修 士 学 位 論 文

題　　目

ニューラルネットワーク言語モデルを用いた含意文生成システム

Entailment Generation with Neural Language Model

主 査　　柳本 豪一 准教授

副 査　　中島 智晴 教授

副 査　　泉　正夫　教授

令 和 2 年（ 2020 年）度 卒 業

(No.　2191104019　)　　趙　暁梅

大阪府立大学大学院 人間社会システム科学研究科
現代システム科学専攻 知識情報システム学分野

ニューラルネットワーク言語モデルを用いた
含意文生成システム
Entailment Generation with Neural Language Model

分　野　名	知識情報システム学分野	氏　名	趙　暁梅
Department	Knowledge and Information Systems	Name	ZHAO　XIAOMEI

Textual entailment needs semantic judgment between two sentences and is a good task to measure text understanding. If we realize entailment generation, we can apply it to summarization that keeps semantics between an original text and a generated text.

However, various entailments can be generated from a premise sentence. Even if they are almost the same as the premise sentences, they are appropriate from the viewpoint of an entailment task. For example, the entailment *"He went to borrow books."* is created by deleting some words from the premise sentence *"He went to the library again to borrow books."*, but another entailment *"This is not his first visit to the library."* is created by different words with a low similarity.

The purpose of this paper is to generate entailment which have a low similarity with the premise sentence. For entailment generation, I used the Sequence-to-Sequence model with Attention mechanism, which is often used in tasks of natural language process. Moreover, in order to limit the similarity between premise sentences and entailments, I applied a similarity loss function to entailment generation task. The loss is based on the similarity between the premise sentences and predicted entailments.

In experiments, we use the corpus of Stanford Natural Language Inference (SNLI). SNLI corpus is a collection of 570k human-written English sentence pairs manually labeled for balanced classification with the label "entailment", "contradiction", and "neutral". In this experiment, to discuss entailment generation, I extracted pairs of sentences representing entailment meanings from SNLI as the dataset. The proposed method and the baseline method are trained 10 epochs with train data, and entailments are generated using test data.

In evaluation experiments, I analyzed SNLI dataset. I found that premise sentences and entailment sentences are shared with many same words. Hence, training data affects the final prediction strongly and entailment tends to share many words with a premise sentence.

By discussing the predicted entailments, I found that there was no significant difference between the baseline and the proposed method from the viewpoint of average sentence similarity. However, the number of entailments with the highest similarity, over 0.9, increases and the proposed method can generate entailments with various words.

In the future, I will focus on reducing the average similarity. Especially, I will increase the epochs of train, and improve the similarity loss function.

ニューラルネットワーク言語モデルを用いた

含意文生成システム

柳本研究室　2191104019　趙 暁梅

1.はじめに

含意関係とは、前提となる文 T と文 H が存在した時、文 T が真の場合に文 H が真であると推論できる関係のことである。含意関係を認識することは、自然言語処理における情報検索、質問応答、情報抽出などの様々なタスクにおいて必要とされる技術である。Dagan ら[1]は、推論ルールを用いて含意関係の抽出を行なっている。

しかし、含意関係にある 2 文には様々なものがあり、ほぼ前提文と同じものであっても含意文と判断されてしまう。例えば、以下の例を用いると、含意文 1 は単語を前提文から単語を削除して作成されているが、含意文 2 は異なった語彙で作成されている。

前提文　He went to the library again to borrow books.
含意文 1　He went to borrow books
含意文 2　This is not his first visit to the library.

本論文では、できるだけ前提文と語彙が重ならないような含意文を作成することを目的とする。含意生成には、機械翻訳でよく用いられる Attention 付きの Sequence-to-Sequence モデルを用いることとする。さらに、前提文と含意文との語彙の重複を制限するため、前提文と生成された含意文の類似度に基づいた損失を定義し、それを用いることで目的を達成する。提案手法を評価するために Stanford Natural Language Inference(SNLI)コーパスを用て実験を行い、前提文と語彙を共有するような含意文を抑えることができた。

2.関連研究

Attention based Sequence-to-Sequence モデル[2][3]は幅広い分野の自然言語処理タスクに応用され、含意文生成タスクのようなテキスト生成タスクにおいて高い性能を示しているモデルである。

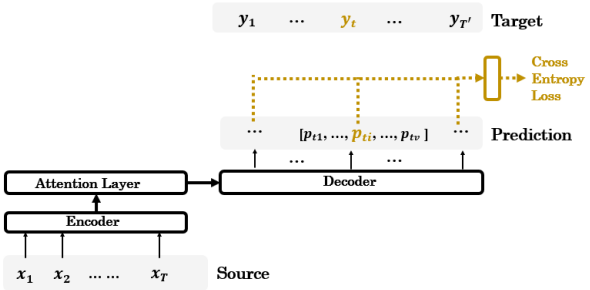

図 1:　ベースライン手法の構成

Attention based Sequence-to-Sequence モデルでは、前提文となる入力は Encoder で圧縮され、Decoder に渡される。Decoder では、Encoder からの情報と前の時刻の出力を計算して含意文を予測する。コーパスの含意文を教師信号として予測された含意文に交差エントロピー誤差を用いて学習される。この学習により、コーパスとして用意された含意文に生成される含意文の質は依存することとなる。

3 提案手法

3.1.類似損失(Similarity Loss)

前提文と生成された含意文との語彙の重なりを小さくすることで、前提文と異なる含意文の生成を目指す。このため、入力文と出力文の類似度を損失に組み込むこととする。図 2 に提案システムの構成を示す。

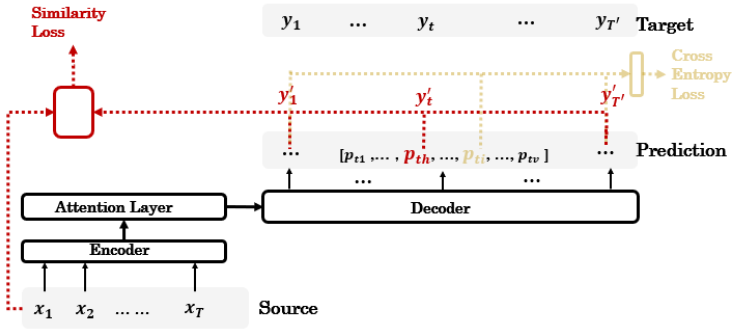

図 2:　提案手法の構成

予測した含意文$[y'_1, \ldots, y'_{T'}]$と入力である前提文$[x_1, \ldots, x_T]$の類似損失を計算する。そして、類似損失と交差エントロピー誤差損失の和を訓練データにおける誤差関数として用いる。

類似損失は$[y'_1, \ldots, y'_{T'}]$と$[x_1, \ldots, x_T]$をもとに計算する。$[y'_1, \ldots, y'_{T'}]$と$[x_1, \ldots, x_T]$で共有されているトークン数を計算して、$[y'_1, \ldots, y'_{T'}]$の長さT'で割ることで、類似損失を定義する。例えば、図 3 では、含意文の長さは 9 で、前提文と共有されているトークンは 3 つであるため、含意文の長さで割ると、類似損失は$\frac{1}{3}$となる。

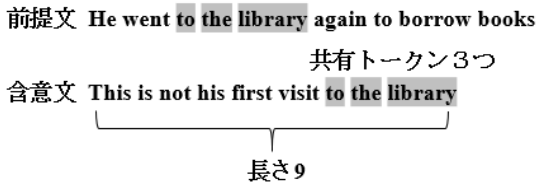

図 3:　類似損失の例

類似損失と交差エントロピー誤差損失の和を訓練データにおける最終的な誤差関数とする。

$$\text{Total Loss} = \text{Similarity Loss}([x_1, \ldots, x_T], [y'_1, \ldots, y'_{T'}]) + \text{Cross Entropy Loss} \quad (1)$$

こうすることで、入力文と近い含意文が生成された場合には誤差が大きくなり、出来るだけ入力文と異なった含意文が生成されるようになる。

3.2 評価手法

生成した含意文と入力となる前提文の文間類似度を求めるために、訓練済みの BERT[4]を用いて、前提文と生成された含意文をベクトルで表現し、その類似度をコサイン類似度により計算する。単語埋め込みサイズは 768 である。生成した含意文と入力となる前提文をそれ

