日本語形態素解析器の知識蒸留

田上 青空（指導教員：戸次大介）

1 はじめに

現在の日本語形態素解析器は非常に高精度であり、様々な自然言語処理技術を用いた実サービスにも応用されている．しかし、従来の形態素解析器は辞書などによりモデルサイズがあまり小さくないため、組み込みサービスを展開する上で課題となる．そこで 2019 年に提案された rakkyo[9] は、Juman++[6] の高速・軽量化を試みている．Juman++を用いて作成した 5 億文のシルバーデータを学習することで、辞書を使わず、モデルサイズを小さくしている．

一方で、近年では事前学習モデルの台頭によりパラメータ数の多いモデルが増加していることから、大規模モデルを圧縮する研究が盛んに行われている．その一つとして大規模モデルの確率分布を教師データとして学習する知識蒸留 [2] という手法が知られている．

本稿では、知識蒸留を用いて rakkyo をさらに圧縮することで、より少ないデータ数で rakkyo の出力に近づけるかを評価する．

2 形態素解析器

形態素解析器は、探索型と点予測型の 2 つに大別される．

2.1 探索型

探索型は、文字列に対して複数の解の経路を構築し、その中で最もコストが低い経路を探索する手法である．最も一般的な構造は、ラティスと呼ばれる形態素の全解候補をグラフ構造で表すものである．ノードは形態素の候補を、エッジは形態素の連接を表現し、それぞれに形態素やその連接の出現しやすさを表したコストを与えて探索を行う．この手法は、辞書に含まれる単語を考慮してラティスを構築するため、大規模辞書が数多く開発・公開されている日本語研究によくみられる．JUMAN[4]、ChaSen[5]、MeCab[3]、Sudachi[8] など、主要な日本語形態素解析器の多くがこの手法を用いている．

2.2 点予測型

点予測型は、文字 1 つ 1 つに対して単語区切りの有無と形態素を予測する手法である．文字 n-gram、文字種 n-gram、辞書素性などから、他の文字とは独立に形態素を予測する．この手法は、辞書が必須ではないため、大規模辞書があまり公開されていない中国語で多く用いられている．探索型形態素解析器に比べ、実装が容易であり、専門用語等辞書に存在しない単語の解析精度が向上することが知られている．点予測型の代表的な日本語形態素解析器として、KyTea[7] があげられる．

また近年では、ニューラルネットワーク (NN) を用いた実装が有効であることも示されている．そのような研究の一つに rakkyo[9] が挙げられる．これは、Juman++[6] を高速・軽量化したモデルであり、Juman++で作成した 5 億文のシルバーデータを用いて学習を行っている．素性として文字ユニグラムのみを用いた点予測を採用していることと、大規模なコーパスを使用することにより辞書が不必要になることから、モデルサイズを大幅に削減している．

3 知識蒸留

近年、高精度な NN モデルが多く提案されているが、実応用においては精度だけでなくモデルサイズも考慮しなければならない．そこで大規模モデルを圧縮する手法として知識蒸留 (Knowledge Distillation) が用いられている．この手法は、正解ラベルを教師信号として用いる代わりに、大規模・高性能なモデル（教師モデル）が出力した確率分布を教師信号として学習する．これにより、小規模な生徒モデルにも教師モデルに近い精度を期待できる．

自然言語処理分野では、BERT[1] の発表を皮切りに事前学習モデルの研究が進められている．これらは高精度であるが年々パラメータ数、モデルサイズが肥大化する傾向にあり、実サービスに組み込むためには小さなモデルとして提供する必要がある．そこで、知識蒸留はそれらの大規模なモデルを圧縮する手法として用いられている．

4 提案手法

本研究では rakkyo[9] を知識蒸留することにより、その確率分布を教師信号として用いて点予測型 NN モデルを構築した．予測する項目は rakkyo と同じく、{B, I, E} からなる単語分割タグ、品詞大分類、品詞細分類、活用型、活用系の 5 つである．{B, I, E} はそれぞれ単語の始まり、中間、終わりを表す．5 項目それぞれについて予測不可を表すタグも含め、4 値分類、16 値分類、38 値分類、34 値分類、82 値分類となっている．

まず文を文字ごとに分割し、入力とする．biLSTM を通して出力された各タイムステップごとの隠れ層に、タグごとの MLP を適用する．その出力ベクトルに対して、Softmax 関数を適用し確率分布を得て、rakkyo が出力する確率分布との間の KLDivergence を損失関

