

科学技術論文における「問題」の周辺文からの問題内容の抽出

平林 照雄 (東京農工大学 生物システム応用科学府) *

古宮 嘉那子 (東京農工大学)

浅原 正幸 (国立国語研究所)

Extracting Problem Contents from Surrounding Texts of a Word “Problem” in Science and Technical Papers

Teruo Hirabayashi (Tokyo University of Agriculture and Technology)

Komiya Kanako (Tokyo University of Agriculture and Technology)

Masayuki Asahara (National Institute for Japanese Language and Linguistics)

要旨

科学論文では、しばしば「問題内容」と「解決法」が主題となる。我々は、日本語科学論文において「問題内容」と「解決法」のペアを効率的に取得するシステムの作成を目指している。Heffernan ら (2018) は、“problem” 及びその類義語に注目し、パターン抽出を用いて英語科学論文における「問題内容」提起箇所の抽出を行った。しかし、日本語は英語とは異なり、決まったパターンがないことから、単純なパターン抽出では「問題内容」を抽出することができない。そのため、本論文では、「問題」という語を含む文とそれらの前後文を少量用意し、それらに「問題内容」が含まれているかどうかのアノテーションを行う。また、それらのアノテーションデータを用いて機械学習を行い、大量のデータに適用することで、自己学習により「問題内容」の有無のタグを付与し、付与されたタグの精度を調査した。

1. はじめに

増え続ける科学技術論文の全てを読解することは非常に困難であるから、その読解を助ける目的で、自動知識抽出システムの研究は盛んである。我々は、主題を自動で抽出するシステムについて研究している。学術論文において、主題は問題解決プロセスであることが多い。一般に、問題を提起し、その解決法として手法を提案し、問題の解決を目指すという流れをとる。そこで、我々はこれらの問題内容と解決法のペアの知識を効率的に取得することで、主題読解を機械的に行うシステムを目指している。以降、取得すべき問題内容および解決法を記述した部分をそれぞれ「問題内容」と「解決法」と表記する。

Heffernan and Teufel (2018) は英語科学論文からパターン抽出を利用して“problem” に注目した「問題内容」と「解決法」のペアの情報抽出を行った。日本語科学論文でも同様に、「問題内容」を明示するため、「問題」（以降、「問題」という語を「問題」と表記する）をはじめとする特徴語が、同一文または、周辺文に存在すると予想される。しかし、日本語においては、「問題は～である。」と明記されていることは稀であり、パターン抽出による情報抽出手法を用い

* s213645z@st.go.tuat.ac.jp

ることが難しい。そこで本論文では、「問題」に注目し、「問題」が含まれる文とその前後文に「問題内容」が含まれるか否かを判定する分類器を作成し、作成した分類器で自己学習することで精度の向上を目指す。具体的には、はじめに、日本語科学技術論文中の「問題」を含む文及び、その前後一文を対象に、文に「問題内容」が含まれるか否かを人手でタグ付けしたコーパスを作成し、それらから文の「問題内容」の有無を判定する分類器を作成し、この精度を調べる。次に作成した分類器を、まだアノテーションがされていない、「問題」を含む文及びその前後一文を対象に適用することで、機械的にタグ付けを行い、コーパスの増補を行う。最後に、増補されたコーパスから新しく分類器を作成し、この精度を調べる。

2. 関連研究

科学技術論文を対象にした自動知識抽出システムの研究は多い。例えば、Peng and McCallum (2006) は、科学技術論文を対象に、論文のヘッダーと呼ばれる冒頭部分から、CRF を用いて、タイトル、著者、所属などの 15 種類の情報を抽出している。

Heffernan and Teufel (2018) は、“problem” とその類義語を利用した情報抽出により、英語科学論文から問題提起箇所を含む「問題内容」と「解決法」コーパスを作成し、さらに、文中に「問題内容」または「解決法」を含むか否かという分類を行った。

