Same Language Translator

—Reformulating Complex Expressions into Plain Words—

March 2021

201913580

Ai Miyuki

University of Tsukuba, School of Informatics

College of Knowledge and Library Sciences

Abstract

In recent years, the population of foreign nationals in Japan has rapidly increased. According to the Organisation for Economic Co-operation and Development (OECD), each year approximately 100,000 foreign-born individuals have become resident in Japan on a long-term or permanent basis in recent years. As of 2019, there were 2.93 million foreign nationals living in Japan, according to the National Statistics Center. Given Japan's labor shortages brought about by an aging population, the number of foreign-born nationals in Japan is expected to continue to increase.

Often, the Japanese language proficiency of foreign nationals is insufficient to understand text or instructions intended for native speakers. While translation tools between languages can offer a partial solution, given the sheer number of target languages good coverage is not always ensured. Plain Japanese has been developed as a simplified subset of Japanese in an effort to make it more intelligible to Japanese learners, bypassing coverage issues.

In this thesis, we first explore the benefits and limitations of plain Japanese with a survey of Japanese learners. Then, we describe a methodology to automatically transform regular Japanese text into plain Japanese by reformulating complex expressions into plain words. Finally, we implement Same Language Translator (SLT) — a prototype of this methodology — and provide an evaluation including both success and failure cases of our prototype.

Contents

Chapter 1

Introduction

The number of non-native Japanese speakers has been increasing in recent years in Japan. According to the National Statistics Center, there are now 2.93 million foreign nationals living in Japan in 2019 [5]. Given that Japan has been suffering from labor shortages because of its aging population, the population of non-native speakers in Japan is expected to keep increasing in the future. As reported by OECD, there have been approximately 100 thousand new foreign-born residents both on a long-term or permanent basis, which was 4.3% more than in 2016. This figure is made up of 53.5% labour migrants, 30.1% family members, 0.1% humanitarian migrants, and 16% other migrants. In terms of nationality, China, Viet Nam, and the Philippines were the top three countries of foreign-born residents in 2017 [1].

However, many non-native Japanese speakers struggle to communicate with native Japanese speakers both in Japanese and English [28, 4]. Often, where multiculturalism is addressed, it tends to focus on translating one language into another language for communication. Although this approach does have clear benefits, it also comes with its limitations. The main issue is that it requires the translation to be done in a large number of target languages, which can be impractical. For example, in the case of a spoken announcement, the announcement time will proportionally increase with the number of translated languages.

As opposed to the translation approach, a study indicates that if non-native Japanese speakers consciously learn and use plain Japanese to non-native Japanese speakers, they are more likely to understand each other [26]. Plain Japanese increases the probability that non-native speakers understand a message the first time they read it; thus, the point of plain Japanese is to communicate clearly and concisely with them. To accomplish this, it avoids difficult words and replaces them with a simpler equivalent word that is more likely to be found in day-to-day conversation. Plain Japanese tries, whenever possible, to enable a non-native Japanese speaker to comprehend a message without having to look up words in a dictionary. The number of Japanese words used in plain Japanese is approximately 2,000, which is about level 3 of the Japanese Language Proficiency Test for Foreigners, also known as

JLPT [59]. Those words also can be understood by a lower grade of elementary school students [27]. Finally, the length of a sentence can have a direct negative impact on its readability [19]. For this reason, keeping every sentence to a minimum length is extremely important, as well as avoiding wordy and formal phrases [29].

In this thesis, we first look into how plain Japanese can help non-native speakers to better understand Japanese. To do so, we conduct a survey in which we ask non-native respondents to read two texts and answer questions about them. We provide details about this survey in Chapter 3. We then propose a hybrid rule-based and data-driven methodology to transform complex Japanese expressions into plain Japanese by leveraging several datasets such as Wikipedia data. We implement our methodology in Python and make our prototype publicly available online[1]. We expand on our methodology and implementation in Chapter 4. We evaluate our methodology on texts from various sources such as news articles or announcements and describe in which ways our system succeeds or not to simplify the input text. Our experiments are described in Chapter 5. We then write about the related work in Chapter 6 before concluding and proposing future work in Chapter 7.

[1] https://slt.aimiyuki.me/

Chapter 2

Background

In this chapter, we first provide the relevant background about the Japanese language that motivates and helps understand the work done in this thesis. Then, we describe the different natural language processing concepts used across our work and explain some of the challenges faced when applying these to the Japanese language.

2.1 Japanese Language and Plain Japanese

Japanese is the official language in Japan. It is a member of the Japonic language family and is made up of several thousand Chinese characters known as Kanji, along with two different alphabets: Hiragana and Katakana. Both are phonetic alphabets and have 46 characters each. Most Kanji have more than one pronunciations, depending on whether they refer to Japanese pronunciation or Chinese pronunciation. Although there are regular rules, the pronunciation of a Kanji can sometimes be ambiguous.

It is said that over 10,000 Japanese words are required to achieve 90 percent coverage in Japanese. Compared to many other languages, this is a considerably large amount of vocabulary [22].

One of the reasons why Japanese requires extensive vocabulary is that it combines different types of words: Japanese words, words of foreign origins and hybrid words. For instance, there is “カステラ” which refers to “Castella,” one of the most popular Japanese sweets, and this name is derived from a Portuguese word meaning “Bread from Castile”. An example of a hybrid word is “抹茶ティーラテ” which is formed by combining two words, a Japanese word “Matcha” and an English word “Tea latte” for “Green tea latte.”

Furthermore, Japanese has several grammatical forms to express politeness, making it even more complicated. Usually, these grammatical forms are determined by the speakers’ positioning of themselves within the Japanese society. Most of the time, this position is established by various factors, essentially, age, job title, and so forth. The speaker in the lower position is expected to opt for a politer form, whereas the other can choose to use less polite forms.

2.1.1 Japanese-Language Proficiency Test

The Japanese-Language Proficiency Test (JLPT) is a standardized written test for the Japanese language learners to evaluate and validate their Japanese proficiency, organized by the Japan Foundation and Japan Educational Exchanges and Services [59]. JLPT is offered at five levels from N1 through N5, with the N1 being the hardest. None of the levels include an oral test. Specifically, passing the N5 shows that the person has the ability to comprehend basic sentences in Japanese. N4 is a benchmark proficiency level for familiar daily topics and N3 for the written materials with specific contents concerning everyday topics. In contrast, the N1 verifies that the person can read difficult content, such as newspaper articles, magazines, or books.

2.1.2 Plain Japanese

Plain Japanese is a form that helps non-native Japanese speakers to understand a message the first time they read it; thus, one of the goals of plain Japanese is to communicate clearly and concisely with non-native Japanese speakers. To accomplish this, it avoids difficult words and uses synonyms that are considered to be more intelligible to a non-native speaker. For example, “右折”, a compound word meaning “to turn right” could be replaced by its decomposed form “右に曲がる”, which is more approachable to a non-native. The JLPT N3 [21] is often used as a reference point to determine if a word is simple enough or not to be included in plain Japanese. Another important feature of plain Japanese is the length of the sentence. Long sentences are known to be more difficult to understand by non-native [19], therefore, plain Japanese also tries to keep sentences short when possible.

2.2 Natural Language Processing

In this section, we define the fundamental concepts of natural language processing (NLP) necessary to understand this thesis. For each section, we give a brief overview of the underlying concept and point the reader towards the relevant literature.

2.2.1 Tokenization

Tokenization is the process of splitting a sentence, which usually comes in the form of a single string, into a list of strings that each represent a single token of the sentence. There is no formal definition of how a sentence should be tokenized but the most common way to do so is to assign each word in the sentence to be a single token. For languages such as English or French, where words are delimited with a white space or punctuation, the tokenization process is relatively easy and can typically be achieved using regular expressions.

However, in scriptio continua languages — i.e. languages without white space delimiters — such as Japanese [60], the process is non-trivial, as in most cases, there is no easy way to determine word boundaries. While some early tokenizers [16] were rule-based, most techniques now use a combination of dictionaries and machine learning algorithms. For example, one of the most commonly used Japanese tokenizer, Mecab [35], uses conditional random fields [37] to infer boundaries and tokenize the sentence. Another well-known tokenization software, ChaSen [41], uses hidden-Markov models [51] while Juman++ [60], a more recent work, uses Recurrent Neural Network (RNN)-based language models [45].

2.2.2 Part-of-Speech Tagging

Part-of-speech (PoS) is the grammatical category of a particular word in the sentence. Most languages share similar PoS, such as verbs, nouns, adjectives or yet adverbs. While determining these PoS is a relatively easy task for a knowledgeable person, it is non-trivial to automate as there are often no clear rules to determine the PoS of a particular word. The process of assigning PoS to words in a sentence is known as PoS tagging and has been studied extensively in the literature [53, 11, 52]. As for Japanese tokenization, PoS has been shown to work better using machine learning techniques. Among other techniques, maximum entropy cyclic dependency network [62] has been popularized by the Stanford CoreNLP natural language processing toolkit [39], a well-known NLP toolkit. At the time of writing this thesis, the state-of-the art is achieved by the NLP toolkt Flair [6], using Bidirectional LSTM-CRF models [20].

