単一回答モデルによる複数回答機械読解

高橋 拓誠　谷口 元樹　谷口 友紀　大熊 智子
富士ゼロックス株式会社
{takahashi.takumi, motoki.taniguchi, taniguchi.tomoki, ohkuma.tomoko}@fujixerox.co.jp

1 はじめに

機械読解タスクは，与えられたテキストを読み解き質問に回答することを目的としており，モデルの読解能力をベンチマークする上でも重要なタスクとして注目されている．SQuAD[7] のような抽出型機械読解では，質問および関連テキスト (コンテキスト) が与えられたとき，最も適切な回答をコンテキストの中から抽出する．いくつかの抽出型機械読解 [7, 6] では，既に多くのモデルによって人間の読解能力を上回る性能を実現することが報告されている [1, 9, 5]．しかし，従来の抽出型機械読解は，図 1 の中段に示すような，質問に対する正解が一つの回答範囲に基づく質問応答 (以下，単一回答 QA) に限定しており，複数の回答が同時に成立するような質問応答には対応できなかった．

DROP[2] では，複数の範囲における回答抽出が必要な質問応答 (以下，複数回答 QA) が新たに追加された．図 1 に示すように，複数回答 QA では与えられた質問とコンテキストに対して，正解が一つの回答範囲に限定されない点で単一回答 QA と異なる．このような複数の範囲における回答を抽出するために，適切な回答を過不足なく抽出するためのモデルがいくつか提案されている [4, 3]．しかしながら，これらのモデルはいずれも複数回答 QA をもつ機械読解のデータを用いて学習することを前提としており，このような複雑なデータを常に用意することは現実的ではない．

単一回答のみ可能とするモデルを適用した場合，モデルにより推定される回答は常に一つに限定される．図 1 に示すように，いくつかの単一回答モデルを複数回答 QA に適用したところ，各モデルの推定する回答が分散することが分かった．同様に，単一/複数回答 QA における各モデルの回答の一致度を比較すると，単一回答 QA のほうが回答の一致度が高い傾向にあることが分かる (図 2)．これは，各単一回答モデルは最適と考える回答を一つしか出力しない制約に起因する．したがって，複数の回答がコンテキストに散在する場合，各モデルの回答はしばしば一致しない．

本研究では，複数の単一回答モデルの回答が複数回答 QA において特に一致しない性質を利用する．具体的には，**単一回答しかできないモデル**を複数組み合わせることで，複数回答可能なモデルを実現する．さらに，本研究では SQuAD のような**単純な単一回答のみ対象とした抽出型機械読解のデータセットだけ**を用いて，複数回答可能なモデルを実現する．本研究の貢献は以下の三点である．

- 単一回答しかできないモデルを複数組み合わせることで，複数回答可能なモデルを実現する．ここで，各単一回答モデルは，SQuAD のような単純な単一回答 QA のみ用いて学習される．
- 単一回答モデルと比較して，提案手法により複数回答 QA の性能が大幅に向上することを示す．

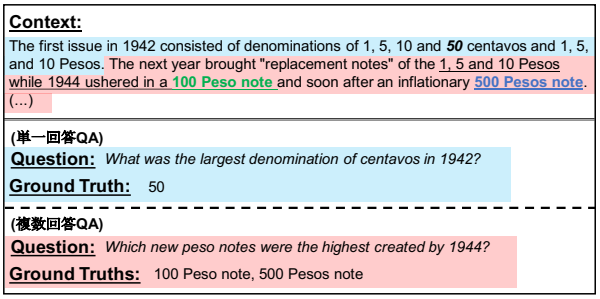

図 1: 単一/複数回答 QA の例．複数回答 QA(下段) に対して，事前に学習した 10 個の単一回答モデル ($\text{BERT}_{\text{LARGE}}$) により抽出した回答をコンテキスト中の下線で示す．太字で着色された箇所は，複数のモデルで一致した回答を表す．

