言語横断検索とリランキングを用いる翻訳メモリ利用型 NMT

田村拓也[1]　王小天[1]　宇津呂武仁[1]　永田昌明[2]
[1] 筑波大学大学院 システム情報工学研究群　[2]NTT コミュニケーション科学基礎研究所

概要

Hossain らが提案した retrieve-edit-rerank [1] は，翻訳や要約において (1) 外部情報から類似事例を検索し，(2) ニューラルモデルにより出力文候補を生成，(3) 予め定義したリランキングスコアによって最良の出力文を自動選択する枠組みである．本論文では，この枠組みにおいて (1) LaBSE や mSBERT を用いて生成した言語に依らない文埋め込みに基づいて類似訳文を言語横断検索する手法と (3) 文長に対する正規化を行ったより良いリランキング手法を提案し，有意に翻訳精度の向上を達成した．

1　はじめに

近年，高品質な対訳文の集合である翻訳メモリを NMT に組み込む研究が多くなされている．Bult é ら [2] [3] は翻訳メモリを NMT に組み込むことで翻訳精度を向上させる NFR モデルを提案した．このモデルは，編集距離や sent2vec [4] に基づいて翻訳メモリの原言語文集合から類似文を検索し，類似訳文を入力原言語文と連結して NMT モデルへの入力とする．このモデルは，NMT モデルへの入力を前処理するだけで良いため，モデルのアーキテクチャを変更せずに翻訳メモリを組み込むことができる．ゆえに，既存のあらゆる NMT モデルとの互換性が高く，実装面での移植性も高い．一方で，この手法では対訳の揃った翻訳メモリを対象に検索せねばならず，目的言語文のみで構成された大規模単言語コーパス (単言語翻訳メモリ) だけでは活用できない．また，入力文長の制約から，これらの手法で利用可能な類似訳文の数は高々数文に限られるため，有益な類似訳文がいくら得られたとしてもそれら全てを活用することはできない．

この制約を克服するため，Cai ら [5] は目的言語の単言語翻訳メモリを利用する手法を提案した．ここでは，Transformer Encoder に基づく検索モデルを提案し，MIPS (Maximum Inner Product Search) によって類似訳文検索を行う．ただし，検索モデルと翻訳モデルは同時に訓練される必要があるため独自のアーキテクチャを必要とし，NFR の利点である既存の NMT モデルへの互換性が大幅に低下する．

また，Hossain ら [1] は，多数の類似文を活用する手法として retrieve-edit-rerank の枠組みを提案した．彼らは，(1) 複数の相異なる類似文を検索して，(2) 異なる複数の出力文候補を生成し，(3) 対数尤度に基づいてリランキングを行う手法を提案した．本論文では，この retrieve-edit-rerank の枠組みを踏襲し，中でも (1) および (3) に着目して研究を行った．(1) では，言語に依らない文埋め込みを生成する事前学習済みモデル mSBERT・LaBSE を用いて目的言語文集合を対象に言語横断検索を行う手法を提案し，(3) リランキングフェーズでは，文長に対する正規化を行ったより良いランキング手法を提案する．

提案法の評価にあたり，ASPEC [6] 英日コーパス，および，EU Bookshop Corpus [7] の英仏コーパスを利用した．その結果，文埋め込みに基づく検索手法を用いた場合は，編集距離で検索する場合に対して有意に BLEU を向上させた．また，異なる類似訳文を用いて生成された異なる出力文候補のリランキングにおいても提案法は既存手法を大幅に上回った．

2　類似訳文検索 (Retrieve)

2.1　翻訳メモリからの類似文検索

翻訳メモリ (TM) は，あらかじめ人手で翻訳された高品質な対訳文の集合である．過去には，Computer-Aided Translation (CAT) など，人手翻訳を補助するツールとして活用され，近年ではニューラル機械翻訳（NMT）へ組み込むことが研究されている．翻訳メモリを用いると，翻訳したい原言語文が既に翻訳に格納された文である場合に，その訳文に置き換えるだけで誤りなく翻訳することができる．また，完全に一致した文が存在せずとも類似度がある程度高ければ，その訳文は翻訳時に参考になる可能性がある．ここで，入力原言語文に類似した原言語文を「類似文」，それに類似した目的言語文を「類