In this thesis, we rely on the Spacy toolkit [14] using a multitask convolutional neural network [34] pre-trained on the Universal Dependencies Japanese treebank [57].

2.2.3 Dependency Parsing

Dependency parsing is the process of inferring the dependency relationships between words in a sentence [58]. There are many types of dependency relationships that can exist among words in a sentence and these are usually expressed as a tree of which the root is the verb of the sentence's main clause. Typically, subjects and objects are both be dependencies of the verb, while pronouns and articles are dependencies of the word they are used in conjunction with. Again, machine learning models are now ubiquitous for this task, with neural-network based models achieving state-of-the-art performance [12].

2.2.4 Unigrams, Bigrams and n-gram

A unigram can be thought of as a single word, a bigram is a sequence of two words and n-gram is the generalization of this concept: a sequence of n adjacent words. For example, given the sentence “The brown fox jumps”, “the”, “brown”, “fox” and “jumps” are all unigrams while there are three bigrams: “the” and “brown”, “brown” and “fox”, “fox” and “jumps”. N-grams can be used to create language models [47, 49]: models that assign an occurrence probability to a sequence of word. The most common way to do so is to count all the n-grams in a given corpus and to define the probability of a certain n-gram to occur to be the number of times it occurs divided by the total number of n-grams in the corpus. A more general language model can then be derived by assigning the probability of a sequence of word to be the product of the probability of each of its n-gram.

2.2.5 Word Embeddings

Embedding words is the process of assigning vectors, typically in $\mathbb{R}^d$ where d is an arbitrary number of dimensions, to words [43]. To understand why this is helpful, we first need to understand how words are typically represented in the context of NLP. Given a vocabulary of $|V|$ words, a single word is often represented as a vector in $\mathbb{N}^{|V|}$ where a single element, the index of the word, is set to 1 and all the others are set to 0. Such a vector is often called a one-hot vector [9]. This means that to represent a sentence of n tokens, a matrix of $\mathbb{N}^{s \times |V|}$ is required. While the memory overhead resulting of such a matrix can be overcome easily with sparse matrices [13], it makes it significantly harder for machine learning models to learn given the large dimensions and sparsity of the domain.

To overcome this, word embeddings have been developed. Assigning a dense vector to a word reduces its dimension from $|V|$, which is typically in the order of 10 or 100 thousand, to an arbitrary d in the order of a few hundreds. Another important feature of these embeddings is that they are generated in such a way that the resulting vectors have useful semantic properties [43, 50] such as synonyms having small distances in the resulting vector space.

Chapter 3

Survey

Before starting the implementation of our work, we performed a survey to validate the utility of plain Japanese and better understand what are the main challenges faced by non-native Japanese speakers when interacting with the Japanese language. In this chapter, we present our survey and highlight our main findings.

3.1 Survey Structure

We designed our survey in the following way: we chose two articles from NHK [3] and their equivalent in plain Japanese from NHK plain Japanese [2]. The first article is written about "Survey of Awareness on promoting plain Japanese in multicultural society". The second article is about "The declining number of doctoral students in recent years in Japan". Both articles are approximately 700 characters long, including more or less 200 of Kanji. We then designed four short questions, 2.5 points each, summing to 20 points. We asked the participants to read the texts, respond to the questions and also to provide some background about their familiarity with Japanese.

3.2 Individual Participants

In this section, we describe the different participants of our survey and discuss their responses. In Table 3.1 we present an overview of the participants. We have four native-French participants and two native-Chinese participants, who have learned Japanese for periods ranging from 1 to 15 years.

3.2.1 Participant 1

Participant 1 is a native French speaker who has studied Japanese for 15 years. His answers show that he understood the plain Japanese better than the standard one. What he has written was close to the correct answer. However, his answer in the first standard version was slightly off toppic. Surprisingly, he even responded ”I don’t understand any of this” in the second

ID	**Native language**	**Learning period**	**JLPT level**	**Text 1 scores**		**Text 2 scores**	
				Complex	Plain	Complex	Plain
1	French	15 years	3	2	5	1	5
2	French	4 years	3	2	5	0	4
3	French	3 years	n/a	2	5	0	5
4	French	4 years	n/a	5	5	3	5
5	Chinese	5 years	1	5	5	5	5
6	Chinese	1 years	3	5	5	5	5
Avg.	-	-	-	**3.5**	**5**	**2.3**	**4.8**

Table 3.1: Summary of survey participants

standard Japanese article, whereas he had no problem answering in the plain Japanese one.

3.2.2 Participant 2

Participant 2 is a native French speaker who has studied Japanese for 4 years. His answers show that reading Japanese articles for him is not easy. However, despite the fact that he actually could not answer most of the questions about standard versions, he had no problem answering questions about plain versions.

3.2.3 Participant 3

Participant 3 is a native French speaker who has studied Japanese for 3 years. His answers about the plain Japanese article were perfect, while his answer about the standard version was a little off from the actual questions, he was able to answer questions about the plain version clearly, even though he answered that “I have no idea” in standard versions.

3.2.4 Participant 4

Participant 4 is a native French speaker who has studied Japanese for 4 years. His level of understanding was overall high. He answered,“I do not understand” in response to the question about the second standard version, while all the answers about the plain Japanese articles were perfect.

3.2.5 Participant 5

Participant 5 is a Chinese native speaker and has learned Japanese for 5 years. His overall comprehension was perfect, as both his answers to the standard and plain Japanese were explained in details and written very clearly. Besides, he even suggested that “The plain Japanese version lacks so much information due to its short length. Conversely, the standard Japanese version felt easier for him to grasp what the whole story is really about.”.

3.2.6 Participant 6

Participant 6 is a native Chinese speaker who has studied Japanese for a year. From his answers, we can tell that his understanding of Japanese is very high despite his short learning period. All of his answers were correct and detailed as to why he answered as he did.

3.3 Summary

Overall, we have found that plain Japanese improves the learners’ understanding of the language. As we see in Table 3.1, the average score for plain Japanese is higher for both texts, with more than twice the score for the second text. Interestingly, the difference in the different level of Japanese comprehension between learners, native French speakers and native Chinese speakers was obvious. Chinese speakers had an equally high level of comprehension regardless of the difficulty of the articles, while French speakers had a much higher level of comprehension for the article in plain Japanese than for the standard version. We hypothesize that this is mainly because there are no common characters between French and Japanese, whereas there are extensive amounts of Japanese Kanji based on Chinese Kanji.

To summarize, we find that for the particular case of Chinese-speakers, who can understand a majority of the symbol from their native language, plain Japanese is not necessary. However, for non-native speakers who do not share Chinese characters, plain Japanese increases significantly the level of understanding.

Chapter 4

Methodology

In this chapter, we describe the methodology that we used to develop our Same Language Translator system. We first give an overview of the system and then describe in more detail the different algorithms. Finally, we present the implementation and datasets used in this thesis.

4.1 Overview

We first give an overview of how we transform sentences. The transformation happens using the following steps, which will be detailed in the following sections.

Step 1. Tokenization: we tokenize our input into its token components

Step 2. Difficulty assessment: we assess the difficulty of each token in the sentence

Step 3. Synonym search: we search if we have an easier replacement for the given token

Step 4. Replacement range identification: we look for the range of tokens that needs to be replaced

Step 5. Replacement denormalization: we denormalize the replacement target to adapt features such as the tense to the initial token set

Step 6. Sentence reconstruction: we reconstruct the original sentence using the tokens from the original sentences and the replaced ones

We show an overview diagram of the process in Figure 4.1. In this diagram, we use the sentence “本日は友人とお食事したあとに、大学に参った”, meaning “Today, I had a meal with my friends, after which I went to university” and show every transformation steps to finally become into “今日は友達と食べたあとに、大学に行った”.

