Extraction of Novel Character Information from Synopses of Fantasy Novels in Japanese using Sequence Labeling

Yuji Oka
Kagawa University
Takamatsu, Kagawa, JAPAN
s20g460@stu.kagawa-u.ac.jp

Kazuaki Ando
Kagawa University
Takamatsu, Kagawa, JAPAN
ando.kazuaki@kagawa-u.ac.jp

Abstract

In this paper, we propose a method of extracting novel character information such as characters' names, gender, age, occupations, and a part of relationships between characters from synopses of fantasy novels in Japanese using sequence labeling by comparing sequence labeling models based on CRF (Conditional Random Fields) and deep learning with CRF. From the experimental results, we confirmed that the BiLSTM-CRF model with the information of part-of-speech of words has achieved the best performance, the precision of 85.40%, the recall of 91.47%, and F1-measure of 88.30% for extracting characters' names. The BiLSTM-CRF model has achieved the best overall performance for extracting all tags.

1 Introduction

In recent years, the spread of electronic books has created an environment in which we can read novels easily. Moreover, websites or applications to post and publish new user-generated novels have also received a lot of attention all over the world. On the other hand, as the total number of published and posted novels increases, it will be difficult to find novels that suit a reader's taste. We can search for novels by authors' names, genres of novels, and keywords set independently by the author, and choose novels from the popularity ranking. However, we cannot search for the contents of novels based on personal preferences.

In order to solve such problems, the following methods can be useful: (1) Search function using personal preferences for novels, (2) Synopsis generation and presentation based on personal preferences, (3) Generation and presentation of a correlation diagram of the characters in the novel.

Personal preferences for novels can be divided into preferences for a story such as "surprise ending" and "happy ending", and preferences for a novel character such as "handsome butler" and "dark hero". As the first step to achieve three methods, we focus on the preferences for the novel characters. In this paper, we consider a method to extract information about novel characters. By extracting and organizing the information about the characters from the novel text, it can be used not only for search targets, but also for generating synopses and correlation diagrams.

Although some of text data such as novel posting sites are available free of charge, most text data of novels published for commercial use must be used for a fee. In addition, when extracting information about characters from text of novels by supervised machine learning, it is very costly to construct training data by annotating the whole text of novels. Therefore, we use "synopses" of novels as text for training data in this paper. A synopsis is attached to almost all novels regardless of whether it is published for commercial use, the length of text is short, and the cost of constructing learning data is low. In addition, we confirmed that many synopses of fantasy novels include information about main characters in our preliminary investigation.

In this paper, we propose a method of extracting novel character information such as characters' names, gender, age, occupations, and a part of relationships between characters from synopses of fantasy novels in Japanese using sequence labeling by comparing sequence labeling models based on CRF (Conditional Random Fields) and deep learning with CRF. Since the proposed method is based on a sentence-by-sentence basis, it can be also applied to the text of novels.

2 Related Work

There are several studies on extracting information about characters from novel text and constructing diagrams or tables of relationships between them. Baba et al. (Baba, 2007) have proposed a method for extracting information about characters from novel text in Japanese by matching with a dictionary and rules. Characters' names are extracted by rules based on part-of-speech information, and attributes of the character are extracted by matching with a dictionary and rules for extraction. A diagram of relationships between characters is constructed based on whether the character exists in a specific scene and the frequency of co-occurrence of characters. From the experimental results, they obtained that the precision of extracting characters' names from novel text was 42.4%, and the recall was about 67.0%.

Yoneda et al. (Yoneda, 2012) have proposed a method for extracting character's name from novel text in Japanese using the local frequency of subjects and information of predicates that co-occur with the subject in a sentence. Based on the hypothesis that a character appears as a subject in a novel, the candidates of the character's name are extracted using particles of Japanese from sentences. Characters' names are identified based on the relationships between each candidate of subjects and predicates that co-occur with the subject in sentences. They showed that the precision of extracting characters' names was 60.3%, the recall was 91.9%, and the F-measure was 71.5%.

