

JParaCrawl における分割学習の提案

村上仁一¹

¹ 鳥取大学工学部 murakami@tottori-u.ac.jp

概要

JParaCrawl は、約 3000 万文を超える日英対訳データである [1]。規模が非常に大きいため、一般の GPU を利用しても、VRAM の容量が足りないため、NMT の学習が困難である。そこで、本研究では、データを分割して学習することで、一般の GPU を利用しても、学習可能な方法を考案した。実験の結果、この学習方法が有効に動作することが示された。

1 はじめに

JParaCrawl は、約 3000 万文を超える日英対訳データである。そのため、一般の GPU を利用しても、VRAM の容量が足りないため、NMT の学習が困難である。そこで、学習済みの NMT のモデルが提供されている。そして、テスト文のドメインアダプテーションをすることで、ある程度高い精度の翻訳が可能である。しかし、この場合、NMT のモデルの構造が、固定されてしまい、多様性に欠ける。基本的には、JParaCrawl を直接学習できることが、好ましい。

本論文では、一般の GPU を使って、JParaCrawl を直接学習できる方法を提案する。この方法は、以下の仮説に基づいている。

1. テスト文に類似した、ある程度の対訳データが存在する。
2. 大量の対訳データを分割して、テスト文に対するアダプテーションをおこなう。

以上を考慮して、JParaCrawl を一般の GPU を使って直接学習する方法を提案する。

2 NMT の学習における仮説

1. 要素合成法の問題

文の翻訳において、単語単位に正確に翻訳しても、文全体からみると不可解な翻訳にある。以下が良い例である。

例 彼女は我を通した She passed me

この問題を避けるには、文全体を考慮して翻訳する必要がある。逆に言えば、大量の対訳文があっても、単語単位に翻訳される。そのため翻訳精度が低下する原因になる。¹⁾

2. 単文翻訳

文の基本要素は単文である。複文は単文に分割することが可能である。

3. NMT の追加学習

NMT の学習は、一度学習を止めて、パラメータを保存し、再び学習をすることが可能である。この再学習するとき、元の学習データと同一である必要は、ない。

4. LSTM

LSTM は、勾配を緩和することで、勾配消失問題を緩和している。これは、一種の Low Pass Filter と見なせる。また、パラメータのスムージングとも見なせる。そのため、不必要なパラメータでも小さな値を維持していることが多い。なお、提案方法では、学習速度を低下させるパラメータを利用する (4f 節)。

以上の事柄を考慮して、大量の学習データがあるときの学習方法を以下に述べる。提案方法は、分割学習と呼べる方法である。

3 提案方法

翻訳する文 (テスト文) は単文とする。そして、以下の対訳データを準備する。

- JParaCrawl のコーパスを分割する。これを $JPARA_1, JPARA_2, JPARA_3, \dots$ とする。
- 対訳単文のコーパスを *Simple* とする

分割学習を、以下にしめす。基本的には、JParaCrawl のコーパスを分割して、個々で学習を行う。

¹⁾ なお、大容量の VRAM を持った GPU を利用して、JParaCrawl を直接学習しても、翻訳精度が、なかなか得られないようだ。

1. *Simple* を学習
2. *Simple* を学習
3. *Jpara₁* を学習
4. *Simple* を学習
5. *Jpara₂* を学習
6. *Simple* を学習
7. *Jpara₃* を学習
8. .
9. .

