Title	自動獲得された因果関係知識に基づく文間の因果関係の推定
Author(s)	山田，涼太
Citation	
Issue Date	2021-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/17129
Rights	
Description	Supervisor:白井 清昭，先端科学技術研究科，修士(情報科学)

修士論文

自動獲得された因果関係知識に基づく文間の因果関係の推定

山田 涼太

主指導教員 白井 清昭

北陸先端科学技術大学院大学
先端科学技術研究科
（情報科学）

令和 3 年 3 月

Abstract

A causal relation is a relation of cause and effect between two events, such as "it rains" and "the ground gets wet." In other words, it is a relation in which an event written in one sentence likely cause an event written in another sentence. A collection of pairs of sentences under the causal relation is called a causality database, which is regarded as one of the common sense knowledge. A causality database is useful knowledge for natural language processing, and used for reasoning in text understanding and for showing evidence of users' evaluation in opinion mining. Studies in natural language processing on the causal relation include retrieval of sentence pairs under the causal relation from a large amount of text, and classification whether two given sentences have the causal relation or not. In this study, a model to perform the latter task is called the causality classification model. Most of the previous studies are based on supervised machine learning that requires labeled training data. However, in general, it is necessary to manually annotate sentence pairs with gold labels indicating whether the causal relation is held or not. It is very costly and time consuming to prepare a large amount of a manually labeled dataset. Therefore, it is preferable to construct training data for the causality classification automatically. The goal of this study is to obtain a causality classification model by supervised machine learning without manually labeled training data.

The proposed method consists of the following steps: "Initial data construction", "Unlabeled data construction", "Training of model", "Evaluation of model", "Causality classification", "Data selection and addition", and "Decision on stop".

First, in "Initial data construction" step, we use the corpus of Mainichi Shimbun news article data to automatically collect sentence pairs under the causal relation using some heuristics. We define a conjunction indicating the causality between two clauses as "causal keyword". Complex sentences including a causal keyword are supposed that two clauses in them have the causal relation. Then, such complex sentences are retrieved from news articles. Next, pairs of clauses (single sentences) connected by the causal keyword are extracted from these sentences. In this study, two conjunctions "から (*kara*)" and "ので (*node*)" are used as the causal keywords. In addition, pairs of sentences that are not under the causal relation are made by randomly shuffling those collected sentence pairs. The constructed initial data is divided into training data, development data, and validation data. In "Unlabeled data construction" step, we collect sentence pairs using the causal keyword "ため (*tame*)" in the same way, where some pairs of sentences have the causal relation and some do not.

The entire procedure of training of the causality classification model is performed by the bootstrapping method. In "Training of model" step, a model is trained using

Bidirectional Encoder Representations from Transformers (BERT) with the initial training data and the development data. The development data is used to optimize the parameters of the model. In "Evaluation of model" step, we apply the trained causality classification model to the validation data and measure its accuracy on the causality classification. In "Causality classification" step, we apply the trained causality classification model to each pair of sentences in the unlabeled data and determine whether they have the causal relation. In "Data selection and addition" step, we get the value of the output node in the BERT model as the reliability of the decision, select instances with the high reliability score, and add them to the training data. By repeating the above procedures, the number of the training data is increased incrementally. In "Decision on stop" step, the iterative procedure is terminated when the accuracy of the current model becomes worse than that of the previous model. Here the accuracy is measured on the validation data in the step of "Evaluation of model". Through the above iterative learning, a large amount of training data is constructed without manual annotation, and a highly precise causality classification model is obtained.

We conducted an experiment to evaluate the proposed method. First, the initial data consisting of 2,236 instances were constructed by our method. Next, the correlation between the reliability of the classification of the BERT model and the accuracy were investigated. It was confirmed that the accuracy became high as the reliability increased, and reached 0.906 at maximum. Next, the causality classification model was applied to the unlabeled data and the most reliable 2,000 instances were chosen and added to the training data at each iteration step. After three iteration steps, the number of the training data was increased from 2,236 to 8,236. We prepared 200 sentence pairs as the evaluation data. Two workers judged whether each pair had the causal relation or not. The inter-annotator agreement was 0.72, and the kappa coefficient was 0.44. When the judgments of two workers did not agree, they discussed and determined the final label. The causality classification model trained by the proposed method was applied to the evaluation data and the accuracy of the classification was measured. The accuracy of the model trained from the initial data only was 0.475, while the accuracy of the model trained from the training data after two iterations was improved to 0.520. However, after the third iteration, the accuracy decreased to 0.495.

Next, the precision, recall, and F-measure on retrieval of positive samples (sentence pairs under the causal relation) as well as negative samples (sentence pairs not under the causal relation) were measured. Through the iterative learning, the F-measure for the positive samples was declined, while that for the negative samples was improved. The initial model failed to classify the causality for the negative samples, but the errors were reduced by incremental enlargement of the training data by the proposed method. From these results, it was confirmed that the training data automatically acquired by the bootstrapping method contributed to improve the quality of the causality classification model. However, the accuracy of the causality classification was not high, 0.520. It should be improved.

One of the future work is to improve the automatic construction of the initial data. Although we supposed that sentences including the causality keyword always represented the causality, we found that it was not always true and some instances were wrongly created as the positive samples. Therefore, it is necessary to develop rules that precisely select pairs of sentences in which the causal relation is truly held. In addition, the method of creating unlabeled data to extend the training data needs to be improved. Although we collected data using "ため (*tame*)" as the causality keyword, it is insufficient to retrieve the cause-effect sentence pairs exhaustively. Some sentence pairs under the causal relation may include another causality keyword, and some may not include any keywords.

It is necessary to develop more patterns that can extract the sentence pairs under the causal relation.

概要

因果関係とは,「雨が降る」「地面が濡れる」のように，2つの事象の間に成立する原因と結果の関係である．一方の文の事象が起こることにより，もう一方の文の事象が起こりうる関係とも言える．このような因果関係を大量に集約した知識は因果関係のデータベースと呼ばれ，常識的な知識の一つと位置付けられる．因果関係のデータベースは自然言語処理に有用な情報であり，テキスト理解のための推論や，評判情報分析における根拠の提示などに活用ができる．因果関係に関する自然言語処理の研究として，大量のテキストから因果関係にある文の組を抽出したり，2つの文の間に因果関係が成立するか否かを判定することが行われている．本研究では，文間の因果関係の有無を判定するモデルを因果関係推定モデルと呼ぶ．その先行研究の多くは教師あり機械学習に基づくが，そのために訓練データを用意する必要がある．しかしながら，訓練データでは因果関係が成立するか否かをラベル付けする必要があり，一般にそのラベル付けは人手で行う．ここでの問題は，人手による因果関係の有無が付与された大規模なデータセットを用意するのは多大なコストと時間を要するということである．したがって，因果関係の訓練データは自動的に構築することが望ましい．本研究は，因果関係の推定モデルを人手で作成された因果関係データを必要としない方法で機械学習することを目的とする．

提案手法は,「初期データ作成」「ラベルなしデータ作成」「推定モデル学習」「因果関係判定」「推定モデル評価」「データの選別と追加」「終了判定」の手順で構成される．

初めに,「初期データ作成」では，毎日新聞の記事データをコーパスとして，ヒューリスティクスによって因果関係が成立する文の組を自動収集する．因果関係を示唆する接続詞を因果関係キーワードとし，これを含む複文は因果関係を表すとする．この文から因果関係キーワードで結ばれた節 (単文) の組を抽出する．本研究では「から」「ので」を因果関係キーワードとして用いる．また，収集された因果関係が成立する文の組をランダムに組み合わせ，因果関係が成立しない文の組も作成する．作成された初期データは，訓練データ，開発データ，検証データに分割する.「ラベルなしデータ作成」では，因果関係キーワード「ため」を用いて，同様に文の組を収集する．このとき，因果関係が成立する文の組としない文の組が混在する．

因果関係推定モデルの学習は，全体的にはブートストラップ法によって行う.「推定モデル学習」では，初期の訓練データと開発データを用いて，因果関係推定モデルを Bidirectional Encoder Representations from Transformers(BERT) を用いて学習する．開発データはモデルのパラメタの最適化に用いる.「推定モデル評価」では，学習した因果関係推定モデルを検証データに適用し，判定の正解率を測る.「因果関係判定」では，学習した因果関係推定モデルをラベルなしデータに適用し，因果関係の有無を判定する.「データの選別と追加」では，BERT の出力ノードの値を判定の信頼度として，信頼度が十分に高い事例を選別し，これを訓練データ

に追加する．上記の操作を繰り返し行うことで，訓練データを漸進的に増加させる.「終了判定」では，一つ前のステップで学習されたモデルと比べて正解率の向上が見られなければ学習を終了する．ここでの正解率は,「推定モデル評価」のモジュールで計測された検証データでの正解率である．以上の反復学習により，人手によるアノテーションなしで大量の訓練データを構築し，それにより精度の高い因果関係推定モデルを学習する．

提案手法の評価実験を行った．初期データとして2,236件のデータを得た．BERTによる判定の信頼度と精度の関係を調べたところ，信頼度が大きいと精度も向上し，最大で0.906まで達することを確認した．因果関係推定モデルをラベルなしデータに適用し，判定の信頼度の高いデータを1回のステップごとに2000件作成し，これを訓練データに追加した．この処理により，訓練データは初期の2,236件から3回の反復学習により8,236件まで増加した．評価用データとして200件の文の組を用意し，因果関係が成立するか否かを2名の作業者が判定した．2者の判定の一致率は0.72，κ係数は0.44であった．判定が一致しないときは，2名の作業者の合議により最終的なラベルを決定した．提案手法で学習された因果関係推定モデルを評価データに適用し，正解率を測った．初期データのみから学習されたモデルの正解率は0.475であったのに対し，反復学習を2回繰り返して得られた推定モデルの正解率は0.520まで向上した．しかし，3回目の反復学習で正解率は0.495と低下した．また，正例(因果関係が成立する文の組)もしくは負例(因果関係が成立しない文の組)を検索するタスクの精度，再現率，F値を測ると，反復学習により，正例のF値はやや低下していったが，負例のF値は向上した．初期モデルでは負例に対する判定を誤ることが多かったが，提案手法による訓練データの増強により誤りを減らすことができた．以上の結果から，ブートストラップ法によって自動獲得された訓練データが因果関係推定モデルの正解率の向上に寄与することを確認した．しかし，判定の正解率自体は0.520と高くはなく，改善の必要がある．

今後の課題として，初期データの自動構築方法の改善が挙げられる．因果関係キーワードを含む文を因果関係が成立するとして扱ったが，誤りも少なからず含まれていることが分かった．そのため，真に因果関係が成立する文の組を正確に選別するルールを開発することが必要である．また，訓練データを拡張するためのラベルなしデータの作成方法にも改善が求められる.「ため」を因果関係キーワードとしてデータを収集したが，因果関係が成立する文の中には，他のキーワードを含むものもあれば，因果関係キーワードがない場合もある．そのため，因果関係を表す文を網羅的に収集しているとは言えない．因果関係を表す文を検出できるより多くのパターンを考慮する必要がある．

目 次

図 目 次

表目次

第1章　はじめに

1.1　背景

因果関係とは,「雨が降る」「地面が濡れる」のように，2つの事象の間に成立する原因と結果の関係である．一方の文の事象が起こることにより，もう一方の文の事象が起こりうる関係のことである．このような因果関係を大量に集約した知識は因果関係のデータベースと呼ばれ，常識的な知識の一つと位置付けられる．因果関係のデータベースは自然言語処理に有用な情報であり，テキスト理解のための推論や，評判情報分析における根拠の提示などに活用ができる．

