Detecting Internet Slang Words with Two Layers Annotation Based on Joint Model of Character and Word Embeddings

LIU YIHONG

Master's Program in Informatics
Degree Programs in Comprehensive Human Sciences
Graduate School of Comprehensive Human Sciences
University of Tsukuba
March 2022

Detecting Internet Slang Words with Two Layers Annotation Based on Joint Model of Character and Word Embeddings

Name: LIU YIHONG

Natural language processing reaches the advances and excellent performance on a variety of tasks, while the presence of non-standard terms such as OOV (Out of Vocabulary) words can still reduce the accuracy of models and task results. It is labor-intensive and time-consuming to build and update a dictionary of new words continuously by taking the semantic information of novel words into account. To address this problem, we usually mask such kinds of new words or ignore them as unknown words. More recently, it has become popular to extract new words by cutting them into fine-grained, smaller units of speech, such as subwords or characters, in the form of spliced words.

Due to the widespread dissemination of Internet slang words, it is necessary to detect them by the word form features and a variety of semantic-changed registered words in social media texts into accounts. Therefore, we have constructed a new 10,000-sentence-corpus by selecting 100 popular Internet slangs, which has become pervasive in recent years. We divided the annotation framework into two-layers, main types and subcategories. With main types, Internet slangs are, based on two features as the words changed semantically from existing words and as newly created strings. We also defined ten subcategories such as "*Gairaigo*", "Japanese-English", "Dialect Borrowing" and so on, according to word-formation style. We proposed a joint embedding method based on character embeddings and word embeddings as a feature representation of new words to label the main types and subcategories sequentially for Internet slang words. Thanks to the strong correlation between the two-layers, we are able to apply the hierarchical shared ELMo multi-task learning method.

The experimental results show that our method shows the second best performance only to multi-task BERT in detecting the main type of Internet slang words, and performed best in detecting the subcategory in the case of shared-LM method. In addition, our model achieves an average 32.5% improvement in terms of F1-score for detecting the main types or subcategories compared to the single-task approach. We concluded that the use of two-layers annotation improves the performance of the models through the relevance, and it facilitated us to better observe and analyze the difference of detection models in details.

Main Academic Advisor: Yohei SEKI
Secondary Academic Advisor: Hai-Tao YU

Contents

1 Introduction **1**
- 1.1 Objective . . . 1
- 1.2 Motivation . . . 2
- 1.3 Contributions . . . 5
- 1.4 Organization . . . 6

2 Related Work **7**
- 2.1 Out of Vocabulary words . . . 8
- 2.2 Embedding Models . . . 8
  - 2.2.1 Word Embedding . . . 8
  - 2.2.2 Subword Embedding . . . 9
  - 2.2.3 Character and Word Embedding . . . 10
- 2.3 Sequence and Context Embedding Models . . . 10
  - 2.3.1 LSTM . . . 10
  - 2.3.2 ELMo . . . 11
  - 2.3.3 BERT . . . 12
- 2.4 Multi-task Learning Method . . . 13
  - 2.4.1 Embedding Shared Model . . . 14
  - 2.4.2 RNN Shared Model . . . 14
  - 2.4.3 Hierarchical Shared Model . . . 14
- 2.5 CRF . . . 16

3 Internet Slang Corpus **18**
- 3.1 Main Type . . . 18
  - 3.1.1 New Semantic Words . . . 18
  - 3.1.2 New Blend Words . . . 19
- 3.2 Fine-grained Subcategory . . . 19
- 3.3 Construction . . . 20

4 Our Proposed Model **25**
- 4.1 System Overview . . . 25
- 4.2 Joint Model Using Character and Word Embeddings . . . 26
- 4.3 ELMo Embedding . . . 26
- 4.4 Multi-Task Learning . . . 27

5 Experiments **29**
5.1 The Dataset and Preprocessing 29
5.1.1 Pre-training Dataset 29
5.1.2 Fine-tuning Dataset 29
5.2 Baseline System 30
5.3 Evaluation 31
5.4 Implementation 32
5.5 Overall Results 32
5.6 Discussion 33
5.6.1 Analysis of the results on main type 33
5.6.2 Analysis of the results on subcategory 35
5.6.3 Comparison of embedding models 35
5.6.4 Comparison of single task with multi-task 35
5.6.5 Comparison of shared-LM with unshared-LM 36

6 Conclusion **37**
6.1 Summary 37
6.2 Future Work 38

Acknowledgements **39**

References **40**

List of Figures

1.1 Examples of tweets with informal context 1
1.2 The distribution of subcategories inside New Semantic Word and New Blend Word 4
1.3 The work flow of Detecting Internet Slang Words 5

2.1 Structure of BiLSTM 11
2.2 Structure of Elmo for NER 12
2.3 Structure of BERT for NER 13
2.4 Embedding Shared Model 15
2.5 RNN shared Model 16
2.6 Hierarchical shared Model (+shared LM) 17
2.7 Hierarchical shared Model (+unshared LM) 17

4.1 Structure of Our Proposed Model 26
4.2 Structure of Our Proposed Model for our two tasks 28

5.1 Structure of BERT for our two tasks 31

List of Tables

1.1 Examples of Japanese Internet Slang Words. 2

3.1 Examples of New Semantic Words. 21
3.2 Examples of New Blend Words. 22
3.3 Examples of Internet Slangs for each Subcategory 23
3.4 Subcategories of New Semantic Words (SEM) and New Blend Words (BLN). 23
3.5 Examples of Annotations of Japanese Internet Slang Words. 24

5.1 Settings of each model 32
5.2 Results of Detecting "New Semantic Words" and "New Blend Words". . . . 33
5.3 F1-scores for Detecting Fine-grained Subcategories. "**" denotes cases where the difference in macro average between Our Model and other methods is statistically significant for $p < 0.01$ using two-tailed paired samples t-tests. . 34

Chapter 1

Introduction

1.1 Objective

With the rise in popularity of social media, people are becoming more inclined to contribute short and colloquial texts to social media. However, such texts are often different from formal written texts and are full of abbreviations, dialects, emoticons, punctuation or auxiliaries, and other semantic elements are missing, which like Fig. 1.1. Especially in the case of Japanese, where there are no spaces to separate words, the text cannot be handled accurately in the usual way.

Figure 1.1: Examples of tweets with informal context

At the same time, on social media, new Internet slang words are becoming the everyday language of Internet users. As they spread, even on other non-social sites, they are being used in articles, comments and other places. The popularity of such Internet slang is not consistent. Still, their meaning often determines the critical meaning of a sentence or plays a vital role in emotional understanding. It, hence, makes sense to be able to extract and understand Internet slang words in order to analyze the behavior of Internet users. In analyzing the famous Internet slang words of recent years, we were able to find many words that were different in many details based on their construction. Therefore, we have decided to adopt a new approach to vectorizing text and identifying a wide range of Internet slang words from this non-standard text. This study introduces a new resource containing

Internet slang that can be specified at two levels: 'main types' or 'fine-grained subcategories', subdividing words according to their semantics and structure.

Our goal is to detect Internet slang words further through refined classifications based on internal structure, and contextual diversity learned through pre-trained embedding models. This research, therefore, focuses on semantic differences in such social networking platforms created and widely used by Internet users. Internet slang words are labeled according the construction of terms the construction of terms using a multilingual model. With our research, we will be able to alleviate fixed vocabulary or terminology-resource constraints and expand the metadata to include informal content such as social tags [36], we propose a method for detecting unknown Internet slang words.

1.2 Motivation

In order to identify a wide range of Internet slang words precisely, we need to collect and annotate enough number of social media texts. However, the existing corpus of social media texts such as tweets and microblogs is limited in number and annotated with semantic and physical features (e.g. named entity recognition and part-of-speech tagging). In addition, the words that are not available in dictionaries but are not explicitly annotated or even replaced with "[UNK]", is not helpful for our study. Besides, no public Japanese slang-word datasets are available. Therefore, we constructed a novel Internet slang word dataset that contains a total of 100 Internet slang words for each language, together with their meanings. In addition, to classify and analyze Internet slang words with various etymologies, we defined new labels framework for annotation. The specificity and relevance of this recognition task necessitate the creation of a new corpus dedicated to the Internet slang words, annotated according to our two-layered labeling framework. Examples of the Internet slang words are shown in Table 1.1.

Table 1.1: Examples of Japanese Internet Slang Words.

Japanese New Semantic Word			Japanese New Blend Word		
Word	Common Usage	Internet Usage	Word	Internet Usage	Etymology
草	grass	interesting	禿同	strongly agree	sounded like "hagesiku doui"
丸い	round	safe	ふぁぼ	favorite	an abbreviation for "favorite"

Although many of the Internet slang words commonly used in recent years are new words that have not appeared before, there are still many that add new meanings and usages to the original words for various reasons. Such inherent words with recent semantic changes should be distinguished from their original meaning by their context and identified as Internet slang words. Therefore, we define two main types of Internet slang words: "new semantic words" (SEM) and "new blend words" (BLN).

After we had divided the two main types of Internet slang words according to their definitions, when we looked deeper into the etymology of the words, we found that there were differences and small patterns within the main types. In the new semantic words, many

of them were added to the original meaning with metaphorical derivations. Some words were used as harmonious words because their sounds were like other words. In addition, some pronunciations added new meanings to the inherent words because they were similar. Some used various characters to simulate such pronunciations and formed new words. We then investigated the construction techniques of neologisms. We found that, whether they were famous Internet slang words or other neologisms (proper nouns that accompanied new things, etc.), they followed several specific constructions. Pinter [31] constructed a dataset of innovative word types based on the New York Times called *NYTWIT*, manually annotating novel categories of words (e.g. lexical derivations, dialectal variants, blends, or compounds) according to how they were formed. A series of contextual understanding-based experiments were conducted to demonstrate that this new dataset can help improve the performance of natural language processing. We therefore defined a series of fine-grained subcategories based on word-formation in addition to the main types.

