特別研究報告書

機械翻訳サービスのための制約に基づく訳語選択

指導教員　　石田 亨 教授

京都大学工学部情報学科

松野 淳

平成22年2月3日

機械翻訳サービスのための制約に基づく訳語選択

松野 淳

内容梗概

インターネット技術の普及により，世界的にインターネット人口は増加の一途をたどっており，エンドユーザレベルでのグローバル化がますます進むものと思われる．エンドユーザには様々な言語や文化をもつ人々が存在しているため，そのような言語・文化の壁を越えた異文化コラボレーションのためには，ある言語を別の言語に機械的に変換する機械翻訳は有用なサービスである．しかし，全ての言語の組み合わせに対して，機械翻訳を開発することは現実的ではないため，英語をハブとした機械翻訳の連携を考える必要がある．英語と非英語言語間の言語資源は，非英語間の言語資源に比べて量，質ともに優れているため，非英語間翻訳においては，英語をハブとした機械翻訳の連携を考えることは自然なことである．そのような英語をハブとした機械翻訳連携は，直接の機械翻訳が存在しないまたはその機械翻訳に十分な品質を得られない言語間において，いっそう質の高い他言語情報を得ることを可能としてくれる．

しかしながら，機械翻訳では，言語が持つ語の多義性と機械翻訳に文間文脈を制御するための機構が存在しないことが原因となって，訳語選択の非一貫性問題が起こる．訳語選択の非一貫性問題とは，同じ意味を持つ同じ語の訳語が周囲の語によって変化してしまう問題である．単独の機械翻訳において，このような問題が生じるのであれば，機械翻訳を連携させた場合には，さらに大きな問題となってしまう．この問題を解決するためには，1 文に対する機械翻訳の翻訳品質を向上させるだけでは不可能である．なぜなら，複数文つまり文間文脈が関わる問題だからである．

本研究では，機械翻訳における訳語選択問題を制約最適化問題と捉えることで，文間文脈を考慮に入れた上での一貫した訳語選択を目指す．本制約最適化問題では，翻訳元文書中の各名詞単語に対応して各変数があり，各変数の翻訳先文書中の訳語を要素として含む有限集合として各変域がある．また，翻訳元文書中で同文共起している変数間には，“ それらの変数の訳語間には意味的関連性がある ”という制約があり，制約に対応して，それらの変数の訳語間の意味的関連性を定量値として出力するコスト関数がある．最終的に，全てのコスト関数の値の総和を最小にするような変数の訳語の組み合わせが最適解である．

複数文で現れる変数は，それぞれの文で共起している変数との間に制約があるため，単語が現れる全ての文を考慮に入れた訳語選択が行われる．また，最適化問題であるため，各単語に対応した単一の訳語選択が行われ，一貫した訳語選択である．つまり，本制約最適化問題を解くことは，文間文脈を考慮に入れた一貫した訳語選択が行われることを意味する．

さらに，本制約最適化問題を機械翻訳連携の訳語選択問題にも適用できるように拡張した．2つの機械翻訳の訳語選択問題を解く制約最適化問題をそれぞれCOP1,COP2とすれば，本来であればCOP1が解かれてしまった後でなければCOP2の問題を定義することができないため，COP1とCOP2を別々の問題として形式化しなければならない．しかし，それでは各機械翻訳で最適な訳語選択が行われてしまい，連携した機械翻訳全体としての最適な訳語選択が行われない．そこで本研究では，条件付き制約充足問題の定義を基に，COP1とCOP2を1つの制約最適化問題として定式化した．この問題を解くことは，連携した機械翻訳全体としての最適な訳語選択が行われることを意味する．

機械翻訳における訳語選択問題，機械翻訳連携の訳語選択問題をそれぞれ解くためのアルゴリズムを示し，機械翻訳において一貫した訳語選択を行うためのシステムを実装した．翻訳元文書で複数回出現していた名詞単語のうち13%の名詞単語において訳語選択の非一貫性問題が生じていたが，訳語選択システムにより，それらの名詞単語に対する一貫した訳語選択を可能とした．また，本システムにより選択された一貫した訳語の80% が文書文脈に適した訳語であった．この結果から，文間文脈を考慮に入れた一貫した訳語選択を行うことにより，文書を対象とした機械翻訳の翻訳品質を向上できる可能性があることを示した．

Constraint-Based Word Selection for Machine Translation Service

Jun MATSUNO

Abstract

The internet population is growing worldwide by the spread of internet technology and it seems that globalization on the level of end users is progressing increasingly. Machine translations translating a language into another language are useful services to realize intercultural collaboration transcending language and cultural barriers because end users have different languages and cultures. However, thinking about pivot translation via English is required because it is not realistic to develop machine translations for a combination of all languages. Thinking about pivot translation via English has a point in translation between non-English languages because language resources between English and a non-English language are superior to them between non-English languages. Pivot translation via English enable more quality information represented in other languages to be got in cases where there are not direct machine translations between non-English languages and quality in them is not enough.

However, the inconsistency of word selection occurs in machine translations due to polysemy of words and nonexistence of mechanism for controlling the context between sentences. This is the problem that the translated word of the same source word containing the same meaning vary depending words around it. If this problem occurs in a single machine translation, it become more serious in pivot translation. If translation quality is increased in machine translations for a sentence, it is impossible to solve this problem. It is because that many sentences, that is the context between sentences, are involved in this problem.

In this paper, we propose the approach for the consistent word selection considering the context between sentences by regarding the word selection problem as the constraint optimization problem. In this constraint optimization problem, there is each variable corresponding to each noun word in the source document and there is each domain as a finite set including translated words of each variable in the target document. Also, there are constraints representing "there is semantic relatedness between translated words of variables" between

them co-occurring in the same sentence of the source document and there are cost functions which output a quantitative value representing semantic relatedness between translated words of variables corresponding to each constraint. As a result, the optimal solution is an assignment of a translated word to every variable with the minimum total cost. The translated word is selected considering all sentences in which a word is contained because a variable contained in multiple sentences have constraints between variables co-occurring with it in each sentence. Also, a single translated word is selected to each noun word and the consistent word selection is performed because of the optimization problem. That is, solving this constraint optimization problem means that consistent words are selected considering the context between sentences.

In addition, we extend this constraint optimization problem that it can be applied to the word selection problem in pivot translation. If each constraint optimization problem solving word selection problems in two machine translation is regarded as COP1 and COP2 respectively, COP1 and COP2 have to be formalized independently because COP2 can be formalized only when COP1 is solved. However, the optimal word selection is performed in each machine translation and the optimal word selection is not performed in whole combined machine translations . Therefore, we formalize COP1 and COP2 as a constraint optimization problem based on the definition of the conditional constraint satisfaction in this paper. Solving this problem means that the optimal word selection is performed in whole combined machine translations.

We present algorithms for solving the word selection problem in the machine translation and it in pivot translation respectively and we implement the system performing the consistent word selection in the machine translation. The inconsistency of word selection occurred in 13% of noun words contained in multiple sentences but, it became possible that the consistent word selection was performed for them using the word selection system. Also, 80% of consistent translated words selected by this system were they which were suitable for the context of the document. This result indicates that performing the consistent word selection considering the context between sentences can increase translation quality in machine translations aimed at the document.

機械翻訳サービスのための制約に基づく訳語選択

目次

第1章　はじめに

インターネット技術の普及により，世界的にインターネット人口は増加の一途をたどっており，エンドユーザレベルでのグローバル化がますます進むものと思われる．エンドユーザレベルでのグローバル化は，情報のグローバル化を意味し，インターネット上は多様な言語で表記された膨大な情報で溢れかえっている．そのような状況下では，ある言語を別の言語に翻訳する機械翻訳は有用なサービスである．ここで，ますます多くの機械翻訳が存在するようになれば，エンドユーザが気軽に機械翻訳を利用できるようになるのだが，n 言語の全ての言語の組み合わせに対して n(n-1)/2 個の直接的な機械翻訳を開発することは現実的ではないため，機械翻訳を連携させることが必要となってくる．

機械翻訳を連携させる際には，英語をハブ言語とした機械翻訳連携が重要である．その理由としては，英語は多くの良質な言語資源を得ることができること，世界的な公用語であるため，非英語と英語間の機械翻訳は数多く存在しその開発も盛んであることが挙げられる．機械翻訳を連携させることによって，直接的な機械翻訳が存在しないまたは，その翻訳品質が十分なものでないといった理由により理解できなかった他言語情報も，理解できるようになるのである．

言語グリッドプロジェクト [1] では，言語・文化の壁を乗り越えた異文化コラボレーションのために，言語サービスの利便性と有用性を高めるための活動を行っている．言語グリッドプロジェクトの目的の一つは，標準言語を扱う既存の言語サービスを連結することである．この目的を達成することにより，世界中の多様な言語を用いるユーザが，他言語のサービスを母国語で利用できるという恩恵を受けることができる．機械翻訳サービスを連携させている機械翻訳連携もこの目的にかなっており，機械翻訳が存在しない言語間での機械翻訳によるインタラクションを可能とする．

しかしながら，機械翻訳を介したコミュニケーションでは，訳語の非一貫性が問題となり，同一物を表す表現を相手が理解することを妨げること [2] が知られているが，この訳語の非一貫性は，翻訳対象として文書を取り扱う機械翻訳においても生じる問題である．単独の機械翻訳でこのような問題が生じるのであれば，機械翻訳を連携させた場合には，もっと大きな問題となってしまう．この問題を解決するために，R. Tanaka et.al [3] は，翻訳元単語と機械翻訳により選択された訳語からなる単語の組を文間文脈情報として伝播させる手法を提

案している．しかし，単語の訳語選択が，その単語が現れている文のうちの 1 文のみに依存しており，文内文脈を文間文脈として伝播してしまっている点が問題として挙げられる．

本研究では，翻訳の対象として文書を扱うとし，文間文脈を考慮に入れた上での一貫した訳語選択を目指す．そのために，機械翻訳における訳語選択問題を制約最適化問題として捉える．本制約最適化問題では，翻訳元文書中の各名詞単語に対応して各変数があり，各変数の翻訳先文書中の訳語を要素として含む有限集合として各変域がある．また，翻訳元文書中で同文共起している変数間には,“ それらの変数の訳語間には意味的関連性がある ”という制約があり，制約に対応して，それらの変数の訳語間の意味的関連性を定量値として出力するコスト関数がある．最終的に，全てのコスト関数の値の総和を最小にするような変数の訳語の組み合わせが最適解である．制約最適化問題では，複数文で現れる単語に対応する変数は，それぞれの文で共起している単語に対応する変数との間に制約があるため，単語が現れる全ての文を考慮に入れた訳語選択が行われる．また，最適化問題であるため，各単語に対応した単一の訳語選択が行われ，一貫した訳語選択である．つまり，本制約最適化問題を解くことは，文間文脈を考慮に入れた一貫した訳語選択が行われることを意味する．