また、「問題内容」を抽出するために、「問題」に注目している研究は他にもあり、Scott (2001) は、「問題」や解決法という語に類似した単語を特定し、それらの単語が問題解決プロセスを示す際にどのように用いられるかを明らかにしていた。

本論文で用いている自己学習手法は、分類器の精度の向上に貢献する。例えば、Yarowsky (1995) や鈴木ほか (2019) は、ブートストラップ法による自己学習を行い、教師なし学習による語義曖昧性解消の精度が向上している。

3. 提案手法

3.1 「問題」の語義曖昧性解消

まずはじめに、平文のコーパスから、「問題」を抽出し、その語義曖昧性解消を行う。先行研究として、平林ほか (2021) で、「問題」には、困っていること、解決したいことを意味する “problematic”、クイズなどのお題という意味の “task” 等の意味があるが、「問題内容」および「解決法」が存在する「問題」は “problematic” の意味のみであるから、「問題」の語義曖昧性解消を必要とすることを述べ、「問題」の語義をタグ付けしたコーパスの作成を行った。先行研究では、コーパス中の「問題」の一部の用例について人手で語義曖昧性解消を行い、残りを機械的に付与したコーパスを作成したが、本研究で用いられる「問題」を含む文は全て人手で語義曖昧性解消を行った文である。

3.2 「問題内容」のアノテーション

本研究では、「問題」の意味をタグ付けしたコーパスから、本研究では、「問題」の意味をタグ付けしたコーパスから、「問題」を含む文及び、その前後一文を対象に、文に「問題内容」が含まれるか否かのタグを付与する。具体的には、対象三文に、人手で一文ずつ、後述のタグ

付けガイドラインに基づき、文に、III(a)~III(c) のタグを付与する。ここで対象とする「問題」は、「問題」の語義を特定済みで、なおかつ「問題内容」とその「解決法」が存在し得る ‘‘problematic’’ の意味のものに限られる。さらに本研究では、「問題」の指し示す「問題内容」の抽出範囲を、その「問題」を含む文か、その周辺文のみとした。タグ付けのガイドラインは以下である。

- I 対象とする三文内で「問題内容」であると判断できない場合、タグを付与しない。
- II 「問題内容」は文中に叙述されている場合のみとし、具体的には「X は問題だ」と言い換えられる場合のみ、タグを付与する。
- III 「問題」との意味関係から「問題内容」を下記の 3 種類に分け、それぞれタグを付与する。

(a) 「問題」が直接指す「問題内容」

[例文] 従って、原則をルールとして生成しても、有効に機能しない場合があるという問題がある。

【出典】『言語処理学会論文誌 LaTeX コーパス』 V12/V12N02-01.tex

(b) 「問題」が直接指す「問題内容」と同じ内容で言い換え

下記例文の二文目に該当する「問題内容」を含む。

[例文] この比喩的な表現の問題を解決するには、比喩に関する人間の常識的な推論が必要である。例えば、「頭が痛い」「寒気がする」「発熱がある」など、疾患・症状が比喩的に使用される例は多くある。

【出典】『言語処理学会論文誌 LaTeX コーパス』 V22/V22N05-02.tex

(c) 「問題」が直接指す「問題内容」とは異なる内容で補足・展開

下記例文の二文目に該当する「問題内容」を含む。

[例文] この選別における問題は、選別の妥当性を確保することである。さらに、選別の対象であるがん用語の候補集合が、なるべく多くのがん用語を網羅していることを保証する必要もある。

【出典】『言語処理学会論文誌 LaTeX コーパス』 V16/V16N02-01.tex

3.3 問題内容を含む文か否かを分類する分類器の作成

作成した「問題内容」のタグを手で付与したコーパスを用いて、「問題」を含む文とその前後一文の計三文を対象に分類器を学習する。この時、対象三文に別の分類器を作成するか否か、使用するモデル、使用する「問題内容」のタグ付け範囲をそれぞれ変化させ、以下の 8 種類の実験を行った。対象三文に共通の分類器を作成するほかに、対象三文に別の分類器を作成する理由としては、「問題」の前後文の分類器に、それぞれ「問題」の前後文であるという情報を分類器に入力として加えることによる精度の変化を調査するためである。

