Regular Paper

Continual Lengthening of Titles: Implications for Deep Learning Named Entity Recognition

Yukihisa Yonemochi[†] and Michiko Oba[‡]

[†]Graduate School of Future University Hakodate, Japan
[‡]Future University Hakodate, Japan
{g3119008, michiko}@fun.ac.jp

Abstract - The objective of this paper is to highlight a problem of deep learning (DL) named entity recognition (NER) for titles of various works. Extracting information from a text input is an important task for interactive text interfaces such as chatbots and voice interfaces. In the field of natural language processing, NER is known as a type of information retrieval. Most of the latest NER methods utilize deep learning with the highest accuracy. As the standard, in most cases, only word sequences have been used as input features. However, these methods have a problem in recognizing unknown longer compound words. Longer titles can be found in titles of works such as novels, manga, animation, and movies. We verified this phenomenon using the following three aspects: First, we verified how standard DL NER has a problem with longer titles. Second, assuming that the addition of lexical features improves the performance, we verified its effectiveness. Third, we verified that such longer titles are distributed within actual real-world titles. Herein, we report the results of the verification and suggest the necessity of considering countermeasures.

Keywords: Named entity recognition, deep learning, feature selection

1 INTRODUCTION

The extraction of proper nouns or unique names such as movie titles from input text is an important part of natural language processing (NLP) when developing interactive systems such as chatbots or voice interfaces. Recognizing the named entity of an artifact, as defined through Message Understanding Conference (MUC) [1] and Information Retrieval and Extraction Exercise (IREX)[2], is the standard method for extracting proper nouns. This is known as the NER in the NLP area. NER has two tasks: tagging and disambiguation. Tagging is the process of generating tag data for the start and end positions of a fragment in a text. Disambiguation involves choosing the correct data from several candidate types.

Table 1 shows an example text that has the fragment "The Bridge on the River Kwai," which can be the title of either a novel or a movie.

B-NOVEL and B-MOVIE indicate the beginning of the title of a novel and a movie, respectively. Likewise, I-NOVEL and I-MOVIE show a range of titles. In addition, O refers to OTHER. This is called the BIO-style label, which is typically used for NER tasks. The NER process must first obtain the correct range, and in this example, the range is 2-7 along with the index number. The NER is then disambiguated, choosing between novels and movies. In this example, B-MOVIE and I-MOVIE are the correct choices.

Although the text initially seems to be saying, "I saw the bridge," it actually states, "I saw the movie." Knowledge of the title can help us reconcile this meaning. Next, we need to guess whether it is a NOVEL or MOVIE. Humans can intuitively recognize that the title can be a movie utilizing the verb "saw."

In NER systems, statistical methods are applied to recognize the range and choose the correct type. In recent years, some DL methods have obtained high NER scores. NLP-progress [3] reported accuacy of more than 90% for several datasets.

We encountered a problem in which unknown, longer unique names often cause errors in the recognition process. In this context, "unknown" indicates a unique name that was not included in the set of training data, and "longer" means a unique name that has several more words of different types than the known name. Such unknown, longer names are commonly seen in novel, manga, cartoon, or movie titles. We refer to such names as "UnknownLonger" throughout this paper. We can assume that some UnknownLonger names will not be correctly extracted in the actual system under the following situation. DL is trained using texts including a list of existing titles. It can extract the existing titles from an input text with high accuracy. However, if a new longer title is announced, it is not correctly extracted.

A typical method for evaluating the accuracy randomly separates a dataset into subsets of 70% for training and 30% for testing. It can hide the problem from the evaluation score because most titles are included in the training data.

The objective of this study is to determine whether it is necessary to add lexical information to a feature when using DL NER in an interactive interface. In this paper, we verify the effectiveness of adding a lexical feature to DL NER for extracting UnknownLonger names from texts. The effectiveness of the technique was measured through experiments conducted using our original dataset with manga, novel, cartoon, and movie titles. For comparison purpose, Japanese and English titles are used because longer titles occur more frequently in Japanese. The dataset was generated using several spoken text patterns with real titles. The text pattern was manually created, and title names were collected from Wikidata

Table 1: Input and Label of NER

index	0	1	2	3	4	5	6	7	8	9
input	**I**	**saw**	**The**	**Bridge**	**on**	**the**	**River**	**Kwai**	**yesterday**	**.**
Label as Movie	O	O	B-MOVIE	I-MOVIE	I-MOVIE	I-MOVIE	I-MOVIE	I-MOVIE	O	O

[4].

In this paper, we verify the problem for the following three aspects.

Verification 1: Existence of the problem
The DL model is prepared, and it is confirmed whether it works well. (Test-1)
Problems are intentionally created by changing the data selection. (Test-2)

Verification 2: Effectiveness of adding lexical features
The DL job is modified to add lexical features and confirm whether the accuarcy improves. (Test-3)

Verification 3: Investigation into the distribution
The distribution of the title lengths is visualized based on the time and type. The difference in the length of the work titles are compared based on the nationality and type.

