

論文 / 著書情報  
Article / Book Information

題目(和文)	自然言語生成における文構造の制御に関する研究
Title(English)	
著者(和文)	丹羽彩奈
Author(English)	Ayana Niwa
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第12462号, 授与年月日:2023年3月26日, 学位の種別:課程博士, 審査員:岡崎 直観,篠田 浩一,徳永 健伸,村田 剛志,井上 中順
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第12462号, Conferred date:2023/3/26, Degree Type:Course doctor, Examiner:,,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

# 自然言語生成における文構造の制御に関する研究

丹羽 彩奈



2023年3月1日

東京工業大学情報理工学院  
情報工学系 知能情報コース

本論文は東京工業大学情報理工学院に  
博士（工学）授与の要件として提出した博士論文である。

審査委員：

岡崎 直観 教授      (主指導教員)  
徳永 健伸 教授  
篠田 浩一 教授  
村田 剛志 教授  
井上 中順 准教授

## 概要

人間のように言葉を操れるコンピュータは、近年さらに現実的なものとなっている。コンピュータが人間の文章の読み書きなどの言葉に関する技術を再現できれば、人間とコミュニケーションをとったり、作文の支援ができるようになる。そして実際に、対話システムや機械翻訳、自動校正など、多くの自然言語生成アプリケーションが人々の生活を仲介し、支援するようになった。この成功の背景にあるのは、多層のニューラルネットワークからなるモデルを用いた深層学習技術の発展である。これにより、入出力間に存在する複雑な関係を学習できるようになり、多くの自然言語処理タスクの性能は飛躍的に向上した。

一方で、深層学習におけるブラックボックス問題が近年問題視されている。これは、深層学習システムがなぜそのような出力をしたのかを人間が理解できない、つまり解釈性に欠けることを指す。そのため、最新の深層学習に基づく自然言語生成技術においても出力文がどのような過程を経て生成されたのかを解釈することができない。この問題は、解釈性だけではなく、出力にある属性をもたせるための制御性の問題にまで波及する。特に自然言語生成においては、同じ内容を複数通りで表現可能であり、それによって文の難しさや効果が変化するため、制御性の高いシステムが強く求められる。しかし、ある属性を制御しようと思っても、それによって出力から制御の対象以外の情報が失われる可能性がある。その場合、その想定外の挙動を軽減するためには、文生成を高度に制御できることが必要である。

そこで本研究では、出力文にもたせたい文構造（単語や句の並びや関係）で書かれた文を制約としてモデルに与え、文生成を制御する。この制御に用いる文を本研究ではテンプレートと呼ぶが、これはスロットをもつ雛形文であるハードテンプレートと、スロットをもたない見本となる用例であるソフトテンプレートのふたつに大別される。そしてそれらのテンプレートに対して、スロットに単語を穴埋めしたり、単語を削除・挿入することで文を生成する。与えられたテンプレートを出発点として文生成を行うことで、生成される文の文構造を制御できるようになる。さらに、文生成の過程をテンプレートに対する一連の編集操作として人間が解釈できる。このテンプレートによって与える制約は、強いほど

忠実にテンプレートの内容を守って文を生成できるため、強い構造制約を持つ言語表現に特に有効である。

そこで、本研究ではまず強い制約を課せるハードテンプレートをモデルに与え、スロットを穴埋めすることで所望の文構造を確実にもった文を生成する解釈性の高い手法に取り組む。今回は、その強い構造制約をもつ文として対句と呼ばれる修辞技法のひとつを題材とし、その自動生成を行う。対句とは、「ひとに都合のいい空気は、地球に都合の悪い空気でした。」のように類似した文構造と対照的な意味を持つ文や句を並列させる技法である。今回用いるハードテンプレートは人手作成が主流だが、その場合コストがかかる。また、対句を生成対象とした研究はなく、言語資源がない。そのため、本研究では対句文の自動生成を対句データセットの構築、ハードテンプレートの自動抽出、そしてハードテンプレートへの穴埋めの3つのサブタスクに分割して取り組んだ。まず、高品質なデータ作成に向けて対句の判定基準を厳密に定義し、クラウドソーシングサービスを用いて対句データセットを構築した。次に自動抽出では、文中の対句に該当する範囲を計算量を削減するための段階的な構造解析アルゴリズムによって同定し、その部分をスロットとすることで対句テンプレートを得る。そしてハードテンプレートへの穴埋めでは、対句の構造を認識した上で単語を穴埋めできる手法を提案した。結果として、定量的・定性的な評価により、提案したハードテンプレートに基づく手法は対句文を生成できることを示した。しかし、この手法は所望の文構造ごとにスロット位置を事前に同定する必要がある。この予め決められたスロットの位置によって生成結果の多様性が大幅に減少してしまう。また、所望の文構造に限られた数のハードテンプレートでカバーできるという仮定に基づいており、適用可能なタスクが限定される。

そこで、次はスロットをもたないソフトテンプレートを弱い構造制約としてモデルに与え、その文構造を参考にしつつ書き換えることによって文を生成するアプローチをとる。大量の文対から検索してきたソフトテンプレートに対し、複数トークンを一度に生成可能な非自己回帰モデルを用いてトークンを挿入・削除することで文を生成する。提案手法は多くのソフトテンプレートが取得可能なタスクやデータセットにおいて特に性能を向上させられることがわかった。また分析結果によりソフトテンプレートによってモデル出力を弱く制御できていることが示唆された。

## 謝辞

本研究を遂行し、博士学位論文としてまとめるにあたって多くの方からご指導・ご支援を賜りました。この場を借りて厚くお礼申し上げます。

主指導教員の岡崎直観教授には、修士課程入学時から約5年にわたり、あたたかいご指導と激励を賜りました。元々外部から入学してきた私にとって、東工大での研究生活は決して平坦な道のりではありませんでした。しかし、常に私の意思を尊重しつつ導いてくださった岡崎先生のお力添えにより、どうにか博士論文をまとめることができました。色々なことにすぐ影響を受けてつまづき落ち込んでしまう私に、岡崎先生は優しくも的確なアドバイスをしてくださり、その度にどうにか立ち直ってここまでできました。ここまで手がかかる博士学生も珍しいのではないのでしょうか。お恥ずかしい限りです。金銭面でも多大なサポートをしていただき、自分のやりたいことに集中する環境を作ってくださいました。また、東工大岡崎研の初期メンバーとして、研究室が大きくなっていくさまを一緒に見られたのもとても良い経験でした。これからも多くの学生を輩出していくであろう岡崎研究室のOGとして、胸を張ってこれからも生きていきたいと思えます。これまでご指導いただき、本当にありがとうございました。

徳永健伸教授には、本論文の審査員としてだけでなく、同じ自然言語処理分野の先生として、審査や学会、日常生活で多くのコメントや激励のお言葉をいただきました。博士課程への進学や博士課程卒業の時期を迷っていたときにも、徳永先生は先を見据えた大きな視点からの的確なアドバイスをしてくださいました。自分の視野が大きく広がったと感じる瞬間が何度もありました。感謝申し上げます。篠田浩一教授には、特に生成の制御に対する本研究の立ち位置に関するコメントをいただき、自然言語生成分野における本研究の立ち位置について客観的に捉える一助となりました。また、予備審査後の励みのお言葉をくださったことで、改めて良い論文にしようと思いが強まりました。村田剛志教授には、

本研究で扱っている2種類のテンプレートの本質的な違いに関する重要なコメントをいただきました。3つの研究を1つの論文にまとめる上で重要となる論文全体の主張の一貫性の向上には、村田先生にいただいたコメントが必要不可欠でした。井上中順准教授には、すでにブラックボックスなモデルでうまく回っている世の中に対して何を生み出せるのかなど、特に実世界応用を見据えた本論文の立ち位置に関する鋭いご指摘を多くいただきました。早いスピードで発展する当該分野における本論文の役割について深く考えるきっかけをいただきました。

秘書・支援員の中山都子さん、佐藤あゆみさん、中川恵理子さん、小西由希子さん、雲財祐子さん、古谷奈緒子さんには、RAの勤怠申請や出張、英文校正や論文出版など研究遂行をスムーズに進めるための事務処理を支えていただきました。特別研究員奨励費の使い方についても事務とやりとりしながら一緒に試行錯誤してくださいました。また、日々の雑談もいつも楽しみにしていました。皆さんがいらっしゃるだけで研究室の空気が明るくなるようでした。

共同研究でお世話になりました、サイバーエージェントの西口佳佑さん、脇本宏平さん、亀山千尋さん、毛利真崇さんには、広告生成という切り口から自然言語処理技術の産業応用に関する研究をさせていただきました。実応用を見据えた、実際にそのシステムを使うクリエイターの方々の意見も伺いながら研究を進めるといふ共同研究ならではの多くの貴重な経験をさせていただきました。また、キャッチコピーという高度な作文技術が詰まった興味深いデータを触らせていただきました。感謝いたします。Megagon Labsの松田寛さんには、修士学生の頃からインターンシップなどで長きにわたりお世話になりました。最近の自然言語処理技術を中心に学んでいる私とは異なり、伝統的な言語学の知識を重んじつつも最新の技術を積極的に取り入れていく松田さんの研究興味やスタイルには私の視野を大きく広げていただきました。また、これらの研究業務をスムーズに進めるにあたり、前田圭一郎さん、柄川麻美さんには暖かいご支援をいただきました。

高瀬翔元助教には、ミーティングの組み立て方から研究の進め方・まとめ方まで研究のサイクル全体について多くのことをご指導いただきました。高瀬さんは学生の全員がほぼ自然言語処理初心者という状況の頃からいらっしゃいました。そのなかで国際会議に次から次へと論文を通していく姿を見て、現役の研究者とはこういうものかととても刺激を受けたことを覚えています。私が思う第一線で活躍し続ける研究者像は高瀬さんからの影響を大きく受けています。

研究生生活を共にしてくださった岡崎研究室のOB・OGを含む皆様からは多くの刺激と示唆を得ました。フルタイム博士課程学生の唯一の先輩であった平岡達也さんには、公私ともに多くの相談にのっていただきました。平岡さんが岡崎研博士第一号として先陣を切って道を切り拓いてくださったおかげで、私は平岡さんの背中を追いかける形で安心して研究生生活を過ごすことができました。また、弱音ばかり吐いている私を毎回凄まじい説得力で元気づけてくださりました。平岡さんが先輩としていらっしゃってくださって本当に良かったと思っています。これからは社会人仲間として、変わらず仲良くしていただけると嬉しいです。金子正弘研究員には、研究能力の高さだけではなく、自然言語処理研究の潮流を的確に捉える能力やチャンスをもものにする姿勢、指導力の高さなどに大きな刺激を受けました。元々NLP Dの会などで接する機会はありましたが、途中でポスドクとして岡崎研究室に入ってこられることを知った時はとても嬉しかったです。学生にも気さくに接してくださり、研究に悩んでいるときは真剣に相談にのってくださりました。研究者としても人間としても尊敬しています。修士の時から先輩である水木栄さんには、研究に対する姿勢やトピックに対する深い造詣をおもちで、多くのことを学ばせていただきました。困った時は水木さんに相談したら大抵のことは解決するのではと思っていましたし、実際に毎回予想以上の有益なコメントをいただいていた。ありがとうございます。楊之申さんと Erick Mendieta Molina さんとは、同学年の仲間として多くの悩みを相談し合いました。最近では直接話せる機会は減ってしまいましたが、私の中で二人の存在はとても大きかったです。谷口大輔さん、石川遼伍さん、昇夏海さんとは、一緒に研究をさせていただきました。共著者としてもっと上手くやれたのではないかと思うことも多々ありますが、全ての研究で論文をまとめるところまで楽しくやらせていただきました。Ma Youmi さんとは博士課程の女子学生仲間として仲良くさせていただきました。何かあるたびに直接お祝いしていただくこと、毎回嬉しく思っていました。博士課程では、COVID-19の影響もあり、自分なりに積極的に外とのつながりをつくったり多くのチャレンジをすることを特に意識していましたが、これは「自分には岡崎研究室という絶対的な帰る場所がある」という安心感があったからこそできたことです。皆さんとの研究に関する議論や日々の何気ない雑談には、精神的に大きく支えられました。岡崎研究室の皆さんは一人一人が多様な趣味や能力を持っていて、接していて本当に楽しかったですし、全員から学ぶことができました。同時期に同じ研究室に所属

し、多くのことを学ばせていただいたこと、大変幸運だったと思っています。

最後になりますが、日々の生活を健康面と経済面の両面からいつも支えてくださりました、父、母、妹、親戚の皆様には深く感謝いたします。私が博士課程進学を決めた時は大変驚いたと思いますが、何も言わずに今まで応援してくれました。そして深夜の作業に根気よく付き合ってくれたチワワのまめに感謝いたします。

# 目次

<b>1</b>	<b>序論</b>	<b>1</b>
1.1	人間と協働する自然言語生成システム	1
1.2	ニューラルネットワークによる自然言語生成の問題	2
1.3	本研究のアプローチ：文構造の制御	3
1.4	本研究の貢献	7
1.5	論文の構成	8
<b>2</b>	<b>関連研究と準備</b>	<b>9</b>
2.1	制御可能な文生成	9
2.2	準備	12
2.2.1	Transformer	12
2.2.2	BERT	15
<b>3</b>	<b>ハードテンプレートに基づく文生成</b>	<b>18</b>
3.1	コーパス構築	20
3.1.1	キャッチコピーデータの収集	20
3.1.2	対句アノテーション	22
3.1.3	対句データセットの統計	24
3.2	対句構造解析によるテンプレート抽出	26
3.2.1	対句構造の言語学的性質	27
	文構造の類似性	27
	句の可換性	28
	句の意味的対照性	28
3.2.2	対句構造と並列構造	28

3.2.3	ハードテンプレートの獲得に関する関連研究 . . . . .	29
3.2.4	対句構造解析に関する関連研究 . . . . .	31
3.2.5	対句構造解析の困難さ . . . . .	34
	語句の省略 . . . . .	34
	対句の統語範疇 . . . . .	34
	対句の出現位置 . . . . .	35
3.2.6	構造解析手法 . . . . .	36
	タスクの定義 . . . . .	36
	対句構造解析における問題と本研究のアプローチ . . . . .	37
	提案手法 . . . . .	37
3.2.7	実験設定 . . . . .	42
3.2.8	実験結果 . . . . .	44
3.2.9	事例分析 . . . . .	46
3.2.10	解析結果からの知識抽出 . . . . .	46
	対句テンプレートの抽出 . . . . .	47
	言語知識の抽出 . . . . .	48
3.3	テンプレートの穴埋めによる対句生成 . . . . .	50
3.3.1	ハードテンプレートへの穴埋めに関する関連研究 . . . . .	51
3.3.2	提案手法 . . . . .	53
3.3.3	実験設定 . . . . .	54
3.3.4	実験結果 . . . . .	56
3.3.5	事例分析 . . . . .	58
3.4	ハードテンプレートに基づく文生成 まとめ . . . . .	59
<b>4</b>	<b>ソフトテンプレートを用いた文生成</b> . . . . .	<b>61</b>
4.1	関連研究 . . . . .	65
4.1.1	ソフトテンプレートに基づく文生成 . . . . .	65
4.1.2	非自己回帰モデルによる文生成 . . . . .	66
4.2	提案手法 . . . . .	67
4.2.1	問題設定 . . . . .	68
4.2.2	準備: Levenshtein Transformer . . . . .	69
	方策分類器 . . . . .	69

学習時 . . . . .	70
4.2.3 ソフトテンプレートの検索 . . . . .	71
語彙マッチング . . . . .	71
文ベクトルの類似度 . . . . .	72
4.2.4 ソフトテンプレートに対する編集操作 . . . . .	72
ソフトテンプレートを編集するためのオラクル方策 . . . . .	72
推論時のソフトテンプレートの活用方法 . . . . .	75
4.3 実験 . . . . .	75
4.3.1 実験全体の詳細 . . . . .	75
4.3.2 ベンチマークでの実験 (JRC-Acquis) . . . . .	77
4.3.3 他のデータセットでの実験 . . . . .	80
Data-to-text タスク (WikiBio) . . . . .	80
機械翻訳タスク (WMT'14) . . . . .	81
4.3.4 ソフトテンプレートの分析 . . . . .	82
4.3.5 実際の実出力例 . . . . .	84
4.4 ソフトテンプレートに基づく文生成 まとめ . . . . .	85
<b>5 結論</b>	<b>86</b>
<b>参考文献</b>	<b>93</b>

# 第 1 章

## 序論

### 1.1 人間と協働する自然言語生成システム

人類は、自分たちのように言葉を操れるコンピュータの実現を長年追及してきた。コンピュータが人間の文章の読み書きなどの言葉に関する技術を再現できれば、人間とコミュニケーションをとったり、人間の言葉に関わる作業を支援することができるようになる。これらの技術は、元々は遠い未来の話とされていたが、近年の深層技術をはじめとするさまざまな技術の発展により現実的なものとなりつつある。

与えられた入力情報から文を生成する技術を自然言語生成と呼ぶ (McDonald 1993, Evans et al. 2002)。これは生成に必要な能力に合わせて以下の例をはじめとするいくつかの形式（タスク）に分類される。

**機械翻訳** ある言語から別の言語へ翻訳する (Hutchins and Somers 1991)

**自動要約** あるテキストを簡潔なテキストへ変換する (Clarke and Lapata 2010)

**文法誤り訂正** 文法的に間違ったテキストから正しいテキストへ変換する (Kukich 1992)

**対話システム** ユーザーの発話に対する回答を生成する (Williams et al. 2005)

**ニュース記事生成** 試合結果や天気予報データからニュース記事を生成する (Theune et al. 2001a, Goldberg et al. 1994)

**広告文生成** 商品情報から広告文を生成する (Özbal et al. 2013)

自然言語生成技術は、深層学習の発展に合わせて近年性能が飛躍的に向上し、産業応用が大きく進んでいる分野の一つである。例えば、コミュニケーションロボットでは従来の非言語コミュニケーション（ジェスチャーや接触行動など）に加え自由な発話が可能な会話ロボットが製品化されている\*。また、通信社がパターン化可能な書類を自動生成するシステムを導入することで人間が取材などのよりクリエイティブな作業に時間を割けるようにしたり (Faggella 2018)、高性能な機械翻訳† や自動校正‡ によって作文作業を効率化させるサービスが幅広く使われている。つい最近には、もっともらしい回答を生成する能力が既存のシステムよりもさらに向上した対話システム ChatGPT (OpenAI 2022) が大きな反響を呼んだ。このように自然言語生成技術を取り巻く状況は日々変化し、コンピュータと人間の協働 (Lertvittayakumjorn and Toni 2019) を推し進めている。

## 1.2 ニューラルネットワークによる自然言語生成の問題

自然言語生成技術の産業化成功の背景には、高い予測性能を持つ深層学習/ディープニューラルネットワーク技術の発展がある。深層学習では、多層のニューラルネットワークからなるモデルを用いて入出力間に存在する複雑な関係をエンドツーエンドで学習することができる。これにより、従来手法の性能を大きく上回り、商業化可能なレベルでテキストを生成できるようになった。

一方で、ニューラルネットワークは複雑かつ非線形な構造を持つため、なぜそのような出力をしたのかがわからず結果の解釈可能性に欠ける。このように性能を向上させるためにモデルの解釈性を犠牲にしているニューラルネットワークの手法はブラックボックス手法と呼ばれる (図 1.1-(b))。

これは自然言語生成においても同様の現象が見られる。ニューラル化以前の自然言語生成モデルは、内容を決定し (テキスト・プランニング)、それをどのように表現するかを決定する (マイクロ・プランニング+表層化) ふたつのサブタスクからなる解釈可能なホワイトボックス手法であった (図 1.1-(a)) (Reiter and Dale 2000)。これにより、そのモデルがどの段階で生成を誤ったのかや、どういう過程を経てその出力がなされたのかを把握できるため、システム

---

\*<https://romi.ai/>

†例：<https://www.deepl.com/translator>

‡例：<https://app.grammarly.com/>

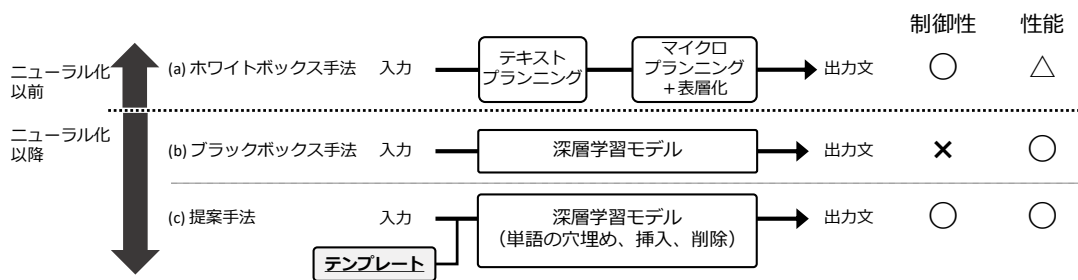


図 1.1: 自然言語生成手法とそれらの制御性

の開発者は性能向上の方針を立てやすく、ユーザーのシステムの予測に対する信用を高めることができる (Adadi and Berrada 2018, Clinciu and Hastie 2019). それニューラル化後はテキストが生成された過程を解釈できないブラックボックスなモデルになってしまった. この結果, ニューラル文生成モデルは入力に矛盾する内容や含まれない内容を出力してしまう (Ji et al. 2022) といった想定外の挙動をするようになった. この挙動の不安定さは, 実用的かつ高度なテキスト生成技術として近年重要視されている文生成の制御 (Zhang et al. 2022) を困難にする. 文生成の制御とは, 特定の単語や文長, 感情極性など特定の性質を出力文にもたせる技術であり, ニューラル文生成モデルに残された大きな課題のひとつとされる (Li et al. 2022). しかし, ある性質を制御しても, それによって出力から制御の対象以外の情報が損なわれることがある (Wiseman et al. 2018). 例えば, ある単語を含むように制御した結果それ以外の重要な単語が出力文から消えてしまったり, ある言い回しを含むように制御した結果流暢性に欠けてしまう. この想定外の挙動をできる限り抑制するには, 高度に生成文を制御する技術が求められる.

本研究では, この文生成の制御を目的とし, まずは単文の生成に取り組む. なお, 複数の文の意味的まとまりからなり生成の難易度が高い談話は本研究の対象からは除外し, 今後の課題とする.

### 1.3 本研究のアプローチ：文構造の制御

制御の対象は, 出力文の内容 (含めたい単語やフレーズ) と文体 (表現方法全般) の二つに大別される. なかでも本研究では, 後者に分類される**文構造**を制約としてモデルに与え, その制約を満たすように, すなわち出力文が与えられ

た文構造を持つように生成を制御するアプローチを提案する。文構造を表す代表的なものには、句構造情報（図 1.2左. 構文木と呼ぶ）や単語同士の修飾関係（図 1.2中. 依存構造と呼ぶ）が挙げられるが、本研究で対象とする文構造は図 1.2右の単語や句の並びや関係である。これは、構文木や依存構造とは異なり文法情報を持たないため外部の解析器が不要である。また、この文構造はトークンの系列で構成されるため、テンプレートと呼ばれる文で表現することができる。このテンプレートには、ハードテンプレートとソフトテンプレートの二種類がある。前者のハードテンプレート<sup>§</sup>は、図 1.3に示したように「スロット」をもつ雛形文のことで、スロットを穴埋めすることで文を生成する。一方後者のソフトテンプレートは見本となる用例のことで、ハードテンプレートとは異なり使用方法は明示されておらず、その用例を参考にして文を生成する（図 1.4）。本研究で制御する文構造は、ハードテンプレートとして与える系列や、ソフトテンプレートとして与える生成したい文に類似した文に該当する。

本研究では、所望の文構造をもつテンプレートを制約としてモデルに与え、テンプレートに適合する単語を穴埋め・削除・挿入して文を生成することで、出力文がその文構造を含むように制御する手法を提案する（図 1.5）。このテンプレートによる文生成の制御は、与えられたテンプレートに基づいて文を生成させる技術と捉えることができる。ハードテンプレートに基づく手法では与えられた文構造を必ず含むように生成されるが、ソフトテンプレートに基づく手法では必ずしもそうとは限らない。そのため両者は制約の強さが異なる制御手法である。また、今回扱う文構造の制御は、「どのように表現するか」を決定する大域的な制約であることから、パイプライン手法での表層化に該当する。自然言語生成では同じ内容を複数通りで表現可能であり、それによって文の難しさや効果が変わる。そのため、文構造で高度に制御可能なテキスト生成手法を実現することは、ユーザが文のスタイルや難易度を自由に変更できるシステムの開発にもつながる。また、文構造は個々の単語ではなくより大域的な系列情報を持つため、文生成の探索空間を効率的に狭められ、生成される文の品質の向上が期待される。さらには韻 (Watanabe et al. 2018, Li et al. 2020) や話者の言葉遣い (Chen et al. 2019) といった特定の構造制約やフォーマットをもつタスク・ドメインにおいてその文体的一貫性を高める効果もある (Deng et al. 2021)。こ

---

<sup>§</sup>穴埋め形式のテンプレートとも呼ばれる

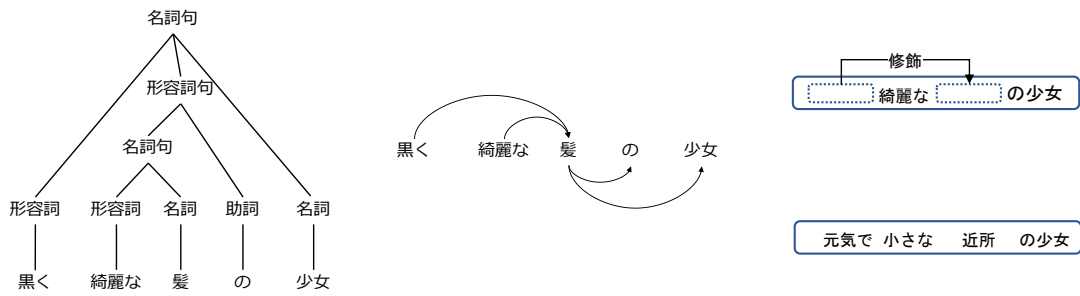


図 1.2: 文構造の種類. 左から構文木, 依存構造, 本研究が対象とする文構造.

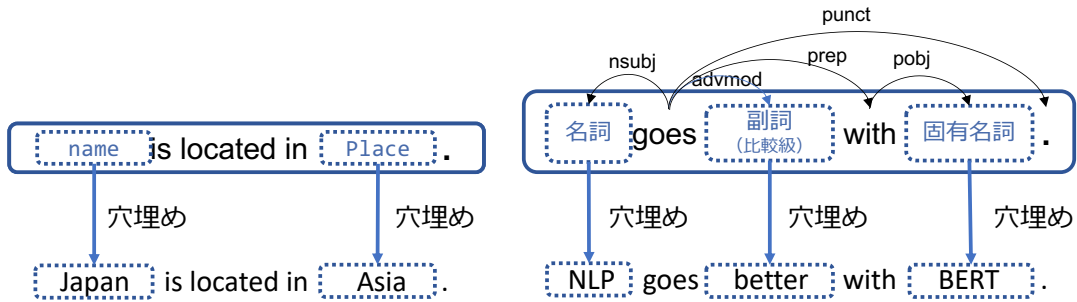


図 1.3: ハードテンプレートの例. 本研究の対象外ではあるが, 右図のように依存構造情報が与えられることもある.

のように, 文構造は文の要素として重要度が高く, その制御も文生成において多くの実用上のメリットがある.

テンプレートはこれまでも文生成の制御で多く用いられてきた. まずハードテンプレートは穴埋め形式で使うことが想定されていることから元々大変制御性の高い手法である. しかしハードテンプレートは人手作成したり, 解析器や単語リストなどを用いて個々のトークンをスロット化するべきか否か判定する手法が主流であり (Ong et al. 2008, Cicekli and Güvenir 2001, Sitompul et al. 2020), 文構造のようなより大域的な性質には対応できない. これに対し, テンプレート内のスロットを含む全ての単語に文構造情報, 正確には依存関係情報を自動付与した研究はあるが, これはコストの面で実用的でない (Hadaytullah et al. 2018). ソフトテンプレートでは, その情報をベクトルに圧縮した上で追加情報としてモデルに入力するブラックボックスなアプローチが主流である (Cao et al. 2018, Gu et al. 2018b). そのため, それらのテンプレートを具体的にどのように用いて文を生成したのかは把握できない. また, 文構造の制御を行った既存研究で

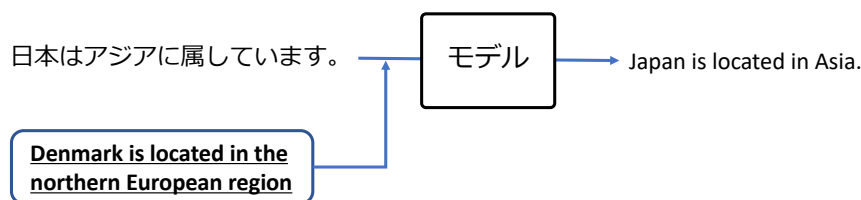


図 1.4: ソフトテンプレートの例. ソフトテンプレートにある ‘○ is located in ○.’ という文構造を参考に文を生成している.

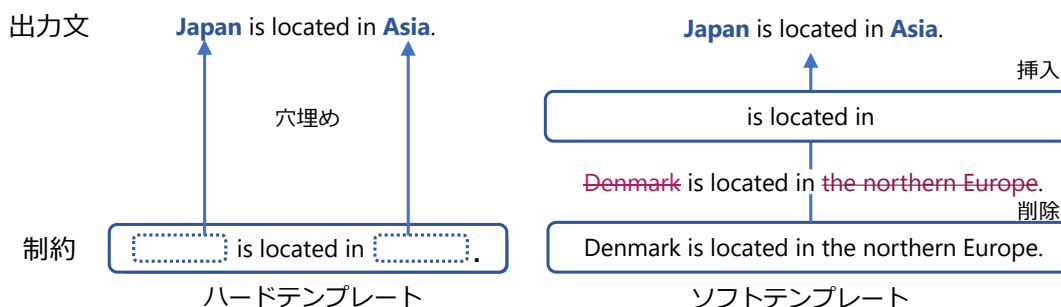


図 1.5: 本研究のアプローチ. 出力文にもたせたい文構造をテンプレートとして与え, その文構造をベースに編集操作を行うことで文を生成する.

は主に外部の解析器で得た木構造をモデルに入力するアプローチをとっているが (Iyyer et al. 2018, Akoury et al. 2019, Yamada and Knight 2001), これらの手法も同様に生成過程がブラックボックスである. しかし提案手法は文生成の全過程をテンプレートを出発点とした一連の編集操作 (単語の穴埋め, 削除, 挿入) として把握できるホワイトボックスな手法である. 生成過程が把握可能なホワイトボックス手法では, 誤りを特定したり, どのような編集でその出力がなされたのかが理解できる. 一般の文生成システムにおいて, ホワイトボックス手法であることは開発者とユーザー双方にメリットがある.