(1) 対象とする三文内で共通の分類器を学習する。

この時、各文はそれぞれ独立して分類器に入力する。

また、3.2 節の III で定義した「問題内容」の分類のうち、(a) と (b) と (c) を正例とする。

- i モデルは、線形の SVM を用いる。
- ii モデルは、BERT(Devlin et al. 2018) を fine-tuning したものをを用いる。

(2) 対象とする三文内で異なる分類器を学習する。

「問題」を含む文の前文の分類器を学習する際には、「問題」を含む文の前文と「問題」を含む文の 2 文を分類器の素性として用いる。

「問題」を含む文の後文の分類器を学習する際には、「問題」を含む文と「問題」を含む文の後文の 2 文を分類器の素性として用いる。

「問題」を含む文の分類器を学習する際には、「問題」を含む文の 1 文のみを分類器の素性として用いる。

i モデルは、線形の SVM を用いる。

- A 3.2 節の III で定義した「問題内容」の分類のうち、(a) のみを正例とする。
- B 3.2 節の III で定義した「問題内容」の分類のうち、(a) と (b) を正例とする。
- C 3.2 節の III で定義した「問題内容」の分類のうち、(a) と (b) と (c) を正例とする。

ii モデルは、BERT を fine-tuning したものをを用いる。

- A 3.2 節の III で定義した「問題内容」の分類のうち、(a) のみを正例とする。
- B 3.2 節の III で定義した「問題内容」の分類のうち、(a) と (b) を正例とする。
- C 3.2 節の III で定義した「問題内容」の分類のうち、(a) と (b) と (c) を正例とする。

線形の SVM による分類器を学習する際には、対象コーパスをランダムに学習データとテストデータに分け、BERT の fine-tuning による分類器を学習する際には、テストデータを線形の SVM と同じ範囲になるように、対象コーパスを学習データと開発データとテストデータに分け、学習データでモデルを学習し、開発データでパラメータ調整を行った後、学習データと開発データを合わせて学習した (図 1)。

その後学習した分類器を用いて、テストデータにより精度の評価を行った。

3.4 自己学習による分類器の追加学習

最後に、作成した分類器を、「問題内容」のタグが付与されていないコーパスにそれぞれ適用することで、「問題内容」のタグを機械的に付与し、コーパスの増補を行う。また、増補したコーパスから、それぞれ分類器を再学習し、その精度を評価する。

この時、線形の SVM による分類器を学習する際には、増補分のコーパスを全て、学習データに加え、コーパス増補前のテストデータで精度を求めた (図 2)。

BERT による分類器を学習する際には、増補分のコーパスをランダムに学習データと開発データに分け、増補前のコーパスの学習データに増補分の学習データを加え、モデルを学習し、増補前のコーパスの開発データと増補分のコーパスの開発データでパラメータ調整を行った後、増補前と増補分の学習データと開発データを合わせて学習し、増補前のテストデータで精度を求めた (図 3)。

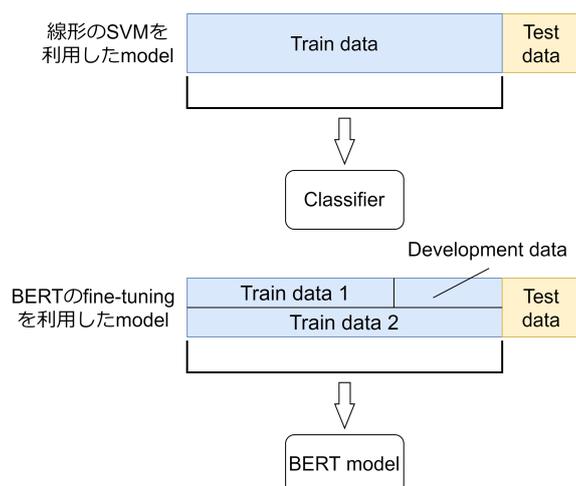


図1 線形のSVMを利用したmodelとBERTのfine-tuningを利用したmodel(ただし、二つのmodel間でTest dataの範囲が同一になるようにとる)