ぞれ BERT に入力して、最後の 4 層の出力の平均ベクトルを文ベクトルにする。生成した含意文と入力となる前提文の文ベクトルをもとに、コサイン類似度を計算する。これを類似度についての評価指標にする。

4.実験

4.1 データセット

Stanford Natural Language Inference(SNLI)コーパスは Samuel R ら[5]によって作成されたコーパスである。自然言語推論タスクでよく使用されており、ラベル付きのコーパスとして代表的なものである。2 文の関係は、「含意(entailment)」「中立(neutral)」「矛盾(contradiction)」という 3 つのラベルが付与されている。実験では、含意文生成をタスクにするため、「含意」ラベルが付与された含意関係である前提と仮説のペアのみ取り出して、実験に用いた。取り出したデータ数は 190,113 であり、訓練データが 183,416 件、検証データが 3,328 件、テストデータが 3,360 件に分割されていた。

類似度問題の原因の一つは、データセットにあると考えられる。SNLI において、訓練データとなる前提文と含意文の文間類似度は平均で 0.7696 である。データセットにおける類似度は[0.2, 1]の範囲に分布され、類似度が[0.5, 1]範囲内のデータは全体の 99.446%を占めている。

4.2 実験設定

エンコーダとデコータの隠れ層はそれぞれ 2 層の Bi-GRU と 2 層の GRU からなるものであり、隠れ層のサイズと単語埋め込みのサイズは 1,024 に設定し、0.2 の Dropout を採用した。注意層では、「Global Attention」と「General Attention Score」を採用した。訓練の段階で、オプティマイザは「Adam」を使用した。バッチサイズは 32 で、GPU 上で学習を行なった。

表 1: 実験設定

アイテム	ベースライン	提案手法
Word Embedding Dim	1024	1024
Encoder	2 層　Bi-GRU	2 層　Bi-GRU
Decoder	2 層　GRU	2 層　GRU
GRU Hidden Size	1024	1024
Dropout	0.2	0.2
Learning Rate	0.0001	0.0001
Optimizer	Adam	Adam
Regularization	L2	L2
Batch Size	32	32
Loss Function	Cross Entropy Loss	Cross Entropy Loss + Similarity Loss

5.結果

提案手法とベースライン手法をそれぞれ訓練データで 10 回学習させて、テストデータによる含意文の生成を行い、前提文と含意文の類似度の平均を表 2 に示す。平均類似度の点からはベースラインと提案手法に大きな差は現れなかった。

表 2: 訓練結果

	ベースライン	提案手法
平均類似度	0.7517	0.7580

両手法で生成された含意文と前提文の類似度の分布を図 3 と図 4 に示す。類似度が 0.9 以上のものは減少していることがわかる。

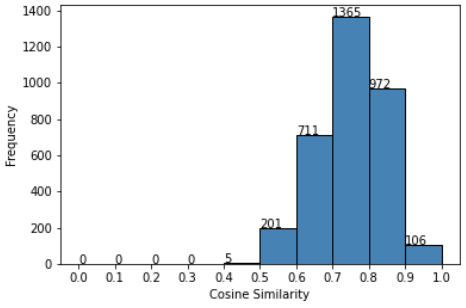

図 3: ベースライン手法による
テストデータの類似度分布

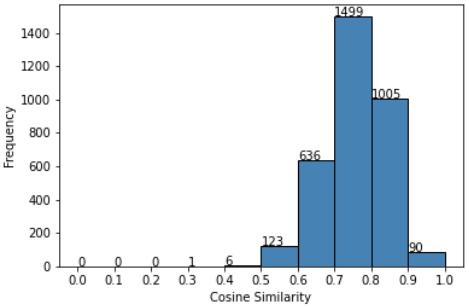

図 4: 提案手法による
テストデータの類似度分布

6.おわりに

本論文では、前提文と生成された含意文の語彙が重ならないように、類似損失を導入した手法を提案した。これにより、前提文の語彙をそのまま使ったような含意文の生成を抑えることができた。

今後の課題としては、全体的に前提文と含意文との語彙の重なりを抑えるようにすることを目指す。また、学習回数を増やすことでどのように変化するかについても検討を行う。

参考文献

[1] Dagan I, Glickman O.　Probabilistic textual entailment: Generic applied modeling of language variability. In *Proceedings of the PASCAL Workshop on Learning Methods for Text Understanding and Mining, Grenoble*, pp26-29, 2004

[2] I. Sutskever, O. Vinyals, QV Le. Sequence to Sequence Learning with Neural Networks. In *Proceeding of* NIPS, pp.3104- 3112,2014.

[3] M. Loung, H. Pham, C. D. Manning. Effective Approaches to Attention-based Neural Machine Translation. In *Proceeding of EMNL*, pp.1412-1421,2015.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceeding of Association for Computational Linguistics*, pp.4171-4186, 2019.

[5] Samuel R. Bowman, Gabor Angeli, Christopher Potts, Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, pp.632–642, Sep.2015.

目次

図目次

表目次

1 はじめに

本稿では、含意文生成(Entailment Generation)をタスクにして取り組んでいく。

含意関係とは、前提となる文 T と文 H が存在した時、文 T が真の場合に文 H が真であると推論できる関係のことである。例えば、以下に示した T と H は含意関係である。含意関係を認識することは、自然言語処理における情報検索、質問応答、情報抽出などの様々なタスクにおいて必要とされる技術である。Dagan ら[1]は、推論ルールを用いて含意関係の抽出を行なっている。

T: *He went to the library again to borrow books.* (前提文)

↓ 推論

H: *This is not his first visit to the library.* (含意文)

本稿では、便宜のために、T と H が含意関係である場合は、T を「前提文」と称し、H を「含意文」とする。つまり、前提文から含意文を生成することをタスクにして取り組んでいく。含意文生成タスクで必要とされる深い理解力と推論力は、自然言語処理における大きな挑戦の一つである。自然言語処理における根本的なトピックを直接扱うため、含意文生成技術は様々な自然言語処理タスクへの応用が期待されている。含意文生成を組み合わせることで、既存の自然言語処理タスクの性能を改善することがよく知られている。例えば、Matsumoto らの含意文生成による質問生成[2]、 Sairaj R らの含意文生成によるエンティティマッピング(Entity Mapping)[4]などがある。一方で、含意文生成が他のタスクとお互いに支え合うことも可能である。例えば、Pasunuru らの含意文生成と文章要約生成のマルチタスク[3]、Guo らの含意文生成と文章要約生成のマルチタスク[5]、 Pasunuru らの含意文生成とビデオキャプション生成のマルチタスク[6]などがある。

しかし、含意関係にある 2 文には様々なものがあり、ほぼ前提文と同じもの

であっても含意文と判断されてしまう。例えば、以下の例を用いると、含意文1は単語を前提文から単語を削除して作成されているが、含意文2は異なった語彙で作成されている。類似度問題の原因の一つは、データセットにあると考えられる。

前提文 *He went to the library again to borrow books.*

含意文1 *He went to borrow books.*

前提文 *He went to the library again to borrow books.*

含意文2 *This is not his first visit to the library.*

本論文では、できるだけ前提文と語彙が重ならないようにして、含意文の類似度を控えることを目的とする。含意生成には、機械翻訳でよく用いられるAttention付きのSequence-to-Sequenceモデルを用いることとする。さらに、前提文と含意文との語彙の重複を制限するため、前提文と生成された含意文の類似度に基づいた損失を定義し、それを用いることで目的を達成する。提案手法を評価するためにStanford Natural Language Inference(SNLI)コーパスを用いて実験を行い、前提文と語彙を共有するような含意文を抑えることができた。