Next, we describe three studies to generate diagrams or tables using the extracted character information. Zhang et al. (Zhang, 2017) have proposed methods for inferring salient attributes to generate the description of main characters by extracting attributes from the source story by ranking candidates or classifying using a list of attributes abstractively. They showed that the abstractive model works better than the extractive model, and both model outperform a SVM-based baseline.

Vani et al. (K, 2019) have proposed a method for producing visual summaries by machine learning. Characters and their aliases are detected by standard natural language processing tools for clustering algorithms and named entity recognition. To generate relationship diagrams the most relevant ones and their relations are evaluated based on simple statistical analysis. The color of characters' nodes and undirected edges between characters are determined by a special sentiment analysis method based on sentence embedding.

Iyyer et al. (Iyyer, 2016) have proposed a method for generating a trajectories of temporal changes in the relationship between two characters by unsupervised neural network. The model jointly learns a set of relationship descriptors as well as a trajectory over these descriptors for each relationship in the raw novel text dataset.

Other studies include speaker identification (Iosif, 2014; Ek, 2018) and personality prediction of characters (Flekova, 2015).

3 Proposed Method

In this paper, we compare methods of extracting novel character information using sequence labeling by CRF (Conditional Random Fields) and by deep learning and CRF. We extract characters' names, gender, age, attributes, occupations, position, and organizations in this paper.

3.1 Collection of Novels' Synopses

"Synopses" of novels in Japanese are used as text for training data. To collect synopses of novels, we use Webcat Plus [1], which is an information service provided by the National Institute of Informatics (NII). Webcat Plus provides various information related to paper books and electronic books. We collect synopses of fantasy novels in BOOK database on Webcat Plus.

Finally, we randomly collected 1,008 synopses of fantasy novels written by novelists extracted from "List of Japanese Fantasy Novelists" in Japanese Wikipedia. The synopses collected consists of two or more sentences.

[1] http://webcatplus.nii.ac.jp/

3.2 Construction of Training Data

Training data used for sequence labeling is constructed by the following steps.

1. Tokenize each sentence into words by a Japanese morphological analysis tool.
2. Tag each word based on the following rules. We use IOB2 labeling scheme.

- Tag a sequence related to a character's name with "NAME"
  Example: Nishio, Nobunaga, Charlemagne
- Tag a sequence related to a gender with "MF"
  Example: man, he, beautiful woman, she
- Tag a sequence related to age with "AGE"
  Example: 16 years old, boy, old man, young, high school student
- Tag a sequence related to appearance and characteristics with "STATE"
  Example: white hair, fineness, domineering, genius, craftsmanship
- Tag a sequence related to occupations and position with "PRO"
  Example: hermit, supreme authority, member, king
- Tag a sequence related to organizations and races with "AFF"
  Example: art team, Japanese government, subjugation army, elf
- Tag a sequence related to other information about characters with "OTHER"
  Example: alien, god, demon, penguin
- Tag a sequence related to place names and architectural structures with "PLACE"
  Example: Mu, Japan, Paris, chapel, wizard school
- Tag a sequence related to relationships between characters with "REL"
  Example: brother, parent, enemy, partner, marriage
- Tag the others with O

Although place names and architectural structures are not information about novel characters, we extract them at the same time, because they are named entities useful for considering the stage of the novel (real world, parallel world, different world, etc.). In addition, we try to extract a part of relationships between characters in order to use it for the label of a correlation diagram of the characters. We will extract the rest of the relationships using information such as dependency relationships, conversation sentences in the future work.

3.3 CRF Model

A Conditional Random Fields (CRF) based model is one of the conventional sequence labeling models. We use the CRF model based on word by word labeling to extract character information from synopses of novels. CRFsuite [2] is used to implement the CRF model. The default values are used for hyperparameters of the CRF model. The window size of the CRF model is set to two.