さらに，制約最適化問題を機械翻訳連携の訳語選択問題にも適用できるように拡張した．2 つの機械翻訳の訳語選択問題を解く制約最適化問題をそれぞれ COP1,COP2 とすれば，本来であれば COP1 が解かれてしまった後でなければ COP2 の問題を定義することができないため，COP1 と COP2 を別々の問題として形式化しなければならない．しかし，それでは各機械翻訳で最適な訳語選択が行われてしまい，機械翻訳連携全体としての最適な訳語選択が行われないことになってしまう．そこで本研究では，条件付き制約充足問題を基に，COP1 と COP2 を 1 つの制約最適化問題として定式化した．この問題を解くことは，機械翻訳連携全体としての最適な訳語選択が行われることを意味する．

第 2 章　背景

2.1　機械翻訳の問題点

コミュニケーションを目的とした機械翻訳の訳語選択において生じる問題には，非一貫性と非対称性の 2 つの問題が存在する [2]．このうち，文書を対象とし

た機械翻訳においても生じる問題は，非一貫性問題である．非一貫性問題とは，同じ語の訳語が周囲の語によって変化する問題である．図 1 では，単語“ paper ”は文章によって“ 紙 ”と訳される場合と“ 論文 ”と訳される場合があり，一貫した訳語選択が行われていない．同様に図 2 では，単語“ subject ”は文章によって，“ 主語 ”および“ 件名 ”と訳されている．このような問題が生じることにより，同一物を指す同じ単語が異なる意味として理解される恐れがあり，文書の理解を妨げる要因となる．ここで，非一貫性問題は，図 1 でも示したように，単独の機械翻訳においても生じる問題であるが，複数の機械翻訳を連携した場合には，翻訳元単語の訳語が一貫して選択されることはさらに困難となる．

機械翻訳連携においては，訳語選択の非遷移性問題が存在する [3]．非遷移性問題とは，翻訳の途中で訳語の意味が変化してしまう問題である．図 3 では，翻訳元文の単語“ 欠点 ”の訳語として，英単語“ fault ”を経て，“ 責任 ”という意味のドイツ語“ Schuld ”が選択されている．これは，英単語“ fault ”に“ 欠点 ”，“ 責任 ”などの意味があるために起こる問題である．

翻訳元文(英)：The paper is excellent. I want to know about the author of the paper.
⇒翻訳先文(日)：紙優れています。私は、論文の著者を知りたい。

図 1: 英日機械翻訳における“ paper ”の訳語選択の非一貫性の例

原文(英)：A sentence usually contains a subject and a predicate. However, Japanese often omit the subject.
⇒訳文(日)：文は通常，主語と述語が含まれます。しかし，多くの場合，件名を省略すると日本語。

図 2: 英日機械翻訳における“ subject ”の訳語選択の非一貫性の例

翻訳元文(日)：彼女の欠点は大きな問題だ.
⇒英文(英)：Her fault is a big problem.
⇒翻訳先文(独)：Ihre Schuld ist ein grosses Problem.

図 3: 日英独機械翻訳連携における訳語選択の非遷移性の例

2.2　文脈を用いた機械翻訳

機械翻訳における非一貫性問題を解決するために，R. Tanaka et.al [3] は，翻訳元単語と訳語の組を文間文脈として機械翻訳サービス間で伝播する手法を提案している．各機械翻訳サービスは，通常受け取った入力文のみを基に訳文を生成するが，訳文生成時の文脈を次のサービスに伝播し，次のサービスは受け取った文脈に従って訳語選択を行う．このようにすることで，文間文脈を考慮に入れた一貫した訳語選択が可能となる．図 1 の例を用いて説明すれば，1 文目で“ paper ”の訳語として“ 紙 ”が選択されたので，(paper, 紙) からなる単語の組を文間文脈とし次の翻訳サービスに伝播し，2 文目の翻訳の際には文間文脈に従った訳語選択，つまり“ paper ”の訳語として“ 紙 ”を選択することとなる．

しかしながら，この手法にはまず初めに選択された訳語が以後選択され続けてしまうという問題点が存在する．また，その訳語は，1 文のみ即ち文内文脈を基に決められた訳語であり，その文脈を文間文脈として伝播することは好ましくないと思われる．提案されている手法が，機械翻訳を用いたコミュニケーション支援を目的としており，即時性が求められているということもあるが，初めから全ての文が把握できる文書翻訳に対しては，適当な手法であるとは考えられない．文書翻訳においては，文書翻訳文全体を考慮に入れた上での一貫した訳語選択が望まれる．もし，各機械翻訳において，複数の訳語が選択されている単語に対して文脈に適した一貫した訳語選択を行うことができれば，各機械翻訳間で文脈に適さない意味を持つ訳語が選択されることを防ぐことができるため，訳語選択の非一貫性問題を解決するだけではなく，非遷移性問題をも解決することにつながる．

第 3 章　制約最適化に基づく訳語選択の定式化

3.1　機械翻訳における訳語選択

一貫しない単語の訳語を，一貫している単語の訳語に基づいて一意に決定するために，本研究では，訳語選択問題を制約最適化問題として定式化する．翻訳元文書中の各名詞単語に対応して各変数があり，各変数がとりうる全ての訳語を要素として含む有限集合として各変域がある．また，同文共起する翻訳元単語の訳語間に制約を課し，最も制約を満たすような訳語の組をその文内で選択されるべき訳語の組とする．このようにすることで，非一貫性問題が生じる

単語，つまり複数文で現れる単語の訳語選択は，それぞれの文で共起している全ての単語の訳語選択の影響を受ける．また，制約最適化問題では，各変数の値が一意に決定されるため，制約最適化問題に基づいて訳語選択問題を解くことは，文間文脈に基づいた一貫した訳語選択が行われることを意味する．

3.1.1 制約最適化問題

まず，制約充足問題の定義 [4] を述べる．制約充足問題は，変数集合 $X = \{x_1, \ldots, x_n\}$,変域集合 $D = \{D_1, \ldots, D_n\}$,制約集合 $C = \{R_1, \ldots, R_m\}$ からなる問題である．変域 D_iは変数 x_i に割り当てられうる値からなる有限集合である．制約 R は，R の範囲である変数の部分集合 $var(R) \in X$ に対する関係を表す．R に含まれるタプルは，R の範囲に含まれる変数に対して許容される値の組み合わせである．解は，全ての制約が満たされるような X に含まれる全ての変数に対する値の組み合わせである．制約最適化問題の定義 [4] では，制約充足問題における制約が，タプル間での優先度を示すコスト関数として表わされる．コスト関数 f はその範囲 $var(f)$ に対して定義され，各タプルに対して非負のコストを返す．ここで，一般性を失わないように，全ての制約 (i.e. soft，hard) をコスト関数として表現するものとする．hard 制約の場合は，許容されるタプルに 0 を許容されないタプルには ∞ を割り当てる 2 値の関数である．このとき，問題を重み付き制約充足問題 [5] と考えれば，目的関数は全ての制約 $C = \{f_1, \ldots, f_m\}$ の和として以下のように表わされる．

$$f^*(\mathtt{X}) = \sum_{j=1}^{m} f_j(\mathtt{X})$$

全ての変数に対する値の割り当てに対するコストが，制約集合により得られるコストの総和となる.ここで,$f_j(X)$ は,X を射影した $var(f_j)$ に対するコストを表すとする. この問題の目的は，コストを最小とするような全ての変数に対する値の割り当てを見つけることである．制約最適化問題の定義を図 4 にまとめる．

3.1.2 制約最適化に基づく訳語選択問題の定式化

3.1.1 で述べた定義を基に，機械翻訳における訳語選択問題を制約最適化問題として定式化する．ここでは，言語 L_1 から言語 L_2 への機械翻訳を考える．まず，n 個の変数 $x_1, \ldots, x_n$ があり，それぞれに対応する変域を $D_1, \ldots, D_n$ とする．各変数は L_1 文書中に現れる名詞単語のうち，L_2 文書中で訳語が名詞単語として得られる各名詞単語に対応して存在しており，n はそのような名詞単語の総数である．D_i は，L_1 単語 x_i の L_2 文書中での訳語を要素として含む有限

変数集合:X=$\{x_1, \ldots, x_n\}$
変域集合:D=$\{D_1, \ldots, D_n\}$
制約集合:C=$\{f_1, \ldots, f_n\}$
目的関数:f^*(X)=$\sum_{j=1}^{m} f_j$(X)
(ここで，f_j(X) は X の射影であり f_j の範囲である var(f_j) に対するコストを表すとする)

最適解:minf^*(X) とする全ての変数に対する値の組み合わせ

図 4: 制約最適化問題の定義

集合である．各変数の値は，対応する変域集合に含まれる要素である訳語から選択される．L_1 文書中で x_i と x_j ($1 \leq i < j \leq n$) が，同文共起しているならば x_i, x_j 間に制約 f_{ij} が存在し，その制約を“ x_i の訳語と x_j の訳語は意味的関連性がある ”として表し，$var(f_{ij}) = \{\ x_i\ ,\ x_j\ \}$ とする．ここで， Gabrilovich et.al [6] が提案している Wikipedia を用いた semantic relatedness の計算手法を用いる．具体的には，x_i の訳語が言語 L_2 の Wikipedia の各記事に出現した回数から，tf/idf score を用いて x_i の訳語と各記事に対する関連の強さ (重み) を決定し，x_i の訳語に対して重み付けされた記事のリスト l_i を得る．そして，x_i の訳語と x_j の訳語のリスト l_i と l_j をコサイン相関値により比較することで，x_i の訳語と x_j の訳語の意味的関連性を定量的に表すことができる．f_{ij} を，x_i の訳語と x_j の訳語を入力として，x_i の訳語と x_j の訳語の意味的関連性を表わす定量値が a であった場合に，1-a を出力する関数とする．このとき，目的関数は，以下のように表わされる．

$$f^*(X) = \sum_{x_i, x_j \in V} f_{ij}(X) \tag{1}$$

(集合 V はその要素に，制約が存在する変数の組を含む．) このとき，$f^*(X)$ を最大にするような全ての変数に対する値の割り当てが，この問題における解であり，求めるべき訳語の組である．最後に，図 5 に制約最適化に基づく訳語選択問題の定式を示す.