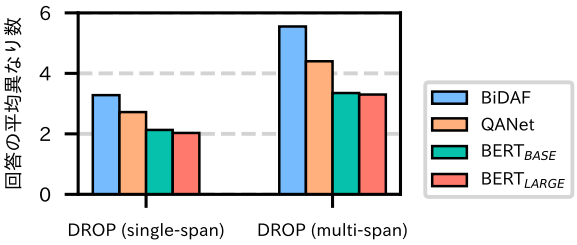

図 2: 複数の単一回答モデルを適用した場合の回答の異なり数 (平均)．各モデルの推定した回答が完全一致した場合 (EM=1)，異なり数=1 とみなす．事前に学習した 10 個の単一回答モデル (4.1 節に記述) を利用した．

- 単一回答モデルの数および多様性が，回答の予測に与える影響について評価する．多様な種類の単一回答モデルを組み合わせることは，特に複数回答 QA の正解率を向上させるために重要であることを示す．

2 関連研究

抽出型機械読解：典型的な抽出型機械読解のデータセットである SQuAD[7] では，質問に対して与えられたコンテキストの中から最も適切な回答を一つ抽出することが目的とされており，既に多くのモデルによって人間の読解能力を上回ることが報告されている [1, 9, 5]．その後，SQuAD 2.0[6] では質問に対する回答がコンテキストに存在しない場合に回答不可と回答するための質問が新しく追加された．

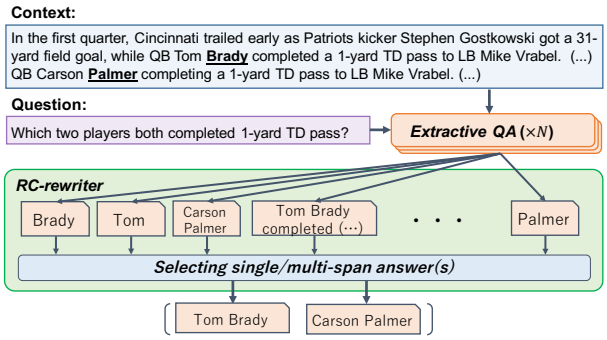

図 3: 提案モデルの概要図．**Extractive QA** では，単一回答 QA を用いて事前に学習した N 個の異なる抽出型機械読解のモデルを使用する．

DROP[2] では，従来取り組まれていなかった複数の回答範囲の抽出 (複数回答 QA) を考慮するための質問が追加された．SQuAD 2.0 は単一回答/回答なしの 2 パターンである一方，DROP は一つ以上の任意の数の回答を求められることから，DROP のほうがより多様な回答パターンを要求されるデータセットであるといえる．

複数回答モデル: Hu ら [4] は，質問に対する回答がコンテキスト中に何個あるか予測し，予測された回答数に基づき回答範囲を繰り返し抽出するモデルにより，DROP の複数回答 QA を回答可能であることを示した．Efrat ら [3] は，回答抽出を系列ラベリング問題と見立て，各トークンに BIO タグを付与する Multi-span Head を提案し，DROP の複数回答 QA において state-of-the-art であることを示した．これらのモデルは複数回答が可能であるものの，学習のために複数回答 QA を含む機械読解のデータをあらかじめ十分に用意しておく必要がある．本研究は，**単一回答のみ対象とした抽出型機械読解のデータセットだけ**を必要とする点と，**単一回答しかできない単純なモデル**を複数組み合わせることで，複数回答 QA を回答可能とする点で異なる．

3 単一回答モデルによる複数回答機械読解

本研究では，単一回答しかできない複数のモデルの回答が，複数回答 QA において特に一致しない性質を利用する．図 3 に提案モデルの概要を示す．**Extractive QA** では，N 個の異なる単一回答モデル ($M = \{m_1, ..., m_N\}$) をあらかじめ学習させておく．**Extractive QA** より出力された N 個の回答のうち，最終的にどの回答を採用すべきかを **RC-rewriter** で選択する．

3.1 Extractive QA

はじめに，単一回答 QA のみから構成される抽出型機械読解のデータセットを用いて，N 個の異なる単一回答モデルをあらかじめ学習させる．本研究では，**Extractive QA** として BERT[1] を使用する．