論文の構成は以下となっている。第1章では含意文生成という自然言語処理タスクを紹介し、類似度についての問題を提起する。第2章では関連研究の紹介と議論を行っていく。ベースラインモデルを紹介した上で、訓練時の損失関数の議論を行う。第3章では含意文生成における文間類似度問題に取り組むために、新しい損失関数を提案する。第4章では実験設定を紹介する。第5章では実験結果に基づいて、提案手法を評価し、今後の課題についての説明を行う。第6章では本稿のまとめとなる。

2 関連研究

自然言語処理系のタスクは、人工知能が取り組むべき重要な課題であり、深層学習などを用いた手法が提案されて以来、ニューラルネットワークによる手法が主流になっている。以下では、ニューラルネットワークによる手法について紹介する。

含意文生成タスクにおいて、生成精度の向上に貢献している研究は数多くなされてきた[3, 5, 6, 10]。

2.1 Attention based Sequence-to-Sequence

Attention based Sequence-to-Sequence モデルは Loung ら[8]に提案されたのである。幅広い分野の自然言語処理タスクに応用され、含意文生成タスクのようなテキスト生成タスクにおいて高い性能を示している [18, 19, 20, 21, 22]。「双方向性ゲート付き回帰型ユニット」と「Sequence-to-Sequence」、「注意機構」三つの部分に分けて紹介していく。

2.1.1 双方向性ゲート付き回帰型ユニット

ゲート付き回帰型ユニット(Gate Recurrent Unit, GRU)[16]は Cho ら[16]に提案され、リカレントニューラルネットワーク(Recurrent Neural Network, RNN)[12, 13, 14] ゲート機構をくみいれたものである。GRU はシンプル RNN と比べ、長いシーケンスにも有効とされるように設計されている。シンプル RNN は誤差逆伝播中の勾配消失や直近依存関係のみを学習するなどの問題が指摘され、長いシーケンスの学習において不安定なところがある[23,24]。GRU は忘却や更新ゲートによって、時間的に離れているステップのトークンを考慮でき、シンプル RNN の問題を軽減できる。また、長短期記憶(Long Short-Term Memory ,LSTM)[12]と比べて計算コストを控えることができ、訓練効率を向上させる。

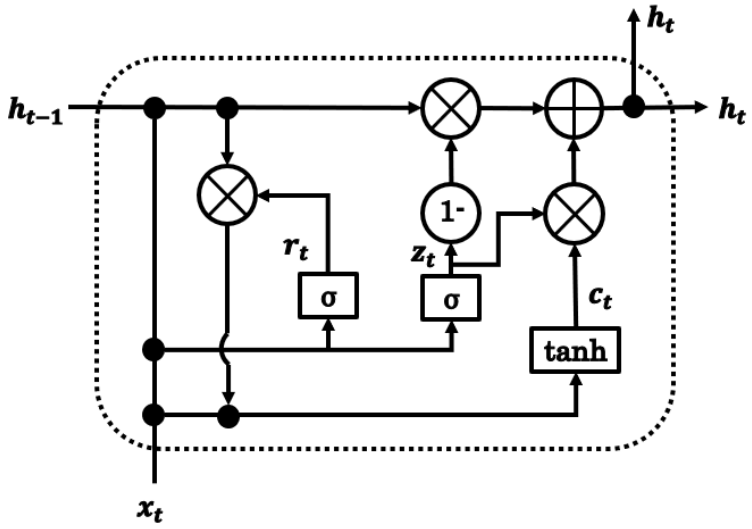

図 2.1　ゲート付き回帰型ユニット

Fig. 2.1　Gate Recurrent Unit

$$z_t = \sigma\left(W^{(z)}x_t + U^{(z)}h_{t-1}\right) \tag{2.1}$$

$$r_t = \sigma\left(W^{(r)}x_t + U^{(r)}h_{t-1}\right) \tag{2.2}$$

$$c_t = tanh\left(W^{(c)}x_t + U^{(c)}(h_{t-1} \odot r_t)\right) \tag{2.3}$$

$$h_t = (1 - z_t) \odot h_{t-1} + c_t \odot z_t \tag{2.4}$$

図 2.1 は時刻 t における GRU セルである。x_tは入力情報、h_tは GRU の出力情報、h_{t-1}は前の時刻の GRU の出力情報である。GRU セルにはリセットゲートr_t、アップデートゲートz_tがある。r_tとz_tは両方ともx_tとh_{t-1}をもとに計算される。r_tは重みとなってh_{t-1}に掛けられ、x_tと合わせてc_tを計算する。シグモイド関数によって 0~1 の範囲に限定されたz_tは、h_{t-1}とc_tをどれだけ無視するかを決定する。h_{t-1}とc_tは決定された比率によって合成され、h_tを計算する。

このようにリセットゲートやアップデートによって、前のステップ情報の維持と削除を適当に選択することが可能なり、離れているステップの情報の記憶維持が容易になり、シーケンスの長期依存が可能となる。GRU は様々なタスクに応用され、よい性能を示している[25,26]。

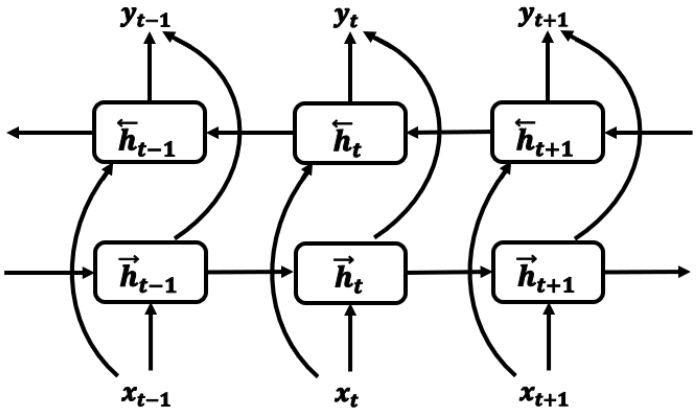

図 2.2　双方向性ゲート付き回帰型ユニット

Fig. 2.2　Bi-directional Gate Recurrent Unit

$$\overleftarrow{h}_t = f(x_t, \overleftarrow{h}_{t+1}) \tag{2.5}$$

$$\vec{h}_t = f(x_t, \overleftarrow{h}_{t-1}) \tag{2.6}$$

$$h_t = [\overleftarrow{h}_t; \vec{h}_t] \tag{2.7}$$

双方向性は Schuster ら[27]に提案されたものである。従来の RNN において前から後ろの順番しかで情報を学習しているという背景で、双方向 RNN が提案された。双方向 RNN において前から後ろの順番と後ろから前の順番、過去と未来の情報両方学習するようになる。双方向性は様々なモデルに応用されている[31, 32]。双方向性ゲート付き回帰型ユニット(Bi-directional Gate Recurrent Unit, Bi-GRU)はその応用の一つである。GRU は前のステップの情報の記憶を維持しやすいが、後ろの情報を扱えない。Bi-GRU は過去と未来両方の記憶を集約して扱えるようになる。図 2.2 のように隠れ層の t 時刻において、後ろから前の情報$\overleftarrow{h}_t$と前から後ろの情報$\vec{h}_t$は連結して隠れ層の出力h_tとなり、良い性能を

示している[28, 29, 30]。

2.1.2　Sequence-to-Sequence

Sequence-to-Sequence は Sutskever ら[7]に提案されたモデルである。モデルはエンコーダ(Encoder)とデコーダ(Decoder)から構成されている。エンコーダとデコーダはそれぞれ RNN からなるものである。

$$p(y_1, \dots, y_{T'} \mid x_1, \dots, x_T) = \prod_{t=1}^{T'} p(y_t \mid v, y_1, \dots, y_{t-1}) \tag{2.8}$$

式 2.8 では、x_1からx_Tは入力シーケンス、y_1から$y_{T'}$は出力シーケンス、入力シーケンスと前の時刻の出力を計算して次のトークンを出力する。例えば、図 2.1 のように、エンコーダは"<start> A B C <end>"を計算して、デコーダは"a b c d e f <end>"を出力する。

