能動的サンプリングを用いた リソース構築共有タスクにおける予測対象データ削減

中山 功太 †,‡　栗田 修平 †　馬場 雪乃 †,‡　関根 聡 †

† 理化学研究所
‡ 筑波大学
{kouta.nakayama, shuhei.kurita, satoshi.sekine}@riken.jp
{baba}@cs.tsukuba.ac.jp

1 はじめに

今日まで、多くの共有タスクが自然言語処理の進歩に大きく貢献してきた。しかし、それらタスクのほとんどは技術開発が目的であり、タスク内で実際に構築されたシステムや予測結果等の共有を目的としない場合が多い。この様な参加者の努力が浪費されている状況は無視できるものではない。近年、タスク参加者による努力をリソース構築といった形で共有する “Resource by Collaborative Contribution(RbCC)”[1] という考え方が考案された。

RbCC に準拠するリソース構築共有タスクでは、参加者は予め指定された範囲のラベル無しデータを受け取り、その全てに対する予測結果を提出する必要がある。評価は一部の隠されたデータにより行われる。参加者から得られた複数の予測結果はアンサンブル等により統合され、最終的に知識ベースなどのリソースとして公開される。実際に RbCC に準拠するタスクが行われ [1, 2]、アンサンブル手法も提案されてきた [3]。RbCC の最終目的はリソース範囲全てのデータに対する予測結果を得ることであり、過去のタスクでは参加者は範囲全てのデータに対する予測の提出が要求されている。しかしリソースのサイズ次第では、予測にかかる計算コストは無視できず、タスク参加への高いハードルとなり得ると考える。タスク参加者ごとで予測範囲を分配することも考えられるが、データ間で品質のばらつきが発生する可能性は捨てきれない。最良スコアを獲得した参加者のみ全件予測を依頼することも可能だが、採択されなかった参加者の努力の浪費につながり、これは RbCC の本意ではない。整理すると、負担軽減の理想は、提出データを削減しつつ最良システム以上のスコアで提出範囲外の最終予測を補完可能なことである。

本研究では、実際に RbCC に準拠するタスクを有する森羅プロジェクトで用いられているアンサンブル手法である Pre-Distillation Ensemble(PDE)[3] に能動学習 [4] の考え方を組み合わせることで、最終的なリソースの品質を落とさず予測対象データを削減し、参加者の負担軽減を試みる。実際のリソース共有タスクにおいて、提案手法が効果的に参加者の負担を削減できることを示すため、森羅プロジェクトの一つである SHINRA2020-ML[5] の結果を用いて予備実験を行う。終わりに、その結果を用いて次年度以降の森羅プロジェクトにおける適用可能性について考える。

2 手法

2.1 Pre-Distillation Ensemble(PDE)

PDE[3] は、教師ありモデルの予測結果を統合する手法である。PDE では、初めに各参加システムの予測結果を学習データとして用い、新規モデルを学習する。モデルの結果から新規モデルを学習することを蒸留と呼ぶ。蒸留に用いるモデルは、図 1 上部に示すような主たる共有モデルに対し各システムの出力値再現を担当する専用モデルをそれぞれ接続する形で構成され、並列に全てのシステムの予測を行う。蒸留時、専用モデルのパラメーター数が小さければ、各システムの出力予測に必要な情報の多くは共有モデルへと埋め込まれる。その後、図 1 下部のように学習済み共有モデルに対し新たな専用モデルを接続し、各参加システムの学習に使用されたラベル付きデータを用いてモデルを再学習する。学習中、新たに接続された出力モデルは、共有モデルに埋め込まれた各システムの情報をもとに、新たな予