変数集合 X=$\{x_1, \ldots, x_n\}$(x_i:翻訳元名詞単語)
変域集合 D=$\{D_1, \ldots, D_n\}$(D_i:x_i の翻訳先文書中の
訳語を要素とする集合)

制約集合 C=$\{f_{ij} \mid x_i, x_j$ が翻訳元文書中で同文共起している $\}$
コスト関数 f_{ij}:翻訳先言語 Wikipedia に基づいて，x_i の訳語と x_j の訳語の意味的関連性を定量値として出力する関数
目的関数:f^*(X)=$\sum_{x_i,x_j \subseteq V} f_{ij}$(X)
(集合 V は要素として，変数間に制約が存在する変数の組を含む)

最適解:minf^*(X) とする全ての変数に対する値の組み合わせ

図 5: 制約最適化に基づく訳語選択問題の定式

3.1.3 訳語選択の例

制約最適化に基づく訳語選択の例として，以下のような 2.1 節の図 2 と同様の英語文から日本語文への機械翻訳を考える．
翻訳元文 (英)：A sentence usually contains a subject and a predicate. However, Japanese often omit the subject.
翻訳先文 (日)：文は通常，主語と述語が含まれます。しかし，多くの場合，件名を省略すると日本語。

このとき，“ sentence ”,“ subject ”,“ predicate ”,“ Japanese ”にそれぞれ対応して変数 x_1，x_2, x_3，x_4 と変域 D_1 =\{ 文 \}, D_2 =\{ 主語，件名 \}, D_3 =\{ 述語 \}, D_4 =\{ 日本語 \} がある．また，x_1x_2 間，x_1x_3 間，x_2x_3 間，x_2x_4 間に制約があり，それぞれの制約に対するコスト関数を f_{12} (x_1 , x_2), f_{13} (x_1, x_3) , f_{23} (x_2, x_3), f_{24}(x_2, x_4) とする．このとき形成される制約ネットワークは，図 6 のように表すことができる．この問題の最適解は，$f^*(X)$ を最小にするような全ての変数に対する値 (英訳語) の割り当てである．図 6 では，“ subject ”の訳語として“ 件名 ”ではなく，“ 文 ”,“ 述語 ”,“ 日本語 ”と意味的関連性の強い“ 主語 ”が選択される．

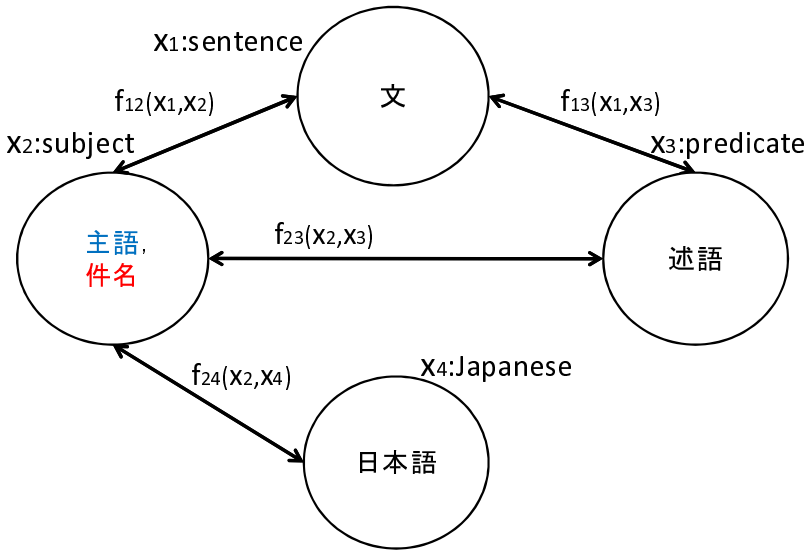

図 6: 機械翻訳における訳語選択問題の制約ネットワーク

3.2 機械翻訳連携における訳語選択

機械翻訳を連携した場合においては，各機械翻訳の訳語選択問題を制約最適化に基づいて，COP1, COP2 として定式化し，それぞれを順番に解くことで一貫した訳語選択は可能となる．だが，COP1 において解が求められ，その解にしたがった訳語の置き換えを英文書に対して行った後でなければ，英文書中の英単語を COP2 における変数に対応させ，その変域を決定することができない．つまり，COP2 は COP1 によりその問題形式が動的に変化する問題である．

COP1 と COP2 を順番に解くことによって，一貫した訳語選択を行うことは可能ではあるが，それは各訳語選択問題における最適な訳語選択を意味する．英語を中間言語とした機械翻訳連携では，全ての言語の Wikipedia の中で最も優れた資源である英語 Wikipedia を用いることができる．そのため，英語と翻訳先言語間での訳語選択に比べて翻訳元言語と英語間での訳語選択の方が信頼性が高いと言え，COP1 を解き一貫した英訳語を決定した後で，COP2 により翻訳先言語の訳語を決定することの方が文脈に適した訳語が選択されるかもしれない．しかしながら，英語 Wikipedia を用いた semantic relatedness だけでは，十分に単語間の意味的関連性を計算することができない可能性があるため，これからますます各言語における Wikipedia の記事が増えていくことを考えれば，翻訳先言語 Wikipedia を用いた英語と翻訳先言語間での訳語選択も考慮に入れた方がより確実な訳語選択を行うことができると思われる．そこで，全体最適

な訳語選択を行うために，COP1, COP2 を 1 つの問題として定式化し，最適解を求める．その定式化を条件付き制約充足に基づいて行う．

3.2.1 条件付き制約充足問題

条件付き制約充足問題 (CondCSP) は，もともと動的制約充足問題 [7] という名前で定義されていた．しかしながら，時間とともに問題が変化した際に，解の再利用により問題を効率良く解いていくような問題の枠組みを，Dechter et.al [8] が既に動的制約充足問題として定義していたことと，問題の初期モデルが問題を解いていく過程において条件付きで変化していくことから，条件付き制約充足問題という名前が新たに付けられた [9].

CondCSP の定義 [10] によれば，CondCSP は $P = \langle X, D, X_I, C_C, C_A \rangle$ として表わされ，$X = \{ x_1, \ldots, x_n \}$ は変数集合，$D = \{ D_1, \ldots, D_n \}$ は各変数 x_i に対して値の集合 $D_i = \{ d_1^i, \ldots, d_n^i \}$ を与える変域集合である．また，集合 $X_I (\subseteq X)$ は P の初期変数集合，C_C は一貫性制約集合，C_A はアクティビティ制約集合である．ここで，これら全ての集合は有限であるとする．

一般的な制約充足問題 (CSP) とは異なり，P の変数は解として含まれるべきか否かを決定するアクティビティ状態を持つ．問題を解く過程で値が割り当てられるのならば，その変数はアクティブである．初期変数集合 X_I は，P の全ての解で値の割り当てがなされる必要のある変数の集合である．したがって，初期変数集合に含まれる変数は常にアクティブである．初期変数集合に含まれない変数 (隠れ変数) は，初期状態ではアクティビティ状態が定義されておらず，アクティビティ制約によりアクティビティ状態を持つようになる．アクティビティ制約は，2 通りの方法で隠れ変数に対して働きかける．1 つは，隠れ変数をアクティブとし，解に含めるようにすること，もう 1 つは，隠れ変数を解から明確に除外するようにすることである．アクティビティ制約によりこのような影響を受けない隠れ変数は，状態が未定義のままであり解に含まれることはない．

引数が j の一貫性制約 c は，変数集合 $x_1, \ldots, x_j$ に対する許容される値の組み合わせを各変数の変域に対する直積集合の部分集合から特定する．つまり，$c(x_1, \ldots, x_j) \subseteq D_1 \times \cdots \times D_j$ である．また，制約を課している変数全てがアクティブの場合にのみ一貫性制約は解探索に関わってくる．アクティビティ制約は，変数のアクティビティに対する制約を与える．ここで，変数をアクティブとするような制約を包含アクティビティ制約と呼び，$c \xrightarrow{\text{incl}} x$ で表わす．c は一般的には一貫性制約と同様の形式であり，c が成り立つときに，x はアクティブ

変数集合 X =$\{x_1, \ldots, x_n\}$
変域集合 D =$\{D_1, \ldots, D_n\}$
初期変数集合 $X_I(\subseteq X)$

一貫性制約集合 C^C:
$c \subseteq C^C$,c$(x_1, \ldots, x_j)\subseteq D_1\times, \cdots, \times D_j$
アクティビティ制約集合 C^A:
$a \subseteq C^A$,a: $c \xrightarrow{\text{incl}} x_k$ または$c \xrightarrow{\text{excl}} x_k$

解:全ての一貫性制約とアクティビティ制約を満たすようなアクティブ変数に対する値の割り当て

図 7: 条件付き制約充足問題の定義

となることを表している．また，変数を除外するような制約を排他アクティビティ制約と呼び，$c \xrightarrow{\text{excl}} x$ で表わし，c が成り立つときに，x が除外されることを表す．

P の解は，全ての一貫性制約とアクティビティ制約を満たすようなアクティブ変数に対する値の割り当てである．具体的には，以下の 2 つの条件を満たすような P に対する割り当て A である．

1. A が各一貫性制約 $c \subseteq C_C$ を満たす．つまり，c の全ての変数が A においてアクティブであれば，c に制約を課せられている変数には c により許容される値が割り当てられている．ただし，c の全ての変数が A においてアクティブでなければ，c は自明に満たされている．
2. A が各アクティビティ制約 $a \subseteq C_A$ ($a: c \xrightarrow{\text{incl}} x_k$ または $c \xrightarrow{\text{excl}} x_k$) を満たす．つまり，A において c の全ての変数がアクティブであり c が満たされているならば，x_k は A においてアクティブであるか A から除外されている．ただし，c の全ての変数がアクティブでないまたはアクティブであるが A において c が満たされていないならば，a は自明に満たされている．

最後に ConCSP の定義を図 7 にまとめる．

3.2.2 条件付き制約充足問題に基づく訳語選択の定式化

機械翻訳連携における訳語選択問題を条件付き制約充足問題に基づいて定式化する．ここでは，言語 L_1 から英語を中間言語とした言語 L_2 への機械翻訳連携を考える．変数を $x_1, \ldots, x_n, x_{n+1}, \ldots, x_{n+|D_1|}, \ldots, x_{n+|D_1|+,\ldots,+|D_{n-1}|+1}, \ldots, x_{n+|D_1|+,\ldots,+|D_{n-1}|+|D_n|}$, それぞれに対応する変域を $D_1, \ldots, D_n, D_{n+1}, \ldots, D_{n+|D_1|}, \ldots, D_{n+|D_1|+,\ldots,+|D_{n-1}|+1}, \ldots, D_{n+|D_1|+,\ldots,+|D_{n-1}|+|D_n|}$ とする．$1 \leq i \leq n$ において，各変数 x_i は L_1 文書に現れる名詞単語のうち，英語文書中で英訳語が名詞単語として得られ，L_2 文書中でその英訳語の L_2 訳語が名詞単語として得られるような各名詞単語に対応しており，n はそのような名詞単語の総数である．また D_i は，L_1 単語 x_i の英文書中での全ての訳語を要素として含む有限集合である．$n + \sum_{l=0}^{j-1} \mid D_l \mid +1 \leq i \leq n + \sum_{l=1}^{j} \mid D_l \mid, 1 \leq j \leq n$ において，各変数 x_i は変数 x_j の変域 D_j に含まれる $i - n - \sum_{l=0}^{j-1} \mid D_l \mid$ 番目の要素である英訳語に対応しており，$\mid D_l \mid$ は変域 D_l に含まれる英訳語の総数である (ただし，$\mid D_0 \mid = 0$)．また D_i は，L_1 単語 x_j の英訳語 x_i の L_2 文書中での全ての訳語を要素として含む有限集合である．各変数の値は，対応する変域集合に含まれる要素である訳語から選択される．ここで，初期変数集合は，$X_I = \{x_1, \ldots, x_n\}$ である．

