機械加工文書における用語入れ子構造とトリガワードを考慮した用語関係同時抽出

稲熊 陸[1] 小島 大[2] 東 孝幸[2] 三輪 誠[1] 古谷 克司[1] 佐々木 裕[1]
[1] 豊田工業大学 [2] 株式会社ジェイテクト

1 研究背景

機械加工はものづくり分野における基本技術の一つであるが，少子高齢化にともないものづくりに携わる技術者，技能者が引退するとともに，新規後継者が減少している．そのため技術者一人あたりの作業負担は増加し，次世代の技術者に熟練技術者と同程度の知識と技術が求められている．

熟練者同等の知識が求められる業務に工程の策定業務が挙げられる．策定業務の遂行のためには，一つの機械加工因子の変化によって，他の機械加工因子が受ける影響をあらかじめ可能な限り網羅する必要がある．機械加工因子とは工具の種類や材料の特性，加工条件と機械加工パラメータなどの機械加工に関わる全ての因子のことをいう．

機械加工因子は専門書や機械加工を対象とする学会で発表される論文，社内技術文書や学生向けの教科書など，文書形式で幅広く存在し，文書内で複雑な関係で結ばれたり，因子間で入れ子構造を形成したりすることが多い．技術者が文書を閲読し，機械加工因子とその関係を獲得することは熟練者同等の知識が必要となる点で属人的である．

そこで本研究では，文書に含まれる機械加工因子と関係の自動抽出を目的に，用語抽出と，関係抽出とのマルチタスク学習を行う深層学習モデルを提案する．用語抽出においては入れ子構造を考慮し，関係抽出においてはトリガワードを考慮したモデル設計を行い，機械加工技術文書における用語と関係の抽出精度の評価を行う．

2 関連研究

属人的方法からの脱却を目指し，機械加工分野の文書から自動で機械加工因子を獲得し，因子を関係という観点で集約する手法として，増田 [1] らが行った Support Vector Machine（SVM）を用いた関係抽出の研究がある．増田らはトリガワードと呼ばれる単語について「トリガワードの出現位置」と「トリガワードが修飾する用語の位置」を定義し，関係抽出の素性に加えた．トリガワードは「物理量を示す二用語間の変化を表す単語」で定義される．図 1 に具体的な例を示す．「切削速度」に対して「増加する」，切削温度に対して「増し」といった単語のことをいう．増田らの研究は機械加工因子をあらかじめ特定させた上での関係抽出であり，新しく発表される機械加工技術文書や論文に示されるような文書への転用には，最初に機械加工因子を獲得するタスクが残っている点で問題が残っている．

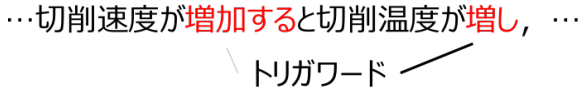

図 1 トリガワードの例

3 提案手法

本提案モデルは，日本語で記述された機械加工技術文書の入れ子構造を対象とした用語抽出タスクと関係抽出タスクを，マルチタスク学習で解く深層学習モデルである．図 2 にモデル全体の概要を示す．図中の格子状の四角はベクトルを表し，行方向に各トークンのベクトルの次元を，列方向にトークンの並びを表現している．また，五角形はぞれぞれ記述された演算処理を表現している．関係抽出タスクにおいては，トリガワードを Attenition 機構を用いて考慮したモデルとなっている．本提案モデルにおけるモデルの入力単位は一文である．

3.1 共有部

共有部（Shared Part）は用語抽出と関係抽出のどちらのタスクを学習する際にも用いられる構成要素である．与えられた各文を日本語版 Wikipedia で事前に学習された Sentencepiece[2] を用いてトークンに分割する．得られたサブワード単位のトークン群