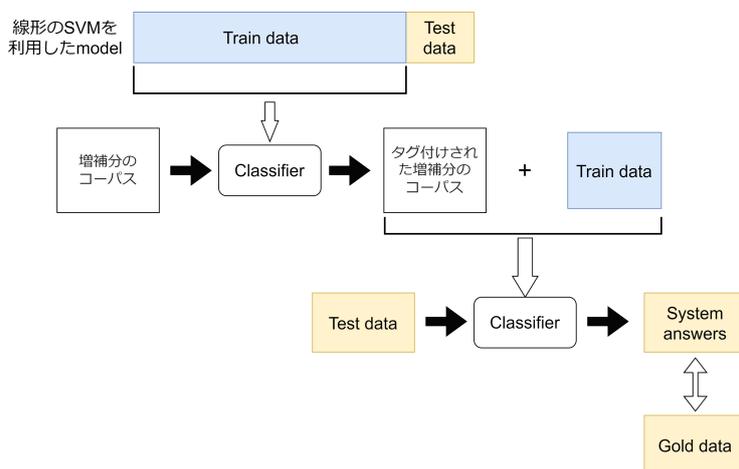


図2 線形のSVMを利用したmodelの再学習とその評価

4. 実験設定

4.1 コーパス情報

本論文ではコーパスとして『言語処理学会論文誌 LaTeX コーパス』⁽¹⁾と、『現代日本語書き言葉均衡コーパス (BCCWJ)』(Maekawa et al. 2014)を利用した。人手で「問題内容」のタグを付与した「問題」を含む文、「問題」を含む文の前文、「問題」を含む文の後文それぞれに対してアノテーションを行い、(a)のみを正例としたときの内訳を表1、(a)と(b)を正例としたときの内訳を表2、(a)と(b)と(c)を正例としたときの内訳を表3に示す。この時、人手でつけた対象コーパスはすべて『言語処理学会論文誌 LaTeX コーパス』内の文である。

⁽¹⁾ https://www.anlp.jp/resource/journal_latex/index.html

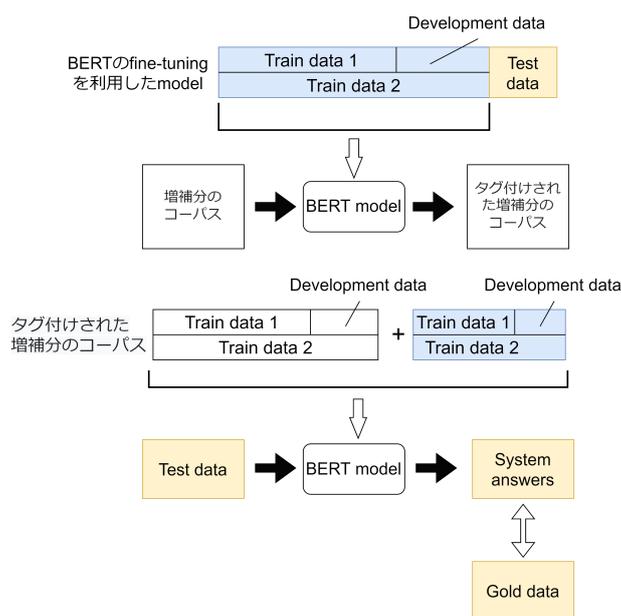


図3 BERTのfine-tuningを利用したmodelの再学習とその評価

表1 (a)のみを正例とするときの人手でタグを付与した対象三文の内訳

	「問題」を含む文		前文		後文		総計
	正例	負例	正例	負例	正例	負例	
人手によるアノテーション総数	133		133		133		399
学習データ	48	31	10	69	1	78	237
開発データ	16	11	3	24	0	27	81
テストデータ	16	11	2	25	0	27	81
総数	80	53	15	118	1	132	399