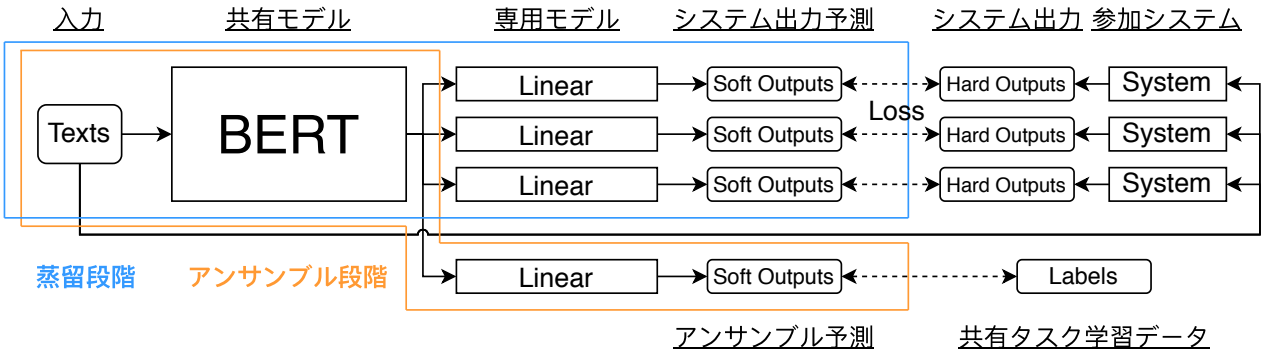

図 1　本研究で使用する PDE モデル

測を出力する。PDE では、このモデルの出力結果を擬似的なアンサンブル結果として扱っている。計 2 回行われる学習はそれぞれ蒸留段階、アンサンブル段階と呼ぶ。

2.2　能動学習

教師ありモデルを学習する場合、ラベル無しデータに対してアノテーションを施し学習データを作成する必要がある。一般的にアノテーションはコストがかかるため、少ないアノテーション数でより高性能なモデルが学習できることが望まれる。能動学習 (Active Learning)[4] は、ある時点で既にアノテーションが施されたデータにより学習されたモデルの予測結果等を用いて、モデルの評価値を最も向上することができるような次のアノテーション対象を能動的にサンプリングする機械学習フレームワークである。一般的に能動学習におけるアノテーション結果は専門家やクラウドワーカーへの問い合わせによって得られる。自然言語処理においても様々なタスクにおいて用いられており、様々な手法が開発されている。能動学習の古典的手法として不確実性サンプリング (Uncertainty Sampling)[6] が挙げられる。不確実性サンプリングでは、モデルの予測値が決定境界に近いインスタンスが次のアノテーション対象として選択される。

2.3　提案手法

RbCC に準拠するタスクにおいて、提出される予測対象の多さが参加者の負担となり得ることは 1 章で述べたとおりである。本論文では、参加者の負担軽減のため、PDE と能動学習を組み合わせることで、最終的なリソースの品質を落とさずに予測対象データの削減を試みる。

PDE では、システムの予測結果は蒸留に使用されるが、直接的にアンサンブル結果に使用されることはない。そのため、PDE に用いるシステムの予測結果は全ての範囲に対するものである必要はない。つまり、最終的なアンサンブル予測を最も洗練できるような蒸留対象を選択することでスコアを担保しながら参加者に要求する予測対象を削減することができると考える。蒸留対象の選択が、参加システムに対する問いあわせであると考えると、能動学習のフレームワークを直接対応できる。本論文では、PDE と能動学習の組み合わせの優位性を予備実験的に示すため、不確実性サンプリングを用いて、タスク参加者に問い合わせるラベル無しデータのサンプルを決定する。

3　SHINRA2020-ML

SHINRA2020-ML[5] は、NTCIR-15[7] の 1 つであり、拡張固有表現 (ENE) 階層 [8] へ 30 言語の Wikipedia 記事を分類するタスクである。ENE 階層は約 200 の末端カテゴリーを持っており、1 つのページが複数の末端カテゴリーに属する場合もある。そのため、SHINRA2020-ML は多ラベル文章分類タスクである。学習データは、日本語 Wikipedia を ENE 階層へと分類した日本語分類済みデータ [9] と、日本語 Wikipedia から他言語への言語間リンクを用いて作成される。予測対象は各言語の全 Wikipedia 記事である。これは各言語ごと異なり、最大で 5,790,377 件、最小で 129,141 件であり、30 言語全ての合計は、32,302,922 件である[1)]。予測対象には学習データの範囲に対する記事も含まれるため、再度予測し提出する必要がある。評価データは各言語 1000 件用意されるが、RbCC の考えに基づき参加者には配布されない。評価は提出データ中の