Detailes of the verification procedure are described in the following sections.

2 DEEP LEARNING NAMED ENTITY RECOGNITION

DL methods have recently been utilized for the NER. NLP-progress reports have shown that the top NER rankings for the CoNLL [5] dataset are CNN [6], RNN [7]+CRF, LSTM [8], Bi-LSTM [9], BERT [10].

The language model GPT-3 [21] by OpenAI [20] or GPT-J [24] by EleutherAI [23] are getting as much attention as BERT. They have numerous more parameteres than BERT and have reported a better performance. They can also be applied for NER too [22][25].

These DL methods are suitable for time-series data. The same set of features are used for the NER, regardless of the method applied. Simultaneously, label data indicating the tag are prepared, which are referred to as "tagged," "annotated," or "labeled." The sequence of tokens is the explanatory variable and the sequence of labels is the objective variable.

Figure 1 shows an example of how DL NER processes input text and labels. DL NER is processed through following steps.

Tokenization
First, the input text needs to be tokenized as a sequence of words or morphemes. Latin-derived languages can be tokenized using space characters, and the Japanese language can be tokenized using a Japanese tokenizer [11].

Encoding
Once the text has been tokenized, the sequence of tokens is translated into vector data, which is called a tensor. The method used to generate a distributed representation has a strong influence on the NER results. The distributed representation of tokens is used as the input feature for DL tools to improve the accuracy. The labels are also translated into tensors, but in simple through a one-hot vector.

Training
The deep learning model is trained to take a word tensor as the input and output a label tensor.

Most studies have used large tagged corpora to create a trained model. However, in [12] and [13], the authors proposed utilizing Wikidata to create large training datasets. In so doing, they support a wide range of vocabulary but still use the same features and labels.

3 DEFINING THE PROBLEM

As mentioned in section 2, standard NER methods use only the surfaces of the word and word sequences as the input feature. This can cause tagging errors in the UnknownLonger titles.

Consider the classification model for the NER, which is trained using training data. The underlined part is the title and must be extracted as an argument for certain applications.

- I want to watch **Star Wars** next week.
- When will **Harry Potter** be released?

The number of words in the titles are both 2, and all of the words are nouns. Using this example, unknown titles such as "Star Trek" can be extracted as a title even if it was not included in the training data. This is called unknown word extraction.

- I want to watch **Star Trek** next week.

However, the trained model failed to extract the title when it was not in the training data and was longer. See the following example:

- I saw **The Bridge on the River Kwai** yesterday.

input	feature tensor						label	label tensor			
I	0	1	0	1	0	0	O	1	0	0	0
saw	0	0	1	1	1	0	O	1	0	0	0
the	1	0	1	0	1	0	B_MOVIE	0	1	0	0
bridge	0	1	1	1	0	0	I_MOVIE	0	0	1	0
on	1	1	1	0	1	0	I_MOVIE	0	0	1	0
the	0	1	0	1	1	1	I_MOVIE	0	0	1	0
river	1	1	1	1	0	1	I_MOVIE	0	0	1	0
kwai	0	1	0	1	1	0	I_MOVIE	0	0	1	0
yesterday	1	0	1	1	1	0	O	1	0	0	0
.	0	0	0	0	1	1	O	1	0	0	0

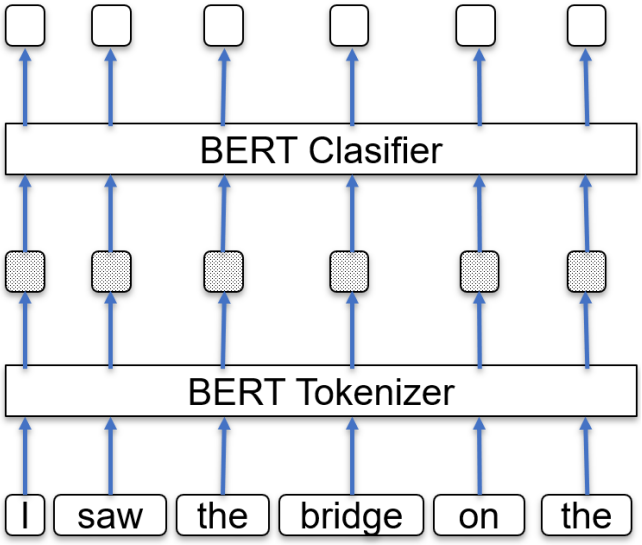


Figure 1: DL NER of a title

This title has six words, i.e., article-noun-preposition-article-noun-noun. This is more complex and longer than that of the examples mentioned above, which can confuse the classification module. This type of situation poses a problem. Artifacts such as manga (comic books), animation (cartoons), novels, and movies are being added daily. However, we cannot train classification models daily, because the computing cost is high.