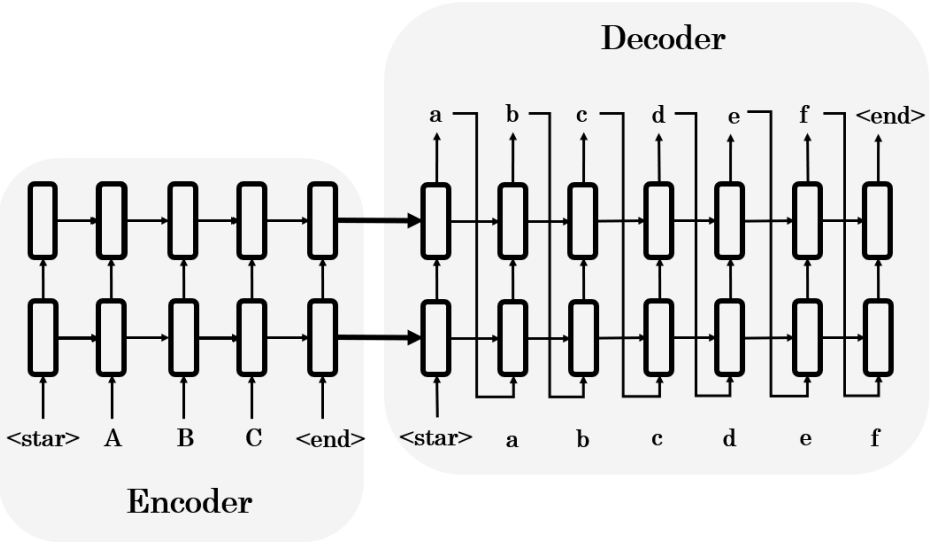

図 2.3　Sequence-to-Sequence の例

Fig. 2.3　Example of Sequence-to-Sequence.

Sequence-to-Sequence は可変長の出力を扱うように設計されている。例えば、図 2.1 のように、シーケンスの終わりを表現する"<end>"が予測されるまで、デコーダは繰り返して次のトークンを出力する。これによって、可変長の出力シーケンスを生成することが可能である。

2.1.3　注意機構

前節で紹介したモデルでは、エンコーダには入力の全ての情報を固定長のベクトルに圧縮しており、入力シーケンスが長くなると、各トークンの情報を適切に圧縮しきれない恐れがあり、デコーダでの対応が難しくなる。この問題を解決するため、注意機構(Attention Mechanism)[8,11]が導入されるようになった。

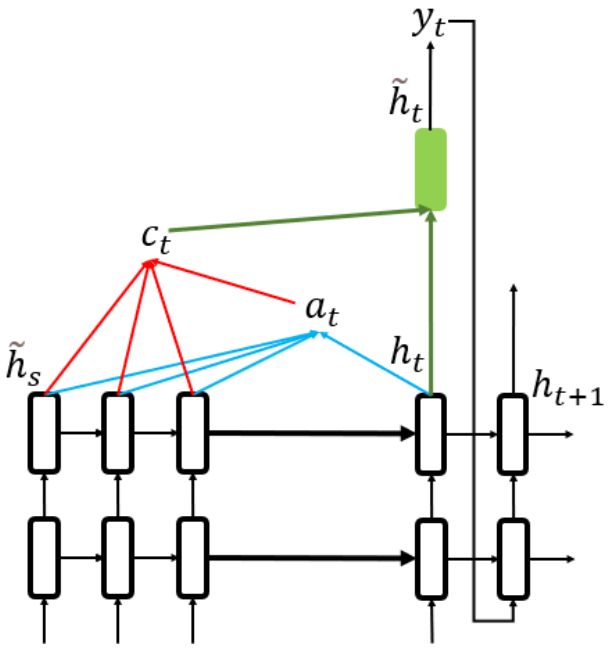

図 2.4　注意機構

Fig. 2.4　Attention Mechanism.

$$score(h_t, \tilde{h}_s) = \begin{cases} h_t^{\mathrm{T}} \tilde{h}_s & dot \\ h_t^{\mathrm{T}} W_a \tilde{h}_s & general \\ v_a^{\mathrm{T}} tanh(W_a[h_t; \tilde{h}_s]) & concat \end{cases} \tag{2.9}$$

$$a_t(s) = align(h_t, \tilde{h}_s) = \frac{\exp(score(h_t, \tilde{h}_s))}{\sum_{s'} \exp(score(h_t, \tilde{h}_{s'}))} \tag{2.10}$$

$$c_t = \sum_s (a_t(s), \tilde{h}_s) \tag{2.11}$$

$$\tilde{h}_t = \tanh(W_c[c_t; h_t]) \tag{2.12}$$

$$p(y_t | y_{<t}, x) = softmax(W_s \tilde{h}_t) \tag{2.13}$$

式(2.2~2.6)のように、エンコーダの各時刻 s の隠れ層状態$\tilde{h}_s$とデコーダのある時刻 t の隠れ層状態h_tを用いて通して、h_tに対する$\tilde{h}_s$のスコア$score(h_t, \tilde{h}_s)$を計算する。各時刻 s のスコアをソフトマックス関数ですべての時刻s'の exp の和で割ることで、重みa_tに変換する。a_tとエンコーダの各時刻 s の隠れ層状態$\tilde{h}_s$の重み付き和より、入力シーケンスの文脈情報を表現したコンテキストベクトルc_tを計算する。最後に、c_tはデコーダに渡され、時刻 t の隠れ層状態h_tと合わせて$\tilde{h}_t$を計算して、出力y_tを予測する。

一般的な Sequence-to-Sequence モデルに比べて、テキストが長くなって

も、デコーダが効率よく入力情報を利用することが可能となる。一方で、注意機構によって、エンコーダがデコーダに渡す加重平均した文脈化ベクトルは時刻に応じて動的に変わり、出力は文脈化情報に基づいて生成することができる。

2.2 交差エントロピー誤差損失

損失関数を用いて、前節で紹介したモデルは誤差逆伝播により学習が可能になる。損失関数はターゲットデータと予測した出力の誤差を表し、モデルの学習の指標となるものである。

本稿では、交差エントロピー誤差損失関数(Cross Entropy Loss)をベースライン損失関数にして検討を行う。

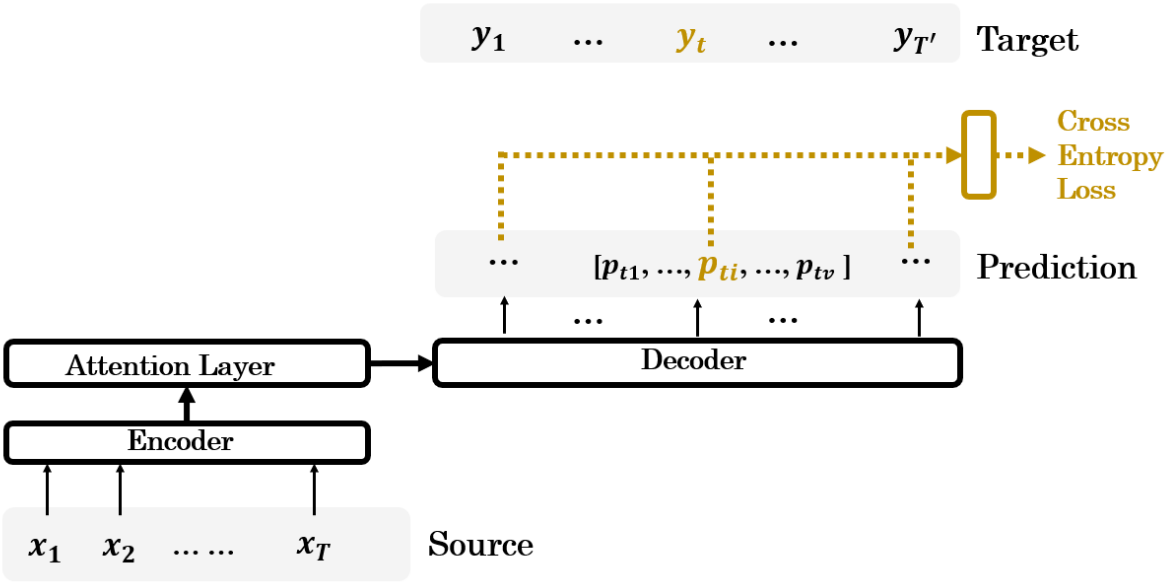

図 2.5　交差エントロピー誤差損失

Fig. 2.5　Cross Entropy Loss.