テンプレートを用いる制御では, ハードとソフトどちらを用いるかによって与えられる制約の強さが異なる. そして, 与える制約が強いほどテンプレートの内容を忠実に守って文を生成することから, 強い構造制約を持つ言語表現に特に有効である. そこで本研究では, まず特定の文構造制約を**確実に**満たした文を生成するため, 強い構造制約を課すことができる**ハードテンプレート**を用いた文生成手法を提案する. 今回は, 与える文構造制約として対句と呼ばれる表現技法を採用する. 対句とは, 「ひとに都合のいい空気は, 地球に都合の悪い空気

でした。」のように類似した文構造と対照的な意味を持つ文や句を並列させる技法である。構造的性質に加えて意味的な制約を併せ持つため、生成の難易度が高く高度な制御技術を要する。対句は、その言語的性質の複雑さや言語資源の不足により、今まで生成対象として研究されてこなかった。そのため、本研究ではまず、対句判定ルールを標準化した上でクラウドソーシングサービスを用いて対句コーパスを構築した。次に、対句文の生成に向け、対句の言語的性質に基づいたテンプレートの自動抽出手法および近年主流なアプローチである事前学習済みモデルを対句生成に適応させるための穴埋め手法を提案した。実験結果より、所望の構造の言語的性質を活用した手法はテンプレートの抽出と穴埋め両者にとって有効であること、また定性評価を含む評価によりテンプレートを用いることで所望の文構造をもつ文が生成できていることを確認した。本アプローチは、テンプレートのスロットに穴埋めすることで確実に制約を守った文を解釈性を保ちながら生成できる。

しかし、この手法は所望の文構造ごとにスロット位置を事前に同定する必要がある。また、テンプレートはスロットの位置が予め決まっておき活用方法に強い制約があるため、生成結果の多様性が大幅に減少してしまう。さらに、所望の文構造が限られた数のハードテンプレートでカバーできるという仮定に基づいており、適用可能なタスクが限定されるという問題もある。

そこで、ハードテンプレートを与えずに弱い文構造制約で制御する方法として、次は**ソフトテンプレート（見本となる用例）**を用いた文生成を行う。これは、ソフトテンプレートとしてスロットをもたない既存の文をモデルに与え、その文構造を参考にしつつ書き換えることによって文を生成する方法である。ここでは、ソフトテンプレートを大量の文対からなるデータストアから検索する。そして、複数トークンを一度に生成可能な非自己回帰モデル (Gu et al. 2018a) を用いて用例に対してトークンを挿入・削除することで文を生成する。提案手法は、多くのソフトテンプレートが取得可能な JRC-Acquis データセットを用いた機械翻訳タスクや WikiBio データセットを用いた Data-to-text タスクにおいて特に性能を向上させられることがわかった。また、分析結果によりソフトテンプレートによってモデル出力を弱く制御できていることが示唆された。本アプローチは、ソフトテンプレートに対して直接編集操作を行うことで、各編集段階でどのトークンが挿入・削除されたかを把握可能である。

以上のように、文構造の制御にあたり解釈性を併せ持ったハード・ソフトテ

ンプレートの活用法を提案し、それぞれの有効性を示した。

## 1.4 本研究の貢献

本研究の貢献は以下の通りである。

**構造解析による対句ハードテンプレートの自動抽出手法の開発** 対句の構造的・意味的な言語的性質を特徴量とした構造解析によるハードテンプレートの自動抽出手法を提案した。構造解析アルゴリズムの計算量を削減するため、段階的な解析を行うことで高速かつ高精度な手法を実現した。

**言語的性質に基づくハードテンプレートの穴埋め手法の提案** 事前学習済みモデルを対句穴埋めタスクに適応させるための学習方法を提案した。定量的・定性的な評価により、提案した手法は対句として適切な単語を穴埋めできることを示した。

**対句を生成対象とした初めての試み** 対句判定ルールの標準化やクラウドソーシングを用いたデータの作成、手法提案、実験、評価まで一連の研究を行った。

**解釈可能なソフトテンプレート手法の実現** ソフトテンプレートを単語の挿入・削除操作によって直接的に書き換えることで、文生成の全過程を解釈可能とするはじめての研究である。

**非自己回帰モデルにおけるソフトテンプレートの有効性の検証** 本研究は、非自己回帰モデルにおいてソフトテンプレートを活用することで解釈性を損なわずに生成文の品質および生成速度を向上させられることを示したはじめての研究である。

## 1.5 論文の構成

第2章では制御可能な文生成に関する関連研究の概観について述べる。次に、第3章「ハードテンプレートに基づく文生成」では、所望の文構造の言語的性質を利用したテンプレートの自動抽出およびスロットへの穴埋めによる文生成手法

を提案する。さらに第4章「ソフトテンプレートに基づく文生成」では、検索した用例に対しトークンを削除したり挿入したりといった編集操作を施すことによりソフトに文構造を制御する手法を提案する。最後に第5章では、本論文の結論と残された研究課題について述べる。

## 第 2 章

### 関連研究と準備

本章では，自然言語生成における制御性に関する既存研究に触れた後，以降の章に必要な準備として基盤となるモデルを紹介する．なお，ハードテンプレートとソフトテンプレートに関する関連研究は，各章内で紹介する．

#### 2.1 制御可能な文生成

主流な深層学習モデルは，学習データの分布を考慮して最も確率の高いトークンを生成して文を構成する．そのため流暢な文が生成できるが，生成結果の制御性は低く，ユーザーが望む性質をもつ文を生成することが難しい (Ji et al. 2022)．そこで，文生成においてモデルの出力分布を誘導し，所望の性質を持った文を生成できるように制御する手法が注目を浴びている．制御対象としては，例えば物語生成におけるエンディング (Egidi and Gerrig 2009) や展開 (Li et al. 2013)，応答生成における話者の言語特性 (Wang et al. 2017)，詩生成における韻 (Ghazvininejad et al. 2017) といったタスク特化のものだけではなく，より一般的な言語現象を制御対象とする研究も多く行われている．このより一般的な制御対象とは，内容の制御と文体の制御の 2 種類である．

内容の制御とは，具体的には含めるべき単語やフレーズを事前に選択し，それらを含むように文を生成する技術である．Kiddon et al. (2016) は，含めたい単語を出力文に確実に含ませるために，文生成モデルに既に生成された単語や生成すべき単語の情報を保持するニューラルチェックリストモジュールを組み込み，各タイムステップでその情報を参照することで高度に内容を制御することを可能にした．Mangrulkar et al. (2018) は，まず複数の出力候補文を生成した後，

必要な単語が含まれない場合や無関係な単語が含まれた場合にペナルティを課して再ランキングするアプローチを提案した。

一方文体とは表現方法全般のことを指し、その制御対象も多岐にわたる。例えば、Hu et al. (2017) は変分オートエンコーダの潜在空間において文の時制や感情極性などの属性表現と文表現を分離することでそれらの属性を制御するアプローチをとった。Martin et al. (2020) は、文の内容を変えずに文法や構造を単純化することで読みやすい文を生成するため、入力文に文字数や文構造の複雑さなどを表す特殊トークンを連結することで制御した。この特殊トークンによるアプローチは、モデルのアーキテクチャを変えずに生成文を制御する単純かつ効果的な手法であり、他にも機械翻訳における文の丁寧さの制御 (Sennrich et al. 2016a) や、文長や記事の出典などをユーザーの好みに合わせた生成型要約 (Fan et al. 2018) でも用いられている。Kikuchi et al. (2016) は生成文の長さを制御するため、追加の入力情報として長さベクトルを与えるアプローチを提案した。

近年では、大規模コーパスで事前学習されたモデルを所望の下流タスクに適応させる（これをファインチューニングと呼ぶ）際に生成文を制御するための取り組みも多く行われている。ここでは、事前学習で得た言語モデリング能力をファインチューニング時に破壊せずに、タスクごとの学習を行う必要がある。Krause et al. (2021) は、指定した属性をもつトークンを生成するための比較的小規模な言語モデルを別途学習し、その予測確率を事前学習済みの予測確率に組み合わせることで少ない追加学習コストで大規模事前学習済みモデルを制御した。この比較的小規模なモデルを追加学習して制御に用いる方法は、その効率の良さから有望視されている (Dathathri et al. 2020)。感情や単語の有害さ、トピックという複数の属性を同時に制御可能な手法として、Qian et al. (2022) らは事前学習済みモデルの入力側の先頭を制御したい属性のベクトルとして割り当て、そのベクトルのみを学習させた。これにより、追加モジュールなしに、また大規模事前学習済みモデルのアーキテクチャを変更せずに軽量の学習によって制御するアプローチを提案した。Ficler and Goldberg (2017) は、複数の属性をキーと値のペア集合（例：Theme:Actor, Sentiment:Positive）として定義し、それぞれの値をエンコードしたベクトルで条件づけて言語モデルを用いるアプローチをとった。

また、本研究のように、文構造を制御する研究もいくつかあるが、外部の解析器によって得られた木構造（を系列化したもの）を用いる手法が主流である。

ニューラル化前からMcKeown (1983)のように人手作成したルールに基づき文構造を制御した研究はあった。Stahlberg et al. (2016)は、従来からあったHierarchical文法 (Chiang 2007) をニューラル機械翻訳モデルに統合することで構文の制御を行った。Iyyer et al. (2018)は、文を構文木に変換し、系列化したもの (例: "She drove home." を (S(NP(PRP))(VP(VBD)(NP(NN)))(.)) に変換) をモデルの追加の入力情報として用いた。文から構文木を予測する学習器を導入し、予測された構文木を用いて文を生成するアプローチもその後提案されている (Akoury et al. 2019, Yamada and Knight 2001)。しかし、これらの手法は外部の構文解析器が必要であり、生成性能が解析器の性能に依存するうえ、解析にコストがかかる。これらに対し、本研究で用いるハードテンプレートは文中のスロットの位置のみを同定することによって得られるため、厳密な構文解析\*が不要である。また、ソフトテンプレートは既存の文をそのまま用いるため解析器は一切不要となっている。なお、解析器を用いずに文構造を制御する研究としては、Chen et al. (2019)のアプローチが最も近いが、その構文制約を潜在表現として暗黙的に扱う彼らの研究とは異なり、本研究はテンプレートを直接操作することで生成を解釈可能な形で制御可能な手法を提案している。

また、解釈可能な文生成もいくつか行われている。ニューラル化以前の自然言語生成モデルでは、生成過程を「何を言うか」を決めるテキスト・プランニングと「どう言うか」を決めるマイクロ・プランニング+表層化のふたつのサブモジュールに分割していた (Reiter and Dale 2000)。この段階的な生成により、文生成過程は解釈性の高いものとなっていた。そこで、近年では従来のニューラル化以前のサブモジュールに基づく文生成が改めて注目を浴びている。具体的には、Reiterらの自然言語生成パイプライン (Reiter and Dale 2000) をニューラル自然言語生成モデルに統合し、サブモジュールが生成する解釈可能な中間表現 (生成文に与えたい制約条件) を用いて文を生成する試みが行われている (Faille et al. 2020, Castro Ferreira et al. 2019, Mayn and van Deemter 2020)。中間表現には、例えば含めたい内容を表す単語のリスト (Shen et al. 2019) や論理形式 (Liu et al. 2021)、そして本研究で用いるテンプレート (Wiseman et al. 2018) などが挙げられる。以下にShen et al. (2019), Liu et al. (2021), Wiseman et al. (2018) らの論文内での例を一部抜粋して示した。

---

\*文の階層構造や品詞情報を得ることを厳密な構文解析とする

**記事** The sri lankan government on Wednesday announced the closure of government schools with immediate effect as a military campaign against tamil separatists escalated in the north of the country.

**単語選択** sri lankan, schools, military campaign

**生成結果** sri lanka shuts down schools amid war fears.

**論理形式**  $\text{eq}\{\text{count}\{\text{filter\_eq}\{\text{all\_rows}; \text{Silver Medal}; 3\}\}; 2\} = \text{True}$

**生成結果** There are two nations getting 3 silver medals in the game.

**テンプレート** `[slot]` is a `[slot]` providing `[slot]` food in the `[slot]` price range.

**生成結果** Cotto is a coffee shop serving English food in the moderate price range.

このように、解釈可能な制約を与えることで、解釈性と制御性の両立が可能となる。実際に、これらのような中間表現を経て生成する方がエンドツーエンドで生成するよりも高品質な文ができることも示唆されている (Castro Ferreira et al. 2019)。本研究でも、これらの知見に基づきニューラルモデルにテンプレートという解釈可能な制約を組み込むことで文生成を制御する。

## 2.2 準備

### 2.2.1 Transformer

本研究で扱う全ての提案手法は、Transformer (Vaswani et al. 2017) と呼ばれるモデルを基盤構造に採用している。Transformer は、近年発表されているほぼ全てのモデルで採用されている汎用性の高いニューラルモデルである。本節では、その Transformer について紹介する。

Transformer はエンコーダ・デコーダモデルの一つであり、入力系列  $\mathbf{X} = (x_1, x_2, \dots, x_n)$  と目的系列  $\mathbf{Y} = (y_1, y_2, \dots, y_m)$  が与えられた時に、 $\mathbf{X}$  を  $\mathbf{Y}$  に変換する目的関数  $p(\mathbf{Y}|\mathbf{X})$  を学習する。まずエンコーダが入力系列  $\mathbf{X}$  を中間表現  $\mathbf{Z} = (z_1, z_2, \dots, z_n)$  に変換する。次に、デコーダがその中間表現  $\mathbf{Z}$  を出力系

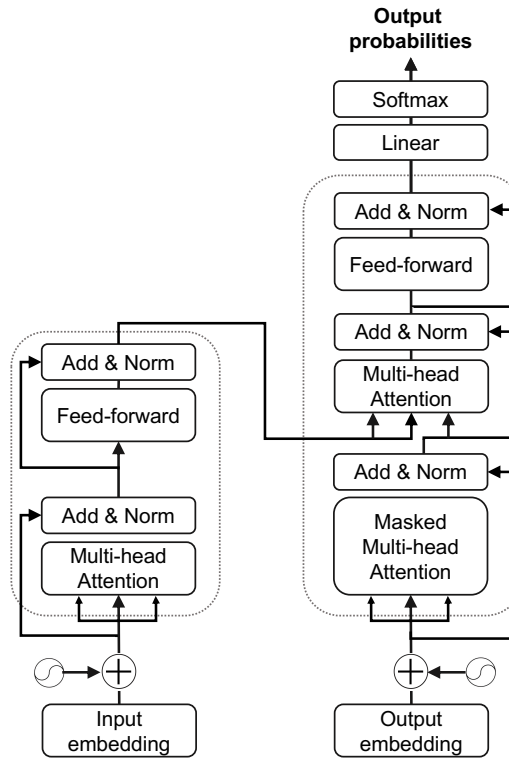


図 2.1: Transformer のアーキテクチャ

列  $\mathbf{Y}$  に変換することで文を生成する。Transformer は時間ステップごとに前ステップまでに生成された系列を参照しながらトークンを生成する自己回帰モデル (Sutskever et al. 2014) に位置付けられる。

エンコーダとデコーダは、図 2.1 に示したようにそれぞれ複数のブロックを積み重ねた構造になっている。エンコーダの各ブロックは、自己注意機構（以下 self-attention と呼ぶ）、単語位置ごとの全結合層（Feed Forward）の 2 つのサブブロックで構成される。一方デコーダのサブブロックは、エンコーダの 2 つのサブブロックの間に入力系列と目的系列間の注意機構（以下 source-target attention と呼ぶ）を加えた 3 つのサブブロックで構成される。エンコーダ・デコーダの各サブブロック間には、ブロックの正規化と残差接続が行われる。

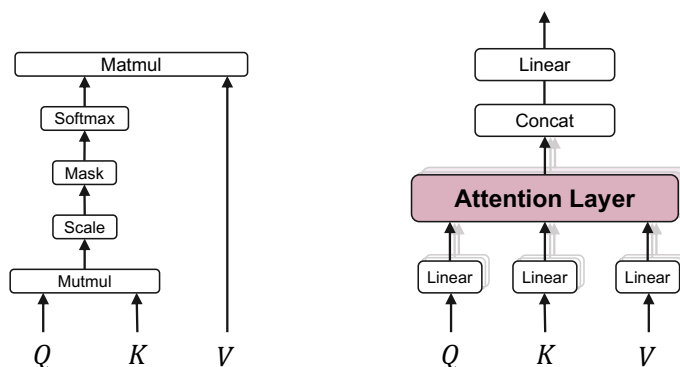


図 2.2: Transformer の Multi-head 機構

**注意機構** self-attention および source-target attention の  $\text{Attention}(\cdot)$  は以下のよう求める。

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2.1)$$

ただし,  $Q, K, V$  はモデルの内部表現である行列を表し,  $d_k$  はそれらの次元数を表す.  $Q$  と  $K$  の内積は,  $Q$  と  $K$  の各要素間の類似度を表し,  $\text{softmax}$  関数によって確率値にすることで  $Q$  と  $K$  の重みを算出する. 注意機構は, この注意の重みを用い,  $V$  との重み付き加重和をとる操作であり,  $Q$  と  $K$  の間の単語の関連の強さを反映した表現を獲得できる. self-attention と source-target attention はこの注意機構の枠組みを異なる  $Q, K, V$  の設定で用いる. まず self-attention では,  $Q, K, V$  は同一の入力源を用いることで同一文内の単語間の関連の強さを計算する. 一方 source-target attention では,  $K$  と  $V$  はエンコーダの隠れ層,  $Q$  はデコーダの隠れ層を入力源とし, 入力系列と目的系列の単語間の関連を計算する. また, 注意機構は図 2.2 に示したように式 2.1 で計算する注意機構を一つのヘッドとみなし, 複数ヘッドを並列化した Multi-head attention を使用する.

この Multi-head attention は以下のように計算する.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \quad (2.2)$$

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \quad (2.3)$$

なお,  $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$ ,  $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$ ,  $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ ,  $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$  はそれぞれ  $d_{\text{model}}$  次元の Q, K, V を線形変換するための重み行列を表す. また,  $d_k$  は W, K の次元数,  $d_v$  は V の次元数,  $\text{Concat}(\cdot)$  は行列の連結を表す.

このようにして計算する  $h$  個のヘッ드의 Multi-head attention により, 異なる位置の異なる部分表現空間を同時に処理することができる.

**フィードフォワードネットワーク** 各レイヤには, 注意機構に加えてフィードフォワードネットワーク (Feed Forward Network; FFN) が含まれる. この FFN は, 以下のようにふたつの線形変換および活性化関数 ReLU からなる.

$$\text{FFN}(x) = \max(0, x\mathbf{W}_1 + b_1)\mathbf{W}_2 + b_2 \quad (2.4)$$

ここで,  $\mathbf{W}_1, b_1, \mathbf{W}_2, b_2$  は FFN のパラメータである.

**位置エンコーディング** 今まで説明してきたアーキテクチャでは, Transformer は各トークンが実際にどの位置にいるのかを認識することはできない. そこで, その位置情報を与えるために位置エンコーディング, つまり入力系列の先頭から何番目の入力かをベクトルで表現したものをを用いる. 元論文では,  $t$  番目のトークンの位置エンコーディングは  $d$  次元のベクトルであり, その中の  $i$  番目の要素を以下のように設計している.

$$\text{PE}_{(t,2i)} = \sin\left(\frac{t}{T^{(k-1)/d}}\right) \quad (2.5)$$

$$\text{PE}_{(t,2i+1)} = \cos\left(\frac{t}{T^{(k-1)/d}}\right) \quad (2.6)$$

なお, この位置エンコーディングは, 位置表現そのものを学習する位置埋め込みに代替されることもある.

## 2.2.2 BERT

3章では, Transformer をベースとしている BERT (Devlin et al. 2019) と呼ばれる事前学習済みモデルを用いているため, 併せて本節で紹介する. BERT は近年主流になりつつある大規模なコーパスにおける事前学習により汎用的な言語知識を

獲得し、個別タスクに応用する（ファインチューニングする）モデルのひとつで、様々なタスクにおいて当時の最先端の性能を更新した。特に画期的だったのは、文を生成する際に左側の文脈しか考慮できなかった従来のモデル (Peters et al. 2018) とは異なり、マスク言語モデリングタスクを解かせることで左側の文脈と右側の文脈両方を考慮できるようになった点であった。事前学習とファインチューニングについて簡単に以下に述べる。

**事前学習** BERTは、マスク言語モデル (Masked LM) と隣接文予測 (NSP; Next Sentence Prediction) の二種類の事前学習タスクを行うことで汎用的な言語知識を獲得する。

#### **マスク言語モデル (Masked LM)**

ランダムに入力トークン系列の一部のトークンを [MASK] トークンに置換した上で、元のトークンを予測するタスクである。例えば以下の形態素解析した後サブワード分割したトークン列を考える。

(1) [CLS] My dog is cute [SEP] She likes play ##ing [SEP]

[CLS] , [SEP] はスペシャルトークンの一種で、それぞれ全てのデータの先頭につくトークンと、二つの文のセパレータの役割をするトークンである。

この内一部のトークンを以下のようにマスキングする。元論文では、ランダムにトークンの中の 15% を [MASK] トークンに置換している。

(2) [CLS] My [MASK] is cute [SEP] She likes play ##ing [SEP]

そして、マスクされた単語 dog を正しく予測できるように学習する。このように、単語予測をマスク言語モデリングタスクとして解くことで該当箇所の前後の文脈を考慮することができ、従来の事前学習モデルではできなかった双方向の特徴量を獲得した。この事前学習によって、BERT のエンコーダは文内におけるトークン間の関係を学習する。

#### **隣接文予測 (NSP; Next Sentence Prediction)**

言語モデルだけでは捉えられない二つの文の関係性を理解する能力も、質問応答や自然言語推論のようなダウンストリームタスクでは重要になる。そこで、先ほどのマスク言語モデリングタスクに加えて、与えられた二つの文が隣接するか否かを二値分類するタスクを扱う。

具体的には、入力事例の 50% は正例としてデータ内の二つの隣接する文を、残りの 50% は負例としてある文とランダムにサンプリングした文を結合して入

力する。

**正例** [CLS] My [MASK] is cute [SEP] She likes play ##ing [SEP]

**負例** [CLS] My [MASK] is cute [SEP] It's such beautiful weather today. [SEP]

この時に、それら二つの文が実際に隣接していれば IsNext ラベルを、隣接していなければ NotNext を振るように学習させる。実際にこのタスクは質問応答や自然言語推論で有効であったことが示されている。この事前学習を通して、BERT のエンコーダは文対の関係を学習する

**ファインチューニング** 先述の通り、BERT のような事前学習モデルの特徴は事前学習した言語知識を個別のタスクに応用することで高い精度を達成できる点である。Transformer のに各タスクに応じた最終層を加えることによって質問応答や文書分類など様々なタスクを扱えるようになる。

このファインチューニングによるアプローチにより、BERT は質問応答データセット SQuAD (Rajpurkar et al. 2016) をはじめとする 11 の自然言語処理タスクにおいて最高精度を示した。

## 第 3 章

# ハードテンプレートに基づく文生成

本章では、ハードテンプレートを用いて文構造を制御する手法を紹介する。

ハードテンプレートとはスロットをもつ雛形文のことで、例えば `[name] is located in [place].` のような文を指す。このテンプレートにおいて、四角で囲まれた `[name]` と `[place]` をスロットと呼ぶ。ハードテンプレートはそのスロットに単語穴埋めをすることで文を生成できる。例えば ‘Narita Airport | location | Chiba’ という関係性を文で記述するとき、そのテンプレートの `[name]` に Narita Airport, `[place]` に Chiba に当てはめて ‘Narita Airport is located in Chiba.’ という文を生成できる。あるいは `The location of [name] is [place].` というテンプレートであれば ‘The location of Narita Airport is Chiba.’ という文が生成できる。このように、ハードテンプレートは生成時の使い方が明示されており、強い構造制約として与えられるため生成文を所望の文構造で確実に制御可能となる。

近年のデファクトスタンダードであるニューラル文生成モデルは、流暢な文を生成できる一方で制御が困難である。例えば入力情報や事実に基づかない文を生成してしまう幻想 (hallucination) の問題が代表的である (Komeili et al. 2022, Wiseman et al. 2017)。これに対し、ハードテンプレートに基づく文生成手法は高い制御性をもつことから、特に内容の忠実性を要する Data-to-text タスク (Luo et al. 2020, Dou et al. 2018, Du et al. 2020) や自動要約 (Oya et al. 2014) などで用いられることが多い。ハードテンプレートは、モデルに対して頑健で信頼性が高く、文法的に正しい文が生成でき、さらに予期せぬエラーを生みづらく生成過程を完全に制御できるなど大変多くの利点がある (Manishina

2016). さらには、スロットへの穴埋めはゼロから文を生成するよりも比較的学習しやすいことから、学習データの少ないドメインにも適用可能である。そのため、創造的な文章の生成 (Özbal et al. 2013) や広告キャッチコピー生成 (山根・萩原 2014, 幅野・浦谷 2014, Tomasic et al. 2014, Hedaytullah et al. 2018), ジョーク生成 (Hempelmann et al. 2006), なぞなぞ生成 (Hong 2008) などでも活用されている。

本研究では、ハードテンプレートを用いて生成する対象として、対句と呼ばれる修辞技法を題材とする。対句とは、「人生は、[近くで見ると悲劇]だが、[遠くから見れば喜劇]である。」の下線が引かれた句ペアのことで、類似した文構造と対照的な意味を持つ文や句を並列させる技法である。あらゆる文生成タスクの中でも対句生成には特に強い構造的制約を要する。さらには構造的な性質のみならず、意味的な性質も併せ持つため、生成対象として挑戦的である。そのため、本研究ではこの対句の生成にハードテンプレートを導入し、確実にそれらの言語的性質を満たせる生成手法を提案する。

管見の限り、対句の生成に取り組んだ既存研究はない。そこで本研究では、対句データセットの作成、ハードテンプレートの獲得、そして最後にスロットへの穴埋めを行う。

まず対句データセットの作成 (3.1節) では、一般的なドメインよりも対句の使用頻度が比較的高い広告キャッチコピーを用いる。そして、定義に曖昧性を含む対句の判定ルールを詳細に設定することで標準化されたアノテーションルールを作成し、クラウドソーシングを用いてデータを作成する。

次にハードテンプレートの獲得 (3.2節) では、対句を含む文から対句に該当する部分を同定し、そこをスロット化することでテンプレートを作成する。また、スロットに埋めるべき意味的対称性を持つ語彙知識も抽出する。対句範囲の同定タスクは、構造解析のひとつとして捉えることができるが、対句構造はその位置を表す手がかり表現がないため、並列キーを手がかりとする並列構造解析手法は計算量の問題により適用できない。そのため、本研究では対句構造の特徴量を考慮した段階的な構造解析手法により、対句候補を絞り込みながら効率的に対句範囲を同定する。そうすることで、計算量を  $O(n^2)$  に抑えつつ、比較手法である関係抽出ベースのモデルに予測精度を上回ることを実験により示す。また、解析結果より対句テンプレートや既存の言語資源からは獲得できない多様な語彙知識を得られた。

最後にスロットへの穴埋め(3.3節)では、対句テンプレートのスロット位置に対して意味的対照性を持つ単語ペアを予測する手法を提案する。ここでは、データ不足の問題を解消するべく大規模なコーパスで事前学習された汎用的な言語知識をもつモデルBERT (Devlin et al. 2019)を対句穴埋めタスクに適応させるアプローチをとる。定量的・定性的な評価定性的実験により、提案した手法は対句として適切な単語を穴埋めできることを示す。

### 3.1 コーパス構築

本節では、対句生成のためのデータセットの構築を行う。対句は、一般的なドメインでは使用率が低く、データを収集することは困難である。そこで、今回は事前調査より対句の使用率が比較的高いことがわかった広告キャッチコピーをベースとして対句データセットを構築する。商品の良さを伝える広告キャッチコピーは、日本語としての文章の正確さだけでなく、メッセージの伝え方を工夫する必要がある。そこで、伝えたいメッセージを強調するために、ものごとを対比的に表現できる対句が使われる。

本研究では、まず、対句データセットのベースとして対句の使用率が比較的高い広告キャッチコピーデータを収集する(3.1.1)。次に、そのキャッチコピーに対して対句範囲のアノテーションを付与することで対句データセットを作成する(3.1.2)。

#### 3.1.1 キャッチコピーデータの収集

まず、対句データセットの構築に向けてキャッチコピーコーパスを作成した。キャッチコピーは、著名なコピーライターの作品集と、公募作品集の二種類の媒体を書き起こして取得した。

**著名なコピーライターによるキャッチコピーのコーパス** 著名なコピーライターによる作品集(仲畑 2018, 谷山 2007, 青田他 2007)に掲載されているキャッチコピーを書き起こしてコーパスを作成した。データ数は、72文字以上のキャッチコピーは削除し、21,296件となった。

キャッチコピー以外に取得できる情報としては、以下のものが挙げられる。

- 発表媒体
  - 新聞 (8,893 件) , 雑誌 (3,317 件) , ポスター (7,154 件) , その他 (1,932 件)
- 会社名
- 業種名
  - 化粧品・薬品・サイエンス・精密機器・食品・飲料など
- コピーライター名
  - 共作であれば全コピーライター名
- 掲載年度
  - 1963 年～2016 年のもの

**公募作品のキャッチコピーのコーパス** ニューラルモデルの学習には、著名なコピーライターによるキャッチコピーのみでは不足である。よって、追加データとして宣伝会議賞\*の一次審査の通過作品がすべて収録されている SKAT (Committee 2003 – 2018) に掲載されているキャッチコピーを取得した。宣伝会議賞とは、様々な企業が課題協賛として商品やサービスなどの課題を出し、その課題の広告アイデアをキャッチコピーで一般人が応募する公募形式のコンクールである。2019 年度には約 60 万作品ほどの応募があり、年一度の日本最大の公募広告賞となっている。

具体的には、SKAT.2～SKAT.17 の 16 冊のコピーを書き起こし、91,682 件のコピーを収集した。キャッチコピー以外に取得できる情報としては、以下のものが挙げられる。

- 発表年度
  - 1962 年～2017 年のもの
- 課題協賛企業

---

\*<https://senden.co/>

- 公的機関（学校）から食品メーカー、化学工業メーカーなど幅広い職種の企業名
- 審査通過状況
  - 1・2（・3）次審査通過作品，最終ノミネート，ファイナリスト
- 受賞状況
  - 金賞，銀賞，道床，審査員奨励賞，協賛企業賞（2004年まで）
  - グランプリ，準グランプリ，コピー部門賞，奨励賞，協賛企業賞（2007年まで）
  - グランプリ，コピーゴールド，シルバー，協賛企業賞
  - 審査員個人が与える賞（年度による）

日本の広告界を代表するクリエイティブディレクターとコピーライターによる厳正な審査を通過した作品のみが収録されているため，一般公募ながら高品質のキャッチコピーとなっている。そのため，さまざまな視点から物事を効果的に対比した表現が得られると考えられる。

このようにして，合計11万件以上のキャッチコピーコーパスを構築した。これは，今までで最大規模のコーパスである。

### 3.1.2 対句アノテーション

次に，3.1節で収集したキャッチコピーをベースに対句データセットを構築する。具体的には，キャッチコピーに対して対句に該当する範囲にアノテーションを付与する。なお，対応する単語ペアではなく対句範囲でのアノテーションとしたのは，事前調査により単語ペアにアノテーションを付与した場合アノテータ間での揺れが大きくなることがわかったためである。