In contrast to English, that used spaces to separate words, Japanese are written using characters, or *Kanji*, without obvious word separators. When dealing with a language system that does not have word separators, the question of how to correctly segment words is a essential. For the web slang in this study, the words that have already registered in the dictionary can be segmented by the language model itself, either by pre-training with corpus or by using a dictionary lookup table. The words composed of various characters and segmented incorrectly by the model, however, will leads to poor results on a range of Japanese language processing tasks. Most of the approaches to language processing were based on word-based units, i.e. word embedding, which was indeed more effective approach for languages with delimiters, compared to the approach combining elements of words and characters. Although word embeddings are learned based on the external context of words, this method only includes limited word-level contextual information [42].

Considering the articles in Japanese, one popular approach to segment new words correctly by expanding the dictionary and use the segmented data to train the model. The other approach is to introduce characters as the basic unit of text, i.e. character level embedding, and annotate them with tags (e.g. BMES tags, BIO tags) at character position, which indicate the position of characters in word species [44]. The use of character embeddings only increases the spatial complexity of the model. By over-calculating the association among characters, it causes weak understanding the relationship between sentences and context. So some results are not necessarily better than those of word embeddings.

As word embeddings are at a disadvantage when dealing with OOV vocabulary due to the sparsity of the word distribution, some researchers have segmented the texts to the characters. They have shown that character-based models consistently outperform word-based models to process Chinese OOV based on deep learning approach. Character-based word representations are one of standard techniques of the neural architectures for natural language processing. Lamble et al. [15] illustrates that character representations can be used to process OOV words in supervised labelling tasks. Pinter et al. [32] aggregates the behavior of these units into the language from the perspective of individual hidden units within a character-level LSTM. Chen et al. [3] proposes multi-prototype character embeddings and effective word selection methods to address the problem of character ambiguity , which is a similar problem for web slang vocabularies. Language models will have a more powerful

ability to encode internal contextual information of characters within word representations. Our study chose to combine character embeddings by relating words to characters and detect Internet slang word by taking the composition of characters within words into accounts.

Our study can be regarded as a two-layered character level sequence labeling task of Internet slang words: one layer based on the determination of major types and the other on the sub-categorization of constructions. By focusing on each labeling task and fed into the model separately, the model will ignore the interconnections between labeling tasks. Therefore, we chose to use a multi-task learning method, which may result in better generalization of multiple tasks by sharing parameters between different tasks. We investigate the distributions of two-layered annotations and found that there should be some correlation between the two levels of annotation, the distribution is showing in Fig. 1.2. In this figure, Rhetoric must be new semantic words. At the same time, abbreviations, compounds, etc., must be new blend words. As a result, we suppose that the detection tasks of two-layered annotation are correlated strongly. By using the multi-task learning method, we can share the parameters of the internal modules of the model and improve the accuracy of detection tasks through the correlation between the two layered annotations. The general workflow of our research is given in Fig. 1.3.

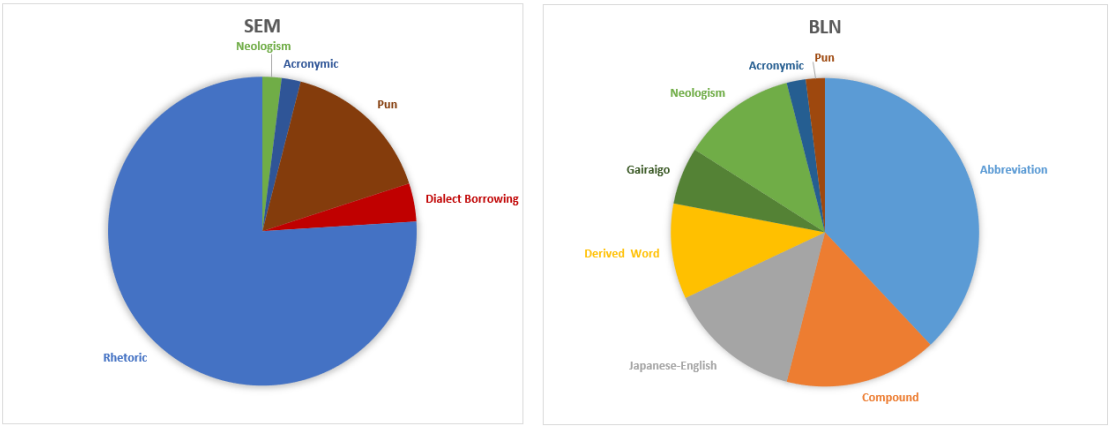

Figure 1.2: The distribution of subcategories inside New Semantic Word and New Blend Word

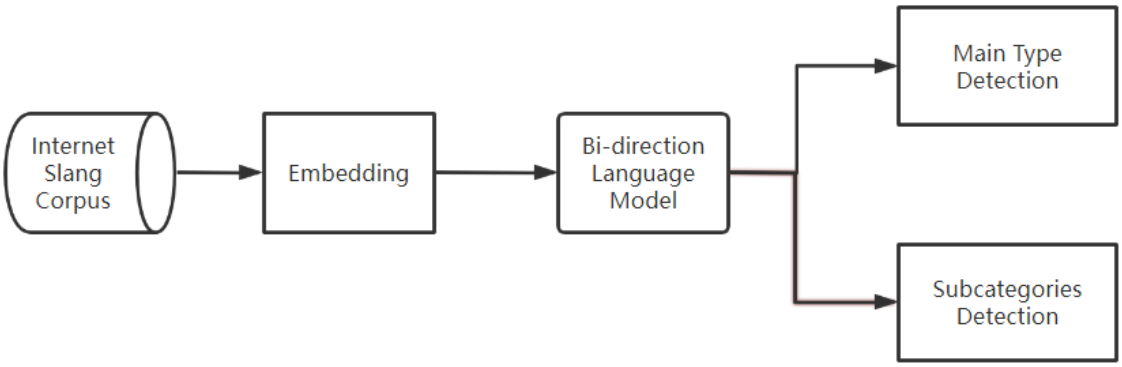

Figure 1.3: The work flow of Detecting Internet Slang Words

1.3 Contributions

The contributions of this paper can be summarized as follows.

1. We have proposed an Internet slang word corpus with a novel approach that classifies Internet slang words into two main types, "new semantic words" and "new blend words," and several subcategories defined in terms of word creation and morphological features.

2. To obtain token representations for Internet slang words effectively, we propose a novel joint embedding method that combines character embedding and word embedding by utilizing ELMo and Word2vec. To compare our method with a subword-based embedding method, we constructed a bidirectional encoder representations from transformers (BERT) model as one of the comparison models.

3. We developed a two-layer multitasking language processing model, where the upper layer draws out the subcategories and the lower layer inherits the memory of the upper layer to recognize the main type.

4. We conducted comparative experiments with multiple levels of embeddings, as character level embedding, word level embedding and subword level embedding. We also compare our joint embeddings approach using BiLSTM, ELMo and BERT models with single and multi-task learning for detecting Internet slang words in order to verify the validity of the model design.

5. Our experimental results showed that the multi-task BERT model could detect the main type most accurately, and our proposed method was better than the other remaining baseline models; while our model with shared-LM achieved the best performance in detecting the subcategory with different construction methods.

1.4 Organization

The rest of this paper is organized as follows. Chapter 2 summarizes some related works, which include the meaning of OOV words, embedding models based on language units, sequence and context embedding models and multi-task learning methods. Chapter 3 describes the components and construction process of Japanese Internet slang corpus from Twitter. Chapter 4 presents the details of the proposed joint embedding and multi-task learning method by ELMo. Chapter 5 describes some experimental settings and the results of experiments. Finally, we conclude our remarks and discuss future works in Chapter 6.

Chapter 2

Related Work

As the basis for natural language processing applications, the primary task of named entity recognition is to identify the textual scope of mentioned named entities and classify them into predefined categories such as people, locations, organizations etc. Along with the development of innovations in various fields, NER has also started to process and recognize words such as neologisms, especially in the biological, medical and business fields.

Li et al. [16] concludes that the NER task model can be divided into three main layers.

1. Distributed representations for input. Character embedding or word embedding expresses the internal information of the word, supplemented by artificial features such as lexical POS (part-of-speech), gazetteers, etc. This gives the model as much additional knowledge such as local dependencies as possible

2. The context encoder, where CNN (Convolutional Neural Network) [48] obtains more substantial local dependencies and multilayer CNN or LSTM obtain linguistic dependencies of longer texts. While Bidirectional LSTM [4] to obtain long-range dependencies along with contextual semantic information, Transformer [46] to obtain long-range dependencies, Recurrent RNN (Recurrent neural network) to obtain linguistic structure information, etc. Depending on the corresponding hypotheses, the models used are selected to give the corresponding prior knowledge on the ground.

3. The tag decoder, which predicts the tags corresponding to the input sequence, is commonly used in Point Network, Softmax, RNN, CRF, etc. The decoding layer directly uses a combination of MLP (Multilayer Perceptron) and Softmax [40] is the most primitive. No additional prior knowledge is needed. The CRF, which has the ability to supervisor the global information when the model decodes, can form constraints on the decoding so that the model knows that logically incorrect sequence labels should not appear, e.g. the position of the current token cannot be labelled as “I-Inside” if there is no “B-Begin” in front of it in the position label.