The features used for the CRF model are as follows: (1) Notation of a word, (2) Character types (Seven types), (3) Part-of-speech, (4) Character 1-gram, (5) Character 2-gram, (6) Tag flag : If any characters in the attention word are included in each list of the ten most frequently used Chinese characters within each tag in the training data, tag flags are set based on the types of tags.

The CRF model based on the notation, character types, and part-of-speech is used for the baseline model (general CRF model). The proposed features, namely, character 1-gram, character 2-gram and tag flags, are combined to the baseline in order to verify their effectiveness.

3.4 Deep Learning Model

Recently, a sequence labeling model that combines CRF with deep learning has been proposed. In this paper, in order to extract character information, we use four models, BiLSTM-CRF proposed by Huang et al. (Huang, 2015), BiLSTM-CNN-CRF proposed by Ma et al. (Ma, 2016), BiLSTM-CRF proposed by Lample et al. (Lample, 2016), and Char-BiLSTM-CRF proposed by Misawa et al. (Misawa, 2017). Then we compare the best performance model of them with the CRF model.

Figure 1 shows BiLSTM-CRF (Huang model (Huang, 2015)). The Huang model gives word vectors, which are obtained by inputting word embedding of each word in a sentence into Bidirectional LSTM, as inputs of CRF.

Figure 2, and Figure 3 show BiLSTM-CNN-CRF

[2] http://www.chokkan.org/software/crfsuite/

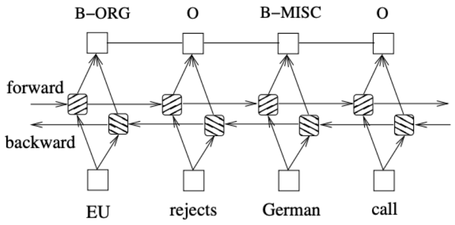

Figure 1: Main Architecture of Huang Model (cited from (Huang, 2015))

(Ma model (Ma, 2016)) and BiLSTM-CRF (Lample model (Lample, 2016)), respectively. Two models improved the performance of the Huang model by using character information included in each attention word. The Ma model inputs word embeddings and character representations computed by CNN into BiLSTM-CRF. The Lample model uses Char-BiLSTM instead of CNN in the Ma model.

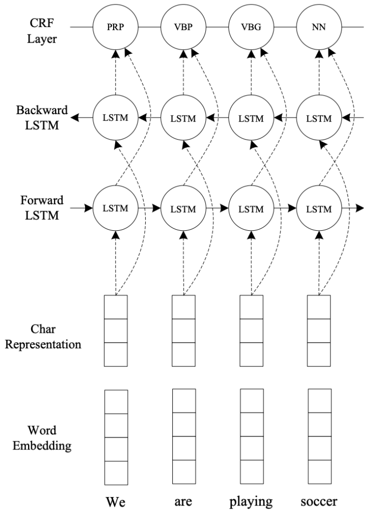

Figure 2: Main Architecture of Ma Model (cited from (Ma, 2016))

Figure 4 shows Char-BiLSTM-CRF (Misawa model (Misawa, 2017)). The Misawa model performs labeling on a character-by-character basis. Labeling on a character-by-character basis has the advantage of not being affected by errors of morphological analysis. The Misawa model inputs each character embedding and word embedding of the attention word containing the characters into BiLSTM-CRF. The Misawa model has achieved the state-of-the-art performance in Japanese NER.

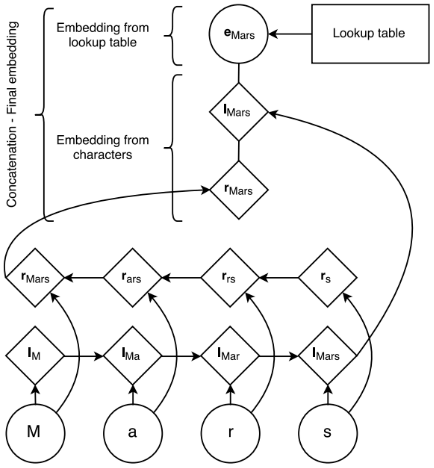

Figure 3: Main Architecture of Lample Model (cited from (Lample, 2016))