本研究では，アノテーションデータの品質を向上させるため，アノテーション作業を(手順1)対句候補の特定と(手順2)対句の範囲に対するアノテーションの2つの段階に分けた。

まず、(手順1)の対句が含まれるか否かの分類では、1つのキャッチコピーに対して5人の作業者を割り当て、作業者はそのキャッチコピーに対句が含まれるかを判定する。この作業は「はい/いいえ/わからない」の三択問題として投入した。作業者をフィルタリングするため、30問に1問簡単なチェック設問を混ぜるよう設定した。このチェック問題に正解するか否かが報酬の付与に関わる。

対句の概念は一般的なものではなく、クラウドワーカーの判定には揺れが想定される。そのため、アノテーションの揺れができる限り少なくなるように、タスクの説明文に以下のような注意書きをした。

**注意：** 単語・フレーズだけ見ると対照的に見えても、キャッチコピー全体として対比になっていないものは対句ではない。

**【例1：対句ではない】** 日本も世界も、何かブレーキがいっぱいかかっているみたいな時代、だからこそ。(「日本 - 世界」だけを見ると対照的に見えるが、文全体では両者を同じ立場で並列に表現している。また、単語と単語の関係なので対句ではない。)

**【例2：対句である】** 日本で二番手でも世界で一番になればいいじゃん。(同じ「日本 - 世界」を使っているが、上記の例は対照的な表現であり、かつ句になっているので対句。)

そうして収集したデータの内、今回はデータの質を優先し、4人以上の作業者が対句構造を含むと指摘した約1万件のデータを(手順2)の範囲のアノテーションデータとして用いた。

(手順2)の対句構造の範囲へのアノテーションは、アノテーションの質が高いと判断した2人の作業者を抽出して行った。その際、対句範囲の揺れを最小限に抑えるため、以下の4つのルールを設定し、アノテーションの統一を図った。

#### **アノテーションの単位**

句、節、文を対象とする。逆に1単語は対象外である。

## アノテーションの範囲

対句と考えられる最も長い範囲とする。また、『「調べればわかる」より、「調べないでもわかる」方がいい。』のように、対応するスパンの両方にかぎ括弧や句読点が含まれる場合は、これらも含める。

### 制限 1

対応する単語ペアは各句内で同じ順番で出現しなければいけない。(例: 「安い車に対しても駐車料金は高い。」は今回は対象外とする)

### 制限 2

上記の全ての条件を満たしていたとしても、三つ以上の句の関係は今回は対象外とする。(例: [子供にとって「すべる」は遊び。][学生にとって「すべる」は挫折。][大人にとって「すべる」は評価。][愛車にとって「すべる」は事故。])

さらに、(手順 1) のアノテーションで間違えて対句でないものに対句タグがついていた場合は随時除いた。また、文字化けなどは随時修正し、判断がつかないものには「迷い」のフラグを立て、後で確認を行った。「作り笑いが嫌いで日本を飛び出すのに、外国で愛想笑いをしますか。」のように対照語の組が複数含まれるが句の構造が異なる場合に「迷い」が生じることが多かった。

最終的に、7,514 件の対句データセットを構築した。以下に対句のアノテーション例 ([ ] 内は対となる句) を示す。

- [運がないなら勇気] を、[勇気がないなら運] で。
- [彼の紹介した漫画] が、[私の好きな漫画] になっていた。
- [アタマが水を求める] 時、[カラダはイオンを求めている]。

### 3.1.3 対句データセットの統計

本節では対句データセットの簡単な統計について紹介する。トークナイザには Juman++(v2.0.0-rc2) (Tolmachev et al. 2018) を用いた。まず、異なるドメインにおけるトークンの平均的な長さを表 3.1 に示す。参考値として新聞の見出しのコーパスである JAMUL (Hitomi et al. 2019) の結果も併記した。対句を含むキャッチコピーは、通常のキャッチコピーより長く、見出しよりやや短くなる傾

表 3.1: 平均トークン長

ドメイン	平均トークン長
対句を含むキャッチコピー	14.07
一般的なキャッチコピー	10.69
新聞見出し (Hitomi et al. 2019)	14.84

表 3.2: アノテーションされたスパンの統語論的な単位

	頻度	例 (括弧内が対句に該当)
文	1,905	[野菜はきらい。] [野菜フライは好き。]
句	859	[孫の言うことを聞きたい] が、[体が言うことを聞かない]。
名詞句	2,424	[勉強しない子ども] より、[遊んでない大人] のほうが心配だ。
動詞句	1,817	あなたを [美しくはできない] が、[キレイにはできる]。
形容詞句	290	[スポーツが苦手な] 人も、[スポーティーは好きだ]。
後置詞句	29	牛乳パワーを、[朝は口から]、[夜は肌から]。

向が見られた。対句の範囲の平均トークン長は 4.98，スパンペア間の平均長の差は 0.78 である。また，半数以上の対句には語長差がなく，80%以上の対句は 1 語未満の語長差である。

また，表 3.2 にアノテーションされたスパンの統語論的な単位の統計と例を示す。文，節，動詞句，名詞句，形容詞句，後置詞句を対象とした。この中でも特に動詞句は比較効果を強めるための多様な活用が見られた。以下に例を示す。

**肯定-否定** [日本は明るくできない] けど、[冷蔵庫の中は明るくできる]。

**能動-受動** [昼は会社に評価され]、[夜は会社を評価する]。

**自動-他動** [部屋の一面変える] と、[部屋一面が変わる]。

**時制** [世界で活躍する日本人のためだった] しょうゆが、[世界中で活躍する人のためになっている]。

先述の通り，対句は意味的な制約として意味的な対照性を持つ単語ペアを含む。この単語ペアには，(いる，いない) のような 2 対相反の単語ペアだけではなく，(母，娘) のようにある観点で対比的な意味をもつ対照語を含むため，多様な語彙知識を要することが想定される。そこで，対句データセットの語彙を分析するため，既存の対義語辞書である WLSP-antonym<sup>†</sup> と比較した。この辞書は，加

<sup>†</sup>National Institute for Japanese Language and Linguistics (2019) WLSP-antonym (ver. 1.0)

藤らが構築した大規模コーパス WLSP から抽出された 12,000 組の単語とフレーズを含んでいる (Kato et al. 2018). 彼らは 20 人のアノテータを採用し、与えられたペアが対義語であるかどうかを判定した. なお, この辞書には (援助, 応援) のように完全な対義語でないペアが含まれている. しかし, 対句には対義語だけでなく対照語も含まれるため, この辞書の語彙知識は有用であると考えられる.

今回は, 対句内に現れる単語のうち, 対義語辞書にも載っている単語の数を数えた. なお, 今回は対句内の単語のアラインメントデータはないため, 対句範囲に含まれる各単語の頻度を数えた. 手順としては, まず各句で重複する単語を削除し (例: {彼にとっては小さな一歩, 人類にとっては偉大な飛躍}ペアから「に」, 「とって」, 「は」を削除), 残りの単語を原形に変換した (例: 「小さな」を「小さい」に変換). そして, WLSP コーパスの対義語エントリと完全一致した数を調査した.

その結果, 対句に含まれる単語の 70.9% が WLSP-antonym 辞書に存在しないことがわかった. これは, 対句に含まれる単語ペアの少なくとも 70% が辞書に含まれないことを意味する. また, する (301), ある (290), わたし (182), 人 (177), あなた (128) という最も頻度が高い 5 つの名詞または動詞 (括弧内は頻度を表す) は一般的な名・動詞であり, 辞書のエントリと一致した単語全体の 11.4% を占めていた. つまり, 今回作成した対句データセット内には対照的な意味を持つ多様な単語対が存在するが, 既存の辞書ではそれらを抽出するには不十分であることがわかる.

以上のように, 対句に関する言語知識は, 文法的側面と意味的側面で非常に多様である. これらの言語知識は既存の辞書ではほとんど獲得できない. そのため, 対句文生成には対句構造解析によるテンプレートや語彙の抽出が必須である.

### 3.2 対句構造解析によるテンプレート抽出

前章では, 対句を含む文の生成に向け, 広告キャッチコピーをベースとした対句データセットを構築した. 分析より, 対句に関する言語知識は, 文構造的側面と意味的側面で非常に多様であり, 既存の知識では対応できないことがわかった. 対句を含む文を生成するには, その文構造を担う対句テンプレートと, スロットに当てはめる対照的な意味を持つ単語ペアといった 2 種類の言語知識を抽出する

例文：ひとに都合のいい空気は、地球に都合の悪い空気でした。

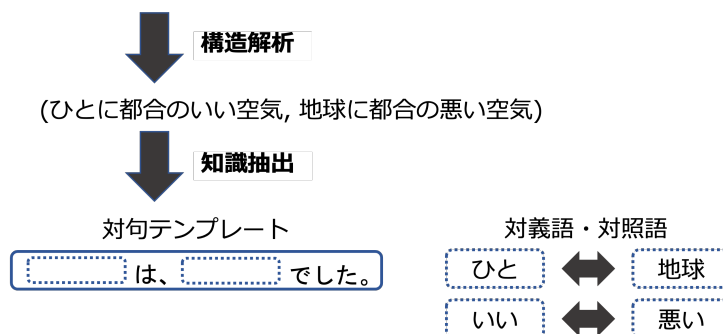


図 3.1: 対句構造解析による言語知識抽出の流れ

ことが必要である。それに向け、本研究では図 3.1に示したように文中の対句範囲を同定する構造解析によって対句文生成に必要なそれらの言語知識を抽出するアプローチをとる。

本節では、まず対句の言語学的性質について説明した後、ハードテンプレートの獲得や対句構造解析に関する関連研究を紹介する。その後、対句の構造解析を可能にする新しい解析手法を提案し、その有効性を実験により示す。

### 3.2.1 対句構造の言語学的性質

対句とは、先述の通り類似した文構造と対照的な意味を持つ文や句を並列させた表現である。

- (a) 人生は、近くで見ると悲劇だが、遠くから見れば喜劇である。。  
この例に見られるように、対句には以下の三つの言語学的性質がある。

#### 文構造の類似性

同一の対句構造に属する対句は、句の構文構造に類似性を持つ。例えば、句の依存関係を示した図 3.2のように「近くで見ると悲劇」と「遠くから見れば喜劇」は構文的に類似している。

#### 句の可換性

対句構造を成す二つの句は、お互いに入れ替えても文の流暢性が保たれる。上記の例では、「人生は、遠くから見れば喜劇だが、近くで見ると悲劇であ

人生は、[近くで見ると悲劇]だが、[遠くから見れば喜劇]である。

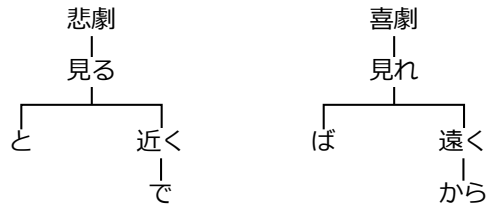


図 3.2: 句構造の類似性例

る。」のように対句を入れ替えても、文法的に成立する。

### 句の意味的対照性

同一の対句構造に属する対句は、それぞれに対応する意味的対照性を持つ語がある。上記の例だと、「近くで - 遠くから」「悲劇 - 喜劇」という意味的に対照な語が句に用いられている。

表 3.3: 対義語と対照語の例

対義語	(男, 女)	(暗い, 明るい)	(良い, 悪い)
対照語	(春, 秋)	(家族, 世間)	(ガラパゴス, 先進国)

表 3.3に示したように、意味的対照性を持つ単語には、明らかに意味が対を成す対義語と、ある観点から見ると意味的対照性を持つ対照語の二種類が見られる。

### 3.2.2 対句構造と並列構造

従来に対句構造解析を行なった研究はない。しかし、句構造の類似性・可換性という言語学的特徴を踏まえると、対句構造は並列構造の特殊なケースであると考えることができる。並列構造とは、「うちの息子は[厳父]と[岳父]の違いも知らない、愚息だ。」のように句を接続させる働きのある等位接続詞などの語（以降並列キーと呼ぶ）に伴い、句や文が並列して出現する構造のことである。並列キーとしては、英語では「and」のような等位接続詞や「as well as」のような副単語表現、日本語では「と」などの等位接続詞が代表的なものとして挙げられる。この並列構造の解析では、並列句の位置の手がかりとなる語（等位接続詞；

並列キー)を検出し、その周辺の表現から並列句の範囲を同定する手法が主流である。ところが、対句構造では、対句を示す手がかり表現が常にあるとは限らない。そのため手がかり表現なしに対句の範囲を同定するモデルの構築が課題である。この場合、並列キーによる範囲の絞り込みができないため、解析時の計算量が大きくなることに注意が必要である。

### 3.2.3 ハードテンプレートの獲得に関する関連研究

テンプレートはニューラル化以前から用いられてきた伝統的な手法である (Kukich 1983) が、テンプレートを完全に人手で作成するとコストがかかる。これによって用意できるテンプレートの数が限られるうえ、ドメインごとに作り直しを要する。そのため、自動抽出や自動学習によってテンプレートを得る研究が主流である。本節ではそれらの関連研究を紹介する。

テンプレートは、主に既存の文のエンティティ (例：スポーツ記事であればチーム名、略歴であれば人物名や出生地、広告キャッチコピーであれば商品名) をスロットに置き換えることで得られる。Lee et al. (2017) は、最初にレポート文を大量の文を人手でオウンゴールやペナルティ、イエローカードなどの内容にカテゴリ化した後、エンティティをスロットに置き換えることでカテゴリごとのテンプレートを得る手法を提案した。この研究で作成されたテンプレートは、その後翻訳サービスを介して別言語に変換されて用いられている (Gatti et al. 2018) 隠れセミマルコフモデルに基づき文中の各トークンに意味カテゴリ、例えば Player Name, Player Points, Minuteなどを割り当てる技術も、文のテンプレート化に応用可能である (Qin et al. 2018)。Özbal et al. (2013) は、収集した全てのデータを依存構造木に変換し、内容語を依存構造情報を保持したままスロット化することでテンプレート化した。そして、ユーザーが指定した単語セットと合致するテンプレートを用いた。また、予め定められた単語リスト (Sitompul et al. 2020) や単語アラインメント (Cicekli and Güvenir 2001) に基づき、文中の個々のトークンを変数化するべきか否かを自動判定する手法も主流である。ハードテンプレート特有の出力文の多様性に欠ける問題を緩和すべく、テンプレート作成時に構文木情報を付与することで、*<goal scorer> scored a goal.* から *<goal scorer> scored his second goal* のように文法性を保ったまま単語を入れ替え可能とする構文テンプレートを提案した研究もある (Theune et al.

2001b).

ハードテンプレートの制御性の高さを表す研究に Hong らが取り組んだなぞなぞ生成がある (Hong 2008, Ong et al. 2008). なぞなぞ生成は発音やスペル, 意味的な類似性や違いまで考慮する必要がある高度なタスクであり, 単語の音韻情報や品詞タグ, 同義語や単語間の意味関係など多くの複雑な制約を併せ持つ. そのため, それらの制約情報を保持したテンプレート化が必要となる. 彼らのテンプレート作成手順を紹介する. 例えば, `What kind of boy burns? A son-burn.` という既存のなぞなぞをテンプレート化することを考える. これは, `boy` (男の子) と `son` (息子) の同義関係, そして `son` (息子) と `sun` (太陽) の音韻, `burn` (燃える) と `sun-burn` (日焼け) の関係性, そして `sun` と `burn` の複合性全てを理解している必要があるなぞなぞである. まず, スロット化される単語は, 似た発音を持つ単語 (例: `waist` と `waste`) や複合語をなす単語 (例: `sub` と `burn`) などに限定する. このとき, スロット化される単語間の関係性も保持しておく必要がある. ここでは, 同義関係, `is-a-word` 関係 (二つの単語が一つの複合語をなす), 発音類似関係, 意味関係の 4 つを対象としている. これらのルールを踏まえると, `What kind of boy burns? A son-burn.` はまず `What kind of <X3> <X4>? A <Y1>-<Y2>.` にテンプレート化される. ここで, `Xn` は質問側のスロット, `Yn` は回答側のスロットである. この時, 単語の関係性は以下の四つの制約に書き下せる. ここで, `Yn-0` は `Yn` と似た発音を持つ単語, つまり `Y1` が `son` の時 `Y1-0` は `sun` である.

**X3 ConceptuallyRelatedTo Y1:** `X3 (boy)` は `Y1 (son)` と意味的に関連している

**X4 ConceptuallyRelatedTo Y1-0:** `X4 (burn)` は `Y1-0 (sun)` と意味的に関連している

**Y1-0 CapableOf Y2:** `Y1-0 (sun)` は `Y2 (burn)` できる

**Y1-0 IsAWord Y2:** `Y1-0 (sun)` は `burn` と一つの複合語をなす

こうすることで, なぞなぞの複雑な性質をスロット間の関係性を厳密に定義することで再現可能とする. これほどに複雑な制約を持つ文の自動生成には, ハードテンプレートが必須である. 他にも, 単語間の意味的關係性に強い制約を持

つ比喩の生成に取り組んだHadaytullah et al. (2018) もテンプレートに依存関係情報を持たせるアプローチをとった。

上記のどちらにも属さない最新のアプローチとして、大量のデータからテンプレートを学習したWiseman et al. (2018) の研究が挙げられる。これはニューラル隠れセミマルコフモデルにより、学習データに見られる頻出する単語の遷移パターンをテンプレート化するというアイデアに基づく手法である。遷移パターンを構成する各隠れ状態にはテキストセグメントが割り当てられる。例えば、The Golden Palace と呼ばれるコーヒーショップの説明文 “The Golden Palace is a coffee shop providing Indian food in the £20-25 price range.” という文をシステムに入力すると以下のように隠れ状態に割り当てられる。数値は対応する隠れ状態の ID であり、例えば以下の The Golden Palace は 55 番目の隠れ状態、providing は 3 番目の隠れ状態に割り当てられていることを指す。

[The Golden Palace]<sub>55</sub> [is a]<sub>59</sub> [coffee shop]<sub>12</sub> [providing]<sub>3</sub> [Indian]<sub>50</sub> [food]<sub>1</sub> [in the]<sub>17</sub> [£ 20-25]<sub>26</sub> [price range]<sub>16</sub> [.]<sub>2</sub>

そして、ここから店舗情報に該当する単語をスロット化すると、“(slot) is a (slot) providing (slot) food in the (slot) price range.” というテンプレートが得られる。この自動学習によるアプローチは画期的であり、ハードテンプレートの研究をさらに促進できる新しい方向性として今後の動向に注目したい。

### 3.2.4 対句構造解析に関する関連研究

本研究では、対句の構造解析により文中の対句の範囲を同定し、スロット化することでテンプレートを得る。対句構造解析を行った研究は過去にないが、先述の通り対句構造は並列構造の一種である。そのため対句構造解析を行うにあたり並列構造解析の手法を参考にする。本節では、対句構造解析に関連する分野として並列構造解析の先行研究について紹介する。

**並列構造解析** 並列構造の範囲を同定する並列構造解析では、従来並列句の類似性と動的計画法に基づくチャートを用いた手法が主流であった。

Kurohashi and Nagao (1994) は、ヒューリスティックなルールに基づき並列句の類似度スコアを算出し、動的計画法を用いて並列構造の範囲を同定した。まず対象文から並列キー（等位接続詞）に相当する表現を検出する。そして、この検出した並列キーを起点として範囲推定を行う。範囲推定には、並列構造を構成

これらの	0	0	2	0	0	0	0	0	0	0	0
曖昧性を	0	0	2	5	0	2	0	0	5	0	2
解消するために	0	0	0	8	0	0	2	0	5	0	0
その	0	0	0	0	0	0	0	0	0	0	0
全ての	2	0	2	0	2	0	2	0	2	0	2
可能性を	0	6	0	5	0	2	0	6	0	5	0
並列キー→ 評価し、	0	4	0	5	0	0	0	4	0	5	0
最適と	0	2	0	2			0	2	0	2	
思われる	0	2	0				0	2	0		
解を	0	2					0	2			
導出する。	0						0				

図 3.3: 黒橋らが用いた類似度計算のための三角行列の例

する文節間の類似性を利用し、動的計画法を用いたマッチング法により、長いスパンの並列句の範囲推定を行った。黒橋らが類似度スコアを計算する際に用いた三角行列の例を図 3.3に示した。この三角行列の対角成分は文節を、 $(i, j)$  要素は  $i$  番目の文節と  $j$  番目の文節の類似度を示す。その類似度スコアの計算には、品詞や文字列の一致、シソーラスによる意味的な近さなど複数の指標を用いる。各文節間の類似度を計算した後は、図 3.3のような三角行列上で、最初に検出した並列キーを起点とする最大スコアの部分経路を探索し、その経路に対応する文節列（すべての可能性を評価し - 最適と思われる解を導出する）を並列構造として出力する。経路探索時には、各ブロックの類似度スコアを加算していただくだけでなく、文節列の区切り要素や経路の方向などによって加点・減点を行う。

黒橋らの手法は、事前定義されたルールに基づき、トークン同士の対応関係を取りながら並列構造を解析できる手法である。その後Hara et al. (2009)がその利点を生かしつつ、ヒューリスティックなルールベースの手法から人手で設計した素性の重みを自動学習できるようにニューラルネットワークベースの手法へ発展させた。

しかし、異なる種類の句が並列した場合や動詞句や文の並列においては並列句は非類似となることがある。この場合、類似性のみを考慮する手法では非類似となる並列句の範囲が捉えられない。その為、名詞並列や形容詞並列と比較して動詞並列及び文並列の解析精度が低いという問題点があった。

そこで、Ficler and Goldberg (2016)は並列句候補のスコアの計算に類似性だ

入力	<i>But</i> <sub>1</sub> it said Charles Johnston, ISI chairman <i>and</i> <sub>9</sub> president, agreed to sell is 60% stake in ISI to Memotec upon completion of the tender offer for a combination of cash, Memotec stock <i>and</i> <sub>37</sub> debentures.
並列句の範囲	<i>but</i> <sub>1</sub> NONE <i>and</i> <sub>9</sub> (8, 8) chairman ; (10,10) president <i>and</i> <sub>37</sub> (33, 33) cash ; (35,36) Memotec stock; (38, 38) debentures
出力	<i>but</i> <sub>1</sub> NONE <i>and</i> <sub>9</sub> (8, 10) chairman, president <i>and</i> <sub>37</sub> (33,38) cash, Memotec stock, debentures

図 3.4: 寺西らの提案モデルの入出力例

けではなく可換性の特徴量を取り入れ、GENIA コーパス (Ohta et al. 2002) を用いた実験で既存手法を上回る精度を示した。しかし、この手法は並列構造を含む文の検出、構文解析、出力された句が実際に並列構造を成すかの識別という3段階からなり、段階ごとの誤りの伝搬や、構文解析段階の精度が外部の構文解析器に依存するといった課題が残った。

これに対し、Teranishi et al. (2017) は外部のシソーラスや構文解析を使わず、文脈情報等の深層的な情報を活用したニューラルネットワークに基づくアプローチをとった。提案手法に最も関連の強いこの寺西らの研究について紹介する。寺西らの手法では、図 3.4 に示したように、個々の並列句 (例: [A] and [B]) の範囲ではなく、ある並列キーに対して並列構造全体 (例: [A and B]) の範囲を直接同定する。具体的には、並列キーに先行する句の終点が並列キーの直前、後続する句の始点が並列キーの直後にあるという仮定のもと、並列構造全体の始点終点の二点を決定する。これにより、計算量を  $O(n^2)$  に抑えることが可能となる。

入力系列をベクトル表現に変換するためのエンコーダには、順方向・逆方向両方の文脈情報を利用可能な双方向 LSTM を採用している。そしてそのエンコーダから得た各トークンのベクトル表現を用いて、並列キーに先行する句候補と後続する句候補のベクトル表現をそれぞれ求める。次に、それらのベクトル表現の組み合わせから算出される類似性・可換性の特徴ベクトルを用いて、各並列句候補ペアに対するスコアを得る。ここでは、入力系列に並列構造が含まれない

場合のスコア (NONE) も計算する。最後に全ての可能な並列句候補ペアの中で最もスコアが大きい候補を求め、最終出力とする。寺西らの手法は、外部の構文解析器やシソーラスに依存することなく、類似性・可換性両方の特徴量を考慮することで、類似度の低い並列句の範囲同定に対しても良い性能を示した。

しかしながら、本研究で扱う対句は解析の手がかりとなる並列キーがない。そのため従来の主流な並列構造解析手法は対句に適用できない。対句においては特に効率的に構造解析を行う仕組みが重要となる。

### 3.2.5 対句構造解析の困難さ

対句構造は3.2.1節で示した言語的性質を持つ。これは解析時の有用な特徴量となり得るが、一方でその範囲の同定は容易ではない。対句構造解析を困難にする3つの特徴について説明する。

#### 語句の省略

対句構造においては、「誰かに借りるより、私に借りるほうが安心だ。」のように対応する句内で反復する語が頻繁に見られるが、同一の対句構造に属する複数の対句において、先行する対句に出現した語句が後続の対句では省略されることがある(例文 a)。また、全てを明記する必要のないキャッチコピーに特有な特徴として、反復が見られなくても「(○○の) 数字の重み」のように語句を意図的に省略する場合がある(例文 b)。このような場合は、「あの人の」という語が「言葉の重み」に係っているのか「数字の重み」に係っているのかを判断することは難しい。

- (a) あのグローバル企業に、[買収されることはあって] も、  
[することはない] だろう。
- (b) [あの人の言葉の重み] は、[数字の重み] でした。

#### 対句の統語範疇

下に示すように、対句は名詞句や形容詞句、動詞句、文といったあらゆる統語範疇の句となり得る。特に動詞句で構成される対句は、意味的な対照性を強調するために文末が「肯定-否定」や「現在形-過去形」などで形式が異なっていたり、

また「させる-させよう」のように動詞性接尾辞を伴っていたりと多様な構造を持つ。

#### 名詞句

[なくてはならない空気]が、[あってはならない空気]に、ならないように。

#### 動詞句

[毎日飲む水にはこだわる]のに、[毎秒吸う空気にはこだわってなかった]。

[不満を爆発させる]より、[才能を爆発させよう]。

#### 形容詞句

部長は [話すとつまらない] が、 [歌うと面白い]。

#### 文

[不快な 28℃もある。][快適な 28℃もある。]

また、対句同士が異なる統語範疇に属する場合もある（例：[札幌の 30℃は快適だった]が、[大阪の 30℃は耐えられなかった]。形容詞句-動詞句）。

#### 対句の出現位置

対句構造は、明確な手がかりなしに様々な接続詞を伴ったり、主語と補語といった異なる文の構成素に属していたり特定の出現位置を持たない。以下に、その内代表的なものを示した。

#### 等位接続詞

[口に入れるもの]と、[お肌につける]もの。同じです。

#### 比較

[初めての「おいしい」]より、[二度目の「おいしい」]。

#### 接続詞（時）

[水を汲みに行く 5 時間が消えた] 時、[学校に行く 5 時間が生まれた]。

#### 接続詞（場所）

[大人が幸せでない] 国で、[子どもが幸せになれる] わげがない。

#### 接続詞（理由）

[井戸から出る] のは飽きたから、[スクリーンから出る]。

#### 接続詞省略

[先が見えないと案じなさんな。][先が見えたと嘆きなさんな。]

#### その他

[あの人の言葉の重み] は、[数字の重み] でした。

以上のように、対句構造はあらゆる統語範疇の句として様々な位置に出現し得るため、構造解析において特定の構文規則に当てはめて解析することは困難である。

### 3.2.6 構造解析手法

本研究では、対句を含むキャッチコピーの自動生成に必要な対句構造解析を行う。

先述の通り、対句構造は性質上並列構造の一種であると考えられるが、対句構造には並列キーのような明示的な手がかりがない。並列構造解析では、並列キーの前後に並列構造が存在するという強い仮定を置いた解析手法が主流だが、対句構造ではその仮定なしに全ての対句候補について考える必要がある。よって、二つのスパンの開始位置・終了位置で四点を予測しなければならず、計算量が  $O(n^4)$  になる。そのため、計算量を削減する工夫が必要となる。

本節では、まず取り組むタスクを定義した後に、提案手法のアーキテクチャについて説明する。

トークン数  $n$  の文  $x = \{x_1, x_2, \dots, x_n\}$  において、先行するスパンを  $s_1 = \{x_i, x_{i+1}, \dots, x_{j-1}, x_j\}$ 、後続するスパンを  $s_2 = \{x_k, x_{k+1}, \dots, x_{l-1}, x_l\}$ 、また提案手法で採用する BERT エンコーダの出力系列を  $\mathbf{H} = \{h_c, h_1, h_2, \dots, h_n\}$  と表記する。なお、 $h_c$  は [CLS] トークンの隠れベクトルを指す。

また、 $|u|$  はベクトル  $u$  の要素毎の絶対値、 $u \circ v$  はベクトル  $u$  と  $v$  の要素積、 $u; v$  はベクトル  $u$  と  $v$  の連結、 $\text{avg}(\cdot)$  は、引数のベクトルの平均値、 $\text{maxpool}(\cdot)$  は、引数のベクトル群に対する最大値プーリングを表す。

## タスクの定義

本研究で取り組む、対句構造解析のタスクの定義について述べる。入出力の例を表 3.4 に挙げた。与えられる文に対して、対句構造を持つスパンの開始位置と終了位置 4 点を出力する。

表 3.4: 入出力の例

入力	人生は、近く <sub>3</sub> で見ると悲劇 <sub>7</sub> だが、 遠く <sub>11</sub> から見れば喜劇 <sub>14</sub> である。
出力	[3, 7] (近くで見ると悲劇) [11, 14] (遠くから見れば喜劇)

## 対句構造解析における問題と本研究のアプローチ

先述の通り、対句ペアを直接同定する際は計算量の問題が生じる。具体的には、明確な手がかり表現（並列キー）のない対句構造において二つのスパンの開始位置・終了位置の合計 4 点を同時に予測する必要があるため、計算量が  $O(n^4)$  にまで膨れ上がってしまう。例えば 19 トークンからなる文に対してはスパン同士のトークン差を 3 以下に設定したとしてもスパンペア候補数は 1,000 件弱におよぶ。これでは実用性に欠けるため、解析時の計算量を削減する手法が求められる。

そこで提案手法では、それらの問題点を解消するため、対句構造解析を (1) 終了位置判定と (2) 対句識別 のふたつのサブタスクに分割する。(1) 終了位置判定では、単語間の関係性を直接モデル化することで対句の終了位置ペアを決定する。ここで、開始位置ではなく終了位置を判定対象としたのは、日本語文の各文節が右側に係っていくという性質があるためである。そして、終了位置判定器に基づき、予測された終了位置を持つスパン候補を開始位置を 1 トークンずつ前に遡ることで列挙する。これにより、開始位置と終了位置を別々に予測できるため、計算量を  $O(n^2)$  に削減できる。提案手法の強みは、スパンペアの単語の対

応関係を捉えられること、また効率的にスパン候補を絞り込むことで現実的な計算量で解析できることである。

## 提案手法

本研究では、終了位置判定器と対句識別器を階層的に学習する手法を提案し、効率よくスパンを絞り込みながら高精度で対句の識別を行う。図 3.5 は提案手法の概要である。提案手法は以下の 5 つの部分で構成される。