テキスト理解のための推論への活用とは，チャットや対話システムなどのテキストによって人と計算機がやり取りをするようなシステムで因果関係のデータベースを利用することを指す．このようなシステムでは，ユーザがその時主題として話している内容，あるいはユーザが求めていることを理解することで，ユーザに対して適切な提案をする，あるいは求めていることに関連する話題を新たな話題として提供することが求められる．この際，因果関係のデータベースを使った常識的な推論により，ユーザの意図や要求をより正確に把握することができる．

評判情報分析における活用とは，口コミサイトにおける知的ユーザインタフェースの構築に因果関係のデータベースを利用することを指す．口コミサイトでは，製品やサービスなどに対して，ユーザはその使用経験や意見などを口コミとして自由に書き込むことができるが，テキストにはユーザが何を思ってその口コミを書いたのかが明示的に書かれていない場合も多く，読み手は口コミを読んでユーザの真の意図を連想したり推測している．因果関係知識を用いることで，何が原因でそのような口コミを書くことになったかという根拠を自動的に推測し，それを提示することで，ユーザにとって利便性の高い口コミ閲覧システムを構築できる．

因果関係に関する自然言語処理の研究として，大量のテキストから因果関係にある文の組を抽出したり，2つの文の間に因果関係が成立するか否かを判定することが行われている．文間の因果関係の有無を判定するモデルは因果関係判定モデルと呼ばれる．その先行研究の多くは教師あり機械学習に基づくが，そのために訓練データを用意する必要がある．しかしながら，訓練データでは因果関係が成立するか否かをラベル付けする必要があり，一般にそのラベル付けは人手で行う．ここでの問題は，人手による因果関係の有無が付与された大規模なデータセットを用意するのは多大なコストと時間を要するということである．したがって，因果関係の訓練データは自動的に構築することが望ましい．

1.2　目的

本研究は，因果関係の判定モデルを人手で作成された因果関係データを必要としない方法で機械学習することを目的とする．因果関係の判定とは，入力として原因文と結果文に相当する 2 つの文を受け取り，原因文と結果文の間に因果関係が真に成立するかを判定することとする．ここで，原因文の事象が起こることで結果文の事象が起こりうる場合，2 つの文の間に因果関係が成立すると定義する．

最初に，確実に因果関係が成立する少量の文の組をキーワードによるパターンマッチングでコーパスから収集し，それらを初期の訓練データとして，因果関係の判定モデルを学習する．続いて，大量の未知の文の組に対して初期の判定モデルを適用し，それらが因果関係が成立するか否かを判定する．判定結果から，判定の信頼度が高い組を新たに訓練データに加え，判定モデルを再学習する．これを繰り返すことで判定モデルの精度を高める．

1.3　本論文の構成

本論文の構成は以下の通りである．2 章では関連研究を紹介し，本研究との違いについて述べる．3 章では提案手法について述べる．初期データの獲得，因果関係判定モデルの学習，訓練データの拡張について説明する．4 章では提案手法の評価について述べる．5 章では本論文のまとめと今後の課題を述べる．

第2章　関連研究

本章では本研究の関連研究について述べる．2.1 節では因果関係の出現特性に関する調査を紹介する．2.2 節では因果関係知識の獲得に関する研究について紹介する．手がかり標識を用いた手法，時系列情報を用いた手法，ブートストラップによる手法のそれぞれについて述べる．2.3 節では因果関係の推定に関する研究について紹介する．Support Vector Machine(SVM) を用いた手法，Multi-Column Convolutional Neural Network(MCNN) を用いた手法のそれぞれについて述べる．最後に，2.4 節では先行研究と本研究の違い，および本研究の特徴を述べる．

2.1　因果関係の出現特性に関する調査

まず，テキスト内に出現する因果関係がどのような特性を持っているかを調査した研究を紹介する．因果関係の特性に関する調査は，因果関係知識の自動獲得や文間の因果関係の有無を自動判定する研究に資するものである．

乾と奥村は，因果関係が文書内でどのように出現するかといった傾向を調査した [11]．まず，調査のために，因果関係がタグ付けされたコーパスを作成した．タグ付けは人間の主観によって行われたが，作業者間の主観の違いによるタグ付けの揺れが起こらないようにするため，言語テンプレートに基づく判断基準を採用している．言語テンプレートとは，2 つの事象 e_1,e_2 に対して，「e_1（という）状態になれば，それに伴い e_2（という）状態になる」といったものであり，各スロットに事象のテキストを入れた際に意味的に正しいと判断されれば因果関係が成立すると判断する．次に，作成されたコーパスを用いて因果関係の出現特性を調査した．因果関係が手がかり標識「ため」「ので」などとともに出現するときより，手がかり標識なしで出現することの方が多かった．また，因果関係において原因および結果を表す出来事が出現する統語カテゴリについて調査したところ，原因を表す出来事も結果を表す出来事も動詞句として表現されることが多いが，名詞句として表現されることも決して少なくなく，その割合は 4 割程度であった．したがって，因果関係知識を自動獲得する際には動詞句だけでなく名詞句も処理の対象として考慮すべきであると論じている．さらに，因果関係を表す出来事は，原因を表す出来事も結果を表す出来事も，文末もしくは文末に近い位置に出現することが多いこともわかった．

2.2　因果関係知識の獲得に関する研究

本節では因果関係知識の獲得に関する研究を紹介する．因果関係知識の獲得とは，大量のテキストから因果関係にある文の組を自動的に抽出することを指す．これにより大規模な因果関係データベースを自動的に構築することも可能となる．因果関係知識の自動獲得に関する先行研究は大きく 3 つに分けられる．2.2.1 項ではに手がかり標識による方法，2.2.2 項では時系列情報を用いた方法，2.2.3 項ではブートストラップによる方法を紹介する．

2.2.1　手がかり標識を用いた手法

因果関係を取得する方法として，手がかり標識を用いたものがある．手がかり標識とは「ため」「ので」のような因果関係の存在を明示的に示す単語のことであり，このような手がかり標識の周辺に出現する文や節は因果関係が成立する可能性が高い．

坂地らは手掛かり標識と構文情報を用いて原因文と結果文を抽出する手法を提案している [12]．ここで，原因文ならびに結果文とは，因果関係を構成する 2 つの文であり，因果関係における原因ならびに結果を表す文である．手がかり標識を起点として，その前後に出現する文を原因文や結果文として抽出するパターンを用意し，これを用いて因果関係を抽出する．ここでのパターンは構文情報も含んでいる．多くの場合，原因文は結果文より前に出現することが多いが，日本語の場合，構文が自由であるため，結果文が原因文よりも前に出現することもある．そのため，結果文が原因文よりも前に出現するパターンも用意する．実験の結果，原因文の抽出で 0.757，結果文の抽出で 0.526 の精度を得たと報告している．

佐藤と堀田は手がかり標識を用いて因果関係を自動抽出し，因果関係を有向グラフとして表した因果ネットワークを構築する手法を提案している [10]．まず，複文を単文に分割し，手がかり標識をもとに単文間の因果関係の有無を判定する．次に，単文の中から重要語を抽出し，単文の内容を重要語の組み合わせ (事象データと呼ぶ) で表現する．最後に，事象データをノード，それらの因果関係をリンクとする有向グラフを作成する．さらに，単文のモダリティを元に因果関係の強さを測り，リンクに付与する．このようにして構築されたグラフを因果ネットワークと呼ぶ．因果ネットワークにより様々な事象間の因果関係を視覚化することで，ある出来事と関連するニュース記事を表示するなどの活用が期待できる．また，ある事象が起こったとき，因果ネットワークを辿ることで，それからどのような別の事象が起こりうるかを推定することもできる．因果ネットワークを辿る際，ネットワーク上で隣接する原因の事象から必ず結果の事象が生じるわけではないことを考慮し，減衰関数 (具体的にはシグモイド関数) を導入することで，2 つの事象の因果関係の強さを見積もる．残された課題として，事象抽出の精度を向上させ

ること，因果関係ではない事象の組を誤って因果ネットワークに取り込むことを減らすことなどを挙げている．

2.2.2　時系列情報を用いた手法

因果関係は原因が起こった後に結果が起こるという時系列的な関係であると考えられる．小野と内海は，イベントに関する記述を抽出し，これらをイベントごとにクラスタリングし，イベントクラスタを時系列順に並べたデータに対してバースト検出を行うことで因果関係知識を獲得する手法を提案している [13]．バーストとは，データが急激に増加する現象のことであり，イベントが集中的に話題になった時期を特定できる．話題になった時期をイベントの発生時期とみなして時系列データとし，古いイベントの後に別のイベントが発生したとき，それらに因果関係がある (古いイベントが新しいイベントを引き起こす) とみなす．具体的には，格助詞でつながる名詞句と動詞句のペアをイベント表現とし，それらを出現する時系列順に並び変え，バースト検知によってイベントの発生時期を推測し，グレンジャー因果性検定を用いてイベント間の因果関係の有無を判定する．実験の結果，7,431 件の因果関係知識を獲得し，ランダムにサンプリングした 40 件を 5 人の判定者に評価させたところ，20 件は過半数の人が因果関係があると判断したと報告している．

2.2.3　ブートストラップによる手法

ブートストラップによって因果関係を抽出する研究が行われている．ブートストラップとは，一般に，少数のシードとなる事例から，事例を抽出するパターンやモデルを学習し，それを適用して新たな事例を獲得し，パターン・モデルの学習と新規事例の獲得を反復することで大量の事例を獲得する方法である．その代表的な例に Espresso アルゴリズムがある．Espresso[8] とは，関係抽出にブートストラップを適用したアルゴリズムである．ここで関係とは，特定の関係が成立する実体の組とする．例えば，抽出対象の関係が人物の職業のとき,「バイデン- 大統領」「隈研吾- 建築家」といった単語の組を抽出する．以下，関係が成立する単語の組を事例と呼ぶ．少量の事例をシードとし，これを含む文をコーパスから検索する．事例は特定のパターンで表現されることが多いため，検索された事例を含む文の集合に頻出する単語列を関係抽出のためのパターンとして自動的に獲得する．さらに，得られたパターンを用いて新しい事例を獲得する．これを繰り返すことで事例とパターンが同時に学習される．

新しいパターンを獲得する際，パターンの信頼度を計算し，それが高いパターンのみを採用する．パターン p の信頼度 $r_\pi(p)$ は式 (2.1) のように定義される．

$$r_\pi(p) = \frac{1}{|I|}\sum_{i \in I}\frac{pmi(i,p)}{max_{pmi}} \times r_l(i) \tag{2.1}$$

ここで，I は既に獲得した事例の集合，$pmi(i,p)$ は事例 i とパターン p が同時に現れる共起度であり，max_{pmi} はその最大値である．また，$r_l(i)$ は後述する事例の信頼度である．この式は，信頼度の高い事例を抽出できるパターンほど信頼度が高いという考えに基づく．

同様に，新たな事例を獲得する際には，事例の信頼度を計算し，それが高い事例のみを採用する．事例 i の信頼度 $r_l(i)$ は式 (2.2) のように定義する．

$$r_l(i) = \frac{1}{|P|}\sum_{i \in P}\frac{pmi(i,p)}{max_{pmi}} \times r_\pi(p) \tag{2.2}$$
$$r_l(i) = 1(i\text{ がシードの時})$$

この式は，信頼度の高いパターンから抽出された事例ほど信頼度が高いという考えに基づく．このようにパターンと事例の信頼度はお互いに依存しているが，初期状態ではシードのみが与えられており，全ての事例の信頼度は 1 となることから，最初に獲得するパターンの信頼度は式 (2.1) によって計算可能である．