1)　森羅プロジェクト統計情報より算出。
http://shinra-project.info/shinra2020ml/statistics/ (参照 2021-01-11)

言語	サンプル数	予測対象	削減率	学習データ数	ベースモデル	ランダム	不確実性	最良システム
アラビア語	34,800	661,205	94.74%	73,054	64.81	**71.57**	**70.78**	76.27
ドイツ語	36,400	2,262,582	98.39%	274,732	80.53	**81.90**	**81.63**	81.86
英語	38,400	5,790,377	99.34%	439,354	81.56	**81.72**	**82.65**	82.73
スペイン語	37,400	1,500,013	97.51%	257,835	80.51	**80.55**	79.98	81.39
フランス語	37,200	2,074,648	98.21%	318,828	81.01	80.41	80.87	81.01
イタリア語	36,200	1,496,975	97.58%	270,295	82.08	**82.88**	**82.89**	82.81
ポルトガル語	36,200	1,014,832	96.43%	217,896	81.01	80.75	**81.82**	83.23
トルコ語	35,200	321,937	89.07%	111,592	82.53	**83.12**	**84.88**	86.50
中国語	37,200	1,041,039	96.43%	267,107	78.15	**78.51**	**78.88**	81.25
マクロ平均	36,556	1,795,956	96.41%	247,855	79.13	80.16	80.49	81.89

表 1　能動的サンプリングを用いた予測対象データ削減結果

相当する部分を用いて行われる。SHINRA2020-ML は、2020 年 8 月まで行われ、合計で 7 チーム 12 システム[2]の結果が提出された [7]。

4　実験

提案手法が RbCC に準拠する共有タスクにおいて効果的に機能することを予備的に示すため、SHINRA2020-ML に提出されたシステム出力を用いて実験を行う。本研究では、SHINRA2020-ML の内、参加システムが最も多かった 8 言語に英語を加えた 9 言語を対象とする。

4.1　能動的サンプリング

タスク参加者に配布するサンプルを能動的に推定するため、SHINRA2020-ML で配布されたラベル付きデータを使用してニューラルモデルを学習する。ラベル付きデータの 90%を学習データとし、残りを開発データとして用いる。モデルは、事前学習済み言語モデルである BERT[10] に 1 層の全結合層を接続する形で構成される。一般的に BERT では、位置埋め込みの都合上扱うことのできる最大トークン長が限られており、我々が実験に使用する XLM-RoBERTa-base[11] では、512 トークンに制限されている。本実験では Wikipedia の先頭 510 トークンのみを使用して分類を行なった[3]。XLM-RoBERTa は、RoBERTa[12] のモデル構造と手法をベースに、100 言語にわたる約 2.5T バイトの CommonCrawl データから事前学習したモデルである。トークナイズと語彙圧縮には SentencePiece[13] を用いており、語彙数は合計で 250,000 トークンである。本研究では、XLM-RoBERTa を各言語ごとに学習するが、その場合、語彙中の多くが低頻度トークンとなる。低頻度トークンに関する計算が、全体の学習速度や使用する GPU メモリーを圧迫することを防ぐため、各言語の学習データ中に高頻度で出現する 24000 トークン[4]のみを使用し、残りのトークンは未知語として扱う。多ラベル分類を、各クラスごとの 2 値分類として考え、誤差関数には交差エントロピーを用いる。その他学習の詳細は付録 A に示す。

多ラベル分類における不確実性サンプリングには様々な手法が提案されているが、本研究では各クラスを 2 値分類タスクとみなし、各クラスごとに 200 件ずつ不確実性の高いインスタンスをサンプリングする。この際、各インスタンスは学習データにおいて出現数が高いクラスから重複が無いようサンプリングされる。また、ラベル付きインスタンスはサンプルの対象としない。最終的にサンプリングされたインスタンス数を表 1 に示す。