In addition, in terms of technology, it was initially popular to use machine learning methods such as SVM (Support Vector Machine) [10], HMM (Hidden Markov Model) [26], etc., but this all required much manual work to construct. Then neural network models that have emerged since then have greatly improved the flexibility and transferability of training. In particular, the combination of state of the art pre-trained models and CRFs,

such as neural network series models and BERT, allows models to perform well on various forms of NER tasks by adding some data-related features or modifications to the model or by fine-tuning the data professionally. In this study, a three-layer NER task model based on OOV words, embedding models for the distributed representations layer, sequence and context embedding models, multi-task learning models for the context encoder layer, and CRFs for the tag decoder layer and illustrated.

2.1 Out of Vocabulary words

When performing the natural language processing task, some of the low-frequency words cannot be included in the word list due to the size limitation of the word list; these words are known as OOV (Out of Vocabulary). OOV is difficult to circumvent for two main reasons completely. Name entities often contain important information, but many of them are also low-frequency words and often cannot be included in the word list. The other is that New words are emerging all the time, and old word lists cannot be updated in a timely manner. Especially in the current situation where the model is getting more prominent, it is costly to retrain the model after adding new words. The three main approaches to cope with OOV (1) Expanded word list: After expanding the word list, some low-frequency words can be incorporated into the word list, but these low-frequency words are often trained with poor word vectors due to the lack of a sufficient number of the corpus, so there is an inevitable bottleneck in expanding the word list in terms of improving the effectiveness of the model. (2) Pointing the Unknown Words based on Attention Mechanism composition (3) Character level processing, Japanese as well as Chinese language tasks mostly use character embeddings, while English tasks use n-grams.

2.2 Embedding Models

2.2.1 Word Embedding

Word-embedding methods, which learn embedding rules according to the external context of words, have been used in many natural-language processing (NLP) tasks. Word embedding is essentially the representation of individual words as real-valued vectors in a predefined vector space. Each word is mapped to a vector, where words with the same meaning have similar representations., and the vector values are learned in a similar way to neural networks, so the technique is often classified as a deep learning domain. The key to the method is the idea of using a densely distributed representation for each word. Each word is represented of tens or hundreds of dimensions. This is in contrast to the thousands or millions of dimensions required for sparse word representations.

Word2vec [25] is a statistical method for learning a standalone character embedding or word embedding from a text corpus. There are two specific methods, Skip-gram and Continuous Bag of Words (CBOW). In CBOW, the context of each word is taken as input and it tries to predict the word that corresponds to the context. Thus, we have seen how contextual words can be used to generate word representations. Skip-gram, on the other hand, uses the target word to predict the context, and in the process we generate represen-

tations. In fact, they contain some limited word-level contextual information [42], due to handling out-of-vocabulary (OOV) words. This is caused by the sparseness of word distributions for such words. To solve this issue, the Chinese texts were segmented into character units, because such character-based models consistently outperforming word-based models for Chinese representations [17]. Lample et al. [15] demonstrated that character-based representations can be used to handle OOV words in a supervised tagging task.

2.2.2 Subword Embedding

The subword-based tokenization algorithms do not split common words into smaller subwords. Instead, rare words are split into smaller meaningful subwords. Subword level embedding, with a granularity between word embedding and character embedding, is now an important performance enhancement method for NLP models. Compared to word embedding, subword tokenization helps models to learn the relationships between affixes and thus better handle unknown or rare words. Some popular subword-based tokenization algorithms are WordPiece [38], BPE [39], ULM [12], and SentencePiece [13].

The WordPiece algorithm [38] is to initialize the vocabulary with individual characters from the language, then the combination of symbols in the vocabulary with the highest likelihood is selected and added to the vocabulary iteratively to train the language model on the new word and repeat the steps until the desired vocabulary size or likelihood threshold is reached. It is implemented in two ways - bottom-up and top-down, with the bottom-up approach based on BPE. Byte Pair Encoding (BPE) [39], is a simple form of data compression in which the most common pair of consecutive bytes data is replaced with bytes that do not exist in that data, and a replacement table is needed to reconstruct the original data when it is used later. This is done by preparing a sufficiently large training corpus, determining the desired size of the subword list, splitting the words into sequences of characters and adding the suffix "</ w>" at the end, counting the frequency of occurrence of each contiguous byte pair. Unigram Language Model (ULM) [12] is another subword separation algorithm, similar to WordPiece It uses a language model to build a subword word list. As a result, the same input can be represented by different encodings, affecting the accuracy of the learned representation. SentencePiece [13] is an unsupervised text tokenizer and detokenizer, mainly used in neural network-based text generation systems, where the size of the word list is determined before entering the neural model for training. SentencePiece allows for a direct training extension to the original sentence. Besides it enables us to build a purely end-to-end system that does not depend on language-specific pre-processing or post-processing.

The subword-based tokenization can effectively balance the size of the vocabulary and the number of steps (the number of tokens required to encode a sentence), and it can have more than one way to encode instances of a particular word. Nevertheless, none of the token combinations are sufficiently comprehensive in terms of priority. Different tokens can represent the same input, which affects the accuracy of the learned representation. When new words with low usage or common misspellings appeared frequently, probabilities should not be estimated precisely for subword judgements.

2.2.3 Character and Word Embedding

Combining the relationship between context and the internal constituents of words, training character embeddings and word embeddings together is a good way to ensure minimal loss of linguistic information. There is no single way to combine the two fine-grained embedding models. Some word embeddings were calculated directly from the embedding of a sequence of characters in the word. Others chose to represent character embeddings based on the frequency of token occurrences for low frequency words, and word embeddings for higher frequency words, but this required consistency in the dimensionality of both embeddings, and lacks comprehensiveness. In addition, selecting character embeddings for OOV words and word embeddings for other words, but such methods require a high rate of update of the word list. This kind of methods using character embeddings partially, is difficult to identify semantic changes of words according to the context. On the other hand, there are ways to integrate character embeddings and word embeddings entirety. Kim et al. [11] trained a recurrent neural network on the words whose embeddings were constructed by convolution on character embeddings; Wieting et al. [45] train the embeddings of character n-grams and then add them to word embeddings. In all these cases, the models for composing the embeddings of subword units into word embeddings were learned by optimizing the targets of a large unlabeled corpus. Pinter et al. proposed a MIMICK Word Embeddings [30] which trained character embeddings into RNN model to predict word vectors for OOV words and also aggregated language-related characteristics [32] from the perspective of individual hidden units within a character-level long short-term memory (LSTM). By the way, Chen et al. [3] proposed multiple-prototype character embeddings and an effective word selection method to address the issues of character ambiguity and noncompositional words, which are also problematic for Internet slang words. Using both character and word representation, language models should have a more powerful capability to encode internal contextual information.

2.3 Sequence and Context Embedding Models

2.3.1 LSTM

LSTM (Long Short-Term Memory) is a temporal recurrent neural network obtained by improving the implicit layer of RNN, which is suitable for processing and predicting important events with relatively long intervals and delays in time sequences. The bidirectional LSTM (BiLSTM) network can pass backwards from the contextual information of the next one and solve the long distance one-band problem [23].The general structure of BiLSTM for labeling task is given in Fig. 2.1. Ma et al. [22] combined character-level information extracted using CNN with word-level representations and fed them into a BiLSTM to model the contextual information of each word, and then decoded the labels of the whole sentence jointly by sequential CRF. Limsopatham et al. [18] challenged the task to recognize the named entities in Twitter messages, which were short, noisy and colloquial. They investigated a method to handle this problem by enabling bidirectional long and short-term memory (LSTM) to automatically learn orthogonal features without feature engineering. This system achieved the most effective performance on both the 'segmentation and classification' and 'segmentation

only' subtasks compared to the other systems involved in the shared task.

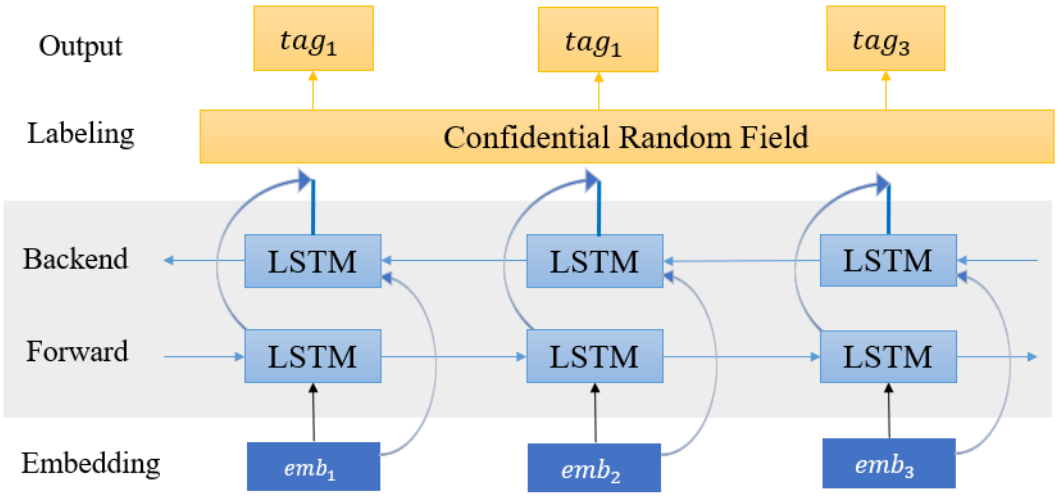

Figure 2.1: Structure of BiLSTM

2.3.2 ELMo

ELMo (Embeddings from Language Models) [28] creates contextualized representations of each token by concatenating the internal states of a two-layer BiLSTM trained on a bidirectional language modeling task. In contrast to neural networks such as LSTM, which can only generate a fixed vector for each token, ELMo's main approach is to train a complete language model first and then use the language model to process the text for the target task to generate the corresponding word vectors. As shown in Fig. 2.2, in order to obtain the ELMo representation, it is necessary to obtain the embedding layer to extract the static vectors that have been pre-trained, and then accept the hidden semantic dependencies from the first and the second layers in BiLSTM to calculate them synthetically. The model of the embedding layer used by ELMo can be more than just a single embedding model like word embedding. It also contains character-based information, which allows the model to form representations of OOV words [43]. With this type of character-based dynamic vector, ELMo considers both character information and the context between words, enabling it to recognize semantic differences most effectively. Bojkovsky´ and Pikuliak [2] claimed that ELMo representation is more flexible and less domain-restricted than traditional word embedding and can improve the accuracy of the model for processing small datasets. Chowdhury et al. [5] constructed a new dataset of disaster-related tweets using the multi-task learning method of ELMo and LSTM to capture informal writing in social media and achieved superiority.