Abe らは，Espresso アルゴリズムを応用し，特定の関係にあるイベントの組とそれを抽出するパターンを自動獲得する手法を提案した [2]．ここでイベントは動詞句で表現されるものとする．また，抽出の対象とする関係はユーザが指定できるため，この手法を用いて因果関係を表す動詞句の組を獲得することもできる．文章において重要な構成要素である動詞に着目し，動詞の目的語を引数として一般化することでイベントの組を抽出するパターンとした．2 つの動詞句を含む複文における動詞句の組を事例，目的語を変数に置き換えて一般化した動詞の構文をパターンとし，Espresso アルゴリズムによる反復学習を行った．実験では，action-effect (例は「運動する-汗をかく」) と action-means (例は「走る- 運動する」) という 2 つの関係を抽出した．20 回の反復を行うことにより，action-effect については 173,806 件の関係と 34,993 件のパターンを，action-means については 237,476 件の関係と 23,281 件のパターンを獲得した．また，サンプリングした 800 件の関係を 2 人の判定者によって正当性を判定させたところ，66%の精度が得られたと報告している．

また，Abe らは，パターンベースの手法とアンカーベースの手法の組み合わせによって特定の関係にあるイベントの組を収集する方法も提案している [3]．パターンベースの手法は前述の Espresso アルゴリズムに基づく手法であり，アンカーベースの手法は 2 つの文で項が共有されているときにイベントの組として抽出する手法である．パターンベースの手法は細かい関係の種類を識別できるが共有項を考えないという問題点がある．一方，アンカーベースの手法は共有項を考慮するが，関係の種類は識別できない (抽出する関係の種類を変更できない) という問題点が

ある．これらの手法はお互いに補完できることを指摘し，これらを組み合わせた手法を考案している．パターンベースの手法で文のペアを獲得し，アンカーベースの手法でこれらの共有項を特定する．実験の結果，パターンベースのみの手法でイベントを獲得した時と比べて再現率が上昇し，精度も最大で 81%に達したと報告している．

2.3　因果関係の推定に関する研究

因果関係の推定とは，ここでは与えられた 2 つの文の間に因果関係が成立するか否を判定することを指す．本節ではその関連研究を紹介する．いずれも教師あり機械学習に基づく手法である．

SVM を用いた手法

Hashimoto らは，文間の因果関係を推定するために Support Vector Machine(SVM) を用いた [5]．まず，単一の文から因果関係が成立する可能性のあるイベントの組を取得し訓練データとする．イベントの組は Web ページから取得し，文の構文情報をテンプレートとして，テンプレートに一致するものを因果関係の可能性のあるイベントの組として抽出した．この訓練データは 3 人の作業者によってアノテーションされた．アノテーションには 9 人月かかったと報告している．この訓練データを用いて SVM モデルを学習した．実験の結果，13%の再現率で 70%の精度を達成した．また，大量に取得した因果関係があるイベントの組の候補をサンプリングし，SVM をそれに適用したときの正解率から，2,451,254 の候補から 69,700 の因果関係のイベントの組を抽出できると推定している．

また，学習した SVM の応用として，得られた原因-結果の因果関係を推移律によって繋げることで，先に起こることを予測するシナリオを生成した．ある因果関係の結果文が別の因果関係の原因文と一致するとき，この 2 つの因果関係を繋げるが，そのときに単純な文字列の一致で結果文と原因文を結びつけようとすると，たとえ 2 つの文が同じ意味を表したとしても，表記の違いによって結びつけることができない．そこで，目的語が同じでかつ文のパターンが同じとき，2 つの因果関係をつなげる．因果関係を繋げてシナリオを作成した後，抽出元の文に単語の重複が存在しない場合は一貫性がないと判断して削除するなどのフィルタリングを行い，シナリオ作成の精度を高めている．実験では，68%の精度で 50,000 のシナリオを作成できたと報告している．

CNN を用いた手法

Kruengkrai らは，訓練データにおける原因文と結果文のそれぞれから得られた単語ベクトルや，著者らが別途構築した質問応答システムで検索された (質問に対

する) 回答から得られた情報から，多重畳み込みニューラルネットワーク (Multi-Column Convolutional Neural Network (MCMM)) を学習する手法を提案した [6]．因果関係の有無が付与された訓練データに加え，パターンを用いて自動収集された因果関係が成立する文の組，質問応答システムの回答から得られたデータ，因果関係を示唆するキーワード (「だから」「ので」など) を含む文といった 3 種類の背景知識も用いて因果関係推定モデルを学習している．訓練データは 3 人の判定者によるアノテーションによって構築した．MCNN を学習する際には，スキップグラムモデルで事前学習された 300 次元の単語埋め込みを用いた．実験の結果，この手法による判定の精度は最大で 55.13%であった．3 種類の背景知識がそれぞれどのように精度に寄与したかを確認したところ，いずれの背景知識も判定の精度向上に貢献し，すべての背景知識を用いた場合に精度が最も高くなったと報告している．

2.4　本研究の特徴

本節では本研究の特徴について述べる．関連研究との違いを説明し，本研究の特色を示す．

2.2 節で紹介した関連研究は，テキストから因果関係が成立する文の組を網羅的に収集し，因果関係の知識データベースを獲得することを目的とするのに対し，本研究では文間の因果関係の有無を判定することを目的としている点が異なる．坂地らの研究 [2] では因果関係の辞書を構築することを，佐藤と堀田の研究 [3] では因果ネットワークを構築することを目的としている．小野と内海の研究 [4] も，時間的に前後に発生するイベントの組を網羅的に収集することを目的としており，本研究の目的とは異なる．

ブートストラップの手法を用いた研究 [2, 3] もまた，因果関係 (正確には任意の関係) が成立する文の組を網羅的に収集することを目的としているが，ブートストラップの手法を用いるという点は本研究と共通している．ただし，ブートストラップの使い方は異なる．Abe らの研究 [6, 7] では，イベントの組とそれを抽出するパターンをブートストラップによって漸進的に獲得している．一方，本研究では因果関係判定モデルの訓練データ，すなわち因果関係が付与された文の組の集合を自動的に構築するためにブートストラップの手法を用いる．初期のラベル付きデータ (シード) から因果関係判定モデルを学習し，そのモデルを用いてラベルなしデータに対して因果関係の有無を判定する．判定の信頼度が十分大きいとき，それをラベル付きデータに追加する．これを繰り返すことで訓練データの量を漸進的に増加させる．

2.3 節で述べた因果関係の推定に関する研究に関しては，人手によるアノテーションを必要としない手法によって文間の因果関係を推定するモデルを学習する点に本研究の特徴がある．先行研究 [5, 6] では判定モデルを学習するための訓練データは人手で作成されているのに対し，本研究は人手によって作成された訓練

データを必要としない．すなわち，訓練データはブートストラップの手法によって自動構築し，シードも人手を介さず手がかり標識を用いて自動生成する．人手によるアノテーションは，それに要するコストや時間が大きいという問題に加え，文献 [11] で述べられている通り，人によって主観が異なるためにアノテーション結果の揺れが生じやすいという問題がある．これに対し，訓練データを自動構築するアプローチではそのような揺れが少なく，均質な訓練データを構築することが可能である．ブートストラップは自然言語処理でよく用いられる手法であるが，これを因果関係判定モデルの訓練データの構築に応用することは初めての試みである．

第3章　提案手法

3.1　概要

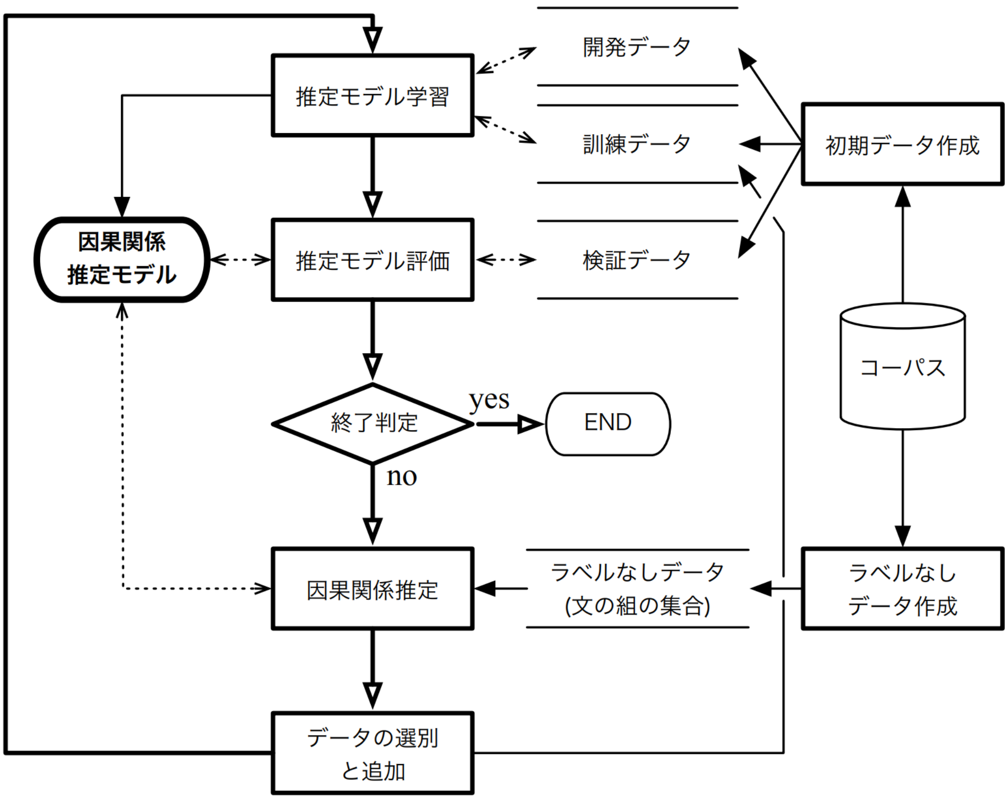
本論文で提案する手法の概要を図 3.1 に示す．初めに,「初期データ作成」では，コーパスからヒューリスティクスによって因果関係が成立する文の組を自動収集する．具体的には，因果関係を示すキーワードを含む文 (複文) を検索し，それから原因を表す文と結果を表す文を抽出する．また，収集された因果関係が成立する文の組を用いて，因果関係が成立しない文の組を作成する．このようにして，正例 (因果関係が成立する文の組) と負例 (因果関係が成立しない文の組) からなる初期データを作成する．作成された初期データは，開発データ，訓練データ，検証データに分割する．

「ラベルなしデータ作成」では，因果関係が成立するかが曖昧な因果関係キーワードを用いて文の組を収集する．収集された文の組は因果関係が成立しないものとしないものが混在する．これらの文の組はラベルなしのデータとして扱い，後の「因果関係判定」と「データの選別と追加」で用いる．

因果関係推定モデルの学習は，全体的にはブートストラップ法によって行う.「推定モデル学習」では，訓練データと開発データを用いて因果関係推定モデルを学習する．開発データはモデルのパラメタの最適化に用いる.「推定モデル評価」では，学習した因果関係推定モデルを検証データに適用し，判定の正解率を測る.「因果関係判定」では，学習した因果関係推定モデルをラベルなしデータに適用し，因果関係の有無を判定する.「データの選別と追加」では，判定の信頼度が十分に高い事例を選別し，これを訓練データに追加する．

上記の操作を繰り返し行うことで，訓練データを漸進的に増加させる.「終了判定」では，一つ前のステップで学習されたモデルと比べて正解率の向上が見られなければ学習を終了する．ここでの正解率は,「推定モデル評価」のモジュールで計測された検証データでの正解率である．

以上の反復学習により，人手によるアノテーションなしで大量の訓練データを構築し，それにより精度の高い因果関係推定モデルを学習する．



開発データ
推定モデル学習
訓練データ
初期データ作成
因果関係
推定モデル
推定モデル評価
検証データ
コーパス
終了判定
yes
END
no
因果関係推定
ラベルなしデータ
(文の組の集合)
ラベルなし
データ作成
データの選別
と追加