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

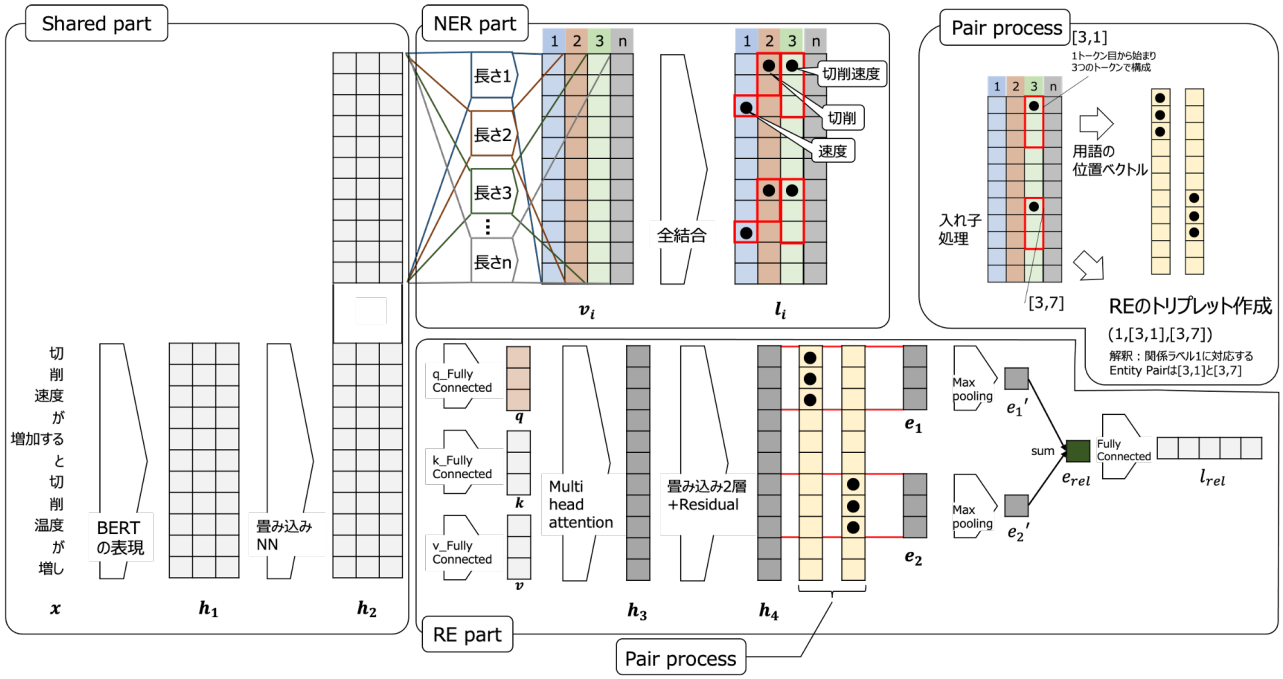

図 2　モデル全体の概要

$\boldsymbol{x}$ に対して日本語版 Wikipedia で事前に学習された BERT[3] を用いてトークンの表現 $\boldsymbol{h_1}$ を獲得する．

$$\boldsymbol{h_1} = \mathrm{BERT}(\boldsymbol{x}) \tag{1}$$

獲得したトークンの表現 $\boldsymbol{h_1}$ を CNN の入力とし，中間表現 $\boldsymbol{h_2}$ を獲得する．

$$\boldsymbol{h_2} = \mathrm{CNN}(\boldsymbol{h_1}) \tag{2}$$

中間表現 $\boldsymbol{h_2}$ は用語抽出と関係抽出のどちらのタスクを解く際にも用いる表現である．従って用語抽出と関係抽出それぞれで発生する損失を用いて $\boldsymbol{h_2}$ を生成するパラメータが更新される．この更新によってマルチタスク学習におけるそれぞれのタスクを解くための表現に新しく情報が追加され，学習が改善することが期待される．

3.2　用語抽出部

入れ子内部の用語と入れ子外部の用語の違いが用語を構成するトークンの数であることに注目し，本提案モデルはトークンの構成数毎に 2 値[1)]の出力をするモデルとなっている．図中の五角形の「長さ i」という演算は，長さ i のトークン列の表現を得る畳み込み演算である．以下の式で表される．なお，モデルは稲熊ら [4] のものを用いている．

$$\boldsymbol{v_i} = \mathrm{spanCNN}_i(\boldsymbol{h_2}) \qquad \forall i \in [1, \ldots, n_{tok}] \tag{3}$$