We conducted a verification using datasets that were arbitrarily divided by the number of words in the target title. Shorter titles were used for the training model, and the remaining longer titles were used for testing.

The existence of this problem in NER tasks was verified through the experimental results described in Section 6.

4 ADDING A BOOLEAN FEATURE

To improve the accuracy of the DL NER of UnknownLonger titles, we propose injecting vocabulary information into the feature. This method, which have previously proposed for improving the NER for gazetteer [14] adds a feature, which is simple a Boolean flag, to the input feature. A flag indicates whether a series of words can be found in the database. In the training data, flags are generated from the labels. In the test data, flags can also be generated from the labels. In the production input data, the flags can be added by searching the word sequences in the database. **Figure 2** shows an example of the addition of flags to a feature. Column "b" indicates the binary feature flag adding vocabulary information to the input.

In this example, "The Bridge on the River Kwai" is the title of a movie. Flags can be generated from the B-MOVIE and I-MOVIE labels during training and testing. During the production time, the flag must be added by searching the name from a database. More precisely, the flags are added to the tensor matrix after the words are translated into a distributed expression.

5 EXPERIMENT

For the first and second verifications, we conducted an experiment using our newly prepared test data.

The purpose of this experiment is to verify whether a typical NER method has a problem with UnkownLonger. Adding lexical features is an effective way to improve the problem.

In desiging the experiment, we need to make sure that the typical NER model does indeed have the UnkonwLonger problem. For this purpose, we prepare a typical NER model that works well on the prepared dataset and we intentionally create the UnknownLonger problem. To intentionally recreate a problems, we will focus on how to split the test data to measure the performance. In measuring the performance of a typical NER model, the test data are divided randomly. In this case, the UnknownLonger problem is hidden. In our experiment, we recreate the UnknownLonger problem by splitting the test data only for sentences that contain long entity names as the test data. This must recreate the problem, which we will observe. Next, we will confirm the improvement by adding lexical features described in Chapter 4.

Through a series of experiments, we aimed to confirm that an NER model with high accuracy becomes less accurate on UnknownLonger, and that the lexical features improve the accuracy significantly. The steps of the experiment are as follows:

(Test-1) A typical valid deep-learning model is prepared.
(Test-2) It is confirmed whether the problem actually occurs.
(Test-3) Whether the problem can be improved by adding a lexical feature is then confirmed.

As the first verification, the result of (Test-1) becomes (Test-2) owing to the influence of UnkownLonger, and as the sec-

input	feature tensor						b	label	label tensor			
I	0	1	0	1	0	0	0	O	1	0	0	0
saw	0	0	1	1	1	0	0	O	1	0	0	0
the	1	0	1	0	1	0	1	B_MOVIE	0	1	0	0
bridge	0	1	1	1	0	0	1	I_MOVIE	0	0	1	0
on	1	1	1	0	1	0	1	I_MOVIE	0	0	1	0
the	0	1	0	1	1	1	1	I_MOVIE	0	0	1	0
river	1	1	1	1	0	1	1	I_MOVIE	0	0	1	0
kwai	0	1	0	1	1	0	1	I_MOVIE	0	0	1	0
yesterday	1	0	1	1	1	0	0	O	1	0	0	0
.	0	0	0	0	1	1	0	O	1	0	0	0

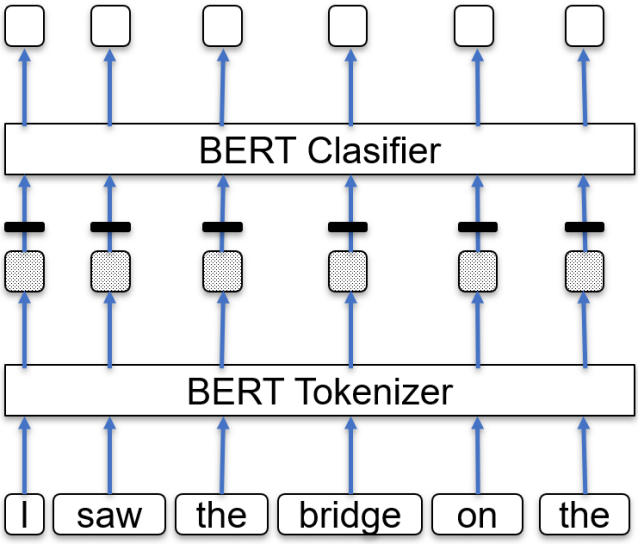


Figure 2: Adding lexical feature for long title

ond verification, (Test-2) extends to (Test-3) owing to the effect of the lexical feature. A more detailed description of the experiment environment is provided in this section.