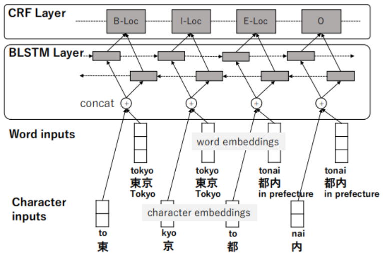

Figure 4: Main Architecture of Misawa Model (cited from (Misawa, 2017))

The parameters used in the deep learning models are shown in Table1. Dropout was adapted before and after input to BiLSTM. The parameters used in the Ma model and Lample model are shown at the bottom of Table1. The dimension of character embeddings used in deep learning models except for the Huang model is set to 50, and the dimension of the hidden layers of CNN of the Ma model and Char-BiLSTM of the Lample model is also set to 50. Each character embedding was initialized within the range of $[-\sqrt{\frac{3}{dim}}, \sqrt{\frac{3}{dim}}]$ with reference to the research (Ma, 2016). Here, dim is the number of dimensions of the character embeddings, and is set to 50 in this experiment. Since the number of characters per word in Japanese is smaller than that in English, the window size of CNN used in the Ma model is set to two. As for the word vectors, we use Wikipedia Entity Vectors [3] which were pre-trained

[3] https://github.com/singletongue/WikiEntVec

in the full text of Japanese Wikipedia. The parameters used in the pre-training are shown in the table 2. The values of word and character embeddings are updated as the model learns.

In addition to word and character embeddings, we also consider the use of part-of-speech vectors as the input of deep learning models. Aguilar et al. state that inputting part-of-speech (pos) information to the model improved the extraction performance for the WNUT2017 dataset constructed from texts in social media (Aguilar, 2017). In this paper, we also consider whether the extraction performance changes by inputting the part-of-speech information into the deep learning models. In order to use the part-of-speech information in the deep learning models, we prepare a part-of-speech vector that is initialized randomly by part-of-speech. Each part-of-speech vector is initialized in the same way as the character embeddings, and the number of dimensions of the vectors is set to zero , five and ten, respectively in the experiment in this paper. The part-of-speech vectors and word embeddings or character embeddings are input to BiLSTM at the same time, and each value of the vectors is updated as the model learns.

Table 1: Hyperparameters of Deep Learning Models

Common parameters	
Demension of BiLSTM hidden layers	128
Number of BiLSTM layers	1
Maximum number of epochs	50
Batch size	32
Learning rate	0.001
Dropout rate	0.5
Gradient clipping	5.0
Optimizer	Adam
Early stopping patience	20
Parameters of Ma model	
Number of CNN filters	50
Window size of CNN	2
Parameters of Lample model	
Dimension of character BiLSTM hidden layer	50
Number of character BiLSTM layers	1

4 Evaluation Experiments

4.1 Evaluation Methods

We evaluate the extraction performance by comparing the results of the combinations of features for the CRF model with the results of four deep learning models. The extraction performance is evaluated by 10-fold cross-validation with precision, recall, and F-measure (F1) as the evaluation criteria. The entities tagged by hand and the results tagged by the machine learning models are compared, and the correctness is judged only when they match perfectly.

Table 2: Hyperparameters of Pre-trained Distributed Representation of Words

Model	CBOW
Number of dimensions	200
Window size	5
Negative sampling	5
Down sampling	0.001

4.2 Dataset

As mentioned in Subsection 3.1, we use 3,679 sentences in 1,008 synopses collected by Webcat Plus for a dataset. In order to construct a dataset, each sentence was segmented by Japanese morphological analyzer MeCab (Kudo, 2004), and character information was tagged in each sentence manually by one person. We are going to tag each sentence by multiple person to ensure accuracy of the dataset in the future work.