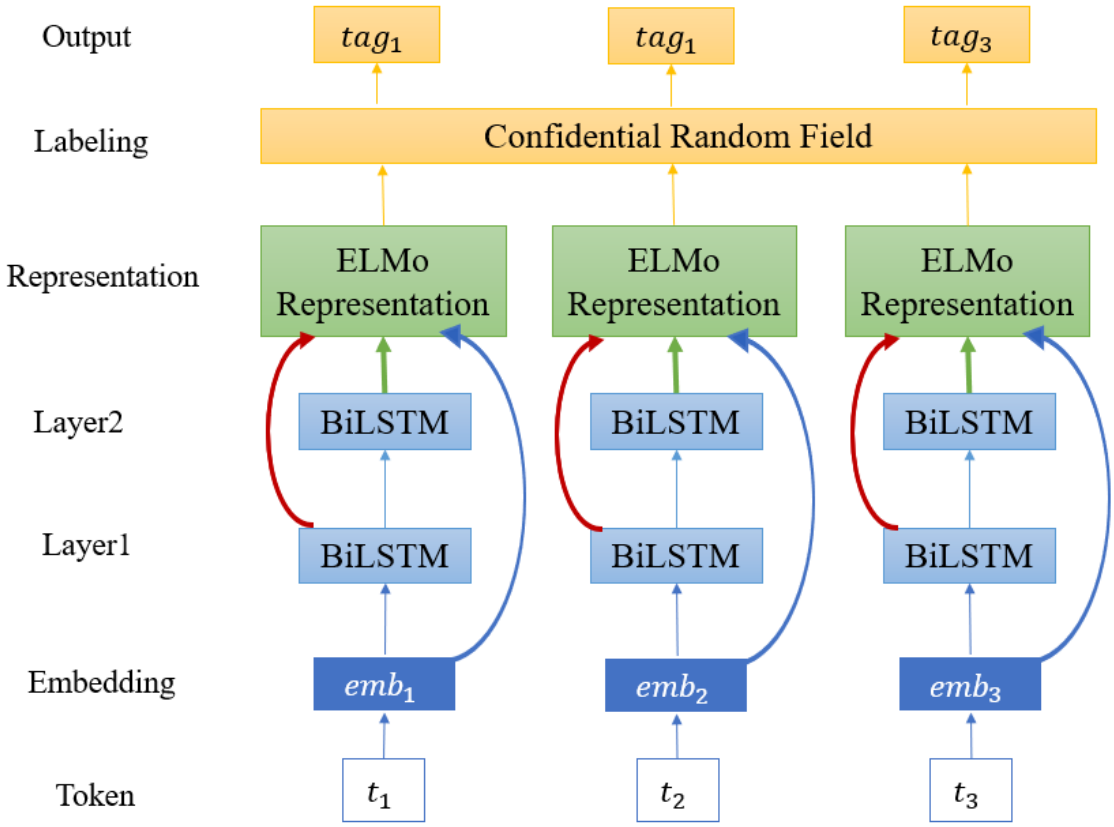

Figure 2.2: Structure of Elmo for NER

2.3.3 BERT

BERT (Bidirectional Encoder Representations from Transformers) [6] is a bidirectional model that is based on a transformer architecture. It replaces the sequential nature of recurring neural networks with a much faster attention-based approach. As it is bidirectional, given the powerful capabilities of the encoder, BERT is quite effective in certain NLP cases [6]. It employs a variety of subword tokenization methods, with byte-pair encoding [39] being the most popular approach to segmenting text into subword units. Guan et al. [8] demonstrated a substantial improvement of BERT with BiLSTM and CRF over BERT with MLP for the NER task. Sun and Yang [41] evaluated a task to identify chemical and protein entities in biomedical Spanish texts and found that fine-tuning BERT trained on English biomedical corpora was still effective because there were a large number of chemical and protein mentions sharing the same name in English and Spanish in biomedical literature. A target domain corpus migrated from a source large-scale dataset finetuning, to identify chemical and protein entities in biomedical texts. Liu et al. [19] combined Wikipedia anchors and DBpedia ontology to build a relatively high quality large-scale NER dataset to pre-train NERBERT, and showed that the model was able to tag low-resource entities relatively well in the Twitter domain. Although the character-based model demonstrated stronger auto-correction performance once a word was judged to be a type error [21], subword-based BERT

were usually used to solve the OOV problem. In this research, we selected subwords as the token representation for comparisons. The general structure for named entity recognition of BERT is given in Fig. 2.3.

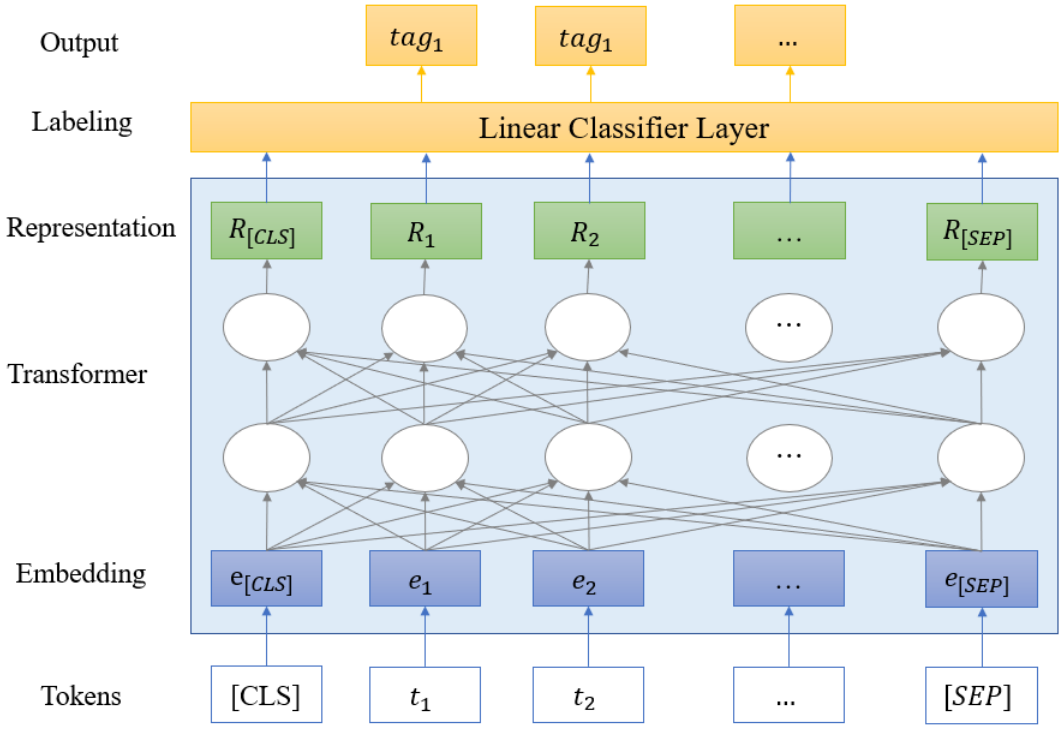

Figure 2.3: Structure of BERT for NER

2.4 Multi-task Learning Method

Multi-task learning task contains the hard sharing and soft sharing [35]. Hard sharing shares the hidden layers among tasks. With more tasks trained simultaneously, the more our model should find a representation that captures all the tasks. Regarding soft sharing, the model and parameters of each task are independent, which regularization encourage similar model parameters between tasks. When it has a high correlation between the tasks, hard sharing should be preferred.

Considering the multi-task learning method with neural networks, one of the tasks will inevitably become a feature of another task, there are challenges in practice. In particular, a uniform loss function needs to be defined for multiple tasks, e.g. when the model converges, some tasks perform better than others, while others perform disastrously. This is because uniform loss functions has different scales for the different tasks, which affects the performance of each tasks. The solution to this problem is to replace the multi-task loss function "simple summation'' with a "weighted summation''. Weighting makes the scale of each loss function consistent but also introduces a new problem: the weighting is challenging

to determine.

Setting up auxiliary sequence labeling in addition to the original sequence labeling can help with multitasking. Meanwhile, learning with auxiliary labels requires an additional set of the dataset, which may be limited for some cases. Thus, several models have been proposed for training sequence labelling tasks with other unsupervised learning tasks. Bjerva [1] verified that using auxiliary tagging in nine languages and over a range of three data scales was effective. Rei and Yannakoudakis [34] showed that the approach of using auxiliary labels on the same dataset allowed regularization of the model with different tasks while still keeping the training data in the domain. In particular, Rei et al. [33] simultaneously trained single-task sequence labelling models with a neural language model. Finally, Pham et al. [29] incorporated a word-level neural language model into both single and multi-task sequence labelling models to improve the performances. In this research, we follow their taxonomy for the multi-task learning model with sequence labeling using auxiliary labels and discuss as follows. It can assist to detect Internet slangs from the non-standard texts regardless of the size of the dataset.