図 3.1: 提案手法の概要

3.2 初期データ作成

まず初めに，判定モデルを学習するための初期データを獲得する．この処理は「原因文と結果文の抽出」と「正例・負例の作成」に分けられる．

3.2.1 原因文と結果文の抽出

原因文と結果文は因果関係キーワードを用いてヒューリスティクスによって取得する．因果関係キーワードとは，ここでは原因を表す文と結果を表す文をつなぐ接続詞とする．本研究では「から」と「ので」を因果関係キーワードとする．これらの因果関係キーワードで結ばれた文には，因果関係が成立する可能性が高いと考える．以下に例を挙げる．

電車が止まったからバスが混む
雨が降ったので地面がぬかるんでいる
今日は晴れだからお出かけ日和だ

最初の文では,「電車が止まった」ことが原因で「バスが混む」ことが起こっている．同様に，2 番目の文では,「雨が降った」ことが原因で「地面がぬかるむ」という現象が発生している．3 番目の文でも,「今日は晴れだ」という状況が原因となって「お出かけ日和だ」という状況が発生している．

コーパスから因果関係が成立する可能性の高い文の組を抽出する．以下，因果関係が成立する文の組を (C, E, yes) と記す．C は原因文，E は結果文，yes は両者の間に因果関係が成立することを表すラベルである．(C, E, yes) は以下の手順で抽出する．

1. 因果関係キーワードを含む文の検出
2. 文節の係り受け解析
3. 原因文の抽出
4. 結果文の抽出
5. 短い文・記号を含む文の除外
6. 事例の作成

以下，それぞれの手続きの詳細を説明する.「因果関係キーワードを含む文の検出」では，まず文の形態素解析を行い，文を単語に分割し，個々の単語の品詞を同定する．形態素解析ツールとして MeCab[9] を用いる．接続詞「から」「ので」(因果関係キーワード) を含み，かつその直前が動詞または助動詞であるとき，その文を因果関係キーワードを含む文として抽出する．直前が動詞または助動詞である文に限定するのは以下のような文を除外するためである．

北海道から来た

上記の例文では，因果関係キーワード「から」の直前は名詞「北海道」である．この文では,「から」が原因ではなく，英語の from に相当するような場所の起点を表す．このように,「から」の直前が動詞ではないときは因果関係を表すことが少ないため，除外する．一方,「から」「ので」の直前が動詞のときは，因果関係を表すことが多い．また，因果関係キーワードの直前が助動詞である文を抽出するのは,「だから」の「だ」,「なので」の「な」が形態素解析によって助動詞と判定されるためである．

「文節の係り受け解析」では，CaboCha[7] を用いて文の文節の係り受け解析を行う．文における文節の境界，文節を構成する単語，個々の文節の係り先の文節,個々の文節の係り元文節のデータを作成し，以後の抽出処理に用いる．

「原因文の抽出」では，因果関係キーワードを含む文節に係り，かつその末尾が助詞である文節を抽出する．続けて，抽出した文節に係り，かつその末尾が助詞である文節も抽出する．この操作を再帰的に繰り返す．最後に，抽出した文節を連結し,「から」「ので」を削除して，原因文 C を得る．この処理で，文から余分な修飾語を省き，文の意味を表す上で重要な要素である動詞，格 (助詞)，格要素 (助詞の前に出現する名詞) から構成される文を原因文とする．

「結果文の抽出」では，最初に文末の単語を検出する．同様に，検出した単語を含む文節に係りかつ末尾が助詞である文節を再帰的に抽出する．ただし，因果関係キーワードより前に出現する文節は検出しない．抽出した文節を連結して結果文 E を得る.「原因文の抽出」と同じく，余分な修飾語を省く処理である．

「短い文・記号を含む文の除外」では，C，E のいずれかの文字数が 7 未満のとき，これを除外する．この処理は以下のような文を除外するためである．

練習を重ねてきたからだ

上記の例文では，因果関係キーワードの直前が動詞となっているため，原因文と結果文の抽出を試みる．原因文として抽出されるのは「練習を重ねてきた」であり，自然な文であるが，結果文として抽出されるのは「だ」のみであり，不自然な文である．7 文字未満の文を除外する処理で，このような意味を持たない短すぎる文を除外する．また，括弧などの記号が含まれている場合は除外する．括弧は他者のセリフの引用などに使われることがあり，そのときは因果関係キーワードがあっても文間の因果関係が成立しない可能性があるからである．

「事例の作成」では，これらの文に因果関係が成立するというラベル「yes」をつけ，(C, E, yes) という組を抽出する．

以下の文を例に，上記の手続きによる抽出の例を具体的に説明する．

パーティーだと人が多くて相手を知るのに苦労だけど，今回は少人数で長時間一緒にいたので相手を理解するのに大変役立った

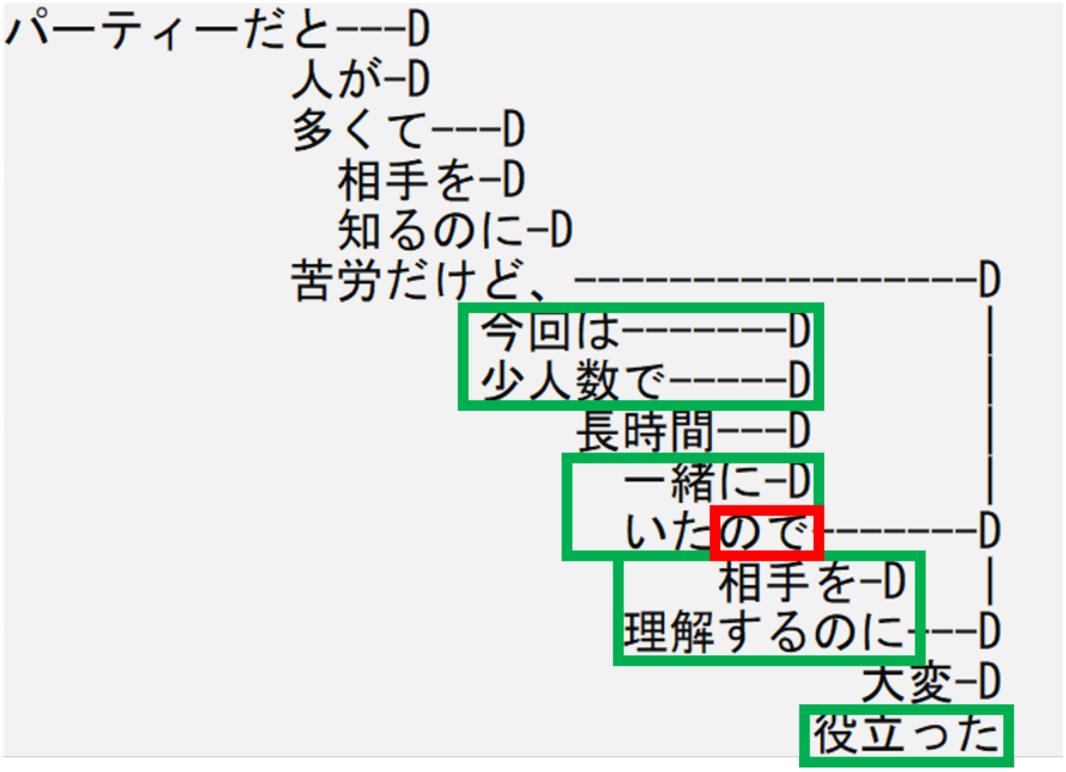


図 3.2: 因果関係抽出の例

図 3.2 は，上記の文を CaboCha によって文節の係り受け解析を行い，文節間の係り受け関係を示したグラフである．まず，因果関係キーワード「ので」を検出する.「ので」の前は「いた」という動詞なので，抽出の条件を満たす．次に，因果関係キーワードを含む「いたので」という文節に助詞を介して係る文節を抽出すると,「今回は」「少人数で」「一緒に」という 3 つの文節が該当する．一方,「長時間」という文節は排除される．これにより,「今回は少人数で一緒にいた」という原因文 C が抽出される．

次に，文末に出現する単語「役立った」を検出する．同様に,「役立った」という文節に助詞を介して係る文節を抽出すると,「理解するのに」という文節が該当する．また，この文節に助詞を介して係る「相手を」も抽出される．一方,「大変」という文節は除外される．また,「一緒に」より前の文節は，因果関係キーワードの前にあるため，抽出されない．したがって,「苦労だけど,」という文節は「役立った」という文節に直接係るが，抽出されない．その結果,「相手を理解するのに役立った」という結果文 E が抽出される．

3.2.2　正例・負例の作成

3.2.1 項で取得した文の組は，因果関係が成立する文の組である．しかしながら，因果関係を推定するモデルを学習するためには，因果関係が成立する文の組 (正例) だけではなく，成立しない文の組 (負例) も必要である．

負例は以下の手続きで作成する．

1. 先の手順で得られた正例の集合を $\{ (C_i, E_i, yes) \}$ と記す．
2. 原因文 C_i に対し，他の組の結果文 E_j $(i \neq j)$ の中からランダムに 1 つを選択し，負例 (C_i, E_j, no) を生成する．
3. この操作を全ての C_i について繰り返す．

これにより，正例と同じ数の負例が得られる．また，このように作成された正例と負例のデータセットでは，1 つの原因文に対し，因果関係が成立する結果文との組と，因果関係が成立しない結果文との組が必ず 1 つずつ存在する．

負例作成の例を示す．以下の文の組が因果関係キーワードを手がかりに得られた正例であるとする．

C_1: 絵本，ベビーベッドも用意している
E_1: 子ども連れでも安心できる

C_2: 筋肉のストレッチングは簡単だ
E_2: 続けていきたいと思います

この文の組に対して，原因文に対して別の文の組の結果文 E_j を組み合わせることで，以下のような因果関係が成立しない負例が作成される．

C_1: 絵本，ベビーベッドも用意している
E_3: チームワークを大切に大会に臨みたい

C_2: 筋肉のストレッチングは簡単だ
E_4: レシピを見ながら挑んだ

上記の手続きで作成された正例と負例の集合を初期データとする．初期データは，あらかじめ 8:1:1 の比率で，訓練データ，開発データ，検証データにランダムに分割する．この時，同じ原因文を持つ正例と負例は，同じデータ内に収まるようにする．したがって，訓練データ，開発データ，検証データにおける正例と負例の数は全て等しい．

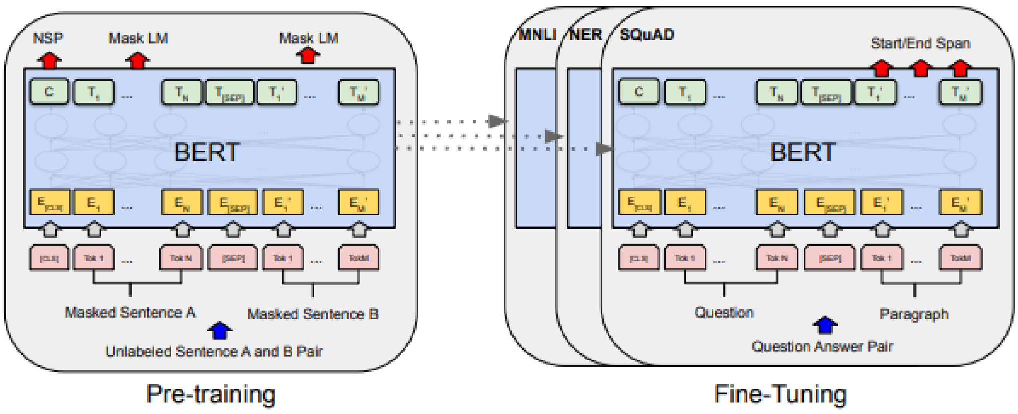


図 3.3: BERT の概要 [4]