図 2.5 は Attention based Sequence-to-Sequence モデルにおける交差エントロピー誤差損失である。エンコーダに入力された長さ T のソースシーケンス$[x_1, x_2, …, x_T]$を考える。学習の目標は長さT'のターゲットシーケンス$[y_1, y_2, …, y_{T'}]$である。

$$\text{Cross Entropy Loss} = -\frac{1}{T'}\sum_{t=1}^{T'} \log(p_{ti}) \tag{2.14}$$

予測の段階では語彙サイズ v のソフトマックスしたベクトル$[p_{t1}, …, p_{ti}, …,$

p_{tv}]が出力され、各語彙の確率を表している。p_{ti}は時刻 t のターゲットy_tに対応する確率を表す。シーケンスにある各時刻の損失の平均を求めると、最終の損失が得られる。

3 提案手法

3.1 類似度問題と交差エントロピー誤差損失

モデル予測の段階で、コーパスの含意文を教師信号として予測された含意文に交差エントロピー誤差を用いて学習される。この学習により、コーパスとして用意された含意文に生成される含意文の質は依存することとなる。モデルの予測したシーケンスとターゲットシーケンスの交差エントロピー誤差を計算して誤差逆伝播して学習させることによって、ターゲットシーケンスに近づいている最適化されたモデルが得られる。しかし、前の 1.2 節で述べたように、コーパスにおいて類似度の高いデータセットが数多く存在しているの問題がある。

前提文と生成された含意文との語彙の重なりを小さくすることで、前提文と異なる含意文の生成を目指す。このため、類似損失を提案する。

3.2 類似損失

含意文生成タスクにおいて前提文と含意文で共通した語彙を減らす目的を達成するためには、交差エントロピー誤差だけでは不十分である。なぜなら、交差エントロピー誤差はターゲットとの一致度合いのみを評価しており、前提文との関係を考慮していないからである。

本稿では、類似損失(Similarity Loss)を提案する。交差エントロピー誤差損失と比べて、入力文と出力文の間で高い類似度を持つ場合に損失関数に反映させるようにして、類似度を控えるようにする。

図 3.1 は類似損失の処理を示す。エンコーダに入力となる前提文は$[x_1, x_2, \ldots, x_T]$である。モデルが予測した含意文は$[y'_1, \ldots, y'_{T'}]$である。まず、予測したテキスト$[y'_1, \ldots, y'_{T'}]$と入力となる前提文$[x_1, \ldots, x_T]$から類似度損失(Similarity Loss)を計算する。そして、類似損失と交差エントロピー誤差損失の和を訓練データにおける誤差関数として用いることにする。こうすることで、入力文と近い含意文が生成された場合には誤差が大きくなり、出来るだけ入力文と異なった含意文が生成されるようになる。

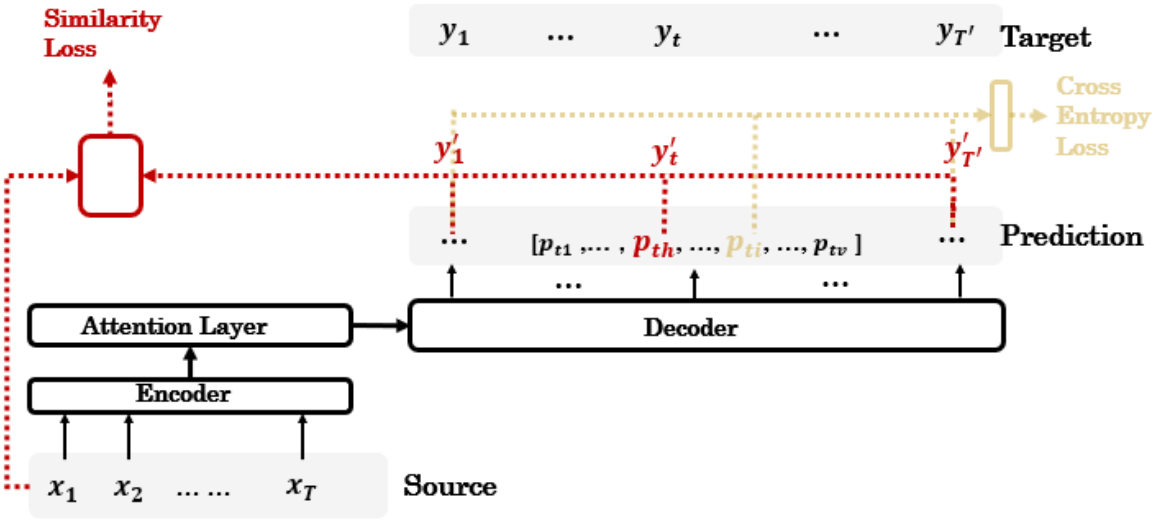

図 3.1　類似損失

Fig. 3.1　Similarity Loss.

$$\text{Total Loss} = \text{Similarity Loss}([x_1, \ldots, x_T], [y'_1, \ldots, y'_{T'}]) + \text{Cross Entropy Loss} \tag{3.1}$$

類似損失は$[y'_1, \ldots, y'_{T'}]$と$[x_1, \ldots, x_T]$をもとに計算するものである。$[y'_1, \ldots, y'_{T'}]$と$[x_1, \ldots, x_T]$が重複しているトークンの数を計算して、$[y'_1, \ldots, y'_{T'}]$の長さT'で割り、類似損失とする。例えば、以下の例文において、前提文"He went to the library again to borrow books"に対して、含意文を二つを考える。それぞれ"He went to borrow books"と"He is not at home now"である。含意文 1 の長さは 5 で、前提文と重ねたトークンは 5 つで、重ねたトークン数を含意文 1 の長さで割ると、類似度損失は 1 となる。これと同様に含意文 2 の長さは 6 で、前提文と重ねたトークンは 1 つで、重ねたトークン数を含意文 2 の長さで割ると、類似度損失は 1/6 となる。

前提文	*He went to the library again to borrow books*	
含意文 1	*He went to borrow books*	重ねたトークン 5 つ
	1 2 3 4 5	

前提文	*He went to the library again to borrow books*	
含意文 2	*This is not at home now*	重ねたトークン 1 つ
	1 2 3 4 5 6	

類似損失と交差エントロピー誤差損失の和を訓練データにおける誤差関数として用いる。こうすることで、訓練データに類似度が高い前提文と含意文の場合、入力文と近い含意文が生成された場合には誤差が大きくなり、罰として高い類似度を損失関数に反映させ、できるだけ入力文と異なった含意文が生成されるようになる。

3.3 評価手法

ベースライン手法と提案手法を比較ための評価手法を紹介する。

生成した含意文と入力となる前提文の文間類似度は訓練済みの BERT[1][4] とコサイン類似度を用いて計算する。類似度はコサイン類似度で、テキスト特徴量は訓練済みの BERT から計算されたものである。訓練済みの BERT は 12 層 12 ヘッドであり、単語埋め込みサイズは 768 である。生成した含意文と入力となる前提文をそれぞれ BERT に入力して、最後の 4 層の出力の平均ベクトルを文ベクトルにする。生成した含意文と入力となる前提文の文ベクトルをもとに、コサイン類似度を計算する。