1)　「機械加工用語である」と「機械加工用語でない」の 2 値

ここで $\mathrm{spanCNN}_i$ はカーネルサイズ i のフィルタによる畳み込み演算を表す．n_{tok} は用語を構成するトークン数に対応しており，学習を行う前に与えるハイパーパラメタである．次に n_{tok} 個の中間表現 $\boldsymbol{v_i}$ に対して共通の全結合層をそれぞれ作用させる．式としては以下のようになる．

$$\boldsymbol{l_i} = \mathrm{FullyConnected}_{one}(\boldsymbol{v_i}) \tag{4}$$

全結合の出力は n_{tok} 個のカーネルサイズ毎に対応したトークン数長のシーケンスになる．出力の中身はカーネルサイズの用語を構成するトークンの開始位置に 1 が立つものである．損失は以下の式で表される．

$$L_{entity} = \sum_{i=1}^{n_{tok}} \sum_{j=1}^{l} y_i^j \log(l_i^j) \tag{5}$$

ここで n_{tok} はモデルの出力数すなわち用意するフィルタ数を表し，l は一文におけるトークン数を表す．各トークンに対する予測 $\hat{l}$ と正解 y との差を交差エントロピーで定義する．各フィルタに対応する出力の和をとったものを損失とする．

3.3　ペア作成部

ペア作成部（Pair Process）において，用語抽出で抽出された用語からペアを作成し，関係抽出用にデータ整形を行う．用語抽出で抽出された用語が入れ子構造を構成する場合は，一番大きい単位の用語

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

のみをペア作成の対象とする．

作られた各ペアに対してアノテーションファイルを参照して関係ラベルを付与し，（関係ラベル，用語 1，用語 2）のトリプレットを作成する．このトリプレットの用語 1 と用語 2 の情報を用いて関係ラベルを当てることが関係抽出タスクになる．ペア作成部では，用語 1 と用語 2 の用語位置ベクトルも作成する．用語位置ベクトルは一文のトークン数の長さの次元数が 1 のベクトルで，用語が存在する位置に 1，用語では無い位置には 0 が立つベクトルである．一文から n_{entity} 個の用語が抽出された場合，2 つの用語を選択してペアを作成するので ${}_{n_{entity}}\mathrm{C}_2$ 個のペアを作成する．

3.4 関係抽出部

関係抽出部はトリガワードを考慮した関係抽出モデルである．(2) 式の中間表現 $\boldsymbol{h_2}$ に対してトリガワードの情報を追加で与えて MultiHeadAttention 層の入力とし，関係を求めるモデルとなっている．最初にトリガワードの情報をアノテーションファイルから参照して $\boldsymbol{v_{trig}}$ を作成する．$\boldsymbol{v_{trig}}$ は長さが一文中のトークン長（SeqLen）で次元が 1 のベクトルである．対応するトークンに対してトークンがトリガワードの場合は 1 が入り，トリガワードではない場合には 0 が要素として定義されるベクトルである．この $\boldsymbol{v_{trig}}$ に対して平均 0 分散 1 の正規分布に基づいた重みで初期化される行列を用いた埋め込み処理を行う．

$$\boldsymbol{h_{trig}} = \mathrm{Embedding}\left(\boldsymbol{v_{trig}}\right) \tag{6}$$

この $\boldsymbol{h_{trig}}$ に対して中間表現 $\boldsymbol{h_2}$ を次元方向に Concat して全結合層に入力して $\boldsymbol{q}$ を生成する．

$$\boldsymbol{q} = \mathrm{FullyConnected}_q\left(\boldsymbol{h_2} \oplus \boldsymbol{h_{trig}}\right) \tag{7}$$