We note several particularities of the original sentence that make it hard for a non-native speaker to understand. First, the word used for “today”, “本日”

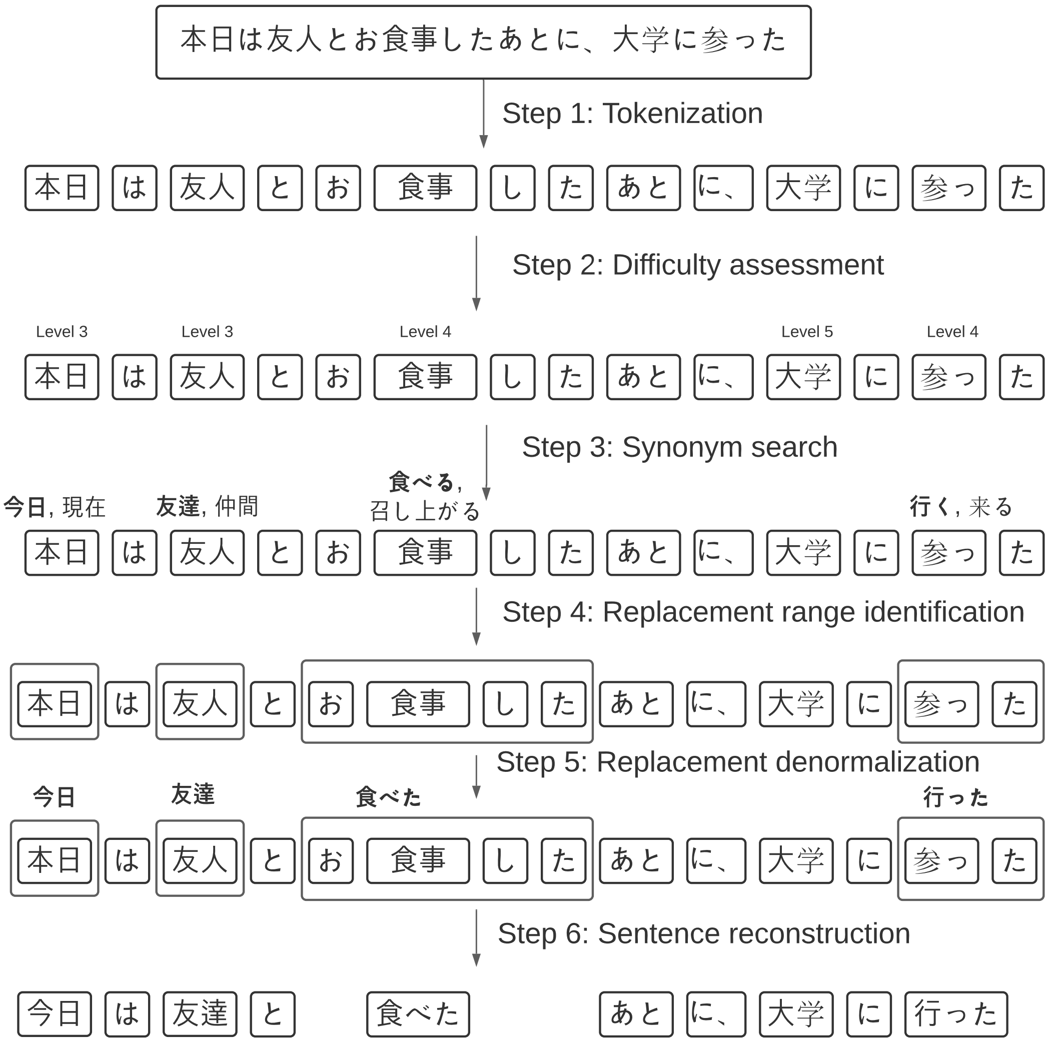

本日は友人とお食事したあとに、大学に参った
Step 1: Tokenization
本日 は 友人 と お 食事 し た あと に、 大学 に 参っ た
Step 2: Difficulty assessment
Level 3
Level 3
Level 4
Level 5
Level 4
本日 は 友人 と お 食事 し た あと に、 大学 に 参っ た
Step 3: Synonym search
今日, 現在
友達, 仲間
食べる, 召し上がる
行く, 来る
本日 は 友人 と お 食事 し た あと に、 大学 に 参っ た
Step 4: Replacement range identification
本日 は 友人 と お 食事 し た あと に、 大学 に 参っ た
Step 5: Replacement denormalization
今日
友達
食べた
行った
本日 は 友人 と お 食事 し た あと に、 大学 に 参っ た
Step 6: Sentence reconstruction
今日 は 友達 と 食べた あと に、 大学 に 行った

Figure 4.1: Overview of the sentence transformation

is formal and typically not used in daily conversations. It is replaced by “今日”, which is the most common way of saying “today” in Japanese. Next, the word “友人” is also rather formal and is replaced by its more regular counterparty, “友達”. The next transformation of the verb “お食事する”, meaning “to have a meal”, is a little trickier: by itself, the token “食事” would translate into “meal”, which is a noun. This makes this transformation context-sensitive: we need to look at not only the token itself but its part-of-speech, to be able to look for the correct replacement.

The replacement found is the verb “食べる”, meaning “to eat”. To replace the verb, our algorithm identifies the dependencies of the “食事” token, namely “お” and “した” and marks it as the range of tokens to replace. Further, this verb is conjugated in the past tense, so a simple replacement using the lemma would modify the semantics of the sentence. We identify the tense of the verb and conjugate the replacement verb accordingly, resulting into the replacement tokens “食べた”. The last replacement of “参った”, being the polite form of “to go”, is performed in a similar manner and results in “行った”, which is the regular form of the same verb.

We will now describes the different steps in more details and justify some of the design decisions of our methodology. We leave the tokenization part to the implementation details, as our work relies on well-known softwares [14, 35].

4.2 Difficulty Assessment

Assessing the difficulty of a sentence, or even part of it, is a difficult task, as has been seen in several previous work [19, 38]. In this work, we try to estimate the difficulty at a token-level rather than at a sentence-level, which makes this task slightly simpler. To do so, we combine two orthogonal approaches: a qualitative and a quantitative approach. In our qualitative approach, we use the JLPT words [23] as a ground-truth to assess the difficulty. This gives us a score from 1 being the most difficult to 5 being the easiest. We never consider words of level 5 as replacement targets as we consider that there are no simpler words. In our quantitative approach, we use the unigram and bigram frequency of the tokens [24] and consider very rare words to be difficult. We define a word to be rare if it appears at least n times less than its closest synonym, where n times is a hyper-parameter of our system. However, given that our dataset, which we will describe further later, is taken from Wikipedia, the distribution of words will tend towards written and formal expressions. Therefore, we do not necessarily consider more frequent words to be simpler when the above property is not fulfilled.

If we look at our previous example and focus on the word “友人”, we find that in terms of JLPT level, it is at level 3, meaning that it is likely not known by a beginner. On the other hand, the word “友達” that is used as a replacement is in the level 5 of the JLPT classification, which means that even a very beginner should know the word. However, the word “友達” appears only 10,303 times in our unigram dataset while the word “友人” has 27,643 occurrences. This is easily explainable by the fact that, while both words are synonyms, “友人” is more appropriate when writing, while “友達” is more often in informal conversations. On the other hand, if we compare the verbs “行く”, identified as level 5, and “参る”, identified as level 4, we see that the former has 29,920 occurrences while the latter only has 2,749. Therefore, the word “参る” fits in both our replacement rules.

4.3 Synonym Search

Once we have a word that we want to replace, we need to find a proper synonym to replace it. As in the previous step, we use a combination of a hand-crafted dataset and a larger, computer generated dataset to look for a synonym. As a hand-crafted dataset, we use a version of WordNet [15] that contains Japanese synonyms [10]. However, naively using a word from the synonyms list given by WordNet results in very sub-optimal results. There are several reasons for this: WordNet is not perfect and sometimes has some errors [25] but more importantly, some words could be marked as synonym but not be used in the same context in a sentence. The first simple improvement that we perform is that we only choose synonyms within wordnet that have the same part-of-speech as our target word. Next, to filter out-of-context words, we again use data from Wikipedia but this time, we use word embeddings [44] generated by FastText [31], an implementation of Word2vec’s continuous bag-of-word (CBOW) model [42].

To compute the replacement of a word, we apply to Equation 4.1, where we note w_1 the word to replace, w_1' the replacement of w_1, S_{w_1} the set of synonyms for w_1, t an arbitrary threshold value and sim(w_1, w_2) the cosine similarity between w_1 and w_2.

$$w_1' = \begin{cases} w_1 & \text{if } \max\limits_{w' \in S_{w_1}} \text{sim}(w_1, w') < t \\ \arg\max\limits_{w' \in S_{w_1}} \text{sim}(w_1, w') & \text{otherwise} \end{cases} \tag{4.1}$$

We can see that Equation 4.1 that the result is dependent on the threshold t, which is another hyper-parameter of our system. Intuitively, this threshold means that if the closest synonyms we found has a very low similarity to the target word, the original word should be kept.

Finally, we use a heuristic approach to avoid wrongly decomposing compound words. For example, the word “者”, meaning in some contexts “person” could in many cases be replaced by “人”, which is the simplest way of expressing a person in Japanese. However, in the context of a compound word such as “関係者”, it would be incorrect to replace “者” by “人”. To avoid this problem, we use our ngram dataset and check if the number of bigrams is m times smaller when replacing a token by its synonym, where m is a hyper-parameter of our system. If this is the case, we discard the synonym. For this particular

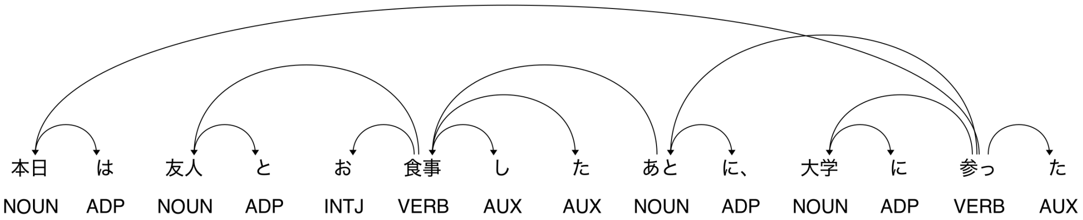


Figure 4.2: Dependencies of our sample sentence

Part-of-Speech	Rule
Noun or Verb	Include the honorific prefix, which is a “お” or “ご” dependency directly at the left of the token
Verb	Include the conjugation, which is a dependency directly at the right of the token
Adjective	Include the conjugation, which is a dependency directly at the right of the token

Table 4.1: Rules to find the replacement range

example, there are 22,357 bigrams for “関係者” while only 180 for “関係人”, which makes it clear that “者” should not be replaced.

