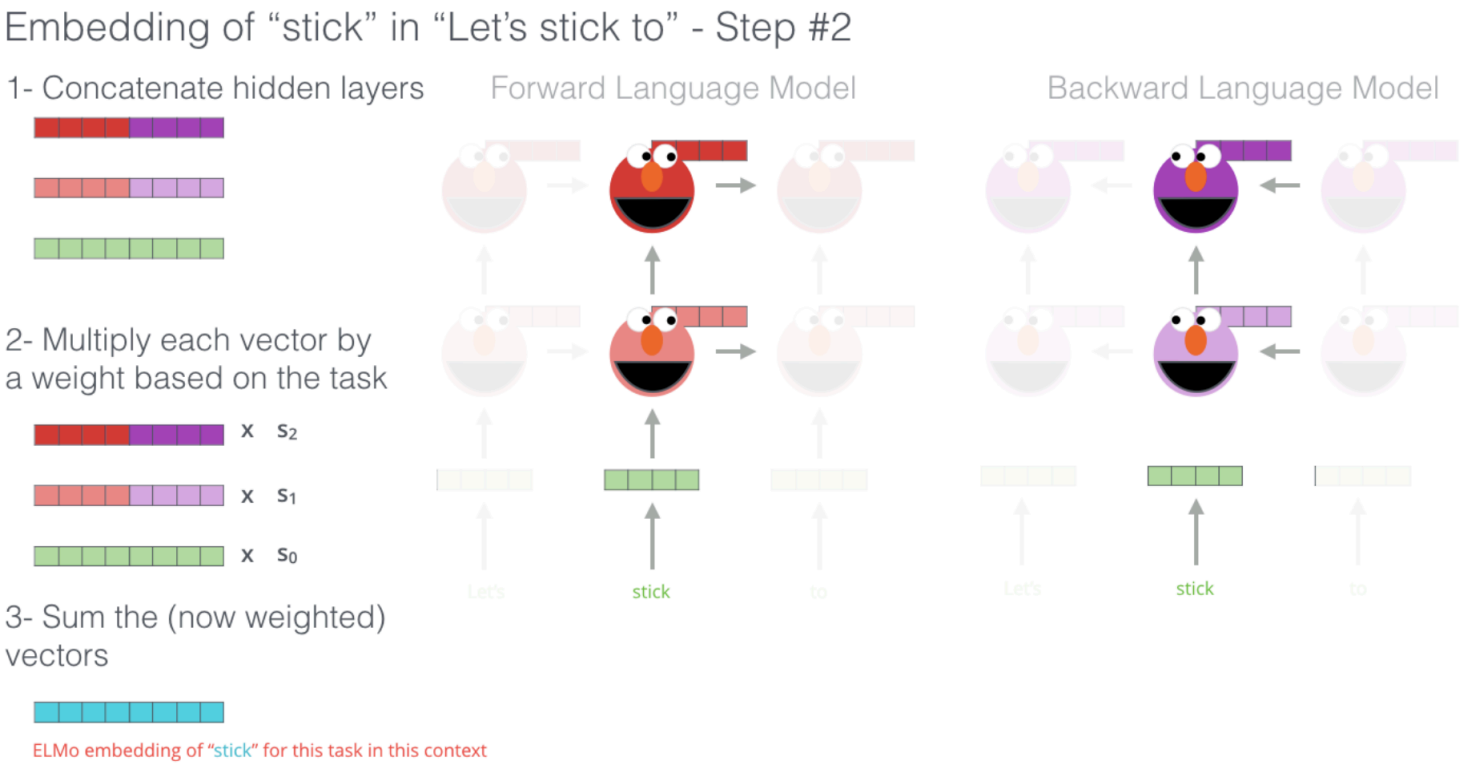
第 2 段階：下流のタスクに従って、第 1 段階の出力ベクトルをこの段階の入力ベクトルとして選択する方法を確認します。具体的には、第 1 段階でトレーニングされた ELMO モデルによって入力文が処理された後、単語を含む 3 つの埋め込みが取得されます。 特徴、構文的特徴（lstm の第 1 層の出力ベクトル）、意味的特徴（lstm の第 2 層の出力ベクトル）、これら 3 つの埋め込みは、ダウンストリームタスクに入力される最終ベクトルとして重み付けおよび合計できます。 「機能ベースの事前トレーニング」と呼ばれます。



注 1：ELMO は層間の残留接続を使用するため、双方向の効果を実現できます。

注 2：ELMO は、フォワード LSTM を使用して上記の情報を取得する双方向 LSTM を使用し、バックワード LSTM 構造を使用して次の情報を取得します。最後に、フォワード LSTM とバックワード LSTM によって生成されたベクトルは、コンテキスト情報の取得を表すためにスプライスされます。

2.5.2　ELMO での単語ベクトルの計算プロセス



上の図に示すように、「Let’s stick to」という文の場合、「stick」の単語ベクトルを計算する場合は、

1. スプライシング：最初に従来の単語ベクトル生成方法を使用して、図の緑色の長い正方形で示されているように、単語ベクトル vec を静的に生成します。次に、vec を最初のレイヤーの前方 lstm と後方 lstm に入力して、非表示のベクトル h11 を取得します。（図の薄いピンクの長い正方形で表される、最初のレイヤーの前方の lstm 計算結果を表します）および h12（図の薄い紫色の長い正方形で表される最初のレイヤーの後方の lstm 計算結果を表します）; 次に、2 番目のレイヤーに h11 を入力します。レイヤーの前方 lstm で、隠れベクトル h21（図の赤い長い四角で示される、2 番目のレイヤーの前方 lstm の計算結果を表す）を取得し、2 番目のレイヤーの後方 lstm に h12 を入力し、最初のレイヤーを示す隠しベクトル h22 を取得します。 2 階の後方 lstm 計算結果は、図の濃い紫色の長い四角で表されています。

静的な単語ベクトル vec は、それ自体をスプライスして大きな緑色の長方形を形成します。最初のレイヤーの非表示のベクトル h11 と h12 はスプライスされて h1 を形成し、2 番目のレイヤーの非表示のベクトル h21 と h22 はスプライスされて h2 を形成します。

2. 重み付け：ステップ 1 の 3 つのベクトルには独自の重みがあり（重みはトレーニン

グされていますか？)、3 つのベクトルは別々に重み付けされます。

3. 合計：ステップ 2 で 3 つのベクトルを合計します。このステップの出力は、「スティック」という単語のコンテキストから取得された単語ベクトルです。

2.5.3　総括する

ELMO はコンテキスト情報を表示できますが、一方向のコンテキスト情報しか表示できません。これは、ELMO の前方 lstm と後方 lstm のネットワークが完全に独立しているためです。つまり、前方 lstm トレーニングを使用する場合 、単語 t はすべての単語 t + 1 以降を見ることができません。同様に、後方 lstm トレーニングを使用する場合、単語 t はすべての単語 t-1 以前を見ることができないため、本質的には一方向しか見ることができません。 情報、いわゆる「双方向」は、情報を 2 方向につなぎ合わせるだけです。これが ELMO の制限です。 （これとは対照的に、バートは単語 t の上下を同時に見ることができるので、バートは本当に双方向の情報を取得しています）

2.6　BERT

2.6.1 はじめに

Google は、論文「BERT：言語理解のためのディープ双方向トランスフォーマーの事前トレーニング」で BERT モデルを提案しました。BERT モデルは、主にトランスフォーマーのエンコーダー構造を使用し、最も原始的なトランスフォーマーを使用します。一般に、BERT には次の特徴があります。

構造：トランスフォーマーのエンコーダー構造を採用していますが、モデル構造はトランスフォーマーよりも深くなっています。 Transformer Encoder には 6 つの Encoder ブロックが含まれ、BERT-base モデルには 12 の Encoder ブロックが含まれ、BERT-large には 24 の Encoder ブロックが含まれます。