一貫性制約については，$1 \leq i < j \leq n$ では，L_1 文書中で，x_i と x_j が，同文共起しているならば x_i , x_j 間に制約 f_{ij} が存在し，その制約を“ x_i の訳語と x_j の訳語は意味的関連性がある ”として表し，$var(f_{ij}) = \{ x_i, x_j \}$ とする．$n+1 \leq i < j \leq n+ \mid D_1 \mid + \ldots + \mid D_{n-1} \mid + \mid D_n \mid$ では，L_1 文書中で，x_i を変域に含む変数と x_j を変域に含む変数が同文共起しているならば，x_i, x_j 間に制約 f_{ij} が存在し，その制約を“ x_i の訳語と x_j の訳語は意味的関連性がある ”として表し，$var(f_{ij}) = x_i, x_j$ とする．ここで，機械翻訳の訳語選択問題の場合と同様に，f_{ij} を x_i の訳語と x_j の訳語の意味的関連性を定量値として出力する関数とする．アクティビティ制約については，$1 \leq i \leq n$ で，変数 x_i が変域 D_i に含まれるいずれの訳語をその値として持つかにより，変域 D_i に含まれる訳語に対応する変数のうちいずれの変数がアクティブになるかが決定される．そのようなアクティビティ制約は，$x_i = d_j^i$ ($\subseteq D_i$) $\xrightarrow{\text{incl}}$ active: $x_{(n+\sum_{l=1}^{i-1}|D_l|+j)}$ ($j = 1, \ldots, \mid D_i \mid$) として表わされる．

このとき，目的関数 $f^*(X)$ を (1) と同様に表わすとすれば，アクティビティ制約を満たした上で $f^*(X)$ を最大にするような全ての変数に対する値の組み合

わせが，この問題における解である．また，結果的に得られる翻訳先言語の訳語の組は，$\{x_{n+1}, \ldots, x_{n+|D_1|}, \ldots, x_{n+|D_1|+\ldots+|D_{n-1}|+1}, \ldots, x_{n+|D_1|+\ldots+|D_{n-1}|+|D_n|}\}$ に含まれる変数に対する値の組合わせである．このようにして得られる訳語の組は，各機械翻訳の訳語選択問題を別々の COP として定式化し，順番に解いて得られた訳語の組とは異なり， 全体最適な訳語の組であると言える．最後に，図 8 に条件付き制約充足問題に基づく機械翻訳連携における訳語選択問題の定式を示す.

第 4 章　解法

4.1　機械翻訳における訳語選択

4.1.1　分枝限定法

分枝限定法 [11] は，制約最適化問題を解くための探索手法である．分枝限定法では，ある 1 つの変数 x_i を選び，現在の問題を変域 D_i に含まれる各値に対して 1 つの問題が対応する部分問題の集合に変換することで，現在のノードを展開していく．探索木において，内部ノードは変数に対する値の割り当てがまだ完全には終わっていないことを表し，葉ノードは変数に対して完全な値割り当てが行われたことを表す．走査は深さ優先探索であるが，走査の間では，それまでに見つけた最良解のコストを保持しており，そのコストは問題の最適解のコストに対する上界 (UB) である．探索空間全体が走査されれば探索は終了であり，その時の UB が問題の最適解のコストである． ここで，ノード t を展開することで発見できる最良解のコストの下界 (LB(t)) を計算することができ，$UB \leq$ LB(t) であると分かれば，t を展開することでコストを改善することはできないため，t 以下の部分木を枝刈りし，一つ前のノードに後戻りすることによって，探索空間を縮小することができる．

アルゴリズム (Algorithm1) は，制約充足問題の解法の 1 つであるフォワードチェッキング [12, 13] に基づく深さ優先探索分枝限定法のアルゴリズムである．P ，F はそれぞれ過去に値の割り当てをした変数の集合，これから値の割り当てをする変数の集合を表す．また，$Domain_m^j$ は，$x_j \longleftarrow d_m^j$ が，P に含まれる各変数に対する値割り当てと一貫性を保っているか否かを表すための二次元配列であり，$Domain_m^j = 0$ であれば一貫性は保たれており，そうでなければ $x_j \longleftarrow d_m^j$ と一貫性を保つことができないような値割り当てが最初に行わ

変数集合 X =$\{x_1, \ldots, x_n, x_{n+1}, \ldots, x_{n+|D_1|}, \ldots, x_{n+|D_1|+,\ldots,+|D_{n-1}|+1}, \ldots,$
$x_{n+|D_1|+,\ldots,+|D_{n-1}|+|D_n|}\}$
変域集合 D =$\{D_1, \ldots, D_n, D_{n+1}, \ldots, D_{n+|D_1|}, \ldots, D_{n+|D_1|+,\ldots,+|D_{n-1}|+1}, \ldots,$
$D_{n+|D_1|+,\ldots,+|D_{n-1}|+|D_n|}\}$
初期変数集合 X_I=$\{x_1, \ldots, x_n\}$

一貫性制約集合
$(1 \leq i < j \leq n)$
$C^C = C_1^C \cup C_2^C$
C_1^C=$\{f_{ij} \mid x_i, x_j$ が翻訳元文書中で同文共起している $\}$
$(n+1 \leq i < j \leq n+ \mid D_1 \mid + \ldots + \mid D_{n-1} \mid + \mid D_n \mid)$
C_2^C=$\{f_{ij} \mid x_i$ を変域に含む変数と x_j を変域に含む変数が同文共起している $\}$
コスト関数 f_{ij}:x_i, x_j の訳語言語 Wikipedia に基づいて x_i の訳語と x_j の
訳語の意味的関連性を定量値として出力する関数.
アクティビティ制約集合
$C^A = C_1^A \cup \cdots \cup C_n^A$
C_i^A:$x_i = d_j^i (\subseteq D_i\ ) \xrightarrow{\text{incl}}$ active:$x_{n+\sum_{l=1}^{i-1}|D_l|+j} (j = 1, \ldots, \mid D_i \mid)$
目的関数:f^*(X)=$\sum_{x_i,x_j \subseteq V} f_{ij}$(X)
(集合 V は要素として，変数間に制約が存在する変数の組を含む)

最適解:アクティビティ制約を満たした上で，minf^*(X) とする全ての変数に
対する値の組み合わせ.
また，結果的に得られる翻訳先言語の訳語の組は,
$\{x_{n+1}, \ldots, x_{n+|D_1|}, \ldots, x_{n+|D_1|+\ldots+|D_{n-1}|+1}, \ldots, x_{n+|D_1|+\ldots+|D_{n-1}|+|D_n|}\}$ に
含まれる変数に対する値の組み合わせである.

図 8: 条件付き制約充足に基づく機械翻訳連携における訳語選択問題の定式

れた変数のインデックスをその値として持つ．以下，アルゴリズムの説明を行う．F から変数 x_i を選び，D_i に含まれる各値 d_l^i に対して繰り返しの処理が行われる．$Domain_l^i = 0$ であれば，x_i に対して d_l^i を割り当て，x_i と d_l^i のペアを解 S に含める．次に，x_i に対する d_l^i の割り当てが可能であるか否かを

LookAhead(Algorithm2) により検証する．もし，検証結果が真であれば，P に x_i を加え，F から x_i を除き再帰呼び出しが行われる．このとき，F が空であれば解 S は現在の最良解を改善するので，UB を更新し，最良解として S が選ばれる．偽であれば，Restore により LookAhead で行われた枝刈りを無効とし，x_i に d_l^i を割り当てる前の状態に戻る．

LookAhead では，F に含まれる各変数の変域に含まれるうちの少なくとも 1 つの値が現在の変数に対する値割り当てと一貫性を保っているならば，返り値として true を返している．一貫性を保っているか否かは，現在の部分解 S に，F に含まれる変数とその変域に含まれる値のうち $Domain = 0$ であるような値を追加した S' に対して，コストの下界を計算し，UB (最良解の上界) と比較することで行う．UB より値が小さければ，今後 UB を更新することが期待されることを示す (このとき，LookAhead のアルゴリズムでは，“ domain wipe-out ” つまり，ある変数の領域に含まれる値の全てが現在の変数に対する値割り当てと一貫性を保つことができないことを表す変数 dwo に false が代入されている)．UB より値が大きいのであれば，今後 UB を更新することは期待されないため，$Domain$ に現在値の割り当てを行っている変数のインデックスを代入することで枝刈りを行う．なおここでは，S' により計算することが可能な各コスト関数 (関数 f の範囲 $varf$ に含まれる変数とその変数に対する値割り当て全てが S' に含まれているような関数) の出力値の総和を LB とする．

4.1.2 分枝限定法に基づく解法例

制約最適化に基づいて定式化された訳語選択問題は，分枝限定法により解を求めることができる．例として，英名詞単語“ subject ”,“ sentence ”,“ author ”,“ paper ”に対応して x_1, x_2, x_3, x_4 と変域 $D_1 =$ { 主語，件名 }，$D_2 =$ { 文 }，$D_3 =$ { 著者 }，$D_4 =$ { 紙，論文 } があり，また x_1x_2 間，x_2x_4 間，x_3x_4 間に制約があり，それぞれの制約に対するコスト関数が $f_{12}(x_1, x_2), f_{24}(x_2, x_4), f_{34}(x_3, x_4)$ であるような訳語選択問題を考える．すると，この問題の最適解を求めるための探索木は，図 9 で表わされる．ここで，分枝限定法に基づいてまず，x_1 に“ 主語 ”，x_2 に“ 文 ”，x_3 に“ 著者 ”，x_4 に“ 紙 ”をそれぞれ割り当てる探索を行い，その時の目的関数の出力値を UB ，最良解を Best Tuple= {{x_1 ，主語 }，{x_2 ，文 }，{x_3 ，著者 }，{x_4 ，紙 }} とする．また，x_4 に対する値割り当ての状態に後戻りし，x_4 に“ 論文 ”を値として割り当てた場合の目的関数の出力値を得て，その出力値が現在の UB よりも小さい値である場合は UB ，$BestTuple$

Algorithm 1 分枝限定法のアルゴリズム BB(P,F)

```
P /*A set of past variables*/
F /*A set of future variables*/
d_l^i /*The value assigned to variable x_i*/
s_i /*A pairs of x_i and d_l^i*/
S /*A set of s_i*/
UB /*The upper bound of the total cost*/
if F = ∅ then
  UB ←ComputeCost(S)
  BestTuple ← S
else
  x_i ← SelectVariable(F)
  for each d_l^i in D_i do
    if Domain_l^i = 0 then
      s_i ← (x_i, d_l^i)
      S ← Append(S, s_i)
      if LookAhead(i, F − {x_i}) then
        BB(P ∪ {x_i}, F − {x_i})
      end if
      Restore(i, F)
      Pop(S)
    end if
  end for
end if
```