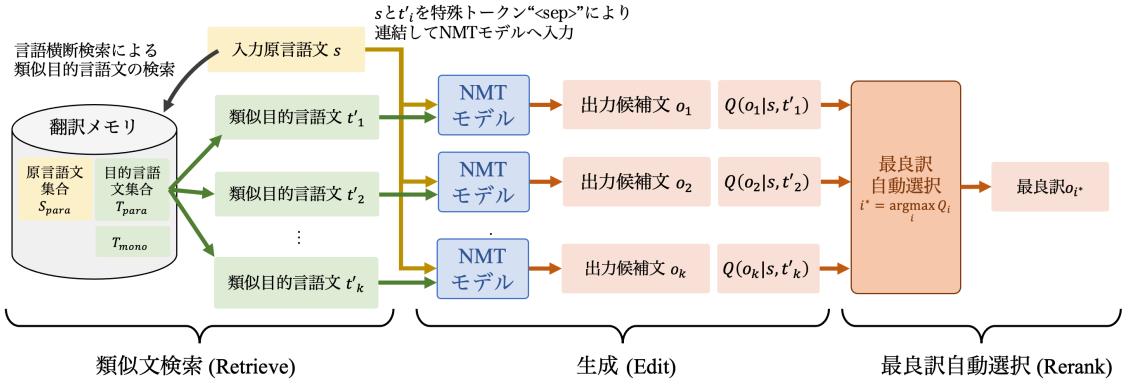


図 1: Retrieve-Edit-Rerank における推論の枠組み

似訳文」と定義する．このとき，原言語文と類似した目的言語文を直接検索することができれば，類似訳文の情報源として原言語-目的言語間の対訳文の揃っている翻訳メモリだけでなく，目的言語文のみで構成された単言語翻訳メモリが利用できる．以後，本論文では，翻訳メモリを原言語文 s と目的言語文 t のペアの集合とし，S_{para} を入力原言語文の集合，T_{para} を目的言語文の集合とする．また，目的言語の単言語翻訳メモリを T_{mono} と表記する．

2.2 編集距離に基づく類似度尺度

編集距離は，1 文字の挿入，削除，置換によって元の文字列を別の文字列に変換するために必要な最小の操作回数と定義される．本論文では，Bultéらの手法に倣って，Vanallemeersch らの [8] の類似度を採用した．

$$\mathrm{sim}(x,y) = 1 - \frac{\Delta_{ed}(x,y)}{\max(|x|_c, |y|_c)}$$

ここで，$\Delta_{ed}(x,y)$ は x と y の間の編集距離を，$|x|_c$ は x の文字数を表す．また，編集距離は同じ言語文のみでしか計算できないことから，類似文検索を行う場合の検索対象は S_{para} に限られる．そのため，検索された類似原言語文の訳文を類似訳文とする．

また，編集距離による類似文検索を行う場合には総当たりで類似度を算出し比較する必要があるため，大規模な翻訳メモリに対する計算コストが著しく大きい．そこで Bultéら [2] は，Python ライブラリ *SetSimilaritySearch*[1] が提供する類似度尺度 $containment_{max}$ を用いて検索された類似文候補集合に対して編集距離を算出する手法を採用した $(sss + ed)$.