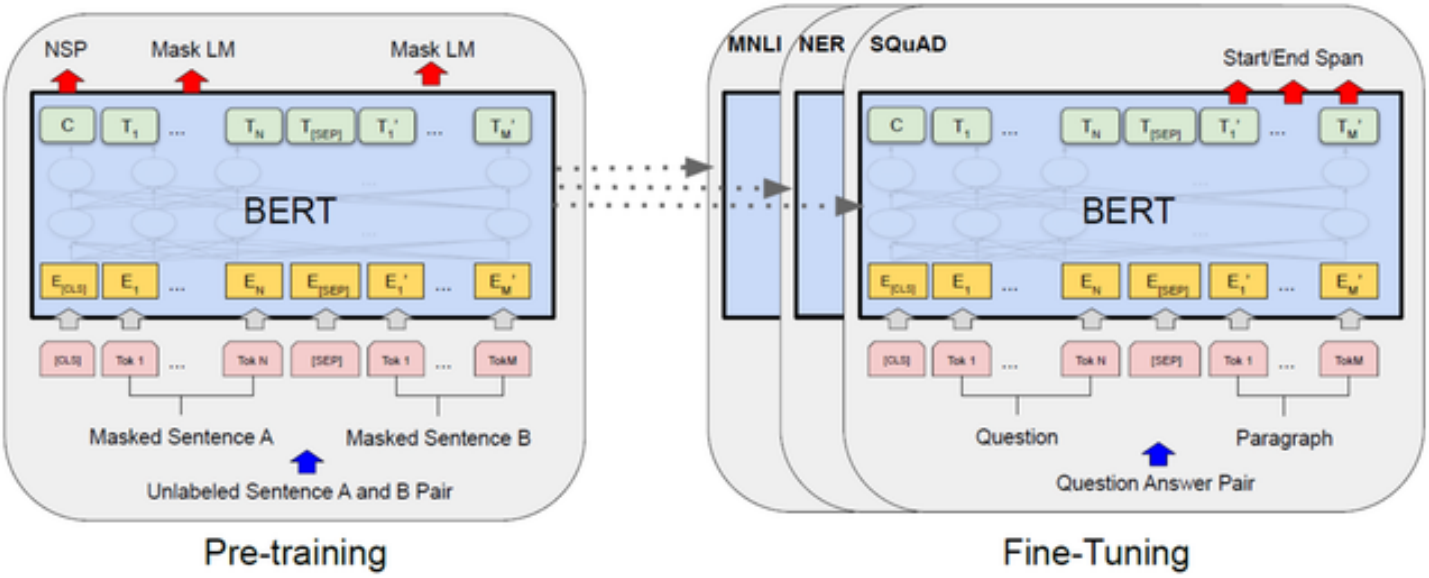
トレーニング：トレーニングは主に、事前トレーニング段階と微調整段階の 2 つの段階に分けられます。事前トレーニング段階は、Word2Vec、ELMo などに似ています。いくつかの事前トレーニングタスクに従って、大規模なデータセットでトレーニングされます。微調整段階は、テキスト分類、音声部分のタグ付け、質問回答システムなど、いくつかのダウンストリームタスクに後で使用されるときに微調整することです。BERT は、構造を調整せずにさまざまなタスクで微調整できます。

事前トレーニングタスク 1：BERT の最初の事前トレーニングタスクは Masked LM

です。これは、文中の単語の一部をランダムにカバーし、コンテキスト情報を使用してカバーされる単語を予測します。これにより、単語の意味が全文からよりよく理解されます。マスクされた LM は BERT の焦点であり、これは後で説明する biLSTM 予測方法とは異なります。

事前トレーニングタスク 2：BERT の 2 番目の事前トレーニングタスクは、次の文の予測タスクである次の文の予測（NSP）です。このタスクは、主にモデルが文間の関係をよりよく理解できるようにすることです。

2.6.2 BERT の構造



上の図は BERT の構造です。左の図は事前トレーニングプロセスを示し、右の図は特定のタスクの微調整プロセスを示しています。

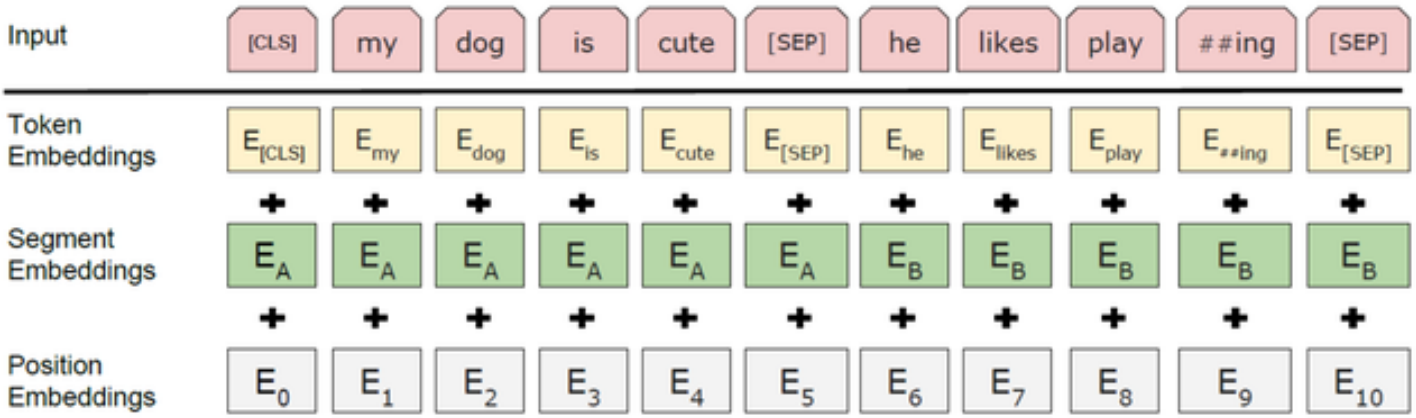
2.6.2.1 BERT 入力

BERT の入力には、文のペア（文 A と文 B）を含めることも、単一の文にすることもできます。同時に、BERT はいくつかの特別なフラグを追加しました：

[CLS] マークは最初の文の上部に配置され、BERT によって取得された表現ベクトル C は、後続の分類タスクに使用できます。 [SEP] マークは、入力文 A と B などの 2 つの入力文を区切るために使用されます。[SEP] マークは、文 A と B の後に追加する必要があります。 [MASK] フラグは、文中のいくつかの単語をカバーするために使用されます。単語が [MASK] でカバーされた後、BERT によって出力される [MASK] ベクトルは、単語が何であるかを予測するために使用されます。たとえば、入力サンプルとして「my dog is cute」と「he likes palying」の 2 つの文を指定すると、BERT は ”[CLS] my dog is cute [SEP] he likes play ing [SEP]” に変換されます。 BERT は WordPiece

メソッドを使用して単語をサブワード単位（SubWord）に分割するため、一部の単語はルートを分割します。たとえば、「palying」は「paly」＋「 ing」になります。

BERT は、入力する文を取得した後、文の単語を Embedding に変換します。 Transformer とは異なり、BERT の入力埋め込みは、Token Embedding、Segment Embedding、および位 Position Embedding の 3 つの部分を追加することによって取得されます。



Token Embedding：トレーニングを通じて取得した [CLS] 犬などの単語の埋め込み。

Segment Embedding：各単語が文 A に属するか文 B に属するかを区別するために使用されます。1 つの文のみが入力された場合、EA のみが使用されます。これはトレーニングを通じて学習できます。

Position Embedding：エンコードされた単語が表示される位置。これは、固定式を使用した Transformer による計算とは異なります。BERT の Position Embedding も学習によって取得されます。BERT では、最長の文が 512 であると想定されます。