The total number of characters in the dataset is 127,486. The average, minimum and maximum number of words in a sentence are 20.60, 2 and 83, respectively. The average, minimum and maximum number of characters in a sentence are 34.65, 3 and 125, respectively. The average, minimum and maximum number of characters in a word are 1.68, 1 and 22, respectively. Figure3 show the information of each tag in the data set.

Table 3: Information about Word and Character in Each Tag

Tag Types	tag	word			character		
	Num	Max	Min	Ave	Max	Min	Ave
NAME	2703	8	1	1.3	18	1	3.5
MF	363	2	1	1.0	3	1	1.5
AGE	436	4	1	1.2	6	1	2.4
STATE	518	8	1	1.8	13	1	3.5
PRO	1358	8	1	1.7	13	1	3.1
AFF	543	8	1	2.0	25	1	4.3
OTHER	825	9	1	1.5	14	1	3.0
REL	1405	8	1	1.6	17	1	3.7
PLACE	722	5	1	1.2	9	1	2.3

4.3 Results

We focus on the results of extracting "NAME" and all tags in this paper, and discuss the extraction performance of each model.

First, we focus on extraction performance of each model for "NAME". Table 4 shows the baseline (general CRF model), the CRF model with combination of the proposed features, and the four deep learning models. The upper part of Table 4 shows the results related to the CRF models. "u", "b", and "f" of the name of the CRF models represent the use of 1-gram, 2-gram, and a tag flag, respectively. "All" of the model's name represents the use of the features obtained from all words in the window size, and "One" represents the use of the features obtained from only the target word. The bottom part of Table 4 shows the results of the deep learning models. BiLSTM-CRF and BiLSTM-CRF-L represent Huang model and Lample model, respectively. "pos5" and "pos10" of the last part of the models' names represent the number of dimensions of the part-of-speech vectors. The underlined values indicate the maximum value of each item in the CRF models. The best performance in all models is highlighted by boldface type.

From the results of uOne and uAll models in the upper part of Table 4, it can be confirmed that when the feature of character 1-gram was added to the baseline, the value of recall improved by 12 points and the value of F1-measure increased by about 7 points. From the results of bOne and bAll models, the model added Character 2-gram to the baseline has a higher precision than the model with character 1-gram, however, it could not improve recall because information could not be obtained from a word consisting of a single character and the comprehensiveness of 2-gram is lower than 1-gram. From the result of fOne and fALL models, it turns out that the extraction performance of the model added tag flags to the baseline was not much improved since the use of tag flags are limited.

As for the deep learning model in the bottom part of Table 4, the extraction performance of all models was improved by the use of part-of-speech vectors. The Char-BiLSTM-CRF-pos5 model had the highest precision, and the BiLSTM-CRF-pos5 model had the highest recall and F1-measure. As for extracting "NAME", the part-of-speech information was effective to improve the extraction performance.

From all the results, we confirmed that the BiLSTM-CRF-pos5 modle has the best extraction performance for "NAME".

Next, we focus on extraction performance of each model for all tags. As for extracting all tags, the uOne-fAll model was the best performance in the CRF models, and the BiLSTM-CRF model without part-of-speech information was the best performance in the deep learning models. Therefore, Table 5 shows performance of the baseline model (general CRF model) and these models.

From Table 5, we can see that the BiLSTM-CRF model achieved the best performance, however, there is no big difference between the extraction performance for "MF". Comparing the extraction performance for all tags of the BiLSTM-CRF model and the BiLSTM-CRF-pos5 model, the extraction performance of BiLSTM-CRF model is higher than that of the BiLSTM-CRF-pos5 for "AGE", "PRO", and "STATE" tags.