5.1 Test Dataset

Because we have a problem with long titles, the dataset was specially designed. The dataset was generated by combining the following two types of text:

- spoken statement patterns
- titles

We chose the titles of mangas, novels, cartoons, and movies as the target areas. The spoken statements were manually created with 20 statements for each area. List 1 shows examples of statements in the movie context.

List 1. Statement patterns

```
I can't wait until x is released
Will you go see x next week?
I need to buy a ticket for x
The movie x will be coming out in theaters
x world premiere
```

Titles were collected from each area of the Wikidata. The work titles on Wikidata cover a wide range of genres, from the very old to the very new. All the necessary information for aggregation is available, such as language, year of release, and nationality. It is updated on a daily basis, making it extremely worthwhile to connect to the actual interactive interface. Of course, if there is a highly comprehensive database of titles of wrok, we can use it for our experiments. However, Wikidata is free, easy to connect, and convenient for scientific experiments. For these reasons, we used Wikidata to collect data for our experiments.

When collecting titles from Wikidata, the instance_of(P31) property is used. For example, when we collect movie titles, the movie item on Wikipedia is wd:Q11424. Wikidata can be searched using a SPARQL[15] query. Code 1 shows the SPARQL query used to collect movie items for US movie titles, with ID and label names.

Code 1. SPARQL query for US movie titles

```
SELECT ?item ?itemLabel
WHERE {
  ?item wdt:P31 wd:Q11424;
  wdt:P495 wd:Q30.
  SERVICE wikibase:label
   {bd:serviceParam wikibase:language "en". }
}
LIMIT 500
```

In the query, P495 denotes the "country of origin," Q30 represents the "USA." The SERVICE argument limits the items to those that have a label for locale "en." Therefore, the query collects English labels of instances of movies with the property, i.e., country of origin, being the USA. For the Japanese work labels from Japan, Q17 is used instead of Q30.

The query times out if a LIMIT is not included. The query limits the number of lines to 500. List 2 shows examples of English movie titles from the United States.

List 2. Examples of movie titles

```
The Brain That Wouldn't Die
Puppet Master: The Legacy
```

Puppet Master 4
The Lusty Men
Curse of the Puppet Master

Table 2 shows the list of items used in the SPARQL query to collect titles from Wikidata.

Table 2: List of id and target labels of Wikidata

Type	instance_of(P31)	Japanese	USA
Manga	wd:Q21198342	MANGA	
TV Animation	wd:Q63952888	ANIME	
Written work	wd:Q47461344		NOVEL
Animated series	wd:Q581714		ANIME
Movie	wd:Q111424	MOVIE	MOVIE

Different sets of items were used in both English and Japanese. Movies, animations, and manga are in Japanese. Because mangas (comic books) were originally created in Japan, it is obvious that TV animation shows and movies will be created from them. However, we tried to collect an instance_of "comic book series" for English, and only 172 items were collected, which is an insuffcient number. Therefore, we chose "written work" (Q47461344) and more than 6,000 instance items were collected. This indicates that they were classified as novels.

By using patterns and titles as examples, and combining the first pattern and first title, the first line of the dataset is as follows:

"I can't wait until *The Brain That Wouldn't Die is released.*"

We used 500 titles and 20 statements for each of the three areas, with a total of 30,000 lines of statements for English and Japanese works.

5.2 Tools

The experiment environment used the existing components. We chose the BERT Tokenizer which is a state-of-the-art technology for NER. **Table 3** lists the environments and tools used during the experiment.

We utilized Python3.0 and TensorFlow [16] 2.0 on Google Colaboratory [17]. The input texts are tokenized using space characters for the English titles, and using Japanese Tokenizer Janome for Japanese titles. The distributed representation is applied as the tensor input feature, and the BertTokenizer with a pre-trained model bert-base-uncased is used for the English titles. In addition, BertJapaneseTokenizer with a pretrained model bert-base-japanese-whole-word-masking[19], which is published by Tohoku University, is used for Japanese titles. TFBertForTokenClassification was used for classification.

5.3 Experimental Steps

The experiments were conducted through the following steps:

1. Dataset preparation
   - Prepare statement patterns
   - Collect titles of written work (USA), including manga (Japanese), animation, and movie from Wikidata
   - Merge, combine patterns and titles, and generate statements and labels
2. Prepare standard NER job
   - TensorFlow
   - BERT Tokenizer (English / Japanese), BERT Classification
3. Test the standard method (Test-1)
   - Randomly divide the dataset into a 7:3 ratio
   - Confirm whether the NER model works well (high score)
4. Verify UnknownLonger problem (Test-2)
   - Divide dataset into shorter and longer titles
   - Verify that the problem exists (lower score)
   - This result is the baseline
5. Verify the effectiveness of the lexical feature (Test-3)
   - Add lexical Boolean feature to the input
   - Compare with the baseline (improve the score)

Table 3: Environment and tools applied

Computing environment	Google Colaboratory
Platform	Python3.0
Tokenized by	space character (for English) Janome (for Japanese)
DL Tool	TensorFlow 2.0
Distributed Representation	BertTokenizer BertJapaneseTokenizer
Classifier	TFBertForTokenClassification
Pretrained model	bert-base-uncased(for English) cl-tohoku/bert-base-japanese-whole-word-masking(for Japanese)
batch size	32

5.4 Dividing database based on the number of words

Figure 3 shows how to intentionally create a problem.