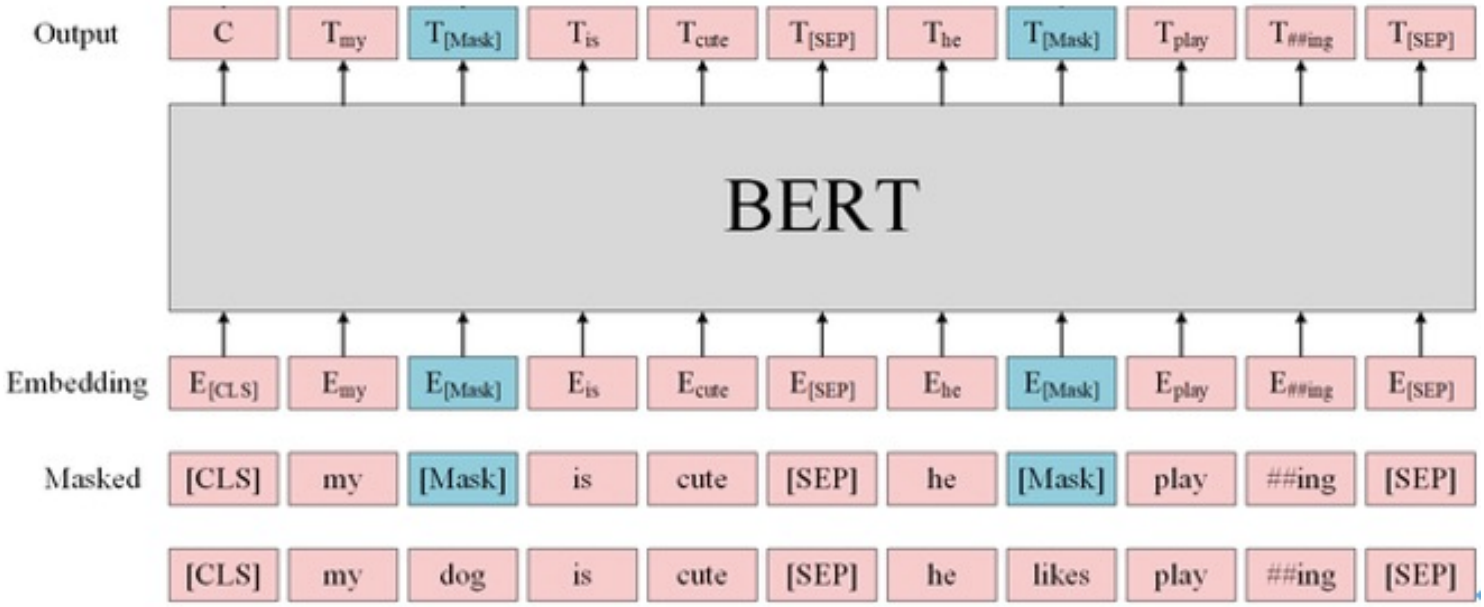
2.6.2.2 BERT の fine-tuning

BERT が文中の単語の埋め込みを入力した後、モデルは事前トレーニングを通じてトレーニングされます。事前トレーニングには 2 つのタスクがあります。

最初のものは MaskedLM です。文中の単語の一部をランダムに [MASK] に置き換え、次に文を BERT に渡して各単語の情報をエンコードし、最後に [MASK] のコーディング情報 T [MASK] を使用して正しい位置を予測します。 語。

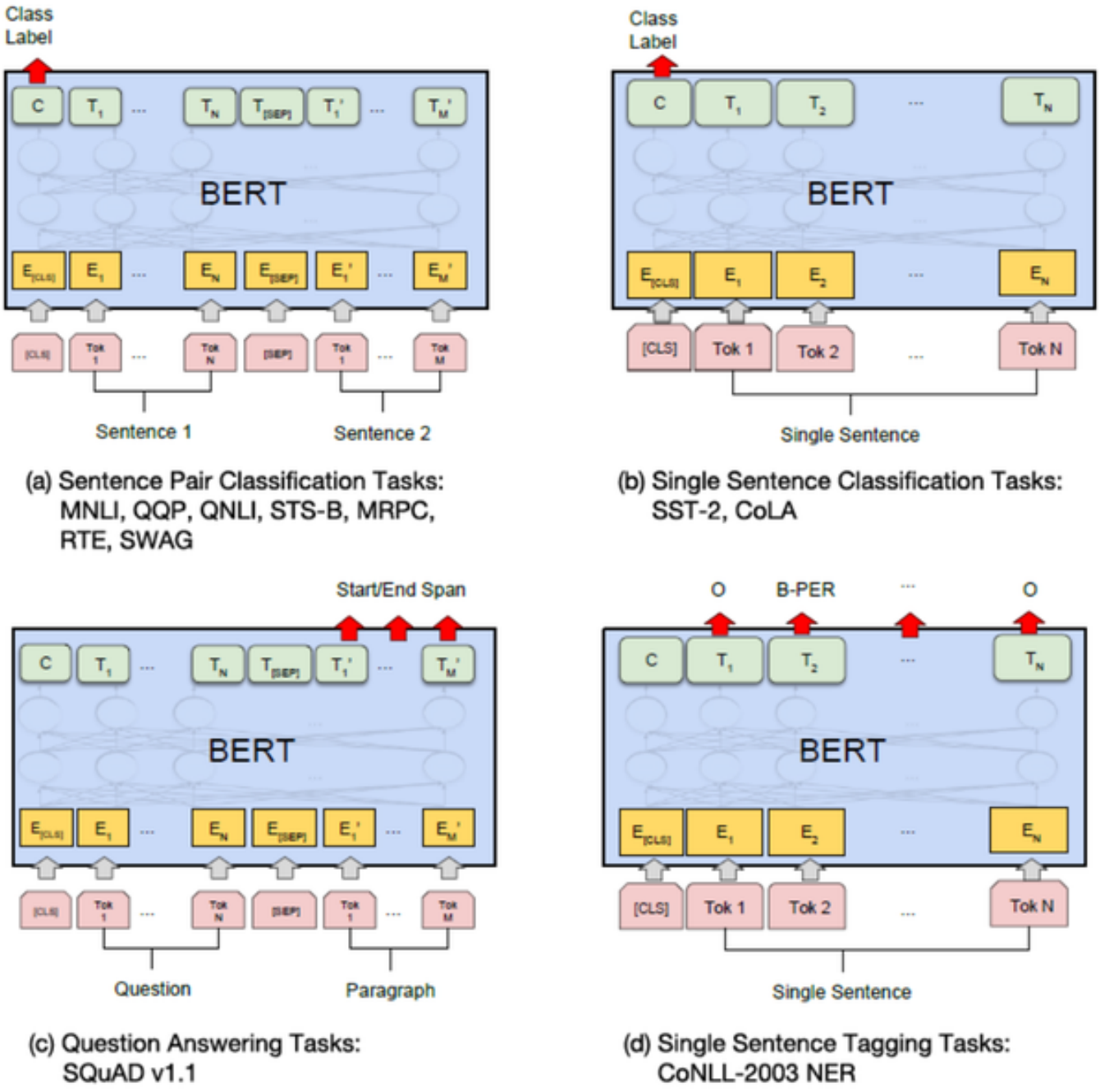
2 つ目は次の文の予測です。BERT に文 A と B を入力し、B が A の次の文であるかどうかを予測し、[CLS] のコーディング情報 C を使用して予測を行います。

BERT の BERT の fine-tuning は、次の図で表示します。



2.6.2.3　BERT は特定の NLP タスクに使用されます

事前トレーニングで得られた BERT モデルは、後で特定の NLP タスクに使用するときに微調整（fine-tuning 段階）できます。BERT モデルは、次の図に示すように、さまざまな異なる NLP タスクに適用できます。



(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

(b) Single Sentence Classification Tasks: SST-2, CoLA

(c) Question Answering Tasks: SQuAD v1.1

(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

自然言語推論（MNLI)、文の意味的同等性（QQP）などの文分類タスクのペア。上の図（a）に示すように、2 つの文を BERT に渡す必要があり、[CLS] の出力値が使用されます。 C は文のペアを分類します。

単一文分類タスク：上図（b）に示すように、文感情分析（SST-2)、文文法が許容できるかどうかの判断（CoLA）など、[SEP] マークなしで 1 文だけ入力する必要があります。【CLS】出力値 C を分類します。

質疑応答タスク：SQuAD v1.1 データセットなど、サンプルは文のペア（Question、Paragraph)、Question は質問、Paragraph は Wikipedia のテキスト、Paragraph には

質問への回答が含まれています。トレーニングの目的は、パラグラフの回答の開始位置（開始、終了）を見つけることです。上の図（c）に示すように、質問と段落が BERT に渡され、BERT は段落のすべての単語の出力に基づいて開始と終了の位置を予測します。

単一文のラベル付けタスク：名前付きエンティティ認識（NER）など、単一文を入力し、個人、組織、場所、その他、またはその他（名前なしエンティティ）に属するかどうかに関係なく、各単語の BERT の出力 T に従って単語のカテゴリを予測します。