Table 4: Extraction Performance of Each Model for NAME

Models	Precision	Recall	F1-measure
baseline	77.79	60.62	68.07
uOne	77.73	71.94	74.68
uAll	78.29	73.65	75.87
bOne	79.23	67.51	72.87
bAll	78.93	67.46	72.71
fOne	79.10	61.80	69.34
fAll	78.14	62.58	69.43
uAll-bAll	78.95	74.50	76.64
uAll-bAll-fOne	78.54	74.53	76.46
uAll-bOne-fOne	78.59	74.79	76.61
BiLSTM-CRF	84.24	90.99	87.48
BiLSTM-CRF-pos5	85.40	**91.47**	**88.30**
BiLSTM-CNN-CRF	85.57	88.16	86.82
BiLSTM-CNN-CRF-pos5	85.93	89.50	87.66
BiLSTM-CRF-L	85.79	88.36	87.04
BiLSTM-CRF-L-pos5	85.84	88.60	87.19
Char-BiLSTM-CRF	85.81	89.75	87.72
Char-BiLSTM-CRF-pos5	**86.12**	90.61	88.29

5 Discussion

We analyze the features of extraction errors. Figure 5 shows the percentage of each error by the baseline model, the best CRF model (uOne-fAll model), and the best deep learning model (BiLSTM-CRF

Table 5: Extraction Performance of the Three Models for Each Tag

Tag	Baseline			uOne-fAll			BiLSTM-CRF		
	Precision	Recall	F1-measure	Precision	Recall	F1-measure	Precision	Recall	F1-measure
NAME	77.79	60.62	68.07	78.23	72.96	75.47	84.24	90.99	**87.48**
MF	94.41	93.33	93.72	93.19	96.56	94.79	96.06	96.03	**95.95**
AGE	92.36	84.76	88.21	91.37	89.47	90.25	92.56	92.83	**92.62**
STATE	53.67	19.59	28.22	59.45	29.64	39.10	58.85	57.72	**57.98**
PRO	68.22	47.93	56.19	71.51	59.55	64.92	80.39	79.92	**80.14**
AFF	69.42	42.48	52.42	72.07	50.16	58.89	72.45	71.51	**71.86**
OTHER	58.78	28.53	38.19	63.72	38.63	47.86	63.62	62.58	**63.02**
PLACE	66.50	43.80	52.72	67.64	53.99	59.99	73.30	78.78	**75.89**
REL	84.35	59.53	69.71	81.80	69.48	75.10	82.99	83.93	**83.33**

model). The extraction errors are classified into the following five types.

- An error of labeling the character information as "O" (ne2oMiss)
- An error of labeling character information tags to a sequence except for character information (o2neMiss)
- A range of labeling is correct, however, a type of the character information tag is wrong (classMiss)
- A type of the character information tag is correct, however, a range of labeling is wrong (rangeMiss)
- An error of labeling both a tag type and a range (r&cMiss)

From "ne2oMiss" of the baseline and the uOne-fAll models in Figure5, it can be seen that the baseline model often tags "O" even if a sentence contains character information. On the other hand, the BiLSTM-CRF model has a low percentage of "ne2oMiss" and a high percentage of other error classes as compared to the CRF models. We consider that errors other than ne2oMiss are possible to modify using other information comparatively. From these results, we found that the CRF models and the deep learning model differ in the tendency of extraction errors, and the deep learning model has more possibility for labeling the character information correctly than the CRF models.

From the results of error analysis for each tag, we found that there is a tendency to mistake "NAME", "AFF", and "PLACE" for two different tags of them. After analyzing the details of the extraction errors, we confirmed that sequences related to three tags often contains Katakana in Japanese. Table 6 shows the percentage of the extraction errors containing Katakana by tag. From Table 6, it can be seen that about 30 to 40% in extraction errors for "NAME", "AFF", and "PLACE" contain Katakana. In Japan, a lot of fantasy novels are produced in the motif of the West, and Katakana tends to be used in characters' names and names of places in the novels. There is a possibility that the extraction performance can be improved by using the information of Katakana.