Algorithm 2 先読み枝刈りを行うアルゴリズム LookAhead(i, F)

```
for each x_j in F do
  dwo = true
  for each d_m^j in D_j do
    if Domain_m^j = 0 then
      S' ←Append(S, (x_j, d_m^j))
      if LB(S') < UB then
        dwo = false
      else
        Domain_m^j ← i
      end if
    end if
  end for
  if dwo then
    return(false)
  end if
end for
return(true)
```

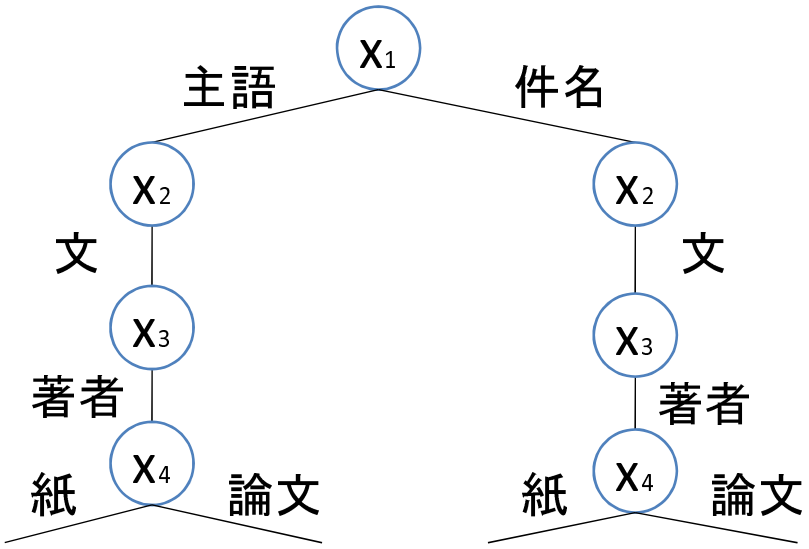

図 9: 訳語選択問題の例から生成される探索木

を更新するが，そうでない場合は，更新をしない．次に，x_1 に対する値割り当ての状態まで後戻りをし，x_1 の値として“ 件名 ”を選択した場合の探索について考える．ここで，x_1 に対する“ 件名 ”の割り当てと一貫性を保つことができないような各変数 x_2, x_3, x_4 の各変域に含まれる値が除かれる．例えば，x_1 に“ 件名 ”，x_2 に“ 文 ”を割り当てた部分解から得られる解の目的関数の出力値の下界 (LB) を計算し，LB が UB よりも大きいのであれば，これ以降の探索は行わない，つまり枝刈りを行う．LB は，現在の部分解で計算できる各コスト関数の出力値の総和から求めるとすれば，コスト関数 $f_{12}(x_1, x_2)$ の出力値を LB とし，UB と比較すればよい．もし,“ 件名 ”と“ 文 ”の意味としての関連性が小さいなどの理由で，$UB \leq LB$ であれば，必要のない探索を行わずに済む．もし，$UB > LB$ であれば x_3, x_4 に対しても同様の処理が行われる．各変数 x_2, x_3, x_4 の各変域に含まれる値が少なくとも 1 つずつ残っている場合は，x_1 に“ 件名 ”を割り当てた後で，今度は x_2 に対する値割り当てを行うことで，探索は続けられる．

4.2 機械翻訳連携における訳語選択

4.2.1 FC に基づく条件付き制約充足問題の解法

条件付き制約充足問題 (CondCSP) の解法として，制約充足問題の解法の一つであるバックトラック法に基づく解法が提案されている [7, 13]．また，バッ

クトラック法以外にもフォワードチェッキングとアーク整合性 [14] に基づく解法も提案されている [10, 15, 16]．本研究では，CondCSP をフォワードチェッキングに基づく解法により解くことを考える．ただし，Gelleet.al [10] と同様に，CondCSP における一貫性制約は 2 値制約であるとし，アクティビティ制約は $x_i = d^i_j \xrightarrow{\text{incl}} x_k$ または $x_i = d^i_j \xrightarrow{\text{excl}} x_k$ として表わされるとする．また，アクティビティ制約について，$x_i = d^i_j$ を条件，x_i を条件変数，x_k を目的変数と呼ぶこととする．CondFC のアルゴリズム (Algorithm3) の基本的な流れは，4.1.1 節での分枝限定法のアルゴリズムと同じである．異なる点は，最適化問題ではなく充足問題であるため，LookForward におけるコストの上界下界を用いた枝刈りではなく，CheckForward(アルゴリズムは付録 A.1 の Algorithm5 を参照) により制約に対して一貫性を保つことができているか否かに基づいた枝刈りが行われている点，CheckActivity により CondCSP 特有のアクティビティ制約に対する確認が行われている点である．

CheckActivity(Algorithm4) では，条件変数が x_i であるようなアクティビティ制約を集合 A に含め，A に含まれる各アクティビティ制約に対して繰り返しの処理を行う．アクティビティ制約の条件における変数と割り当てられる値のペアが，s_i $(=(\ x_i, d^i_j\ ))$ と一致するならば，変数 $target$ にアクティビティ制約の目的変数を，$action$ にアクティビティ制約に従って incl または excl を含める．ここで，$action$ が incl の場合と excl の場合とで場合分けがされる．前者の場合は，$target$ に含まれている変数が既にこれまで除去した変数集合 $ExcludedVariables$ に含まれているのであれば返り値として false を返し，$target$ に含まれている変数がこれまでアクティブとした変数集合 $IncludedVariables$ に含まれていないのであれば，$target$ を $x_i = d^i_j$ によりアクティブとなった変数集合 $Newvariables$ に含めることとする．後者の場合は，$target$ に含まれている変数が既にこれまでアクティブとなった変数集合 $IncludedVariables$ に含まれているのであれば返り値として false を返し，$target$ に含まれている変数がこれまで除去した変数集合 $ExcludedVariables$ に含まれていないのであれば，$target$ を $x_i = d^i_j$ により除去された変数集合 $ExcludedVals$ に含めることとする．ここまでの処理により，アクティビティ制約により新たにアクティブとなる変数を特定するとともに，アクティブ制約を満たしているか否かを検証している．ここで，新たにアクティブとなった変数集合 $Newvariables$ に含まれる各変数に対して，CheckBackward(アルゴリズムは付録 A.1 の Algorithm6 を参照) により現在の部分解 S と一貫性を

保つことのできない値を対応する変域から除去する．もし，変域に一貫性を保つことのできる値が 1 つも存在しないような変数が $Newvariables$ に存在していれば，Restore を呼び返り値として false を返す．最後に，新たにアクティブとなった各変数を F と $IncludedVariables$ に含めるとともに，新たに除去された変数を $ExcludedVariables$ に含めるようにし，返り値として true を返すこととする．

Algorithm 3 FC に基づく CondCSP のアルゴリズム FCforCondCSP(P,F)

```
if F = ∅ then
  Solution ← Solution ∪ S
else
  x_i ←SelectVariable(F)
  for each d_l^i in D_l do
    if Domain_l^i = 0 then
      s_i ← (x_i, d_l^i)
      S ← Append(S, s_i)
      if CheckForward(i, F − {x_i}) and CheckActivity(i, F, s_i, S) then
        FCforCond(P ∪ {x_i}, F − {x_i})
      end if
      Restore(i, F)
      Pop(S)
    end if
  end for
end if
```

4.2.2 条件付き制約充足問題の解法に基づく解法

4.1.1 節で述べた分枝限定法のアルゴリズムを基に，4.2.1 節で述べた FC に基づく CondCSP のアルゴリズムの最初の 2 行 (F が空である場合に，ComputeCost により解 S のコストを計算し，S を最良解とする) および 9 行目 (CheckForward を LookAhead にする) を変更し，アクティビティ制約確認アルゴリズムの Check-Backward を取り除くことで，制約最適化問題を解くためのアルゴリズムとする．このようにすることで，条件付き制約充足問題に基づいて定式化された機械翻訳連携における訳語選択問題を解くことが可能となる．

Algorithm 4 アクティビティ制約確認アルゴリズム CheckActivity(i, F, s_i, S)

```
Newvariables /*A set of variables included by s_i*/
ExcludedVals /*A set of variables excluded by s_i*/
IncludedVariables /*A set of all included variables*/
ExcludedVariables /*A set of all excluded variables*/
A ←activity constraints whose conditoins involve x_i
for each a in A do
  if s_i isconsistent with a's condition then
    target ←targetvariable by a
    action ←activity performed by a
    if action includes target then
      if target ∈ ExcludedVariables then
        return(false)
      end if
      if target ∉ IncludedVariables then
        Newvariables ← Newvariables ∪ target
      end if
    else
      if target ∈ IncludedVariables then
        return(false)
      end if
      if target ∉ ExcludedVariables then
        ExcludedVals ← ExcludedVals ∪ target
      end if
    end if
  end if
end for
for each x_i in Newvariables do
  if CheckBackward(i, x_i, S) then
    Restore(i, Newvariables)
    return(false)
  end if
end for
for each newvar in Newvariables do
  F ← F ∪ newvar
  IncludedVariables ← IncludedVariables ∪ newvar
end for
ExcludedVariables ← ExcludedVariables ∪ ExcludedVals
return(true)
```

第5章　実装と評価

5.1　訳語選択システムの実装

3.1 節および 4.1 節で述べた定式化及び解法に基づいて，機械翻訳における文間文脈に基づいた一貫した訳語選択を行うシステムを実装した．図 10 では，翻訳元文書中の 1 文を Get Source Sentence により得て，その文を機械翻訳サービス (MT) により翻訳先文に翻訳し，翻訳元文 ss と翻訳先文 ts からなる 2 つの文の組を Get A Pair of Sentences により得ている．次に，Get Pairs of Noun Words により先ほど得た文のペアを用いて，ss に含まれる名詞単語と ts に含まれるその名詞単語の訳語からなる名詞単語のペア集合を得る．この処理を翻訳元文書内の全ての文に対して繰り返し行うことで，最終的に各文に対する名詞単語のペア集合を得る．ここで得られた各文に対する名詞単語のペア集合から，訳語選択問題を Formulate Problem により制約最適化問題として定式化する．

図 11 の Solve Problem では，図 10 で定式化された制約最適化問題を，コスト関数を用いて分枝限定法により解いている．この解は，3 章でも述べたように，翻訳元言語名詞単語に対応する変数 x_i とその翻訳先名詞単語 d_i のペアからなる集合であり，各翻訳元言語名詞単語に対して唯一つの翻訳先言語名詞単語が得られる．