表2 (a)と(b)を正例とするときの人手でタグを付与した対象三文の内訳

	「問題」を含む文		前文		後文		総計
	正例	負例	正例	負例	正例	負例	
人手によるアノテーション総数	133		133		133		399
学習データ	48	31	11	68	7	72	237
開発データ	17	10	5	22	2	25	81
テストデータ	15	12	3	24	2	25	81
総数	80	53	19	114	11	122	399

表3 (a)と(b)と(c)を正例とするときの人手でタグを付与した対象三文の内訳

	「問題」を含む文		前文		後文		総計
	正例	負例	正例	負例	正例	負例	
人手によるアノテーション総数	133		133		133		399
学習データ	45	34	15	64	10	69	237
開発データ	20	7	4	23	6	21	81
テストデータ	19	8	7	20	5	22	81
総数	84	49	26	107	21	112	399

コーパスの増補時に用いた対象文は、2021年9月6日ー2021年9月8日に実施した yahoo! クラウドソーシングにより整備した。実施時の作業員への質問例を図4に示す。

設定した設問ID : 6222ZV16N03Z01

以下の【問題】という単語の使い方について当てはまるものすべてにチェックをいれてください

また、精度を低下させる原因として、属性・属性値とfacetを含む関係を上位下位関係と誤判定する【問題】が多いことも分かった。

- 「--は【問題】でない」「--は【問題】にならない」と同じ意味である
- 「○○【問題】」など複合語の一部である
- 「【問題】点」に言い換えられる
- 「タスク」に言い換えられる
- 「問い」に言い換えられる
- 「疑問点」に言い換えられる
- 「困難」に言い換えられる
- 今の技術では簡単には解けそうにはない【問題】である
- 「対処する(される)」「解消する(される)」【問題】である

[6222ZV16N03Z01]

図4 クラウドソーシング作業員への質問例

コーパスの増補分の、「問題」を含む文数を表4に示す。

4.2 作成した分類器

SVM は Scikit-learn ライブラリの linearSVC モデルをデフォルト値で使用した。また、事前学習済み BERT モデルは、東北大学乾研究室で公開されている “cl-tohoku/bert-base-

表4 クラウドソーシングによる対象文数と、コーパスの増補に用いた文数の内訳

クラウドソーシング実施対象文数	10,000
内、「問題」を含む文数 (コーパスの増補分の文数)	1,353
内、『言語処理学会論文誌 LaTeX コーパス』内の文数	450
内、BCCWJ 内の文数	903
学習データ	1,082
開発データ	271

japanese-v2⁽²⁾を使用した。また、いずれの分類器も入力として、事前学習済み BERT モデルから出力された分散表現を用いた。対象三文内でそれぞれ異なる分類器を作成する時、「問題」を含む文から作成される分類器を以降の表内で「抽」と表記し、同様に「問題」を含む文の前文から作成される分類器を「前」、「問題」を含む文の後文から作成される分類器を「後」と表記する。

人手でタグを付与した「問題内容」コーパスに対して fine-tuning により学習した BERT モデルの学習率と epoch 数を表5に示す。

表5 人手でタグを付与した「問題内容」コーパスに対して学習した BERT モデルの各種パラメータ

分類器の番号	(1)-BERT		(2)-BERT
学習率	0.001		
epoch 数	3	A	抽 5
			前 0
			後 0
		B	抽 3
			前 4
			後 0
		C	抽 7
			前 6
			後 7