2.6.3 fine-tuning mission

事前トレーニングの部分が BERT の焦点です。次に、BERT 事前トレーニングの詳細を理解しましょう。 BERT には、2 つの事前トレーニングタスク MaskedLM と次の文の予測が含まれています。

2.6.3.1 Masked LM

「I / like / Learning / natural / language / processing」という文を例として使用して、以前の言語モデルの事前トレーニング方法を確認してみましょう。 言語モデルをトレーニングする場合、通常、情報漏えいを防ぐためにいくつかのマスク操作を実行する必要があります。情報漏えいとは、「 natural」という言葉を予測するときに「 natural」情報を事前に知っていることを指します。 TransformerEncoder の情報漏えいの理由については後で説明します。

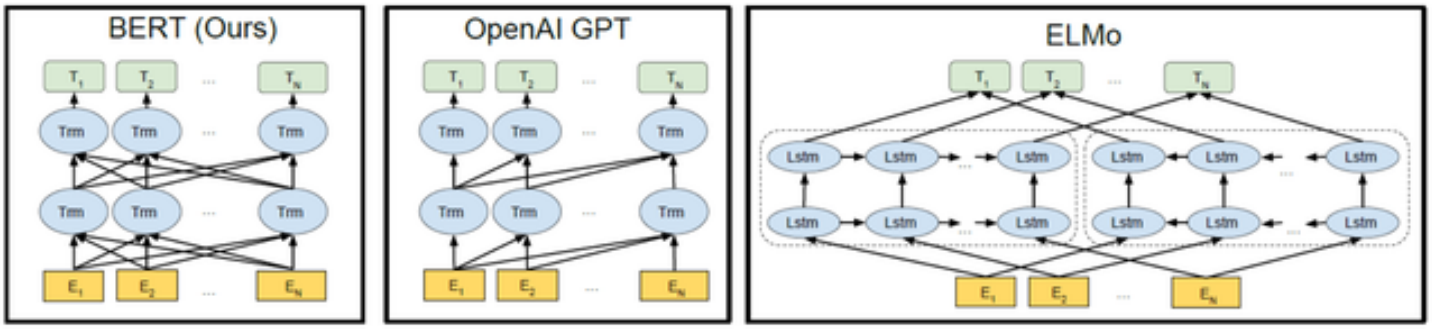
Word2Vec の CBOW：単語 i の上下の情報から単語 i を予測しますが、ワードバッグモデルが使用されており、単語の順序情報は不明です。たとえば、「自然」という言葉を予測する場合、上記の「I / Like / Learn」と次の「Language / Processing」の両方が予測に使用されます。 CBOW は、トレーニング中に「ナチュラル」という単語をマスキングすることと同じです。

ELMo：ELMo はトレーニング中に biLSTM を使用します。「ナチュラル」を予測する場合、フォワード LSTM は「ナチュラル」の後のすべての単語をマスクし、上記の「I / like / learn」予測を使用します。バックワード LSTM は「ナチュラル」をマスクします。前の単語については、以下の「言語/処理」を使用して予測してください。次に、フォワード LSTM とバックワード LSTM の出力が一緒にスプライスされます。したがって、ELMo は、予測にコンテキスト情報を同時に使用するのではなく、予測にコンテキスト情報を分離します。

OpenAI GPT：OpenAI GPT は、Transformer を使用して言語モデルをトレーニン

グする別のアルゴリズムですが、OpenAI GPT は、一方向構造である Transformer のデコーダーを使用します。 「自然」を予測するときは、上記の「I / Like / Learn」のみを使用してください。デコーダーには、現在予測されている単語の後に単語をマスクするマスク操作が含まれています。

次の図は、BERT と ELMo および OpenAIGPT の違いを示しています。



BERT の作成者は、単語を予測する場合、最良の予測を行うには、左（上）と右（下）の情報を同時に使用する必要があると考えています。 左から右と右から左を別々に実行する ELMo のモデルは、浅い双方向モデル（浅い双方向モデル）と呼ばれます。BERT は、TransformerEncoder 構造で深い双方向モデルをトレーニングすることを望んでいます。 トレーニングにはマスク LM 法が提案されています。

マスク LM は、情報漏えいを防ぐために使用されます。たとえば、「ナチュラル」という単語を予測する場合、入力部分の「 natural」マスクを削除しないと、予測出力から「 natural」情報を直接取得できます。

BERT は、トレーニング中に [Mask] 位置の単語のみを予測するため、コンテキスト情報を同時に使用できます。 ただし、その後の使用では、[Mask] という単語は文に表示されないため、モデルのパフォーマンスに影響します。 そのため、トレーニングでは次の戦略を使用します。文中の単語の 15 %がランダムにマスクに選択されます。マスクとして選択された単語のうち、80 %が実際に [Mask] を置換に使用し、10 %が置換せず、残りの 10 %がマスクに使用します。 ランダムな単語の置換。

たとえば、「my dog ishairy」という文では、「hairy」という単語がマスクとして選択され、次のようになります。

80 %の確率で、「my dog ishairy」という文は「mydogis [Mask]」という文に変換されます。

10 %の確率で、「my dog is hairy」という文を変更しないでください。

10 %の確率で、「hairy」という単語を「apple」などの別のランダムな単語に置き換え

ます。 「mydogishairy」という文を「mydogisapple」という文に変換します。

上記は、BERT の最初の事前トレーニングタスク MaskedLM です。

2.6.3.2 Next Sentence Prediction (NSP)

BERT の 2 番目の事前トレーニングタスクは、次の文の予測である次の文の予測（NSP）です。2 つの文 A と B が与えられた場合、文 B が文 A の次の文であるかどうかを予測する必要があります。

BERT がこの事前トレーニングタスクを使用する主な理由は、質問回答（QA）や自然言語推論（NLI）などの多くのダウンストリームタスクでは、2 つの文の関係を理解するためのモデルが必要ですが、言語モデルのトレーニングでは達成できないためです。 この目的のために。

BERT がトレーニング中の場合、接続された 2 つのセンテンス AB を選択する確率は 50 %であり、2 つのセンテンス AB を取得するために接続しないことを選択する確率は 50 %であり、[CLS] フラグの出力 C を介してセンテンス A を予測します。 次の文は B ですか？

入力= [CLS] 私は [マスク] リーグ [SEP] をプレイするのが好きです私の最高の [マスク] は安雄 [SEP] ですカテゴリー= B は A の次の文です

入力= [CLS] プレイしたい [マスク] リーグ [SEP] 今日の天気はとても [マスク] [SEP] カテゴリー= B は A の次の文ではありません

2.6.4 総括する

Masked LM は BERT の事前トレーニングで使用されるため、各バッチでトレーニングされるのは単語の 15 %のみであり、より多くの事前トレーニング手順が必要です。ELMo などのシーケンシャルモデルは、すべての単語を予測します。

BERT は、トランスフォーマーのエンコーダーとマスクされた LM の事前トレーニング方法を使用するため、双方向の予測を実行できます。一方、OpenAI GPT は、トランスフォーマーのデコーダー構造を使用し、デコーダーのマスクを使用します。