BERT に基づくモデルでは，質問 Q およびコンテキスト C が与えられたとき，$x = [\texttt{CLS}]Q[\texttt{SEP}]C[\texttt{SEP}]$ を BERT への入力とする．[CLS] はペア文全体の表現を表すトークン，[SEP] は入力ペアの分割を明示するためのトークンを表す．したがって，x を BERT へ入力することで，長さ L のトークン系列に対する d 次元の分散表現 $\boldsymbol{H} \in \mathbb{R}^{d \times L}$ が得られる．

回答範囲の抽出のため，2 種類の全結合層を用いて回答の開始点と終了点を予測するための確率分布を以下の式に基づき計算する．なお，$\boldsymbol{W}_s$，$\boldsymbol{W}_e$ は重み行列，$\boldsymbol{b}_s$，$\boldsymbol{b}_e$ はバイアス項である．

$$p_{start} = \mathrm{softmax}(\boldsymbol{W}_s\boldsymbol{H} + \boldsymbol{b}_s), \quad (1)$$

$$p_{end} = \mathrm{softmax}(\boldsymbol{W}_e\boldsymbol{H} + \boldsymbol{b}_e) \quad (2)$$

最終的に，単一回答モデル m_i は，p_{start} と p_{end} が最大となるトークンを回答の開始点および終了点として回答 a_{m_i} を抽出する．このように回答の抽出は確率分布の最大値に基づくため，単一回答モデル m_i はコンテキスト C に潜在する回答の個数にかかわらず，回答を一つだけ抽出する．

3.2 RC-rewriter

Extractive QA(3.1 節) では，N 個の異なる単一回答モデル ($M = \{m_1, ..., m_N\}$) に基づき，独立な N 個の回答候補 ($A = \{a_{m_1}, ..., a_{m_N}\}$) を得た．**RC-rewriter** を用いた回答選択では，コンテキスト中の**特定のトークンに回答範囲が集中すれば対象トークンが単一回答**であり，**複数のトークンに回答範囲が分散すれば複数回答**になると仮定した回答選択を行う．以降では，回答選択するための具体的な方法について説明する．

最初に各回答の一致度を計算するために，コンテキスト C に含まれているすべてのトークン $T = \{t_1, ..., t_{|C|}\}$ に対して，各トークンが回答候補 A に出現した頻度を計算する．トークン t_i が回答候補 A に出現した頻度 f_i は，

$$f_i = \sum_{a \in A} match(t_i, a), \quad (3)$$

$$match(t_i, a) = \begin{cases} 1 & (t_i \in a) \\ 0 & (t_i \notin a) \end{cases} \quad (4)$$

のように計算される．この計算をコンテキスト中のすべてのトークンに対して適用することで，トークン単位で回答候補 A に出現した頻度を表す系列 $F = \{f_1, ..., f_{|C|}\}$ を得る．

続いて，各トークンの回答候補における出現頻度 F を参照しながら，閾値 u, l に基づき最終的な回答 A^* を得る (ただし，$u \geq 0.5 \times N$ とする)．具体的には，下記の操作に基づきコンテキスト中のすべてのトークンを選別する．

操作 1 $f_i \geq u$ を満たす場合は，t_i を単一回答のトークンとみなし A^* に追加 (**単一回答 QA**)．

操作 2 $u > f_i \geq l$ を満たす場合は，t_i を複数回答のトークンとみなし A^* に追加 (**複数回答 QA**)．

操作 3 $l > f_i$ を満たす場合は，t_i を回答のトークンでないとみなす．

なお，**操作 1** または**操作 2** において，連続する複数のトークン ($T' = \{t_i, t_{i+1}, ...\}$) に対して同じ操作が繰り返される場合は，各トークンを個別に A^* に追加するのではなく，系列の長さが最大となる T' を A^* に追加する．また，t_j に対して**操作 1** を実行した後，t_{j+1} に対する処理が**操作 1** 以外であった場合は，**操作 1** で得られたトークン系列 T' を単一回答として A^* に追加し，回答の選択を終了する ($T' = \{t_j\}$ の場合は，t_j のみ A^* に追加する)．

RC-rewriter では，**Extractive QA** が抽出した回答候補をトークン単位の頻度に基づき再構成するため，図 3 のように “Tom” と “Brady” という別々に抽出されたトークンから “Tom Brady” といった回答を構成可能である．

dataset	train	dev.
DROP(single-span)	23,068	2,749
DROP(multi-span)	-	553

表 1: 実験に使用したデータセットの統計量.