コーパス増補後に、分類器 I-SVM の再学習を行う際の各データの内訳を表6、分類器 I-BERT の再学習を行う際の各データの内訳を表7、分類器 II-SVM-A の再学習を行う際の各データの内訳を表8、分類器 II-SVM-B の再学習を行う際の各データの内訳を表9、分類器 II-SVM-C の再学習を行う際の各データの内訳を表10、分類器 II-BERT-A の再学習を行う際の各データの内訳を表11、分類器 II-BERT-B の再学習を行う際の各データの内訳を表12、分類器 II-BERT-C の再学習を行う際の各データの内訳を表13、に示す。ただし、表内では『言語処理学会論文誌 LaTeX コーパス』を JNLP コーパスと記述する。

⁽²⁾ <https://huggingface.co/cl-tohoku/bert-base-japanese-v2>

表 6 分類器 I-SVM の再学習を行う際の各データの内訳

		「問題」を含む文		前文		後文		総計
コーパス増補後の文数		1,486		1,486		1,486		4,458
		正例	負例	正例	負例	正例	負例	
学習データ	JNLP コーパス	246	283	146	383	118	411	1587
	BCCWJ	575	355	399	531	390	540	2790
	計	821	638	545	914	508	951	4377
テストデータ	JNLP コーパス	19	8	7	20	5	22	81
総数		840	646	552	934	513	973	4458

表 7 分類器 I-BERT の再学習を行う際の各データの内訳

		「問題」を含む文		前文		後文		総計
コーパス増補後の文数		1,486		1,486		1,486		4,458
		正例	負例	正例	負例	正例	負例	
学習データ	JNLP コーパス	119	317	36	400	26	410	1308
	BCCWJ	36	689	11	714	7	718	2175
	計	155	1006	47	1114	33	1128	3483
開発データ	JNLP コーパス	46	74	9	111	11	109	360
	BCCWJ	13	165	4	174	5	173	534
	計	59	239	13	285	16	282	894
テストデータ	JNLP コーパス	19	8	7	20	5	22	81
総数		233	1253	67	1419	54	1432	4458

表 8 分類器 II-SVM-A の再学習を行う際の各データの内訳

		「問題」を含む文		前文		後文		総計
コーパス増補後の文数		1,486		1,486		1,486		4,458
		正例	負例	正例	負例	正例	負例	
学習データ	JNLP コーパス	333	196	31	498	1	528	1587
	BCCWJ	244	686	394	536	0	930	2790
	計	577	882	425	1034	1	1458	4377
テストデータ	JNLP コーパス	16	11	2	25	0	27	81
総数		593	893	427	1059	1	1485	4458

表 9 分類器 II-SVM-B の再学習を行う際の各データの内訳

		「問題」を含む文		前文		後文		総計
コーパス増補後の文数		1,486		1,486		1,486		4,458
		正例	負例	正例	負例	正例	負例	
学習データ	JNLP コーパス	314	215	53	476	27	502	1587
	BCCWJ	384	546	415	515	73	857	2790
	計	698	761	468	991	100	1359	4377
テストデータ	JNLP コーパス	15	12	3	24	2	25	81
総数		713	773	471	1015	102	1384	4458

表 10 分類器 II-SVM-C の再学習を行う際の各データの内訳

		「問題」を含む文		前文		後文		総計
コーパス増補後の文数		1,486		1,486		1,486		4,458
		正例	負例	正例	負例	正例	負例	
学習データ	JNLP コーパス	299	230	85	444	39	490	1587
	BCCWJ	447	483	381	549	132	798	2790
	計	746	713	466	993	171	1288	4377
テストデータ	JNLP コーパス	19	8	7	20	5	22	81
総数		765	721	473	1013	176	1310	4458

表 11 分類器 II-BERT-A の再学習を行う際の各データの内訳

		「問題」を含む文		前文		後文		総計
コーパス増補後の文数		1,486		1,486		1,486		4,458
		正例	負例	正例	負例	正例	負例	
学習データ	JNLP コーパス	328	100	10	418	1	427	1284
	BCCWJ	467	266	0	733	0	733	2199
	計	795	366	10	1151	1	1160	3483
開発データ	JNLP コーパス	101	27	3	125	0	128	384
	BCCWJ	107	63	0	170	0	170	510
	計	208	90	3	295	0	298	894
テストデータ	JNLP コーパス	16	11	2	25	0	27	81
総数		1019	467	15	1471	1	1485	4458