1） https://github.com/ardate/SetSimilaritySearch

2.3 文埋め込みに基づく類似度尺度

本節では，文埋め込みに基づく文の類似度尺度について述べる．文埋め込みは文を高次元の実数ベクトルに写像したもので，文書分類や感情分析，対訳文検索に用いられる．以後，文 x を文埋め込みへ変換したものを $E(x)$ と表記する．本論文では，文埋め込み生成モデルとして，Multilingual Sentence-BERT[2)3)] [9] [10]，および，LaBSE [11] を利用した．Multilingual SBERT は，NLI データセットで訓練し STS タスクで高い精度を達成した英語版 SBERT を知識蒸留により多言語化したもので，言語に依らない文埋め込みを生成できる．LaBSE は，対訳文を用いて対照学習された多言語文埋め込み生成モデルであり，対訳文検索を行う BUCC タスクで高い精度を達成した．ここで，2 文 x と y の間の類似度 sim を以下のように定義する．

$$\mathrm{sim}(x,y) = \frac{E(x) \cdot E(y)}{|E(x)||E(y)|}$$

本論文では，入力原言語文 s を文埋め込み生成モデルにより変換して得られた文埋め込み $E(s)$ をクエリとして，$T_{para} \cup T_{mono}$ を対象にベクトルの近傍探索を行い，k 文の類似訳文 $t'_1, t'_2, \ldots, t'_k$ を抽出した．なお，ベクトルの近傍探索には FAISS [12] を利用した．

3 翻訳モデルによる生成 (Edit)

2） https://github.com/UKPLab/sentence-transformers
3） 本論文の実装では mSBERT の事前学習済みモデルとして，paraphrase-multilingual-mpnet-base-v2 を利用した．

3.1 翻訳モデルの訓練

翻訳モデルの訓練では Blut é ら [2] [3] と同様の手順で実施した．具体的には，まず入力原言語文 s をクエリとして，編集距離または文埋め込みによって翻訳メモリ内から k_t 文の類似訳文 $t'_1, \ldots, t'_{k_t}$ を検索する．その後，NFR モデルと同様に特殊トークン “<sep>” を挟んで s と t'_i を連結し，翻訳モデルへの入力とする．

3.2 翻訳モデルの推論

モデルの推論手順を図 1 に示す．まず，入力原言語文 s をクエリとし，訓練時と同様の手順で k_p 文の類似訳文 $t'_1, t'_2, \ldots, t'_{k_p}$ を検索する．次に，翻訳モデルを用いて k_p 回のデコーディングを行い k_p 文の出力文 o_i とデコーダの出力確率 p_{MT} に基づくリランキングスコア Q_i を得る．このリランキングスコアについては，4 節にて詳説する．

4 出力文のリランキング (Rerank)

最後のフェーズでは，生成フェーズで得られたリランキングスコア Q_i を最大化するような $i = i^*$ を選択し，最終的な出力文 o_{i^*} とする．ここで，本論文では 2 通りのリランキングスコアを検討した．1 つ目は Hossain ら [1] と同様の手法で，対数尤度に基づくスコアである．

$$Q_i^{(\text{Hossain})} = \log_2 p_{MT}(o_i|s, t'_i)$$

2 つ目は，文長による正規化を行った平均対数尤度に基づくランキング手法である．ここで，出力文 o_i の文長として，サブワード化された出力を一度復元してから単語数をカウントすることとし，以下では $|\text{deSW}(o_i)|$ と表記する．

$$Q_i^{(\text{proposed})} = \frac{\log_2 p_{MT}(o_i|s, t'_i)}{|\text{deSW}(o_i)|}$$

5 実験

5.1 データセット

本論文では，提案法の評価のためにアジア学術論文英日コーパス ASPEC [6]，欧州の諸機関からの出版物を元に作成された EU Bookshop Corpus [7] (以後 EUBC と表記) のうち英仏コーパスを利用し，翻訳方向はそれぞれ英日・英仏とした．翻訳モデルの訓練文には，各コーパスからランダムサンプリングされた 10 万文対のみ利用し，残りについては単言語翻訳メモリとした．表 1 に，データセットの詳細な文数を示す．また，和文に対しては MeCab[4] を，英文・仏文に対しては Moses tokenizer[5] を用いてトークナイズしたのち，Byte Pair Encoding (BPE)[6] [13] を用いて，操作数 32,000 でサブワードに分解した．