2.4.1 Embedding Shared Model

To run multiple tasks at the same time, the simplest method is using a same-level-shared model, that both primary and auxiliary tasks are trained and predicted at the same-level layer. Fig. 2.4 depicts the embedding shared model, which uses the same embedding layer for both primary and auxiliary tasks, and separate LSTM and CRF layers for each. For an input token sequence $x = (x_1, x_2, \ldots, x_T)$, $h^{auxiliary}$ and h^{main} are computed as follows:

$$\mathbf{h}^{\text{auxiliary}} = \mathbf{h}^{\text{main}} = \mathbf{BiLSTM}(\mathbf{x}) \tag{2.1}$$

2.4.2 RNN Shared Model

The other way is the recurrent neural network(RNN) shared model, which uses the same embedding and LSTM layers for both primary and auxiliary tasks, and separates CRF layer for each task. In the RNN shared model in Fig. 2.5,$h^{auxiliary}$ and h^{main} are the same and are computed from one BiLSTM layer:

$$\mathbf{h}^{\text{auxiliary}} = \mathbf{BiLSTM}^{\text{auxiliary}}(\mathbf{x}) \tag{2.2}$$

$$\mathbf{h}^{\text{main}} = \mathbf{BiLSTM}^{\text{main}}(\mathbf{x}) \tag{2.3}$$

2.4.3 Hierarchical Shared Model

In hierarchical-shared models, we train and predict different supervised tasks at different levels. Gong et al. [7] constructed a Hierarchical LSTM+CRF framework consisted of characters, subwords and context-aware predictors from segmentors to capture different levels of linguistic knowledge in Chinese and achieve significant improvements over other benchmark methods. Luo et al. [20] proposed a sentence representation with enhanced learning through unshared BiLSTM with an attention mechanism for label embedding with the key-value

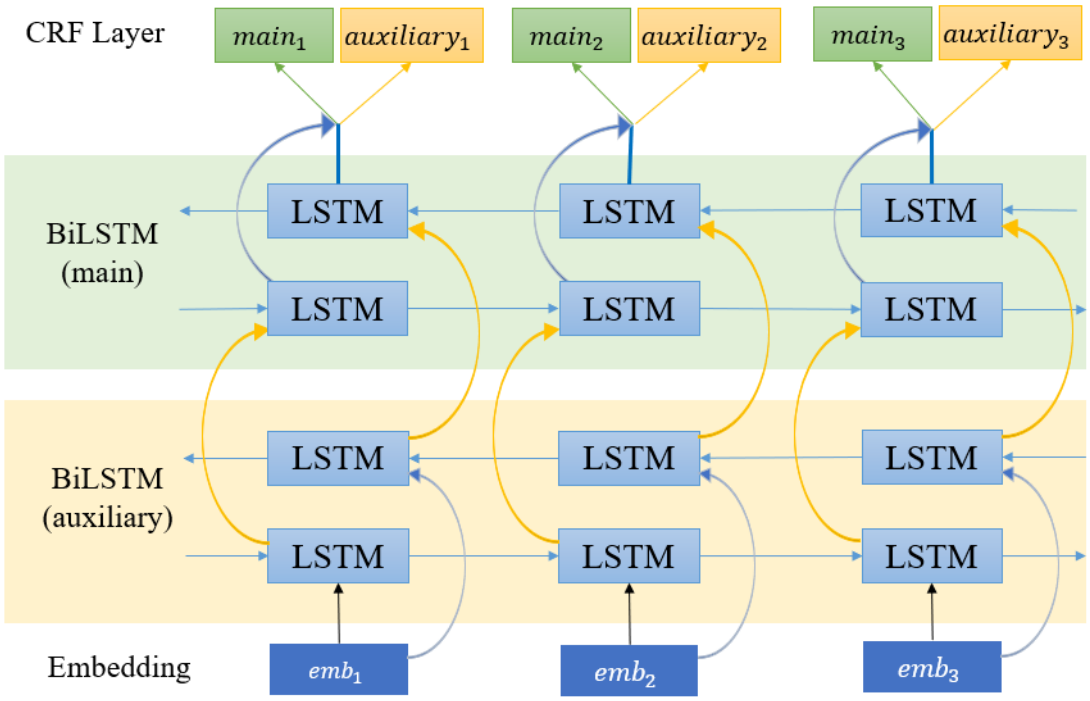

Figure 2.4: Embedding Shared Model

memory component to memorize document-level multi-layered contextualized representation for each unique word, which is sensitive to the similarity of contextual information. Hu et al. [9] proposed a hierarchical lexicon-based embedding architecture using and auxiliary words to capture key information in Chinese NER texts. The model helped to capture useful information by sharing the parameters of both main and auxiliary word categories.

According to Fig. 2.6, by analyzing the low-level and high-level layers, we can predict auxiliary tasks as well as the main task. Word representations are fed into both low-level and high-level layers to avoid catastrophic interfering with each other. This model puts the hidden state of the BiLSTM at each time step into the Softmax layer to predict adjacent words in the context. Each of the forward and backward LSTM uses two separate language models. The current objective function combines sequence label and language model objective function. Regarding hierarchical-shared models, $h^{auxiliary}$ and h^{main} are computed as follows:

$$\mathbf{h}^{\text{auxiliary}} = \mathbf{BiLSTM}^{\text{auxiliary}}(\mathbf{x}) \tag{2.4}$$

$$\mathbf{h}^{\text{main}} = \mathbf{BiLSTM}^{\text{main}}\left(\left[\mathbf{x}; \mathbf{h}^{\text{auxiliary}}\right]\right) \tag{2.5}$$

In Fig. 2.7, hierarchical-shared model(unshared-LM) which uses separate neural language model for each task is the other way to detect two-level tags. Unlike shared-LM unshared-LM uses separate language models for each task and does not share the hidden

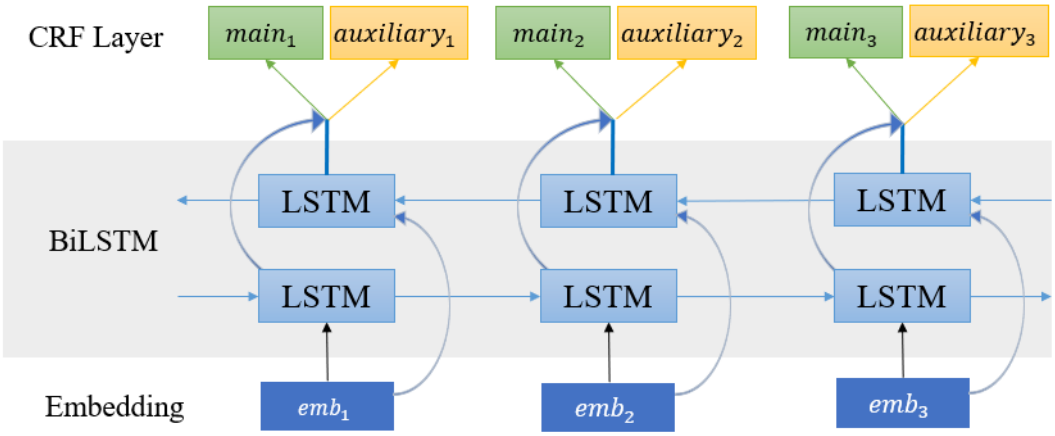

Figure 2.5: RNN shared Model

layer between the two BiLSTM layers, in addition the different parameter settings of each language model.

2.5 CRF

Conditional Random Fields (CRF) is a class of statistical modelling methods commonly used in pattern recognition and machine learning and for structured prediction. CRF belongs to the distinguished undirected probability graph model. Discrete classifiers predict the labels of individual samples without taking the influence of neighboring samples into account. CRF can refer to context; for example, linear-chain CRF (its popular in natural language processing) predicts the sequence of labels of a sequence of input samples.

CRF is widely used in sequence labeling tasks such as named entity recognition, etc. Recently, CRFs were used as a post-processing tool (with CNN or LSTM was used for segmentation). Indeed, parametrization of CRFs using LSTM allows us to be combined in an end-to-end manner. The CRFs are used to train a well-scored and labelled corpus for feature extraction, which prompts the estimation of the model parameters and ultimately generates the label reasoning model we want.

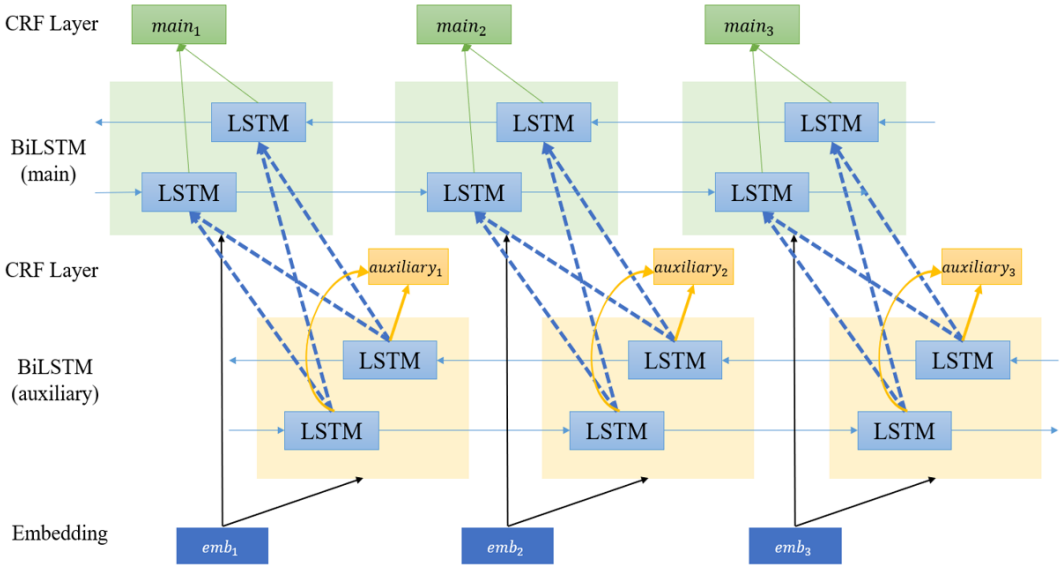

Figure 2.6: Hierarchical shared Model (+shared LM)