第 3 章

SQUAD

3.1 概要

3.1.1 SQuAD とは

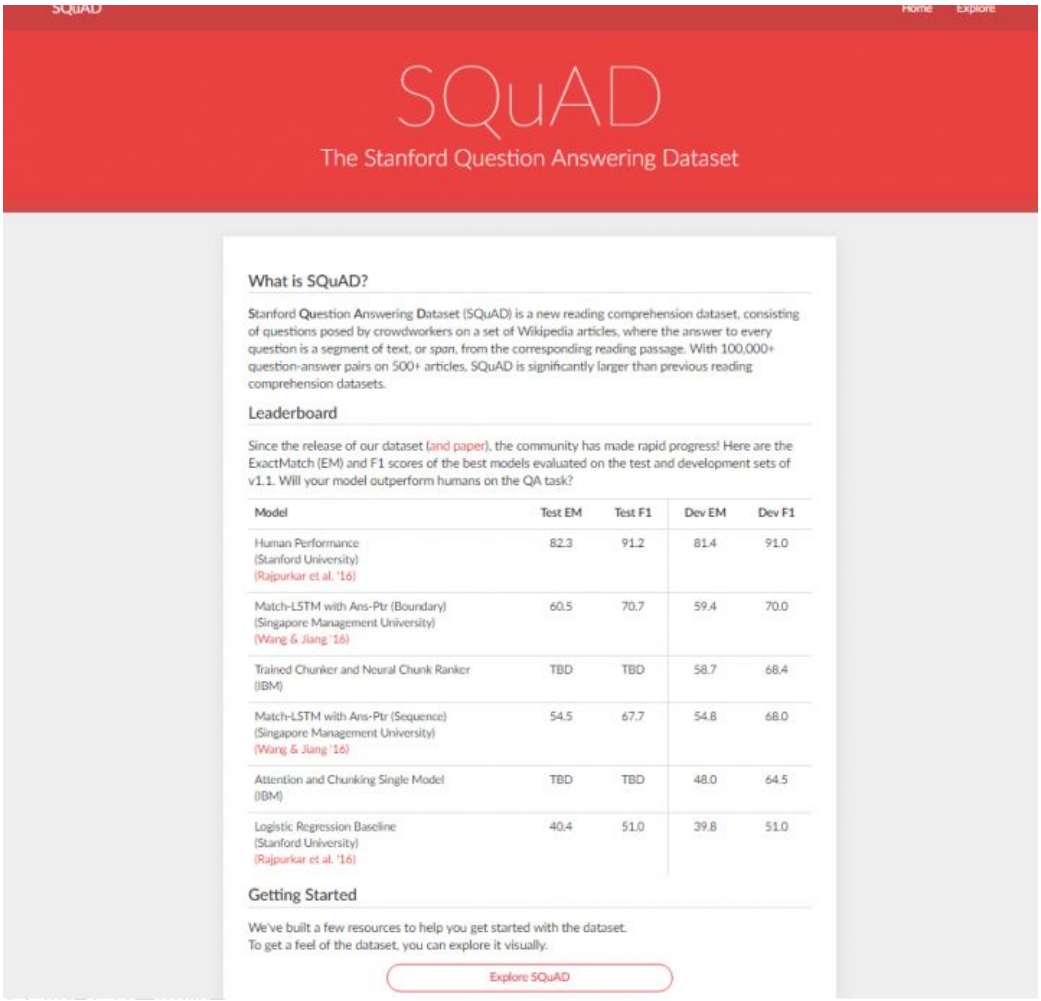
SQuAD は、2016 年にスタンフォード大学によって立ち上げられたデータセットです。これは、読解力データセットです。記事が与えられた場合、対応する質問を準備するには、質問に答えるアルゴリズムが必要です。 このデータセットのすべての記事は Wikipedia から選択されており、データセットの量は他のデータセット（たとえば、WikiQA）の数十倍です。 合計 107,785 の質問と 536 の補足記事があります。 データセットの寄稿者は StanfordPercy Liang などで、Percy Liang は自然言語処理の分野で万能であり、セマンティック解析、QA、最適化などのさまざまな分野で重要な貢献をしてきました。

Dataset	Question source	Formulation	Size
SQuAD	**crowdsourced**	**RC, spans in passage**	**100K**
MCTest (Richardson et al., 2013)	crowdsourced	RC, multiple choice	2640
Algebra (Kushman et al., 2014)	standardized tests	computation	514
Science (Clark and Etzioni, 2016)	standardized tests	reasoning, multiple choice	855
WikiQA (Yang et al., 2015)	query logs	IR, sentence selection	3047
TREC-QA (Voorhees and Tice, 2000)	query logs + editor	IR, free form	1479
CNN/Daily Mail (Hermann et al., 2015)	summary + cloze	RC, fill in single entity	1.4M
CBT (Hill et al., 2015)	cloze	RC, fill in single word	688K

現在の公開データセットを次のように比較します。MCTest、Algebra、Science は 3 つの公開読書理解データセットです。Squad はこれらの 3 つのデータセットの数をはるかに上回っているため、このデータセットのトレーニングが行われます。 大規模で複雑なアルゴリズムが可能になります。 同時に、2 つの有名な質疑応答データセットである WikiQA と TrecQA と比較して、Squad もその数をはるかに上回っています。 CNN メールと CBT は大きいですが、これら 2 つのデータセットは両方ともくり抜かれ、推測された単語であり、本当の意味での質問と回答ではありません。

3.1.2 SQuad データセットを構築する方法は

次に、このデータセットの構成について詳しく紹介します。まず、このデータセットの美しいインターフェイスを感じてみましょう。



図から、検証セットとテストセットのレベルでそれを見ることができます。 テストセットでは、実行可能なプログラムを送信する必要があります。 最後の場所と最初の場所は、それぞれ著者が作成したベースラインと人々が答えることができるレベルです。リリースは 1 か月しかありませんが、シンガポールと IBM の一部の大学はすでにこのタスクを試していることがわかります。 次の図は、このデータセットの例です。最初に記事を与えてから質問を始めます。最初の質問「雨の原因」に対する答えは重力によって引き起こされます。 質問は非常に難しく、推論が必要ですが、答えはまだテキストに表示されています。

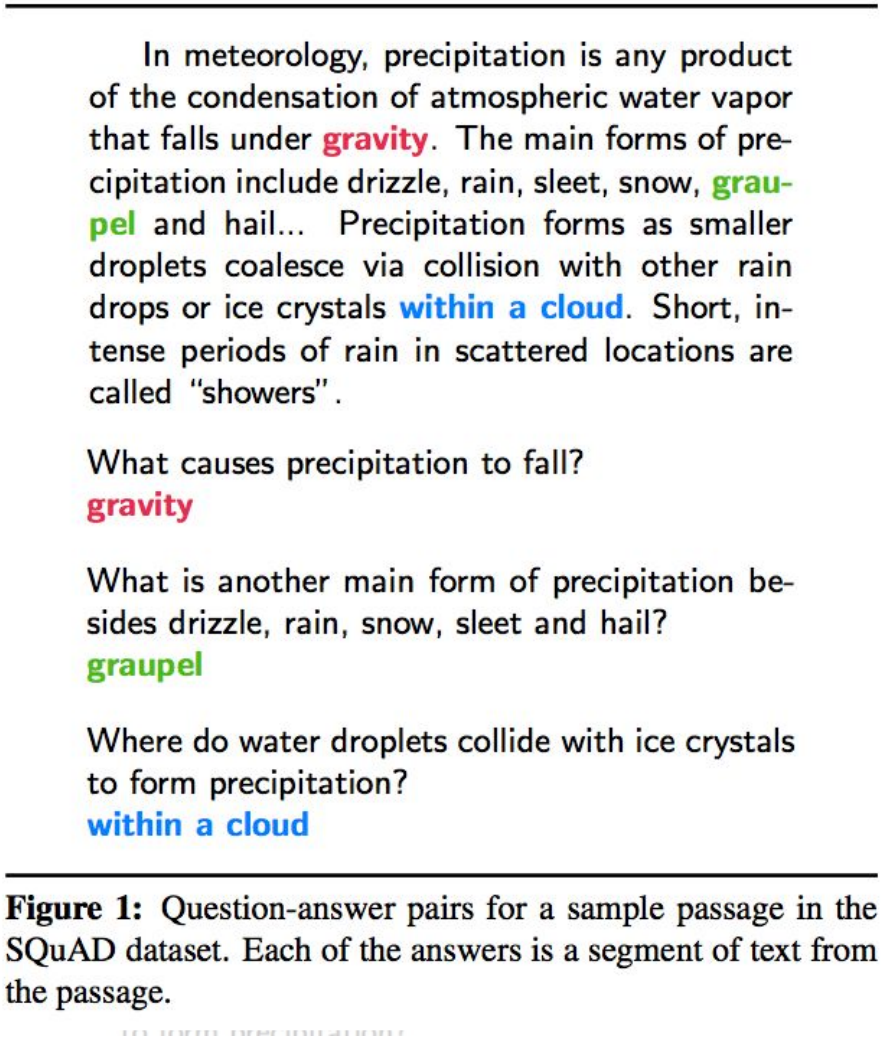
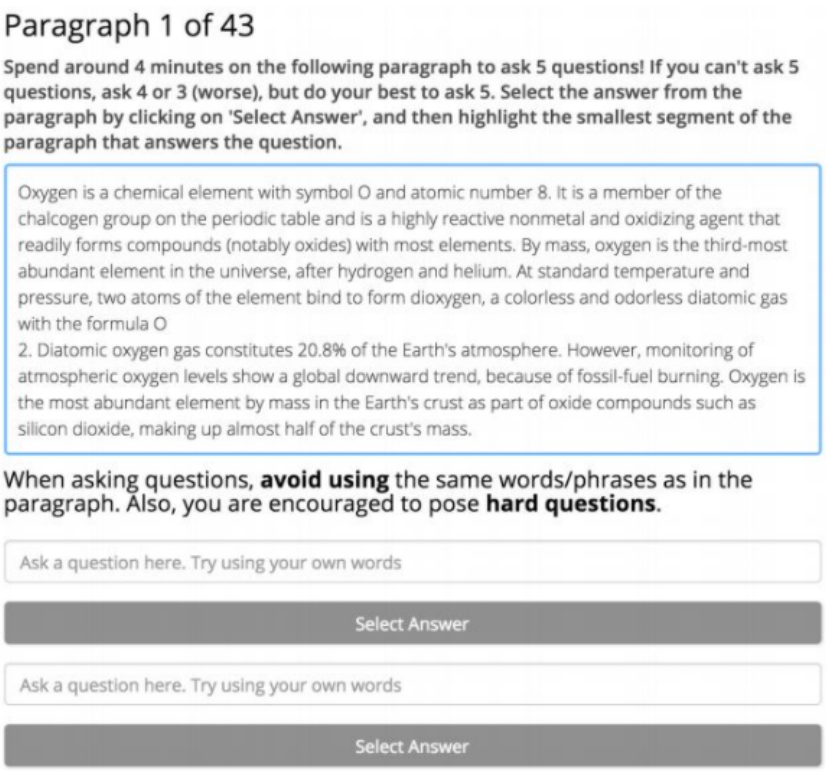