表 1: データセット

	ASPEC 英→日	EUBC 英→仏
訓練文	100,000	100,000
開発文	1,790	2,000
テスト文	1,812	2,000
翻訳メモリ	2,000,000 (日)	8,421,120 (仏)

5.2 実験設定

類似訳文の検索では，編集距離 (sss+ed)，mSBERT，LaBSE の 3 通りを比較した．このうち，sss+ed は同言語間でしか類似訳文を検索できないため，検索対象は訓練文の原言語文 10 万文に限られる．一方で，mSBERT，LaBSE は目的言語文を直接言語横断検索できるため検索対象は ASPEC で 200 万文，EUBC で 842 万文となる．訓練時には，類似文検索を行わない通常の手法（検索なし）と，最大 4 文の類似訳文を利用する手法 (top1〜top4) を比較した．推論時には，類似文を利用しない手法（類似文なし），オリジナルの NFR と同様の上位 1 文の類似訳文のみを利用する手法（$k_p = 1$），$Q^{(\text{Hossain})}$ に基づいてリランキングする手法，提案法である $Q^{(\text{proposed})}$ に基づく手法を比較した．このうち sss+ed を一つ目のベースラインとし，各検索手法において $Q^{(\text{Hossain})}$ に基づいてリランキングする手法を二つ目のベースラインとする．また，リランキングによる翻訳精度向上の上限を評価するため，各出力文候補の中で Sentence-BLEU が最も高い文を選択するオラクルについても評価した． 実験にあたっては PyTorch 実装の Transformer モデルを採用した．エンコーダとデコーダは各 6 層とし，隠れ次元は 512 次元，FF 層の次元は 2048 次元，マルチヘッド数は 8 とした．また，バッチサイズは 96 文，ウォームアップ 6,000 ステップのもとで 30 エポックの訓練を行い，開発文の BLEU が最も高いエポックにおけるテスト文の BLEU を採用した．

4) https://github.com/neologd/mecab-ipadic-neologd
5) https://www.statmt.org/moses/
6) https://github.com/rsennrich/subword-nmt

表 2: 各リランキング手法における翻訳モデルの実験結果 (訓練時に利用する類似訳文数は $k_t = 2$.「検索なし」 は類似訳文を利用しない vanilla Transformer の結果を,「$k_p = 1$」 は NFR と同様に最も類似度の高い類似訳文を利用した場合の結果を示す.「$Q^{(\text{Hossain})}$」は,対数尤度に基づくスコアを,「$Q^{(\text{proposed})}$」は,文の長さによる正規化を伴うスコアによる結果を示す.† は「$k_p = 1$(ベースライン 1)」に対する有意差 (p<0.05) を,‡ は「$Q^{(\text{Hossain})}$(ベースライン 2)」に対する有意差 (p<0.05) を示す.)

データセット	モデル	リランキングなし		リランキングあり ($k_p = 32$)		
		類似訳文なし	$k_p = 1$	$Q^{(\text{Hossain})}$ (ベースライン 2)	$Q^{(\text{proposed})}$	オラクル
ASPEC 英→日	検索なし	26.2	-	-	-	-
	sss+ed	-	26.2	26.6	26.8	28.5†‡
	mSBERT	-	26.5	26.4	26.9	29.7†‡
	LaBSE	-	27.1	27.4	**28.1**†	31.8†‡
EUBC 英→仏	検索なし	20.2	-	-	-	-
	sss+ed	-	20.2	20.3	20.3	20.3
	mSBERT	-	20.8	19.9	**22.1**†‡	25.2†‡
	LaBSE	-	21.0	19.6	21.7‡	25.6†‡

表 3: 各検索手法における実験結果 (「類似訳文なし」類似訳文を利用しない vanilla Transformer を,sss+ed は NFR と同様に編集距離に基づく手法を示す.† は sss+ed(ベースライン 1) に対する有意差 (p<0.05) を示す.)