ここで，コスト関数とは，2 つの単語の意味としての関連の強さを定量値として出力する関数である．図 12 で示されているように，まず入力として受け取られた 2 つの単語は，それぞれ semantic interpreter により各 Wikipedia 記事に対してその関連の強さから重みづけをされたリストとして表現される．これは，それぞれの単語が各 Wikipedia 記事に出現した回数を基とした，tf/idf score により事前に得ておくことが可能である．それらのリスト間のコサイン相関値から入力として受け取った単語間の意味としての関連の強さを 0 以上 1 以下の値 a として出力する．a の値が 1 に近いほど単語間の意味としての関連は強いが，制約最適化問題ではコストを最小とすることを目的とするため，実際に問題と解く際には，コスト関数は 1-a を出力することとしている．

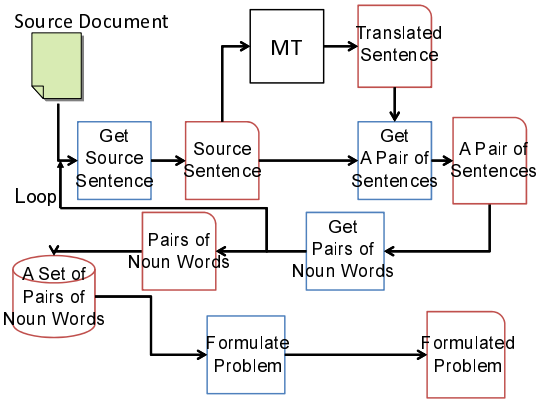

図 10: 機械翻訳における訳語選択問題の制約最適化問題としての定式化過程

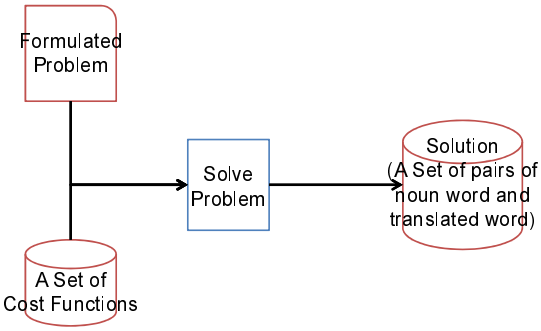

図 11: 機械翻訳における制約最適化問題として定式化された訳語選択問題を解く過程

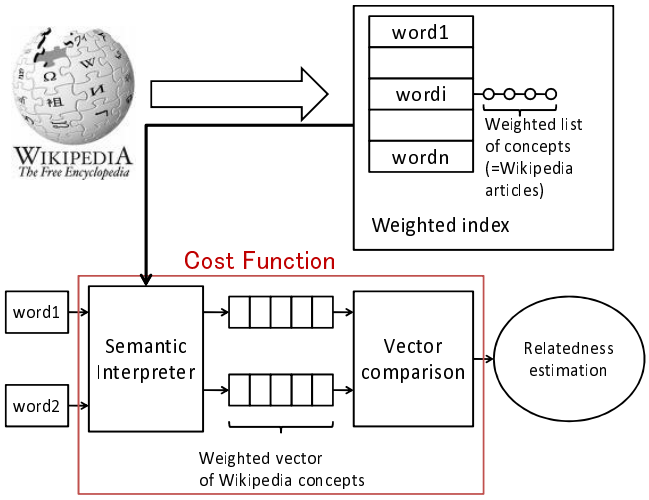

図 12: コスト関数の仕組み

5.2 評価

5.2.1 訳語選択システムの評価

5.1 節で実装した機械翻訳における訳語選択システムの評価を英日機械翻訳において行った．翻訳する対象の記事には，英語 Wikipedia の世界 7 大陸の各記事の各パラグラフ (ただし，See also 以下のパラグラフを除く) の記事文書を用いた．パラグラフごとに評価した理由としては，記事全体で一貫した訳語選択をすると，同じ記事内でもパラグラフごとに述べられている内容に隔たりがあるため，文脈に適さないような誤った訳語選択が多く行われてしまう可能性があると判断したからである．評価方法としては，各パラグラフの記事文書に対して，Google Translate を用いた英日機械翻訳を行い，(1)“ 記事内で複数回出現しており，訳語選択の非一貫性問題が生じなかった名詞単語 ”，(2)“ 記事内で複数回出現しており，訳語選択の非一貫性問題が生じた名詞単語 ”をそれぞれカウントした．なお，名詞単語に対する訳語の同定は，形態素解析と対訳辞書などを用いたプログラムを実装することにより行った．次に，各パラグラフの記事文書の Google Translate を用いた英日機械翻訳において，訳語選択システムにより各英語名詞単語に対して一貫した訳語選択を行い，(3)“ (2) に分類されていた名詞単語のうち，訳語選択システムにより文脈に適した一貫した訳語選択が行われた名詞単語 ”，(4)“ (2) に分類されていた名詞単語のうち，訳語選択システムにより文脈に適切ではない一貫した訳語選択が行われた名詞単語 ”，(5)“ (2) に分類されていた名詞単語のうち，訳語を一貫させる必要がないにも関わらず，訳語選択システムにより一貫した訳語選択が行われた名詞単語 ”，をそれぞれカウントした．

表 1 は，各記事に対して，その記事に含まれるパラグラフの (1) と (2) に含まれる名詞単語の合計をそれぞれ示した表であり，表 2 は，各記事に対して，その記事に含まれるパラグラフの (3)，(4) および (5) に含まれる名詞単語の合計をそれぞれ示した表である (各パラグラフごとの詳細な結果は付録 A.2 の表を参照)．各パラグラフの記事内で複数回出現していた名詞単語 859 語のうち 116 語 (13%) において訳語の非一貫性問題が生じており，それらの名詞単語に対しては，訳語選択システムにより文間文脈に基づいた一貫した訳語選択を可能とした．また，訳語選択システムにより，訳語が一貫していなかった名詞単語 116 語のうち 93 語 (80%) に対して文書文脈に適した訳語を与えることが可能

であった.

(2) に含まれていた名詞単語には，訳語が一貫してはいないものの，ほぼ同じ意味を表しており文書の理解に支障をきたさないと思われる単語，および訳語が一貫しておらず訳語の意味の違いが文書の理解に支障をきたす単語の 2 種類が存在する．前者の例としは,“ name ”(訳語:名前，名),“ west ”(訳語:西，西方) が挙げられ，後者の例としては,“ capital ”(訳語:資本，首都),“ current ”(訳語：電流，海流) が挙げられる．訳語が一貫しておらず (2) に含まれていた名詞単語は，さらに文間文脈に基づいた一貫した訳語選択が訳語選択システムにより行われ，(3)〜(5) のいずれかに分類される．(3) に含まれる名詞単語の例としては，先の例で挙げた“ capital ”(文脈に適した訳語:首都，文脈に適した訳語:資本),“ current ”(文脈に適した訳語:海流，文脈に適した訳語:電流) が挙げられ，それらはそれぞれ付録 A.3 の図 A.1 で示されるような文においては，異なる訳語が選択されていたが，訳語選択システムにより，文書文脈に適した一貫した訳語選択が行われるようになった．また,“ name ”,“ west ”などの，訳語が一貫してはいないが，それらの訳語がほぼ同じ意味を表している名詞単語については，一貫した訳語としてどちらを選択した場合でも (3) に含めることとした．一方，(4) に含まれる名詞単語の例としては,“ party ”(文脈に適さない訳語:党，文脈に適した訳語:パーティー),“ system ”(文脈に適さない訳語:システム，文脈に適した訳語:体制) が挙げられる．それぞれ，付録 A.3 の図 A.2 で示されるような文においては正しい訳語が選択されていたにも関わらず (“ パーティー ”が適した訳語だとは言えないかもしれないが,“ 党 ”と比較して文脈に適した訳語であると判断した)，訳語選択システムにより，文書文脈に適していない一貫した訳語が選択されてしまった．(5) に含まれる，訳語を一貫させることによりかえって文書の理解を妨げる要因となった名詞単語の例としては,“ area ”が挙げられ，付録 A.3 の図 A.3 に示されるように“ area ”を“ 面積 ”または“ 地域 ”(“ 地域 ”が適した訳語とは言えないかもしれないが,“ 面積 ”と比較して文脈に適した訳語であると判断した) と翻訳される文に応じて訳語を選択する必要があるにも関わらず，一貫した訳語として“ 地域 ”を選択してしまった．

5.2.2 考察

(3) に含まれていた名詞単語 93 語の中には，先に例として挙げた“ name ”,“ west ”のような訳語が一貫していないが，いずれの訳語を選択したとしても，それらの訳語の意味の違いが，文書の理解に支障をきたすことがないと考えら

記事タイトル	(1)	(2)	総数 (1)+(2)
Asia	79	14	93
Africa	136	24	160
North America	74	10	84
South America	98	12	110
Antarctica	149	23	172
Europe	177	28	205
Australia	30	5	35
総計	743	116	859

表1: Goolgle Translate を用いた英日機械翻訳における訳語選択の非一貫性に基づく名詞単語の分類

記事タイトル	(3)	(4)	(5)	総数 (2)
Asia	10	3	1	14
Africa	18	2	4	24
North America	8	1	1	10
South America	11	1	0	12
Antarctica	18	4	1	23
Europe	23	3	2	28
Australia	5	0	0	5
総計	93	14	9	116

表2: Goolgle Translate を用いた英日機械翻訳において訳語選択システムにより選択された一貫した訳語が文脈に適しているか否かに基づいた名詞単語の分類

れる名詞単語 60 語が含まれていた．仮にそれらの名詞単語の訳語を一貫させることが，翻訳品質の向上につながらないと考えるならば，残りの 33 語のみが翻訳品質の向上につながる名詞単語の数と考えることができ，一方で，文脈に適さない一貫した訳語が選択された名詞単語の数と訳語を一貫させる必要がなかった名詞単語の数を合わせた 23 語を，翻訳品質の低下につながる名詞単語の数と考えることができる．つまり，訳語選択システムにより訳語を一貫させることが翻訳品質に影響を与えた名詞単語 56 語のうち 33 語 (59%) のみが翻訳品質の向上につながると考えられる名詞単語であると言える．しかしながら，翻訳品質の向上につながった名詞単語 33 語と比較して，翻訳品質の低下につながった名詞単語 23 語には，一貫していない訳語間で意味に隔たりのある単語を訳語として持つ名詞単語 (例えば，current(訳語：海流，電流)，right(訳語：権利，右)) の含まれている数が少なかった．

そこで，SR を 2 つの名詞単語を引数にとり，それらの意味的関連性を Wikipedia に基づいて定量的に 0 以上 1 以下の値として出力する関数とし，ある名詞単語 w の訳語として，$tw_1, tw_2, \ldots, tw_n$ が存在しているときに，訳語選択システムに