4.4 Replacement Range Identification

To identify the range of tokens that needs to be replaced, we use dependency parsing [36] in combination with several heuristics that we will describe here.

In the first step, we find the dependencies of each token we need to replace. We show the dependencies of our sample sentence in Figure 4.2. We can see that “食事”, which is a verb that we want to replace, has “友人”, “お” and “した” as direct dependency.

In this particular case, all the dependencies but “友人” should be replaced with the verb, as “お” is an honorific particle and “した” is the conjugation of the verb. In Table 4.1, we list the rules that we used to identify the replacement range.

We note that the rule for suffix computation needs to be adjusted to cases where part of the suffix needs to stay, for example in the case of a conjunction with “が” or “けど”. However, this part is left to the next step of the pipeline that we will describe in the next section.

4.5 Replacement Denormalization

Synonyms are looked up in normalized form, i.e. using their lemmas, meaning that in the above example, “お食事した” will be looked up as “食事” and the

Algorithm 1 Sentence Reconstruction Algorithm

```
Input: $original, synonyms$
Output: $simplified$
  $simplified \leftarrow ()$                ▷ create an empty sequence for simplified sentence
  $seen \leftarrow \{\}$                    ▷ empty set of seen indices
  for $(index, token) \in original$ do
      if $index \notin seen$ then
          if $index \in synonyms$ then
              $(tokens\_before, tokens\_after, replacement) \leftarrow synonyms[index]$
              $simplified \leftarrow drop\_right(simplified, tokens\_before)$
              $seen \leftarrow seen \cup \{j \mid j \in [index, index + tokens\_after]\}$
              $simplified \leftarrow simplifed \bigoplus replacement$     ▷ append replacement
          else
              $simplified \leftarrow simplifed \bigoplus token$
          end if
      end if
  end for
```

resulting synonym will be “食べる”, the lemmatized form of the verb. Once the synonym is found and the range to be replaced has been identified, we need to denormalize it so that it fits in the initial sentence.

The main type of denormalization is conjugation, where we need to first identify the form of the replaced verb or adjective and then to conjugate the replacement to the adequate form. We use a table-based approach [48] to achieve this. The steps are as follow:

1. Strip and save suffixes of the verb to replace (e.g. “べきだ”)
2. Inspect the ending of the verb to replace and identify the tense
3. Inspect the ending of the verb to denormalize and identify its group
4. Use information from the two previous steps and conjugate the verb to denormalize to the same tense as the one identified for the verb to replace
5. Add the suffixes to the conjutaged verb

Although this approach is almost purely rule-based, the rather regular nature of Japanese conjugation makes it possible to cover most cases with a small number of rules.

4.6 Sentence Reconstruction

Once we have computed all the synonyms and their replacement range, we need to reconstruct the new sentence with the replaced tokens. Although this

Description	**Variable**	**Value**
Similarity threshold to consider as replacement	t	0.4
Difference ratio between n-grams to consider a word rare	n	5
Difference ratio between n-grams to reject word	m	10

Table 4.2: List of hyper-parameters used in our final system

process is relatively straightforward, we note that the replacements are not one-to-one, meaning that we need to exclude spans of the initial sentence and include new spans instead, rather than simply tokens. We perform this process iteratively, by looking at each token in the initial sentence and computing the range of the replacement every time. We show the full reconstruction algorithm in Algorithm 1. We note that the synonyms are stored as a triplet containing the number of tokens before and after, as described in Section 4.4 and the replacement. We define $drop\_right$ to be a function that removes n elements at the right of a sequence such that $drop\_right((1,2,3,4),2) = (1,2)$ and note $\bigoplus$ as the append operation to a sequence such that $(1,2,3) \bigoplus 4 = (1,2,3,4)$.

4.7 Implementation

In this section, we give further details about the implementation of the above methodology and the different datasets that we used to build our prototype.

4.7.1 Implementation details

We implement our prototype in approximately 2000 lines of Python and choose spaCy [14] as our NLP toolkit. In particular, we rely on its pre-trained Japanese multi-task CNN that allows for POS tagging and dependency parsing[1]. To implement our web interface, we use the Flask framework[2] and host an online demo at the following address: `https://slt.aimiyuki.me/`.

We note that all the computations performed by our prototype are relatively light-weighted and we are able to run our demo interface with all the required models in memory using roughly 4GB of memory, most of the memory being consumed by the word embeddings.

We list the hyper-parameters that we use in our final implementation in Table 4.2. As mentioned above, the similarity threshold is computed using a pre-trained CBOW model. The difference ratio between n-grams is computed as the number occurrences of an n-gram divided by the number of occurrences the n-gram it is being compared with.

[1] `https://spacy.io/models/ja#ja_core_news_sm`

[2] https://flask.palletsprojects.com/

4.7.2 Datasets

In addition to the pre-trained model of our NLP toolkit, we use the following datasets.

Wikipedia data. We use a full dump of Wikipedia data to generate n-grams of the Japanese language. The raw XML dump is roughly 1.8GB compressed and from this, we generate the 100,000 most frequent unigrams and 300,000 most frequent bigrams.

Common Crawl[3]. We use word vectors generated using the Common Crawl dataset for Japanese to compute the similarity between words. For this task, we use Japanese word vectors trained using CBOW [18].

Japanese Wordnet. We use a version of Wordnet containing Japanese synonyms to look for synonyms [10].

JLPT vocabulary. We use a list of more than 8,000 words present in the JLPT vocabulary to assess the level of each word.

[3] http://commoncrawl.org/

Chapter 5

Experiments

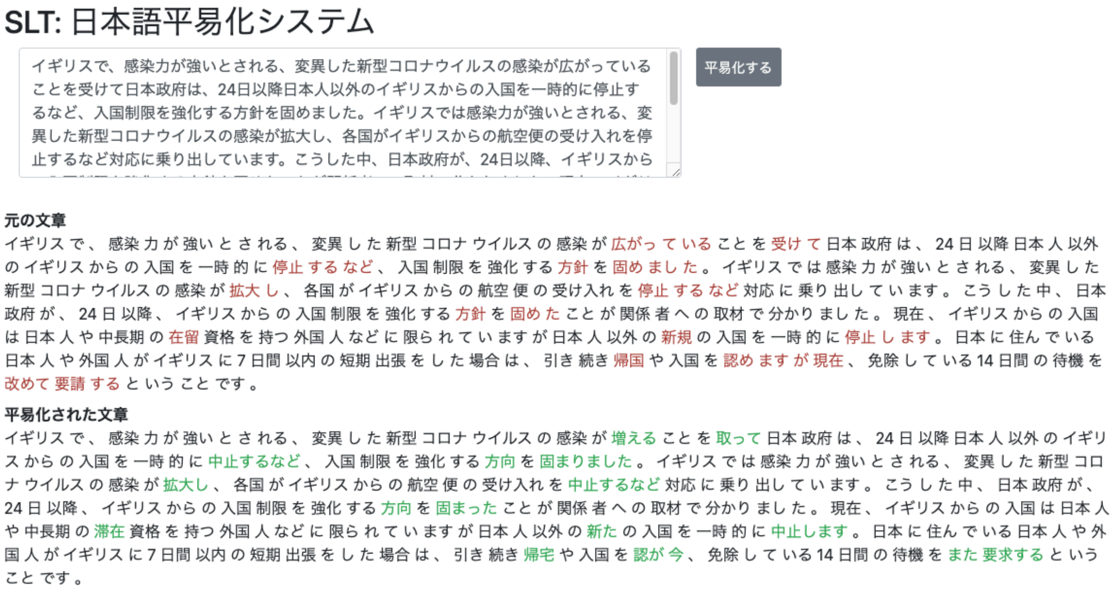
In this chapter, we present the experimentations that we conducted using four different texts to evaluate our system.

We selected four Japanese texts in order to reformulate into plain Japanese using SLT, two articles and two announcements. For the articles, we chose one from NHK and another from TechCrunch Japan. Then, we chose instructions from Tokyo Metro and the Immigration Bureau of Japan. For each text, we report the words that the system correctly or incorrectly translated, at well as some complex expressions that were not replaced.