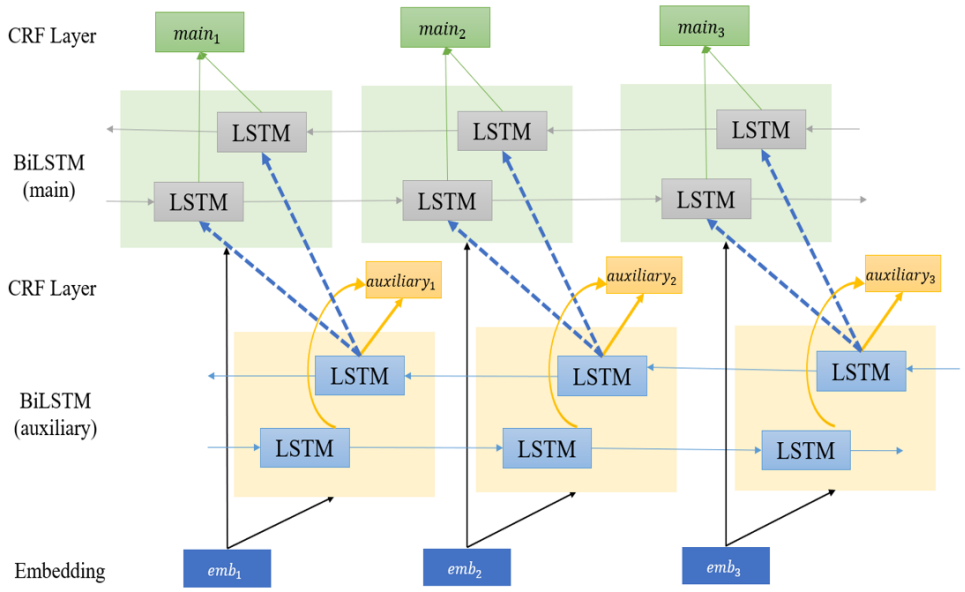

Figure 2.7: Hierarchical shared Model (+unshared LM)

Chapter 3

Internet Slang Corpus

The richness of social media platforms, such as Twitter and Weibo, enables users from different countries, even from different races and using different languages, to communicate effectively. Such a large diversity has led to the semantic development of terminology on the Internet, including Internet slang words. Many Internet slang words have evolved from existing words (denoted "new semantic words") or are newly created (denoted "new blend words"). These are constantly being introduced by users via social media platforms and can quickly become popular. It has widely been used and affects our daily life than ever. Semantic meaning of Internet slang words, however, could not be collected into the dictionary as well as a corpus in time, which is essential for people or machines to understand them.

In this study, we introduce a new resource containing Internet slang words specifiable at two levels: as a "major type" or as a "fine-grained subcategory" that subdivides words according to their semantics and construction. We aim to further analyze Internet slang words through fine-grained subcategories based on the internal construction and diversity of context which are learned with pre-trained embedding models. Then, a dual-task model is constructed to perform both two level classifications simultaneously, and the association between the two classifications is considered in the processing.

3.1 Main Type

By looking up popular Internet slang words in recent years, there are many new words using as a new meaning away from the usage written in the dictionary, while others are mainly new combinations of characters or words paired together by various new characters. Based on this characteristic, they were manually divided into two main types: "new semantic words" (SEM) and "new blend words" (BLN). Moreover, from a more detailed perspective, it is possible to find various patterns in the way new words are formed. In order to be able to further illustrate and analyze Internet slang words, combining this new word-formation with a specialized approach for Internet slang words, several fine-grained subcategories have been identified to help classify the details of lexical changes in the slang words.

3.1.1 New Semantic Words

"New semantic words" involve entries initially recorded in the dictionary and used daily. These words, however, now have new meanings that have become popular because of their

similarities to other terms or popular iconic events.

Although such vocabulary can be segmented directly, the context-based meaning is often very different from the original, and even the parts of speech can change. The New semantic words we selected of Japanese are shown in Table 3.1.

3.1.2 New Blend Words

"New blend words" often borrow foreign words, dialects, numeric elements, and icons by focusing on sound similarity. They often combine definitions, homonyms, abbreviations, repetitions, and other word-formation methods. They can also involve unconventional grammars [14]. Internet language has achieved the effect of "novelty" through its unconventional nature and nonstandard usage. The New blend words we selected of Japanese are shown in Table 3.2.

3.2 Fine-grained Subcategory

We identify subcategories by considering the formation of the Japanese Internet slang words. Examples for each subcategory are given in Table 3.3. Beside, a summary is given in Table 3.4.

1) *Gairaigo*[1]: the foreign words transliterated into Japanese and usually written in the "*Katakana*" phonetic script.

2) Japanese–English: the Japanese words look or sound just like English, but have different meanings from their English origins.

3) Dialect Borrowing: the words are derived from nonstandard Japanese dialects but have different meanings and contexts from the original words.

4) Compound Word: words created by joining together two (or more) root words.

5) Derived Word: words created by attaching affixes (which cannot be used in isolation) to the root words.

6) Abbreviation: words that omit some characters of existing words to create a shorter word form.

7) Acronymic: words composed of acronyms from multiple words.

8) Pun: each word is replaced with another word that sounds very similar but has a different meaning, thereby making the expression humorous.

9) Rhetoric: words with new and more specific meanings based on figurative expressions.

10) Neologism: words whose compositional characteristics are difficult to recognize from existing root words and affixes according to word-formation subcategories in 1–9.

[1] "Foreign words" in Japanese

3.3 Construction

The dataset used in our experiments was constructed from Japanese-language tweets accessed via Twitter's application programmer's interface[2]. A preprocessing stage removed any emoji or *kaomoji* (emoticon) data in the tweets. We then selected 100 Japanese Internet slang words whose meanings in Internet usage were specified in an online Japanese slang words collection[3], with 50 of the words being identified as "new semantic words" and the remainder as "new blend words." Among these words, we eliminated some ineligible words, such as those that had been updated or added to the standard Japanese dictionary, those that did not originate on the Internet, and those that had extremely low usage. Then based on the word formation features, 10 subcategories were finally determined for this set of Japanese Internet slang words. Next, we collected 50 sentences containing Internet slang used as such. For comparison, we collected another 50 sentences containing the same words but used in a general sense. Please note that the texts in which contain the selected Internet slang words may also contain other Internet slangs, including those outside the 100 words we selected. We have also annotated those words, therefore the actual samples in our corpus contain more Internet slangs than we planned at first.

In the annotation step, three native Japanese annotators identified Internet slang words in the sentences and labeled their types and subcategories. Most of the Internet slang words can be clearly distinguished from the words with common usage, but in a few cases annotators were unable to agree the results, because of incorrect usage by the tweet users or insufficient information in the short tweets. Finally, we decided to exclude such ambiguous cases from the corpus.

The collected Japanese sentences were then segmented into character units, subword units (via the SentencePiece algorithm[4]), and word units (via MeCab[5]). Finally, we tagged the characters, using the *BIO (B-Begin I-Inside O-Others)* tagging style to represent positional information. Examples are given in Table 3.5.

[2]https://developer.twitter.com/en/docs
[3]https://numan.tokyo/words/
[4]https://github.com/google/sentencepiece
[5]https://www.mlab.im.dendai.ac.jp/~yamada/ir/MorphologicalAnalyzer/MeCab.html

Table 3.1: Examples of New Semantic Words.

No.	**SEM**	**Netword Meaning**	**Etymology**	**sub**
1	草	grass	interesting	n
2	語彙力	vocabulary	strong vocabulary made people impressed	r
3	安定	stability	It's OK if you start to do it	r
4	定期	regular	common sense	r
5	裏山	back hills	envy	p
6	虹	rainbow	2 dimensional	p
7	鯖	mackerel	server	p
8	杉	cedar	too much	p
9	池沼	pond	intellectual disability	p
10	垢	plaque	Account	p
11	炎上	under fire	flame	r
12	鉄板	iron plate	a sure thing	r
13	地雷	mine	minefield	r
14	沼	swamp	to be addicted to	r
15	密林	jungle	Amazon	r
16	丸い	Round	safe	r
17	乙	the 2nd	It's a hard/great day today.	p
18	空気	air	Thin presence, atmosphere	r
19	安価	low price	Anchors	p
20	囲い	enclosure	passionate fans	r
21	三密	three dense	Three C's	ac
22	写メ	photo	Shooting without the intention of transmitting	r
23	ギガ	giga	go over the limitation of mobile data	r
24	左側	left side	The active partner in boys' love	r
25	ガッツ	guts	fan support with guts	r
26	尊い	precious	highly regarded	r
27	砂を吐く	spitting sand	ship	r
28	強火	strong fire	the passionate love of a particular idol	r
29	地蔵	Jizo	motionless artist	r
30	してもろて	do for me	"guide" in game	db
31	知らんけど	I don't know	closing statement after claims	db
32	枯れる	wither	selling out of idols' product	r
33	投げ銭	throwing money	social tipping	r
34	単騎	single driver	attending concerts or coterie magazine sales alone	r
35	通常運転	normal operation	as usual	r
36	履修	taking a course	collecting information on all genres related to the otaku circles	r
37	新規	new	a new fan of doujin and idol	r
38	積む	loading	leave a pile of games or books	r
39	全通	all through	attending all the performances	r
40	ブーメラン	boomerang	the phenomenon that criticism and bad words that you utter come back to yourself	r
41	沸いた	boiling	excitement	r
42	茶の間	tea room	fan support at home	r
43	天井	ceiling	maximum amount of *gacha* for social games	r
44	最大手	biggest organization	major circles of doujin field	r
45	遠征	expedition	travel long distance for otaku events	r
46	過疎	underpopulation	old fashion in otaku field	r
47	聖地	holy land	A place that has been the setting for or is associated with a manga, anime or film.	r
48	学級会	student council	A fan's actions or comments can lead to other fans arguing about etiquette, leading to a huge debate within or outside the genre.	r
49	在宅	at home	A fan who cheers at home, or his or her actions.	r
50	世紀末	end of the century	A decadent view of the world	r

Table 3.2: Examples of New Blend Words.