$\boldsymbol{k}$，$\boldsymbol{v}$ は以下の式に沿って作成し，$\boldsymbol{q}$，$\boldsymbol{k}$，$\boldsymbol{v}$ を MultiHeadAttention 層の入力とし，$\boldsymbol{h_3}$，$\boldsymbol{W}$ を生成する

$$\boldsymbol{k} = \mathrm{FullyConnected}_k(\boldsymbol{h_2}) \tag{8}$$

$$\boldsymbol{v} = \mathrm{FullyConnected}_v(\boldsymbol{h_2}) \tag{9}$$

$$\boldsymbol{h_3}, \boldsymbol{W} = \mathrm{MultiHead}(\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v}) \tag{10}$$

この Attention は query に対してトリガワードの情報を加えたものである．query に対しての情報付加なので「文中における関連度を知りたいもの」としてトリガワードを加えていることになる．key，value には $\boldsymbol{h_2}$ の情報が含まれていることを考慮すると，得られる $\boldsymbol{h_3}$ は「文全体におけるトークンの内トリガワードに関連するトークンに対して，強く注意がかかった表現」という解釈が可能である．これは「トリガワードが存在する場合，トリガワードと文全体の文脈との関連性」を考慮することで関係抽出の精度が向上することを期待している．

次に得られた $\boldsymbol{h_3}$ に対してスキップコネクションと畳込み層を 2 層を通して中間表現 $\boldsymbol{h_4}$ を作成する．

$$\boldsymbol{h_3'} = \mathrm{CNN}(\boldsymbol{h_3}) + \boldsymbol{h_3} \tag{11}$$

$$\boldsymbol{h_4} = \mathrm{CNN}(\boldsymbol{h_3'}) + \boldsymbol{h_3} \tag{12}$$

次にペア作成部で作成した用語位置ベクトルと $\boldsymbol{h_4}$ を用いて関係抽出に用いる 2 つの用語の表現 $\boldsymbol{e_1}$ と $\boldsymbol{e_2}$ を獲得する．

$$\boldsymbol{e_i} = \boldsymbol{pe_i} \odot \boldsymbol{h_4} \qquad \forall i \in [1, 2] \tag{13}$$

ここで $\boldsymbol{pe_i}$ は i 個目の用語位置ベクトル，$\odot$ は要素積を表す．得られた $\boldsymbol{e_1}$ と $\boldsymbol{e_2}$ に対して Maxpooling の演算を行いそれぞれの 1 つのベクトルに整形する．

$$\boldsymbol{e_i'} = \mathrm{Maxpooling}(\boldsymbol{e_i}) \qquad \forall i \in [1, 2] \tag{14}$$

ここで $\boldsymbol{e_i'}$ は複数トークンから構成される用語である．この $\boldsymbol{e_i'}$ をそれぞれ全結合層を入力して用語としての表現を獲得し，それらのベクトルの和をとってから最後に全結合層を通して用語間に存在する各関係の確率を出力する．以下の式で表される．

$$\boldsymbol{e_i''} = \mathrm{FullyConnected}\left(\boldsymbol{e_i'}\right) \qquad \forall i \in [1, 2] \tag{15}$$

$$\boldsymbol{e_{rel}} = \boldsymbol{e_1''} + \boldsymbol{e_2''} \tag{16}$$

$$\boldsymbol{l_{rel}} = \mathrm{FullyConnected}(\boldsymbol{e_{rel}}) \tag{17}$$

関係抽出を行うことによる一文の損失は以下のようになる．

$$L_{relation} = \sum^{{}_{n_{entity}}\mathrm{C}_2} \left(\sum_{i=1}^{r} y_i \log\left(\boldsymbol{l_{rel}}\right)_i \right) \tag{18}$$

ここで r は関係ラベルの種類数であり，n_{entity} は用語抽出が一文から抽出した用語数である．各関係に対して損失が発生し，それらの全ての和をとったものを関係抽出の損失とする．一文に対するマルチタスク学習モデル全体の損失としては，用語抽出の損失と合わせて以下のようになる．

$$L_{all} = L_{entity} + nL_{relation} \tag{19}$$

この損失をすべての文に対して，最小化することでマルチタスク学習を実現する．

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

表 1　関係抽出の対象ラベル数

	None	Relation	Negative	Positive	Sub
学習	12,028	710	76	90	78
評価	1,335	49	9	10	21

4　実験

4.1 節で今回実験に用いるデータについて述べ，4.2 節で実験設定について述べる．4.3 節ではマルチタスク学習モデルで行った用語抽出と関係抽出の実験について述べる．