- (1) **BERT エンコーダ**: 判定する文のトークン列を BERT エンコーダに入力し、各トークンの隠れベクトルを得る。
- (2) **終了位置判定器**: 二つのスパンの終了位置ペアを判定する。
- (3) **スパン候補の列挙**: 終了位置判定器が出力した句の終了位置に対して可能なスパン候補を列挙する。
- (4) **特徴抽出**: 全てのスパン候補のペアについて特徴ベクトルを抽出する。
- (5) **対句識別器**: 対句となるスパンのペアを選択する、あるいは対句「無し」と判定する。

これらのネットワーク構造の詳細を説明する。

(1) **BERT エンコーダ** 形態素解析後のトークン列を入力し、バイトペアエンコーディング (BPE) によりさらに細かく分割する。そうすることで出現頻度の低い単語を出現頻度の高いサブワードで表す。そして BPE トークン列を事前学習された BERT に入力し、入力系列の隠れベクトル  $H$  を得る。

(2) **終了位置判定器** 終了位置らしさを表すスコアの計算には、トークンペアの関係性を直接的にモデル化するため、式 3.1 で定義した内積注意の重みを用いた。この行列  $A$  は全てのトークンペア  $(j, l)$  の関係性スコアとなっている。つまり  $A(i, j)$  は  $i$  番目のトークン  $x_i$  と  $j$  番目のトークン  $x_j$  の関係性を表す。そして

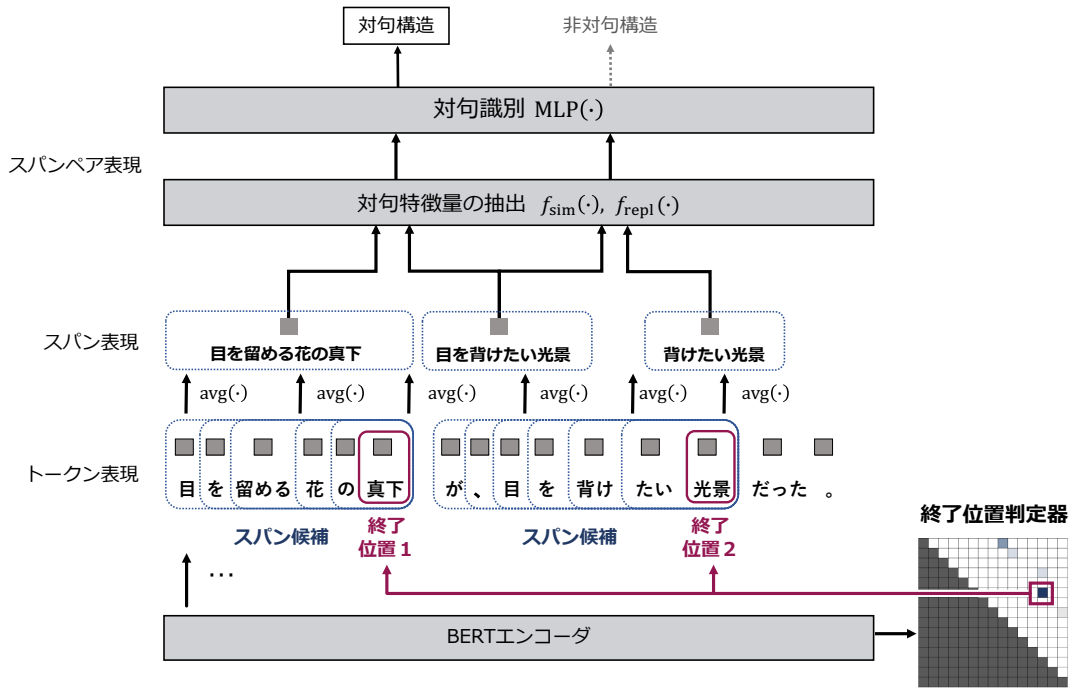


図 3.5: 提案手法の概要図

それらの内最もスコアの高い候補を softmax 関数によって求め、そのトークンペア  $\hat{y}$  を終了位置としてスパン候補の列挙に用いる。

$$\mathbf{A} = \frac{(\mathbf{H}\mathbf{Q})(\mathbf{H}\mathbf{K})^\top}{\sqrt{d_k}} \quad (3.1)$$

$$\mathbf{E} = \text{softmax}(\mathbf{A}), \quad (3.2)$$

$$\hat{j}, \hat{l} = \underset{j,l}{\text{argmax}} E_{j,l}. \quad (3.3)$$

ただし、 $\mathbf{W}^Q$  と  $\mathbf{W}^K$  は  $\mathbf{H}$  を線形変換するための学習パラメータ、 $d_k$  は  $\mathbf{W}^Q$  および  $\mathbf{W}^K$  の次元数を表す。  $d_k$  でスケールすることで、 $d_k$  の値が大きい時に勾配が小さくなることを防ぐことができる (Vaswani et al. 2017)。

(3) **スパン候補の列挙** 終了位置判定器で出力した終了位置を持つスパンを列挙する。そして、列挙したスパンに対して以下の式を適用することでスパン表現を求める。

$$\mathbf{v}_{i,j} = \text{avg}(\mathbf{h}_i, \mathbf{h}_{i+1}, \dots, \mathbf{h}_j) \quad (3.4)$$

(4) **特徴抽出** 提案手法では，対句の言語的性質に基づく特徴量として，類似構造・非類似構造の両方を捉えるため，句の類似性  $f_{\text{sim}}(\cdot)$  と句の可換性  $f_{\text{repl}}(\cdot)$  を導入する．入力されたトークン列の中で， $[i, j]$  と  $[k, l]$  の範囲の2つのスパン表現  $\mathbf{v}_{i,j}, \mathbf{v}_{k,l}$  がスパン候補として与えられたとする．

このとき，この2スパンの類似性特徴量  $f_{\text{sim}}(\mathbf{v}_{i,j}, \mathbf{v}_{k,l})$  は，Teranishi et al. (2017) を始め多くの研究でベクトルの類似度を求める際に用いられている式 3.5 を用いる．

$$f_{\text{sim}}(\mathbf{v}_{i,j}, \mathbf{v}_{k,l}) = [|\mathbf{v}_{i,j} - \mathbf{v}_{k,l}|; \mathbf{v}_{i,j} \circ \mathbf{v}_{k,l}] \quad (3.5)$$

**可換性特徴量**  $f_{\text{repl}}(\mathbf{h}_{1:N}, i, j, k, l)$  は， $[i, j]$  と  $[k, l]$  の範囲の2つのスパンに対して，各スパンの開始・終了トークンの入れ替えやすさ，前後のトークンとの接続の良さを定量化する．なお，前半のスパンが先頭トークンから始まる場合，または後半のスパンが最終トークンで終わる場合は，その部分の接続の良さは考慮しない．

$$\begin{aligned} f_{\text{repl}}(\mathbf{h}_{1:N}, i, j, k, l) \\ = [|\mathbf{h}_i - \mathbf{h}_k|; \text{avg}(\mathbf{h}_{i-1} \circ \mathbf{h}_k, \mathbf{h}_i \circ \mathbf{h}_{k-1}); \\ |\mathbf{h}_j - \mathbf{h}_l|; \text{avg}(\mathbf{h}_j \circ \mathbf{h}_{l+1}, \mathbf{h}_l \circ \mathbf{h}_{j+1})] \end{aligned} \quad (3.6)$$

(5) **対句識別器** 全ての可能なスパン候補のペアに対してスコアを予測する． $m$  個のスパン候補があった時，スパンペアの候補は  $\frac{m(m-1)}{2}$  個になる．スパンペア  $(s_1, s_2)$  のスコアは，スパンペア表現  $\text{pair\_repr}(s_1, s_2)$  と2層からなる多層パーセプトロン (MLP) を用いて計算する (式 3.7)．なお，MLP の活性化関数には

ReLUを使用した.

スパンペア表現  $\text{pair\_repr}(s_1, s_2)$  は、類似性ベクトル  $f_{\text{sim}}(\cdot)$  と可換性ベクトル  $f_{\text{repl}}(\cdot)$  を連結させたものである. また、全てのペアが対句ではない場合のスコア  $\text{score}(\text{NONE})$  は式 3.9 に示すように [CLS] トークンの隠れベクトル  $\mathbf{h}_c$  を線形変換することで定義する (式 3.9).

$$\text{score}(s_1, s_2) = \text{MLP}(\text{pair\_repr}(s_1, s_2)), \quad (3.7)$$

$$\text{pair\_repr}(s_1, s_2) = [f_{\text{sim}}(\cdot); f_{\text{repl}}(\cdot)], \quad (3.8)$$

$$\text{score}(\text{NONE}) = \text{MLP}(\mathbf{W}\mathbf{h}_c + \mathbf{b}), \quad (3.9)$$

なお、 $\mathbf{W} \in \mathbb{R}^{d_h \times d_f}$  と  $\mathbf{b} \in \mathbb{R}^{d_f}$  は線形変換に用いるパラメータである.

対句識別器の最終的な出力は、すべてのスパンペア候補の中で最も高いスコアを持つスパンペアである. これは、予め決定された終了位置 2 点  $(\hat{j}, \hat{l})$  に対して最適な開始位置 2 点  $(\hat{i}, \hat{k})$  を決定することと同義である.

$$\hat{i}, \hat{k} = \underset{(i,k) \in \mathcal{P}}{\text{argmax}} \text{score}(s_{i:\hat{j}}, s_{k:\hat{l}}). \quad (3.10)$$

ここで、 $\mathcal{P}$  は全ての開始位置候補の組み合わせ、 $s_{n:o}$  は  $[n, o]$  の範囲にあるスパンである.  $\text{score}(\text{NONE})$  が  $\text{score}(s_{i:\hat{j}}, s_{k:\hat{l}})$  よりも大きい時、あるいは終了位置判定器が終了位置候補がないと判定した時は、対句識別器も対句なしであると判断する.

**学習** 手法の学習には、終了位置判定器の交差エントロピー誤差  $\mathcal{L}^{\text{end}}$  と対句識別器の交差エントロピー誤差  $\mathcal{L}^{\text{ant}}$  の線形和を用いる.

$$\mathcal{L} = \sum_{d \in \mathcal{D}} \mathcal{L}_d^{\text{end}} + \alpha \sum_{d \in \mathcal{D}} \mathcal{L}_d^{\text{ant}}, \quad (3.11)$$

$$\mathcal{L}_d^{\text{end}} = - \sum_{(j,l) \in \mathcal{E}} \{t_{j,l}^{\text{end}} \log E_{j,l} + (1 - t_{j,l}^{\text{end}}) \log(1 - E_{j,l})\}, \quad (3.12)$$

$$\begin{aligned} \mathcal{L}_d^{\text{ant}} = & - \sum_{(i,k) \in \mathcal{P}} \{t_{(s_{i:\hat{j}}, s_{k:\hat{i}})}^{\text{ant}} \log \sigma(\text{score}(s_{i:\hat{j}}, s_{k:\hat{i}})) \\ & + (1 - t_{(s_{i:\hat{j}}, s_{k:\hat{i}})}^{\text{ant}}) \log(1 - \sigma(\text{score}(s_{i:\hat{j}}, s_{k:\hat{i}})))\} \\ & - \{t_{\text{NONE}}^{\text{ant}} \log \sigma(\text{score}(\text{NONE})) \\ & + (1 - t_{\text{NONE}}^{\text{ant}}) \log(1 - \sigma(\text{score}(\text{NONE})))\}, \end{aligned} \quad (3.13)$$

$$t_{\text{NONE}}^{\text{ant}} = \begin{cases} 1 & (\sum_{(i,k) \in \mathcal{P}} t_{(s_{i:\hat{j}}, s_{k:\hat{i}})}^{\text{ant}} = 0) \\ 0 & (\text{otherwise}) \end{cases} \quad (3.14)$$

$\mathcal{D}$ は学習事例の集合、 $\mathcal{E}$ は終了位置判定器に入力されるトークンペアの集合、 $\sigma(\cdot)$ はシグモイド関数である。また、トークンペア  $(j, l)$  の正解ラベル  $t_{j,l}^{\text{end}}$  はトークンペア  $(j, l)$  が正しい終了位置ペアだった時のみ1の値をとり、その他の場合は0をとる。同様に、スパンペア  $(s_{i:\hat{j}}, s_{k:\hat{i}})$  の正解ラベル  $t_{(s_{i:\hat{j}}, s_{k:\hat{i}})}^{\text{ant}}$  は、 $(s_{i:\hat{j}}, s_{k:\hat{i}})$  が対句をなす時のみ1の値をとり、その他の場合は0をとる。NONEの正解ラベル  $t_{\text{NONE}}^{\text{ant}}$  は、すべてのスパンペアが対句を構成しない場合にのみ1をとる。また、 $\alpha$ は終了位置判定器と対句識別器の損失の重みをコントロールするハイパーパラメータである。

BERT、終了位置判定器、対句識別器のパラメータはすべてこれらの損失関数に基づいて学習中に更新される。

### 3.2.7 実験設定

本研究では、3.1.2で構築した対句データセットを用いて対句構造解析を行う。

**データセット** 対句データセットを学習データ、開発データ、評価データに分割した。サイズはそれぞれ21,033件（内対句ありは5,256件）、4,506件（内対句ありは1,125）、4,510件（内対句ありは1,126件）である。対句を含まないデータは、3.1.2の（手順1）において3人以上のアノテータが対句を含まないとした

事例からランダムに抽出したものである。

**ベースライン手法** 対句構造解析タスクは、二つの句を組み合わせることで対句関係として解釈が可能になることから、関係抽出タスクの一種として捉えることができる。関係抽出とは、「スティーブ・ジョブズはサンフランシスコ生まれである」という文があったときに、スティーブ・ジョブズとサンフランシスコの間に出生地という関係性があることを認識するタスクである。このタスクでは固有表現と呼ばれる組織・地名・製品名などの固有名詞（例：スティーブ・ジョブズ、サンフランシスコ）が既知である。本研究では、その固有表現の認識（スパンの同定）と固有表現のペアがなす関係の認識（対句識別）のふたつのタスクを同時に解く最新の関係抽出モデル SpERT (Eberts and Ulges 2019) を高性能な比較手法として採用する。

SpERT は、入力文を BERT でエンコードし、各トークンの隠れ状態ベクトルからスパン分類（固有表現抽出）、スパンフィルタリング、関係分類を行うエンドツーエンドのモデルである。今回はその SpERT を対句構造解析に適用し、スパン分類器を対句スパン候補の列挙に用い、関係分類器は各スパンが対句の候補であるかどうかを判断する対句識別に用いる。また、対句の類似性、可換性に基づく特徴も導入する。スパン分類器は、スコアの高いものから少なくとも 2 つのスパンを出力する。学習時、スパン分類器と対句分類器の入力に用いるスパンは、事前にランダムにサンプリングされる。この段階的なアプローチにより、SpERT は提案手法と同様に、 $O(n^2)$  の計算量で対句構造解析を行える。このベースラインを以降 SpERT ベースラインと呼ぶ。

また、スパンではなくトークンに基づくベースラインとして、BERT の出力ベクトルをラベル分類器に直接入力する系列ラベリングモデルを採用した。これは、各トークンが対句の範囲内にあるか否かを系列で予測する手法である。

**パラメータ設定** 提案手法の最終的なハイパーパラメータの設定を表 3.5 に示した。SpERT ベースラインのハイパーパラメータのほとんどはオリジナルの SpERT (Eberts and Ulges 2019) のパラメータと同様のものであり、事例のサンプリング方法は対句データセットの統計に合わせて変更したものをを用いた（表 3.6）。また、全てのモデルをエポック数 20 で学習した。

表 3.5: 提案手法のハイパーパラメータ

パラメータ	値
エポック数	10
学習率	2e-5
MLP の隠れ層のユニット数	2,400
$Q, R$ の次元 $d_k$	100
損失関数の重み $\alpha$	0.5

表 3.6: SpERT ベースラインのハイパーパラメータ

パラメータ	値
最大スパン長さ	10
スパンペアに含まれるトークンの長さの差	[0,3]
サンプリングするスパン数	100
サンプリングするスパンペア数	100

**評価指標** SpERT ベースライン手法と提案手法は、スパン分類と対句識別の 2 つのサブタスクそれぞれについて評価した。スパン分類では、各スパンが対句を構成し得るか否かを分類し、スパン候補を列挙する。提案手法は、終了位置判定によりこれを行うが、SpERT ベースライン手法は事前にサンプリングしたスパンを分類することによりこれを行う。終了位置判定に関しては、予測された終了位置ペアが正しいかどうかを精度で正解率で評価した。また、SpERT ベースライン手法では、予測されたスパンが正しいスパンと完全に一致しているかどうかを正解率で評価した。対句識別の性能評価は、トークン単位でスパンの一致度合いを評価することで行った。SpERT ベースライン手法と提案手法は二つのスパンペアが対句を構成するか否かを識別する一方、系列ラベリングに基づくベースラインでは、直接文中の対句範囲を同定する。

なお、性能の評価は 5 回の実験の平均値で行う。

**実験設定** 異なる対句の特徴量の効果を検証するために、対になるスパンの 2 つの表現をそのまま連結して対句の特徴量なしにスパンペア表現とする場合と、スパンペアから抽出した対句の特徴量ベクトルをスパンペア表現とする場合を比較する。また、SpERT ベースラインでも、スパンペア表現として対句の特徴量を用いない場合と用いる場合を比較する。対句の特徴ベクトルを用いる場合は、SIM (類似性特徴量のみ)、REPL (可換性特徴量のみ)、BOTH (類似性・可換

表 3.7: 対句識別タスクの結果. P は精度, R は再現率, F は F1 スコアを表す.

	スパン候補の絞り込み			対句識別		
	P	R	F	P	R	F
系列ラベリング	-	-	-	0.725	0.761	0.742
SpERT ベースライン	0.792	0.755	0.770	0.590	0.803	0.680
SpERT ベースライン-SIM	0.793	0.757	0.774	0.554	0.806	0.655
SpERT ベースライン-REPL	0.783	0.759	0.771	0.577	<b>0.807</b>	0.672
SpERT ベースライン-BOTH	0.787	0.758	0.772	0.559	<b>0.807</b>	0.659
	終了位置判定					
提案手法	0.884 (0.948)			0.773	0.767	0.770
提案手法-SIM	0.886 (0.950)			0.773	0.780	<b>0.776</b>
提案手法-REPL	0.874 (0.939)			0.781	0.757	0.769
提案手法-BOTH	0.879 (0.943)			<b>0.788</b>	0.757	0.772

表 3.8: 1 エポックあたりの学習に要する平均時間. 単位は秒.

特徴量	SpERT ベースライン	提案手法
SIM	3978	524
REPL	12079	731
BOTH	15186	727

性特徴量の両方) の 3 種類で表記する.

文エンコーダには, 日本語 Wikipedia で学習された Transformer の層数 12, 隠れ層の次元 768, self-attention ヘッドの数 12 からなるパラメータ数 1.1 億の BERT<sub>BASE</sub> の事前学習モデル (柴田他 2019) を用いた. また, トークナイザには Juman++ (v2.0.0-rc2) (Tolmachev et al. 2018) を採用した.

### 3.2.8 実験結果

表 3.7 に, スパン分類と対句識別の性能を示した.

単語間の対応関係を考慮する提案手法は SpERT ベースラインと比較して F1 スコアを 9 ポイント以上向上させ, 明らかな優位性を示した. また, 終了位置判定の平均 F スコアは, 両終了位置が完全一致する場合は 0.874 から 0.886, 少なくとも片方の終了位置が一致する場合は 0.939 から 0.950 に向上した. これは, 終了位置判定がほとんどの終了位置を正しく検出できるため, 各手法が大きな誤差の伝播なしに単語間の対応を捉えることができることを示している.

一方、対句の3つの特徴量による性能に顕著な違いは見られなかった。また、対句の特徴を持たない提案手法の性能は、可換性のみのを考慮した手法より性能が高い。さらに、スパンペアを考慮しない系列ラベリングの性能は、対句の特徴量を考慮する SpERT ベースラインよりも良い性能を示した。これらのことから、BERT はファインチューニングの際に暗黙的に対句の特徴量を自動学習する能力が高いため、LSTM ベースの既存研究 (Teranishi et al. 2017) とは異なり対句の特徴量があまり役に立たないことが示唆される。また、self-attention により離れた単語間の関係を直接捉えることができることも、対句の特徴量なしに BERT が高い性能を達成できる理由と考えられる。

以上より、終了位置と開始位置を段階的に決定する提案手法は、計算量を効果的に削減し、単語間の対応関係に基づく高精度な識別を行えることを示した。一方で、予想外にも対句の特徴量は性能に大きく寄与しなかった。これは、BERT が元々構造的特徴や意味的特徴を捉える能力を有していたためと考えられる。また、提案手法の SpERT ベースラインに対する優位性は、関係抽出モデルを対句識別に適用することが困難であることを示すものである。関係抽出では、人名や地名など、認識された固有表現 (スパン) に対して意味カテゴリが定義される一方で、対句は各スパンの意味カテゴリが明確ではなく、2つのスパンをペアにすることで初めて対句であると解釈される。そのため、スパンペアを考慮せずに直接スパン候補を列挙することは非効率的である。

### 3.2.9 事例分析

提案手法の具体的な出力例を図 3.6 に示した。(a) のように肯定-否定といったバリエーションを持つ動詞句ペアや (b) のような文ペアも正しく認識できている。また、(c) のような対応する名詞句が括弧を伴う場合でも、括弧まで正しく含めることができる。

一方で、図 3.7 に示したように提案手法が不正解する傾向のある事例もある。例 1, 2 から分かる通り、正解スパンとの関係性が強いトークンは一緒にスパンとして出力されることがある。これらの例は、今回考慮している特徴量である類似性や可換性を同時に満たしているためである。このような場合は、類似性・可換性だけでは対処できないため、例 2 であれば「神話 - 事実」「は-が」「いない-いる」といったように、トークン単位での対応関係をモデルに組み込み、

- (a)[コップを洗って] も、[水道管は洗わない]。
- (b)[一人のときは、いつものカード。][二人のときは、ゴールドカード。]
- (c)…[「食べてみたいもの」]から[「つくってみたいもの」]になると思う。

図 3.6: 提案手法による出力例

- 正解1 [「がんばれ」も嬉しい]ですが、[「おいしい」はもっと嬉しい]  
 予測1 [「がんばれ」も嬉しいです]が、[「おいしい」はもっと嬉しい]
- 正解2 安全に、[神話はいらぬ]。[事実がある]。  
 予測2 [安全に、神話はいらぬ]。[事実がある]。
- 正解3 赤ちゃんできた。[ひとりじゃない、うれしさ]と[ひとりじゃなくなる不安]。  
 予測3 [赤ちゃんできた。ひとりじゃない]、[うれしさとひとりじゃなくなる]不安。
- 正解4 [大嫌いだった父]を撮った。レンズ越しの父は、[私の好きな父]だった。  
 予測4 大嫌いだった父を撮った。[レンズ越しの父]は、[私の好きな父]だった。

図 3.7: 提案手法による解析の失敗傾向 ([ ] 内が予測/正解スパン)

対応先がない「安全に、」をスパンから取り除く機構が必要である。例3のように、句読点の位置に混乱してしまうものもあった。また、例4では「～父」という名詞句が三箇所出てくるが、「大嫌い-好き」という対照語を理解していれば正解できるところを誤答してしまった。提案モデルでは、対照語を明示的に扱っていないため、このような誤答パターンが見られた。よって、外部知識として対照語を加える必要がある。

### 3.2.10 解析結果からの知識抽出

本解析の目的は、構造解析の結果から対句テンプレート及び対義語・対照語の語彙知識を抽出することである。本節では、抽出できた言語知識の分析結果を報告する。

#### 対句テンプレートの抽出

対句テンプレートについては、同定された範囲をスロット化することで得られる。ここでは、そうして得られたテンプレートからランダムにサンプリングした161件の事例から対句の文構造を分析した。この分析では、先行する句を [span1],

表 3.9: 対句の構造パターンと出現頻度

パターン	頻度	例
(A)	17	[この街が好き．][この部屋は嫌い．]
(B)	64	[作りたてではない]けれど，[出来立てではある]．
(C)	80	いい人生は，[クチコミに始まり]，[書き込みに終わる]．

後続する句を [span2] に置き換え，対句の構造パターンを抽出している．その結果，対句の文構造は以下の3つのパターンのいずれかに従うことが分かった．

(A) 対句ペア 例： [span1][span2]

(B) 対句ペア+機能語 例： [span1] {だが, か, ...} [span2],

(C) 対句ペア+内容語 例： 大切なのは， ... [span1] ... [span2] ...

パターン B では，機能語のみが対句の2つのスパンをつないでいる一方で，パターン C では，1つ以上の内容語が対句に文脈情報を与えている．表 3.9 にそれぞれの構造パターンの出現頻度と例を示した．

これによると，パターン A とパターン B は全体の半分近くを占めている．対句生成をすることを考えると，パターン A と B は対句スパン以外の内容語を生成する必要がないため，生成テンプレートに変換しやすい．一方，パターン C は，対句の内容と意味的に関連するような内容語（上記の例では人生）を生成する必要があるため，テンプレート化が困難な場合がある．パターン C に適した内容語をどのように選択するかは，今後の興味深い方向性であろう．

追加分析として，パターン B におけるスパンをつなぐ接続詞や接続助詞の種類を調べた．その結果，以下の8つの構造パターンが確認された．

- (1) 連結： [日本の自然] と [日本人の夢]。
- (2) 選択： [ごめんなさいを言いたい日] か， [ありがとうと言いたい日]。
- (3) 逆説： [髪型を変えても気づかれない] のに， [シャンプーを変えたら気づかれた]。
- (4) 条件： [上司が変わろう] と， [注ぐ酒は変わらない]。
- (5) 比較： [世界とつながっている] より， [私の世界とつながりたい]。

- (6) **理由と結論**： [肌年齢が 18 歳] だから、 [私は 18 歳] です。
- (7) **変化**： [「本当にキレイになるの?」] から [「本当にキレイになるの!」] へ。
- (8) **その他 (接続詞なし)**： [日本で一番新しい空港] は [日本で一番有名な空港] です。

これらの例から、対句に用いられる接続詞や接続助詞の種類は限定的であることがわかる。このことから、パターン B の (1) から (7) までの 7 つの構造パターンが対句テンプレートとして容易に活用できる。一方、接続詞を持たない (8) のパターンもパターン B の事例全体の 15% 以上を占めており、任意の文型で対句を生成することも今後の課題である。

### 言語知識の抽出

語彙知識については単語単位のアラインメントが必要であり、解析結果から直接得ることはできない。そのため、対句と同定された範囲に含まれる単語を品詞に変換し、対応すると考えられる単語ペアを抽出した。手順は以下の通りである。

1. 対句を含むと判定された文を Juman++ (Tolmachev et al. 2018) を用いてトークンに分割し、内容語 (名詞, 動詞, 形容詞, 副詞, 副名詞, 間投詞) のみを残した後原形に変換する
2. 各句に含まれる単語の先頭から、品詞が同一である単語ペアを探し、そのペアを対照語ペアとして抽出する
3. 手順 2. の過程で、両方のスパンに含まれる単語を削除する

以下が「あなたを [美しくすることはできない] が、 [綺麗にすることはできる]。」という事例から言語知識を抽出した際の具体例である。

1. [美しく/形容詞 する/接尾辞 こと/名詞 は/助詞 できる/動詞 ない/接尾辞], [綺麗に/形容詞 する/接尾辞 こと/名詞 は/助詞 できる/動詞]
2. (美しく/形容詞, 綺麗に/形容詞), (する/接尾辞, する/接尾辞), (こと/名詞, こと/名詞), (は/助詞, は/助詞), (できる/動詞, できる/動詞)
3. (美しく/形容詞, 綺麗に/形容詞)

表 3.10: 抽出できた単語ペアの関係性

関係性	頻度
対義語	9
対照語	100
単語の誤分割を含む対照語	39
抽出ミス	52

辞書にない対義語	(飽食, 飢餓)
	(未来, 過去)
対照語	(競合, 協力)
	(捨てる, 直す)
	(理解, 実証)
	(社員, アルバイト)
	(恋人, 愛人)
	(お金, カード)

図 3.8: 抽出できた単語ペアの例.

このプロセスにより, 10,502 件の単語ペアを抽出できた. このうち 96%は既存の対義語辞書 WLSP-antonym (3.1.3) には見られなかった.

次に, 各単語ペアが対義語か対照語かをランダムに抽出された 200 ペアに対して手作業で分析した (表 3.10). この結果, 提案手法により, 誤った単語分割を含む 139 組の対照的な単語ペアが抽出できた. 合計 52 組の抽出ミスは, ほとんどが構造解析の誤りによるものであった. 抽出された単語ペアの例を図 3.8 に示す. 本手法は, 対義語だけでなく, 対照語を抽出することにも成功した.

### 3.3 テンプレートの穴埋めによる対句生成

本章では, 対句テンプレートに対して単語を穴埋めすることで対句を生成する. その時, 3.2.10 節で示したように意味的な対照性を考慮して単語ペア (以降簡単のため対義語と対照語を合わせて対義語と表記) を埋める必要がある. このための最も単純なアプローチは, 対義語辞書をひいて, 記載されている単語を埋めることである. ところが, 穴埋めができる対義語はテンプレートの文脈によって変化する. 例えば, 次に示すように**母**という単一の単語に対しても, **父**,

妻, 娘など複数の単語が対義語として考えられる.

(A) 父は保険で楽をした。母は保険で苦労した。

(B) 妻になって、値段が気になった。

母になって、成分が気になった。

(C) 娘はエステに行き、母は整体に行く。

また、対義関係には二律相反のペア（素数—非素数）だけではなく、相互の役割などの関係性によるもの（生徒—教師）、ある基点を挟んだ相対的・対照的な概念を表すもの（明日—昨日）など多様な関係性が含まれる (Hoffmann 1993, Leech 1976, 三省堂編修所 2017). しかし、辞書には対義語ペアのみが収録されており、その文脈情報は収録されないことが多い上に掲載されている対義関係の種類も限定的である. これに対し、対句生成のための対句穴埋めタスクでは、文脈情報や多様な対義関係を考慮して対義語を予測できるモデルが求められる.

近年、モデルを事前学習およびファインチューニングするアプローチが多くタスクで高い性能を収めている (Devlin et al. 2019, Yang et al. 2019). そこで本研究では、代表的な事前学習モデルである BERT を利用し、対句テンプレートの内容である文脈を考慮した対句穴埋めを行う. しかし、モデルをファインチューニングするための教師データ、すなわち対照的な文脈を持つ対義語ペアを含む文の収集は容易ではない. そこで本研究では、BERT モデルで対句穴埋めを行うための手法として、(1) 対句データを用いたドメイン適応、(2) 対句穴埋めに特化させるための対照的マスキング、(3) 文の対句構造を捉えさせるための対句位置エンコーディング、(4) 対義語辞書を活用した自動アノテーションによる擬似教師データ収集 を提案する.