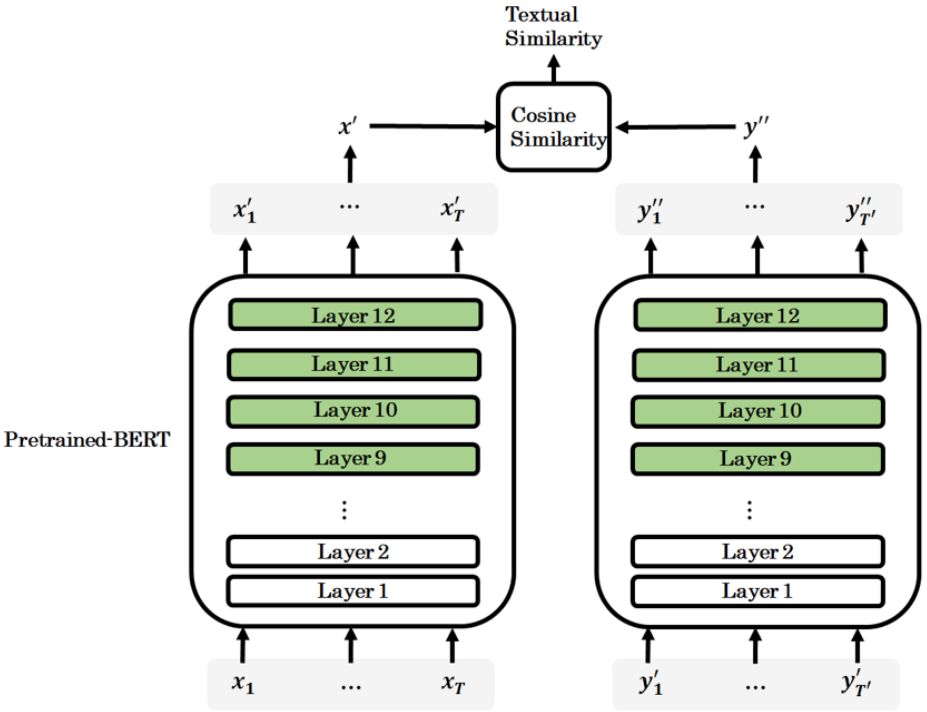

図 3.2　評価手法

Fig. 3.2　Calculation Method of Textual Similarity.

[1] 12/768 (BERT-Base)のダウンロードサイト:https://github.com/google-research/bert

$$x_t' = x_t^9 + x_t^{10} + x_t^{11} + x_t^{12} \quad (3.2)$$

$$y_t'' = y_t'^9 + y_t'^{10} + y_t'^{11} + y_t'^{12} \quad (3.3)$$

$$x' = \frac{1}{T}\sum_{t=1}^{T} x_t' \quad (3.4)$$

$$y'' = \frac{1}{T'}\sum_{t=1}^{T'} y_t'' \quad (3.5)$$

$$\text{Textual Similarity} = \text{Cosine Similarity}(x', y'') \quad (3.6)$$

図 3.2 のように、入力となる前提文は $[x_1, \dots, x_T]$であり、モデルが生成した含意文は$[y_1', \dots, y_{T'}']$である。$[x_1, \dots, x_T]$と$[y_1', \dots, y_{T'}']$をそれぞれ訓練済みの 12 層の BERT に入力して、Layer9 から Layer12 までという最後の 4 層の出力ベクトルx_t^9、x_t^{10}、x_t^{11}、x_t^{12}を計算する。x_t^9、x_t^{10}、x_t^{11}、x_t^{12}の平均ベクトルx''を前提文の文ベクトルにする。含意文も同様に、$y_t'^9$、$y_t'^{10}$、$y_t'^{11}$、$y_t'^{12}$の平均ベクトルy''を含意文の文ベクトルにする。最後に、x''とy''のコサイン類似度を計算する。実験結果の評価として、テストデータの文間類似度を評価指標にする。

4 実験

4.1 データセット

Stanford Natural Language Inference(SNLI)コーパス[2]は Samuel R ら[15]に提出されたコーパスである。自然言語推論タスクでよく使用されており、ラベル付きのテキストペアからなるものである。「含意(entailment)」「中立(neutral)」「矛盾(contradiction)」という三つのラベルはテキストペア間の関係を表す。例えば、表 4.1 において、前提" An older and younger man smiling."と仮説" Two men are smiling and laughing at the cats playing on the floor."は「中立」のラベルが付与されている。前提" A man inspects the uniform of a figure in some East Asian country."と仮説" The man is sleeping"は「矛盾」のラベルが付与されている。前提"A soccer game with multiple males playing."と仮説"Some men are playing a sport."は「含意」のラベルが付与されている。

表 4.1　SNLI の例

Table 4.1　Examples of　SNLI.

Text	Judgments	Hypothesis
A man inspects the uniform of a figure in some East Asian country.	contradiction	The man is sleeping.
An older and younger man smiling.	neutral	Two men are smiling and laughing at the cats playing on the floor.
A black race car starts up in front of a crowd of people.	contradiction	A man is driving down a lonely road.
A soccer game with multiple males playing.	entailment	Some men are playing a sport.
A smiling costumed woman is holding an umbrella.	neutral	A happy woman in a fairy costume holds an umbrella.

[2] SNLI コーパスダウンロードサイト：https://nlp.stanford.edu/projects/snli/

本稿では、含意文生成をタスクにするため、「含意」ラベルが付与された含意関係である前提と仮説のペアを取り出してデータセットにする。取り出したデータの数は合わせて 190,113 件で、その中には、トレーニングデータが 183,416、開発データは 3,328 件、テストデータは 3,360 件がある。

表 4.2　データセット数

Table 4.2　Number of Dataset.

Train	Development	Test
183416	3328	3360

SNLIの訓練データにおいて、183416の前提文と含意文のペアの文間類似度は平均で0.7696である。図4.1で示されたように、[0, 1]の類似度の範囲に対して、データセットにおける類似度は[0.2, 1]の範囲に分布され、類似度が[0.5, 1]範囲内のデータは全体の99.446%を占めている。

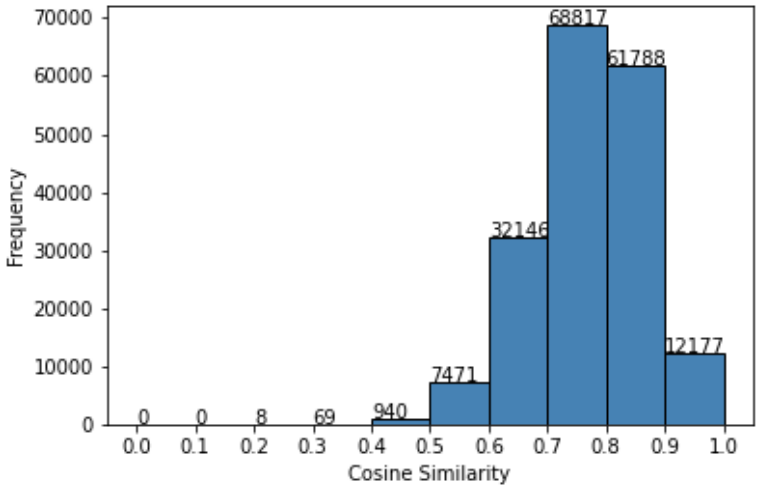

図 4.1　訓練データにおける類似度分布

Fig. 4.1　Frequency of Similarity in the Train Data.

以下の例文はデータセットにある文間類似度が0.9を超えた前提文と含意文のペアである。アンダースコアでハイライトされた部分では、前提文と含意文が全く一緒の単語を共有している。このような類似度が0.9を超えたデータは全体の6.639%を占めている。

前提文　*A　foreign　family is walking along a dirt path next to the water .*

含意文　*A　foreign　family walks by a dirt trail along a body of water .*

前提文　*A woman stands behind an outdoor grill with a blue basket of food in her hands .*

含意文　*A woman standing behind a grill outside with a blue basket of food in her hands .*

前提文　*Two children in hats play in an open , rocky field .*

含意文　*The children are playing in a rocky field .*

4.2 実験設定

本稿では、交差エントロピー誤差損失をベースライン手法はにする。提案手法は類似度損失と交差エントロピー誤差損失を組み合わせたものである。両方とも Attention based Sequence-to-Sequence モデルに基づいて含意文生成を行う。類似度損失の有効性を検証するため、ベースライン手法と比較しながら評価する。損失についての設定以外は、ベースライン手法と提案手法に全く同じようなハイパーパラメータで設定する。

エンコーダとデコータの隠れ層はそれぞれ 2 層の Bi-GRU と 2 層の GRU からなるものであり、隠れ層のサイズと単語埋め込みのサイズは 1,024 に設定し、0.2 の Dropout を採用した。注意層では、M. Loung ら[8]が提案した「Global Attention」と「General Attention Score」を採用した。訓練の段階で、オプティマイザは「Adam」を使用した。バッチサイズは 32 で、GPU 上で学習を行なった。

表 4.3　実験設定

Table 4.3　Settings of Experiment.