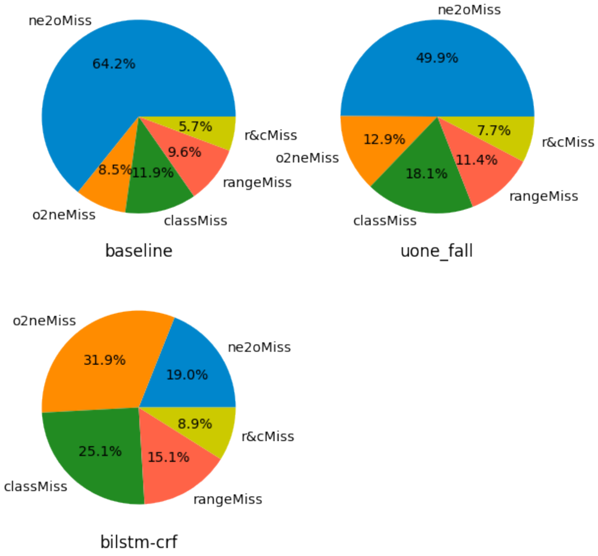

Figure 5: Extraction Errors by Three Models

6 Additional Experiment

Since synopses of fantasy novels in Japanese are targeted for extraction in this paper, names such as characters, organizations, and places tend to be written in Katakana. Focusing on names containing Katakana, We confirmed that there are names consisting of only Katakana and names consisting

Table 6: Extraction Errors Containing Katakana

Tag Types	Percentage of Katakana
NAME	**44.68**
MF	0.0
AGE	4.35
STATE	10.38
PRO	21.02
AFF	**37.10**
OTHER	20.42
PLACE	**33.33**
REL	7.47

of Katakana and other character types such as Hiragana and Kanji. Therefore, we construct a kChar-BiLSTM-CRF-pos5 model with the new character embeddings generated by compressing a sequence of Katakana to a single character and examine the performance of the model by comparing the best deep learning model for "NAME" (BiLSTM-CRF-pos5 model), the Char-BiLSTM-CRF-pos5 model and the kChar-BiLSTM-CRF-pos5 model.

Table 7 shows the results of experiments conducted with the same settings as in Section 3. The kChar-BiLSTM-CRF-pos5 model obtained all value was lower than that of the Char-BiLSTM-CRF-pos5 model. Therefore, it can be said that the extraction performance for NAME is not improved by the new character embeddings focusing on Katakana.

Comparing the extraction errors of the Char-BiLSTM-CRF-pos5 model with the kChar-BiLSTM-CRF-pos5 model, the latter can predict suitable tags for the sequences which appear more than once in the training data. However, we found a tendency that it was difficult to predict suitable tags for sequences never appeared in the training data. From the results, it is thought that the generalization ability of the model was degraded because the amount of information used for prediction was reduced by compressing a sequence of Katakana to a single character.

Table 8 shows the values of F1-measure of three models. By comparing the Char-BiLSTM-CRF-pos5 model with the kChar-BiLSTM-CRF-pos5 model, the value of F1-measure for "AFF" by the model with the new character embeddings is about 0.5 point higher than the normal model, however, the values of F1-measure for "NAME " and "PLACE" of the the model with the new character embeddings are about 2 and 0.5 point lower, respectively. The extraction performance for all tags by the kChar-BiLSTM-CRF-pos5 model is lower than that of the best deep learning model for "NAME", BiLSTM-CRF-pos5. From the results, the new character embeddings focusing Katakana did not much contribute to extraction performance. We will continue to consider the effective use of Katakana information.

Table 7: Extraction Performance for NAME by the Best Deep Learning Model with New Character Embeddings

Model	Precision	Recall	F1-measure
BiLSTM-CRF-pos5	85.40	**91.47**	**88.30**
Char-BiLSTM-CRF-pos5	**86.12**	90.61	88.29
kChar-BiLSTM-CRF-pos5	85.25	86.70	85.95

Table 8: F1-measure of Three Models on Extracting "NAME", "AFF", and "PLACE"