In Test-1, similar to the typical method of testing the accuracy of deep learning models, the dataset is randomly devided into a ratio of 70% to 30%. In Test-2, the set of titles is divided into training and testing data using the number of words in each title. Designating the number of words in the title by N, the following were used as the training data for the experiments.

- For English text: $N < 3$.
- For Japanese text: $N < 4$.

Table 4: Experiment results(test-1,2,3)

language	division	count of data training	testing	lexical feature	Epoc	F1 score NOVEL	ANIME	MOVIE	weighted avg.
English	random 7:3	21264	9114		20	0.997	1.000	1.000	0.999
	train w/N <3	21198	9180		15	0.7987	0.7593	0.8111	0.7889
	train w/N <3	21198	9180	added	24	**1.0000**	**0.9998**	**0.9959**	**0.9988**
language	division	training	testing	feature	Epoc	MANGA	ANIME	MOVIE	weighted avg.
Japanese	random 7:3	21264	9114		20	0.997	0.998	0.986	0.994
	train w/N <4	21771	8607		6	0.5837	0.6212	0.3045	0.4605
	train w/N <4	21771	8607	added	47	**0.9932**	**0.9820**	**0.9734**	**0.9803**

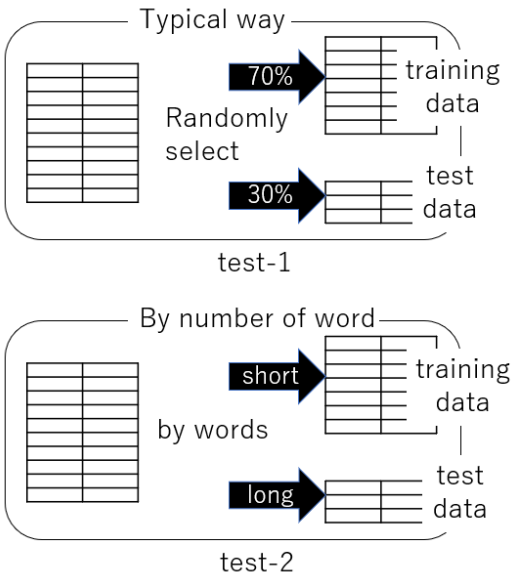


Figure 3: Intentionally Re-create the Problem

We chose these numbers to split the dataset into a ratio of approximately 7:3. This indicates that English titles are shorter than Japanese titles.

6 RESULTS AND DISCUSSION

Table 4 presents the experimental results. The experiment results, listed in **Table 4**, are described in this section. The lines for each language in the results of Tests 1, 2, and 3 are discussed in Section 5.3.

6.1 Preparation of Classification Job

The first lines for each language in **Table 4** show the results of Test-1, where the training job using the standard method randomly divided the dataset into 70% for training and 30% for testing. The weighted average F1 score was 99.9% for English and 99.4% for Japanese. These results are excellent, and we thus assume that the dataset was prepared arbitrarily and is too easy for the classification task. However, it shows that the prepared job works for this dataset.

6.2 Verifying the Existence of the Problem

The second line for each language shows the results of Test-2, the training job using the traditional method of dividing the dataset using the number of words in the title as the criterion. The test data lines with a title having less than three words for English and four words for Japanese were used as the training data. The remaining data were used as the test data. The weighted average F1 score was 78.9% for English and 46.1% for Japanese. From these results, we can verify that the accuracy of the classification decreases when testing only UnkownLonger titles. This difference indicates that Japanese texts are more strongly affected by UnknownLonger titles than English texts, which was considered the baseline.

6.3 Verifying the Effectiveness of the Lexical Features

The third line for each language in **Table 4** shows the results of Test-3, i.e., the training job with an added lexical feature that divides the dataset according to the criteria of the number of words in the title. The data division criteria were the same as those for Test-2. The weighted average F1 score was 99.9% for English and 98.0% for Japanese. This indicates that adding lexical features is effective for this job and the dataset.

6.4 Discussion and the Mechanism

Figure 4 shows a visualization of the results of Tests-1,2, and 3 for Verifications 1 and 2.