数として学習する．

$$D_{KL}(P||Q) = \sum_{x \in X} P(x) \log \left(\frac{P(x)}{Q(x)} \right) \quad (1)$$

モデルの構築は、Torch の Haskell インターフェースである hasktorch[1]を用いて行った．

5 実験と結果

5.1 実験設定

実験環境には、産総研 AI 橋渡しクラウド (ABCI) の rt_C.large (Intel Xeon Gold 6148、20 コア、120GiB) を 1 ノード使用した．

国語研 BCCWJ コーパスを用いて、rakkyo から抽出した確率分布を教師データとして学習を行った．モデルのパラメータを表 1 に示す．文字埋め込みは学習データに現れる文字のみを用いている．rakkyo の出力を正解ラベルとし、単語区切りのみ、単語区切りと品詞大分類、さらに全ての項目での精度を測定した．コーパスはそれぞれ 273 文、570 文、767 文、1018 文、1253 文で実験を行い、データ数による精度の増減を確認した．

表 1: 提案モデルのパラメータ

文字埋め込み次元	512
LSTM 層の数	1
隠れ層の次元	512
学習率	10e-2
バッチサイズ	16
epoch	100

5.2 実験結果

実験結果を、表 2 に示す．データ数に比例して精度が上昇していることが見受けられる．今後さらにデータ数を増やしていけば、精度が上がることが期待できる．

表 2: 精度

データ数	単語区切り	+品詞大分類	全体
273 文	69.76%	58.76%	60.08%
570 文	76.93%	67.48%	66.56%
767 文	78.01%	69.86%	68.68%
1018 文	82.85%	74.13%	71.86%
1253 文	84.76%	76.71%	73.94%

6 おわりに

本研究では、日本語形態素解析器 rakkyo の知識蒸留を行い、データ数による精度の増減を評価した．今後はさらにデータ数を増やし精度との関係を実験を重ね確かめていきたい．また、ハイパーパラメータの調整やモデルサイズの比較など最適なアーキテクチャを検討するとともに、ゴールドデータを用いて、rakkyo の精度とどれほど差を縮められるかを評価したいと考える．さらに、少ない文で高精度な予測をできることを活用し、特定ドメインの分野適応についても検討していきたい．

謝辞：早稲田大学基幹理工学部・河原大輔教授には、rakkyo の学習済みモデルを提供して頂いた．本研究の一部は、JSPS 科研費 JP18H03284、および JST CREST JPMJCR20D2 の助成を受けたものである．また、評価の際は産総研の AI 橋渡しクラウド (ABCI) を利用した．

[1] http://hasktorch.org/

参考文献

[1] Devlin, J., Chang, M., Lee, K. and Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, *CoRR, arXiv:1810.04805* (2018).

[2] Hinton, G., Vinyals, O. and Dean, J.: Distilling the Knowledge in a Neural Network, in *NIPS Deep Learning and Representation Learning Workshop, arXiv:1503.02531* (2015).

[3] Kudo, T., Yamamoto, K. and Matsumoto, Y.: Applying Conditional Random Fields to Japanese Morphological Analysis, in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, Barcelona, Spain (2004).

[4] Kurohashi, S.: Improvements of Japanese Morphological Analyzer JUMAN, *Proceedings of the International Workshop on Sharable Natural Language Resources (SNLR), 1994*, pp. 22–28 (1994).

[5] Matsumoto, Y.: Japanese Morphological Analysis System ChaSen Version 2.0 Manual, *Technical Report* (1999).

[6] Morita, H., Kawahara, D. and Kurohashi, S.: Morphological Analysis for Unsegmented Languages using Recurrent Neural Network Language Model, in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal (2015).

[7] Neubig, G., Nakata, Y. and Mori, S.: Pointwise Prediction for Robust, Adaptable Japanese Morphological Analysis, in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, Oregon, USA (2011).

[8] Takaoka, K., Hisamoto, S., Kawahara, N., Sakamoto, M., Uchida, Y. and Matsumoto, Y.: Sudachi: a Japanese Tokenizer for Business, in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan (2018), European Language Resources Association (ELRA).

[9] Tolmachev, A., Kawahara, D. and Kurohashi, S.: Shrinking Japanese Morphological Analyzers With Neural Networks and Semi-supervised Learning, in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota (2019).