We provide an overview our results in Table 5.1. The columns are defined as follow: the first column is the source of the article, the next one is the total number of tokens in the article. We show the number of tokens rather than the number of characters as it is the smallest unit that can be replaced by our prototype. The next two columns are the number of replacements performed, the first one being the number of correct ones and the second one the number of incorrect ones, where incorrect usually means that the replacement does not fit correctly in the simplified sentence. The last column is an aggregate of the two previous ones, showing the accuracy of the systems in terms of correct replacements divided by the number of total replacements, expressed as a percentage.

		Replacements count		
Source	**Tokens**	**Correct**	**Incorrect**	**Accuracy**
NHK	236	9	6	60%
TechCrunch	243	17	8	68%
Immigration Services Agency of Japan	240	23	3	88%
Tokyo Metro	65	4	1	80%

Table 5.1: Summary of the results of our system on different texts



SLT: 日本語平易化システム

イギリスで、感染力が強いとされる、変異した新型コロナウイルスの感染が広がっていることを受けて日本政府は、24日以降日本人以外のイギリスからの入国を一時的に停止するなど、入国制限を強化する方針を固めました。イギリスでは感染力が強いとされる、変異した新型コロナウイルスの感染が拡大し、各国がイギリスからの航空便の受け入れを停止するなど対応に乗り出しています。こうした中、日本政府が、24日以降、イギリスから

元の文章

イギリス で 、 感染 力 が 強い と さ れる 、 変異 し た 新型 コロナ ウイルス の 感染 が 広がっ て いる こと を 受け て 日本 政府 は 、 24 日 以降 日本 人 以外 の イギリス から の 入国 を 一時 的 に 停止 する など 、 入国 制限 を 強化 する 方針 を 固め まし た 。 イギリス で は 感染 力 が 強い と さ れる 、 変異 し た 新型 コロナ ウイルス の 感染 が 拡大 し 、 各国 が イギリス から の 航空 便 の 受け入れ を 停止 する など 対応 に 乗り 出し て い ます 。 こう し た 中 、 日本 政府 が 、 24 日 以降 、 イギリス から の 入国 制限 を 強化 する 方針 を 固め た こと が 関係 者 へ の 取材 で 分かり まし た 。 現在 、 イギリス から の 入国 は 日本 人 や 中長期 の 在留 資格 を 持つ 外国 人 など に 限ら れ て い ます が 日本 人 以外 の 新規 の 入国 を 一時 的 に 停止 し ます 。 日本 に 住ん で いる 日本 人 や 外国 人 が イギリス に 7 日間 以内 の 短期 出張 を し た 場合 は 、 引き 続き 帰国 や 入国 を 認め ます が 現在 、 免除 し て いる 14 日間 の 待機 を 改めて 要請 する と いう こと です 。

平易化された文章

イギリス で 、 感染 力 が 強い と さ れる 、 変異 し た 新型 コロナ ウイルス の 感染 が 増える こと を 取って 日本 政府 は 、 24 日 以降 日本 人 以外 の イギリス から の 入国 を 一時 的 に 中止するなど 、 入国 制限 を 強化 する 方向 を 固まりました 。 イギリス で は 感染 力 が 強い と さ れる 、 変異 し た 新型 コロナ ウイルス の 感染 が 拡大し 、 各国 が イギリス から の 航空 便 の 受け入れ を 中止するなど 対応 に 乗り 出し て い ます 。 こう し た 中 、 日本 政府 が 、 24 日 以降 、 イギリス から の 入国 制限 を 強化 する 方向 を 固まった こと が 関係 者 へ の 取材 で 分かり まし た 。 現在 、 イギリス から の 入国 は 日本 人 や 中長期 の 滞在 資格 を 持つ 外国 人 など に 限ら れ て い ます が 日本 人 以外 の 新た の 入国 を 一時 的 に 中止します 。 日本 に 住ん で いる 日本 人 や 外国 人 が イギリス に 7 日間 以内 の 短期 出張 を し た 場合 は 、 引き 続き 帰宅 や 入国 を 認が 今 、 免除 し て いる 14 日間 の 待機 を また 要求する と いう こと です 。

Figure 5.1: Passage from a NHK article

5.1 NHK News Article

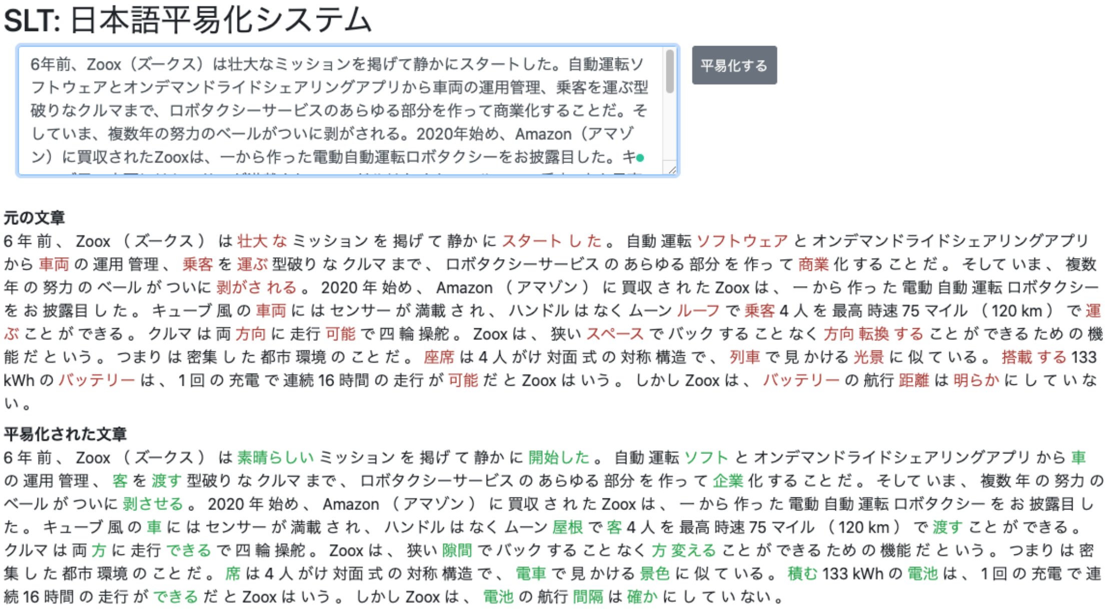
We first test our system with an article taken from NHK news and show the results in Figure 5.1. The article is about the increase of the number of Corona cases in the United Kingdom and is targeted at native Japanese speakers.

In this article, the word "停止する", meaning to stop or to interrupt, comes up often and is successfully replaced by our system by the simpler "中止する" which preserves the meaning of the sentence. We note that the conjugated form "停止します" is also replaced by its conjugated equivalent "中止します". It also successfully transforms "改めて" into "また", which is significantly simpler while conveying the same meaning.

On the other hand, our system wrongly translates some of the words. For example, "受けて" is replaced by "取って". Both words refer to "to take" but in this particular context, "受けて" means to receive, while the replacement "取って" does not fit in. However, both being synonyms, our system performs the replacement as "取って" is deemed simpler and the contextual information is not enough for the synonym to be discarded. Supporting such cases would require to provide more context to our system. Another miss here is the verb "認めます", that is replaced by the token "認". This is likely due to a pending bug in our conjugation module and should, unlike the previous issue, be fixable with little changes to the system.

5.2 Passage from a TechCrunch article

We test our system with an article taken from TechCrunch, shown in Figure 5.2. The article is about a company developing autonomous vehicles. In this article, the word "壮大な", meaning magnificent, was reformulated into



SLT: 日本語平易化システム

6年前、Zoox（ズークス）は壮大なミッションを掲げて静かにスタートした。自動運転ソフトウェアとオンデマンドライドシェアリングアプリから車両の運用管理、乗客を運ぶ型破りなクルマまで、ロボタクシーサービスのあらゆる部分を作って商業化することだ。そしていま、複数年の努力のベールがついに剥がされる。2020年始め、Amazon（アマゾン）に買収されたZooxは、一から作った電動自動運転ロボタクシーをお披露目した。キ

元の文章

6 年 前 、 Zoox （ ズークス ） は 壮大 な ミッション を 掲げ て 静か に スタート し た 。 自動 運転 ソフトウェア と オンデマンドライドシェアリングアプリ から 車両 の 運用 管理 、 乗客 を 運ぶ 型破り な クルマ まで 、 ロボタクシーサービス の あらゆる 部分 を 作っ て 商業 化 する こと だ 。 そして いま 、 複数 年 の 努力 の ベール が ついに 剥がさ れる 。 2020 年 始め 、 Amazon （ アマゾン ） に 買収 さ れ た Zoox は 、 一 から 作っ た 電動 自動 運転 ロボタクシー を お 披露目 し た 。 キューブ 風 の 車両 に は センサー が 満載 さ れ 、 ハンドル は なく ムーン ルーフ で 乗客 4 人 を 最高 時速 75 マイル （ 120 km ） で 運ぶ こと が できる 。 クルマ は 両 方向 に 走行 可能 で 四 輪 操舵 。 Zoox は 、 狭い スペース で バック する こと なく 方向 転換 する こと が できる ため の 機能 だ と いう 。 つまり は 密集 し た 都市 環境 の こと だ 。 座席 は 4 人 がけ 対面 式 の 対称 構造 で 、 列車 で 見 かける 光景 に 似 て いる 。 搭載 する 133 kWh の バッテリー は 、 1 回 の 充電 で 連続 16 時間 の 走行 が 可能 だ と Zoox は いう 。 しかし Zoox は 、 バッテリー の 航行 距離 は 明らか に し て い ない 。