表 12 分類器 II-BERT-B の再学習を行う際の各データの内訳

		「問題」を含む文		前文		後文		総計
コーパス増補後の文数		1,486		1,486		1,486		4,458
		正例	負例	正例	負例	正例	負例	
学習データ	JNLP コーパス	366	82	65	383	7	441	1344
	BCCWJ	571	142	33	680	0	713	2139
	計	937	224	98	1063	7	1154	3483
開発データ	JNLP コーパス	90	18	19	89	2	106	324
	BCCWJ	154	36	2	188	0	190	570
	計	244	54	21	277	2	296	894
テストデータ	JNLP コーパス	15	12	3	24	2	25	81
総数		1196	290	122	1364	11	1475	4458

表 13 分類器 II-BERT-C の再学習を行う際の各データの内訳

		「問題」を含む文		前文		後文		総計
コーパス増補後の文数		1,486		1,486		1,486		4,458
		正例	負例	正例	負例	正例	負例	
学習データ	JNLP コーパス	282	151	65	368	31	402	1299
	BCCWJ	342	386	74	654	28	700	2184
	計	624	537	139	1022	59	1102	3483
開発データ	JNLP コーパス	82	41	16	107	16	107	369
	BCCWJ	78	97	21	154	5	170	525
	計	160	138	37	261	21	277	894
テストデータ	JNLP コーパス	19	8	7	20	5	22	81
総数		803	683	183	1303	85	1401	4458

コーパス増補後の「問題内容」コーパスに対して fine-tuning により学習した BERT モデルの学習率と epoch 数を表 14 に示す。

表 14 コーパス増補後の「問題内容」コーパスに対して学習した BERT モデルの各種パラメータ

分類器の番号	(1)-BERT		(2)-BERT
学習率	0.001		
epoch 数	26	A	抽 29
			前 0
			後 0
		B	抽 22
			前 20
			後 0
		C	抽 24
			前 1
			後 3

5. 実験結果

対象三文内でそれぞれ異なる分類器を作成する時、対象三文の分類器による判定が全て正しい時の精度を表中で「計」と表記する。コーパス増補前の分類器の正解率を表 15 に示す。

コーパス増補後の分類器の正解率を表 16 に示す。

表 15 コーパス増補前の分類器の正解率

分類器の番号	(1)-SVM	(1)-BERT		(2)-SVM	(2)-BERT
正解率	0.70	0.79	A	抽 0.56	抽 0.70
				前 0.89	前 0.85
				後 1.0	後 1.0
				計 0.48	計 0.63
			B	抽 0.63	抽 0.89
				前 0.89	前 0.85
				後 0.93	後 0.89
				計 0.59	計 0.67
			C	抽 0.70	抽 0.78
				前 0.85	前 0.85
				後 0.78	後 0.78
				計 0.48	計 0.48

表 16 コーパス増補後の分類器の正解率

分類器の番号	(1)-SVM	(1)-BERT		(2)-SVM	(2)-BERT
正解率	0.72	0.69	A	抽 0.78	抽 0.67
				前 0.93	前 0.85
				後 1.0	後 1.0
				計 0.74	計 0.52
			B	抽 0.67	抽 0.56
				前 0.85	前 0.89
				後 0.85	後 0.89
				計 0.52	計 0.41
			C	抽 0.85	抽 0.70
				前 0.74	前 0.74
				後 0.85	後 0.81
				計 0.52	計 0.30

6. 考察・展望

表 15 と表 16 を見比べると、(2)-SVM-B の分類器を除いて、SVM を用いた分類器では自己学習による精度の向上がみられる。一方、BERT を用いた分類器では、コーパス増補後は一律精度が下がっている。これには二つの可能性が考えられる。一つは、学習データに開発データを加えて再学習する手法が BERT の fine-tuning とは相性が悪い可能性が考えられる。学習データに開発データを加えて再学習する手法は、今回のように、学習データが非常に少ない