アイテム	ベースライン	提案手法
Word Embedding Dim	1024	1024
Encoder	2 層　Bi-GRU	2 層　Bi-GRU
Decoder	2 層　GRU	2 層　GRU
Attention	Global	Global
Attention Score	General	General
GRU Hidden Size	1024	1024
Dropout	0.2	0.2
Learning Rate	0.0001	0.0001
Optimizer	Adam	Adam
Regularization	L2	L2
Batch Size	32	32
Device	GPU	GPU
Loss Function	Cross Entropy Loss	Cross Entropy Loss + Similarity Loss

5 結果

5.1 訓練結果

提案手法とベースライン手法によって、それぞれ訓練データを 10 回学習させる。訓練中のトレーニング損失推移は以下となる。ベースライン手法において、交差エントロピー誤差損失は 3.647 から 2.592 まで下がった。提案手法において、交差エントロピー誤差損失は 3.529 から 2.580 まで下がり、類似損失は 1.079 から 1.035 まで下がった。

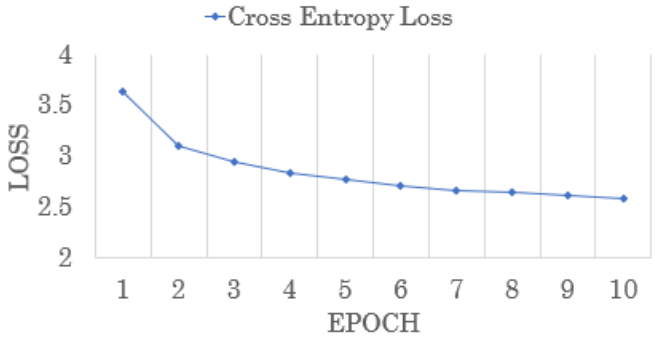

図 5.1　ベースラインモデルのトレーニング損失の推移

Fig. 5.1　Training Loss of the Bseline.

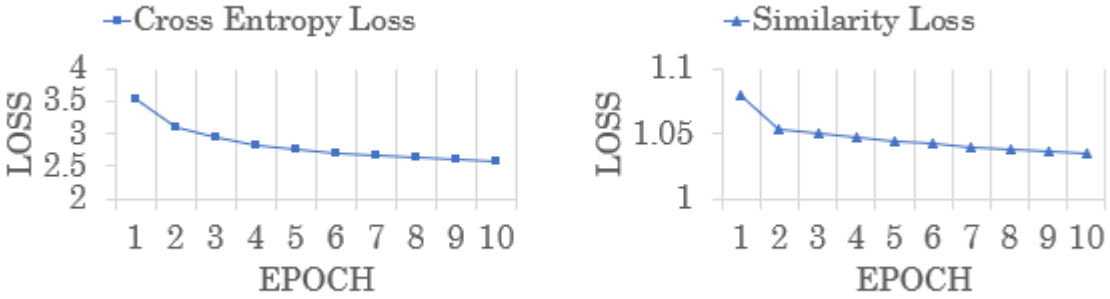

図 5.2　提案モデルトレーニング損失の推移

Fig. 5.2　Training Loss of the Proposal.

また、10 回訓練済みのモデルでテスト損失を求める。ベースライン手法において、交差エントロピー誤差損失は 3.529 である。提案法において、交

差エントロピー誤差損失は 3.590 で、類似損失は 1.063 である。

5.2 類似評価

前節「3.2 評価手法」で紹介した類似度評価手法で、ベースライン手法と提案手法について類似評価を行う。テストデータの平均類似度を計算して、結果は以下表 5.1 となる。ベースライン手法による類似度評価は 0.7517、提案手法による類似度評価は 0.7580 である。平均類似度の点からはベースラインと提案手法に大きな差は現れなかった。

表 5.1　実験結果

Table 5.1　Result of Experiment.

	ベースライン手法	提案手法
平均類似度	0.7517	0.7580

また、両手法で生成された含意文と前提文の類似度の分布を図 5.3 と図 5.4 に示す。類似度が 0.9 以上のものは減少していることがわかる。類似度 0.9 を超えたものにおいて、ベースライン手法は 106 件、提案手法は 90 件である。提案手法は類似度 0.9 を超えた含意文の生成を控えたことが分かった。

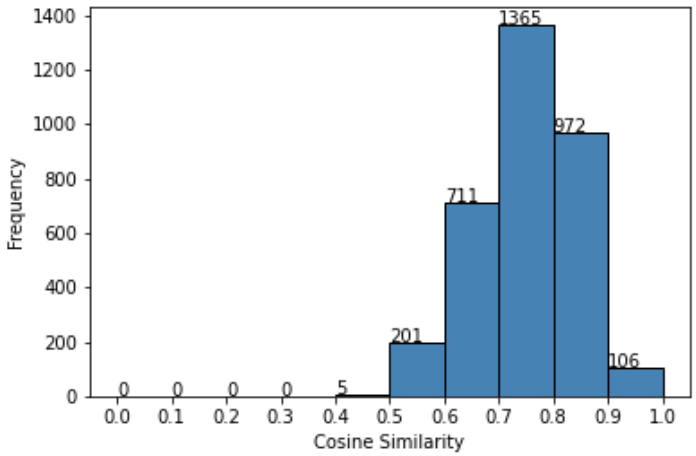

図 5.3　ベースライン手法

Fig. 5.3 Frequency of Textual Similarity with Baseline.

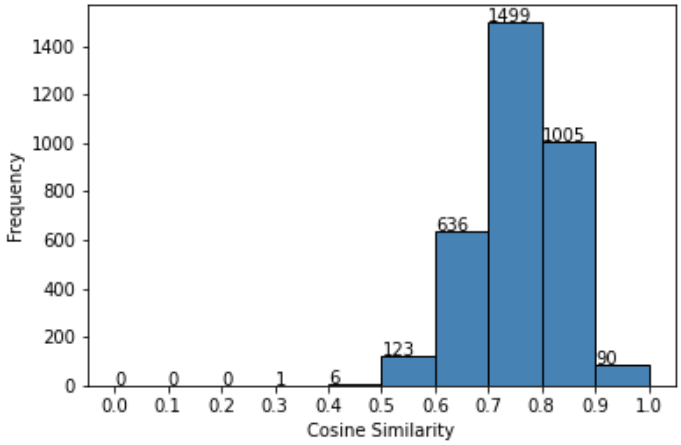

図 5.4　提案手法

Fig. 5.4 Frequency of Textual Similarity with Proposal.