平易化された文章

6 年 前 、 Zoox （ ズークス ） は 素晴らしい ミッション を 掲げ て 静か に 開始した 。 自動 運転 ソフト と オンデマンドライドシェアリングアプリ から 車 の 運用 管理 、 客 を 渡す 型破り な クルマ まで 、 ロボタクシーサービス の あらゆる 部分 を 作っ て 企業 化 する こと だ 。 そして いま 、 複数 年 の 努力 の ベール が ついに 剥させる 。 2020 年 始め 、 Amazon （ アマゾン ） に 買収 さ れ た Zoox は 、 一 から 作っ た 電動 自動 運転 ロボタクシー を お 披露目 し た 。 キューブ 風 の 車 に は センサー が 満載 さ れ 、 ハンドル は なく ムーン 屋根 で 客 4 人 を 最高 時速 75 マイル （ 120 km ） で 渡す こと が できる 。 クルマ は 両 方 に 走行 できる で 四 輪 操舵 。 Zoox は 、 狭い 隙間 で バック する こと なく 方 変える こと が できる ため の 機能 だ と いう 。 つまり は 密集 し た 都市 環境 の こと だ 。 席 は 4 人 がけ 対面 式 の 対称 構造 で 、 電車 で 見 かける 景色 に 似 て いる 。 積む 133 kWh の 電池 は 、 1 回 の 充電 で 連続 16 時間 の 走行 が できる だ と Zoox は いう 。 しかし Zoox は 、 電池 の 航行 間隔 は 確か に し て い ない 。

Figure 5.2: Passage from a TechCrunch article

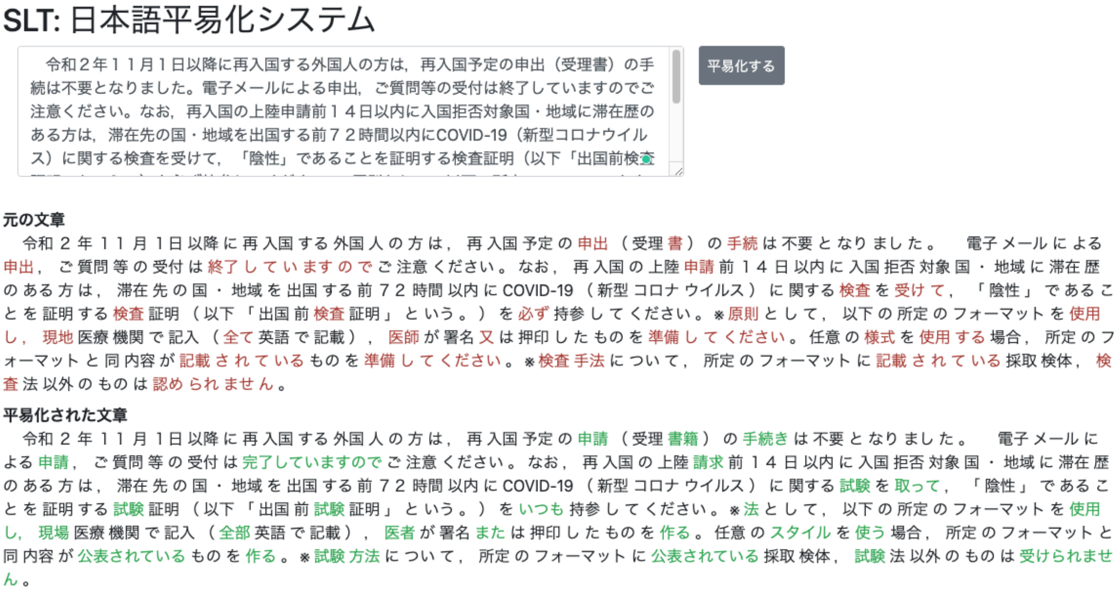
“素晴らしい”, having a similar meaning. We note that “壮大な” is composed of two tokens: “壮大” and “な”, but the dependency was correctly identified by our prototype and replaced accordingly.

A more debatable replacement is the one of “スタート” into “開始”, which both mean start. While the semantic of the sentence is kept, “スタート” would likely be simpler to understand by English speakers while “開始” might be easier for other Japanese learners.

We also note an interesting failure of our prototype. Indeed, the expression “方向転換する”, meaning to change direction, has wrongly been replaced by “方変える”. While this should have been replaced as a single expression, our system has first replaced “方向” by “方” and then replaced “転換する” by “変える”, resulting in a noun and a verb missing a particle between them. This issue could potentially be fixed by improving our sentence reconstruction algorithm to avoid replacing a token that is a dependency of another replacement.

5.3 Announcement at Immigration Services Agency of Japan

We test our system with a part of the instructions regarding travel advice during the pandemic taken from the Immigration Services Agency of Japan’s website and show the results in Figure 5.3. In this announcement, the words“申出”, meaning “request”, was reformulated into “申請”, meaning application. Although both words are non-trivial, the latter is indeed taught earlier than the former. In the same manner, “検査” is replaced with “試験”, which refers to



SLT: 日本語平易化システム

　令和２年１１月１日以降に再入国する外国人の方は，再入国予定の申出（受理書）の手続は不要となりました。電子メールによる申出，ご質問等の受付は終了していますのでご注意ください。なお，再入国の上陸申請前１４日以内に入国拒否対象国・地域に滞在歴のある方は，滞在先の国・地域を出国する前７２時間以内にCOVID-19（新型コロナウイルス）に関する検査を受けて，「陰性」であることを証明する検査証明（以下「出国前検査

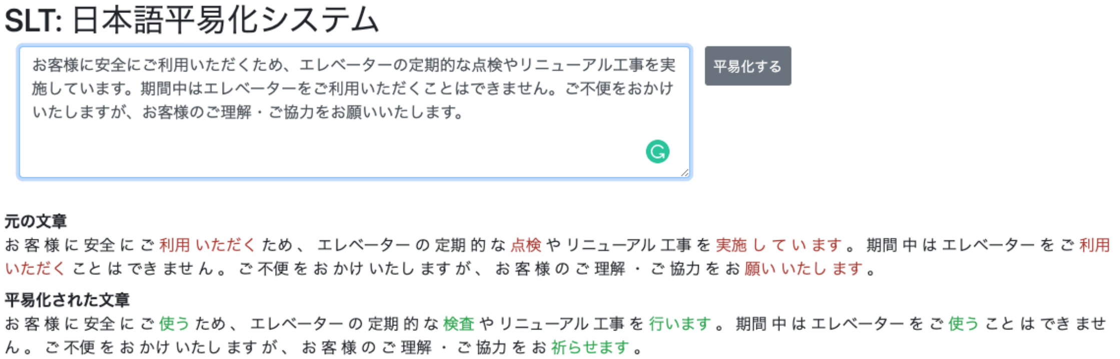
元の文章

　令和２年１１月１日以降に再入国する外国人の方は，再入国予定の申出（受理書）の手続は不要となりました。　電子メールによる申出，ご質問等の受付は終了していますのでご注意ください。なお，再入国の上陸申請前１４日以内に入国拒否対象国・地域に滞在歴のある方は，滞在先の国・地域を出国する前７２時間以内にCOVID-19（新型コロナウイルス）に関する検査を受けて，「陰性」であることを証明する検査証明（以下「出国前検査証明」という。）を必ず持参してください。※原則として，以下の所定のフォーマットを使用し，現地医療機関で記入（全て英語で記載），医師が署名又は押印したものを準備してください。任意の様式を使用する場合，所定のフォーマットと同内容が記載されているものを準備してください。※検査手法について，所定のフォーマットに記載されている採取検体，検査法以外のものは認められません。

平易化された文章

　令和２年１１月１日以降に再入国する外国人の方は，再入国予定の申請（受理書籍）の手続きは不要となりました。　電子メールによる申請，ご質問等の受付は完了していますのでご注意ください。なお，再入国の上陸請求前１４日以内に入国拒否対象国・地域に滞在歴のある方は，滞在先の国・地域を出国する前７２時間以内にCOVID-19（新型コロナウイルス）に関する試験を取って，「陰性」であることを証明する試験証明（以下「出国前試験証明」という。）をいつも持参してください。※法として，以下の所定のフォーマットを使用し，現場医療機関で記入（全部英語で記載），医者が署名または押印したものを作る。任意のスタイルを使う場合，所定のフォーマットと同内容が公表されているものを作る。※試験方法について，所定のフォーマットに公表されている採取検体，試験法以外のものは受けられません。