Figure 1: Question-answer pairs for a sample passage in the SQuAD dataset. Each of the answers is a segment of text from the passage.

データセットの具体的な構成は次のとおりです。

1. 記事はランダムなサンプル Wiki であり、合計 536 の Wiki が選択されました。 各ウィキは段落に分割され、23215 の自然な段落になります。 その後、これらの 23215 の自然な段落の理解、または自動質問と回答を読んでください。

2. その後、スタンフォードはクラウドソーシングを使用して、特定の記事に手動でラベルを付け、質問をし、回答を与えました。 彼らはさまざまな人々に 20,000 以上の段落を与え、各段落に 5 つの質問をしました。



3. 他の人に、記事の最短の抜粋でこの質問に答えてもらうように依頼します。そうでない場合、または答えが記事に表示されていない場合、他の人は答えることができません。 検証後、質問の種類の分布は十分に多様であり、推論が必要な質問が多いため、このセットは非常に困難です。 下の図に示すように、著者はデータセットに対する回答のカテゴリ分布をリストしています。日付、人の名前、場所、番号などがすべて同じ比率で含まれていることがわかります。

Answer type	Percentage	Example
Date	8.9%	19 October 1512
Other Numeric	10.9%	12
Person	12.9%	Thomas Coke
Location	4.4%	Germany
Other Entity	15.3%	ABC Sports
Common Noun Phrase	31.8%	property damage
Adjective Phrase	3.9%	second-largest
Verb Phrase	5.5%	returned to Earth
Clause	3.7%	to avoid trivialization
Other	2.7%	quietly

4. このデータセットには 2 つの評価基準があります。1 つ目は F1、2 つ目は EM です。EM は完全一致の略語であり、正確にするには、マシンによって指定されたものと同じである必要があります。 一文字違うとしても間違っているでしょう。 そして、F1 は答え

のフレーズを単語にカットし、リコール、プレシジョン、F1 をその人の答えと一緒にカウントします。つまり、いくつかの単語に一致してもすべてが正しくない場合でも、スコアとしてカウントされます。

5. このデータセットでは、特徴を抽出し、LR アルゴリズムを使用して特徴を組み合わせることでベースラインを作成し、最終的に 40.4em と 51f1 に到達しました。 現在、IBM とシンガポール管理大学の両方が、深層学習モデルを使用してこのアルゴリズムを突破しました。

3.2　現存する主要な SQUAD1.1 と SQUAD2.0 データ

3.2.1　SQUAD1.1 と SQUAD2.0 違う場所

読解システム（モデル）は通常、コンテキストドキュメントで質問に対する正解を見つけることができますが、コンテキストで正解がない場合の質問に対する回答ほど信頼性は高くありません。 既存のデータセットは、回答可能な質問のみに焦点を当てるか、データセットとして簡単に認識できる自動生成された回答不可能な質問を使用します。 これらの欠点を補うために、この記事では、スタンフォード質問と回答データセット（SQuAD）の最新バージョンである SQuAD 2.0 を紹介します。これは、既存の SQuAD の回答可能な質問と、公務員によって書かれた 50,000 を超える回答が難しい質問を統合したものです。 難しい質問は、答えられる質問に似ています。 SQuAD 2.0 のパフォーマンスを向上させるには、システムは可能な場合に質問に回答するだけでなく、段落のコンテキストが回答をサポートしていない場合を判断し、質問への回答を回避する必要があります。SQuAD 2.0 データセットは、自然言語理解タスクにおける既存のモデルへの挑戦です。

データセット：クラウドソースのスタッフは、答えられない質問を書くために Daemo プラットフォームで雇われています。各タスクは、SQuAD1.1 の記事全体で構成されています。記事の各段落について、スタッフは段落だけでは答えられない最大 5 つの質問をすることができます。同時に、段落に表示されているエンティティを参照して、合理的な答えを出すことができます。同時に、SQuAD 1.1 の質問を参考としてスタッフに見せ、回答可能な質問と同様に回答しにくい質問を作成してください。

この論文では、2 つのデータセットに対する 3 つの既存のモデルアーキテクチャのパフォーマンスを評価します。これにより、これらのモデルは回答の分布を学習するだけでなく、質問に回答できない可能性も予測します。質問に答えられない確率が特定のしきい

値を超えるとモデルが予測すると、モデルは答えの分布の学習をあきらめます。次の表は、2 つのデータセット（SQuAD1.1 および SQuAD2.0）での 3 つのモデルのパフォーマンスを示しています。結果は次のとおりです。

最高のパフォーマンスを発揮するモデル（DocQA + ELMo）は、SQuAD 2.0 で人間と 23.2 のギャップがあります。これは、モデルに改善の余地がたくさんあることを意味します。

2 つのデータセットで同じモデルアーキテクチャを使用すると、SQuAD1.1 と比較して、最適なモデルと人の F1 値の間のギャップが、SQuAD 2.0 で大きくなります。これは、SQuAD2.0 が既存のモデルで学習するのが難しいデータであることを示しています。

System	SQuAD test		SQuADRUn dev		SQuADRUn test	
	EM	F1	EM	F1	EM	F1
BNA	68.0	77.3	59.8	62.6	59.2	62.1
DocQA	72.1	81.0	61.9	64.8	59.3	62.3
DocQA + ELMo	**78.6**	**85.8**	**65.1**	**67.6**	**63.4**	**66.3**
Human	82.3	91.2	86.3	89.0	86.9	89.5
Human–Machine Gap	3.7	5.4	**21.2**	**21.4**	**23.5**	**23.2**

SQuAD 2.0 で質問に答えるのが難しいことを証明するために、この記事では TFIDF とルールを使用して、SQuAD 1.1 データセットでいくつかの難しい質問をランダムに生成し、比較のために同じモデルを使用します。 結果は（下の表に示すように）SQuAD 2.0 データセットで最良のモデルが依然として最低であることを示しています。これは、SQuAD2.0 が既存の言語理解モデルにとって難しい課題であることを改めて証明しています。

System	SQuAD + TfIdf		SQuAD + RuleBased		SQuADRUn dev	
	EM	F1	EM	F1	EM	F1
BNA	72.7	76.6	80.1	84.8	59.8	62.6
DocQA	75.6	79.2	80.8	84.8	61.9	64.8
DocQA + ELMo	**79.4**	**83.0**	**85.7**	**89.6**	**65.1**	**67.6**