When testing UnknownLonger, we can observe a clear decrease in accuracy (Verification1,) and we can observe that it improves by adding lexical features (Verification2.)

From the experiment, we confirmed that the problem exists and we can improve it by adding a lexical feature. Now discuss the mechanism. A typical NER has only a sequence of words as input. The names labeled during training are stored in the trained model as a lexicon, but only peripheral words are used for an unknown word estimation. UnknownLonger names often appear like excerpts from sentences, and they may include words that are included in the periphery. In such cases, the boundary between the name and the peripheral words is extremely vague. The lexical features as a Boolean vector that strongly suggest the possibility of a boundary between names and peripheral words, and the discriminator re-

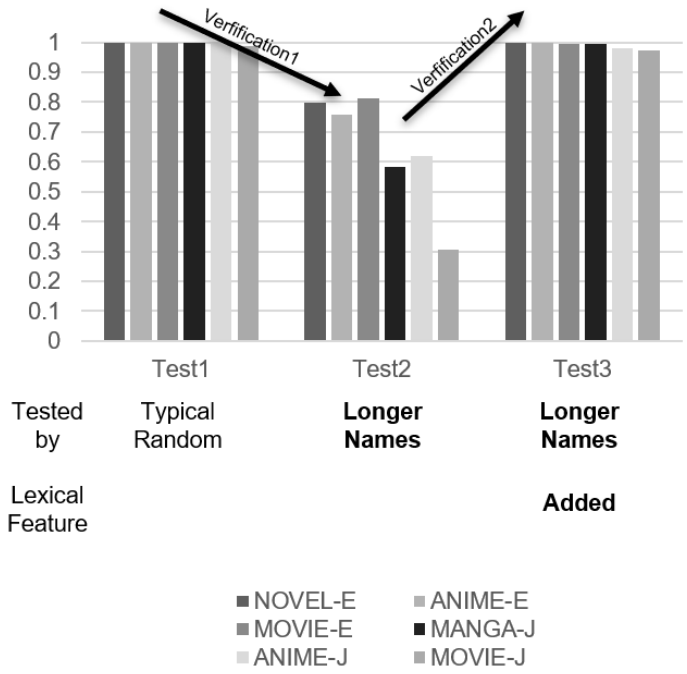


Figure 4: Verification 1 and 2

sponds strongly to them, which is considered to be a significant contribution to solving the problem.

7 VERIFYING THE DISTRIBUTION OF LONGER TITLES

As a third verification, we investigated the distribution of the length of the titles. We assume that not only the number of words, but also the types of words that make up the name of an entity, affect the recognition. The types of words are known as parts of speech. Parts of speech include nouns, verbs, adjectives, adverbs, conjunctions, and articles. Words that fall into different parts of speech, such as nouns, verbs, adjectives, and adverbs, should be placed far apart in a distributed representation space by the BERT Tokenezer. UnknownLonger names are often composed of many parts of speech, and the type of part of speech is a matter of interest.

A POS analysis was studied earlier. In the experiments conducted in this study, we use OpenNLP for English names and a Japanese morphological analyzer for Japanese names. The words and POS were counted for each title. In English titles, words are separated by space characters, and the POS is tagged using Apache OpenNLP [18]. In Japanese titles, words are separated using the Japanese Tokenizer Janome, and the POS is tagged at the same time.

Table 5 shows an example of the POS of the title of the work, as analyzed by Apache OpenNLP.

Table 5: Example POS

The	Bridge	on	The	River	Kwai
DT	NNP	IN	DT	NNP	NNP

In this example, DT is a determiner, NNP is a singlar proper noun, and IN is a preposition or subordinating conjunction.

There are six words. In addition, there are three types of POS, i.e., articles, nouns, and prepositions.

The number of words and the POS are summarized by year to confirm the trend of these lengths. The number is also summarized by these work types to confirm which type has a long name.

7.1 Increasing length

Figure 5 shows a summary of the published year. From the visualization, the trend in the length of the work titles increases yearly. The trend is the same for the main title and the subtitles in both Japan and the US. Thus, we verified that the length of work titles increased annually. This means that UnknownLonger titles may appear after the recognition model is trained.

7.2 Longer Titles for Japanese Manga and Movie

Figure 6 shows the distribution of the number of words and POS based on the type of work. **Table 6** shows the independent t-test results for each work type. The t-test calculations confirmed that Japanese manga and movie titles are significantly longer than English titles. This confirms that the accuracy of the results for the Japanese manga and movie titles in **Table 4** was strongly affected.

8 CONCLUSION AND FUTURE WORK

We verified that the accuracy of the BERT classifier is affected by the length of unknown words and can be recovered by adding a lexcal feature. As indicated by the existing titles of various works, Japanese manga and movie titles are longer than English titles. The lengths of such titles are increasing every year. It was therefore suggested that measures such as adding lexical features are needed to improve the accuracy of identifying UnknownLonger titles. Particular attention should be paid to the Japanese titles of manga and movies.