Figure 5.3: Announcement at Immigration Services Agency of Japan



SLT: 日本語平易化システム

お客様に安全にご利用いただくため、エレベーターの定期的な点検やリニューアル工事を実施しています。期間中はエレベーターをご利用いただくことはできません。ご不便をおかけいたしますが、お客様のご理解・ご協力をお願いいたします。

元の文章

お客様に安全にご利用いただくため、エレベーターの定期的な点検やリニューアル工事を実施しています。期間中はエレベーターをご利用いただくことはできません。ご不便をおかけいたしますが、お客様のご理解・ご協力をお願いいたします。

平易化された文章

お客様に安全にご使うため、エレベーターの定期的な検査やリニューアル工事を行います。期間中はエレベーターをご使うことはできません。ご不便をおかけいたしますが、お客様のご理解・ご協力をお祈らせます。

Figure 5.4: Announcement at Tokyo metro station

examination. Although it might sound slightly awkward to a native speaker, it has a meaning close enough “検査” while being simpler. An interesting failure of our system is “原則として” being replaced by “法として”. Both do include a similar concept of “rule” but the former is an idiom meaning roughly “in principle” and should be kept as is. This should be the job of our n-grams filtering but this does not work as expected because “として” is split in three tokens and our filtering rule only uses up to bigrams. This means that the bigrams that are actually compared are not “原則 として” and “法 として” but rather “原則 と” and “法 と”. Furthermore, “法 と” could be interpreted as “law and” and is therefore a rather common bigram. After checking our data, we find that the “原則 と” appeared 9,123 times, while “法 と” appeared 6,923, which is enough for it not to be discarded.

5.4 Announcement at Tokyo Metro Station

In Figure 5.4, we show the results of our system on a passage regarding the travel information from the Tokyo Metro's website.

The first replacement is interesting as it is a non-trivial one. The word "利用" is a noun meaning "usage" but in conjunction with "いただく", it is a polite way of saying "to use". Our system has detected the usage of "利用" as a verb and has chosen a simpler synonym: "使う". Then, it correctly marked "いただく" as part of the replacement range and replaced the whole expression correctly. Our system also correctly replaced "実施しています" in the simpler "行います" successfully. However the final verb "お願いいたします", which is typically used for a request, was wrongly reformulated into "祈らせます", which means to pray. The first verb do has a meaning somewhat close but is hard to think of as a synonym. While we would have expected our similarity filter to discard this replacement, it appears that these two words have a similarity score of approximately 0.47, which is higher than our value of $t = 0.4$. This could of course be fixed by increasing t to a value such as 0.5 but this could result in some valid synonyms being wrongly filtered and therefore needs to be adjusted with care.

5.5 Summary

We have tested our system on four different texts, two articles and two announcements. We have seen that overall, our system has a much higher number of correct replacements than incorrect ones, with 23 correct to 3 incorrect, or 88% accuracy, in the best case and 9 to 6, or 60% accuracy, in the worst case. We have seen that our system works generally well for words which are not highly dependent on their context and can be replaced by a synonym with little impact on the meaning of the sentence. However, even in such cases, we have seen that the choice of synonyms is sometimes sub-optimal, resulting in outputs that are not natural or sometimes even wrong. Improving on this would require either to increase the required similarity for replacements by increasing our threshold t, which as mentioned above, could result in missing some replacement, or by using a better curated set of synonyms. The other main issue that we have observed is the lack of ability of our system to properly observe the context of the words it replaces. Although we do look at the bigrams, we notice that this is often not enough and we would need to incorporate more contextual information in our methodology to be able to correctly choose whether or not to replace words in some cases.

Chapter 6

Related Work

In this chapter, we present work related to paraphrasing and readability, in particular in the context of the Japanese language. We divide the related work into three sections: we first give an overview of existing research about Japanese paraphrasing, then we present systems related to difficulty and readability assessment for Japanese and finally, we present other related work such as a paraphrasing systems in other languages.

6.1 Japanese paraphrasing

There has been a lot of previous research about paraphrasing Japanese that had goals ranging from simplification to improving sentence aesthetics.

In early 2000, Torisawa [61] presents a paraphrasing system focusing on reformulating expressions of the form "AのB", where "の" can be used to express the possession of "B" by "A" but also other types of relationships that often contain implicit information. The author proposes a method to transform expressions such as "AのB" into a relative clause, for example, "着物の女性", having an implicit meaning of "the woman wearing a kimono", into its explicit clause: "着物を切る女性". To do so, the author utilizes an expectation maximization [46] based unsupervised learning method and try to maximize the co-occurrence probability of the explicit clause given the probability of the two components "A" and "B".

Tanabe et al. [56] focus on paraphrasing Japanese modality expressions. Japanese has many modal expressions often expressing similar concepts, for example, "してくれますか" and "してください" both express a request to the interlocutor with the former being slightly reserved while the latter is a polite order. In this work, the authors express these modality expressions in the form of logical relationships and define equivalence rules in order to paraphrase them.

Masuno et al. [40] present a system focusing on paraphrasing Japanese adjectives. They provide an extensive taxonomy of the different types and conjugations of Japanese adjective and propose a rule-based method allowing to convert adjectives, supporting double-negation and other complex patterns.

Shioda et al. [54] develop a Japanese simplification system based on Twitter data. To generate a simplification dataset, they rely on the number of occurences of the n-gram in the data as well as the number of users using a given n-gram. Using this data, they generate a list of simple words and they transform the input to have as many words as possible in the output sentence that belong to the simple words list.

6.2 Japanese readability

In the context of our thesis, paraphrasing is of course important but another critical point is to be able to estimate the difficulty of a given sentence. In this section, we present research focusing on assessing the difficulty of Japanese sentences.

Hasebe et al. [19] present a readability evaluation system called jReadability that analyzes Japanese input text and emits a readability score. The score is based on several features such the length of the sentence, the ratio of "kango" and "wago" as well as the ratio of verb and particles. The weights assigned to each of these variables are computed by fitting them using existing texts classified into different difficulty levels.

Sato et al. [30] present a system called "Obi" that evaluates the difficulty of a Japanese sentence. Unlike jReadability, Obi is created in a almost fully automated way by looking at the words present in Japanese primary school textbooks. It emits a readability score by choosing the grade in which the text has the highest probability to belong to, with respect to the textbooks corpus.

6.3 Other related work

Paraphrasing has also been explored in other languages, such as English. In this section, we will briefly introduce some work about non-Japanese paraphrasing.

Barzilay et al. [8] develop a method to extract paraphrases from a parallel corpus, such as two transactions of the same text. While this is a helpful, parallel corpus are more common in a bilingual context, which is explained by Bannard et al. [7], who show how to generate paraphrases using such a corpus. Using similar methods, Ganitkevitch et al. [17] create The Paraphrase Database, a parallel corpora that contains over 220 million paraphrase pairs. However, such corpus are not easy to build for less common languages and the cost of building in such a case is very high. To improve on this issue, Kajiwara et al. [33] propose a method to automatically generate monolingual parallel corpus. Kajiwara [32] then leverages such corpus and proposes a method that leverages advances in neural machine translations but takes into account the particularities of paraphrasing to improve the quality of the paraphrases.

Chapter 7

Conclusion and Future Work

7.1 Summary

In this thesis, we explained what is plain Japanese and described why it is becoming more and more important that it becomes a common tool, with the increasing foreign population in Japan. We then described a survey that we performed with six non-native participants to assess the efficiency of plain Japanese and showed that it was in particular helpful for non-native speakers whose language do not include Kanji characters. Consequently, we presented our the methodology used to develop our system Same Language Translator that reformulates complex Japanese expressions into plain Japanese. We gave an overview of the methodology and described the different datasets and algorithms that we used. We provided an evaluation of our system on several types of sentences and showed that it managed to simplify part of the sentence while making a relatively small amount of mistakes. Finally, we presented work related to this thesis, with an emphasis on other Japanese paraphrasing systems.

7.2 Future Work

In this thesis, we have presented a methodology and a first prototype to transform complex Japanese expressions into plain Japanese. We have seen that our current system successfully transforms some words and expressions into simpler ones but also has some major limitations. One of the main limitations is that it is incapable of reasoning about the context of a word in a sentence further than comparing the n-grams frequency. This leads to cases where a work is wrongly replaced with a similar one that does not fit in the context. One of the main avenue of future research will be to improve the methodology to obtain a better understanding of the context. As a starting point, we could use a more complex language model, such as recurrent neural networks [55], to choose whether a word fits in a context or not. A next step from there could be to use parallel corpora of complex and plain Japanese and use supervised machine learning techniques to improve the results of the reformulation.

Bibliography

[1] Foreign-born population – 2018 international migration outlook 2019 oecd. https://www.oecd-ilibrary.org/sites/e025d47d-en/index.html?itemId=/content/component/e025d47d-en. Accessed on 12/19/2020.

[2] NEWS WEB EASY やさしい日本語で書いたニュース nhk. Accessed on 12/19/2020. URL: https://www3.nhk.or.jp/news/easy/.