	訓練時の類似文数 k_t	ASPEC 英→日	EUBC 英→仏
類似訳文なし	-	26.2	20.2
sss+ed (ベースライン 1)	1	26.4	20.2
	2	26.2	19.5
	3	26.1	18.3
	4	25.7	16.4
mSBERT	1	25.8	20.5
	2	26.5	20.8
	3	26.4	19.9
	4	26.2	19.0
LaBSE	1	25.8	20.9
	2	**27.1**†	**21.0**†
	3	26.5	20.4
	4	26.3	19.3

6 実験結果

各検索手法を用いて類似訳文を検索し,翻訳モデルを訓練した場合の結果を表 3 に示す.ただし,訓練時に利用する類似訳文数を $k_t = 1, 2, 3, 4$ とし,推論時に利用する類似訳文数を $k_p = 1$ とした.類似文検索を行わない場合は,ASPEC・EUBC それぞれ 26.2 ポイント・20.2 ポイントであったのに対し,sss+ed ではそれぞれ最大 26.4・20.2 ポイントと有意な BLEU の向上は見られなかった.一方で,mSBERT ではそれぞれ最大 26.5 ポイント・20.8 ポイントと有意差はないものの BLEU が向上したものが多く,LaBSE では最大 27.1 ポイント・21.0 ポイントと sss+ed に対して有意に高い BLEU が得られた.特に,mSBERT,LaBSE ともに上位 2 文を利用する場合に最も高い BLEU が得られ,上位 3 文以上を利用すると逆に精度が低下することが確認できる.

次に,retrieve-edit-rerank の枠組みに倣ってリランキングを行った場合の結果を表 2 に示す.まず,$Q^{(\text{Hossain})}$ によるリランキング手法に着目すると,いずれも有意な BLEU の向上は得られなかった.特に,EUBC において mSBERT や LaBSE を利用する場合有意に BLEU が低下する.一方で,$Q^{(\text{proposed})}$ によるリランキング手法では,sss+ed では有意な BLEU 向上が得られないものの,mSBERT や LaBSE を利用する場合は多くの事例で有意に BLEU が向上した.また,Sentence-BLEU が最大の文を取り出すオラクルはリランキングの上限値を示しているが,sss+ed は mSBERT や LaBSE に比べ低く,さらなる改善の余地が小さいことを示唆する.

7 おわりに

本論文では,retrieve-edit-rerank の枠組みのもとで,文埋め込みによる類似訳文の言語横断検索の活用と,文長に対する正規化を行ったより良いランキング手法を提案した.先行研究で利用された編集距離による類似訳文の検索手法よりも,mSBERT や LaBSE といった言語に依存しない文埋め込み生成モデルを用いたベクトルの近傍探索による検索が翻訳精度の向上に貢献することを示した.また,複数の相異なる類似訳文を利用して複数の訳文候補を生成しリランキングによって最良訳自動選択を行う際,サブワード化を復元した文の長さに対する正規化を行うことにより大幅な翻訳精度の向上を達成した.

参考文献

[1] N. Hossain, M. Ghazvininejad, and L. Zettlemoyer. Simple and effective retrieve-edit-rerank text generation. In **Proc. 58th ACL**, pp. 2532–2538, 2020.

[2] B. Bulté and A. Tezcan. Neural fuzzy repair: Integrating fuzzy matches into neural machine translation. In **Proc. 57th ACL**, pp. 1800–1809, 2019.

[3] A. Tezcan, B. Bulté, and B. Vanroy. Towards a better integration of fuzzy matches in neural machine translation through data augmentation. **Informatics**, Vol. 8, No. 1, 2021.