We assume that GPT-3 and GPT-J have the same problems and rather exacerbate the problem in terms of training costs. We will also need to make sure that GPT-3 and GPT-J have the same problem and that the proposed method is effective.

As an assumption, adding lexical features has an unintended effect on short names. If the title of the database is only a single common word, it may be labeled a title whenever the word is used in the input text. We need to study how to work around this problem to utilize lexical features.

In a text interface, people do not accurately or perfectly input long names. Abbreviations or shortened names are used. In the future, we will study the tendency of people to abbreviate or shorten long titles, and devise a method for recognizing shortened names from input text using lexical information.

9 ACKNOWLEDGEMENT

We thank Google LLC for providing the Google Colaboratory used in our experiment. We would like to thank Editage

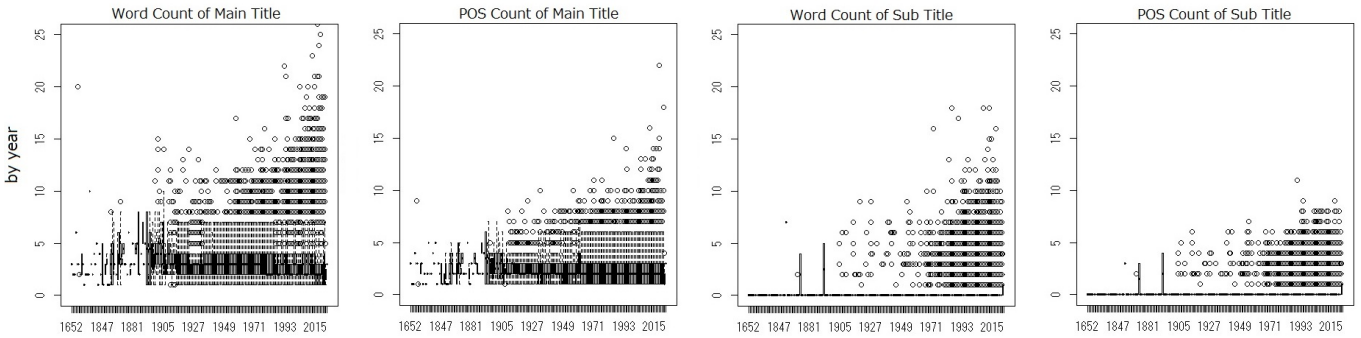


Figure 5: Count of Words and POS of titles by year

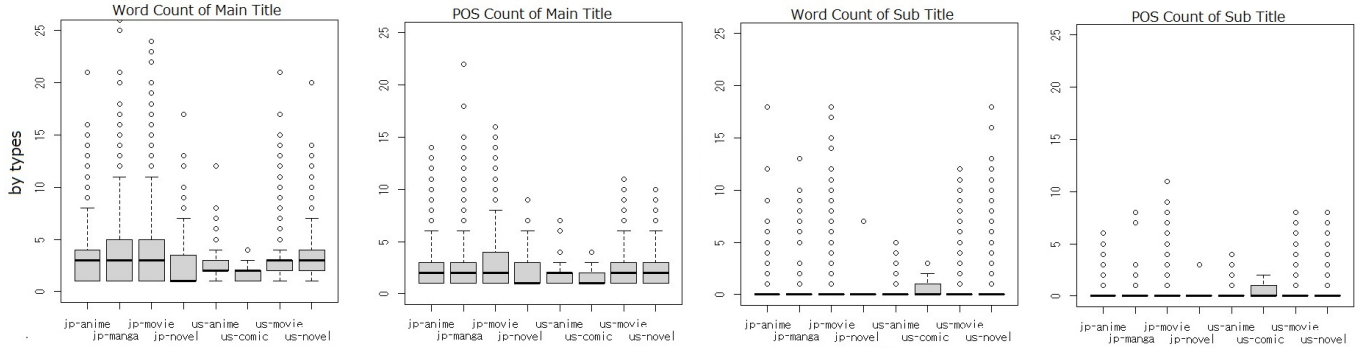


Figure 6: Count of Words and POS of titles by types

(www.editage.com) for English language editing.

REFERENCES

[1] N. Chinchor and E. Marsh. MUC-7 information extraction task definition, In Proceeding of the Seventh Message Understanding Conference (MUC-7), Appendices, pp. 359-367 (1998).

[2] S. Sekine and H. Isahara. Irex: Irie evaluation project in Japanese. In Proceedings of the 13th International Conference on Language Resources and Evaluation (LREC), pp. 1977–1980 (2000).

[3] S. Ruder. Nlp-progress. London (UK): Sebastian Ruder (accessed 2020-01-18). https://nlpprogress.com (2020).