[3] NHK NEWS WEB nhk. Accessed on 12/19/2020. URL: https://www3.nhk.or.jp/news/.

[4] 「やさしい日本語」についてBureau of Tokyo 2020 Olympic and Paralympic Games Preparation. https://www.2020games.metro.tokyo.lg.jp/multilingual/references/easyjpn.html. Accessed on 12/19/2020.

[5] 国籍・地域別　在留資格（在留目的）別　在留外国人 法務省統計局. Accessed on 12/19/2020. URL: https://www.e-stat.go.jp/stat-search/files?page=1&layout=datalist&toukei=00250012&tstat=000001018034&cycle=1&year=20190&month=24101212&tclass1=000001060399&tclass2val=0.

[6] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. Flair: An easy-to-use framework for state-of-the-art nlp. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, 2019.

[7] Colin Bannard and Chris Callison-Burch. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 597–604, 2005.

[8] Regina Barzilay and Kathleen McKeown. Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th annual meeting of the Association for Computational Linguistics*, pages 50–57, 2001.

[9] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.

[10] Francis Bond, Timothy Baldwin, Richard Fothergill, and Kiyotaka Uchimoto. Japanese semcor: A sense-tagged corpus of japanese. In *Proceed-*

ings of the 6th Global WordNet Conference (GWC 2012), pages 56–63, 2012.

[11] Eric Brill. Some advances in transformation-based part of speech tagging. *arXiv preprint cmp-lg/9406010*, 1994.

[12] Danqi Chen and Christopher D Manning. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 740–750, 2014.

[13] Iain S Duff. A survey of sparse matrix research. *Proceedings of the IEEE*, 65(4):500–535, 1977.

[14] Explosion AI. spacy · industrial-strength natural language processing in python, 2020. URL: `https://spacy.io/`.

[15] Christiane Fellbaum. Wordnet. *The encyclopedia of applied linguistics*, 2012.

[16] Takeshi Fuchi and Shinichiro Takagi. Japanese morphological analyzer using word co-occurrence-jtag. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 409–413, 1998.

[17] Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. Ppdb: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764, 2013.

[18] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. Learning word vectors for 157 languages. *arXiv preprint arXiv:1802.06893*, 2018.

[19] Yoichiro Hasebe and Jae-Ho Lee. Introducing a readability evaluation system for japanese language education. In *Proceedings of the 6th international conference on computer assisted systems for teaching & learning Japanese*, pages 19–22, 2015.

[20] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

[21] Hideki Tanaka, Hideya Mino, Shinji Oichi, Motoya Shibata and others. やさしい日本語ニュースの公開実験サイト 「NEWS WEB EASY」 の評価実験. 研究報告自然言語処理 (NL), 2012(9):1–9, 2012.

[22] 佐藤政光. 日本語学習者の語彙習得に関する調査研究-(1) 基本語彙の問題点について. 1999.

[23] 内山和也. 日本語能力試験出題基準語彙表. `http://web.ydu.edu.tw/~uchiyama/data/noryoku.html`. (Accessed on 12/19/2020).

[24] 小島健輔 and 佐藤理史 and 藤田篤. 文字 bigram モデルを用いた日本語テキストの難易度推定. 言語処理学会第 15 回年次大会 (NLP-2009), pages 897–900, 2009.

[25] 平尾拓也 and 宮田光樹 and 鈴木孝彦 and 廣川佐千男. 日本語 WordNet 類義語の誤り検出: コーパス利用の試み (人工知能と知識処理). 電子情報通信学会技術研究報告= IEICE technical report: 信学技報, 114(339):13–18, 2014.

[26] 松田陽子 and 前田理佳子 and 佐藤和之. 災害時の外国人に対する情報提供のための日本語表現とその有効性に関する試論. 日本語科学, 7:145–159, 2000.

[27] 柴田実. やさしい日本語の試み. 放送研究と調査, 56(2):36–42, 2006.

[28] 熊野正 and 田中英輝. 統計機械翻訳によるやさしい日本語書き換えの性能向上. 言語処理学会第 22 回年次大会発表論文集, pages 713–716, 2016.

[29] 西村彩香. 「表記」 から見る 「やさしい日本語」:「やさしい日本語」 使用の実態調査と有効性の検証を通して. 語文論叢, (33):62–45, 2018.

[30] 高津弘明 and 福岡維新 and 藤江真也 and 林良彦 and 小林哲則. 快適な情報享受を可能とする音声対話システム. 言語処理学会第 22 回年次大会発表論文集, pages 302–305, 2016.

[31] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

[32] Tomoyuki Kajiwara. Negative lexically constrained decoding for paraphrase generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6047–6052, Florence, Italy, July 2019. Association for Computational Linguistics. URL: `https://www.aclweb.org/anthology/P19-1607`, `doi:10.18653/v1/P19-1607`.

[33] Tomoyuki Kajiwara and Mamoru Komachi. Building a monolingual parallel corpus for text simplification using sentence similarity based on alignment between word embeddings. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1147–1158, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee. URL: `https://www.aclweb.org/anthology/C16-1109`.

[34] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014.

[35] Taku Kudo. Mecab: Yet another part-of-speech and morphological analyzer. 2006. URL: `http://mecab.sourceforge.jp`.

[36] Taku Kudo and Yuji Matsumoto. Japanese dependency analysis using cascaded chunking. In *Proceedings of the 6th Conference on Natural*

Language Learning - Volume 20, COLING-02, pages 1–7, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi:10.3115/1118853.1118869.

[37] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[38] Dekang Lin. On the structural complexity of natural language sentences. In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*, 1996.

[39] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.

[40] Shigeaki Masuno, Ryuji Urata, Satoshi Sato, and Takehito Utsuro. 語構成パターンに応じた変換規則による形容詞の言い換え. 2005.

[41] Yuji Matsumoto, Akira Kitauchi, Tatsuo Yamashita, Yoshitaka Hirano, Hiroshi Matsuda, Kazuma Takaoka, and Masayuki Asahara. Japanese morphological analysis system chasen version 2.0 manual. *NAIST Techinical Report*, 1999.

[42] Mikolov et al. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

[43] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[44] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. Advances in pre-training distributed word representations. *arXiv preprint arXiv:1712.09405*, 2017.

[45] Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Extensions of recurrent neural network language model. In *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5528–5531. IEEE, 2011.

[46] Todd K Moon. The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6):47–60, 1996.

[47] Thomas R Niesler and Philip C Woodland. A variable-length category-based n-gram language model. In *1996 IEEE International Conference*

on Acoustics, Speech, and Signal Processing Conference Proceedings, volume 1, pages 164–167. IEEE, 1996.

[48] Yoshiki Ohira. kotodama - github. Accessed on 12/19/2020. URL: `https://github.com/tennmoku71/kotodama/blob/master/kotodama/data/kotodama_dic.csv`.

[49] Adam Pauls and Dan Klein. Faster and smaller n-gram language models. In *Proceedings of the 49th annual meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 258–267, 2011.

[50] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[51] Lawrence Rabiner and BiingHwang Juang. An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16, 1986.

[52] Helmut Schmid. Part-of-speech tagging with neural networks. *arXiv preprint cmp-lg/9410018*, 1994.

[53] Helmut Schmid. Improvements in part-of-speech tagging with an application to german. In *Natural language processing using very large corpora*, pages 13–25. Springer, 1999.

[54] Kento Shioda, Kajiwara Tomoyuki, and Mamoru Komachi. 日本語学習者の文章読解支援のための語彙制限. 2015.

[55] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. Lstm neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*, 2012.

[56] Toshifumi Tanabe, Kenji Yoshimura, and Kosho Shudo. Modality expressions in japanese and their automatic paraphrasing. In *NLPRS*, pages 507–512, 2001.

[57] Takaaki Tanaka, Yusuke Miyao, Masayuki Asahara, Sumire Uematsu, Hiroshi Kanayama, Shinsuke Mori, and Yuji Matsumoto. Universal dependencies for japanese. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1651–1658, 2016.

[58] Pasi Tapanainen and Timo Jarvinen. A non-projective dependency parser. In *Fifth Conference on Applied Natural Language Processing*, pages 64–71, 1997.

[59] The Japan Foundation. N1～N5:認定の目安 | 日本語能力試験 JLPT. (Accessed on 12/19/2020). URL: `https://www.jlpt.jp/about/levelsummary.html`.

[60] Arseny Tolmachev, Daisuke Kawahara, and Sadao Kurohashi. Juman++: A morphological analysis toolkit for scriptio continua. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 54–59, 2018.

[61] Kentaro Torisawa. A nearly unsupervised learning method for automatic paraphrasing of japanese noun phrases. In *The Sixth Natural Language Processing Pacific Rim Symposium (NLPRS2001) Post-Conference Workshop, Automatic Paraphrasing: Theories and Applications*, pages 63–72, 2001.

[62] Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 conference of the North American chapter of the association for computational linguistics on human language technology-volume 1*, pages 173–180. Association for Computational Linguistics, 2003.