提案方法は、上記の分割学習を、さらに尤度が収束するまで繰り返す。

4 実験

実験に利用したデータを以下に示す。

1. JParaCrawl V3.0[1]
 - ただし、以下の条件を加えた。
 - 日本文において文末が”。
 - ”英文において文末が”。”
 - 日英同一の対訳文は、1文にまとめる。(sort および uniq を取る。)
 - 使用した対訳文は約 1600 万文になった。
2. 対訳単文
 - 電子辞書などから抽出した、単文・動詞が 1 つの文・約 16 万文 [2]。
3. テスト文
 - 単文 10000 文
4. 学習条件
 - (a) 分割数
 - JParaCrawl を 16 分割
 - (b) 分割学習の回数
 - 分割学習を 10 回
 - (c) 利用した NMT
 - Open-NMT
 - (d) 個々の学習の繰り返し回数
 - 5000step (例 *Jpara₁* を学習する回数)
 - (e) vocabulary size
 - GPU の memory の制限 (12G) に由来して、10 万単語とした。
 - (f) 学習パラメータ
 - 提案方法は、大量の対訳コーパスを分割して学習する。そのため、各分割したコーパスにおいて未知語が存在する。したがって、通常の学習より収束を遅くする必要がある。そこで、以下のパラメータにする。

- learning-rate: 0.9
 - pos-ffn-activation-fn: gelu
- その他のパラメータは default とする。

5. 形態素
 - 日本語は mecab で形態素解析をおこなった。
 - 英語は、文末の”。” および文中の”,” は、前後に空白を加え 1 単語とした。
6. ベースライン
 - 比較実験のため、JParaCrawl を利用しなくて、対訳単文 16 万文だけを学習した実験をベースラインとする。

5 実験結果

5.1 自動評価

自動評価による実験結果を表 6 に示す。評価文は 10000 文である。

評価方法	BLEU	meteor	TER	RIBES
提案手法	0.1991	0.4843	0.5845	0.7848
ベースライン	0.1775	0.4486	0.6197	0.7631

以上の結果より、自動評価において、提案手法の有効性が見られた。

5.2 人手評価

提案手法とベースラインの対比較評価を行った。対象は約 100 文である。評価者は 4 名である。結果の平均を表 2 に示す。

提案手法 > ベースライン	45.2%
提案手法 < ベースライン	7.4%
提案手法 = ベースライン	47.3%

以上の結果より、人手評価においても、提案手法の有効性が見られた。

5.3 出力例

翻訳例を表 3 に示す。表中のベースはベースラインを意味する。

表3 翻訳例

入力	信号が青より赤に変わった。
提案	The signal turned green to red .
ベース	The light turned red to red than green .
参照	The signal changed from green to red .
入力	私は電車事故で足留めを食った。
提案	I was <unk> in a train accident .
ベース	I was robbed in the train accident .
参照	I was stranded as a result of the train accident .
入力	彼女は我を通した。
提案	She <unk> herself .
ベース	She has come .
参照	She had her own way .
入力	ぶらんこが揺れている。
提案	<unk> are swinging .
ベース	The <unk> is swinging .
参照	The swing is swinging .
入力	エイズの確実な治療法はまだわかっていない。
提案	The cure for AIDS is not yet known .
ベース	The fundamental method of AIDS is not yet known .
参照	No sure cures for AIDS are known yet .
入力	その新人女優は体当たりの演技で新人賞を獲得した。
提案	The new actress won a new award with the <unk> performance .
ベース	The new actress won <unk> performance in <unk> performance .
参照	That starlet put everything she had into her part and won the prize for new talent .
入力	12号線環状部は一九九二年に着工、来年十二月に開業を予定している。
提案	The 12 lines are scheduled to start in 1992 , and in December , we plan to open our business .
ベース	The <unk> line is scheduled to begin operation in 1970 , in 1954 .
参照	Construction on the subway line began in 1992 , and the line is scheduled to be put into operation in December next year .