4.2　PDE 学習

図 1 に本研究で使用する PDE モデルを示す。共有モデルには、4.1 でも用いた XLM-RoBERTa-base を用い、同様の語彙数削減を行う。専用モデルには、1 層の全層結合を用いる。誤差関数も 4.1 章と

2）　Official submission と Late submission 間での同じシステムは合わせて 1 システムとして換算

3）　BERT では入力の先頭と末尾に特殊トークンを接続する必要があるため、入力文に使用できる最大長は 510 トークンとなる。

4）　XLM-RoBERTa で使用される特殊トークンを含む。

同じものを用いるが、蒸留段階のみ陽性ラベルに対して重みを付与して計算する。重みは、各システム出力ごとに陰性ラベルの総数を陽性ラベルの総数で割った数値を用いる。蒸留段階における各システム出力との複数誤差は算術平均により統合する。その他学習の詳細は付録 B に示す。

5 実験結果

実際に不確実性サンプリングにより得られたサンプルに対し PDE を適用した結果を表 1 に示す。比較のため、同じ数のサンプルをランダムサンプリングにより取得し、同様に PDE を適用した。ベースモデルは、不確実性サンプリングに使用されたモデルのスコアを示しており、最良システムは SHINRA2020-ML に参加したシステムのうちの最良スコアを示している。また、ベースモデルより良いスコアを太字、最良システムより良いスコアを下線で示している。削減率は、本来の予測対象を能動的なサンプリングによりどれだけ削減できたかを示しており、この値が高いほど参加者の負担軽減につながる。評価指標は、SHINRA2020-ML と同じく F1 値のマイクロ平均である。

不確実性サンプリングを用いた場合、スペイン語、フランス語以外での言語でベースモデルより良い性能を達成していることから、モデルの性能を向上できるようなサンプルが導出できていることが分かる。しかし、アラビア語、ドイツ語、スペイン語ではランダムサンプリングを用いた場合において、不確実性サンプリングを用いた場合よりスコアが高い。これは、蒸留段階における本来のクラス分布と大きく異なったサンプルの選択が、アンサンブル段階後のモデルの出力分布に負の影響を及ぼす可能性を示唆している。この問題は、各クラスごと均等数の不確実性サンプリングではなく、実際の学習データの分布を用いてサンプルの割合を決定することで解決できる可能性がある。

最良システムとの比較では、イタリア語において不確実性サンプリングを用いた手法が良いスコアを残している。イタリア語では予測対象データの 97% 以上の削減に成功しており、タスク参加者の負担軽減に大きく貢献している。残りの言語においては、最良システムに対し不確実性サンプリングが劣っている結果になっているが、これらは PDE に用いたモデルが文章の先頭のみの情報から分類を行なっていることが要因の一つであると考えられる。実際に SHINRA2020-ML に参加したシステムには、より広い範囲の文章を参照するシステムや、言語間リンクや記事に付与された画像等追加の付加情報を使用するシステムがあり、現状のモデルとの間で入力情報量に差異が生まれてしまっていることは否定できない。本研究は予備実験のため、簡易なモデル設計を行なったが、今後実際に共有タスクで適用する場合には、PDE モデルは様々な入力に対応できる設計が好ましい。また、アラビア語では、不確実性サンプリングを用いた場合のスコアが大きく劣っているが、ベースモデルのスコアはさらに大きく劣っていることから、対象言語におけるモデルの学習法自体に大きな問題が存在する可能性があり、これについては今後学習法の見直しを考えている。

6 おわりに-今後の展望

本研究では、実際に RbCC に準拠するリソース構築タスクで使用されているアンサンブル手法である PDE と、能動学習のフレームワークを組み合わせることで、タスク参加者が提出を求められる予測対象データを削減する手法を提案した。SHINRA2020-ML の結果を用いて予備実験を行ったところ、提案手法は 9 カテゴリーのうち 1 カテゴリーで 97%以上の予測対象データを削減した上で、最良システム以上のスコアで削減された範囲を補完できた。また、残りの 8 カテゴリー中 6 カテゴリーにおいても、F1 値の劣化を 1.7 以内に収めた上で、89%〜99%予測対象を削減できることが分かった。