より選択された一貫した訳語 tw_k が，その他の訳語との間に SR(tw_k, tw_i) $< b$ (ただし，$i \neq k, 1 \leq n$ であり，b は閾値) が成り立つ時のみ tw_k を w の訳語として選択することを考える．今回は，$b = 0.08$ として実際に，翻訳品質の向上につながった名詞単語 33 語および翻訳品質の低下につながった名詞単語 23 語に対して，訳語の選択を行った．その結果，翻訳品質の向上につながった名詞単語 33 語のうち 12 語が，翻訳品質の低下につながった名詞単語 23 語のうち 7 語が一貫した訳語として選択された．この翻訳品質の低下につながった名詞単語 7 語の中には,“ community ”が 2 語含まれていたが，それは“ community ”の訳語として“ 社会 ”と“ 地域社会 ”が考えられるときに文脈に適さない訳語“ 社会 ”を選択してしまったからであった．しかし，今回は“ 地域社会 ”のような名詞単語は“ 地域 ”と“ 社会 ”に分けられてそれぞれの名詞単語の各 wikipedia 記事に対して重み付されたリストが得られてしまっていたため，結果的に“ 地域社会 ”のリストの値は 0 となり，SR(“ 社会 ”,“ 地域社会 ”)=0 となってしまっていた．もし，SR(“ 社会 ”,“ 地域社会 ”) ≤ 0.8 であると考えるならば，翻訳品質の低下につながった名詞単語 23 語のうち 5 語が一貫した訳語として選択されたため，名詞単語 17 語のうち 12 語 (70%) が翻訳品質の向上につながると言える．

ここで述べた手法では，翻訳品質を向上させる名詞単語の多くを除去してしまい，翻訳品質を低下させる名詞単語を十分に除去することができなかったが，今後は，文脈に適した一貫した訳語のみを選択し，文脈に適さないまたは一貫すべきでない単語の訳語を選択することを防ぐことにより，翻訳文書の翻訳品質を低下させないような訳語選択手法を考える必要があると思われる．

第 6 章　おわりに

多様な言語で表記された膨大な情報で溢れかえっているインターネット上では，機械翻訳を連携させることで多くの言語間での機械翻訳を可能とし，より多くの情報を理解できるようになる．しかし，機械翻訳連携には同じ単語の訳語がその翻訳文書内で一貫しないという問題，訳語選択の非一貫性問題が存在し，翻訳の質を低下させる．

この問題を解決するために，本研究では，まず機械翻訳における訳語選択問題を制約最適化問題として定式化した．同文共起する翻訳元単語の訳語間に制

約を課し，最も制約を満たすような訳語の組をその文内で選択されるべき訳語の組とする．このようにすることで，非一貫性問題が生じる単語，つまり複数文で現れる単語の訳語選択は，それぞれの文で共起している全ての単語の訳語選択の影響を受けるため，文間文脈を考慮に入れた一貫した訳語選択が可能となる．機械翻訳連携においても，各機械翻訳での訳語選択問題を制約最適化問題に基づいて解くことにより，一貫した訳語選択が可能となった．また，機械翻訳連携における訳語選択問題を，条件付き制約充足問題に基づいて 1 つの制約最適化問題として定式化した．これにより，連携した各機械翻訳の訳語選択問題を制約最適化問題として別々に解いた場合とは異なり，機械翻訳全体としての最適な訳語選択が可能となった．

訳語選択問題を解くためのアルゴリズムを示すとともに，実際に機械翻訳において，文間文脈に基づいた一貫した訳語選択を行うシステムを実装し，システムの評価をした．記事内で複数回出現していた名詞単語 859 語のうち 116 語 (13%) において訳語の非一貫性問題が生じており，それらの名詞単語に対しては，訳語選択システムにより一貫した訳語選択を可能とした．また，訳語選択システムにより，訳語が一貫していなかった名詞単語 116 語のうち 93 語 (80%) に対して文書文脈に適した一貫した訳語を選択することができた．この結果から，文間文脈を考慮に入れた一貫した訳語選択を行うことにより，文書を対象とした機械翻訳の翻訳品質を向上できる可能性があることが分かった．

しかしながら，これら 116 語のうち本質的に翻訳品質に影響を与えるような名詞単語の数は 56 語であり，そのうち翻訳品質を向上させると思われる文書文脈に適した一貫した訳語が選択された名詞単語の数は 33 語であり，翻訳品質を低下させると思われる文書文脈に適さない一貫した訳語が選択された名詞単語および訳語を一貫すべきでなかった名詞単語の数は 23 語であった．今後の課題としては，一貫した訳語選択により翻訳品質を向上させる名詞単語の数を減らすことなく翻訳品質を低下させる名詞単語の数を減らすための訳語選択手法を考案することが挙げられる．

謝辞

本研究を行うにあたり，熱心なご指導，ご助言を賜りました石田亨教授に厚く御礼申し上げます．また，有益な助言を与えてくださいました松原繁夫准教

授をはじめ，石田研究室の皆様方に心より感謝いたします.

参考文献

[1] T.Ishida. Language Grid: An Infrastructure for Intercultural Collaboration. IEEE/IPSJ Symposium on Applications and the Internet (SAINT-06),pages 96-100, 2006.

[2] N.Yamashita and T.Ishida. Effects of Machine Translation on Collaborative Work. In Proceedings of International Conference on Computer Supported Cooperative Work (CSCW-06), pages 515-523, 2006.

[3] R. Tanaka, Y. Murakami, and T. Ishida. Context-based approach for pivot translation services. In Proceedings of International Joint Conference on Artificial Intelligence (IJCAI-09), pages 1555-1561,2009.

[4] J.Larrosa and R.Dechter. Boosting search with variable elimination in constraint optimization and constraint satisfaction problems. Constraints 8(3), pages 303-326,2003.

[5] T.Schiex, H.Fargier and G.Verfaille.Valued Constraint Satisfaction Problems: Hard and Easy Problems. In Proceedings of International Joint Conference on Artificial Intelligence (IJCAI-95), pages 631-637,1995.

[6] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In Proceedings of International Joint Conference on Artificial Intelligence (IJCAI-07),pages 1606-1611,2007.

[7] S.Mittal and B.Falikenhainer. Dynamic constraint sa-tisfaction problems. In Proceedings of National Confe-rence on Artificial Intelligence(AAAI-90), pages 25-32, 1990.

[8] R.Dechter and A.Dechter. Belief maintenance in dynamic constraint networks. In Proceedings of National Conference on Artificial Intelligence(AAAI-88), pages 37-42,1988.

[9] M. Sabin and E.C. Freuder. Detecting and resolving inconsistency in conditional constraint satisfaction problems. In Proceedings of National Conference on Artifi-cial Intelligence (AAAI-99) Workshop on Configuration,

pages 90-94,1999.

[10] E. Gelle and M. Sabin. Solving methods for conditional constraint satisfaction. In Proceedings of International Joint Conference on Artificial Intelligence(IJCAI-03) Workshop on Configuration, pages7-12, 2003.

[11] R.M. Haralick and G.L. Elliott. Increasing tree search efficiency for constraint satisfaction problems. Artificial Intelligence 14 ,pages 263-313,1980.

[12] F.Bacchus and A.Grove . On the forward checking algorithm. In Proceedings of International Conference on Principles and Practice of Constraint Programming (CP-95), pages 292-308,1995.

[13] T.Soininen, E.Gelle and I.Niemela, A fixpoint definition of dynamic constraint satisfaction.In Proceedings of International Conference on Principles and Practice of Constraint Programming(CP-99),Alexandria, VA, Lecture Notes in Computer Science, Vol. 1713, Springer, pages 419-433,1999.

[14] R.Mohr and T.C.Henderson. Arc and path consistency revisited. Aritificial Intelligence 28, pages 225-233,1986.

[15] M. Sabin, E. C. Freuder, and R. J. Wallace.Greater efficiency for conditional constraint satisfaction.In Proceedings of nternational Conference on Principles and Practice of Constraint Programming(CP-03), pages 649-663, 2003.

[16] E.Gelle and M.Sabin. Solver framework for conditional constraint satisfaction problems. In Proceeding of European Conference on Artificial Intelligence(ECAI- 06) Workshop on Configuration, pages 14-19,2006.

付録

A.1 CheckForward と CheckBackward のアルゴリズム

4.2.1 節で示した FC に基づく CondCSP のアルゴリズム FCforCondCSP において呼ばれている CheckForward のアルゴリズムおよびアクティビティ制約確認アルゴリズム CheckActivity において呼ばれている CheckBackward のアルゴリズムを示す.

Algorithm 5 先読み枝刈りを行うアルゴリズム CheckForward(i, F

```
for each x_j in F do
  dwo = true
  for each d_m^j in D_j do
    if Domain_m^j = 0 then
      if C_{i,j} ≠ ∅ then
        if (s_i, d_m^j) ∈ C_{i,j} then
          dwo = false
        else
          Domain_m^j ← i
        end if
      else
        dwo = false
        break
      end if
    end if
  end for
  if dwo then
    return(false)
  end if
end for
return(true)
```

Algorithm 6 部分解とアクティブとなった変数との制約確認アルゴリズム

CheckBackward(i, x_j, S)

```
for each (x_k, s_k) in S do
  dwo = true
  for each d_m^j in D_j do
    if Domain_m^j = 0 then
      if C_{k,j} ≠ ∅ then
        if (s_k, d_m^j) ∈ C_{k,j} then
          dwo = false
        else
          Domain_m^j ← i
        end if
      else
        dwo = false
        break
      end if
    end if
  end for
  if dwo then
    return(false)
  end if
end for
return(true)
```

A.2 英語 Wikipedia の世界 7 大陸の各記事の各パラグラフの記事文書に対する訳語選択システムの評価結果

5.2.1 節において述べた英語 Wikipedia の世界 7 大陸の各記事の各パラグラフの記事文書に対して，Google Translate を用いた英日機械翻訳を行い，(1)“ 記事内で複数回出現しており，訳語選択の非一貫性問題が生じなかった名詞単語 ”，(2)“ 記事内で複数回出現しており，訳語選択の非一貫性問題が生じた名詞単語 ” をそれぞれカウントした際の結果の表を示す．また，各パラグラフの記事文書の Google Translate を用いた英日機械翻訳において，訳語選択システムにより各英語名詞単語に対して一貫した訳語選択を行い，(3)“ (2) に分類されていた名詞単語のうち，訳語選択システムにより文脈に適した一貫した訳語選択が行われている名詞単語 ”，(4)“ (2) に分類されていた名詞単語のうち，訳語選択システムにより文脈に適切ではない一貫した訳語選択が行われている名詞単語 ”，(5)“ (2) に分類されていた名詞単語のうち，訳語を一貫させる必要がないにも関わらず，訳語選択システムにより一貫した訳語選択が行われている名詞単語 ”，をカウントした際の結果の表を示す．

パラグラフ名	(1)	(2)	総数 (1)+(2)
Asia	5	0	5
Etymology	5	3	8
Definition and boundaries	13	2	15
Territories and regions	0	0	0
Economy	18	2	20
Early history	9	4	13
Languages and literature	15	2	17
Beliefs	14	1	15
総計	79	14	93