ベースライン手法と提案手法による生成した含意文の例は以下なる。以下の例文で、提案手法において、前提文と生成された含意文との語彙の重なりが小さくなり、類似度が低くなる。

前提文：　*white dog playing in the snow .*
目標：　*the dog enjoys the snow .*
ベースライン：*a dog is playing in the snow .*
提案手法：　*a dog is outside .*

前提文：　*construction workers are standing upon on a wooden bridge .*
目標：　*the bridge is wood .*
ベースライン：*construction workers are standing on a bridge .*
提案手法：　*there are people on a bridge .*

前提文：　*two soccer players are going after the ball .*
目標：　*there are two soccer players .*
ベースライン：*two soccer players are going to the ball .*
提案手法：　*two soccer players are playing soccer .*

今後の課題としては、平均的に前提文と含意文との語彙の重なりを抑えるようにすることを目指す。また、学習回数を増やすことでどのように変化するかについても検討を行う。

6 おわりに

本論文では、できるだけ前提文と語彙が重ならないような含意文を作成することを目的とする。含意生成には、機械翻訳でよく用いられる Attention 付きの Sequence-to-Sequence モデルを用いることとする。さらに、前提文と含意文との語彙の重複を制限するため、前提文と生成された含意文の類似度に基づいた損失を定義し、それを用いることで目的を達成する。モデル予測の段階で、コーパスの含意文を教師信号として予測された含意文に交差エントロピー誤差を用いて学習される。この学習により、コーパスとして用意された含意文に生成される含意文の質は依存することとなる。前提文と生成された含意文との語彙の重なりを小さくすることで、前提文と異なる含意文の生成を目指す。このため、類似損失を提案する。提案手法を評価するために SNLI コーパスを用いて実験を行い、平均類似度の点からはベースラインと提案手法に大きな差は現れなかった。また、類似度 0.9 を超えたものにおいて、ベースライン手法は 106 件、提案手法は 90 件である。提案手法は類似度 0.9 を超えた含意文の生成を控えたことが分かった。前提文と語彙を共有するような含意文を抑えることができた。類似度が 0.9 以上のものは減少していることがわかる。

今後の課題としては、平均的に前提文と含意文との語彙の重なりを抑えるようにすることを目指す。また、学習回数を増やすことでどのように変化するかについても検討を行う。

謝辞

本論文を進めるにあたり、本研究を遂行する上で終始丁寧かつ熱心なご指導を賜りました柳本豪一准教授に心より感謝申し上げます。研究室に配属されて以来、自然言語処理に関する基礎知識や研究に取り組む姿勢等について一からご指導いただいたことに対して、柳本豪一准教授に改めて深く感謝いたします。副査を担当していただいた中島智晴教授、泉正夫教授に心より感謝申し上げます。最後に、いつも心の支えになってくれた両親に心から感謝します。ありがとうございました。

令和3年1月19日

参考文献

[1] Dagan I, Glickman O. Probabilistic textual entailment: Generic applied modeling of language variability. In *Proceeding of the PASCAL Workshop on Learning Methods for Text Understanding and Mining,* Grenoble, pp26-29, 2004.

[2] Takaaki Matsumoto, Kimihiro Hasegawa, Yukari Yamakawa, Teruko Mitamura. Textual Entailment based Question Generation. In *Proceeding of the Workshop on Intelligent Interactive Systems and Language Generation,* pp15-19, Nov.2018.

[3] Ramakanth Pasunuru, Han Guo, Mohit Bansal. Towards Improving Abstractive Summarization via Entailment Generation. In *Proceeding of the Workshop on New Frontiers in Smmarization*, Copenhagen, Denmark, pp27-32, Sep.2017.

[4] Tharaniya Sairaj R, Balasundaram S.R. An Entailment Analysis Based Entity Mapping To Improve Automatic Question Generation. In *Proceeding of CODS COMAD*, pp424, Jan2021.

[5] Han Guo, Ramakanth Pasunuru, Mohit Bansal. Soft Layer-Specific Multi-Task Summarization with Entailment and Question Generation. In *Proceedings of the Association for Computational Linguistics,* Melbourne, Australia , pp687–697, Jul.2018.

[6] Ramakanth Pasunuru, Mohit Bansal. Multi-Task Video Captioning with Video and Entailment Generation. *In Proceedings of the Association for Computational Linguistics*, Vancouver, Canada, pp1273–1283, 2017.

[7] I. Sutskever, O. Vinyals, QV Le. Sequence to Sequence Learning with Neural Networks. In *Proceeding of* NIPS, pp.3104- 3112, 2014.

[8] M. Loung, H. Pham, C. D. Manning. Effective Approaches to Attention-based Neural Machine Translation. In *Proceeding of EMNL*, pp.1412-1421, 2015.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceeding of the* Association for Computational Linguistics, pp.4171-4186, 2019.

[10] Maosheng Guo, Yu Zhang, Dezhi Zhao, Ting Liu. Generating Textual Entailment Using Residual LSTMs. In *Proceeding of CCL,*

pp.263-272, 2017.

[11] Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceeding of ICLR,* 2015.

[12] Sepp Hochreiter, Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation 9(8)*, pp.1735-1780, 1997.

[13] P.J. Werbos. Backpropagation through time: what it does and how to do it. In *Proceedings of IEEE*, pp.1550-1560, Oct.1990.

[14] David E. Rumelhart, Geoffrey E. Hinton, Ronald J. Williams. Learning representations by back-propagating errors. *Nature* 323, pp.533–536, 1986.

[15] Samuel R. Bowman, Gabor Angeli, Christopher Potts, Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, pp.632–642, Sep.2015.

[16] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, pp.1724–1734, Oct.2014.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceeding of Association for Computational Linguistics*, pp.4171-4186, 2019.

[18] Haoshen Fan, Jie Wang, Bojin Zhuang, Shaojun Wang, Jing Xiao. A Hierarchical Attention Based Seq2seq Model for Chinese Lyrics Generation. In *Proceeding of PRICAI*, pp.279-288, 2019.

[19] Lijun Wu, Fei Tian, Li Zhao, Jianhuang Lai, Tie-Yan Liu. Word Attention for Sequence to Sequence Text Understanding. In *Proceeding of AAAI-18*, pp.5578-5585.

[20] Ademi Adeniji, Nate Lee, Vincent Liu. Sequence-to-Sequence Generative Argumentative Dialogue Systems with Self-Attention. 2019.

[21] Haoyang Wen, Yijia Liu, Wanxiang Che, Libo Qin, Ting Liu.

Sequence-to-Sequence Learning for Task-oriented Dialogue with Dialogue State Representation. In *Proceedings of the 27th International Conference on Computational Linguistics*, Santa Fe, New Mexico, USA, pp.3781–3792, Aug.2018.

[22] Ruiyi Zhang, Changyou Chen, Xinyuan Zhang, Ke Bai, Lawrence Carin. Semantic Matching for Sequence-to-Sequence Learning. In *Proceeding of EMNLP*, pp.212-222, Nov.2020.

[23] Y. Bengio, P. Simard, P. Frasconi. Learning long-term dependencies with gradient descent is difficult. In *IEEE Transactions on Neural Networks (Volume: 5, Issue: 2)*, pp.157-166, Mar.1994.

[24] Razvan Pascanu, Tomas Mikolov, Yoshua Bengio. On the difficulty of training Recurrent Neural Networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning* ,pp. III1310-III1318, Jun.2013.

[25] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, Yoshua Bengio. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. In *Proceedings of NIPS 2014 Workshop on Deep Learning*, Dec.2014.

[26] Zhiyuan Tang1, Ying Shi1, Dong Wang. Visualization Analysis for Recurrent Networks. Center for Speech and Language Technologies, Research Institute of Information Technology, Tsinghua University. 2016.

[27] M. Schuster, K.K. Paliwal. Bidirectional recurrent neural networks. In IEEE Transactions on Signal Processing (Volume: 45, Issue: 11), pp.2673-2681, Nov.1997.

[28] Zhenyu Jiao, Shuqi Sun, Ke Sun. Chinese Lexical Analysis with Deep Bi-GRU-CRF Network. arXiv:1807.01882, Jul.2018.

[29] Vedran Vukotić, Christian Raymond, Guillaume Gravier. step beyond local observations with a dialog aware bidirectional GRU network for Spoken Language Understanding. In *INTERSPEECH 2016*, San Francisco, USA, pp.3241-3244, Sep2016.

[30] Cong Ma, Changshui Yang, Fan Yang, Yueqing Zhuang, Ziwei Zhang, Huizhu Jia, Xiaodong Xie. Trajectory Factory: Tracklet Cleaving and Re-connection by Deep Siamese Bi-GRU for Multiple Object Tracking. In *Proceedings of IEEE International Conference on Multimedia and Expo*, Jul.2018.

[31] Alex Graves, Jürgen Schmidhuber. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. In *Proceedings of 2005 IEEE International Joint Conference on Neural Networks*, Montreal, Que, Canada, Jul.-Aug.2005.

[32] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, Hannaneh Hajishirzi. Bidirectional Attention Flow for Machine Comprehension. In *Proceedings of ICLR*, 2017.