3.2.2　まとめ

SQuAD 2.0 で質問に答えるのが難しいことを証明するために、この記事では TFIDF とルールを使用して、SQuAD 1.1 データセットでいくつかの難しい質問をランダムに生成し、比較のために同じモデルを使用します。 結果は（下の表に示すように）SQuAD 2.0 データセットで最良のモデルが依然として最低であることを示しています。これは、SQuAD2.0 が既存の言語理解モデルにとって難しい課題であることを改めて証明しています。

第 4 章

実験

4.1　利用したデータセット

今回の実験データは wiki に記録した文章と日本語能力実験（NLPT）N1 の文章を利用しました。文章には物を記録の文章と物を見て感想がてる文章、二つ種類があります。wiki から物を記録の文章を取ります。日本語能力実験から感想文を取りました。記録文は 100 文章を探しました。一つ文章に問題三を聞きます。感想文は 60 文章を探しました。一つ文章に問題三を聞きます。

実験文章の一部を論文に書きます。

日本語の文章：

私たちは、古いものを捨てることが進歩だと信じてきた。伝統的な生き方を尊重し、誇りを持つことより、もっと便利なもの、効率のよいものを生活様式の中に取り入れつづけてきた……。時間換算された仕事をこなすために、遠い職場まで通いつめ、流行とされている服を何度も買い直し、楽しい時間をすごすために、高速道路を車で飛ばし、レストランに通い、ビデオを見て暮らしている。いらなくなったものは、ゴミとしてビニール袋に詰め込んだ家の前で出しておけばそれでよい。一見、豊かそうに見える私たちの暮らしだけれど、果たしてそうなのだろうか。

日本語の問題：

１：私たちはどうな生活が好きですか？

２：いらないものはどうしますか？

英語に翻訳した文章：

We have believed that discarding old things is progress. Rather than respecting

the traditional way of life and taking pride, we have continued to incorporate more convenient and efficient things into our lifestyles... In order to do time-converted work, he travels to distant workplaces, buys fashionable clothes again and again, and to spend a good time, drive down the highway, go to restaurants and watch videos. living. If you don't need anything, just put it out in front of your house, which is packed in a plastic bag as garbage. At first glance, our lives look rich, but is that true?

英語に翻訳した問題：

１：What kind of life do we like?

２：What do you do with unnecessary things?

英語の答え：

１：rich

２：none

英語の答えを日本語に翻訳

１：リッチ

２：none

私が正しいと思う答え：

１；もっと便利なもの、効率のよいものを生活様式の中に取り入れつづけてきた……。

２；ゴミとしてビニール袋に詰め込んだ家の前で出して

日本語の文章：

花の絵を描き始める時、心は画用紙のように真っ白であなりたいと思っている。同じ名前がついている花でも、よく見ると一つ一つが、人間の顔が違うようにそれぞれの表情を持っているからである。また、同じ花でも朝と昼とでは、ほんのわずか色が変わっている場合が多い。

日本語の問題：

１：同じ花ならば色がいつも同じですか？

２：花の絵を描き始める時に心は何をなりたいですか？

英語に翻訳した文章：

When I start drawing flowers, my heart wants to be pure white, like drawing paper. This is because even if the flowers with the same name are looked closely, each one

has its own facial expression so that the human face is different. In addition, the same flower often changes color slightly between morning and day.

英語に翻訳した問題：

１：If the flowers are the same, are the colors always the same?

２：What does your heart want to do when you start drawing flowers?

英語の答え：

１：NONE

２：be pure white

英語の答えを日本語に翻訳

１：NONE

２：真っ白に

私が正しいと思う答え：

１；同じ花でも朝と昼とでは、ほんのわずか色が変わっている場合が多い。

２；心は画用紙のように真っ白であなりたいと思っている。

日本語の文章：

いくら見慣れた花でも、「この花はこういう形をしているんだ。」などと、先入観を持って書き始めると、花にそっぽを向かれてしまうことがある。花屋さんでは、開き過ぎたものは売り物にならないようだけれど、開き過ぎて雌蕊や雄蕊が飛び出したのも、時にはっとするぐらい美しい表情を見せてくれることがある。花びらが一、二枚落ちてしまったのも、虫が食っているのも、いいなあと思う。咲き終わって、花びらが茶色くなってしまったのも。それは決して死んだ花ではなく、一生懸命生きて、今、実を結び始めた最も素晴らしい時期を迎えているのだと思う。

日本語の問題：

１：花びらが落ちると何が起こりますか？

２：花びらが茶色になると私はどう思いますか？

英語に翻訳した文章：

No matter how familiar a flower is, if you start writing with a preconceived notion, such as ”This flower has this shape,” it sometimes turns to the flower. At the florist, it seems that things that are open too much are not for sale, but sometimes when the pistil or stamen pops out due to opening too much, it sometimes shows a stunningly

beautiful expression. I think it's good that one or two petals have fallen off or insects are eating. After the blooming, the petals have turned brown... I think it's not a dead flower, it's the best time to live hard and start to bear fruit.

英語に翻訳した問題：

１：What happens if the petals fall?

２：What do I think if the petals turn brown?

英語の答え：

１：turned brown

２：dead flower

英語の答えを日本語に翻訳

１：褐色になった

２：枯れた花

私が正しいと思う答え：

１；虫が食っている

２；それは決して死んだ花ではなく、一生懸命生きて、今、実を結び始めた最も素晴らしい時期を迎えているのだと思う。

日本語の文章：

風で折れてぶら下がっているのもあれば、病気か何かで歪んで咲いているものある。日向で勢い良く咲いているのもあるが、根元の方では、雨の日に土の跳ね返りを受けて、薄汚くなったのもある。そういうのを見ていると、人間の社会と同じだなあと思ったりする。はしっこい人もいれば、のんきな人のいる。美しい人も、そうでない人も、病気の人も、健康な人も。いろいろな人がいる。

日本語の問題：

１：花の様子を見て私はどう思いますか？

英語に翻訳した文章：

Some are broken by the wind and hanged, while others are distorted due to illness or something. Some of them are blooming vigorously in the sun, but some of them at the roots have become soiled due to the rebound of the soil on a rainy day. When I see that kind of thing, I sometimes think that it is the same as human society. Some are smart and some are carefree. Beautiful people, not so good people, sick people,

healthy people. There are various people.

英語に翻訳した問題：

１：How do you feel when you see the flowers?

英語の答え：

１：it is the same as human society

英語の答えを日本語に翻訳

１：人間社会と同じ

私が正しいと思う答え：

１；人間の社会と同じだなあと思ったりする。

日本語の文章：

しかし、私自身、「あいつは、ああいう奴なんだ。」とほんのわずかしか知らないうちに決め付けてしまうことが、なんと多いのだろう。花の色が一日にして変化するのだから、まして心を持っている人間を見るとき、自分のわずかなばかりで決めつけてしまうのなんて、全く間違っていると思う。

日本語の問題：

１：花の色でも一日中に変わりますのにほんのわずかしか知らない人を決め付けていいですか？

英語に翻訳した文章：

However, how often I personally decide that ”He’s such a guy” while knowing very little. The color of the flower changes in a day, so when you look at a person who has a heart, it is totally wrong to make a decision with just a little of yourself.

英語に翻訳した問題：

１：Even if the color of the flower changes all day long, is it okay to brand someone who knows very little?

英語の答え：

１：NONE

英語の答えを日本語に翻訳

１：NONE

私が正しいと思う答え：

１；全く間違っていると思う

日本語の文章：

山手線は、日本の首都である東京の都心部で環状運転を行い、多くの駅において、都心から各方面へと伸びる JR 東日本（在来線・新幹線）や私鉄各社の放射路線および都心部を走る地下鉄各線に接続している。1 周の長さは 34.5 km、1 周の所要時間は内回り、外回りとも標準で 59 分、朝ラッシュ時は 61 分、夕方ラッシュ時は 60 - 61 分（いずれも大崎駅での停車時間を除く）である。

日本語の問題：

１：山手線はどこで環状運転を行いますか？

２：山手線は 1 周の長さはどのくらいですか？

３；山手線朝ラッシュ時に 1 周の所要時間はどのくらいですかどうか？

英語に翻訳した文章：

The Yamanote Line operates a loop operation in the central part of Tokyo, the capital of Japan, and at many stations, JR East Japan (conventional line / Shinkansen) and private railway companies' radial lines and the central part of the city that extends from the center to each direction It is connected to each subway line that runs. The length of one lap is 34.5 km, the time required for one lap is 59 minutes for both inner and outer laps, 61 minutes during the morning rush hour, 60-61 minutes during the evening rush hour (excluding the stop time at Osaki station)).