6 考察

6.1 自動評価と人手評価

人手評価では、自動評価と大きな差が出た。これは、自動評価では、文の一部を評価しているのに対し、人手評価では、文全体を評価することで、差が出ていると考えている。今後、文全体を評価する自

動評価方法を考えていく必要がある。この方法の1つは、参照文との一致率 [3] を見る方法と考えている。

6.2 GPU の VRAM と NHT の vocabulary size と未知語

NMT の vocabulary size と GPU の VRAM 量は、ほぼ比例の関係にある。実験に使用した GPU は 12G である。そのため今回の実験では、vocabulary size を 100000 にせざるを得なかった。そのため、出力文において<UNK>が、多く出力される。

この問題点を避けるために、形態素を SentencePiece[4] に変更する方法がある。SentencePiece では、固有名詞を文字に分割するため、未知語は出力されない。しかし、分割することの問題もでる。簡易実験を行ったところ、“知床”を“knowledge floor”と翻訳された。そして全体の翻訳精度が低下した。sentencePiece を利用するときには学習パラメータの再調査が必要である。

6.3 学習パラメータ

今回の実験では、学習速度を低下させるために、以下のパラメータを用いた。

- learning-rate: 0.9
- pos-ffn-activation-fn: gelu

他にも、多くのパラメータがある。これらを追求して行きたい。

6.4 google 翻訳との比較

現在 web 上の google 翻訳が利用可能である。提案手法と google 翻訳を比較した。テスト文は約 100 文である。評価者は 4 名の対比較試験である。結果を表 4 に示す。

表4 対比較評価(人手)

提案手法 > google	9.4%
提案手法 < google	40.2%
提案手法 = google	50.2%

結果として、提案手法は、google 翻訳には、まだ追いついていない。この最大の原因は、未知語の存在である。例を以下に上げる。

表5 google 翻訳との比較

入力	信長勢は再び京へ上った。
提案	The <unk> went to Kyoto again .
google	Nobunaga's army went up to Kyoto again .
参照	Nobunaga's army went up to Kyoto again .

語彙数 100000 では、入力文において未知語が存在する。この例では、“信長勢”が未知語になる。そのため、出力文に未知語が出力される。そのため、google 翻訳が良いと判断される。このような例が非常に多い。

6.5 JParaCrawl のみの学習

対訳単文のコーパス *Simple* を利用しないで、JParaCrawl のコーパスのみを用いて分割学習を行った。テスト文は 10000 文である。結果を表??に示す。この表から JParaCrawl のコーパスのみを用いて学習しても、翻訳精度は低いことがわかる。

表 6 JParaCrawl のみの学習

評価方法	BLEU	meteor	TER	RIBES
JParaCrawl のみ	0.1400	0.4176	0.6521	0.7360

6.6 大量の対訳データを学習する方法

大量の対訳データを学習する方法は、本論文で述べた分割学習を用いる方法以外にも、以下の方法がある。

1. fine tuning

この方法は、一般的な方法である。大量の VRAM を持つ GPU で、1 度 JParaCrawl の対訳データを学習して、base モデルとする。このモデルに対して、翻訳対象の分野の対訳データに fine tuning する。ただし、この方法は、NMT のモデルが、base モデルに依存する。そのため拡張性に欠ける欠点がある。

2. 類似文

翻訳対象の文に類似した文を JParaCrawl から抽出し、この対訳文を学習データに加える。この方法は、テスト文が異なると、再度 NMT を学習する必要がある。

これらの方法は、一長一短ある。データベースの種類や量やテストデータの依存する。今後、最適な方法を調査したい。なお予備実験では、類似文の追加が最も良い翻訳性能を得ている。

7 おわりに

JParaCrawl は、約 3000 万文を超える日英対訳データである。規模が非常に大きいため、NMT の学習が困難である。そこで、本研究では、JParaCrawl を分割して、個々に学習する方法を提案した。個々に学習した場合、GPU のメモリが少なくでも計算可能である。実験の結果、この方法が有効に動作するこ

とが示された。そして、特に人手の評価に大きな有効性が得られた。

ただし提案方法には、未知語において大きな問題点がある。また、多くの拡張方法がある。これらを追求していきたい。

謝辞

人手評価に参加した以下の方々に感謝します。

矢野 貴大, 本田 涼太, 三木 謙志, 名村 太一, 丸山 京祐