3.3 因果関係推定モデルの学習

原因文 C と結果文 E の組が与えられたとき，それらの間に因果関係が成立するか否かを判定するモデルを学習する．このモデルは，因果関係が成立する場合に正，成立しない場合に負と判定する二値分類器である．因果関係推定モデルは Bidirectional Encoder Representations from Transformers (BERT)[4] を用いて学習する．近年，BERT は自然言語処理の分野において様々なタスクで高い成果が得られており，注目されている学習モデルである．特に，汎用性の高い言語モデルを学習できることが知られている．言語モデルとは，ここでは文の抽象表現 (ベクトル表現) を指し，汎用性が高い言語モデルとは様々なタスクに適した文の抽象表現を指す．

BERT での学習は図 3.3 に示す通り，pre-training(事前学習) と，fine-tuning(再学習) の 2 つのステップから構成される．pre-training は具体的な問題 (タスク) を解くモデルを学習する前に，あらかじめ文や単語の抽象表現を学習する処理である．事前学習済みのモデルは研究者によっていくつか公開されており，英語，日本語をはじめ様々な言語のモデルが存在する．fine-tuning は求められるタスクに合わせてモデルの調整を行う処理である．pre-training で自然言語処理を行う上で必要となる文法や意味といった基礎的な知識を包括的に学習し，fine-tuning でそれぞれのタスクに対して専門性を高める．pre-training と fine-tuning は同じアーキテクチャによって実装されており，pre-training の結果得られたモデルを初期値として fine-tuning を行う．また，同じアーキテクチャで様々なタスク (文の極性の分類，文間の同値関係の分類，固有表現抽出，質問応答など) に適用できるため，BERT は汎用性が高い．

BERT では，1 つの文を分類することもできるし，2 つの文の組を分類することもできる．ここでは 2 つの文の間に因果関係が成立するか否かを判定するため，BERT の入力は 2 つの文の組，すなわち原因文と結果文の組となる．BERT の学

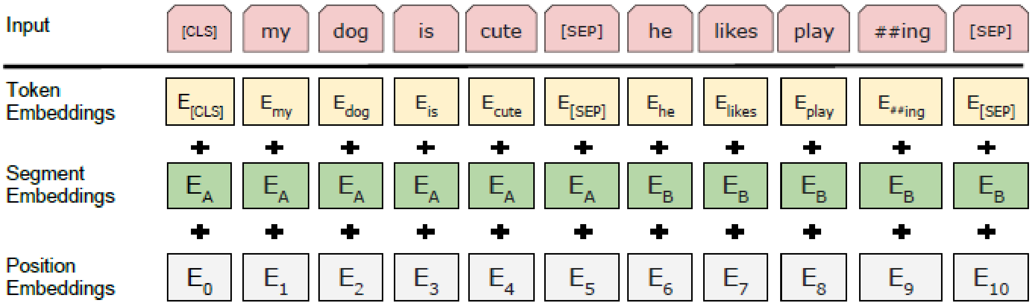


図 3.4: BERT での文の処理 [4]

習の際には，入力となる文の組のデータを以下のような系列に変換する．

$$\text{[CLS]}\ cw_1 \cdots cw_n\ \text{[SEP]}\ ew_1 \cdots ew_m\ \text{[SEP]}$$

[CLS] は文の組の分類のための抽象表現を得るためのトークン，[SEP] は 2 つの文の境界を示すトークン，cw_i は原因文を形態素解析して得られた単語，ew_i は結果文を形態素解析して得られた単語を表す．変換されたデータは図 3.4 のように単語ごとに分けられ，文脈情報を保持したままベクトル化される．それぞれの単語において，文中の単語の位置や複文であった場合にはどの文に含まれるかなどの情報を埋め込み表現として保持しているため，文法や文脈を学習することができる．

本研究では，BERT を学習する際には transformers-bert[1]をライブラリとして使用する．transformers は深層学習モデルであり，BERT の開発の元となったモデルである．

事前学習済みの言語モデルとして，日本語版 Wikipedia から事前学習され，京都大学によって公開されているモデル [1] を使用する．一方，fine-tuning のステップでは 3.2 節で得られた因果関係の有無のラベルが付与された訓練データと開発データを用いる．学習時のパラメータは以下の通りである．

- 学習率 (learning rate) 2^{-5}
- バッチサイズ 32
- 最大シーケンス長 128
- エポック数 10

3.4 訓練データの拡張

BERT によって因果関係推定モデルを学習後，訓練データを拡張する．その処理の流れを図 3.5 に示す．ラベルなしデータをあらかじめ用意しておき，そのデー

[1]https://github.com/huggingface/transformers

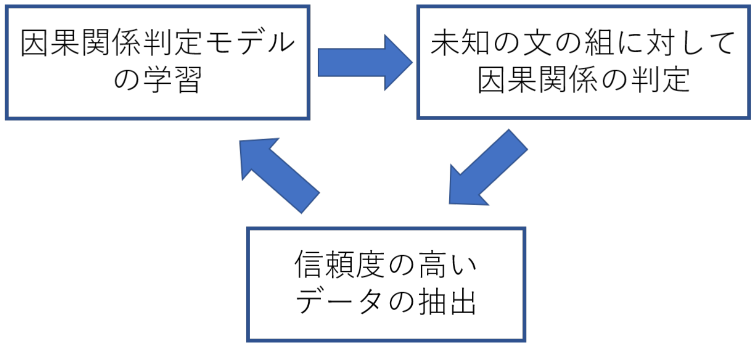


図 3.5: 訓練データの拡張のサイクル

タに対してその時点でのモデルを適用し，個々の文の組の因果関係の有無を判定する．また，判定の信頼度も算出する．そして，信頼度の高い事例を訓練データに新たに追加する．なお，図 3.1 おける開発データと検証データは拡張せず，ブートストラップの過程において常に初期データを用いる．

ラベルなしのデータは，接続詞「ため」をキーワードとし，初期データの作成と同様の手続きで原因文 C と結果文 E の候補の組を抽出する．「ため」をキーワードとして原因文と結果文を抽出した場合，キーワードの前後に出現する文は，以下の 2 つの文のように因果関係が成立する場合もあれば，しない場合もある．

文 1: 雪が降ったため遠足は中止になった
文 2: 学会で発表するため何回も練習した

文 1 では，キーワード「ため」の前の文「雪が降った」は，後の文「遠足は中止になった」の原因となっているため，因果関係が成立している．一方で，文 2 では「学会で発表する」は「何回も練習した」という行為の目的を表すため，因果関係は成立していない．このように，「ため」の前の文は原因を表すこともあれば目的を表すこともあるため，「ため」を因果関係キーワードとして用いて原因文と結果文の組を抽出することで，正例と負例が混在したデータが得られることが期待できる．

n 回目の反復ステップで学習された因果関係推定モデルを M_n と記す．すなわち，M_{n-1} を用いて訓練データを拡張し，拡張後のデータで M_n を再学習する．また，初期データから学習された因果関係推定モデルは M_0 とする．一方，n 回目の反復で訓練データ拡張のために用いるラベルなしデータの集合を U_n と記す．また，$U_n = \{\ (C_i, E_i, ?)\ \}$ とする．「?」は因果関係の有無が不明であることを表す記号である，この集合は因果関係判定モデルの n 回目のステップにおける反復学習のたびに，別のデータセットを用意する．すなわち，$U_n (n = 1, 2, 3, \ldots)$ は互いに異なるデータセットとする．

以下の手順で訓練データを拡張する．

1. 初期データから因果関係推定モデル M_0 を学習する．

2. モデル M_{n-1} を用いてラベルなしデータ U_n の文の組に対して因果関係の有無を判定する．

3. U_n の中から信頼度の大きいデータを選別し，訓練データとして追加する．

4. 上記の操作を繰り返す．

ラベルなしデータに対して因果関係の有無を判定するとき，その判定の信頼度も算出する．判定の信頼度は，ここではBERTによる因果関係推定モデルにおける出力ノードの値とする．BERTの出力ノードの値は，各クラスごとの確率分布である．今回の場合は正例と負例の2値分類であるため，正例である確率と負例である確率が出力される．確率の高い方が判定結果となる．

予備実験では，信頼度が上位のデータの多くが，因果関係推定モデルによって2つの文の間に因果関係が成立すると判定されていた．つまり，判定の信頼度が上位のデータのほとんどが正例であった．そのため，訓練データに追加するデータを作成する際に，正例と負例のバランスを取る．本研究では，初期データを正例と負例が同数となるように作成したのと同様に，因果関係推定モデルを学習するための訓練データは，正例と負例の間に偏りがない方が適していると考える．

具体的には，追加データの数を N_{add} と設定するとき，信頼度の大きい順に正例の数が $N_{add}/2$ 件に到達するまで追加データを取得する．この中に含まれる負例の数が N_{neg} のとき，$N_{add}/2 - N_{neg}$ 件の負例を新たに作成する．この負例は，初期データの作成時と同様に，ラベルなしデータ U_n の中から原因文と結果文をランダムに組み合わせて作成する．最終的に正例と負例の数が等しい N_{add} 件のデータを拡張データとし，これを訓練データに追加する．以降，拡張した訓練データを用いて因果関係推定モデルを再学習する．

上記で述べた訓練データの拡張の手続きをまとめる．

1. 初期データから因果関係推定モデル M_0 を学習する．

2. $n = 1$ とする．

3. モデル M_{n-1} を用いて，U_n 内の文の組 $(C_i, E_i, ?)$ に対して，因果関係の有無を判定する．同時に判定の信頼度も求める．

4. U_n の事例を判定の信頼度の大きい順に並べる．正例の数が $N_{add}/2$ 件になるまで，信頼度が大きい順に事例を選択する．これらのうち，正例の集合を P_n，負例の集合を N_n とおく．負例の数を $N_{neg}(= |N_n|)$ とする．

5. P_n の結果文を U_n における別の事例の結果文とランダムに入れ換えることにより，$N_{add}/2 - N_{neg}$ 個の負例を生成し，これを N_n に加える．

6. 正例の集合 P_n と負例の集合 N_n を訓練データに追加する．追加される事例の数は N_{add} 件である．

7. 拡張した訓練データと (初期の) 開発データを用いて，因果関係推定モデル M_n を再学習する．

8. モデル M_n の検証データにおける正解率が M_{n-1} の正解率より低いとき，処理を終了する．

9. $i \leftarrow i + 1$ とし，ステップ 3 に戻る．

第4章　評価

本章では，3 章で述べた提案手法の評価実験について述べる．4.1 節では初期データの作成について説明する．4.2 節では BERT による判定の信頼度の有用性について検証する．4.3 節では拡張データの評価を行う．4.4 節では提案手法によって学習された因果関係推定モデルを評価する．

4.1　初期データ獲得の結果

3.2 節で述べたように，コーパスから原因文と結果文を抽出し，正例と負例を作ることで初期データを獲得する．ここでは，その手法によって得られた初期データの結果を示す．

コーパスは，2009 年から 2013 年の毎日新聞の記事データを使用した．表 4.1 に各年の文の総数と因果関係キーワードを含む文の数を示す．新聞記事データを文に分割し，それぞれの文に因果関係キーワード「から」「ので」が含まれるかをチェックし，キーワードを含む文を抽出した．これを初期データを作成するための文の集合とした．これに対し，3.2 節で説明した手続きに従い，初期データを作成した．結果として，正例，負例が 1,398 件ずつ，合計 2,796 件の文の組からなる初期データを得た．この初期データを 8:1:1 に分割して，初期の訓練データ，開発データ，検証データを得た．その内訳を表 4.2 に示す．表 4.1 では「から」「ので」を含む文がおよそ 4 万件または 5 万件ほど得られているのに対し，初期データの数が少ないのは，3.2 節で述べた条件に合わない文が削除されたためである．因果関係キーワード「から」「ので」の直前が動詞ではない文のとき，原因文や結果文が短いとき，原因文や結果文が括弧を含むときは，初期データとして抽出していない．