実験の結果、提案したファインチューニング手法により、BERT モデルを文脈を踏まえた対句穴埋めタスクへ適応させ、正解の単語をより予測できるようになることを確認した. また、文脈を考慮した対句穴埋めタスクでは複数の正解が許容されるため、唯一の正解による自動評価指標による精度は低く見えるが、対義語としての適切さや文脈における流暢さをもつ事例は 85%以上を占めることから、提案手法は対句生成モデルとして十分な性能を示していることを報告する.

### 3.3.1 ハードテンプレートへの穴埋めに関する関連研究

従来の穴埋め手法では、使いたい単語リストを予め準備しておき、テンプレートに対する適合性や、穴埋めした時の文の意味など定められた基準を満たすように単語の置換を繰り返す手法が主流であった。この単語穴埋めに際し、いくつかの単語候補がある中でより適切なものを探索するため、例えば遺伝的アルゴリズムによってより意味的・文法的に正しい単語を選択できるようにしている (Hadaytullah et al. 2018)。ハードテンプレートに基づく生成手法は、データ数が少ないドメインにも適用であることから、例えば広告キャッチコピーの生成などでも用いられている。山根・萩原 (2014) は、消費者の嗜好を考慮したキャッチコピーを自動生成するため、まず既存のキャッチコピーをユーザの嗜好が反映されていると考えられる単語カテゴリに含まれる単語で置換していくつかの候補文を生成する。さらにこの候補から、文の自然さや意味的関連性に基づくスコアが最大のキャッチコピーを出力する。Tomasic et al. (2014) は穴埋め後の文の長さや多様性、指定した単語を含むか否か、主観性の高さなど複数の観点で評価し、生成された文のスコアが予め決められた基準を越えるまで単語の置換を繰り返すことで高品質な文を生成するアプローチをとった。

近年では、ニューラルモデルがすべての語彙から適切な単語を選択して穴埋めするアプローチが主流となった。ハードテンプレートは、スロットを穴埋めする際にその左側の文脈と右側の文脈両方を考慮する必要がある。そのため、双方向言語モデルによる穴埋めが最も適している (Devlin et al. 2019, Ghazvininejad et al. 2019)。本研究でも、この双方向言語モデルを用いて対句生成を行う。なお、最近ではハードテンプレートへの穴埋めアプローチは文生成に限らない幅広いタスクで用いられている (Liu et al. 2021)。これは、タスクごとの入力  $x$  をテンプレートを用いてスロットを持つプロンプトに修正し、言語モデルを用いてスロットに該当する単語を予測させることで最終出力を得るアプローチである。例えば、‘I love this movie.’ という文章  $x$  の感情を予測する感情分析タスクに取り組むとき、感情分析用テンプレートとして ‘[ $x$ ] The movie is [slot].’ を準備する。そして入力文を ‘I love this movie. The movie is [slot].’ に書き換えてモデルに入力すると、スロットに対して ‘great, great, …’ などの肯定的な単語を返す。機械翻訳タスクでも、‘Je vous aime.’ という仏文を英文に翻訳するためのテンプレート ‘French: [ $x$ ] English: [slot].’ を用いて入力文を ‘French: Je vous aime. English:

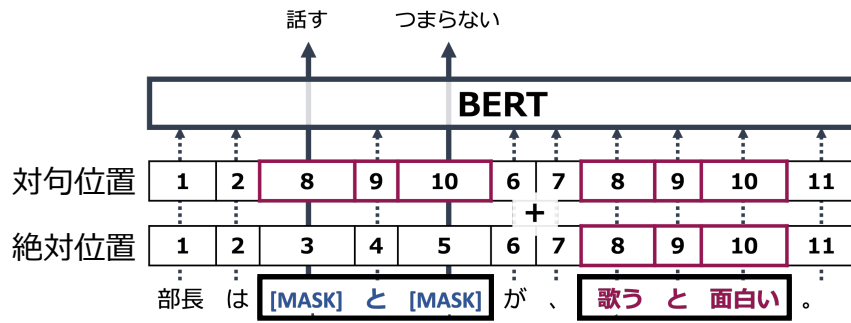


図 3.9: 提案手法の概要

‘slot’に書き換えて入力すると ‘I love you’ という出力が得られる．このように、適切なテンプレートに基づき入力文を書き換えることで、モデルの挙動を制御し、事前学習済みの言語モデルを用いて、さまざまなタスクを解くことができる．このアプローチは、タスク固有の追加学習なしに使えるケースもあり、近年注目を浴びている (Raffel et al. 2020, Petroni et al. 2019).

### 3.3.2 提案手法

概要を図 3.9に示したように、位置  $m$  ( $1 \leq m \leq n$ ) に [MASK] トークン (スロットに該当) を持つ長さ  $n$  の系列  $x_1, \dots, x_n$  を与えられた時、穴埋めするトークン  $y_m$  の条件付き確率を BERT モデル (Devlin et al. 2019) を用いて以下のようにモデル化する．

$$P(y_m | x_1, \dots, x_m, \dots, x_n). \quad (3.15)$$

入力情報の双方向の文脈で条件づけることにより、BERT モデルは [MASK] トークンの周辺の文脈を捉えられる．その BERT モデルを対句構造を持つコーパスでファインチューニングすることで、ドメイン適応により入力文中の文脈と対義語ペアの組み合わせを考慮可能となることが期待される．

BERT モデルを対句穴埋めに適応させる際、ニューラルモデルの学習に対して比較的サイズが小さい対句データセットを有効活用するために以下の三つのアプローチを取る．

**対照的マスキング** 文中に対義関係を持ち得るトークンを [MASK] トークンに置換することでファインチューニングのための教師データを作成する。例えば、「[値段に始まり]、[味わいに終わる]。」という文から、「[MASK]に[MASK]、味わいに終わる。」、「値段に始まり、[MASK]に[MASK]。」という二件の学習事例を得る。これらのマスクされるトークンには、対応する句に現れないものを選択する。このマスキングの方法を**対照的マスキング**と呼ぶ。オリジナルのBERTモデルは [MASK] トークンをランダムに作成するが、**対照的マスキング**は、対句穴埋めのための教師データを選択的に作成する。

**対句位置エンコーディング** 次に、二つ目のアプローチとして、BERTモデルに入力文中の対句構造を考慮させるために位置エンコーディングを拡張する。まず、対句構造を構成する二つのスパン  $[i, j)$  ([MASK] トークンが含まれる) と  $[k, l)$  ( $1 \leq i < j \leq k < l \leq n$ ) を含む文を考える。この時、スパン  $[i, j)$  にスパン  $[k, l)$  が対応することを教えるために、対句構造に特化したインデックスを以下のように定義する。

$$a_t = \begin{cases} k + \left\lfloor \frac{(l-k)(t-i)}{j-i} \right\rfloor & (t \in [i, j)) \\ t & (\text{otherwise}) \end{cases} \quad (3.16)$$

インデックス  $a_t$  は  $t \notin [i, j)$  だったときはトークン  $x_t$  の絶対位置を、そうでない場合は対応するスパン  $[k, l)$  の絶対位置を与える。本稿では、この手法を**対句位置エンコーディング**と呼ぶ。今回は、オリジナルのBERTモデルで用いられている絶対位置の埋め込み表現と対句位置  $a_t$  の埋め込み表現の平均を取ったものをBERTモデルに入力する。

**辞書を用いた擬似教師データの作成** 教師データには、3.1節で作成した対句データセットを用いる<sup>‡</sup>。しかし、教師データだけでは学習事例数が少なく、BERTモデルをファインチューニングするには不十分である。そこで、文を自動的にアノテートすることを考える。具体的には、既存の対義語辞書(三省堂編修所 2017)に掲載されている対義語ペアを含むキャッチコピーを用いる。このプロセスによ

<sup>‡</sup>3.1節で作成したデータからアノテーションルールに反していた4件のキャッチコピーを除いた7,511件を用いる。

り、対句データセットに含まれないキャッチコピーから 1,894 件を抽出した<sup>§</sup>。各キャッチコピーに対して、対義語を片方ずつ [MASK] トークンに置換することで二件の学習事例を作成した。この外部知識を活用したデータ作成により、対義語の語彙知識を擬似教師データとして BERT モデルに学習させる。

### 3.3.3 実験設定

**データセット** 7,511 件の対句構造を持つキャッチコピーを学習データ、検証データ、評価データに分割した上で、[MASK] トークンを含む穴埋め事例に変換する。対句構造には二つの対照的な意味を持つ句が含まれることから、各キャッチコピーから二件の学習事例を作成できる。このようにして、11,923 件の学習データ、1,496 件の検証データ、1,247 件の評価データを作成した。この評価データの作成方法は 3.3.3 に示した。また、擬似教師データとして 3,788 件の学習データを用いる。

**ベースライン** 今回の実験では、ベースライン手法として辞書を用いた手法と、ファインチューニングなしの事前学習済み BERT モデルを採用した。辞書を用いた手法では、各空欄を埋めるべき正解単語が、対応する句に含まれる単語の対義語として辞書 (三省堂編修所 2017) に含まれるかを調査する。このベースラインは、正解単語がわかっている状態で評価するため、辞書を用いた手法の性能の上限として考えられる。ファインチューニングなしの事前学習済み BERT モデルでは、対句穴埋めに特化した学習なしに、BERT モデルが言語モデルとして文脈を考慮して対義語を予測する能力について調査する。また、3 名の作業員に対して穴埋めタスクを実施した結果 (以後作業員と表記) も評価する。具体的には、各空欄に対して最大 5 つの単語を回答させた。3 名の作業員は、本タスクの難易度を考慮し、作家やブロガーなど、作文経験の豊富な作業員を選んだ。なお、事前学習済みモデルには、日本語 Wikipedia で学習された Transformer の層数 12、隠れ層の次元 768、self-attention ヘッドの数 12 からなるパラメータ数 1.1 億の BERT<sub>BASE</sub> の公開モデル<sup>¶</sup> を用いた。トークナイザには Mecab<sup>||</sup> を用いた。

<sup>§</sup>対句データセットに含まれないキャッチコピーは、コーパス構築の一段階目における厳しい基準でフィルタリングされたため、実際には対句を含むものが含まれている。

<sup>¶</sup><https://github.com/cl-tohoku/bert-japanese>

<sup>||</sup><https://taku910.github.io/mecab/>

**評価** 本実験では、マスクされた位置に正しい対義語が予測されたか否かを top-1, top-10 の正解率で評価する。評価データには、[MASK] トークンがサブワード単位で分割されている事例が含まれる。しかしながら、作業者はサブワード単位で空欄を埋めることはできない。そのため、評価データのうち [MASK] トークンがサブワード分割されていない事例を含むサブセット（以後単語単位と呼ぶ）を作成し、作業者が埋めた単語とモデルが予測した単語を比較する。また、作業者は必ずしも一つの空欄に対して5つの回答を思いつくとは限らないため、top-1 と top- $n$  の正解率でも評価する。この数値  $n$  は、各事例に対する回答数によって異なる。

**評価データの作成方法** 評価データ内で空欄とする単語（列）には、3.3.2で説明した対照的マスキングの対象となった単語（列）から、以下の基準に基づき選択した。

1. 評価事例一件につき一単語（句は除外）
2. 単語の品詞は名詞，動詞-自立，形容詞-自立，形容動詞のいずれか
3. 企業情報などの外部知識がなければ解けない事例は可能な限り除外

本来は一件のキャッチコピーから前半の句と後半の句をそれぞれマスキングすることで二件の評価事例を作成できる。しかし、クラウドソーシングプロセスの簡略化のため、単語単位ではキャッチコピー一件につきランダムに評価事例を一件選択して使用した。そのため、元々の評価データ全体に比べて小規模になっている。

### 3.3.4 実験結果

対句穴埋めの結果を表 3.11に示した。

提案手法は、全評価データに対して top-1 正解率で 23.4%，top-10 正解率で 47.9%，単語単位では top-1 正解率で 30.4%，top-10 正解率で 49.1%を達成した。ファインチューニングなしの事前学習済み BERT モデルは、提案手法に比べて top-1 正解率が大きく低下している。これは、一般的なマスク言語モデルでは、対句構造の文脈を与えられていたとしても、対義語を予測するには不十分であることを示している。言い換えれば、提案手法はモデルを対句構造における単

表 3.11: 対句穴埋めの正解率. Acc. は正解率を表す.

	全データ		単語単位	
	Acc.@1	Acc.@10	Acc.@1	Acc.@n
辞書を用いた手法	-	-	9.6	-
BERT モデル (ファインチューニングなし)	14.5	41.1	15.7	39.1
BERT モデル (ファインチューニングあり)	20.6	45.2	25.0	44.4
- ランダムマスキング + 対照的マスキング	21.7	46.0	27.4	47.4
+ 対句位置エンコーディング	22.6	47.5	27.4	48.0
+ 擬似教師データ	<b>23.4</b>	<b>47.9</b>	<b>30.4</b>	<b>49.1</b>
作業者 (正解率が最小値となった場合)	-	-	31.5	52.3
作業者 (正解率が最大値となった場合)	-	-	34.5	59.1
作業者 (3名の回答を統合した場合)	-	-	51.8	66.6

表 3.12: 各指標を満たす 100 件あたりの事例数

	意味的対照性	文の自然さ・流暢さ
作業者	94	90
提案手法	88	85

語予測タスクに適応させるのに有効であることがわかる。しかしながら、ファインチューニング無しと有りの場合で top-10・top-n 正解率における性能の差は top-1 に比べると小さい。これは、BERT モデルが事前学習時に文脈に基づき正解単語を正解候補として予測する能力を獲得していることを示唆している。

教師データでファインチューニングをした場合は、特に top-1 正解率が大きく向上した（全評価データで+6.1 ポイント、単語単位で+9.3 ポイント）。また、対照的マスキングにより全ての正解率が、対句位置エンコーディングにより特に top-10・top-n 正解率が向上した（前者は+0.8~+3.0 ポイント、後者は+1.5, +0.6 ポイント）。さらに、擬似教師データの活用により、特に top-1 の正解単語の向上を確認した（+0.8, +3.0 ポイント）。これらの提案手法が性能向上に寄与していることから、BERT モデルに対して対句穴埋めに特化したファインチューニングを行うことが重要であるとわかる。

辞書を用いたベースラインでは、単語単位の top-1 正解率で 9.6% という結果になった。この正解率の低さは、辞書に含まれるエントリのカバー率の低さに起因する。例えば、評価データに含まれる対義語の 39.3% しか辞書に掲載されていなかった。

表 3.11 は、穴埋めタスクを解いた時に正解率が最小・最大となった作業者の

結果も示している。これを見ると、正解率は作業員ごとに揺れが見られるものの提案手法よりも高い。しかしながら、正解率は最大でも top-1 で 34.5%、top-n で 59.1%と低く、このタスクの難しさが現れている。一方で、作業員 3 名の回答を統合し、その中に正解単語が含まれれば正解とする最も緩やかな評価を行うと、top-1 正解率は 51.8%、top-n 正解率で 66.6%となった。この top-1 正解率の大幅な向上は、評価データ内の各空欄を埋める際に複数の可能な単語が考えられることを示唆している。この正解単語が複数考えられる事象について調査するため、正解単語と一致しなかった作業員の回答と提案手法の予測結果の質について、別の作業員による主観評価を実施した。各回答・予測結果の内、意味的対照性と文としての自然さ・流暢さを満たす事例数を表 3.12 に示した。この分析では、作業員の回答と提案手法の予測単語両方が不正解だった事例から 100 件ランダムに抽出したものをを用いた。なお、作業員の回答は、3 名の回答からランダムに 1 事例につき 1 件を選択した。この結果、作業員の回答・提案手法の予測単語ともに、85%以上の単語が対義語として適切であることがわかった。

以上より、対句テンプレートに対する単語穴埋めでは、複数通りの正解単語が存在するため、単一の正解単語を用いた自動評価（正解率）では性能が過小評価されてしまうことがわかった。しかし、提案手法による予測結果を主観評価したところ、対義語としての適切さ、文としての自然さ・流暢さとも非常に高い正解率となっており、人間の目から見ると適切に対句生成ができていることがわかる。

### 3.3.5 事例分析

ベースライン（ファインチューニングなしの BERT モデル）、提案手法、各作業員による出力・回答例を表 3.13 に示した。

事例 (A) の（出会い—別れ）のように、対比関係がわかりやすい単語が存在する場合は、提案手法も作業員も正解単語を第一候補として出力することができる。また、予測すべき単語の周辺のみを考慮しているベースラインの出力と比べると、提案手法は「別れの曲—出会いの曲」という文中の大域的な構造に着目できていることがわかる。

事例 (B) は、正解単語は予測・回答されていないものの、対義語として機能している。この問題では、「地球」という単語に対して、規模の大きさや身近さ

表 3.13: 対句穴埋めタスクの出力・回答例

(A) 別れの曲だったのに、[MASK]の曲になった。	
正解	出会い
ベースライン	別れ, 最後, 今, 人生, 卒業
提案手法	出会い, 憧れ, 人生, 最高, 始まり
作業員 1	出会い, 再会, 初恋, 永遠, 永久
作業員 2	出会い, 始まり, スタート, 開始, 邂逅
作業員 3	出会い, 出逢い
(B) 地球の環境より、まず [MASK]の環境。	
正解	心
ベースライン	宇宙, 水, 地球, 太陽, 植物
提案手法	家族, 私, 周り, トイレ, 家
作業員 1	自宅, 家, 自分, 部屋, 職場
作業員 2	自分, 私, 周辺, 室内, 家内
作業員 3	国, 家庭, 町, 街, 周り
(C) [MASK]軽さから、忘れる軽さへ。	
正解	許せる
ベースライン	忘れる, 壊れる, 思い出す, 嬉しい, 積み重ねる
提案手法	忘れる, 思い出す, 覚える, 思い出す, 気づかない
作業員 1	持てる, 運べる, 重ねられる, 運搬できる, 探せる
作業員 2	感じる, 受ける, 分かる, 感じ取る, 勘づく
作業員 3	思い出す, 残る, 驚く

の度合いで対比できる単語を出力する。その点、提案手法の予測結果・作業員が作成した回答ともに自分の周りにある物や人を中心とした単語が並んでおり、どれも意味的対照性と文としての自然さを満たしている。しかし、正解単語はそれらの観点に加えて「物理的・精神的な対比」を持つ「心」となっている。このような事例に対応するためには、「二つの単語をどのような観点で対比させるのか」を明確にすることが必要であると考えられる。

また、人間にとっても正解することが難しい他の事例として、事例 (C) のような文が挙げられる。この事例で正解単語を予測するには、このキャッチコピーの言及対象がメガネ製品であるという事前知識と、その対象に対する推測やイメージを要する。このような事例にも対応するため、文の言及対象に関する追加情報を入力することも重要である。

### 3.4 ハードテンプレートに基づく文生成 まとめ

本研究では、強い構造制約としてハードテンプレートをモデルに与え、文を生成するアプローチに取り組んだ。制約なしに生成が困難な対象として、構造的・意味的な制約を併せ持つ対句を選択し、その生成に向けたテンプレートの自動抽出および穴埋めを行った。

ハードテンプレートの自動抽出では、対句文の生成に要する言語知識、具体的には文構造テンプレートと対照語・対義語の言語知識を構造解析により自動抽出するアプローチをとった。対句構造は、従来盛んに研究されている並列構造の特殊なケースであるとも考えられるが、解析上の明確な手がかりがないことから従来のモデルは計算量の問題により適用が難しい。そこで、本研究では単語の対応関係に注目し、さらに対句構造の特徴を組み込んだ解析手法の提案を行った。提案手法は、シンプルなアーキテクチャながら、比較手法である関係抽出ベースのモデルに比べ短い学習時間で予測精度を上回った。また、解析結果よりテンプレートと言語知識を抽出し、対句が用いられる文構造パターンは限られるためテンプレートを用いた文生成が可能であること、また既存の辞書にはない多様な単語ペアが用いられることを分析により示した。

また、ハードテンプレートへの穴埋めによる対句生成では、テンプレートの文脈を踏まえてスロットに対義語を穴埋めする手法を提案した。BERT モデルを対句穴埋めに適応させる方法として、対句データを活用したドメイン適応、対照的マスキング、対句位置エンコーディング、擬似教師データの収集方法を提案した。提案手法は、評価データにおいて、23.4%の top-1 正解率および 47.9%の top-10 正解率を達成した。これらの値は低く見えるが、空欄を埋められる正解単語は複数考えられるため、単一の正解単語に基づく自動評価では性能が過小評価されてしまう。しかし、主観評価により、提案手法による予測結果は、85%以上の事例が対義語として適切であることがわかった。つまりこれは対句文として判定可能な文が生成できていることを指す。

ハードテンプレートによる文生成は、特に対句のような複雑な構造的性質を持つ文にも対応可能であり、制御性が高いアプローチである。しかしながら、この手法は所望の文構造ごとにスロット位置を事前に同定する必要がある。今回は構造解析によるテンプレートの自動抽出を行ったが、対句の言語的性質を利用した手法であり、他の言語表現には適用できない。そして生成対象ごとに解析

方法を開発するのは非現実的である。また、テンプレートはスロットにのみ単語を当てはめることで文を生成するという強い活用方法に関する制約があるため、生成結果の多様性が大幅に減少してしまう。また、ハードテンプレート化しづらいタスクに対しては適用が困難であるという問題もある。そこで、ハードテンプレートを与えずに文構造を制御する方法が求められる。

## 第 4 章

### ソフトテンプレートを用いた文生成

前章で扱ってきたハードテンプレート（雛形文）に対して、スロットを持たないソフトテンプレート（見本となる用例）というものがある。ソフトテンプレートではスロット位置の同定が不要であり、既存の文をそのまま用いることができる。また、スロットへの穴埋めが求められるハードテンプレートとは異なり使い方が明示されない。このため、3章で対象とした対句のように強い構造制約を持つ表現ではなく、例えば能動態・受動態どちらにするか、何を主語や動詞にするかなど「この文に近いものを生成したい」という弱い制約による制御が可能となる。なお、この文構造制約は必ずしも守られるとは限らない。ソフトテンプレートに基づく文生成手法は、その参考にさせたい文が存在するときに適用可能であり、ある入力文  $x$  を出力文  $y^*$  に変換する際の追加入力としてソフトテンプレートをモデルに与え、出力文を制御する。

近年最も盛んに研究されているアプローチは、大規模なデータストアからソフトテンプレートを検索し、文生成モデルを補強するメモリ拡張 (Khandelwal et al. 2021, Borgeaud et al. 2022) である。入力文  $x$  に最も近い文  $x'$  を大量の文対を含むデータストアから見つけ、その  $x'$  に対応する正解文  $y'$  をソフトテンプレートとして得る (Cao et al. 2018)。このため、ある程度の規模のデータストアが必要である。また、類似した文を得やすい条件としては、例えば特定のドメインに特化した文生成タスクや、入出力間にある一定のパターンがあるタスクが挙げられる。実際に、例えば特許文や法文書といった特定のトピックに絞った機械翻訳でソフトテンプレートは特に有効であるとされる (Xu et al. 2020)。

ソフトテンプレートによる弱い制御の例として、図 4.1 に 3 種類のソフトテ

表 4.1: ソフトテンプレート (ST) により文構造を制御する日英翻訳の例

ソース文	ポルトガルで列車が脱線し, 3人が亡くなった.
ST-A	Two <b>died in</b> a Britain train collision.
出力 A	Three <b>died in</b> Portuguese train derailment.
ST-B	Two <b>people were killed in</b> Britain train collision.
出力 B	Three <b>people were killed in</b> Portuguese train derailment.
ST-C	A train collision in Canada <b>killed two people</b> .
出力 C	Portuguese train derail <b>killed three people</b> .

ンプレートで出力を制御した機械翻訳タスクでのイメージを示した。太字になっている部分が結果的に制御されたことになる。

例えばソフトテンプレート (ST) A では, 死亡した人数が主語にくる能動態の文になっており, B では受動態になっている。一方 C では列車が主語の能動態の文章になっている。この例だと, 例えば, 死亡した人数を強調したければ A や B のソフトテンプレートが好ましく, 逆にどこの列車による事故なのかを強調したければ C のソフトテンプレートが好ましいと考えられる。このように, どの出力文も正解ではあるが, どのソフトテンプレートを与えるかによって多様な文を生成できる。これは任意の文生成タスクで求められる能力のひとつである。

本研究では, このソフトテンプレートを用いて出力文の文構造を制御することを考える。ハードテンプレートと比較して明示的な制御はできないが, テンプレート化のプロセスが不要であり, より幅広いタスクに適用可能である。

このソフトテンプレートを用いて生成を制御するため, 本研究ではソフトテンプレートを書き換えることによって文を生成する手法 NeighborEdit を提案する。この編集操作は, トークンの挿入操作と削除操作のふたつからなる。例えば “I have an apple.” という英文を “Ich habe einen Apfel.” という独文に翻訳する例を考える。この時, まず英文のソフトテンプレート “I have a banana. – Ich habe eine Banane.” を学習データから検索する。次に, その検索された独文をソフトテンプレートとしてモデルに与え, ‘eine’ と ‘Banane’ を削除し, ‘einen’ と ‘Apfel’ を挿入することで翻訳結果 “Ich habe eine Banane.” が生成できるという流れである。

ソフトテンプレートは, 古くから品詞タグ付け (Daelemans et al. 1996) や用例ベースの機械翻訳 (Nagao 1984) などの幅広いタスクで活用されてきた。近年では主流な文生成モデルである自己回帰モデル (AR) (Sutskever et al. 2014) に適

用され、ニューラル機械翻訳 (Gu et al. 2018b, Xu et al. 2020, Khandelwal et al. 2021), 自動要約 (Cao et al. 2018, Peng et al. 2019) などのタスクで高い性能を達成してきた。

ソフトテンプレートを書き換える際は、文全体を把握した上で不要なトークンと必要なトークンを見分ける必要がある。そこで、双方向の文脈を考慮して複数のトークンを同時に予測可能な非自己回帰モデル (NAR) (Gu et al. 2018a, Ghazvininejad et al. 2019, Lee et al. 2020, Qian et al. 2021) をベースとする。そして、編集操作を行う NAR デコーダの入力にソフトテンプレートを用いる。あわせてソフトテンプレートに対する編集操作を学習させるための方策も提案する。ベースのアーキテクチャには、トークンの削除と挿入の二種類の編集操作を行える Levenshtein transformer (Gu et al. 2019) を採用する。

ソフトテンプレートに基づく手法は、一般にソフトテンプレートが生成に役立つ、つまり見本となる文であることを念頭においた設計になっている。しかしながら、このソフトテンプレートは大量の文から語彙などの類似度に基づき検索してくるため、常に参考になる文が得られるとは限らない。そして、もし本来参考にならないはずの文をソフトテンプレートとして与えられた場合、その事例に対して編集操作を学習・適用することは非常に困難である。したがって、本研究ではソフトテンプレートと出力文の語彙的な違いに基づく学習戦略も提案する。具体的には、ソフトテンプレート中心方策とターゲット文中心方策の二つの方策を統合する (4.2.4)。

実験結果より、提案手法である NeighborEdit はソフトテンプレートを用いた研究のベンチマークである JRC-Acquis 英独データセットでより高い品質の文 (vanilla Transformer に比べて+1.69 ポイント) をより少ないデコーディング回数 (1/18) で達成した。また、Data-to-Text タスク (WikiBio) や他の翻訳タスク (WMT'14 英独) においても NAR ベースラインの性能を上回った。最後に、提案手法が用いているソフトテンプレートについて分析する。

本手法には以下のメリットがある。

**解釈性を併せ持ったソフトテンプレートによる文生成の制御** 既存のすべてのソフトテンプレートに基づく文制御手法は、ソフトテンプレートをひとつのベクトルに圧縮して追加入力としている (Peng et al. 2019, Pandey et al. 2018, Guu et al.

2018, Song et al. 2018, Cao et al. 2018). これらの手法は実際に有効であることは知られているが、ソフトテンプレートが具体的に生成でどう使われたのかがわからないブラックボックスな手法である。一方で本研究ではソフトテンプレートに対して繰り返し編集操作を明示的に行うことで、文生成の全過程を把握可能なホワイトボックス手法である。

**適切なソフトテンプレートを得られない場合にも対応可能** 一般に、ソフトテンプレートは生成したい文（の入力側）との類似度に基づいて検索する。そのため、参考になる文とならない文両方がソフトテンプレートになり得るが、既存手法ではすべてのソフトテンプレートに対して同様の活用方法をとる。これは、参考にならない文を参考にすることによる悪影響を受けやすい。一方で本研究ではソフトテンプレートの類似度の高さによって学習・推論戦略を切り替えるため、その影響を軽減できることが期待される。

また、本研究は非自己回帰モデルの性能向上のための手法としても位置付けられる。非自己回帰モデル (Gu et al. 2018a) は、近年活発に研究が進められているテキスト生成の枠組みのひとつである。自己回帰モデル (Sutskever et al. 2014) では、デコーダの入力層から出力層への順伝播の計算を生成されるトークンの数だけ繰り返す必要があるが、非自己回帰モデルは複数のトークンを同時に予測するため、順伝播の計算が1回で済み、生成が速い。一方で、生成されるトークン間の依存関係を十分に捉えられず、自己回帰モデルに比べて生成文の品質が劣る傾向にある。これに対し、非自己回帰モデルの生成を複数回繰り返すことで生成文の質を向上させる手法が提案されているが、生成速度の優位性を維持したまま自己回帰モデルと同等の生成品質に引き上げることは達成できていない (Lee et al. 2020, Ghazvininejad et al. 2019)。本研究の手法は、ソフトテンプレートをデコーダの初期値として組み込む。これにより、提案手法には以下のふたつの副次的なメリットがある。

**非自己回帰モデルが生成する文の品質を向上させる** 非自己回帰モデルは生成されたトークン間の依存関係を捉えることが難しく、NAR モデルが生成した文の品質は AR モデルに劣る。これは既存の NAR が推論時にはデコーダは無あるいはスペシャルトークンから生成を始める必要があることが大きな要因である。

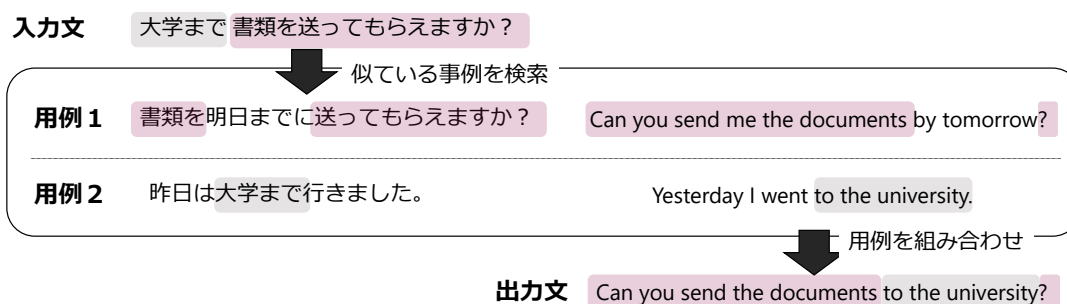


図 4.1: 用例ベースの機械翻訳のイメージ図.

この結果最初のデコーダの出力は低品質になるが、最初に生成された文の品質が低い場合、反復的に生成を繰り返したとしてもその品質を十分に上げることが難しい (Huang et al. 2022). これに対し、本研究ではデコーダの初期値をソフトテンプレートとすることで、より品質の高い文から生成を開始でき、最終的な出力文の品質も向上すると考えられる.