No.	**BLN**	**Netword Meaning**	**Coming From**	**sub**
1	わかりみ	understanding	“わかる”(understand) with nominalization	dw
2	禿同	strongly agree	sounded like “激しく同意”	p
3	ふぁぼ	Favorite	from English “favorite”	je
4	ワンチャン	one chance	from English “One chance”	je
5	そマ	Is that really?	from “それマジ”	ab
6	すこ	Favorite	derived from “好”	dw
7	イケボ	a voice from handsome guy	“イケメン”(handsome guy) + “ボイス”(voice)	je
8	パリピ	Party people	transmitted from “パーティピーポー”	je
9	わろた	LOL	derived from “笑える”(laugh)	dw
10	秒で	quickly	“秒”(second) +“で”(at)	c
11	やばたん	dangerous	“やばい”(dangerous) + light-hearted expression “たん”	dw
12	かまちょ	need you attention	from “かまってちょうだい”	ab
13	すきピ	people I like	from “好(す)きなピーポー”	ab
14	メンブレ	mental break	from “メンタルブレイク”	je
15	ガチ勢	prudent person	from “ガチ”(seriousness) and a noun “勢”(an army or force)	dw
16	マジ卍	unbelievable	“マジ”(really) + an emotional exclamation “卍”	c
17	ヌルゲー	easy game	from “ヌルいゲーム”	ab
18	リアタイ	real time	from “リアルタイム”	ab
19	とりま	For the time being, well	from “とりあえず まぁ”	ab
20	kp	cheers before drinking	Acronymic of “乾杯”(kanpai)	ac
21	タヒる	die	a horizontal line with katakana “タヒ” resembles the character for “死”(death)	n
22	推し事	activities to support idols	from “推しを応援するためにする活動の事”	ab
23	過去1	No. 1 things in the past	from “過去に経験した中でこれは1番...”	ab
24	絶起	got up late	from “絶望の起床”	ab
25	ヌッパン	TV show where guests lie down	from Korean word	g
26	陰キャ	introverted	from “陰気なキャラクター(character)”	ab
27	陽キャ	Outgoing	from “陽気なキャラクター(character)”	ab
28	常考	in general	an abbreviation of “常識的に考え...”	ab
29	はにゃ	exclamation	an doubtful exclamation word	n
30	逆さ撮り	unpopular photography	“逆さ”(reverse) + “撮り”(photography)	c
31	がこおわー	finish school	from “学校(がっこう)が終(おわ)わったよ”	ab
32	ぴえん	sad	exclamation words that express sad feelings	n
33	ぽえん	happy	exclamation words that express happy feelings	n
34	ばおん	antonym	antonym, a word that expresses the opposite feeling	n
35	ミーム	meme	from English “meme”	g
36	チルする	chill	English word “chill” with a verb affix“する”	c
37	アセアセ	very panic	“汗(アセ)”(means sweating)	r
38	キャス変	cast changed	used when the cast (actor) who was originally in charge of the stage or musical has changed for some reason.	ab
39	チー牛	introverted	from “チーズ牛丼”	ab
40	専オタ	professional nerd(fan)	fans who only come to support the scene where a specific idol appears	ab
41	おしゃピク	fashionable picnic	an abbreviation of “おしゃれなピクニック”	ab
42	人生RTA	competing in real-time to clear the game quickly	“人生(life)” with RTA(Real-time attack)	c
43	無理ぽ	it seems impossible	from ”もう無理っぽい”	ab
44	きゅんです	throbbing	an onomatopoeia ”きゅん” + a noun affix “です”	c
45	テッテレー	cheers	a kind of fanfare and sound effect for success	n
46	神現場	the best live performance of an idol	“神”(god)+“現場”(live)	r
47	金コマ	debtor	from “お金に困(コマ)ったいる人”	ab
48	おけまる	ok.	“おけ”(OK) + “まる(.)”	je
49	ツイ廃	Twitter addicts	”ツイッター(twitter)” + “廃人”(addicts)	je
50	モッパン	eating show	from Korean	g

Table 3.3: Examples of Internet Slangs for each Subcategory

Subcategory	**Word**	**Etymology**
Gairaigo	ミーム	From English Slang Word *meme*.
Japanese-English	ワンチャン	With the same pronunciation as *One Chance*.
Dialect Borrowing	してもろて	From the dialect in Kansai district of “してもらって” (let someone do).
Compound Word	秒で	Combination of the noun “秒” (second) and the auxiliary “で” (at).
Derived Word	わかりみ	A verb “わかる” (understand) with a noun suffix “み” (-ing) to nominalize the verb.
Abbreviation	そマ	An abbreviation of the sentence “それマジ” (Is that really?)
Acronymic	三密	*Three Cs* (crowded places, close contact settings, and closed spaces)
Pun	鯖	With the same pronunciation as *server*.
Rhetoric	世紀末	A decadent worldview extended from the end of the century.
Neologism	タヒる	Katakana characters combination “タヒ” is morphologically similar to “死” (death).

Table 3.4: Subcategories of New Semantic Words (SEM) and New Blend Words (BLN).

Subcategory	**Tag**	**SEM**	**BLN**	**Total**
Gairaigo	g	\	3	3
Japanese-English	je	\	7	7
Dialect Borrowing	db	2	\	2
Compound Word	c	\	8	8
Derived Word	dw	\	5	5
Abbreviation	ab	\	19	19
Acronymic	ac	1	1	2
Pun	p	8	1	9
Rhetoric	r	38	\	38
Neologism	n	1	6	7

Table 3.5: Examples of Annotations of Japanese Internet Slang Words.

New Semantic Words
-Internet Usage 初/O **鯖**/ B-sem | p の/O 初/O 心/O 者/O に/O 迷/O惑/O か/Oけ/Oる/O な/O !/O
-Common Usage 脂/O の/O 乗/O っ/O た/O **鯖**/O の/O 塩/O 焼/O き/O と/O か/O と/O 合/O わ/O せ/O た/O い/O
New Blend Words
-Internet Usage **そ**/B-bln|ab **マ**/I-bln | ab **？**/O 行/O け/O る/O 時/O 言/O っ/O て/O バ/O イ/O ト/O 無/O け/O れ/O ば/O ワ/O イ/O も/O 行/O く/O わ/O
-Common Usage 今/O 日/O こ/O **そ**/O **マ**/O へ/O ラ/O し/O ま/O す/O !/O

Chapter 4

Our Proposed Model

4.1 System Overview

Our proposed approach combines character embedding and word embedding for each token in the embedding layer to obtain a new ELMo representation. The hierarchical shared BiLSTM is used in the context encoder to obtain the dependencies between the two layers of tags and the semantic information corresponding to the words according to the Internet slang corpus. For the two layers of BiLSTM, we also set up two separate ways of sharing hidden layer information and independent. In the tag decoder, a CRF is used as the annotator of the sequences to ensure the logical order between the sequences.

A joint embedding based on character embedding and word embedding is used to capture the relationship between contextual words simultaneously, and to obtain features for this internal structure. In addition, the ELMo model can distinguish dynamic representations generated by semantic differences through contextual information. As the subcategories are subdivided according to the main types, we consider that there is a strong correlation between the two levels of labels set, and during the training phase it was also decided to build a multitasking language model that can process both tasks and pass the parameters of the predecessor task between the levels like a back-end task, thus allowing the model to learn the relationship between the two labels. The multi-task language model first detects the main type at the upper level and then passes it to the lower level tasks to identify fine-grained sub-categories.

The addition of a CRF layer on the output layer of the BiLSTM neural network model retains the advantage of contextual information and enhances the relevance of contextual information. For sequential annotation, an annotation result is given for each word in a given sentence. High-dimensional features are extracted for the i-th word in the sentence, and by learning the mapping of features to annotation results, the probabilities of features to arbitrary labels are obtained. The best sequential results can be estimated with these probabilities. Thanks to the CRF layer, the influence between the before and after annotations at the sentence level can be taken into account, rather than just a simple dynamic programming process on the output of the neural network layer.

4.2 Joint Model Using Character and Word Embeddings

The structure of Joint Embedding is given in Fig. 4.1.