[4] F. Erxleben, M. Günther, Markus Krötzsch, Julian Mendez, and Denny Vrande˘ci´c. Introducing wikidata to the linked data web. In Proceedings of International Semantic Web Conference (ISWC), pp. 50–65. (2014).

[5] ACL SIGNLL. The signll conference on computational natural language learning. https://conll.org/ (2020).

[6] Y. LeCun et al. LeNet-5, Convolutional Neural Networks. URL: http://yann.lecun.com/exdb/lenet, Vol. 20, No. 5, p. 14 (2015).

[7] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences, Vol. 79, No. 8, pp. 2554–2558 (1982).

[8] S. Hochreiter and J̈ Schmidhuber. Long shortterm memory. Neural Computation, Vol. 9, No. 8, pp. 1735–1780 (1997).

[9] M. Schuster and K. K Paliwal. Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, Vol. 45, No. 11, pp. 2673–2681 (1997).

[10] J. Devlin, M.-W. Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018).

[11] Janome. https://mocobeta.github.io/janome/ (2020).

[12] A. L. F. Shanaz and R. G. Ragel. Named entity extraction of wikidata items. In 2019 14th Conference on Industrial and Information Systems (ICIIS), pp. 40–45. IEEE, (2019).

[13] S. Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 708–716, (2007).

[14] S. Magnolini, V. Piccioni, V. Balaraman, M. Guerini, and B. Magnini. How to use gazetteers for entity recognition with neural models. In Proceedings of the 5th Workshop on Semantic Deep Learning (SemDeep-5), pp. 40–49 (2019).

[15] w3c. Sparql query language for rdf. (2008).

[16] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: A system for large-scale machine learn-

Table 6: t-test Movie Titles between Japan and US(P=0.05)

	ANIME		MANGA		MOVIE		NOVEL	
	ja-JP	en-US	ja-JP	en-US	ja-JP	en-US	ja-JP	en-US
Mean	3.05	2.74	**3.49**	2.04	**4.00**	2.78	2.83	3.17
Variance	5.89	2.57	10.45	1.08	10.76	1.82	7.59	2.71
Count	1225	139	1295	26	6978	65371	192	7153
t	2.00		6.56		31.03		-1.72	
P($T \leqq t$)	0.023		6.24E-08		7E-199		0.043	

ing. In 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), pp. 265–283 (2016).

[17] E. Bisong. Google colaboratory. In Building Machine Learning and Deep Learning Models on Google Cloud Platform, pp. 59–64, (2019).

[18] Apache, https://opennlp.apache.org/ (2021).

[19] Tohoku University, cl-tohoku models, https://huggingface.co/cl-tohoku (2021)

[20] OpenAI, https://openai.com/ (2021)

[21] Floridi, L., Chiriatti, M., GPT-3: Its Nature, Scope, Limits, and Consequences., Minds & Machines 30, 681–694 (2020), https://doi.org/10.1007/s11023-020-09548-1(2021)

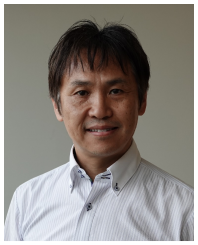
[22] R. Ma, Using GPT-3 for Named Entity Recognition, https://ricky-ma.medium.com/using-gpt-3-for-named-entity-recognition-83a95389408e (2021)

[23] EleutherAI, https://www.eleuther.ai/ (2021)

[24] EleutherAI, Mesh Transformer JAX, https://github.com/kingoflolz/mesh-transformer-jax/ (2021)

[25] A. Diaz, NLP – What happens in Entity Extraction and its value, https://www.marscrowd.com/blog/text/nlp-entity-extraction-and-its-value/ (2021)

(Received: November 3, 2021)
(Accepted: February 10, 2022)



Yukihisa Yonemochi Graduated from Gunma National College of Technology, the Department of Mechanical Engineering in 1987. In the same year, he joined IBM in the area of software product support, selling, marketing, and research. From 2012, he was a lecturer at Future University Hakodate. In 2015, he joined Honda Research Institute Japan as a manager. In 2020, he founded Pandrbox. He is a member of the Information Processing Association of Japan (IPSJ), Japanese Society for Artificial Intelligence (JSAI), and The Association for Natural Language Processing (ANLP).

Michiko Oba Michiko Oba received a PhD. in engineering from Osaka University, Japan, in 2001. She worked in the Systems Development Laboratory and the Software Division of Hitachi Ltd. from 1982. She is currently a professor in the Department of Media Architecture, Future University Hakodate, Japan. Prof. Oba is a member of IEEE Computer Society, the Information Processing Society of Japan (IPSJ), and the Institute of Electrical Engineers of Japan (IEEJ). She became a Council Member of Science Council of Japan in 2020. She became a IPSJ fellow in 2020.