**非自己回帰モデルの生成速度を向上させる** 非自己回帰デコーディングは一度の順伝搬計算を行うだけで済み高速である. しかしながら NAR は生成される文の品質に劣るため、デコーディングの過程の繰り返しがほぼ必須である. 生成を繰り返すことで、生成されたトークンで条件付けられた分布を学習することになり、出力トークン間の依存関係を減らすことができる (Gu and Kong 2021). しかしながら本来の強みであった生成速度の速さが損なわれてしまうという問題がある. それに対し本研究では正解文により近い文からデコーディングを開始できるため、より少ない反復数で生成できることが期待される.

## 4.1 関連研究

### 4.1.1 ソフトテンプレートに基づく文生成

ソフトテンプレートもハードテンプレートと同様に文生成手法としては伝統的なものである\*. 長尾ら (Nagao 1984) は、図 4.1に示したように機械翻訳タスクにおいて生成したい文に類似した事例を対訳コーパスから検索し、フレーズ単位で組み合わせることで新しい文を生成した. それまでの翻訳手法は人手作成

\*ソフトテンプレートと呼ばれ始めたのは比較的最近のことである.

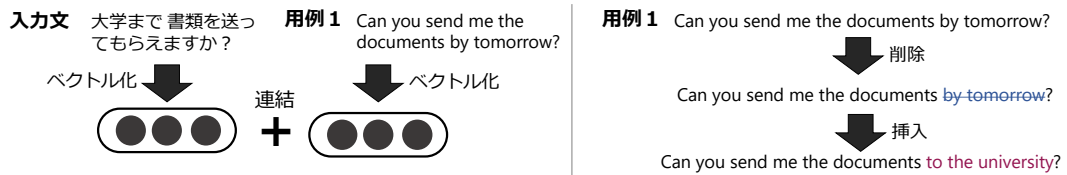


図 4.2: ソフトテンプレートに基づくニューラル文生成のイメージ図.

したルールに基づいておりコストがかかっていたため、ルール不要で対訳コーパスさえあれば翻訳できる手法としてインパクトを与えた。この手法は単純な文であれば十分に生成可能であり、生成の制御性および解釈性に優れている。なお、この既存の文をフレーズ単位で切り貼りするアイデアは最近になってニューラルモデルに組み込まれ、最新のモデルにおいても性能だけでなく解釈性や制御性に優れていることが確認されている (Wiseman et al. 2021)。

近年では Gu らがソフトテンプレートをアテンションベースのエンコーダ・デコーダモデルに組み込んだ機械翻訳モデルを最初に提案した (Gu et al. 2018b)。その後、ソフトテンプレートは機械翻訳に限らず自動要約や言語モデリングなど幅広いタスクで活用されるようになった (Cao et al. 2018, Xu et al. 2020, Bulte and Tezcan 2019, Borgeaud et al. 2022)。しかし、これらの研究はそのソフトテンプレートを入力文に連結したり、ソフトテンプレート用のエンコーダを用意するなどしてひとつのベクトルに圧縮している (図 4.2 の左側のイメージ図)。この仕組みは性能向上の面では有効であるが、追加のメカニズムやパラメータを要する上、生成過程は解釈できない。一方で、提案手法は追加のコンポーネントやパラメータなしに学習戦略のみを変更することで文生成にソフトテンプレートを組み込み制御性を向上させられる。さらには、一連の編集操作によりソフトテンプレートから出力文への変換を完全に解釈可能である。また、本研究は非自己回帰モデルにソフトテンプレートを組み込んだ最初の試みである。ソフトテンプレートは必ずしも文単位とは限らず、トークンレベル (Khandelwal et al. 2021, Zheng et al. 2021) やチャンクレベル (Borgeaud et al. 2022)、文レベル (Peng et al. 2019, Cao et al. 2018) で活用されている。トークンレベルやチャンクレベルのソフトテンプレートは正解文により近いソフトテンプレートを得ることができる一方で、各タイムステップごとに検索を繰り返す必要があり学習や推論の速度を下げる。一方で、提案手法はひとつの入力事例

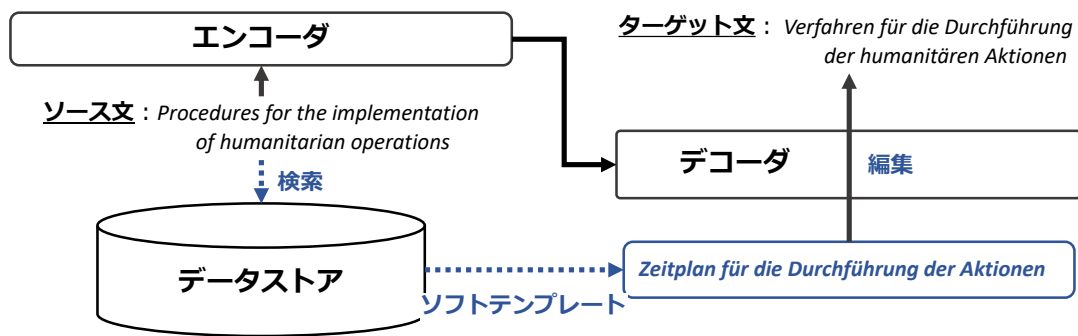


図 4.3: 機械翻訳タスクにおける提案手法の簡略図.

に対し一度の検索で済むため、速度を落とすことはない。また、提案手法ではノイズの多い文レベルのソフトテンプレートが検索された場合に対応できるよう、ソフトテンプレート中心方策やターゲット文中心方策などによって学習時と推論時でソフトテンプレートの使い方を調整している。

#### 4.1.2 非自己回帰モデルによる文生成

NAR デコーディングによる文生成はその生成速度の速さから近年多くの注目を集めている (Ghazvininejad et al. 2019, Lee et al. 2020, Gu et al. 2019, Stern et al. 2019, Lee et al. 2018). NAR モデルに残されている最も重要な課題は生成される文の品質の向上であるが、そのための主流なアプローチは反復的なデコーディング (Ghazvininejad et al. 2019) や AR モデルを用いた知識蒸留 (Kim and Rush 2016) などである。特に AR モデルなしに性能を向上させられる反復デコーディングでは、トークン置換 (Ghazvininejad et al. 2019, Qian et al. 2021) や挿入 (Stern et al. 2019), 削除・挿入 (Gu et al. 2019) を繰り返すことで文を段階的に洗練させていくことができるが、生成される文の品質と生成速度にはトレードオフの関係がある。本研究ではその編集操作に基づく反復デコーディングに基づく手法にソフトテンプレートを組み込むことで、制御性・解釈性を向上させつつそのトレードオフを改善させる。

## 4.2 提案手法

図 4.3に英文 “Procedures for the implementation of humanitarian operations” を独文 “Verfahren für die Durchführung der humanitären Aktionen,” に翻訳す

る場合の提案手法の簡略図を示した。提案手法はまず学習セット内の全ての文対からなるデータストアからソフトテンプレート “Zeitplan für die Durchführung der Aktionen (Schedule for implementation of measures)” を検索する。次に、その検索されたソフトテンプレートに対してトークンを削除・挿入することで文を生成する。

本節では、まず提案手法が取り組むソフトテンプレートに基づく文生成の問題設定とベースとなるアーキテクチャについて説明した後 (4.2.1, 4.2.2), 提案内容 (4.2.3, 4.2.4) について述べる。

#### 4.2.1 問題設定

本研究では、ソース文  $\boldsymbol{x}$  とソフトテンプレート  $\boldsymbol{z}_0$  を用いてターゲット文  $\boldsymbol{y}^*$  を生成する系列変換タスクに取り組む。ソフトテンプレートは、大量の文対を含むデータストアから検索することで得られる。今回、このデータストア  $(\mathcal{K}, \mathcal{V})$  は全ての学習事例  $(\mathcal{X}, \mathcal{Y})$  のソース文  $\boldsymbol{x}'$  のベクトル表現  $f(\boldsymbol{x}')$  をキー、ターゲット文  $\boldsymbol{y}'$  を値とするペアの集合として構築する。

$$(\mathcal{K}, \mathcal{V}) = \{(f(\boldsymbol{x}'), \boldsymbol{y}') \mid (\boldsymbol{x}', \boldsymbol{y}') \in (\mathcal{X}, \mathcal{Y})\} \quad (4.1)$$

ここで、 $f(\cdot)$  はトークン列を高次元ベクトル表現へマッピングする関数である (4.2.3)。そして、ソース文  $\boldsymbol{x}$  をベクトル表現  $f(\boldsymbol{x})$  に変換して検索のクエリとし、データストアからコサイン類似度が最も大きい文  $\boldsymbol{x}'$  を見つけ、その対応する値  $\boldsymbol{y}'$  をソフトテンプレート  $\boldsymbol{z}_0$  とする。

このようにして得たソフトテンプレート  $\boldsymbol{z}_0$  に対して編集操作を行うことで新しい文を生成する。学習時は  $\boldsymbol{z}_0$  からターゲット文  $\boldsymbol{y}^*$  への編集操作を行うが、ここでは模倣学習を用いる。模倣学習は強化学習の一種で、エキスパートの一連の行動を模倣し、同様の行動ができるように学習する。まず、文生成 (編集) プロセスをタプル  $(\mathcal{Y}, \mathcal{A}, \mathcal{E}, \mathcal{R}, \boldsymbol{z}_0)$  で定義されるマルコフ決定過程として考える。ここで、 $\boldsymbol{y} \in \mathcal{Y}$  はデコーディングの過程で部分的に編集された系列を表す。この系列を既存研究 (Stern et al. 2019, Wiseman et al. 2021) と同様にキャンバスと呼ぶ。 $\boldsymbol{y}$  の初期系列がソフトテンプレート  $\boldsymbol{z}_0$  である。また、 $\mathcal{A}$  は行動 (編集操作) の集合、 $\mathcal{E}$  は編集操作  $\boldsymbol{a}$  とキャンバス  $\boldsymbol{y}$  を受け取り、編集済みの系列を返す環境、 $\mathcal{R}$  は報酬関数である。

モデルはステップ  $k$  で前ステップで編集された  $n+1$  トークンからなるキャンバス  $\mathbf{y}^{k-1}$  ( $k=1$  ではソフトテンプレート  $\mathbf{z}_0$ ) を受け取り, 次の系列  $\mathbf{y}^k = \mathcal{E}(\mathbf{y}^{k-1}, \mathbf{a}^k)$  を生成するための行動 (編集操作)  $\mathbf{a}^k \in \mathcal{A}$  を選択する. ここで  $y_0 = \langle s \rangle$  と  $y_n = \langle /s \rangle$  は系列の最初と最後のトークンを表す.  $n$  の値は, 反復する過程でトークンの削除・挿入によって変わる. そして,  $n+1$  トークンからなる次の系列  $\mathbf{y}^k = \mathcal{E}(\mathbf{y}^{k-1}, \mathbf{a}^k)$  を生成するための行動 (編集操作)  $\mathbf{a}^k \in \mathcal{A}$  を選択し, 報酬  $r^k = \mathcal{R}(\mathbf{y}^k)$  を得る. 報酬関数  $\mathcal{R}$  は生成結果と正解系列間の距離を測る.

この行動を選択するための方策  $\pi$  は, キャンバス  $\mathbf{y}^{k-1}$  に対するデコーダの出力  $(\mathbf{h}_0, \mathbf{h}_1, \dots, \mathbf{h}_n)$  から行動空間  $\mathcal{A}$  への写像として定義される一般に, 方策  $\pi$  はパラメータ  $\theta$  をもつ線形分類器で近似するのが一般的である. なお, 本設定における模倣学習では, ある系列を別の系列に変換するための編集距離が最小となる編集操作は一意に定まるため, 実質的には教師あり学習である. この行動 (編集操作) を決定するための方策分類器について, 次節で説明する.

#### 4.2.2 準備: Levenshtein Transformer

NeighborEdit は, ベースのアーキテクチャに編集操作に基づく非自己回帰エンコーダ・デコーダモデルである Levenshtein Transformer (Gu et al. 2019) を採用する. 方策  $\pi$  は, (1) 削除すべきトークン位置の分類器  $\pi^{\text{del}}$ , (2) [PLH] を挿入すべき位置と数の分類器  $\pi^{\text{plh}}$ , そして (3) [PLH] に挿入すべきトークンの分類器  $\pi^{\text{tok}}$  の三つからなる.

#### 方策分類器

(1) トークン削除分類器  $\pi_\theta^{\text{del}}$  は各トークン  $y_i$  ( $i \in \{1, \dots, n-1\}$ ) が削除されるべきか ( $d_i = 0$ ) されないべきか ( $d_i = 1$ ) を二値分類で予測する.

$$\begin{aligned} \pi_\theta^{\text{del}}(d_i | i, \mathbf{y}) &= \text{softmax}(\mathbf{h}_i \mathbf{A}) \\ \mathbf{A} &\in \mathbb{R}^{d_{\text{model}} \times 2}, \forall i : i \in \{1, \dots, n-1\} \end{aligned} \quad (4.2)$$

(2) プレースホルダー分類器  $\pi_\theta^{\text{plh}}$  は全ての隣あうトークン間に挿入されるべきプレースホルダー ([PLH]) の数  $p_i \in \{0, \dots, K_{\text{max}}\}$  を予測する.  $p_i = 0$  の時,

モデルは  $y_i$  と  $y_{i+1}$  の間にはプレースホルダーは挿入しない.

$$\begin{aligned} \pi_{\theta}^{\text{plh}}(p_i|i, \mathbf{y}) &= \text{softmax}([\mathbf{h}_i; \mathbf{h}_{i+1}])\mathbf{B} \\ \mathbf{B} &\in \mathbb{R}^{(2d_{\text{model}}) \times (K_{\text{max}}+1)}, \forall i : i \in \{0, \dots, n-1\} \end{aligned} \quad (4.3)$$

ここで,  $[\mathbf{a}; \mathbf{b}]$  はベクトル  $\mathbf{a}$  と  $\mathbf{b}$  の連結を表す.

(3) トークン挿入分類器  $\pi_{\theta}^{\text{tok}}$  はプレースホルダー [PLH] に埋めるべきトークン  $t_i$  を予測する.

$$\begin{aligned} \pi_{\theta}^{\text{tok}}(t|i, \mathbf{y}) &= \text{softmax}(\mathbf{h}_i\mathbf{C}), \\ \mathbf{C} &\in \mathbb{R}^{d_{\text{model}} \times |\mathcal{V}|}, \forall i : y_i = [\text{PLH}] \end{aligned} \quad (4.4)$$

全体として, 方策分類器はパラメータ  $\theta = (\mathbf{A}, \mathbf{B}, \mathbf{C})$  からなる. モデルはキャンバス  $\mathbf{y}$  を受け取り, (1) トークン削除  $d_i$  ( $i \in \{0, \dots, n\}$ ), (2) プレースホルダー挿入  $p_i$  ( $i \in \{0, \dots, n-1\}$ ), and (3) トークン挿入  $t_i$  ( $\forall i : y_i = [\text{PLH}]$ ) の順序で編集操作を適用する.

## 学習時

モデルパラメータ  $\theta$  は削除  $\mathcal{L}_{\theta}^{\text{del}}$ ・挿入  $\mathcal{L}_{\theta}^{\text{ins}}$  それぞれの損失値の和に基づいて学習する. ここで,  $\mathbf{y}_{\text{ins}}$  と  $\mathbf{y}_{\text{del}}$  はそれぞれ挿入・削除対象のキャンバスである.

$$\mathcal{L} = \mathcal{L}_{\theta}^{\text{del}} + \mathcal{L}_{\theta}^{\text{ins}} \quad (4.5)$$

$$\mathcal{L}_{\theta}^{\text{del}} = - \sum_{\mathbf{y}^i \in \mathbf{y}_{\text{del}}, d_i^* \sim \mathbf{d}^*} \log \pi_{\theta}^{\text{del}}(d_i^*|i, \mathbf{y}_{\text{del}}) \quad (4.6)$$

$$\begin{aligned} \mathcal{L}_{\theta}^{\text{ins}} &= - \sum_{\mathbf{y}^i \in \mathbf{y}_{\text{ins}}, p_i^* \in \mathbf{p}^*} \log \pi_{\theta}^{\text{plh}}(p_i^*|i, \mathbf{y}_{\text{ins}}) \\ &\quad - \sum_{t_i^* \in \mathbf{t}^*} \log \pi_{\theta}^{\text{tok}}(t_i^*|i, \mathbf{y}'_{\text{ins}}) \end{aligned} \quad (4.7)$$

また,  $\mathbf{y}'_{\text{ins}}$  は [PLH] を  $\mathbf{y}_{\text{ins}}$  に挿入した後のキャンバスを表す. エキスパート操作  $\mathbf{a}^* \in \{\mathbf{d}^*, \mathbf{p}^*, \mathbf{t}^*\}$  はエキスパート  $e$  と編集済みのキャンバス  $\mathcal{E}(\mathbf{y}, \mathbf{a})$  のレーベンシュ

ソース文 (クエリ  $q$ ) : *Procedures for the implementation of humanitarian operations*

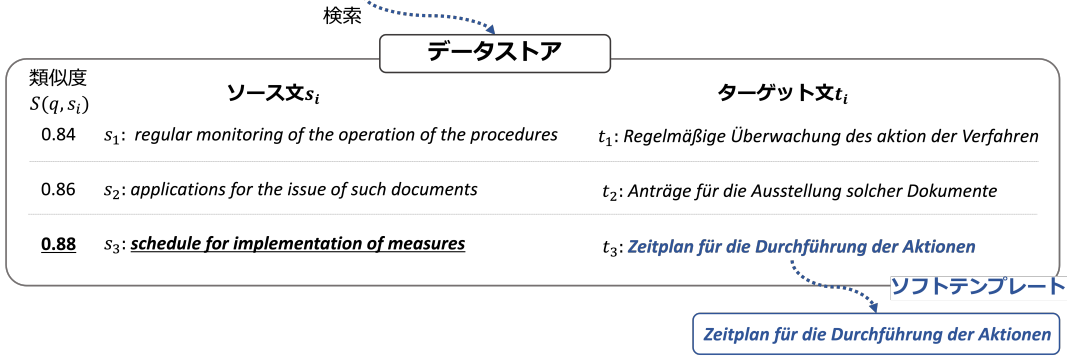


図 4.4: ソフトテンプレートのデータストアからの検索

タイム距離  $\mathcal{D}$  (Levenshtein 1966) を最小化するように選択される.

$$\mathbf{d}^* = \arg \min_d \mathcal{D}(\mathbf{e}, \mathcal{E}(\mathbf{y}_{\text{del}}, \mathbf{d})) \quad (4.8)$$

$$(4.9)$$

Levenshtein Transformer に関するさらなる詳細については Gu らの論文を参照されたい (Gu et al. 2019).

#### 4.2.3 ソフトテンプレートの検索

4.2.1に示した通り, ソフトテンプレート  $z_0$  を得るために学習データ内に含まれる全ての文対からなるデータストアから与えられた入力文に最も近い事例をデータストアから検索する. 図 4.4に示したように, 検索クエリとなる入力文を関数  $f(\cdot)$  を用いてベクトル化し, コサイン類似度が最も大きい事例をソフトテンプレートとして用いる. この検索フェーズは近傍検索とも呼ばれる. 学習時は, まず Top- $k+1$  のソフトテンプレートを検索し, そこから自身の事例を除くことで Top- $k$  の事例を得る. 関数  $f(\cdot)$  の設計を含むソフトテンプレートの検索方法は (1) 語彙マッチング (Xu et al. 2020, Gu et al. 2018b, Bulte and Tezcan 2019) と文ベクトルの類似度 (Khandelwal et al. 2021, Borgeaud et al. 2022) のふたつに大別される.

#### 語彙マッチング

既存研究ではファジーマッチングや n-gram マッチング (Xu et al. 2020, Gu et al. 2018b, Bulte and Tezcan 2019) など様々な方法が用いられてきた. なかでも本

研究では、大規模なデータにもスケールするようにPeng et al. (2019)と同様にトークンの出現頻度を用いる。具体的には、関数  $f(\cdot)$  には TFIDF 特徴量  $\text{tfidf}(\cdot)$  を用いて、ふたつの文  $s_i$  と  $s_j$  の類似度スコア  $S_{\text{TFIDF}}$  が最も大きい事例を検索する。

$$f(s_i) = \text{tfidf}(s_i) \quad (4.10)$$

$$S_{\text{TFIDF}}(s_i, s_j) = \frac{f(s_i) \cdot f(s_j)}{\|f(s_i)\| \|f(s_j)\|} \quad (4.11)$$

### 文ベクトルの類似度

関数  $f(\cdot)$  には既存のエンコーダを採用し、そのエンコーダから計算される文ベクトルのコサイン類似度  $S_{\text{SentVec}}$  が最も大きい事例を検索する。

$$f(s_i) = \mathbf{h}_i \quad (4.12)$$

$$S_{\text{SentVec}}(s_i, s_j) = \frac{f(s_i) \cdot f(s_j)}{\|f(s_i)\| \|f(s_j)\|} \quad (4.13)$$

ここで、 $\mathbf{h}_i, \mathbf{h}_j$  はそれぞれ文  $s_i, s_j$  をエンコードして得られた文ベクトルである。提案手法はトークンレベルでソフトテンプレートを編集するため、ソフトテンプレートはターゲット文と共通する単語を含むのが好ましい。したがって、本研究では  $k$  個のソフトテンプレート候補を  $S_{\text{SentVec}}$  に基づき検索し、その候補を語彙マッチング  $S_{\text{TFIDF}}(s_i, s_j)$  でリランキングする。

#### 4.2.4 ソフトテンプレートに対する編集操作

NeighborEdit は Transformer (Vaswani et al. 2017) をベースとした NAR モデルである。ソース文  $\mathbf{x}$  を入力として、検索されたソフトテンプレート  $\mathbf{z}_0$  をデコーダの初期値として受け取り、そのソフトテンプレートをトークン削除・挿入を繰り返すことで編集する (図 4.5)。このデコーディングの過程にどのようにしてソフトテンプレート  $\mathbf{z}_0$  を組み込むかが提案手法の設計において重要である。

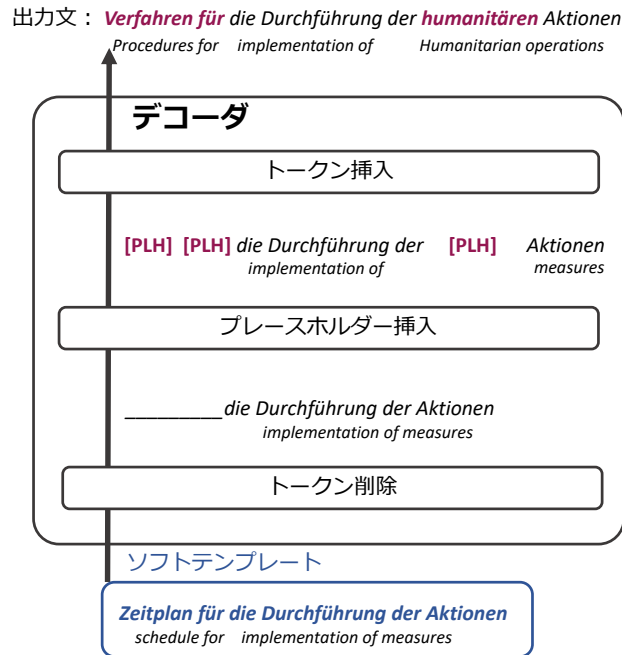


図 4.5: 提案手法によるソフトテンプレートの編集操作.

### ソフトテンプレートを編集するためのオラクル方策

模倣学習の過程で、モデルはオラクル方策  $\pi^*$  の行動を模倣するようにパラメトライズされた方策  $\pi_\theta$  を学習する。本研究では、ソフトテンプレート  $z_0$  を編集してターゲット文を効率的に生成するためのオラクル方策を得るための新しい戦略を提案する。ベースとなるアイデアは、推論時に現れる中間系列（部分的に編集されたソフトテンプレート）に対する編集操作を再現するような方策である。したがって、提案方策はソフトテンプレートとターゲット文の語彙的な違いに着目する。しかしながら、それらの類似度が低い場合、モデルはソフトテンプレートを完全に異なる文に書き換える必要性が生じてしまい、学習が困難になる。このような問題を軽減するため、ソフトテンプレート中心方策  $\pi_s^*$  とターゲット文中心方策  $\pi_t^*$  のふたつを混合させた方策を設計する。前者はソフトテンプレート  $z_0$  とターゲット文  $y^*$  の語彙的な類似性や違いを利用する一方で、後者はターゲット文  $y^*$  のみに着目する。

学習時は、モデルはランダムに  $\pi_s^*$  と  $\pi_t^*$  をバッチごとに選択する。さらに、ソフトテンプレートとターゲット文の類似度  $\text{sim}(z_0, y^*)$  がある閾値  $\beta$  よりも低い場合、入力操作に関する方策  $\pi_s^*$  は事例ごとに  $\pi_t^*$  に切り替える。なお、推論時

にソフトテンプレートから不要なトークンを確実に削除させるため、削除操作に関する方策には閾値は設定しない。まとめると、提案方策  $\pi^*$  は二種類の方策  $\pi_n^*, p_t^*$  を以下のように混合する。

$$\begin{aligned} \pi^* &= (\pi^* \text{ for deletion}, \pi^* \text{ for insertion}) \\ &= \begin{cases} (\pi_n^*, \pi_n^*) & (u < \alpha \wedge \text{sim}(\mathbf{z}_0, \mathbf{y}^*) > \beta) \\ (\pi_n^*, \pi_t^*) & (u < \alpha \wedge \text{sim}(\mathbf{z}_0, \mathbf{y}^*) \leq \beta) \\ (\pi_t^*, \pi_t^*) & (\text{otherwise}), \end{cases} \end{aligned} \quad (4.14)$$

$\alpha \in [0, 1]$  はハイパーパラメータ、 $u$  は  $[0, 1]$  の一様分布からサンプルした値である。ここで、類似度  $\text{sim}(\mathbf{z}_0, \mathbf{y}^*)$  は重複したトークンの数をソフトテンプレートないのトークンの数で割ったトークンの重複度合いで定義する。

上記の学習戦略により、提案手法はソフトテンプレートから不要なトークンを削除し、必要なトークンを挿入できるようになる。キャンバス  $\mathbf{y}$  を与えられた時、系列  $\mathbf{y}$  からエキスパートとなる文  $e$  を生成するための正しい行動を示すオラクル方策  $\pi^*$  は以下のように設計する。

**削除操作:**  $\pi^*(\mathbf{y}_{\text{del}})$  この編集操作では、キャンバス  $\mathbf{y}_{\text{del}}$  から不要なトークンを削除する。ソフトテンプレートに対する編集操作において重要なのは、ソフトテンプレート内に含まれる不要なトークンを削除することである。さらに、前ステップでモデルが挿入した不要なトークンも削除する必要がある。したがって、ソフトテンプレート  $\mathbf{z}_0$  からターゲット文  $\mathbf{y}^*$  に現れない単語を削除するか、 $\pi_\theta^{\text{del}}$  によってトークンを挿入された  $\mathbf{y}'$  である  $\mathcal{E}(\mathbf{y}', (\tilde{p}, \tilde{t}))$  から不要な単語を削除するように学習する。

$$\pi_n^*(\mathbf{z}_0) = M(\mathbf{y}^*, \mathbf{z}_0) \quad (4.15)$$

$$\pi_t^*(\mathcal{E}(\mathbf{y}', (\tilde{p}, \tilde{t}))) = M(\mathbf{y}^*, \mathbf{y}_{\text{del}}) \quad (4.16)$$

$$\tilde{p} \sim \pi_\theta^{\text{plh}}, \tilde{t} \sim \pi_\theta^{\text{tok}}$$

関数  $M(\mathbf{a}, \mathbf{b})$  は系列  $\mathbf{a}$  と  $\mathbf{b}$  の共通単語列を返す関数である。例えば、 $M(\text{“ABCDE,”}, \text{“ARE”})$  は  $\text{“AE”}$  を返す。

**挿入操作:**  $\pi^*(\mathbf{y}_{\text{ins}})$  ここでは、キャンバス  $\mathbf{y}_{\text{ins}}$  に必要なトークンを挿入してターゲット文  $\mathbf{y}^*$  を生成する.

$$\pi_n^*(M(\mathbf{y}^*, \mathcal{E}(\mathbf{z}_0, \tilde{d}))) = \mathbf{y}^* \tilde{d} \sim \pi_\theta^{\text{del}} \quad (4.17)$$

$$\pi_t^*(\mathcal{E}(\mathbf{y}^*, \tilde{d})) = \mathbf{y}^* \tilde{d} \sim \pi^{\text{RND}} \quad (4.18)$$

ここで、 $\pi^{\text{RND}}$  はランダムに単語を削除する方策、 $\mathcal{E}(\mathbf{y}^*, \tilde{d})$  は方策  $\pi^{\text{RND}}$  に基づきランダムにトークンを削除された後の系列を表す。  $M(\mathbf{y}^*, \mathcal{E}(\mathbf{z}_0, \tilde{d}))$  は、方策  $\pi_\theta^{\text{del}}$  に基づきソフトテンプレートからトークンを削除し、 $\mathbf{y}^*$  にはあって削除済みの系列にはないトークンを挿入するためのものである。

### 推論時のソフトテンプレートの活用方法

モデルは初期状態であるソフトテンプレートを受け取り、キャンバス  $\mathbf{y}$  に対して最も確率の高い編集操作を繰り返し選択・適用することで文を生成する。生成プロセスは、最大反復数に達するか連続して同じ編集操作が行われた時に終了する。

しかしながら、もしデータストア内に近いソフトテンプレートが存在しない場合にそれを初期状態として使うことで性能を悪化させる可能性がある。そこで、本研究ではオラクル方策と同様に切り替えアプローチを採用する。具体的には、もしソース側のソフトテンプレートとの類似度  $\text{sim}(\mathbf{z}_0, \mathbf{x})$  が式 (4.14) で使われている閾値  $\beta$  よりも低い場合、デコーダの初期状態にはソフトテンプレートの代わりに他の NAR モデルと同様スペシャルトークン  $\langle s \rangle$  を使うこととする。

## 4.3 実験

### 4.3.1 実験全体の詳細

本研究では、複数のタスクやデータセットを用いた実験により提案手法の有効性を示す。まず、主な結果としてソフトテンプレートを用いた研究のベンチマークとして用いられる JRC-Acquis データセットを用いた機械翻訳タスクでの性能を報告する (4.3.2)。そして、追加実験として WikiBio データセットを用いた

表 4.2: 実験で用いたデータセットの統計

データセット	学習データ	開発データ	評価データ
JRC-Acquis	475,330	2,945	2,937
WikiBio	582,659	72,831	72,831
WMT'14	3,961,179	3,000	3,003

Data-to-text タスク, WMT'14 データセットを用いた機械翻訳タスクでの結果も示す (4.3.3).