[4] M. Pagliardini, P. Gupta, and M. Jaggi. Unsupervised learning of sentence embeddings using compositional n-gram features. In **Proc NAACL HLT 2018**, pp. 528–540, 2018.

[5] D. Cai, Y. Wang, H. Li, W. Lam, and L. Liu. Neural machine translation with monolingual translation memory. In **Proc. 59th ACL and 11th IJCNLP**, pp. 7307–7318, 2021.

[6] T. Nakazawa, M. Yaguchi, K. Uchimoto, M. Utiyama, E. Sumita, S. Kurohashi, and H. Isahara. ASPEC: Asian scientific paper excerpt corpus. In **Proc. 10th LREC'16**, pp. 2204–2208, 2016.

[7] R. Skadiņš, J. Tiedemann, R. Rozis, and D. Deksne. Billions of parallel words for free: Building and using the EU bookshop corpus. In **Proc. 9th LREC**, pp. 1850–1855, 2014.

[8] T. Vanallemeersch and V. Vandeghinste. Assessing linguistically aware fuzzy matching in translation memories. In **Proc. 18th EAMT**, pp. 153–160, 2015.

[9] N. Reimers and I. Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In **Proc EMNLP-IJCNLP 2019**, pp. 3982–3992, 2019.

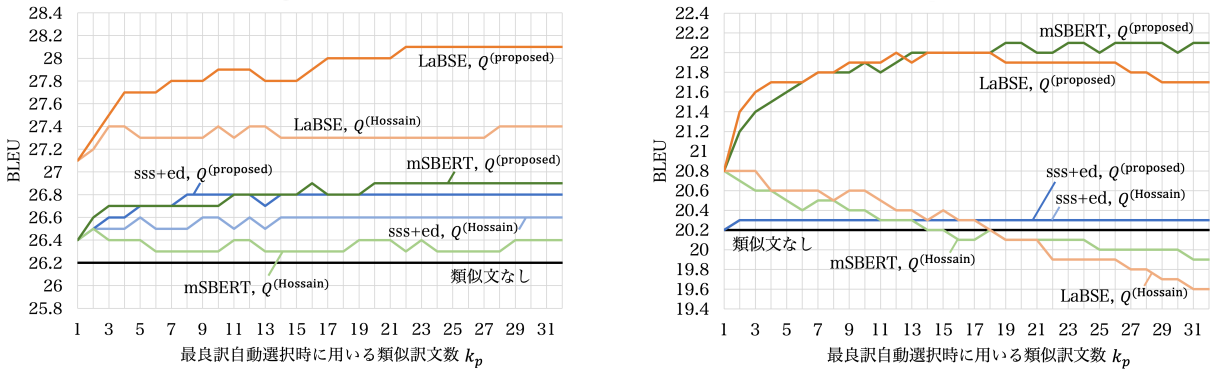
[10] N. Reimers and I. Gurevych. Making monolingual sentence embeddings multilingual using knowledge distillation. In **Proc. 16th EMNLP**, pp. 4512–4525, 2020.

[11] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, and W. Wang. Language-agnostic BERT sentence embedding. In **Proc. 60th ACL**, pp. 878–891, 2022.

[12] J. Johnson, M. Douze, and H. Jégou. Billion-scale similarity search with gpus. **CoRR**, Vol. abs/1702.08734, , 2017.

[13] R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. In **Proc. 54th ACL**, pp. 1715–1725, 2016.