4.1　対象データ

用いたデータは機械加工の切削加工に関する教科書文で文中に関係ラベルが存在する 794 の文を学習 714 と評価 80 に分割する．データ中の用語ラベルは「機械加工用語である」という Machining である．また，この 794 文の内トリガワードが含まれる文は 149 文で全体の 18.4%であった．関係ラベルは，用語 A と用語 B に「A が大きくなれば B が大きくなる」という正の相関を表す Positive の関係，「A が大きくなると B は小さくなる」という負の相関を表す Negative の関係，「A は B の一種である」という Sub の関係，「A は B と何らかの定性的な関係がある」という Relation の関係の四種類である．

4.2　実験設定

本節では行った実験の全体に共通する設定を述べる．最適化アルゴリズムは学習率 0.001 の Adam を用いた．用語抽出における用語のフィルタ数 n_{tok} は 4 とした．これは対象のデータにおける用語を構成するトークン数が四つまでで全体のトークンの 96%を越えることが分かったためである．

4.3　マルチタスク学習による関係抽出

タスクの難易度に差がある用語抽出タスクと関係抽出タスクにおいて，用語抽出と関係抽出の同時学習モデルによるマルチタスク学習を行うことによる関係抽出の精度評価実験について述べる．このマルチタスク学習において，学習初期から用語抽出と関係抽出を同時に行うことは精度が安定しない．その理由として，関係抽出の対象とする用語ペアが用語抽出で抽出された用語から作成されることが挙げられる．実際に，最初からマルチタスク学習を行うと，用語の抽出精度が安定していないため，関係抽出が対象とする用語ペアの数が多くなってしまうことがあり，GPU メモリに載らず，処理がそもそも不可能であった．そこでマルチタスク学習モデルにおける関係抽出は学習全体の半分は用語抽出のみを行い，半分以降は関係抽出も行うマルチタスク学習モデルとして評価を行った．表 2 に実験結果を示す．ベースラインモデルは用語抽出と関係抽出を別々で行ったモデルである．結果を見ると用語抽出と関係抽出どちらのタスクにおいてもマルチタスク学習モデルの方が高い精度が得られることが分かった．この結果から，マルチタスク学習モデルにおける共有部の中間表現 $\boldsymbol{h_2}$ が用語抽出・関係抽出の両タスクにとって必要な情報を獲得していることを示唆している．

表 2　学習法の違いによる評価データでの F 値

モデル	用語抽出部	関係抽出部
ベースライン	0.882	0.216
マルチタスク学習	0.898	0.320

5　おわりに

本研究は熟練者同等の知識が必要となる中で，技術文書等のテキストから機械加工因子と関係を抽出することを目指し，入れ子構造を考慮した用語抽出とトリガワードを考慮した関係抽出を同時に学習するマルチタスク学習モデルを提案した．マルチタスク学習をおこなうことで，用語抽出，関係抽出ともにベースラインより高い抽出精度が実現できることを示した．

今後の課題としては，マルチタスク学習モデルの学習方法についてのより詳細な検討，結果のより詳細な解析が挙げられる．

参考文献

[1] 増田ら. Trigger word と部分文字列を用いた機械加工用語の関係抽出. 言語処理学会 第 22 回年次大会 発表論文集, pp. 573–576, 2016.

[2] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *CoRR*, Vol. abs/1808.06226, , 2018.

[3] Yohei Kikuta. Bert pretrained model trained on japanese wikipedia articles. `https://github.com/yoheikikuta/bert-japanese`, 2019.

[4] 稲熊陸, 小島大, 東孝幸, 三輪誠, 古谷克司, 佐々木裕. 入れ子構造を考慮した機械加工用語抽出. 言語処理学会第 26 回年次大会 (NLP2020) P5-31, 2020.

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).