**モデルの詳細とパラメータ設定** NeighborEdit は,  $d_{\text{model}} = 512$ ,  $d_{\text{hidden}} = 2048$ ,  $n_{\text{head}} = 8$ ,  $n_{\text{layer}} = 6$  とする. 4.2.4で提案したオラクル方策では, 全ての設定でハイパーパラメータ  $\alpha = 0.6$  とした. また, JRC-Acquis の全ての検索法と WikiBio の TableMatch, WMT'14 の TFIDF は  $\beta = 0.3$ , WMT'14 の SentVec+TFIDF では  $\beta = 1.0$  とした. これらの数値は, 他のハイパーパラメータと同様開発データでの性能で自動的に決定した. 閾値  $\beta = 1.0$  は推論時に全ての事例でデコーダの初期値をソフトテンプレートではなくスペシャルトークンとすることを指す. しかし, 学習時は削除操作を学習するために全事例でソフトテンプレートを用いる. 方策分類器では,  $K_{\text{max}} = 255$  とした. ソース側とターゲット側の単語埋め込み行列およびデコーダ内の3つの編集操作のパラメータはそれぞれ共有する.

**近傍検索の詳細** 4.2.3で説明した TFIDF, SentVec+TFIDF ( $k=50$ ) でデータストアを検索し, 入力に最も近い事例を選択する. 文ベクトルのエンコーダには多言語コーパスで事前学習済みの SBERT モデル<sup>†</sup> (Reimers and Gurevych 2019) を, 近傍検索には faiss<sup>‡</sup> を用いた.

TFIDF については, まず高次元の疎な TFIDF ベクトルを作成したあと, 特異値分解 (SVD) によって 512 次元まで圧縮する. ソフトテンプレートは faiss で検索したあと, Top-50 のソフトテンプレート候補を圧縮前の TFIDF ベクトルを用いてソフトテンプレート候補とソース文のコサイン類似度を求め, リランキングすることで情報損失を軽減する. SentVec については, 事前学習されたエンコーダを用いて求めた 512 次元のベクトルをそのまま検索に用いる. faiss を用いた近傍検索では, IndexFlatIP インデックスを構築し, 線形検索により完全

<sup>†</sup><https://huggingface.co/sentence-transformers/distiluse-base-multilingual-cased-v1>

<sup>‡</sup><https://github.com/facebookresearch/faiss>

照合探索を行なうことでベクトルのコサイン類似度が最も大きな事例を検索する。近傍検索はNARモデルのデコーディングよりも速く、それによるオーバーヘッドは無視できるほど小さい。

**学習の詳細** 語彙の構築にはBPE (Sennrich et al. 2016b) を用いた。マージ操作数は、JRC-Acquisで20,000, WMT'14で40,000である。なお、WikiBioではサブワード分割は行わない。

パラメータの初期化はBERT (Devlin et al. 2019) のスキーマに従った。オプティマイザにはAdam (Kingma and Ba 2015) ( $\beta = (0.9, 0.98)$ ) を採用し、ドロップアウト率は0.3, 重み減衰は0.01とした。学習率スケジュールにはInverse Square Root Decayを用い、最大の学習率は $5 \times 10^{-4}$ , Warmupステップは10,000とした。JRC-AcquisとWikiBioの全てのモデルはNVIDIA Tesla P100 GPUを四枚, WMT'14のモデルはNVIDIA V100 GPUを16枚用いて学習した。

**評価** 各タスクでの生成文の品質評価に加え、一事例あたりの推論に要した平均反復回数と平均生成時間を測定する。平均生成時間は、先行研究 (Xu and Carpuat 2021) に従い、1個のGPUを用いてバッチサイズ1で文を生成し、一事例を生成し終えるまでの生成時間の平均値を求めた。なお、提案手法の平均生成時間にはソフトテンプレートの検索にかかる時間も含めた。

推論に用いるモデルは、開発データ上でのBLEUスコアが最大となったチェックポイントを採用した。SacreBLEUのハッシュ値はBLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.5.1, 統計的有意差検定ではpaired bootstrap resampling法 (Dror et al. 2018) を採用した。

#### 4.3.2 ベンチマークでの実験 (JRC-Acquis)

**データセット詳細** 本実験では法文書コーパスであるJRC-Acquis<sup>§</sup> (英独) を利用する。本コーパスは意味的に関連する文を多く含むことから、ソフトテンプレートを活用した多くの既存研究でベンチマークとして用いられている (Li et al. 2016, Gu et al. 2018b, Xu et al. 2020, Cai et al. 2021)。本実験では、重複する対訳対の事例を除外した後、Gu et al. (2018b) と同様の前処理を行なった。データサイズは表 4.2に示した。

<sup>§</sup><https://opus.nlpl.eu/JRC-Acquis.php>

表 4.3: 比較手法と提案手法のデコーダの初期値一覧

	モデル	デコーダの初期値
自己回帰モデル	Transformer	[BOS]
非自己回帰モデル	GLAT+CTC	Encoder( $\boldsymbol{x}$ )
	NAT-ITR	$\boldsymbol{x}$
	NAT-CRF	[PAD], ..., [PAD], [EOS]
非自己回帰モデル	CMLM	[MASK], ..., [MASK]
(反復デコーディング)	LevenshteinT	<s></s>
	NeighborEdit	TFIDF, SentVec+TFIDF

**モデル** 自己回帰モデル (AR) のベースラインには Transformer (Vaswani et al. 2017) を用いる。非自己回帰モデル (NAR) のベースラインには、1 度のみデコーディングを行う GLAT+CTC (Qian et al. 2021) と反復的デコーディングを行う Levenshtein Transformer (Gu et al. 2019) (LevenshteinT), NAT-ITR (Lee et al. 2018), NAT-CRF (Sun et al. 2019), CMLM (Ghazvininejad et al. 2019) を採用する。それぞれのデコーダの初期値は表 4.3 にまとめた。推論はビーム幅 4 で行う。モデルは fairseq (Ott et al. 2019)<sup>¶</sup> をベースに実装した。学習は全て 1 ステップあたりのバッチサイズを 32,768 前後、ステップ数は 100,000 回とした。

**評価** 生成の品質評価では、SacreBLEU (Post 2018), ChrF (Popović 2015), COMET (Rei et al. 2020) を用い、大文字小文字を区別して比較する。

**実験結果** 表 4.4 に生成の品質及びデコーディングの速度の結果を示した。

ソフトテンプレートを初期状態に用いる NeighborEdit は、非自己回帰モデルのベースラインを上回った (+7.04 BLEU スコア, +3.59 ChrF スコア, +0.013 COMET スコア)。さらには、反復 NAR モデル間で平均反復回数も最小となった。NeighborEdit はデコードを 1 反復あたり削除 1 回、挿入 2 回で合計 3 回繰り返すため、平均反復回数の少なさに対して平均生成時間はそこまで短いわけではない。しかしながら、提案手法が生成文の品質を大きく向上させたことは注目に値する結果と言える。ソフトテンプレートの検索法については、結果は語彙マッチングと文ベクトル両方を使った場合 (SentVec+TFIDF) の方が前者のみを使った場合 (TFIDF) よりも僅かに性能が高い結果となった。これは語彙

<sup>¶</sup><https://github.com/pytorch/fairseq>

表 4.4: JRC-Acquis データセットにおける BLEU, ChrF, COMET, 平均反復回数, 平均生成時間 [ms]. (太字は非自己回帰モデルでの最良値, 下線は有意差検定により LevenshteinT に対して有意水準  $p < 0.05$  での有意な性能向上であると判断された数値を表す.)

モデル	BLEU	ChrF	COMET	平均反復数	生成時間 [ms] (検索時間)
Transformer	53.90	72.51	0.681	30.69	300.8
GLAT+CTC	46.75	66.53	0.329	1.00	19.8
NAT-ITR	38.53	60.71	-0.034	3.57	46.7
NAT-CRF	26.94	50.39	-0.698	3.25	64.7
-----	47.81	67.55	0.548	10.00	137.8
CMLM	46.98	67.06	0.490	6.00	88.0
-----	44.12	65.17	0.293	3.00	50.6
LevenshteinT	48.36	69.00	0.592	2.51	93.0
NeighborEdit (TFIDF)	<u>55.40</u>	72.59	0.603	1.66	75.4 (0.17)
(SentVec+TFIDF)	<b>55.59</b>	<b>72.74</b>	<b>0.605</b>	1.65	77.0 (0.08)

的な類似度と大域的な意味的類似度の両方を用いることで出力文により近いソフトテンプレートを得られることを示唆している。

さらに, このデータセットにおいて NeighborEdit は強力な AR モデルのベースラインに BLEU スコアで +1.50 上回った. 平均反復数は AR モデルのベースラインの 1/18 にまで, 平均生成時間も 1/2 まで削減できた.

これらの結果は NeighborEdit は一貫してデコーディングの速度を犠牲にせずに生成の品質を向上させられる, つまり NAR モデルにソフトテンプレートを組み込むことは有効であることを示唆している.

本手法では, ソフトテンプレートに対する編集操作と, それを学習させるための方策を提案している. それぞれが性能向上にどの程度寄与するかを調べるため, オラクル方策内のソフトテンプレートとデコーダの初期値の影響をそれぞれ調査した. また, ランダムに検索した事例をデコーダの初期値とする RAND ベースラインの結果も調べた.

表 4.5によると, ソフトテンプレートをオラクル方策とデコーダの初期値両方に組み込むことで性能が大きく向上した. さらに, ランダムな事例で初期化すると性能が著しく低下する. これらの結果から, 性能向上の鍵は, 有用な近傍探索法を選択してソフトテンプレートをオラクル方策やデコーダの初期化に効

表 4.5: JRC-Acquis データセットにおける, オラクル方策とデコーダの初期値にソフトテンプレートを活用する場合としない場合の BLEU 値の比較. 近傍検索には最も性能が良かった SentVec+TFIDF を用いた.

オラクル方策に ソフトテンプレートを組み込むか	デコーダの初期値に用いる系列		
	<s></s>	$z_0$	RAND
No ( $\pi_t^*$ )	48.36	42.05	3.00
Yes ( $\pi_t^* + \pi_n^*$ )	45.67	55.59	44.09

果的に取り入れることであることが示唆された.

### 4.3.3 他のデータセットでの実験

また, ソフトテンプレートを用いた先行研究においてあまり検討されていない他のデータセットについても実験を行った. これらの実験では, NAR モデルのベースラインとして Levenshtein Transformer を採用した.

#### Data-to-text タスク (WikiBio)

WikiBio データセット<sup>||</sup> (Lebret et al. 2016) を用いて, データから文への変換タスクにおける NeighborEdit の有効性を評価した. データサイズは表 4.2 に示した. このタスクでは, モデルは人物情報を記述したファクトテーブルを受け取り, 略歴を生成する. 略歴はある程度決まったフォーマットをもつため, このタスクのターゲット側にはパターンが存在すると考えられる.

ソフトテンプレートを得るために, Wiseman et al. (2021) と同様に, 2つの系列化された infobox  $s_i$  と  $s_j$  間の類似度スコア  $S_{TM}$  を算出した.

$$S_{TM}(s_i, s_j) = F_1(\text{fields}(s_i), \text{fields}(s_j)) + 0.1F_1(\text{values}(s_i), \text{values}(s_j)) \quad (4.19)$$

ここで,  $\text{fields}(s)$  は infobox  $s$  内のフィールドタイプ (例 name) を抽出したものの,  $\text{values}(s)$  は infobox  $s$  内の値 (例 ‘Obama’) に現れるユニグラムを抽出したものの,  $F_1$  は F1 スコアを表す. この検索法を TableMatch と呼ぶ. 学習は全て 1 ステップあたりのバッチサイズを 16,384 前後, ステップ数は 300,000 回とした.

<sup>||</sup><https://rlebret.github.io/wikipedia-biography-dataset/>

表 4.6: WikiBio データセットにおける BLEU, NIST の値と平均反復数, 平均生成時間. **太字** になっている数値は非自己回帰モデルの中で最も良い性能, 下線の引かれた値は LevenshteinT に対して統計的有意差が認められた性能 ( $p < 0.05$ ) を表す.

モデル	BLEU	NIST	反復数	生成時間 [ms]
Transformer	46.91	9.48	23.70	186.4
LevenshteinT	43.09	8.79	2.25	56.4
NeighborEdit	<b><u>44.96</u></b>	<b>9.24</b>	<b>2.08</b>	55.8

性能は BLEU, NIST\*\*, 平均反復数で評価する.

表 4.6に結果を示した. NeighborEdit は NAR ベースラインを BLEU スコアで 1.87 ポイント上回り AR ベースラインとの性能差をより少ない反復回数で縮められた. これにより, このデータセットにおいても提案手法は有効であることが示された.

### 機械翻訳タスク (WMT'14)

これまでの実験では, ソフトテンプレートが有効であると期待されるデータセットにおける NeighborEdit の有効性を検証した. 本節では, 機械翻訳タスクでよく用いられる WMT'14 データセット (英独) (Bojar et al. 2014) を用いる. WMT'14 は, JRC-Acquis とは異なり幅広いトピックをカバーしている. データサイズは表 4.2に示した. データセットのダウンロードおよび前処理には, fairseq で公開されているスクリプト<sup>††</sup>を用いた. 学習は全て 1 ステップあたりのバッチサイズを 65,536 前後, ステップ数は 300,000 回とした.

その結果, SentVec+TFIDF を用いた提案手法が Levenshtein Transformer を 0.63 ポイント上回った (表表 4.7). しかしながら, BLEU スコアの増加幅は他のデータセットよりも小さいうえ, 反復数も増加している. また, ソース文にソフトテンプレートを連結して Transformer に入力した AR モデル (+CONCAT) の性能を調べても, ソフトテンプレートによる性能向上は見られなかった. 以上より, WMT'14 英独データセットは JRC-Acquis や WikiBio と異なる性質を持つ可能性がある.

\*\*We used this evaluation script: <https://github.com/tuetschek/e2e-metrics>.

††<https://github.com/pytorch/fairseq/tree/main/examples/translation#WMT'14-english-to-german-convolutional>

表 4.7: WMT'14 英独データセットにおける BLEU, 平均反復数, 平均生成時間. **太字** になっている数値は非自己回帰モデルの中で最も良い性能, 下線の引かれた値は LevenshteinT に対して統計的有意差が認められた性能 ( $p < 0.05$ ) を表す.

Model	BLEU	反復数	生成時間 [ms]
Transformer	27.06	28.10	219.0
+CONCAT	26.91	28.02	216.5
LevenshteinT	23.98	2.83	74.2
NeighborEdit	TFIDF 24.17	2.60	72.1
	SentVec+TFIDF <b>24.61</b>	3.05	83.8

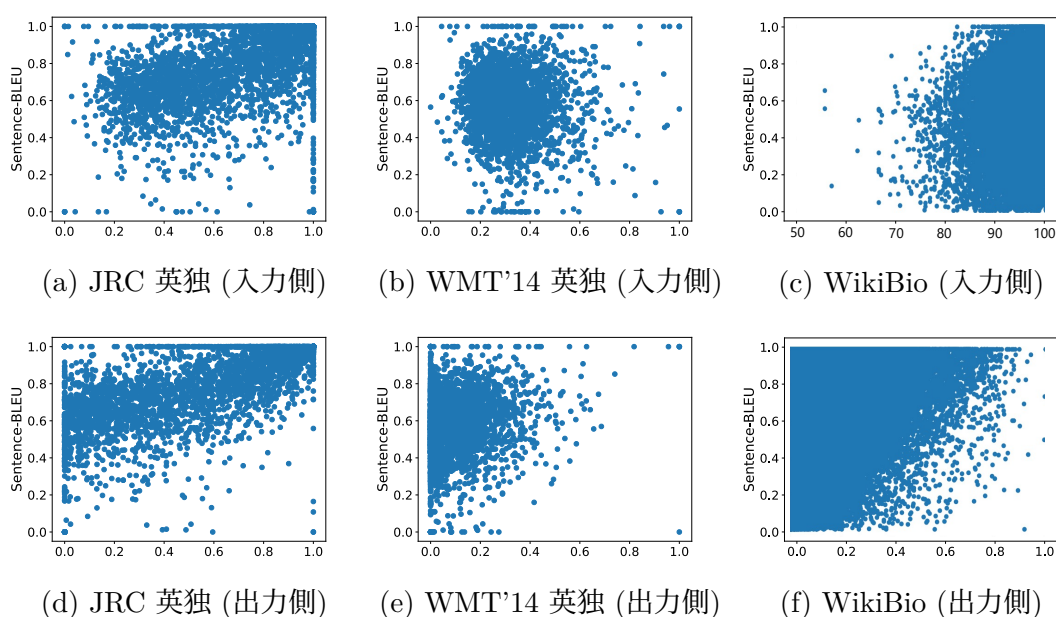


図 4.6: 開発データにおける各事例とそのソフトテンプレートの類似度 ( $x$  軸) とその sentence-BLEU ( $y$  軸) の関係を示した図.

#### 4.3.4 ソフトテンプレートの分析

本節では, 提案手法が有効である条件, 特にソフトテンプレートの品質やその性能との関係などデータセットごとの性質について調査する. 図 4.6は開発データの各事例と対応するソフトテンプレートの類似度 ( $x$  軸) とその生成結果の sentence-BLEU ( $y$  軸) の関係を表す. 入力側の図 4.6a, 4.6b, 4.6cの  $x$  軸は入力文  $\boldsymbol{x}$  とソフトテンプレートに対応する入力側の文  $\boldsymbol{x}'$  の tf-idf ベクトルのコサ

イン類似度を表す。逆に、出力側の図 4.6d, 4.6e, 4.6f の  $x$  軸は正解文  $y$  とソフトテンプレート  $z_0$  の tf-idf ベクトルのコサイン類似度を表す。つまり入力側の図の横軸は検索クエリに対してどれくらい近い事例を検索できるか、出力側は検索されてきたソフトテンプレートが実際にどの程度正解文に近いのかを表す。

図 4.6d によると、JRC-Acquis データセットは類似度の高い事例が含まれており、平均コサイン類似度は 0.521 であった。また、検索された事例とターゲット文との類似度が高いほど BLEU スコアも高くなる傾向が読み取れる。

一方で、図 4.6e は WMT'14 データセットが類似度の高い事例をあまり含まないことを明らかに示唆している。平均コサイン類似度は 0.105 であった。これは、NeighborEdit が遠い事例から多くの編集操作を行う必要があるということを示す。この難しさは、表 4.7 の平均反復数の大きさからも読み取れる。言い換えれば、WMT'14 データセットには似ている事例が少なく、初期値として有効なソフトテンプレートを得ることが難しいため NeighborEdit があまり有効ではないということになる。さらに、WMT'14 のソース側とターゲット側（それぞれ図 4.6b, 4.6e）の傾向は JRC-Acquis（図 4.6a, 4.6d）とは大きく異なる。ソース側とターゲット側の類似度の相関係数は JRC-Acquis では 0.839 であったのに対し WMT'14 では 0.421 であった。これは、WMT'14 はトピックと同様に文に多くのバリエーションがあり、ソース側で似ている事例でもターゲット側ではそうではない傾向がより強いことを意味する。この WMT'14 における低い相関が、ソース側の語彙的な類似度に基づく提案手法の有効性を下げるひとつの要因であると考えられる。したがって、例えば言い換え表現に頑健な手法など、ソース側とターゲット側のギャップを埋めるような近傍検索法が必要である。

WikiBio データセットでは、ソース側のコサイン類似度（図 4.6c）は 99.15 と高かった。これは、ふたつの infobox テーブルが同じフィールド名を含むだけで類似度が高くなるためである。対照的に、ターゲット側のコサイン類似度（図 4.6f）は 0.13 と低い結果となった。これらの結果は、infobox と文で異なる類似指標を用いたことでソース側とターゲット側で類似度分布が大きく異なったことを示唆する。今回は既存研究に基づき 4.3.3 節で説明したようなソース側の類似度指標を用いたが、略歴パターンの類似度をより反映させたソース側の類似度指標、例えばエンティティの職業により着目するなどの工夫が必要かもしれない。

このようにデータセットごとに得られるソフトテンプレートや適切な類似度指標は異なるが、総じてソフトテンプレートとターゲット側の文の類似度が高

い場合は文単位の BLEU が高くなる。BLEU は局所的な単語の並びの一致度を表すため、この結果はソフトテンプレートによって文構造（単語や句の並びや関係）を弱く制御できていることを示唆している。

#### 4.3.5 実際の実出力例

図 4.7 に Levenshtein Transformer と NeighborEdit の JEC-Acquis データセットでの実際の実出力例を示した。挿入されたトークンを赤色、削除されたトークンを青色で表記した。提案手法は、ソフトテンプレートから不要なトークンを削除し、必要なトークンを挿入することでより少ない反復回数で正しい文を生成できている。

### 4.4 ソフトテンプレートに基づく文生成 まとめ

本研究では、スロットを持たないソフトテンプレートで文生成を文構造で弱く制御する手法を提案した。具体的には、非自己回帰モデルにソフトテンプレートを組み込み、明示的にトークンの挿入や削除を行った。提案手法は生成過程を一連の編集操作として把握できるホワイトボックスな手法であるという強みもある。実験結果より、全てのデータセットにおいて既存の NAR モデルよりも少ない反復回数で生成文の質を改善できることを報告した。また、ソフトテンプレートによって出力文の構造を弱く制御できていることが示唆された。NeighborEdit は、特許や議会の議事録など作文パターンがあるドメインで特に有効であることが期待される。

ソース文	( 48 ) Capacity utilisation increased by 56 % between 1996 and the IP.
ターゲット文	( 48 ) Die Kapazitätsauslastung stieg von 1996 bis zum UZ um 56 %.
<b>Levenshtein Transformer</b>	
削除 1	
挿入 1	[PLH] [PLH] [PLH] [PLH] [PLH] [PLH] [PLH] [PLH] [PLH] [PLH] [PLH] [PLH] [PLH] [PLH] [PLH] [PLH] [PLH] ( 48 ) Die Kapazitätsauslastung stieg von 1996 bis dem UZ UZ um 56 %.
削除 2	( 48 ) Die Kapazitätsauslastung stieg von 1996 bis dem UZ um 56 %.
挿入 2	( 48 ) Die Kapazitätsauslastung stieg von 1996 bis [PLH] dem UZ um 56 %. ( 48 ) Die Kapazitätsauslastung stieg von 1996 bis zu dem UZ um 56 %.
<b>NeighborEdit</b>	
ST (英)	( 56 ) Export prices of the two cooperating Thai producers have increased by 6 % between 1996 and the IP.
ST (独)	( 56 ) Die Ausfuhrpreise der beiden kooperierenden thailänd@@ ischen Hersteller stiegen von 1996 bis zum Untersuchungszeitraum um 6 %
削除 1	( ) Die von 1996 bis zum um %.
挿入 1	( [PLH] ) Die [PLH] [PLH] von 1996 bis zum [PLH] um [PLH] %. ( 48 ) Die Kapazitätsauslastung stieg von 1996 bis zum UZ um 56 %.

図 4.7: Levenshtein Transformer と NeighborEdit の翻訳例. ST はソフトテンプレートを表す.

## 第 5 章

### 結論

本研究では、深層学習のブラックボックス問題によって失われた自然言語生成モデルの制御を可能にすることを目的とし、出力文を文構造テンプレートで制御する手法を提案した。本研究で対象とした文構造は、テンプレートによって与えられる単語や句の並びや関係のことを指す。文構造を制御することで、生成される文の言い回しを変えたり品質を向上させることができる。

提案手法では、ハードテンプレートとソフトテンプレートという異なる二種類のテンプレートを対象とし、それらに対して単語の穴埋めや挿入、削除を行うことで、文生成を制御することを可能にした。まず、ハードテンプレートに基づく文構造の制御では、強い構造的性質を持つ対句を生成対象とし、テンプレートの抽出および穴埋めによる文生成に取り組んだ。テンプレート抽出は、文中の対句の範囲を構造解析によって同定し、その部分をスロットとすることで実現した。この時、構造解析アルゴリズムの計算量を削減するため、対句の言語的性質を考慮した段階的な解析手法を提案した。これにより、効率的かつ高性能なテンプレート抽出が行えた。次に、対句テンプレート内のスロットに対して単語を穴埋めすることで対句を生成した。ここでは、対句データの不足に対応するため、大規模コーパスで事前学習されたモデル BERT を対句穴埋めタスクに適応させるための手法を提案した。定量的・定性的な評価により、提案手法は対句として判定可能な文を生成できたことを確認した。ハードテンプレートはスロットに単語を当てはめることで文を生成するため、探索空間を大幅に削減できる。これにより、7,500 件という深層学習モデルの学習には小さなサイズのデータセットでも十分対句として適切な文を生成できたことから、データ量に対する頑健

性もハードテンプレートのもつ一つの強みだと考えられる。

次にソフトテンプレートに基づく文構造の制御では、その文構造を参考にしつつ書き換えることによる文生成に取り組んだ。ここでは、大量の文対が含まれるデータストアから、語彙的・分布的な類似度に基づきソフトテンプレートを検索した。そして、検索してきたソフトテンプレートに対し、トークンを挿入・削除する編集操作を繰り返すことで文を生成する。提案手法は、機械翻訳や Data-to-text タスクにおいて特に性能を向上させられることがわかった。また分析により、ソフトテンプレートによって生成された文の構造が弱く制御できていることが示唆された。

本研究は文構造をテンプレートに対する編集操作によって制御したことで、副次的なメリットとして解釈性も向上している。今まで制御性と解釈性それぞれ個別の問題に取り組んだ研究は多いが、両者を両立させるニューラル手法についてはあまり研究されてこなかった。実際すでに実用化されている自然言語処理サービスは全てがブラックボックスである。しかし、それらが真に人間と協働可能なシステムになり得るかと言われると現状そうではない。挙動の不安定さや信頼性の低さの解決が急務であり、そのためには解釈可能な制約によって出力文を高度に制御できる研究の促進および実用化が必要不可欠である。これはユーザーだけではなくシステムの開発者にとってもメリットである。なぜなら、生成過程を観測できるということは、例えば編集操作のどの過程でどのトークンを間違えたのかエラー分析を行えるため、性能を向上させるための方針を立てやすいためである。近年はより多くのパラメータやデータを使ってブラックボックス手法の性能をあげていくアプローチが主流にはなっているが、ユーザー視点、また開発者視点でシステムをどう改善していくべきかをさらに考えていく必要があると考える。

今後の展望としては以下が挙げられる。

**談話生成における文構造の制御** 本研究では主に単文の生成を対象としており、文間の関係性を捉える必要のある談話生成はできない。多くの自然言語処理データセットは本研究と同様に短文や単文を対象としている。しかし、実世界に見られる多くのテキスト、例えば書籍や論文などを扱うためにはそれらよりも長く談話構造を捉える必要があるため、それらにも対応できる手法が近年求められて

いる (Shaham et al. 2022). これに伴い、談話生成における文構造の制御も重要性が増すと考えられる.

**より厳密な文構造の制御** 本研究では扱わなかった木構造に基づく厳密な文構造は、言葉の階層性を明示的に扱えるため文生成では有効であると考えられる. そのため、系列化された木構造を編集して文を生成する手法が考えられる. 一方で、今回は段階的に文を編集しているため構造的な情報が暗に学習されている可能性もあり、本当に厳密な文構造情報が必要なのかも含めて検証の余地がある.

**ハードテンプレートの抽出コストの削減** 本研究では構造解析により文中の対句の範囲を同定することでハードテンプレートを抽出した. これにより、対句を含む文ごとに1つのテンプレートが抽出できたが、実際にはより少ないテンプレートでほとんどのテンプレートをカバーできると考えられるため、コスト削減の余地がある. そのため、まず解析前の対句文を木構造化するなどして文構造情報を抽出し、出現頻度が高い文構造を含む文を絞り込む. このプロセスはノンパラメトリックに行える. その次に構造解析によって対句範囲を同定することで大幅に抽出コストを減らすことができる.

**ソフトテンプレートに基づく任意の文構造による制御** 提案手法は生成過程を観測可能なホワイトボックス手法ではあるが、その各過程をユーザーが直接制御できる枠組みはまだない. 提案したソフトテンプレートに基づく手法は、用例を具体的にどのように用いて文を生成するかはモデルに委ねられているため、具体的に制御したい構造を指定することはできない. しかし、ユーザーにとっての使いやすさを考慮すると、そのどの側面で制御するかを容易に指定できるようなシステムが好ましい. そこで、ソフトテンプレートに対する編集操作を学習する際、ユーザーが指定した文構造を削除した場合にペナルティを課すようなオラクル方策を設計する. これにより、特定の文構造は確実に残しつつその他の部分を柔軟に生成可能となるため、ソフトテンプレートを用いた任意の文構造による制御が実現できると考えられる. なお、さらにスコープを広げると言語表現には文構造だけではなくキーワードや長さ、使う単語の難しさなど多くの側面がある. 既存研究では、複数の言語制約を同時に指定可能な制御手法も提案されているが (Gu et al. 2022), 解釈性は考慮されていないため、提案手法

をこれらの側面に対応可能にするという方向性も考えられる。

**ハードテンプレートとソフトテンプレートを組み合わせた手法** 先述の通り、両テンプレートは得意不得意が互いに補完的であり、設定によって適したテンプレートを選択する必要がある。対句のように全ての事例が強い文構造制約を持つ場合は一律にハードテンプレートを選択すれば良いが、他のデータでは事例によってはハードテンプレート化しやすいものとしづらいものが混在すると考えられる。これに対し、ハードテンプレートが容易に作成できるような入力に対してはハードテンプレートを、そうでなければソフトテンプレートを用いるように切り替える仕組みの導入が考えられる。

## 業績リスト

### ジャーナル論文

1. Ayana Niwa, Sho Takase, Naoaki Okazaki. Nearest Neighbor Non-autoregressive Text Generation. Journal of Information Processing, 10 pages (double column), (to appear), April 2023.
2. Ayana Niwa, Naoaki Okazaki, Kohei Wakimoto, Keisuke Nishiguchi, Masataka Mouri. Construction of a Corpus of Rhetorical Devices in Slogans and Structural Analysis of Antitheses. ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), 20(6):1-26, November 2021.

### 国際学会発表

1. Masahiro Kaneko, Sho Takase, Ayana Niwa, and Naoaki Okazaki. Interpretability for Language Learners Using Example-Based Grammatical Error Correction. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL), pages 7176–7187, May 2022.
2. Ayana Niwa, Keisuke Nishiguchi, and Naoaki Okazaki. Predicting Antonyms in Context using BERT. In Proceedings of the 14th International Conference on Natural Language Generation (INLG 2021), pages 48–54, September 2021.

### 招待講演

1. 丹羽 彩奈. 女子中高生のための「数理・情報・A I」最前線 女子中高生向けオンラインセミナー, 2022年11月.
2. 丹羽 彩奈. NeurIPS Meetup Japan 2021, Women in Machine Learning (WiML). 2021年12月.

3. 丹羽 彩奈. 自然言語生成におけるテンプレートの導出と活用. 人工知能学会 合同研究会 2021 第 118 回人工知能基本問題研究会, 2021 年 11 月. (招待講演)
4. 丹羽 彩奈. NeurIPS 2019 における自然言語処理. AI トレンド・トップカンファレンス報告. 人工知能学会 NeurIPS 2021 報告会, 2020 年 4 月.