時に、学習データを増やす目的で利用され、平林ほか (2021) でも同様の手法をとり、今回の実験結果と同様、線形の SVM では効果があることが確認されている。しかし、BERT での fine-tuning では初めての試みであったため、これらの手法が有効であるか確認できていない。もう一つは、異なる分野での学習の影響を BERT での fine-tuning では強く受けてしまう可能性が考えられる。本論文では日本語の科学技術論文における「問題内容」のアノテーションを目指しているが、日本語の科学技術論文が十分に集まらなかったため、他分野コーパスを用いて精度向上を狙った。しかし、他分野コーパスの文が多い学習データで、日本語の科学技術論文のテストデータをあてるのが BERT での fine-tuning による分類器には難しいタスクであった可能性がある。

これらの可能性を踏まえ、増補分のコーパスの精度を人手でタグをつけることで、それぞれの分類器のタグ付け精度を求め、追加のコーパスを精度良く機械的にタグ付けを行えるシステムを探していく。

また、複数回実験を行うことによる正解率の差が大きいため、十分な実験数をとって追報告したい。

7. おわりに

本論文では、日本語科学技術論文中の「問題」を含む文及び、その前後一文を対象に、「問題内容」が含まれるか否かを人手でタグ付けしたコーパスを作成し、そのコーパスから作成される分類器の精度と、それをを用いてコーパスを増補し、自己学習を行い分類器の精度の向上を目指した。その結果、線形の SVM からなる分類器の精度の向上は概ね見られたが、BERT の fine-tuning からなる分類器の精度の向上をすることが出来なかった。自己学習の手法を見直し、さらなる分類器の精度の向上により、機械的に文に「問題内容」を含むかをタグ付けし、最終目標である日本語論文において「問題内容」と「解決法」のペアを効率的に取得するシステムの作成を目指したい。

謝辞

本研究は、国立国語研究所共同研究プロジェクト「コーパスアノテーションの拡張・統合・自動化に関する基礎研究」により作成されたコーパスを利用したものです。また、本研究は JSPS 科研費 18K11421、17KK0002 の助成を受けています。深く感謝いたします。

文献

- Kevin Heffernan, and Simone Teufel (2018). “Identifying problems and solutions in scientific text.” *Scientometrics*, 116:2, pp. 1367–1382.
- Fuchun Peng, and Andrew McCallum (2006). “Information extraction from research papers using conditional random fields.” *Information Processing & Management*, 42:4, pp. 963–979.
- Mike Scott (2001). “Mapping key words to problem and solution.” *Patterns of Text: in Honour of Michael Hoey. Benjamins, Amsterdam*, pp. 109–127.

- David Yarowsky (1995). “Unsupervised word sense disambiguation rivaling supervised methods.” *33rd annual meeting of the association for computational linguistics*, pp. 189–196.
- 鈴木類・古宮嘉那子・浅原正幸・佐々木稔・新納浩幸 (2019). 「概念辞書の類義語と分散表現を利用した教師なし all-words WSD」 *自然言語処理*, 26:2, pp. 361–379.
- 平林照雄・河野慎司・古宮嘉那子・新納浩幸 (2021). 「日本語の論文コーパスにおける「問題」の語義アノテーション」 *言語処理学会第 27 回年次大会*, pp. 1151–1155.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). “Bert: Pre-training of deep bidirectional transformers for language understanding.” *arXiv preprint arXiv:1810.04805*.
- Kikuo Maekawa, Makoto Yamazaki, Toshinobu Ogiso, Takehiko Maruyama, Hideki Ogura, Wakako Kashino, Hanae Koiso, Masaya Yamaguchi, Makiro Tanaka, and Yasuharu Den (2014). “Balanced corpus of contemporary written Japanese.” *Language resources and evaluation*, 48:2, pp. 345–371.