4.2　信頼度の有用性の検証

3.4 節では，BERT モデルの出力ノードの値を判定の信頼度とし，これが高いデータを訓練データに追加する手法を提案した．しかしながら，BERT モデルの出力ノードの値が本当に判定の信頼度を表すのか，すなわち出力ノードの値が高いときほどその判定が正しい可能性が高くなっているのかについては，確認する必要がある．もし，判定の信頼度が信用できない値であるならば，これが高いデータを訓練データに追加する提案手法は妥当であるとは言えない．

表 4.1: 因果関係キーワードを含む文の抽出

	文の総数	から	ので
2009 年	2,302,031	9,335	11,842
2010 年	2,116,336	7,762	9,415
2011 年	2,192,350	7,814	10,233
2012 年	2,048,843	6,992	10,224
2013 年	1,993,419	6,827	9,687
合計	10,652,979	38,730	51,401

表 4.2: 初期データ

	訓練データ	開発データ	検証データ
正例数	1,118	140	140
負例数	1,118	140	140
全て	2,236	280	280

上記のことを検証するために，判定の信頼度と判定精度の相関関係を調査した．まず，判定の信頼度の閾値 t を設定する．あるデータに対して，判定の信頼度が t 以上のとき，そのデータに対する因果関係の有無を決定する．閾値 t を越えないときには，因果関係の有無は判定できないものとする．このときの判定の精度，すなわちモデルによって因果関係を推定することができたデータのうち，判定が正しかったものの割合を求める．閾値を t に設定したときの精度を P_t とする．その定義を式 (4.1) に示す．

$$P_t = \frac{\text{信頼度が } t \text{ 以上でかつ判定が正しいデータ数}}{\text{信頼度が } t \text{ 以上となるデータ数} } \tag{4.1}$$

t を変化させたときの精度 P_t の変動を調べる．判定の信頼度と精度の間に正の相関があるとき，すなわち t を大きく設定するほど P_t が高くなるとき，BERT モデルの出力ノードの値は判定の信頼度として妥当であると言える．

検証には因果関係推定モデル M_0 を用いる．これは，データを拡張する前の初期データから学習した因果関係推定モデルである，P_t を調べるためのデータは 4.1 節で説明した，自動作成によって得られた 280 件の検証データを用いる．閾値 t は，0.5 から 1.6 まで 0.1 刻みで変動させる．

結果を表 4.3 に示す．この表は，精度 P_t(2 列目)，閾値 t 以上のデータにおける正例と負例の数 (3 列目)，そのうち判定結果が正しかったときの正例と負例の数 (4 列目) を示す．2 列目の括弧内は，判定の信頼度が閾値 t 以上のデータ数と，そのうち判定が正しかったデータ数を示す．すなわち式 (4.1) の分子と分母に相当する値である．一方，表 4.3 に示した t と P_t の関係をグラフで示したのが図 4.1 である．

閾値 t が大きいほど判定の精度が高いことから，BERT モデルの出力ノードの値を判定の信頼度とすることは妥当であるといえる．閾値 t が 1.4 の時点で精度は

表 4.3: 判定の信頼度と判定精度の関係

t	精度 P_t	正例:負例 (全体)	正例:負例 (正解のみ)
0.5	0.692 (128/185)	99:86	71:57
0.6	0.710 (115/162)	91:71	69:46
0.7	0.717 (99/138)	85:53	65:34
0.8	0.733 (88/120)	76:44	62:26
0.9	0.787 (70/89)	63:26	57:13
1.0	0.797 (55/69)	58:11	54:1
1.1	0.839 (47/56)	47:9	47:0
1.2	0.860 (43/50)	43:7	43:0
1.3	0.897 (35/39)	35:4	35:0
1.4	0.906 (29/32)	29:3	29:0
1.5	0.895 (17/19)	17:2	17:0
1.6	1.000 (7/7)	7:0	7:0

0.9 を超えた．この精度は，自動拡張されたデータの品質を保証するためには十分に高いと言える．しかしながら，$t = 1.4$ のとき，閾値以上となるデータの数が 32 件に減っている．これは全データ数 280 件の 11%に相当する．また，判定の信頼度が高くなると，負例の数が正例の数よりもかなり少なくなる傾向も見られた．すなわち，判定の信頼度が高いとき，判定が正例に偏る傾向が強いことがわかった．

4.3 拡張データの評価

ここでは，ブートストラップ法によって拡張された訓練データを評価する．増加した訓練データの数など，ブートストラップ法による反復処理の詳細について報告する．

提案手法では，検証データの正解率が向上しなくなった時点で訓練データの追加を停止するが，今回の実験では試験的に反復回数を 3 回と設定する．

3.4 節で述べた通り，訓練データの拡張はラベルなしデータを用いて作成する．ラベルなしデータは，初期データと同様に毎日新聞の新聞記事データから獲得した．拡張データを作成するためのラベルなしデータを U_i とする．i は反復ステップの数を示す．本研究では U_1 は 2013 年，U_2 は 2012 年，U_3 は 2011 年の毎日新聞の記事データから獲得した．表 4.4 に各年の文の総数と抽出されたラベルなしデータの数を示す．

U_i はそれぞれ別の年のデータから取得したものであるため，互いに重なりはない．また，抽出に用いた因果関係キーワードが，初期データでは「から」と「ので」

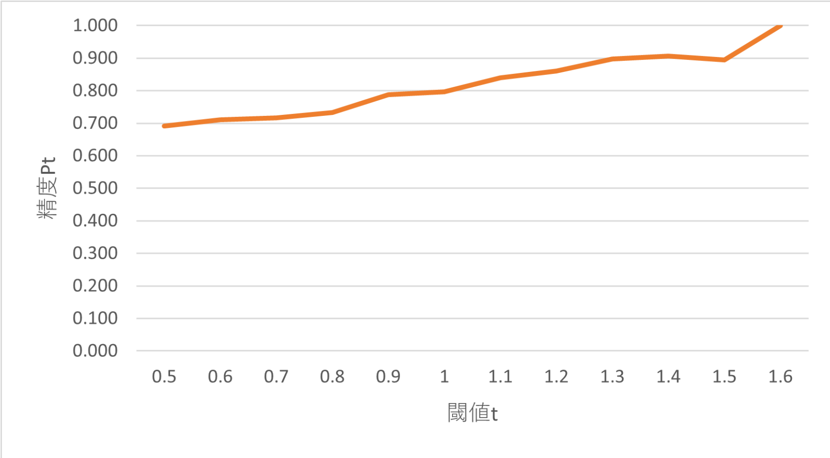


図 4.1: 判定の信頼度と判定精度の関係

表 4.4: ラベルなしデータの詳細

	発行年	文の総数	データ数
U_1	2013 年	1,993,419	5,581
U_2	2012 年	2,048,843	6,063
U_3	2011 年	2,192,350	6,350

であるのに対し，ラベルなしデータの作成に用いた因果関係キーワードは「ため」であり，因果関係キーワードが異なるため，U_i は初期データとも重なりはない．

表 4.5 は，それぞれの反復ステップ i について，ラベルなしデータ U_i の数 (3 列目)，訓練データの総数 (4 列目)，および検証データでの正解率 (5 列目) を示している．今回の実験では，一回の反復で追加するラベル付きデータの数 N_{add} を 2000 と設定した．そのため，各ステップごとに 2000 ずつ訓練データは増加し，初期データの 2,236 件から 8,236 件まで増加した．各ステップにおける N_{add} と N_{neg}(正例の数が $N_{add}/2$ 件になるまで判定の信頼度の上位のデータを取得したときに含まれる負例の数) の数を表 4.6 に示す．信頼度の高いものは，今回の実験ではすべて正例と判定されたため，N_{neg} は常に 0 であった．したがって，拡張データに追加する負例 1,000 件は，すべてランダムに文を組み合わせることで作成した．

検証データでの正解率は，1 回目の反復で向上し，0.657 となったが，それ以降は変動はあるものの，この正解率を越えることはなかった．このことから，訓練データを拡張することによって正解率の向上は見込めるが，拡張データの作成や追加の方法については検討の余地があることがわかった．拡張データのラベル付けは自動で行われているため，たとえ判定の信頼度が高いデータを選択したとしても，訓練データに追加する事例に誤りが含まれることを完全に妨げることはできない．例えば，表 4.3 で閾値を $t = 1.4$ と設定したときは，90%の確率で正しい結果が得られるが，残り 10%は誤りである．誤った事例が一度訓練データに追加

表 4.5: 因果関係推定モデルの反復学習の結果

i	モデル	U_i	訓練	正解率
0	M_0	–	2,236	0.639
1	M_1	5,581	4,236	0.657
2	M_2	6,063	6,236	0.636
3	M_3	6,350	8,236	0.650

表 4.6: 訓練データの反復拡張における N_{add} と N_{neg}

i	N_{add}	N_{neg}
1	2000	0
2	2000	0
3	2000	0

されると，それが及ぼす悪影響がモデルの反復学習を繰り返すたびに伝播し，結果として反復回数が増えると判定の正解率が低下すると考えられる．今回の実験では $Nadd = 2000$ と設定したが，これをもっと低い値に設定すれば，誤りの事例が追加される可能性を低くできる．あるいは，追加件数を設定するのではなく，信頼度が閾値以上のデータを追加するという方式も考えられる．一方，1 回の反復で追加する事例の数を減らすと，十分な量の訓練データが得られるまでに要する反復回数が増加し，結果として反復学習全体の計算時間が増大するというデメリットもある．

拡張データとして得られた正例の例を以下に示す．これらは，因果関係キーワード「ため」を含む文が抽出され，因果関係判定モデルで正例と判断されたものである．

C: 女性が告訴を取り下げた
E: 不起訴になっていた

C: ３月の募集では応募がなかった
E: 再募集することになった

拡張データとして得られた負例の例を以下に示す．既に述べたように，判定の信頼度の高いものは全て正例と判断されたため，これらの負例は結果文をランダムに選んで作成されたものである．

C: 受動喫煙を防止する
E: 収入が不安定です

C: 個人の観光客のニーズが多様化している
E: 入り口前で停車中に誤ってアクセルを踏んだとみている

表 4.7: 二者の判定の分割表

判定者 1 \ 判定者 2	因果関係あり	因果関係なし
因果関係あり	56	46
因果関係なし	10	88

表 4.8: 二者の判定の一致率と κ 係数

一致率	κ 係数
0.72	0.44

4.4 因果関係推定モデルの評価

ここでは，3.4 節で述べた因果関係推定モデルを評価する．最初に評価用のデータの作成について述べる．次に，これを用いて因果関係推定モデルを評価した結果を示す．最後にエラー分析を行い，提案手法の利点と欠点を論じる．

4.4.1 評価用データの作成

4.3 節では因果関係推定モデルの評価を 4.1 節で構築した検証データを用いて行っていたが，これは自動構築されたものであるため，モデルの性能を測るデータセットとしては必ずしも適切ではない．そのため，これとは別に，因果関係推定モデルの性能を正確に測るため，評価用データを人手で作成した．人手作成された評価用データを用いることで，自動構築した因果関係推定モデルによる判定が人による判定とどれだけ近いものであるかを客観的に評価できる．