Model	NAME	AFF	PLACE
BiLSTM-CRF-pos5	**88.30**	**71.73**	**76.97**
Char-BiLSTM-CRF-pos5	88.29	69.73	72.98
kChar-BiLSTM-CRF-pos5	85.95	70.21	72.49

7 Conclusion

In this paper, we have compared methods for extracting novel character information using sequence labeling by CRF and by deep learning with CRF. The model with the part-of-speech vectors added to the input of the BiLSTM-CRF model has achieved the best performance for extracting "NAME", and the BiLSTM-CRF model has achieved the best overall performance for extracting all tags. Focusing on that Katakana tends to be used in "NAME", "AFF", and "PLACE" in the fantasy novels, we have considered the extraction performance of the kChar-BiLSTM-CRF-pos5 model with the character embeddings generated by compressing a sequence of Katakana to a single character, however, it could not confirm the beneficial effect.

In the future, we are going to consider the effective use of Katakana information, expand the data set, and consider a method of linking a character's name to other character information.

References

Takaaki Yoneda, Takahiro Shinozaki, Yasuo Horiuti, and Shingo Kuroiwa. 2012. Extraction of Novel Characters using Predicate Information (述語情報を利用した小説の登場人物の抽出). In *Proceedings of the Annual Meeting of the Association for Natural Language Processing*. pp.855-858. (in Japanese)

Kozue Baba and Atsushi Fujii. 2007. Extraction and Systematization of Character Information from Novel Text (小説テキストを対象とした人物情報の抽出と体系化). *In Proceedings of the Annual Meeting of the Association for Natural Language Processing*. pp.574-577. (in Japanese)

Zhiheng Huang, Wei Xu, and Kai Yu, 2015. Bidirectional LSTM-CRF Models for Sequence Tagging. *arXiv preprint arXiv: 1508.01991*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Dyer Chris. 2016. Neural Architectures for Named Entity Recognition. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*.

Shotaro Misawa, Motoki Taniguchi, Yasuhide Miura, and Tomoko Ohkuma. 2017. Character-based Bidirectional LSTM-CRF with words and characters for Japanese Named Entity Recognition. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*. pp.97-102.

Gustavo Aguilar, Adrian Pastor López Monroy, Fabio González, and Thamar Solorio. 2018. Modeling Noisiness to Recognize Named Entities Using Multitask Neural Networks on Social Media. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. volume 1. pp. 1401–1412.

Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying Conditional Random Fields to Japanese Morphological Analysis. In *Proceedings of the Empirical Methods in Natural Language Processing*. pp.230–237.

Weiwei Zhang, Jackie Chi Kit Cheung, and Joel Oren. 2019. Generating Character Descriptions for Automatic Summarization of Fiction. In *Proceedings of the ThirtyThird AAAI Conference on Artificial Intelligence, 2019*. pp. 7476–7483.

K Vani and Alessandro Antonucci. 2019. NOVEL2GRAPH: Visual Summaries of Narrative Text Enhanced by Machine Learning. In *Proceedings of Text2Story - Second international Workshop on Narrative Extraction from Text co-located with 41th European Conference on Information Retrieval*. pp.29-37.

Mohit Iyyer, Anupam Guha, Snigdha Chaturvedi, Jordan Boyd-Graber, and Hal Daumé III. 2016. Feuding Families and Former Friends: Unsupervised Learning for Dynamic Fictional Relationships. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pp.1534-1544.

Elias Iosif and Taniya Mishra. 2014. From Speaker Identification to Affective Analysis: A Multi-Step System for Analyzing Children ’ s Stories. In *Proceedings of the Third Workshop on Computational Linguistics for Literature*. pp.40-49.

Lucie Flekova and Iryna Gurevych. 2015. Personality profiling of fictional characters using sense-level links between lexical resources. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. pp.1805-1806.

Adam Ek, Mats Wirén, Robert Östling, Kristina N. Björkenstam, Gintarė Grigonytė and Sofia Gustafson Capková. 2018. Identifying Speakers and Addressees in Dialogues Extracted from Literary Fiction. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*.