	♣	♢	♡	♠
バッチサイズ	60	32	12	24
エポック数	50	50	10	5
学習率	$1e^{-3}$	$1e^{-3}$	$3e^{-5}$	$3e^{-5}$
最大トークン数	512	400	512	512

表 2: ハイパーパラメータの設定．♣ は BiDAF，♢ は QANet，♡ は $\text{BERT}_{\text{BASE}}$，♠ は $\text{BERT}_{\text{LARGE}}$ を示す.

4 評価実験

4.1 実験設定

データセット: DROP[2] に含まれる抽出型機械読解のうち，回答が一つのものを単一回答 QA(single-span)，回答が二つ以上のものを複数回答 QA(multi-span) として使用した．なお，本実験では DROP の複数回答 QA は学習データとして使用せず，評価時のみ使用する．したがって，学習時には DROP(single-span) のみ使用した．表 1 に実験に使用した単一/複数回答 QA の統計量を示す.

Extractive QA: 単一回答の抽出型機械読解のモデルとして，BiDAF[8](♣)，QANet[10](♢)，BERT[1] を使用した．BERT は事前学習済みモデル[1]の $\text{BERT}_{\text{BASE}}$(♡) および $\text{BERT}_{\text{LARGE}}$(♠) を用いた．**Extractive QA** として使用するために，異なる seed を設定した 20 個のモデルを事前に学習した．学習時には Adam による最適化を行った．表 2 に各モデルで使用したハイパーパラメータを示す.

RC-rewriter: 回答選択する際の各トークンの頻度に対する閾値を $u = 0.9 \times N$，$l = 0.2 \times N$ に設定した．なお，N は **Extractive QA** で用いるモデルの数に対応する (本実験では，$N = 20$).

比較手法: Extractive QA として用意したモデルと同様の条件で学習した単一回答モデル (BiDAF(♣)，QANet(♢)，$\text{BERT}_{\text{BASE}}$(♡)，$\text{BERT}_{\text{LARGE}}$(♠)) をベースラインとして用意した．さらに，単一回答モデルにおける性能の上限値として $\text{Oracle}_{\text{single}}$ を用意した．$\text{Oracle}_{\text{single}}$ では，各質問に付与された複数の真の回答のうち，F1 の値が最も高くなる回答を一つ抽出する.

評価指標: Dua ら [2] に倣い，DROP の評価用に変更した EM(Exact Match) および F1(macro-averaged) を使用した．EM では，正解に含まれるすべての回答とモデルの予測した回答が完全に一致することで評価値が 1 となる．また，F1 は正解と予測回答でアライメントを取り，すべての組み合わせに対して F1 を計算したのち規格化を行うため，複数の回答を過不足なく得ることで評価値が最大化される.

4.2 単一/複数回答 QA に関する実験結果

表 3 に DROP の複数回答 QA(multi-span) および単一回答 QA(single-span) の実験結果を示す.

[1]本実験では，BERT-Base, Uncased/BERT-Large, Uncased を使用した．`https://github.com/google-research/bert`

Models	multi-span EM	multi-span F1	single-span EM	single-span F1
BiDAF(♣)	0	15.9	49.6	55.7
QANet(♢)	0	18.4	53.6	59.8
$\text{BERT}_{\text{BASE}}$(♡)	0	19.1	53.5	60.0
$\text{BERT}_{\text{LARGE}}$(♠)	0	23.4	**57.8**	**65.0**
♣(×20) + **RC-rewriter**	3.25	27.2	37.7	55.4
♢(×20) + **RC-rewriter**	3.80	28.2	42.5	58.1
♡(×20) + **RC-rewriter**	2.89	28.8	42.8	59.1
♠(×20) + **RC-rewriter**	**5.61**	**33.2**	43.9	62.7
$\text{Oracle}_{\text{single}}$	0	42.0	100	100

表 3: DROP(dev.) の複数回答 QA(multi-span) および単一回答 QA(single-span) における実験結果.