評価用データは，初期データの作成と同様の手法を用いて抽出した．すなわち，因果関係キーワードを含む文から原因文と結果文の組を抽出した．接続詞「から」を因果関係キーワードとして抽出した文の組を 50 件,「ので」を因果関係キーワードとしたものを 50 件,「ため」を因果関係キーワードとしたものを 100 件，合計 200 件の文の組を評価用データとして抽出した．これらは常に因果関係が成立するわけではないことに注意していただきたい．また，評価用データは毎日新聞の 2008 年のデータから抽出した．初期データは 4.1 節で述べた通り 2009 年から 2013 年の記事データから抽出し，拡張データは 2011 年から 2013 年の記事データから抽出したため，評価用データとの重複はない．

こうして得られた 200 件の文の組に対して，2 名の作業者が独立に因果関係の有無をアノテーションした．二者の判定の分割表を表 4.7 に示し，判定の一致率と κ 係数を表 4.8 に示す．一致率は 72%とやや高いが，κ 係数は低い．二者で判定が分かれるのは，因果関係が成立するかを判定する際に，どれだけ常識的知識を使っ

表 4.9: 評価用データ

	正例	負例	合計
「から」	18	32	50
「ので」	18	32	50
「ため」	33	67	100
全て	69	131	200

てテキストそのものにはない情報を補うかに関して見解が分かれることが主な原因であった．以下に例を挙げる．

データ 1
C: ２歳だった
E: 原爆の記憶はない
データ 2
C: 届いたパソコンに全データを移していた
E: 致命的事態は避けられた

データ 1 では，年齢の情報によって因果関係が成立するかに対して意見が分かれた．1 名の判定者は 2 歳という幼ない年齢では記憶が残らないと判断し，因果関係があると判定したが，もう 1 名の判定者は 2 つの文に強い関連性がないと判断し，因果関係がないと判定した．データ 2 では,「致命的事態」に対して意見が分かれた．1 名の判定者は「致命的事態」が何に対しての致命的事態なのかがわからないため因果関係がないと判定したが，もう 1 名の判定者は原因文 C の中にパソコンという単語があり，致命的事態はパソコン上の何らかのシステムに関することであると考え，因果関係があると判定した．このように，文間の因果関係を判定する際に，どれだけ常識的知識を補完するかで見解が分かれた．

判定が異なる文の組については，著者 2 名の合議により最終的なラベルを決めた．テストデータの正例数は 69，負例数は 131 となった．各因果関係キーワードごとの正例と負例の数を表 4.9 に示す．

4.4.2 実験結果と考察

因果関係推定モデルを評価する際には，以下の 3 つの評価基準を用いる．

- 正解率
- 「因果関係あり」クラスに対する精度，再現率，F 値
- 「因果関係なし」クラスに対する精度，再現率，F 値

表 4.10: モデルによる予測と正解の分割表

モデルの予測結果 / 正解	正例	負例
正例	TP(True Positive)	FN(False Negative)
負例	FP(False Positive)	TN(True Negative)

正解率は，モデルによって予測された因果関係の有無の判定の結果が正解と一致する割合である．モデルの予測結果と真の結果 (正解) との対応関係を表 4.10 のように示す．TP は True Positive，FP は False Positive，FN は False Negative，TN は True Negative である．このとき，正解率は式 (4.2) のように定義される．

$$\text{正解率} = \frac{TP + TN}{TP + FP + FN + TN} \tag{4.2}$$

「因果関係あり」クラスに対する精度，再現率，F 値は，データセットの中から「因果関係あり」を正解ラベルに持つデータ，すなわち因果関係が成立する文の組を検索するタスクにおける精度 (precison)，再現率 (recall)，F 値 (F-measure) である．以下，それぞれを P_{pos}，R_{pos}，F_{pos} という記号で表す．pos は正例 (postivie sample) に対する指標であることを表す．これらは因果関係が成立する文の組をどれだけ正確に分類できるかを表す指標である．P_{pos}，R_{pos}，F_{pos} の定義を式 (4.3), (4.4)，(4.5) にそれぞれ示す．

$$P_{pos} = \frac{TP}{TP + FP} \tag{4.3}$$

$$R_{pos} = \frac{TP}{TP + FN} \tag{4.4}$$

$$F_{pos} = \frac{2 \cdot P_{pos} \cdot R_{pos}}{P_{pos} + R_{pos}} \tag{4.5}$$

「因果関係なし」クラスに対する精度，再現率，F 値は，データセットの中から「因果関係なし」を正解ラベルに持つデータ，すなわち因果関係が成立しない文の組を検索するタスクにおける精度 (precision)，再現率 (recall)，F 値 (F-measure) である．以下，それぞれを P_{neg}，R_{neg}，F_{neg} という記号で表す．neg は負例 (negative sample) に対する指標であることを表す．これらは因果関係が成立しない文の組をどれだけ正確に分類できるかを表す指標である．P_{neg}，R_{neg}，F_{neg} の定義を式 (4.6)，(4.7)，(4.8) にそれぞれ示す．

$$P_{neg} = \frac{TN}{TN + FN} \tag{4.6}$$

$$R_{neg} = \frac{TN}{TN + FP} \tag{4.7}$$

$$F_{neg} = \frac{2 \cdot P_{neg} \cdot R_{neg}}{P_{neg} + R_{neg}} \tag{4.8}$$

因果関係推定モデルによって 4.4.1 項で述べた評価用データの因果関係の有無を判定し，モデルの性能を上記の評価指標で測った．実験に用いた因果関係推定モデルは，初期データから学習されたモデル M_0，および拡張データを用いて学習されたモデル M_1，M_2，M_3 である．結果を表 4.11 に示す．また，図 4.2 は正解率，図 4.3 は「因果関係あり」クラスに対する精度，再現率，F 値，図 4.4 は「因果関係なし」クラスに対する精度，再現率，F 値をグラフで示したものである．

正解率は 1 回目と 2 回目の反復で向上し，3 回目の反復で減少した．反復回数が 2 のときに最大で，0.520 となった．初期データのみから学習したモデル M_0 と比べて 0.045 ポイント上昇したことから，訓練データの拡張は効果があることが確認された．正解率自体は 5 割程度であり，2 人の判定者による判定の一致率 72%と比べても決して高くなく，改善が必要である．また，すべてのモデルにおいて自動作成した検証データでの結果 (表 4.5) よりも正解率は低くなった．これは検証データと評価用データにおける負例の違いに起因するものと考えられる．検証データでは，負例はランダムに選択した文の組であるため，2 つの文は全く無関係なことがほとんどであり，因果関係を持つ正例との識別が比較的容易であると考えられる．一方，評価データでは正例も負例も因果関係キーワードの前後に出現する文であり，因果関係以外の何らかの関係を持っている可能性が高く，正例と負例の識別が難しいと推測される．

「因果関係あり」クラスの F 値は反復が進むにつれて低下するが,「因果関係なし」クラスの F 値は反復回数が 2 までは向上する．このことから，訓練データの拡張は，因果関係が成立しない文の組に対して正しく判定ができるようになる効果が大きいと言える．

因果関係推定モデルの反復学習の過程における正解率の変動において，正解率が一番高くなるのは，表 4.5 の検証データの正解率では反復回数が 1 のとき，表 4.11 の評価用データの正解率では反復回数が 2 のときと，一致していない．これは，検証データと評価用データの性質が異なることが原因のひとつと考えられる．検証データはコーパスから自動作成されたものであり，評価用データは人手でラベル付けされているという点で性質が異なる．また，検証データは因果関係キーワードが「から」「ので」である文から作成されているが，評価用データは「から」「ので」「ため」を含む文から作成されていることも異なる点である．

表 4.11: 因果関係推定モデルの評価

モデル		M_0	M_1	M_2	M_3
正解率		0.475	0.515	0.520	0.495
因果関係あり	精度 (P_{pos})	0.368	0.383	0.378	0.364
	再現率 (R_{pos})	0.725	0.667	0.609	0.623
	F 値 (F_{pos})	0.488	0.487	0.467	0.460
因果関係なし	精度 (P_{neg})	0.703	0.713	0.697	0.683
	再現率 (R_{neg})	0.344	0.435	0.473	0.427
	F 値 (F_{neg})	0.462	0.540	0.564	0.526

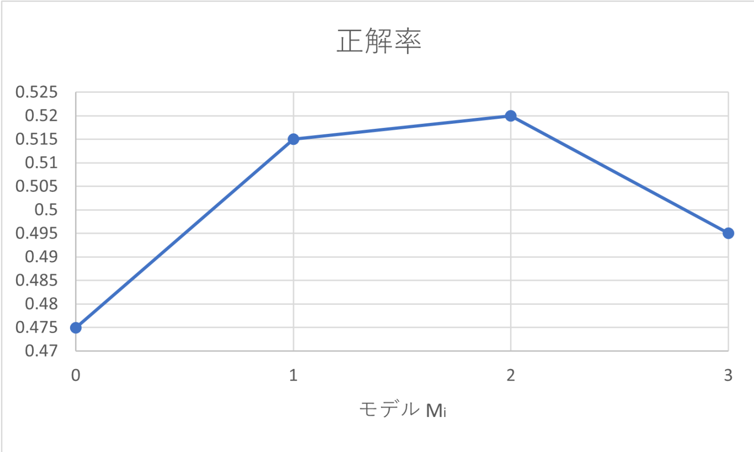


図 4.2: 正解率

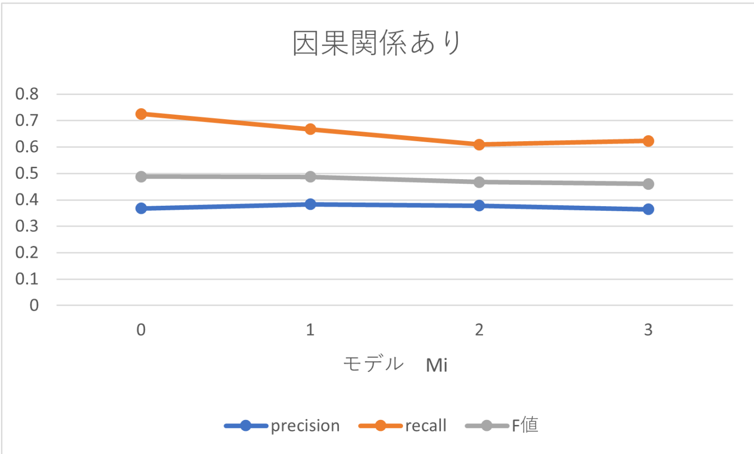


図 4.3: 「因果関係あり」クラスに対する精度，再現率，F 値

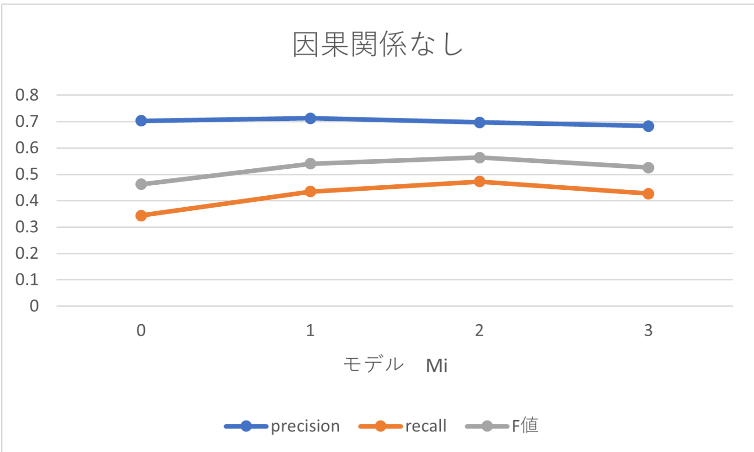


図 4.4: 「因果関係なし」クラスに対する精度，再現率，F 値

4.4.3 エラー分析

この節では因果関係推定モデルによる判定のエラー分析を行う．評価用データに含まれる事例を以下の 4 つの場合に分類し，それぞれ分析を行う．

1. システムの予測が因果関係あり，正解が因果関係あり (TP)
2. システムの予測が因果関係なし，正解が因果関係なし (TN)
3. システムの予測が因果関係あり，正解が因果関係なし (FP)
4. システムの予測が因果関係なし，正解が因果関係あり (FN)