A　リランキング時に用いる類似訳文数 k_p の影響



(a) ASPEC (En → Ja)　　**(b)** EBCU (En → Fr)

図 2: リランキング時に用いる類似訳文数 k_p を変化させた際のリランキング後の BLEU

B　具体例

表 4 に，ASPEC を用いた評価実験の結果の具体例を示す．この事例は，マウスの小腸におけるグルコースの吸収について述べた文である．表内の「類似訳文」には各検索手法とリランキング手法によって得られた最も良い類似訳文を示し，「出力文」には翻訳モデルによる出力結果を示す．その文に対する Sentence-BLEU をそれぞれ算出した．sss+ed によって検索された類似訳文に着目すると，まず top1 の類似訳文は生物分野の文ではなく，$Q^{(Hossain)}$ や $Q^{(proposed)}$ による類似訳文は生物分野の文ではあるものの内容として参考にできる情報は少ない．一方で，LaBSE によって検索された類似訳文に着目すると，top1 の類似訳文でも「グルコース」について述べた文が得られており，$Q^{(proposed)}$ では「マウス」における糖類の吸収について述べた文が得られた．また，出力文の Sentence-BLEU についても LaBSE+$Q^{(proposed)}$ において最も高い値を得る．

表 4: ASPEC を用いた評価実験の結果の具体例

原言語文	Study of the effect on the glucose absorption power , the TCDD exposed C57BL / 6J mouse increased the glucose absorption power in the intestinum tenue .
参照訳文	グルコース吸収能に対する影響を検討した結果，ＴＣＤＤ暴露Ｃ５７ＢＬ／６Ｊマウスは小腸におけるグルコース吸収能が増加した．

	類似訳文	**出力文**	**Sentence BLEU**
類似訳文なし	-	グルコース吸収能に対する効果の研究では，ＴＣＤＤがＰＢＬ／６Ｊマウスに暴露されたものは小腸のグルコース吸収能を増加させた．	21.0
sss+ed +top1	また，ポンプ光と信号光との間の偏光角を変化させて，スイッチングに与える影響を実験により確認した．	耐糖能吸収能への影響の研究では，ＴＣＤＣＤが５７ＢＬ／６Ｊを搭載したが，小腸では耐糖吸収能は増加していた．	18.8
sss+ed +$Q^{(Hossain)}$	また，ハタケシメジ投与は糞中胆汁酸排せつ量と，コレステロールを胆汁酸に変換するコレステロール７α-ヒドロキシラーゼの活性を上昇させた．	耐糖能吸収能への影響の研究では，ＴＣＤＣＤが５７ＢＬ／６Ｊを搭載したＴＣＤＤは小腸中での耐糖吸収能を増加した．	24.5
sss+ed +$Q^{(proposed)}$	また，ハタケシメジ投与は糞中胆汁酸排せつ量と，コレステロールを胆汁酸に変換するコレステロール７α-ヒドロキシラーゼの活性を上昇させた．	耐糖能吸収能への影響の研究では，ＴＣＤＣＤが５７ＢＬ／６Ｊを搭載したＴＣＤＤは小腸中での耐糖吸収能を増加した．	24.5
LaBSE +top1	グルコースとしょ糖間の相互転化を触媒する細菌性（大腸菌）キシロースイソメラーゼの発現によって，発育中のジャガイモ塊茎の代謝におけるヘキソースの役割について調べた．	グルコース吸収電力に対する影響の研究，ＴＣＤＤがＣＭＳ／６ＪマウスにＣＣＤは小腸のグルコース吸収電力を増加させた．	14.0
LaBSE +$Q^{(Hossain)}$	ヤギにおけるコレシストキニン（ＣＣＫ）の食欲減退効果を評価する目的で，２２時間粗食させた５頭の去勢ヤギに，塊状注入もしくは連続注入で第三脳室内にＣＣＫ（硫酸エステルＣＣＫ-８）を投与した．	グルコース吸収電力に対する効果の研究，ＴＣＤＤがＣｕｐ５７ＢＬ／６Ｊマウスに暴露された．	24.0
LaBSE +$Q^{(proposed)}$	正常マウスやＫＫ-Ａｙマウスを用いて，小腸におけるα-グルコシル二糖類の消化／吸収能力を比較した．	グルコース吸収電力，ＴＣＤＤ露光ＣＭＳ／６Ｊマウスのグルコース吸収電力に対する影響を研究した結果，小腸におけるグルコース吸収電力を増加させた．	27.3