複数回答 QA の評価結果: ベースラインの単一回答モデルと **RC-rewriter** を用いた提案手法を比較すると，すべてのモデルで F1 が約 10pt 向上することを確認した．さらに，単一回答モデルは回答を一つしか抽出しないため，$\text{Oracle}_{\text{single}}$ を含むすべてのモデルで EM=0 となる．一方で，提案手法は最大 5.61%の複数回答 QA でコンテキスト中のすべての回答を正しく抽出できることを示した．提案手法では，特定の単一回答モデルに依存することなく，複数回答 QA における性能を大幅に向上させることが可能である.

単一回答 QA の評価結果: ベースラインの単一回答モデルと比較して，**RC-rewriter** を用いた提案手法では F1 でほとんど同程度の性能となる一方で，EM では大幅に性能低下してしまうことが分かった．EM が大幅に低下した要因としては，**RC-rewriter** により正解を抽出したものの，同時に余分な回答も抽出したためであると分かった．EM と F1 における性能差の要因分析については，5 節で議論する.

4.3 単一回答モデルの数/多様性に関する実験結果

本節では，**Extractive QA** として用いる単一回答モデルの数および種類を変化させた場合の性能を評価する.

単一回答モデルの数に応じた性能の評価: 図 4 に結果を示す．単一回答モデルの数を最小 (N=1) にした条件と比較して，モデルの数を増やすことで性能向上することが確認できる．一方で，モデルの数が増えすぎると性能が低下する傾向にあることが分かった．したがって，ある程度の数の単一回答モデルを用意することは，複数回答 QA を解くために必要であるものの，**モデル数を際限なく増やすことによる性能向上は見込めない**といえる.

異なる単一回答モデルの組み合わせによる性能の評価: 表 4 に結果を示す．比較するモデル間で単一回答モデルの数を統一するため，組み合わせモデル (♣ + ♢ + ♡ + ♠ + **RC-rewriter**) の各モデルの数は 5 個とし，合計 20 個の単一回答モデルを用いた．multi-span において，組み合わせモデルを ♠(×20) + **RC-rewriter** と比較した場合，EM で 0.54pt，F1 で 2.9pt 性能向上することを確認した．さらに，表 3 に示した BiDAF(♣) と比較すると，F1 で 20pt 以上性能向上することが分かる．一方で，single-span においては，**RC-rewriter** を用いた各モデルの性能の平均と同程度の評価値になることが示された．したがって，**Extractive QA** として用いる単一回答モデルは，**最も性能の高い単一回答モデルのみ使用するのではなく，多様な種類の単一回答モデルを組み合わせる**ことが，複数回答 QA の正解率を向上させる上で重要であるといえる.

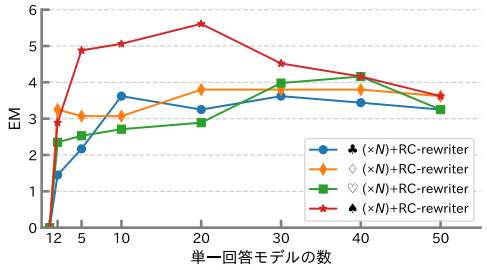

図 4: multi-span におけるモデル数に応じた性能の変化.

Models	multi-span EM	multi-span F1	single-span EM	single-span F1
♠(×20) + **RC-rewriter**	5.61	33.2	**43.9**	**62.7**
♣ + ♢ + ♡ + ♠ + **RC-rewriter**	**6.15**	**36.1**	40.0	59.7

表 4: multi/single-span における異なる単一回答モデルの組み合わせによる性能評価．異なる構造の単一回答モデルを組み合わせた条件では，各モデルを 5 個ずつ使用した．