1，2 はシステムの判定と正解が一致したもの，3，4 はシステムの判定と正解が一致しなかったものである．エラー分析に用いた因果関係推定モデルは，精度が最も高かった 2 回目のモデル M_2 を用いる．システムの判定と正解が一致した事例を表 4.12，一致しなかった事例を表 4.13 に示す．

まず，モデルが正解した 1，2 の事例について分析する．表 4.12 の上部 (2〜6 行目) はシステムも正解も因果関係ありの事例である．これらが正解できたのは，2 つの文の間で文脈的なつながりがあると因果関係が成立することが学習できたからであると考えられる．4 行目の例では「業者」という単語が原因文と結果文両方に共通しており，6 行目の例では「景気悪化」と「不景気」といったほぼ同義の単語を含んでいるため，2 つの文の間の文脈的なつながりが明確であったことが正解した要因であると考えられる．また，3 行目の例では,「しんどい」に対して「のんびり」といった反義の単語があることで，同様に文間の文脈的なつながりによって因果関係があると判定されたと考えられる．

表 4.12 の下部 (7〜11 行目) はシステムも正解も因果関係なしの事例である．この事例では同じ単語は含まれておらず，文脈的なつながりを示す単語の組もないため，因果関係がないと判断されたものと思われる．

次に，モデルが判定を誤った 3，4 の事例について分析する．表 4.13 の上部 (2〜6 行目) は，正解が因果関係がなしであるのに因果関係があると判定した事例である．これらについて因果関係がないと作業者がアノテーションした理由は，情報の欠落によるものである．2 行目の例では,「もっと上に行けると思っていた」では何の上なのかがわからず，因果関係が成立するとは言い難い．4 行目の例では,結果文「直接道に相談した」は原因文とのつながりが読み取れない．しかし，5 行目の例の「難易度が増す」に対して「延期」という文脈的なつながりが想定可能な場合や，6 行目の例の「男性」と「人」のように上位下位関係の単語がある場合もあり，これらから 2 つの文の間に文脈的なつながりがあるとシステムが解釈したため，判定を誤ったものと思われる．

表 4.13 の下部 (7 行目から 11 行目) は，正解が「因果関係あり」であるのに因果関係がないと判定した事例である．上記の場合と逆で，原因文，結果文の内容から両者の因果関係から読み取れたため，因果関係があるとアノテーションされた

表 4.12: システムと正解が一致した事例

C	E	システム	正解
質問されて困ることもあります	常に勉強しなければいけない	1	1
長男は学校でしんどい思いをしていると思う	家ではのんびりさせてあげたい	1	1
業者を代えると品質管理が面倒になる	特定業者を指名したがるのでは	1	1
夜学に通ってました	無理が来たのかもしれません	1	1
景気悪化の影響も遅れて現れる	不景気の波は年明け以降になるのだろう	1	1
決勝の相手は予選で負けたチームだった	一段とうれしい	0	0
届いたパソコンに全データを移していた	致命的事態は避けられた	0	0
テレビ番組はハードディスクに録画する	ＤＶＤは持っていない	0	0
私の自宅は山商の近くだ	選手の皆さんがランニングしている姿を見かけます	0	0
内村が踏ん張っていた	楽にさせてやりたかった	0	0

ものである．アノテーション時には 8 行目の例の「コンクリート」と「部材」，9 行目の例の「10 時」と「朝寝坊」といった互いに関連する単語があるため，常識的な知識で情報を補完しなくても因果関係が成立するとした．しかし，10 行目の例の「頑張ってきた」と「最高」，11 行目の例の「人間の技」と「重文 (重要文化財)」のような，文脈的なつながりが判断しにくいものもあり，これらに対してシステムが関連性を見い出すことができなかったため，因果関係なしと誤判定したと思われる．

表 4.13: システムと正解が一致しなかった事例

C	E	システム	正解
もっと上に行けると思っていた	悔いが残る終わり方だ	1	0
左手を角材に添えて右手ののみで彫る	小山さんの左手は傷だらけだ	1	0
１０月には医師の診断が別の医師の診断と異なった	直接道に相談した	1	0
工事の難易度が増す	延長せざるを得なくなったという	1	0
客層が中高年の男性に集中しがちだ	人にも来てもらおうと支店を開いた	1	0
夏から選手間でミーティングを繰り返して改善点を話し合ってきた	チームの団結力は強い	0	1
原子炉補助建屋でコンクリートの強度不足が分かった	他の部材も状態を慎重に調べる	0	1
インターネットの接続工事を１０時に頼んである	それまではゆっくりと朝寝坊しよう	0	1
全国大会に行こうと頑張ってきた	最高の夏だった	0	1
そこに人間の技が加わった	重文になったのです	0	1

第5章　おわりに

5.1　まとめ

本論文は，ブートストラップの手法を用いて，人手によるアノテーションなしに文間の因果関係を推定するモデルを学習する手法を提案した．

まず，毎日新聞の記事データから，因果関係の指標となるキーワードとして「から」と「ので」を使用し，因果関係が成立する可能性の高い文を抽出した．文節の係り受け解析を行い，その結果を元に余分な修飾語などを省略し，動詞，格 (助詞)，格要素 (名詞) から構成された短縮された文を抽出した．抽出した原因文と結果文の組は正例 (因果関係が成立する文の組) とした．一方，原因文に対して結果文をランダムに選んだ文の組を作成し，これを負例 (因果関係が成立していない文の組) とした．以上の正例と負例を合わせて初期データを作成した．

次に，因果関係を示唆するキーワードとして「ため」を使用し，同様の手続きでラベルなしのデータを抽出した.「ため」の前に出現する文は,「ため」の後に出現する文の原因を表す (因果関係である) こともあれば目的を表す (因果関係ではない) こともあるため，正例と負例が混在したデータが得られた．ラベルなしデータはブートストラップの反復ステップ毎に別のデータを用意した．

次に，初期データを用いて因果関係判定モデルを学習した．判定モデルの学習には BERT を用いた．これをラベルなしにデータに適用し，因果関係が成立するか否かを判定し，またその判定の信頼度を算出した．判定の信頼度の上位のデータから正例と負例をそれぞれ 1000 件取得し，訓練データに追加した．これを繰り返すことで，訓練データ量の増加と因果関係判定モデルによる判定精度の向上を図った．

提案手法を評価する実験を行ったところ，初期データのみから学習されたモデルの正解率は 0.475 であったのに対し，反復学習を 2 回繰り返して得られた判定モデルの正解率は 0.520 まで向上した．また，負例の F 値が向上しており，初期モデルは負例に対する判定を誤ることが多かったが，モデルの反復学習によりこれが改善された．このことから，ブートストラップ法によって自動獲得された訓練データが推定モデルの正解率の向上に寄与することを確認した．しかし，判定の正解率自体は 0.520 とは高くはなく，改善の必要がある．

5.2 今後の課題

最後に本研究の今後の課題について述べる．

まず，自動構築した初期データの品質が十分高いのかに疑問が残る．初期データを作成する際には，キーワードを含む文は必ず正例になると考えた．しかし，初期データを精査すると，正例の中には誤りも少なからず含まれていることが分かった．そのため，真に因果関係が成立する文の組をより正確に選別するルールを開発することが喫緊の課題として挙げられる．同様に，負例についても正確に作成する必要がある．本研究では負例をランダムに文を組み合わせて作成した．ランダムに選択された文の組はほとんど無関係であり，因果関係が成立する文の組と比較的容易に識別できると考えられる．一方，実際に文間の因果関係を判定する場面では，因果関係が成立しなくても，互いに関連がある文の組が判定の対象となることが多いと考えられる．つまり，容易に正例と負例を識別できるデータから学習された判定モデルは，識別が難しい実際の文の組に対して正確に働かない可能性がある．また，ランダムに文を選ぶことで2つの文は無関係となり，因果関係は成立しないと考えたが，因果関係が偶然成立する可能性を否定できない．より適切な負例の作成方法の探究も今後の重要な課題である．

次に，訓練データを拡張するために用いるラベルなしデータの作成方法にも改善が求められる．本研究では，ラベルなしデータは「ため」を因果関係キーワードとして作成した．キーワードを1つに限定したため，収集した文の組の多様性が乏しい可能性がある．因果関係は，本研究で用いた「から」「ので」「ため」以外にも，別の標識とともに出現することもあるし，標識なしで出現することもある．一方，因果関係の有無を判定するモデルを機械学習するためには，その訓練データには様々なタイプの因果関係が含まれていることが望ましい．また，初期データは「から」と「ので」をキーワードとしていたため，ラベルなしデータとは言語的特徴が異なる因果関係が含まれていると考えられる．初期データと拡張に用いるラベルなしデータの性質の違いがブートストラップによる因果関係判定モデルの学習にどのような影響を与えるかは精査する必要がある．

関連図書

[1] BERT 日本語 pretrained モデル. http://nlp.ist.i.kyoto-u.ac.jp/index.php?ku_bert_japanese. (2020 年 12 月閲覧).

[2] Shuya Abe, Kentaro Inui, and Yuji Matsumoto. Acquiring event relation knowledge by learning cooccurrence patterns and fertilizing cooccurrence samples with verbal nouns. In *Proceedings of the Third International Joint Conference on Natural Language Processing*, pp. 497–504, 2008.

[3] Shuya Abe, Kentaro Inui, and Yuji Matsumoto. Two-phased event relation acquisition: Coupling the relation-oriented and argument-oriented approaches. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING 2008)*, pp. 1–8, 2008.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–4186, 2019.

[5] Chikara Hashimoto, Kentaro Torisawa, Julien Kloetzer, Motoki Sano, István Varga, Jong-Hoon Oh, and Yutaka Kidawara. Toward future scenario generation: Extracting event causality exploiting semantic relation, context, and association features. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 987–997, 2014.

[6] Canasai Kruengkrai, Kentaro Torisawa, Chikara Hashimoto, Julien Kloetzer, Jong-Hoon Oh, and Masahiro Tanaka. Improving event causality recognition with multiple background knowledge sources using multi-column convolutional neural networks. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 3466–3473, 2017.

[7] 工藤拓, 松本裕治. チャンキングの段階適用による日本語係り受け解析. 情報処理学会論文誌, Vol. 43, No. 6, pp. 1834–1842, 2002.

[8] Patrick Pantel and Marco Pennacchiotti. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pp. 113–120, 2006.

[9] Mecab: Yet another part-of-speech and morphological analyzer. http://taku910.github.io/mecab/.

[10] 佐藤岳文, 堀田昌英. Web マイニングを用いた因果ネットワークの自動構築手法の開発. 社会技術研究論文集 Vol.4, 2006, pp. 66–74, 2006.

[11] 乾孝司, 奥村学. 文書内に現れる因果関係の出現特性調査. 情報処理学会研究報告 = IPSJ SIG technical reports, pp. 81–88, 2005.

[12] 坂地泰紀, 竹内康介, 関根聡, 増山繁. 構文パターンを用いた因果関係の抽出. 言語処理学会第 14 回年次大会論文集, 2008, pp. 1144–1147, 2008.

[13] 小野博紀, 内海彰. イベントの時系列分析による因果関係知識の獲得. 人工知能学会論文誌 30 巻 1 号 B, pp. 66–74, 2015.