表 A.1: 記事“ Asia ”の各パラグラフの記事文書に対する Goolgle Translate を用いた英日機械翻訳における訳語選択の非一貫性に基づく名詞単語の分類

パラグラフ名	(3)	(4)	(5)	総数 (2)
Asia	0	0	0	0
Etymology	3	0	0	3
Definition and boundaries	2	0	0	2
Territories and regions	0	0	0	0
Economy	1	1	0	2
Early history	3	1	0	4
Languages and literature	1	0	1	2
Beliefs	0	1	0	1
総計	10	3	1	14

表 A.2: 記事“ Asia ”の各パラグラフの記事文書に対する Goolgle Translate を用いた英日機械翻訳において訳語選択システムにより選択された一貫した訳語が文脈に適しているか否かに基づいた名詞単語の分類

パラグラフ名	(1)	(2)	総数 (1)+(2)
Africa	3	0	3
Etymology	6	1	7
History	54	11	65
Geography	15	2	17
Politics	9	3	12
Economy	6	0	6
Demographics	13	2	15
Languages	6	2	8
Culture	21	1	22
Religion	3	2	5
Territories and regions	0	0	0
総計	136	24	160

表 A.3: 記事‘ Africa ”の各パラグラフの記事文書に対する Goolgle Translate を用いた英日機械翻訳における訳語選択の非一貫性に基づく名詞単語の分類

パラグラフ名	(3)	(4)	(5)	総数 (1)+(2)
Africa	0	0	0	0
Etymology	1	0	0	1
History	11	0	0	11
Geography	2	0	0	2
Politics	2	0	1	3
Economy	0	0	0	0
Demographics	0	1	1	2
Languages	0	1	1	2
Culture	1	0	0	1
Religion	1	0	1	2
Territories and regions	0	0	0	0
総計	18	2	4	24

表 A.4: 記事‘ Africa ”の各パラグラフの記事文書に対する Goolgle Translate を用いた英日機械翻訳において訳語選択システムにより選択された一貫した訳語が文脈に適しているか否かに基づいた名詞単語の分類

パラグラフ名	(1)	(2)	総数 (1)+(2)
North America	9	1	10
Etymology	8	1	9
History	9	2	11
Geography and extent	34	4	38
Countries and territories	14	2	16
Communications	0	0	0
総計	74	10	84

表A.5: 記事“ North America ”の各パラグラフの記事文書に対する Goolgle Translate を用いた英日機械翻訳における訳語選択の非一貫性に基づく名詞単語の分類

パラグラフ名	(3)	(4)	(5)	総数 (1)+(2)
North America	1	0	0	1
Etymology	1	0	0	1
History	2	0	0	2
Geography and extent	2	1	1	4
Countries and territories	2	0	0	2
Communications	0	0	0	0
総計	8	1	1	10

表A.6: 記事“ North America ”の各パラグラフの記事文書に対する Goolgle Translate を用いた英日機械翻訳において訳語選択システムにより選択された一貫した訳語が文脈に適しているか否かに基づいた名詞単語の分類

パラグラフ名	(1)	(2)	総数 (1)+(2)
South America	3	1	4
History	38	8	46
Politics	2	0	2
Geography	14	1	15
Demographics	13	1	14
Economy	3	0	3
Tourism	0	0	0
Culture	25	1	26
総計	98	12	110

表 A.7: 記事“ South America ”の各パラグラフの記事文書に対する Goolgle Translate を用いた英日機械翻訳における訳語選択の非一貫性に基づく名詞単語の分類

パラグラフ名	(3)	(4)	(5)	総数 (1)+(2)
South America	1	0	0	1
History	7	1	0	8
Politics	0	0	0	0
Geography	1	0	0	1
Demographics	1	0	0	1
Economy	0	0	0	0
Tourism	0	0	0	0
Culture	1	0	0	1
総計	11	1	0	12

表 A.8: 記事“ South America ”の各パラグラフの記事文書に対する Goolgle Translate を用いた英日機械翻訳において訳語選択システムにより選択された一貫した訳語が文脈に適しているか否かに基づいた名詞単語の分類

パラグラフ名	(1)	(2)	総数 (1)+(2)
Antarctica	6	0	6
History	10	2	12
Geography	17	3	20
Geology	23	4	27
Climate	7	2	9
Population	12	0	12
Flora and fauna	18	4	22
Politics	8	2	10
Economy	7	0	7
Transport	2	0	2
Research	15	3	18
Ice mass and global sea level	11	1	12
Effects of global warming	8	0	8
Ozone depletion	5	2	7
総計	149	23	172

表A.9: 記事“ Antarctica ”の各パラグラフの記事文書に対する Goolgle Translate を用いた英日機械翻訳における訳語選択の非一貫性に基づく名詞単語の分類

パラグラフ名	(3)	(4)	(5)	総数 (1)+(2)
Antarctica	0	0	0	0
History	0	2	0	2
Geography	3	0	0	3
Geology	4	0	0	4
Climate	1	1	0	2
Population	0	0	0	0
Flora and fauna	3	1	0	4
Politics	2	0	0	2
Economy	0	0	0	0
Transport	0	0	0	0
Research	3	0	0	3
Ice mass and global sea level	1	0	0	1
Effects of global warming	0	0	0	0
Ozone depletion	1	0	1	2
総計	18	4	1	23

表A.10: 記事“ Antarctica ”の各パラグラフの記事文書に対する Goolgle Translate を用いた英日機械翻訳において訳語選択システムにより選択された一貫した訳語が文脈に適しているか否かに基づいた名詞単語の分類

パラグラフ名	(1)	(2)	総数 (1)+(2)
Europe	13	2	15
Definition	11	1	12
Etymology	3	3	6
History	55	6	61
Geography and extent	18	2	20
Climate	4	1	5
Geology	8	4	12
Biodiversity	22	2	24
Demographics	10	4	14
Political geography	0	1	1
Economy	22	2	24
Language	7	0	7
Religion	3	0	3
Culture	1	0	1
総計	177	28	205

表A.11: 記事“ Europe ”の各パラグラフの記事文書に対するGoolgle Translateを用いた英日機械翻訳における訳語選択の非一貫性に基づく名詞単語の分類

パラグラフ名	(3)	(4)	(5)	総数 (1)+(2)
Europe	2	0	0	2
Definition	1	0	0	1
Etymology	3	0	0	3
History	6	0	0	6
Geography and extent	1	1	0	2
Climate	1	0	0	1
Geology	4	0	0	4
Biodiversity	1	0	1	2
Demographics	2	1	1	4
Political geography	1	0	0	1
Economy	1	1	0	2
Language	0	0	0	0
Religion	0	0	0	0
Culture	0	0	0	0
総計	23	3	2	28

表A.12: 記事“ Europe ”の各パラグラフの記事文書に対するGoolgle Translateを用いた英日機械翻訳において訳語選択システムにより選択された一貫した訳語が文脈に適しているか否かに基づいた名詞単語の分類

パラグラフ名	(1)	(2)	総数 (1)+(2)
Australia (continent)	9	1	10
Geography and nomenclature	5	2	7
Geology	1	0	1
Biogeography	15	2	17
総計	30	5	35

表A.13: 記事“ Australia(continent) ”の各パラグラフの記事文書に対する Goolgle Translate を用いた英日機械翻訳における訳語選択の非一貫性に基づく名詞単語の分類

パラグラフ名	(3)	(4)	(5)	総数 (1)+(2)
Australia (continent)	1	0	0	1
Geography and nomenclature	2	0	0	2
Geology	0	0	0	0
Biogeography	2	0	0	2
総計	5	0	0	5

表A.14: 記事“ Australia(continent) ”の各パラグラフの記事文書に対する Goolgle Translate を用いた英日機械翻訳において訳語選択システムにより選択された一貫した訳語が文脈に適しているか否かに基づいた名詞単語の分類

A.3　訳語選択システムの訳語選択例

英語 Wikipedia の世界 7 大陸の各記事の各パラグラフの記事文書に対する Google Translate を用いた英日機械翻訳において，訳語選択システムを用いない場合に，記事内で複数回出現しており，訳語選択の非一貫性問題が生じた名詞単語のうち，“ 訳語選択システムにより文脈に適した一貫した訳語選択が行われている名詞単語 ”，“ 訳語選択システムにより文脈に適切ではない一貫した訳語選択が行われている名詞単語 ”，“ 訳語を一貫させる必要がないにも関わらず，訳語選択システムにより一貫した訳語選択が行われている名詞単語 ”にそれぞれ分類された名詞単語に対する訳語選択の例を以下に示す.

翻訳元文(英): Holding their capital at the great cougar-shaped city of Cusco, the Inca civilization dominated the Andes region from 1438 to 1533.
⇒翻訳先文(日): ホールディングの資本金は、偉大なクーガー-クスコの模様の都市では、インカ文明は1438年から1533年にはアンデス地域を支配した。

翻訳元文(英): About 40 Ma Australia-New Guinea separated from Antarctica, so that latitudinal currents could isolate Antarctica from Australia, and the first ice began to appear.
⇒翻訳先文(日): 約40馬オーストラリア、ニューギニア南極大陸から分離さように、緯度電流オーストラリアから、最初の氷の南極大陸を分離できるようになった。

図 A.1: 訳語選択システムにより文脈に適した一貫した訳語が選択されるようになった文の例

翻訳元文(英): During the Nimrod Expedition led by Ernest Shackleton in 1907, parties led by T. W. Edgeworth David became the first to climb Mount Erebus and to reach the South Magnetic Pole.
⇒翻訳先文(日): ニムロッド遠征中にアーネストシャクルトンが1907年に、パーティー、トレッド幅エッジワースデビッド率いるとエレバス山に登るには南磁極に到達する第一になったなった。

翻訳元文(英): With the fall of communism in Eastern Europe in 1991 the Eastern states had to adapt to a free market system.
⇒翻訳先文(日): 1991年に東ヨーロッパにおける共産主義の東欧州自由市場経済体制に適応していたの秋と。

図 A.2: 訳語選択システムにより文脈に適さない一貫した訳語が選択されるようになった文の例

翻訳元文(英): Of Europe's approximately 50 states, Russia is the largest by both area and population, while the Vatican City is the smallest.
⇒翻訳先文(日): 一方、バチカン市国は最小のはヨーロッパでも約50カ国のうち、ロシアは、両方の面積と人口で最大です。

翻訳元文(英): In addition, people living in insular areas such as Ireland, the United Kingdom, the North Atlantic and Mediterranean islands and also in Scandinavia may routinely refer to continental or mainland Europe simply as Europe or the Continent.
⇒翻訳先文(日): 加えて、人々は孤立した地域でアイルランドなどの生活は、イギリスでは、大西洋と地中海の島々 、またスカンジナビアでは日常的に大陸やヨーロッパ大陸にヨーロッパや大陸単に参照することがあります。

図 A.3: 訳語を一貫させる必要がないにも関わらず訳語選択システムにより一貫した訳語が選択されるようになった文の例