5 エラー分析

4.2 節の結果を受けて，提案手法の single-span における EM と F1 の性能差について考察する．♠(×20) + **RC-rewriter** の予測回答を分析対象とした．まず，ランダムサンプリングした 200 件の single-span のうち，完全一致 (EM) で正解した事例は 88 件 (44%) あり，表 3 の結果と概ね一致した．続いて，完全一致で不正解となった事例 112 件を対象に，**EM を低下させる要因**という観点から，提案手法が誤った事例を以下の (a)-(d) のカテゴリに基づいて人手による分類を行った．

(a) 複数回答として回答した誤り： 112 件中，35 件 (31.3%) で回答数に起因する誤りが発見された．これは図 5 の (a) に示すように，正解を抽出したものの余分な回答も同時に抽出してしまったことで，不正解 (EM=0) となった事例が該当する．さらに，$\text{BERT}_{\text{LARGE}}$(♠) による予測回答は，35 件中 21 件が正解 (EM=1) となったことから，当該カテゴリによる誤りは **RC-rewriter** によるところが大きいと考えられる．

(b) 意味的に同じ回答 (表記ゆれ) による誤り： 112 件中，14 件 (12.5%) で表記ゆれに起因する誤りが発見された．これは図 5 の (b) に示すように，抽出した範囲は異なるものの，正解と予測回答の意味は全く同じ事例が該当する．また，$\text{BERT}_{\text{LARGE}}$(♠) による予測結果は全て不正解 (EM=0) であったことから，当該カテゴリによる誤りは単一回答モデルにおいても同様に発生する現象であることが分かる．

(c) 全く異なる範囲を抽出したことによる誤り： 112 件中，45 件 (40.2%) で全く異なる範囲における誤りが発見された．$\text{BERT}_{\text{LARGE}}$(♠) による予測結果では，45 件中 5 件が正解 (EM=1) となる事例であった．したがって，当該カテゴリによる誤りは，**RC-rewriter** の導入の有無にかかわらず発生する問題であるといえる．

(d) 複合的な要因による誤り： 112 件中，18 件 (16.1%) で (a)-(c) の複数の要因による誤りが発見された．具体的には，“複数回答した上で，全ての範囲が異なっていた場合” や “複数回答した中に正解を含まないが，表記ゆれレベルの誤りを含む場合” が該当する．

(a) 複数回答として回答した誤り：

Context:
(...) 29.3 % were of united states , 22.2 % germans , 12.1 % english people and 10.9 % irish people ancestry according to 2000 United States Census.

Question: *Which ancestral group is smaller: Irish or English?*
Ground Truth: irish

(b) 意味的に同じ回答（表記ゆれ）による誤り：

Context:
(...) Oakland would get the early lead in the first quarter as quarterback JaMarcus Russell completed a 20-yard touchdown pass to rookie wide receiver Chaz Schilens . (...)

Question: *Who threw the longest pass?*
Ground Truth: Russell

図 5: 提案手法により誤った事例．コンテキスト中の下線付きのテキスト (青色) は，モデルの予測した回答範囲を表す．

6 おわりに

本研究では，**単一回答のみ対象とした抽出型機械読解のデータセットだけを用いて，単一回答しかできないモデル**を複数組み合わせることで，複数回答可能なモデルを実現した．実験では，提案手法により複数回答 QA の性能が大幅に向上することを示した．さらに，**多様な種類の単一回答モデルを組み合わせる**ことは，複数回答 QA の正解率を向上させるために重要であることを示した．

今後の課題として，特に単一回答 QA における EM の向上が挙げられる．このため，経験的に決定した **RC-rewriter** の閾値 u, l を最適化する方法を検討する予定である．

謝辞： 本研究では，産総研の AI 橋渡しクラウド (ABCI) を利用した．

参考文献

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

[2] Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *NAACL*, 2019.

[3] Avia Efrat, Elad Segal, and Mor Shoham. Tag-based multi-span extraction in reading comprehension. *arXiv preprint arXiv:1909.13375*, 2019.

[4] Minghao Hu, Yuxing Peng, Zhen Huang, and Dongsheng Li. A multi-type multi-span network for reading comprehension that requires discrete reasoning. In *EMNLP-IJCNLP*, 2019.

[5] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[6] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *ACL*, 2018.

[7] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *EMNLP*, 2016.

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

[9] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019.

[10] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. Qanet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541*, 2018.