### 国内口頭発表

1. 丹羽 彩奈, 岡崎 直観. 事前学習済みモデル T5 における近傍分布の有効性の調査. 言語処理学会第 29 回年次大会 (NLP2023), 2023 年 3 月.
2. 谷口 大輔, 脇本 宏平, 丹羽 彩奈, 岡崎直観. 大規模言語モデルにおける文生成方向に関する依存性の検証. 言語処理学会第 29 回年次大会 (NLP2023), 2023 年 3 月.
3. 谷口 大輔, 脇本 宏平, 黒田 和矢, 川本 峻頌, 西口 佳佑, 丹羽 彩奈, 岡崎 直観. 商品レビューと商品特徴を用いた広告文制作支援. 2022 年度人工知能学会全国大会 (JSAI2022), 3Yin2-07, pp. (4 pages), 2022 年 6 月.
4. 石川 遼伍, 丹羽 彩奈, 水木 栄, 岡崎 直観. 疑似訓練データによる格助詞の省略に頑健な係り受け解析. 言語処理学会第 28 回年次大会 (NLP2022), pp. 1808–1813, 2022 年 3 月.
5. 丹羽 彩奈, 高瀬 翔, 岡崎 直観. 近傍の事例を用いた非自己回帰生成. 言語処理学会第 28 回年次大会 (NLP2022), pp. 1306–1311, 2022 年 3 月. (言語処理学会第 28 回年次大会委員特別賞)
6. 丹羽 彩奈, 松田 寛. 個人間の感情理解の揺れを考慮した感情分析に向けた試み. 2021 年度人工知能学会全国大会 (JSAI 2021), 2021 年 6 月.
7. 丹羽 彩奈, 西口 佳佑, 岡崎 直観. 文脈を考慮した対義語穴埋め. 言語処理学会第 27 回年次大会 (NLP 2021), pp. 1702–1707, 2021 年 3 月.
8. 昇 夏海, 平岡 達也, 丹羽 彩奈, 西口 佳佑, 岡崎 直観. 企業情報を考慮したキャッチコピーの自動生成. 言語処理学会第 27 回年次大会 (NLP 2021), pp. 450–454, 2021 年 3 月.

9. 丹羽 彩奈, 西口 佳佑, 岡崎 直観. マスク言語モデルを用いた対句キャッチコピー生成. 第 15 回 NLP 若手の会シンポジウム, 2020 年 8 月.
10. 昇 夏海, 平岡 達也, 丹羽 彩奈, 岡崎 直観, 脇本 宏平, 黒田 和矢, 西口 佳佑. 企業情報を考慮した広告文の自動生成. 第 15 回 NLP 若手の会シンポジウム, 2020 年 8 月.
11. 丹羽 彩奈, 脇本 宏平, 西口 佳佑, 毛利 真崇, 岡崎 直観. 単語の対応関係を利用したスパン候補の絞り込みによるキャッチコピーの対句構造解析. 第 34 回人工知能学会全国大会 (JSAI 2020), 2020 年 6 月.
12. 丹羽 彩奈, 岡崎 直観, 西口 佳佑, 亀山 千尋, 毛利 真崇. 修辞技法を考慮したキャッチコピー自動生成に向けた研究. 第 14 回 NLP 若手の会シンポジウム, p. 63, 2019 年 8 月.
13. 丹羽 彩奈, 岡崎 直観, 西口 佳佑, 亀山 千尋, 毛利 真崇. キャッチコピーの自動生成に向けた分析. 言語処理学会第 25 回年次大会 (NLP 2019), pp. 558–561, 2019 年 3 月.

## 参考文献

- Amina Adadi and Mohammed Berrada (2018) “Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI),” *IEEE Access*, Vol. 6, pp. 52138–52160.
- Nader Akoury, Kalpesh Krishna, and Mohit Iyyer (2019) “Syntactically Supervised Transformers for Faster Neural Machine Translation,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1269–1281, Florence, Italy: Association for Computational Linguistics, July.
- Ondřej Bojar, Christian Buck, Christian Federmann et al. (2014) “Findings of the 2014 Workshop on Statistical Machine Translation,” in *Proceedings of the Ninth Workshop on Statistical Machine Translation*, Baltimore, Maryland, USA, June.
- Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann et al. (2022) “Improving Language Models by Retrieving from Trillions of Tokens,” in *Proceedings of the 39th International Conference on Machine Learning*, Baltimore, Maryland, USA, July.
- Bram Bulte and Arda Tezcan (2019) “Neural Fuzzy Repair: Integrating Fuzzy Matches into Neural Machine Translation,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, July.
- Deng Cai, Yan Wang, Huayang Li, Wai Lam, and Lemao Liu (2021) “Neural Machine Translation with Monolingual Translation Memory,” in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*

and the 11th International Joint Conference on Natural Language Processing, Online, August.

Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei (2018) “Retrieve, Rerank and Rewrite: Soft Template Based Neural Summarization,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, Melbourne, Australia, July.

Thiago Castro Ferreira, Chris van der Lee, Emiel van Miltenburg, and Emiel Krahmer (2019) “Neural data-to-text generation: A comparison between pipeline and end-to-end architectures,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 552–562, Hong Kong, China: Association for Computational Linguistics, November.

Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel (2019) “Controllable Paraphrase Generation with a Syntactic Exemplar,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5972–5984, Florence, Italy: Association for Computational Linguistics, July.

David Chiang (2007) “Hierarchical Phrase-Based Translation,” *Computational Linguistics*, Vol. 33, No. 2, pp. 201–228.

Ilyas Cicekli and Halil Altay Güvenir (2001) “Learning Translation Templates from Bilingual Translation Examples,” *Appl. Intell.*, Vol. 15, pp. 57–76, 07.

James Clarke and Mirella Lapata (2010) “Discourse Constraints for Document Compression,” *Computational Linguistics*, Vol. 36, No. 3, pp. 411–441, September.

Miruna-Adriana Clinciu and Helen Hastie (2019) “A Survey of Explainable AI Terminology,” in *Proceedings of the 1st Workshop on Interactive Natural Language Technology for Explainable Artificial Intelligence (NL4XAI 2019)*, pp. 8–13: Association for Computational Linguistics.

- Sendenkaigi Award Committee (2003 – 2018) *SKAT.2–SKAT.17*: Sendenkaigi Co., Ltd.
- Walter Daelemans, Jakub Zavrel, Peter Berck, and Steven Gillis (1996) “MBT: A Memory-Based Part of Speech Tagger-Generator,” in *Fourth Workshop on Very Large Corpora*, Herstmonceux Castle, Sussex, UK, June.
- Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu (2020) “Plug and Play Language Models: A Simple Approach to Controlled Text Generation,” in *International Conference on Learning Representations*.
- Liming Deng, Long Wang, Binzhu Wang, Jiang Qian, Bojin Zhuang, Shaojun Wang, and Jing Xiao (2021) “Structure Controllable Text Generation.”
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019) “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, Minnesota, June.
- Longxu Dou, Guanghui Qin, Jinpeng Wang, Jin-Ge Yao, and Chin-Yew Lin (2018) “Data2Text Studio: Automated Text Generation from Structured Data,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 13–18, Brussels, Belgium: Association for Computational Linguistics, November.
- Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart (2018) “The Hitchhiker’s Guide to Testing Statistical Significance in Natural Language Processing,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, Melbourne, Australia, July.
- Yuheng Du, Shereen Oraby, Vittorio Perera, Minmin Shen, Anjali Narayan-Chen, Tagyoung Chung, Anushree Venkatesh, and Dilek Hakkani-Tur (2020) “Schema-Guided Natural Language Generation,” in *Proceedings of the 13th*

- International Conference on Natural Language Generation*, pp. 283–295, Dublin, Ireland: Association for Computational Linguistics, December.
- Markus Eberts and Adrian Ulges (2019) “Span-based Joint Entity and Relation Extraction with Transformer Pre-training,” *CoRR*, Vol. abs/1909-07755.
- Giovanna Egidi and Richard J Gerrig (2009) “How valence affects language processing: Negativity bias and mood congruence in narrative comprehension,” *Memory & cognition*, Vol. 37, No. 5, pp. 547–555.
- Roger Evans, Paul Piwek, and Lynne Cahill (2002) “What is NLG?” in *Proceedings of the International Natural Language Generation Conference*, pp. 144–151, Harriman, New York, USA: Association for Computational Linguistics, July.
- Daniel Faggella (2018) “News Organization Leverages AI to Generate Automated Narratives from Big Data,” [Online; accessed 16-December-2022].
- Juliette Faille, Albert Gatt, and Claire Gardent (2020) “The Natural Language Pipeline, Neural Text Generation and Explainability,” in *2nd Workshop on Interactive Natural Language Technology for Explainable Artificial Intelligence*, pp. 16–21, Dublin, Ireland: Association for Computational Linguistics, November.
- Angela Fan, David Grangier, and Michael Auli (2018) “Controllable Abstractive Summarization,” in *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pp. 45–54, Melbourne, Australia: Association for Computational Linguistics, July.
- Jessica Fidler and Yoav Goldberg (2016) “A Neural Network for Coordination Boundary Prediction,” in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 23–32: Association for Computational Linguistics, November.
- (2017) “Controlling Linguistic Style Aspects in Neural Language Generation,” in *Proceedings of the Workshop on Stylistic Variation*, pp. 94–104,

- Copenhagen, Denmark: Association for Computational Linguistics, September.
- Lorenzo Gatti, Chris van der Lee, and Mariët Theune (2018) “Template-based multilingual football reports generation using Wikidata as a knowledge base,” in *Proceedings of the 11th International Conference on Natural Language Generation*, pp. 183–188, Tilburg University, The Netherlands: Association for Computational Linguistics, November.
- Marjan Ghazvininejad, Xing Shi, Jay Priyadarshi, and Kevin Knight (2017) “Hafez: an Interactive Poetry Generation System,” in *Proceedings of ACL 2017, System Demonstrations*, pp. 43–48, Vancouver, Canada: Association for Computational Linguistics, July.
- Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer (2019) “Mask-Predict: Parallel Decoding of Conditional Masked Language Models,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, Hong Kong, China, November.
- E. Goldberg, N. Driedger, and R.I. Kittredge (1994) “Using natural-language processing to produce weather forecasts,” *IEEE Expert*, Vol. 9, No. 2, pp. 45–53.
- Jiatao Gu and Xiang Kong (2021) “Fully Non-autoregressive Neural Machine Translation: Tricks of the Trade,” in *Findings of the Association for Computational Linguistics*, Online, August.
- Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher (2018a) “Non-Autoregressive Neural Machine Translation,” in *Proceedings of the Sixth International Conference on Learning Representations*, Vancouver, Canada.
- Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor OK Li (2018b) “Search engine guided neural machine translation,” in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, USA.

- Jiatao Gu, Changhan Wang, and Junbo Zhao (2019) “Levenshtein Transformer,” in *Proceedings of the Thirty-third Conference on Neural Information Processing Systems*, Vol. 32, Vancouver, Canada.
- Yuxuan Gu, Xiaocheng Feng, Sicheng Ma, Lingyuan Zhang, Heng Gong, and Bing Qin (2022) “A Distributional Lens for Multi-Aspect Controllable Text Generation,” in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Abu Dhabi, United Arab Emirates, December.
- Kelvin Guu, Tatsunori B. Hashimoto, Yonatan Oren, and Percy Liang (2018) “Generating Sentences by Editing Prototypes,” *Transactions of the Association for Computational Linguistics*, Vol. 6, pp. 437–450.
- Hadaytullah Hadaytullah, Khalid Alnajjar, and Hannu (TT) Toivonen (2018) “"Talent, Skill and Support.": A Method for Automatic Creation of Slogans,” in Francois Pachet, Anna Jordanous, and Carlos Leon eds. *Proceedings of the Ninth International Conference on Computational Creativity*, pp. 88–95, United Kingdom: Association for Computational Creativity, 6.
- Kazuo Hara, Masashi Shimbo, Hideharu Okuma, and Yuji Matsumoto (2009) “Coordinate Structure Analysis with Global Structural Constraints and Alignment-Based Local Features,” in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pp. 967–975: Association for Computational Linguistics, August.
- Christian Hempelmann, Victor Raskin, and Katrina Triezenberg (2006) “Computer, Tell Me a Joke ... but Please Make it Funny: Computational Humor with Ontological Semantics.,” Vol. 13, pp. 746–751, 01.
- Yuta Hitomi, Yuya Taguchi, Hideaki Tamori, Ko Kikuta, Jiro Nishitoba, Naoaki Okazaki, Kentaro Inui, and Manabu Okumura (2019) “A Large-Scale Multi-Length Headline Corpus for Analyzing Length-Constrained Headline Generation Model Evaluation,” in *Proceedings of the 12th International Conference*

- on Natural Language Generation*, pp. 333–343: Association for Computational Linguistics.
- Thomas R. Hoffmann (1993) *Realms of meaning: An introduction to semantics*: Longman.
- Bryan Anthony Hong (2008) “Template-Based Pun Extractor and Generator,” Ph.D. dissertation.
- Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing (2017) “Toward Controlled Generation of Text,” in Doina Precup and Yee Whye Teh eds. *Proceedings of the 34th International Conference on Machine Learning*, Vol. 70 of Proceedings of Machine Learning Research, pp. 1587–1596: PMLR, 06–11 Aug.
- Xiao Shi Huang, Felipe Perez, and Maksims Volkovs (2022) “Improving Non-Autoregressive Translation Models Without Distillation,” in *Proceedings of the 10th International Conference on Learning Representations*, Online.
- William J. Hutchins and Harold L. Somers (1991) *An introduction to machine translation*.
- Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer (2018) “Adversarial Example Generation with Syntactically Controlled Paraphrase Networks,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1875–1885, New Orleans, Louisiana: Association for Computational Linguistics, June.
- Ziwei Ji, Nayeon Lee, Rita Frieske et al. (2022) “Survey of Hallucination in Natural Language Generation,” February.
- Sachi Kato, Masayuki Asahara, and Makoto Yamazaki (2018) “Annotation of ‘Word List by Semantic Principles’ Labels for the Balanced Corpus of Contemporary Written Japanese,” in *Proceedings of the 32nd Pacific Asia Conference*

*on Language, Information and Computation*: Association for Computational Linguistics.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis (2021) “Nearest Neighbor Machine Translation,” in *Proceedings of the 9th International Conference on Learning Representations*, Online.

Chloé Kiddon, Luke Zettlemoyer, and Yejin Choi (2016) “Globally Coherent Text Generation with Neural Checklist Models,” in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 329–339, Austin, Texas: Association for Computational Linguistics, November.

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura (2016) “Controlling Output Length in Neural Encoder-Decoders,” in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1328–1338, Austin, Texas: Association for Computational Linguistics, November.

Yoon Kim and Alexander M. Rush (2016) “Sequence-Level Knowledge Distillation,” in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas, November.

Diederik P. Kingma and Jimmy Ba (2015) “Adam: A Method for Stochastic Optimization,” in *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, CA, USA.

Mojtaba Komeili, Kurt Shuster, and Jason Weston (2022) “Internet-Augmented Dialogue Generation,” in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8460–8478, Dublin, Ireland: Association for Computational Linguistics, May.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani (2021) “GeDi: Generative Discriminator Guided Sequence Generation,” in *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 4929–4952,

Punta Cana, Dominican Republic: Association for Computational Linguistics, November.

Karen Kukich (1983) “Design of a Knowledge-Based Report Generator,” in *21st Annual Meeting of the Association for Computational Linguistics*, pp. 145–150, Cambridge, Massachusetts, USA: Association for Computational Linguistics, June.

——— (1992) “Techniques for Automatically Correcting Words in Text,” *ACM Comput. Surv.*, Vol. 24, No. 4, p. 377–439, December.

Sadao Kurohashi and Makoto Nagao (1994) “A syntactic analysis method of long Japanese sentences based on the detection of conjunctive structures,” *Computational Linguistics*, Vol. 20, No. 4, pp. 507–534.

Rémi Lebret, David Grangier, and Michael Auli (2016) “Neural Text Generation from Structured Data with Application to the Biography Domain,” in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas, November.

Chris van der Lee, Emiel Kraemer, and Sander Wubben (2017) “PASS: A Dutch data-to-text system for soccer, targeted towards specific audiences,” in *Proceedings of the 10th International Conference on Natural Language Generation*, pp. 95–104, Santiago de Compostela, Spain: Association for Computational Linguistics, September.

Jason Lee, Elman Mansimov, and Kyunghyun Cho (2018) “Deterministic Non-Autoregressive Neural Sequence Modeling by Iterative Refinement,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, October–November.

Jason Lee, Raphael Shu, and Kyunghyun Cho (2020) “Iterative Refinement in the Continuous Space for Non-Autoregressive Neural Machine Translation,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, Online, November.

Geoffrey Leech (1976) *Semantics*: Penguin Books.

Piyawat Lertvittayakumjorn and Francesca Toni (2019) “Human-grounded Evaluations of Explanation Methods for Text Classification,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5195–5205, Hong Kong, China: Association for Computational Linguistics, November.

Vladimir Iosifovich Levenshtein (1966) “Binary codes capable of correcting deletions, insertions, and reversals,” in *Soviet physics doklady*, Vol. 10, pp. 707–710, Soviet Union.

Boyang Li, Stephen Lee-Urban, George Johnston, and Mark Riedl (2013) “Story Generation with Crowdsourced Plot Graphs,” *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 27, No. 1, pp. 598–604, Jun.

Liangyou Li, Andy Way, and Qun Liu (2016) “Phrase-Level Combination of SMT and TM Using Constrained Word Lattice,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, August.

Piji Li, Haisong Zhang, Xiaojiang Liu, and Shuming Shi (2020) “Rigid Formats Controlled Text Generation,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 742–751, Online: Association for Computational Linguistics, July.

Wei Li, Wenhao Wu, Moye Chen, Jiachen Liu, Xinyan Xiao, and Hua Wu (2022) “Faithfulness in Natural Language Generation: A Systematic Survey of Analysis, Evaluation and Optimization Methods.”

Ao Liu, Congjian Luo, and Naoaki Okazaki (2021) “Improving Logical-Level Natural Language Generation with Topic-Conditioned Data Augmentation and Logical Form Generation.”

- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig (2021) “Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing.”
- Bingfeng Luo, Zuo Bai, Kunfeng Lai, and Jianping Shen (2020) “Make Templates Smarter: A Template Based Data2Text System Powered by Text Stitch Model,” in *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1057–1062, Online: Association for Computational Linguistics, November.
- Sourab Mangrulkar, Suhani Shrivastava, Veena Thenkanidiyoor, and Dileep Aroor Dinesh (2018) “A Context-aware Convolutional Natural Language Generation model for Dialogue Systems,” in *Proceedings of the 19th Annual SIG-dial Meeting on Discourse and Dialogue*, pp. 191–200, Melbourne, Australia: Association for Computational Linguistics, July.
- Elena Manishina (2016) “Data-driven natural language generation using statistical machine translation and discriminative learning,” Ph.D. dissertation.
- Louis Martin, Éric de la Clergerie, Benoît Sagot, and Antoine Bordes (2020) “Controllable Sentence Simplification,” in *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 4689–4698, Marseille, France: European Language Resources Association, May.
- Alexandra Mayn and Kees van Deemter (2020) “Towards Generating Effective Explanations of Logical Formulas: Challenges and Strategies,” in *2nd Workshop on Interactive Natural Language Technology for Explainable Artificial Intelligence*, pp. 39–43, Dublin, Ireland: Association for Computational Linguistics, November.
- David D. McDonald (1993) “Issues in the choice of a source for Natural Language Generation,” *Computational Linguistics*, Vol. 19, No. 1, pp. 191–197.
- Kathleen R. McKeown (1983) “Paraphrasing Questions Using Given and new information,” *American Journal of Computational Linguistics*, Vol. 9, No. 1, pp. 1–10.

- Makoto Nagao (1984) “A Framework of a Mechanical Translation between Japanese and English by Analogy Principle,” in *Proceedings of the International NATO Symposium on Artificial and Human Intelligence*.
- Tomoko Ohta, Yuka Tateisi, and Jin-Dong Kim (2002) “The GENIA Corpus: An Annotated Research Abstract Corpus in Molecular Biology Domain,” in *Proceedings of the Second International Conference on Human Language Technology Research, HLT '02*, p. 82–86, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Ethel Ong, Bryan Anthony Hong, and Vince Andrew Nuñez (2008) “Automatically Extracting Templates from Examples for NLP Tasks,” in *Proceedings of the 22nd Pacific Asia Conference on Language, Information and Computation*, pp. 452–459, The University of the Philippines Visayas Cebu College, Cebu City, Philippines: De La Salle University, Manila, Philippines, November.
- OpenAI (2022) “ChatGPT: Optimizing Language Models for Dialogue,” [Online; accessed 16-December-2022].
- Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli (2019) “fairseq: A Fast, Extensible Toolkit for Sequence Modeling,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, Minneapolis, Minnesota, June.
- Tatsuro Oya, Yashar Mehdad, Giuseppe Carenini, and Raymond Ng (2014) “A Template-based Abstractive Meeting Summarization: Leveraging Summary and Source Text Relationships,” in *Proceedings of the 8th International Natural Language Generation Conference (INLG)*, pp. 45–53, Philadelphia, Pennsylvania, U.S.A.: Association for Computational Linguistics, June.
- Gözde Özbal, Daniele Pighin, and Carlo Strapparava (2013) “BRAINSUP: Brainstorming Support for Creative Sentence Generation,” in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*

(*Volume 1: Long Papers*), pp. 1446–1455: Association for Computational Linguistics, August.

Gaurav Pandey, Danish Contractor, Vineet Kumar, and Sachindra Joshi (2018) “Exemplar Encoder-Decoder for Neural Conversation Generation,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1329–1338, Melbourne, Australia: Association for Computational Linguistics, July.

Hao Peng, Ankur Parikh, Manaal Faruqui, Bhuwan Dhingra, and Dipanjan Das (2019) “Text Generation with Exemplar-based Adaptive Decoding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, Minnesota, June.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer (2018) “Deep Contextualized Word Representations,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2227–2237, New Orleans, Louisiana: Association for Computational Linguistics, June.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller (2019) “Language Models as Knowledge Bases?” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2463–2473, Hong Kong, China: Association for Computational Linguistics, November.

Maja Popović (2015) “chrF: character n-gram F-score for automatic MT evaluation,” in *Proceedings of the Tenth Workshop on Statistical Machine Translation*, Lisbon, Portugal, September.

Matt Post (2018) “A Call for Clarity in Reporting BLEU Scores,” in *Proceedings*

- of the *Third Conference on Machine Translation: Research Papers*, pp. 186–191, Brussels, Belgium: Association for Computational Linguistics, October.
- Jing Qian, Li Dong, Yelong Shen, Furu Wei, and Weizhu Chen (2022) “Controllable Natural Language Generation with Contrastive Prefixes,” in *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2912–2924, Dublin, Ireland: Association for Computational Linguistics, May.
- Lihua Qian, Hao Zhou, Yu Bao, Mingxuan Wang, Lin Qiu, Weinan Zhang, Yong Yu, and Lei Li (2021) “Glancing Transformer for Non-Autoregressive Neural Machine Translation,” in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, Online, August.
- Guanghui Qin, Jin-Ge Yao, Xuening Wang, Jinpeng Wang, and Chin-Yew Lin (2018) “Learning Latent Semantic Annotations for Grounding Natural Language to Structured Data,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3761–3771, Brussels, Belgium: Association for Computational Linguistics, October–November.
- Colin Raffel, Noam Shazeer, Adam Roberts et al. (2020) “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer,” *Journal of Machine Learning Research*, Vol. 21, No. 140, pp. 1–67.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang (2016) “SQuAD: 100,000+ Questions for Machine Comprehension of Text,” in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas: Association for Computational Linguistics, November.
- Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie (2020) “COMET: A Neural Framework for MT Evaluation,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, Online, November.
- Nils Reimers and Iryna Gurevych (2019) “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks,” in *Proceedings of the 2019 Conference*

*on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, Hong Kong, China, November.

Ehud Reiter and Robert Dale (2000) *Building Natural Language Generation Systems*, Studies in Natural Language Processing: Cambridge University Press.

Rico Sennrich, Barry Haddow, and Alexandra Birch (2016a) “Controlling Politeness in Neural Machine Translation via Side Constraints,” in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 35–40, San Diego, California: Association for Computational Linguistics, June.

——— (2016b) “Neural Machine Translation of Rare Words with Subword Units,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725, Berlin, Germany: Association for Computational Linguistics, August.

Uri Shaham, Elad Segal, Maor Ivgi et al. (2022) “SCROLLS: Standardized Comparison Over Long Language Sequences.”

Xiaoyu Shen, Jun Suzuki, Kentaro Inui, Hui Su, Dietrich Klakow, and Satoshi Sekine (2019) “Select and Attend: Towards Controllable Content Selection in Text Generation,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 579–590, Hong Kong, China: Association for Computational Linguistics, November.

Opim Salim Sitompul, Erna Budhiarti Nababan, Dedy Arisandi, Indra Aulia, and Hengky Wijaya (2020) “Template-Based Natural Language Generation in Interpreting Laboratory Blood Test.”

Yiping Song, Cheng-Te Li, Jian-Yun Nie, Ming Zhang, Dongyan Zhao, and Rui Yan (2018) “An Ensemble of Retrieval-Based and Generation-Based Human-Computer Conversation Systems,” in *Proceedings of the Twenty-Seventh In-*

*ternational Joint Conference on Artificial Intelligence, IJCAI-18*, pp. 4382–4388: International Joint Conferences on Artificial Intelligence Organization, 7.

Felix Stahlberg, Eva Hasler, Aurelien Waite, and Bill Byrne (2016) “Syntactically Guided Neural Machine Translation,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 299–305, Berlin, Germany: Association for Computational Linguistics, August.

Mitchell Stern, William Chan, Jamie Kiros, and Jakob Uszkoreit (2019) “Insertion Transformer: Flexible Sequence Generation via Insertion Operations,” in *Proceedings of the 36th International Conference on Machine Learning*, Long Beach, California, USA.

Zhiqing Sun, Zhuohan Li, Haoqing Wang, Di He, Zi Lin, and Zhihong Deng (2019) “Fast Structured Decoding for Sequence Models,” in *Proceedings of the Thirty-third Conference on Neural Information Processing Systems*, Vol. 32, Vancouver, Canada.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le (2014) “Sequence to Sequence Learning with Neural Networks,” in *Proceedings of the Twenty-eighth Conference on Neural Information Processing Systems*, Vol. 27, Montréal, Canada.

Hiroki Teranishi, Hiroyuki Shindo, and Yuji Matsumoto (2017) “Coordination boundary identification with similarity and replaceability,” in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 264–272: Asian Federation of Natural Language Processing.

M. Theune, E. Klabbers, J.R. de Pijper, E.J. Krahmer, and J. Odijk (2001a) “From Data to Speech: A Generic Approach,” *Natural Language Engineering*, Vol. 7, No. 1, pp. 47–86, Pagination: 40.

- M. Theune, E. Klabbers, J. R. De Pijper, E. Krahmer, and J. Odijk (2001b) “From Data to Speech: A General Approach,” *Nat. Lang. Eng.*, Vol. 7, No. 1, p. 47–86, March.
- Arseny Tolmachev, Daisuke Kawahara, and Sadao Kurohashi (2018) “Juman++: A Morphological Analysis Toolkit for Scriptio Continua,” in *Proc. of the EMNLP*, pp. 54–59, Brussels, Belgium, November.
- Polona Tomasic, Martin Znidarsic, and Gregor Papa (2014) “Implementation of a slogan generator,” in *Proceedings of 5th International Conference on Computational Creativity*, pp. 340–343: Association for Computational Creativity.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017) “Attention is all you need,” in *Proceedings of the Thirty-first Conference on Neural Information Processing Systems*, Long Beach, California, USA.
- Di Wang, Nebojsa Jojic, Chris Brockett, and Eric Nyberg (2017) “Steering Output Style and Topic in Neural Response Generation,” in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2140–2150, Copenhagen, Denmark: Association for Computational Linguistics, September.
- Kento Watanabe, Yuichiroh Matsubayashi, Satoru Fukayama, Masataka Goto, Kentaro Inui, and Tomoyasu Nakano (2018) “A Melody-Conditioned Lyrics Language Model,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 163–172, New Orleans, Louisiana: Association for Computational Linguistics, June.
- Jason D. Williams, Pascal Poupart, and Steve Young (2005) “Partially Observable Markov Decision Processes with Continuous Observations for Dialogue Management,” in *Proceedings of the 6th SIGdial Workshop on Discourse and Dialogue*, pp. 25–34, Lisbon, Portugal: Special Interest Group on Discourse and Dialogue (SIGdial), September.

- Sam Wiseman, Stuart Shieber, and Alexander Rush (2017) “Challenges in Data-to-Document Generation,” in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2253–2263, Copenhagen, Denmark: Association for Computational Linguistics, September.
- (2018) “Learning Neural Templates for Text Generation,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3174–3187, Brussels, Belgium: Association for Computational Linguistics, October–November.
- Sam Wiseman, Arturs Backurs, and Karl Stratos (2021) “Data-to-text Generation by Splicing Together Nearest Neighbors,” in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Online and Punta Cana, Dominican Republic, November.
- Jitao Xu, Josep Crego, and Jean Senellart (2020) “Boosting Neural Machine Translation with Similar Translations,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, July.
- Weijia Xu and Marine Carpuat (2021) “EDITOR: An Edit-Based Transformer with Repositioning for Neural Machine Translation with Soft Lexical Constraints,” *Transactions of the Association for Computational Linguistics*, Vol. 9.
- Kenji Yamada and Kevin Knight (2001) “A Syntax-based Statistical Translation Model,” in *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pp. 523–530, Toulouse, France: Association for Computational Linguistics, July.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le (2019) “XLNet: Generalized Autoregressive Pretraining for Language Understanding,” in *Advances in Neural Information Processing Systems*, Vol. 32, pp. 5753–5763: Curran Associates, Inc.

Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song (2022) “A Survey of Controllable Text Generation using Transformer-based Pre-trained Language Models.”

Xin Zheng, Zhirui Zhang, Junliang Guo, Shujian Huang, Boxing Chen, Weihua Luo, and Jiajun Chen (2021) “Adaptive Nearest Neighbor Machine Translation,” in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 368–374, Online: Association for Computational Linguistics, August.

三省堂編修所（編）（2017）『反対語対立語辞典』，三省堂。

山根宏彰・萩原将文（2014）「ウェブにおけるユーザの嗜好を反映させたキャッチコピー自動生成システム」，『日本感性工学会論文誌』，第13巻，第4号，493–500頁。

柴田知秀・河原大輔・黒橋禎夫（2019）「BERTによる日本語構文解析の精度向上」，『言語処理学会第25回年次大会(NLP2019)』，P205–208 (4 pages) 頁，名古屋大学（愛知県），3月。

青田光章・秋山晶・東秀紀他（2007）『最新約コピーバイブル』，株式会社宣伝会議。

谷山雅計（2007）『広告コピーってこう書くんだ!読本』，株式会社宣伝会議。

仲畑貴志（2018）『ホントのことを言うと、よく、しかられる。勝つコピーのぜんぶ』，株式会社宣伝会議。

幅野裕貴・浦谷則好（2014）「あらすじとレビューを用いた映画のキャッチコピー作成支援手法の提案」，『言語処理学会第20回年次大会』，864–867頁。