英語に翻訳した問題：

１：Where does the Yamanote Line carry out loop driving?

２：How long is one lap of the Yamanote line?

３；How long does one lap take in the morning rush hour on the Yamanote Line?

英語の答え：

１：the central part of tokyo , the capital of japan

２：34 . 5 km

３；60 - 61 minutes

英語の答えを日本語に翻訳

１：日本の首都、東京の中心部

２：34 . 5 km

３；60〜61 分

私が正しいと思う答え：

１；日本の首都である東京の都心部で

２；34.5 km

３；61 分

日本語の文章：

山手線は、日本の文明開化期に私鉄の日本鉄道が当時国内有数の貿易港であった横浜港と関東地方内陸部の各地（埼玉県、群馬県、栃木県）さらに東北地方および北陸地方方面を結ぶ貨物線として建設した、赤羽駅 - 品川駅（および大井町駅）間を結ぶ鉄道路線（当初は品川線と呼称）である。

日本語の問題：

１：当時国内有数の貿易港は誰ですか？

２：関東地方内陸部はどこですか？

３；赤羽駅 - 品川駅間を結ぶ鉄道路線の呼称はなんですか？

英語に翻訳した文章：

The Yamanote Line is the port of Yokohama where private railways of Japan Railway were Japan's leading trading port at the time of the civilization of Japan, and various parts of the inland Kanto region (Saitama prefecture, Gunma prefecture, Tochigi prefecture) and the Tohoku and Hokuriku regions. Built as a freight line that connects, Akabane station-Shinagawa station (and Oimachi station) is a railroad line connecting (initially called the Shinagawa line).

英語に翻訳した問題：

１：Who is the leading trading port in Japan at the time?

２：Where is the inland Kanto region?

３；What is the name of the railway line connecting Akabane Station-Shinagawa Station?

英語の答え：

１：yamanote line

２：saitama prefecture

３；shinagawa line

英語の答えを日本語に翻訳

１：山手線

２：埼玉県

３；品川線

私が正しいと思う答え：

１；横浜港

２；埼玉県、群馬県、栃木県

３；山手線

日本語の文章：

旅客輸送は、開業当初は新橋駅 - 品川駅 - 新宿駅 - 赤羽駅間を往復する列車が 1 日数往復のみ運行され、その後、東京山手の人口増に伴い上野駅を起点として池袋駅、新宿駅、渋谷駅、品川駅、新橋駅を経て東京駅方面に至る環状運転が開始され、その後、上野駅 - 東京駅間の開通により京浜線と東北本線の相互直通運転が開始された時期と同じくして現在の運行形態へと移行、定着した. 明治後期から昭和期にかけての私鉄各社は、地下鉄道であれば東京地下鉄道や東京高速鉄道のように東京 15 区内に路線を敷設できた。

日本語の問題：

１：開業当初に止まる駅の名前はなんですか？

２：なんで環状運転が開始しましたか？

３；いつから私鉄各社は東京 15 区内に路線を敷設できた？

英語に翻訳した文章：

At the beginning of passenger transportation, trains that return from Shimbashi Station-Shinagawa Station-Shinjuku Station-Akabane Station operate only a few round trips a day. At the same time as the loop operation that started from Shibuya Station, Shinagawa Station, Shimbashi Station to Tokyo Station started, and then the mutual direct operation of Keihin Line and Tohoku Main Line started due to the opening of Ueno Station-Tokyo Station. From the latter half of the Meiji era to the Showa era, private railway companies were able to lay routes within the 15 wards of Tokyo like Tokyo Subway or Tokyo High Speed Rail.

英語に翻訳した問題：

１：What is the name of the station that stops when it opens?

２：Why did the loop operation start?

３；Since when have private railway companies been able to lay routes within the 15 wards of Tokyo?

英語の答え：

１：ueno station - tokyo station

２：the opening of ueno station - tokyo station

３；from the latter half of the meiji era to the showa era

英語の答えを日本語に翻訳

１：上野駅-東京駅

２：上野駅の開業-東京駅

３；明治後期から昭和まで

私が正しいと思う答え：

１；新橋駅 - 品川駅 - 新宿駅 - 赤羽駅

２；東京山手の人口増

３；明治後期から昭和期にかけて

4.2　実験結果

記録文 300 問と感想文 150 問の正確率から見ると感想文より記録文の方が高いです。83 パーセントになりました。5W2H の問題ならばほぼ正しいですが論理的に少し考えて答えられる問題でしたらほぼ当ていません。例えば例えば文章は：朝ラッシュ時の混雑率が 250 パーセントを越えていた時期もあったが、地下鉄網の発達や並行する山手貨物線の旅客化、山手線自身の 6 ドア車導入による 11 両化、上野東京ラインの開業等、新線開業が相次いだことにより混雑は大幅に緩和された。2015 年度以降の朝ラッシュ時混雑率は外回り、内回りとも 170 パーセントを下回っている。問題は 2 つがあります。一つ：2015 年以前の混雑率は？ 2 つ：2015 年以後の混雑率は？答え（両方）：170 パーセントでも少し考えると人ならば 250 パーセントと 170 パーセントを答える。後は場所の名前の発音が一緒ならば答えられません。例えば文章には日立を書いていますが、英語に翻訳すると HITACHI になります、英語の答えは HITACHI ですが日本語に翻訳するときに常陸になる。感想文の場合の正確率は 23 パーセントしかないです。

第 5 章

考察

5.1　今後の課題

SQUAD には 1.1 バージョンと 2.0 バージョン二つがあります.SQuAD 1.1 の 100,000 の質問と回答と比較して、SQuAD 2.0 はさらに 50,000 の人間が書いた質問を追加します-そして、質問には対応する回答がない場合があります。今後 SQUAD2.0 を使用して、SQUAD1.1 では解決できない問題を解決できるかどうかを確認します.

第 6 章

結論

人々が段落（ニュースなど）を読むとき、彼らはいくつかの簡単な質問をするかもしれません、そしてこれらの質問への答えはテキストで直接見つけることができます。 ただし、直接答えることができない関連する質問をする可能性が高くなります。 強力なデータセットは、両方を同時にモデル化する必要があります。 データセットで回答された質問のみがある場合、モデルは常にテキストで見つけることができるいくつかの回答を与えることを学習します。 誰かがテキストに直接表示されていない質問をすると、当然間違った答えが返されます。

SQuAD には多くの問題がありますが、実用的な記事は少なく、短いため、データセット全体の語彙やトピックの多様性が制限されます。

したがって、SQuAD で適切に機能するモデルをより複雑な問題に使用する場合、スケーラビリティと適用性の両方に問題があります。

ですがしかし、この論文の目的のためだけに、翻訳ソフトウェアの使用は、日本語のデータセットがないという問題を補うことができます。

謝辞

中間審査および最終審査では、各位教授と准教授より、貴重なご指導とご助言を賜りました。感謝申し上げます。

本研究を進めるに当たり、指導教官の新納教授には研究の着想から、調査、論文執筆まで多くのご指導をいただきました。心から感謝申し上げます。

最後に、所属する自然言語処理ゼミのみなさまには多くのご支援をいただきました。お礼申し上げます。

厚く御礼を申し上げ、感謝する次第です。

参考文献

word2vec:Le Q, Mikolov T. Distributed representations of sentences and documents[C]// International conference on machine learning. 2014: 1188-1196.

doc2vec:Quoc Le.Tomas Mikolov.Distributed Representations of Sentences and Documents// Google Inc, 1600 Amphitheatre Parkway, Mountain View, CA 94043

ELMO:Matthew E. Peters,Mark Neumann †,Mohit Iyyer,Matt Gardner. Deep contextualized word representations// arXiv:1802.05365v2 [cs.CL] 22 Mar 2018

bert: Peters, M. E. et al. Deep contextualized word representations. naacl (2018).

bert: Peters, Radford, A.Salimans, T. Improving Language Understanding by Generative Pre-Training. (2018).

squad1.1:BERT: Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova.Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805