今後は全てのカテゴリーにおける最良システム以上の補完を目指し、サンプリング手法やサンプルサイズを変化させた場合の実験や、より多くの情報を活用したモデルの設計等の改善を行いたいと考えている。また、改善手法は 2021 年に開催する SHINRA2021-ML タスクに対し適用する予定である。具体的には、本手法により能動的に推定されたサンプルをリーダーボードの提出データに含めることを考えている[5)]。リーダーボードに提出したデータ内で最終評価を行うことも視野に入れており、積極的に一部予測のみの最終提出を受け入れる予定である [14]。

参考文献

[1] Satoshi Sekine, Akio Kobayashi, and Kouta Nakayama. SHINRA: Structuring Wikipedia by Collaborative Con-

5) リーダーボード上における評価は、その一部で行われる。

tribution. In *Submitted to Automated Knowledge Base Construction*, 2019.

[2] 小林暁雄, 中山功太, 安藤まや, 関根聡. Wikipedia 構造化プロジェクト「森羅 2019-JP」. 言語処理学会第 26 回年次大会, 2020.

[3] 中山功太, 栗田修平, 小林暁雄, 関根聡. Pre-Distillation Ensemble:リソース構築タスクのためのアンサンブル手法. 言語処理学会第 26 回年次大会, 2020.

[4] Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[5] Satoshi Sekine, Masako Nomoto, Kouta Nakayama, Asuka Sumida, Koji Matsuda, and Maya Ando. Overview of shinra2020-ml task. In *Proceedings of the NTCIR-15 Conference*, 2020.

[6] David D. Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In William W. Cohen and Haym Hirsh, editors, *ICML*, pp. 148–156. Morgan Kaufmann, 1994.

[7] Yiqun Liu, Makoto P. Kato, and Noriko Kando. Overview of ntcir-15. In *Proceedings of the NTCIR-15 Conference*, 2020.

[8] Sekine Satoshi. Extended named entity ontology with attribute information. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May 2008. European Language Resources Association (ELRA).

[9] Masatoshi Suzuki, Koji Matsuda, Satoshi Sekine, Naoaki Okazaki, and Kentaro Inui. A joint neural model for fine-grained named entity classification of wikipedia articles. *IEICE Transactions on Information and Systems*, Vol. E101.D, No. 1, pp. 73–81, 2018.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. 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)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[11] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8440–8451, Online, July 2020. Association for Computational Linguistics.

[12] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, Vol. abs/1907.11692, , 2019.

[13] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 66–71, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[14] 関根聡, 野本昌子, 中山功太, 隅田飛鳥, 松田耕史, 安藤まや. SHINRA2020-ML:30 言語の Wikipedia ページの分類. 言語処理学会第 27 回年次大会, 2021.

パイパーパラメーター	ベースモデル			蒸留			アンサンブル		
	ar	en	その他	ar	en	その他	ar	en	その他
Epoch	10			20			10		
Batch size	128	512	256	128			128	512	256
Hidden layer dropout	0.1								
Attention dropout	0.1								
Learning rate	5e-5								
Adam $\beta 1$	0.9								
Adam $\beta 2$	0.999								
Adam ϵ	1e-8								
Weight decay	0.05								

表 2　本研究で用いたハイパーパラメーター

A　ベースモデル学習

学習に使用したハイパーパラメーターを表 2 に示す。学習データのサイズに合わせて、ar のバッチサイズを半分、en のバッチサイズを倍に設定している。学習時の計算コストを削減するため、apex[6] (opt_level には O1 を設定) を用いて混合精度学習を行っている。

B　PDE モデル学習

蒸留段階、アンサンブル段階に使用したハイパーパラメーターを表 2 に示す。アンサンブル段階では、ベースモデルの学習と同様に、言語ごとでのバッチサイズを設定している。学習時の計算コストを削減するため、apex(opt_level には O1 を設定) を用いて混合精度学習を行っている。

6）　https://github.com/NVIDIA/apex