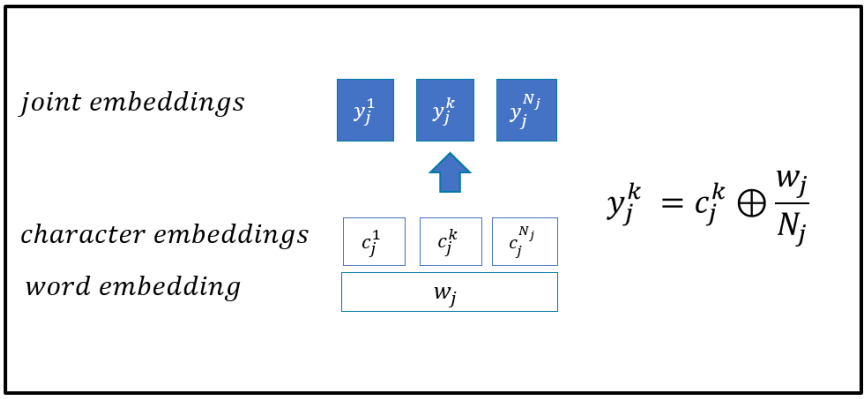

Figure 4.1: Structure of Our Proposed Model

In Fig. 4.1, the parameter w_j is the word embedding of token j, N_j is the number of characters in this token, c_j^k is the embedding of the k-th character, and y_j^k is joint embedding for the character-level unit annotation, for which the relationship between these parameters is given in equation (4.1).

$$y_j^k = c_j^k + \frac{w_j}{N_j} \tag{4.1}$$

4.3 ELMo Embedding

To pre-train the ELMo model [28] , we also used Word2vec Skip-gram with Negative Sampling [24] to obtain the standard word dictionary and word vectors by the Japanese Wikipedia Dataset[1], for which all of the layers were combined via a weighted-average pooling operation. The output of the ELMo model is given in equation (4.2).

$$\mathbf{ELMo}_k = \gamma \sum_{j=0}^{L} s_j \mathbf{h}_{k,j} \tag{4.2}$$

Here, k represents the position of each (character-level) token and j is the number of layers. $\mathbf{h}_{k,j}$ is the hidden state output for each BiLSTM layer. The parameter s represents the Softmax-normalized weight, and the scalar parameter γ allows the task model to scale the entire ELMo vector. γ could enhance the optimization process.

[1]https://dumps.wikimedia.org/jawiki/latest/ [accessed on October 2020]

4.4 Multi-Task Learning

Because the first layer of BiLSTM cannot determine with certainty both shared-LM and unshared-LM based on prior research before the experiment, it obtains the dependencies of subcategories based on the context and imports the CRF layer to determine the most probable label for the token, and then passes the hidden layer information of the model into the next layer of BiLSTM. Moreover, in the process of passing shared-LM will pass both forward and backward weight information, while unshared-LM will pass only the backward output information. In the second layer of BiLSTM, after receiving the information about the main type, the main types are processed according to the context and the CRF is also imported to infer the tag corresponding to the token. The $h^{subcategory}$ and $h^{maintype}$ are given in equation (4.3) and equation (4.4).

$$\mathbf{h}^{subcategory} = \mathbf{BiLSTM}^{subcategory}(\mathbf{x}) \tag{4.3}$$

$$\mathbf{h}^{maintype} = \mathbf{BiLSTM}^{maintype}\left(\left[\mathbf{x}; \mathbf{h}^{subcategory}\right]\right) \tag{4.4}$$

The information transfer ground equation is as follows, where $\boldsymbol{\lambda}$ is the parameter controlling the influence of the language modelling task to the sequence annotation task and $\overleftarrow{E}_{LM}$ and $\vec{E}_{LM}$ are the objective functions for the forward and backward language models.

$$E_{\text{joint}} = E + \lambda\left(\overrightarrow{E}_{LM} + \overleftarrow{E}_{LM}\right) \tag{4.5}$$

$$\overrightarrow{E}_{LM} = -\sum_{t=1}^{T} \log\left(P\left(w_{t+1} \mid \overrightarrow{h_t}\right)\right) \tag{4.6}$$

$$\overleftarrow{E}_{LM} = -\sum_{t=1}^{T} \log\left(P\left(w_{t-1} \mid \overleftarrow{h_t}\right)\right) \tag{4.7}$$

where $\overrightarrow{h}_t, \overleftarrow{h}_t$ are the hidden states of the forward and backward LSTMs and w_{t-1}, w_{t+1} are the preceding and latter tokens.

We construct bilayer BiLSTM network as the hierarchical multi-task implementation module, and the forward and backward inputs in BiLSTM are connected as the output of each layer of BiLSTM network. In shared-LM, this connection is shown in the Equation (4.8), the forward and backward LSTMs of the first subcategory layer will all be passed to the second layer main type layer. In unshared-LM, the fusion of forward and backward information will be input the CRF layer to determine the subcategory's tag, and when tranmitted to the main type layer, only the backward information is passed as in the Equation (4.9).

$$E_{\text{shared}} = E + \lambda\left(\overrightarrow{E}_{LM} + \overleftarrow{E}_{LM}\right)_{subcategory} \tag{4.8}$$

$$E_{\text{unshared}} = E + \lambda\left(\overleftarrow{E}_{LM}\right)_{subcategory} \tag{4.9}$$

The structure of our model is given in Fig. 4.2.

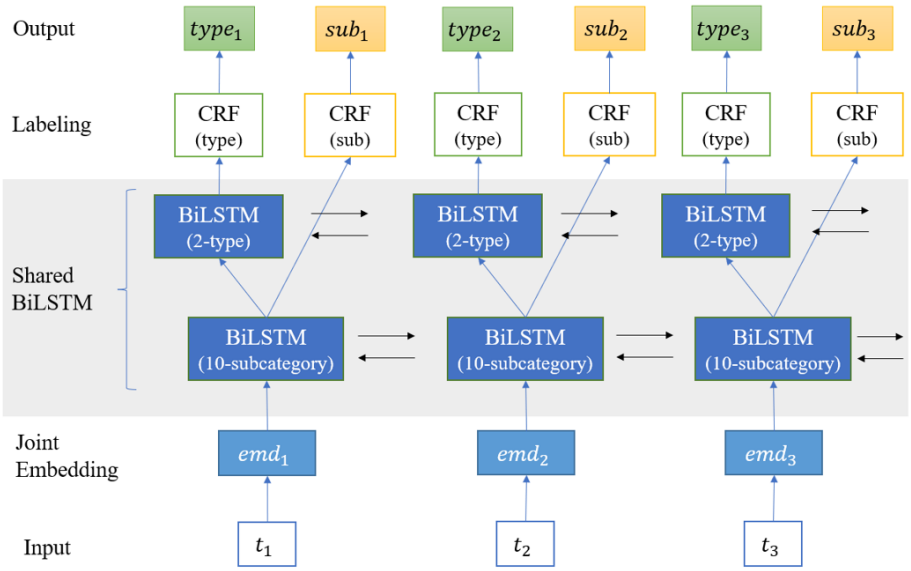

Figure 4.2: Structure of Our Proposed Model for our two tasks

Chapter 5

Experiments

The purpose of this experiment is to demonstrate whether our proposed model is effective in detecting online phrases and whether it can correctly determine the categories of online phrases. First, we describe the corpus preparation for the input model in the experiment and the baseline methods used for comparison. After that, we compare the parameters of our model with those of the baseline model for the experiments. Finally, in order to test the effectiveness of our proposed ELMo multi-task learning model based on the joint embedding of character and word embedding, we compare the effect of single task and multi-task models separately and use precision, recall and F1-score as evaluation metrics for the main type, and F1-score is used as evaluation metrics for subcategory.

5.1 The Dataset and Preprocessing

5.1.1 Pre-training Dataset

To unify all the pre-trained models, we chose the Japanese Wikipedia dataset. As a pre-training dataset, the number of words and characters is 312,000 and 10,000, respectively. Pre-training the model enables it to learn the basic standard Japanese language and thus obtain the corresponding token representation. For LSTM, we use Word2vec Skip-gram to obtain static character embeddings and word embeddings.

In this case, the vector dimension of character embedding is set to 200 and the context window size is 10. While for word embedding, the vector dimension is 200 and the window size is 5. We also adjusted other settings to match the settings in the Japanese Wikipedia Entity Vector[1].

5.1.2 Fine-tuning Dataset

After pre-training the models, we trained the models by dividing the Internet slang corpus into five small datasets using the five-fold cross-validation method; each dataset had 80% of training data and 20% of test data. In addition, 20% of the training data were randomly selected as the validation set in order to tune the parameters (determine the required batch size and epoch number for training completion, etc.). Thus, the training set has 6,400 samples, the validation set has 1,600 samples, and the test set has 2,000 samples.

[1] http://www.cl.ecei.tohoku.ac.jp/ ~m-suzuki/jawiki_vector/

5.2 Baseline System

The aim of this study is to investigate the performance of the joint embedding model of characters and words in terms of detecting Internet slang words using ELMo embedding, and to explore the superiority of multi-tasking over single-tasking. We compared it with three baseline methods based on LSTM models, ELMo models, and BERT method as follows.[2].To examine the differences in multi-task, we also used both shared-LM and unshared-LM regarding multi-task learning methods of LSTMs and ELMo .

1. **LSTM-c**: Bidirectional LSTM network based on character-only embedding.
2. **LSTM-w**: Bidirectional LSTM network based on word-only embedding.
3. **LSTM-cw**: Bidirectional LSTM network based on character and word embedding.
4. **ELMo-c**: ELMo method based on character-only embedding.
5. **BERT**: BERT method based on subword embedding.

The pre-trained BERT model we use is Laboro-BERT-Japanese [47], which is also trained by the Japanese Wikipedia dataset. In terms of multi-task methods, the token was annotated from two-layers separately in order to be consistent with our proposed model. We refer to PhoNLP [27] for modification. As shown in the Fig. 5.1, the first layer Attention learns the information of subcategory and passes it to the next layer to learn the main type. We trained and tested the model with each of the five datasets after cross-validation segmentation and calculated the averaged precision, recall, and F1-score of the Internet slang assigned to the main type or subcategory.

[2]"c" and "w" denote character and word embeddings, "cw" denotes the joint embedding of character and word embeddings

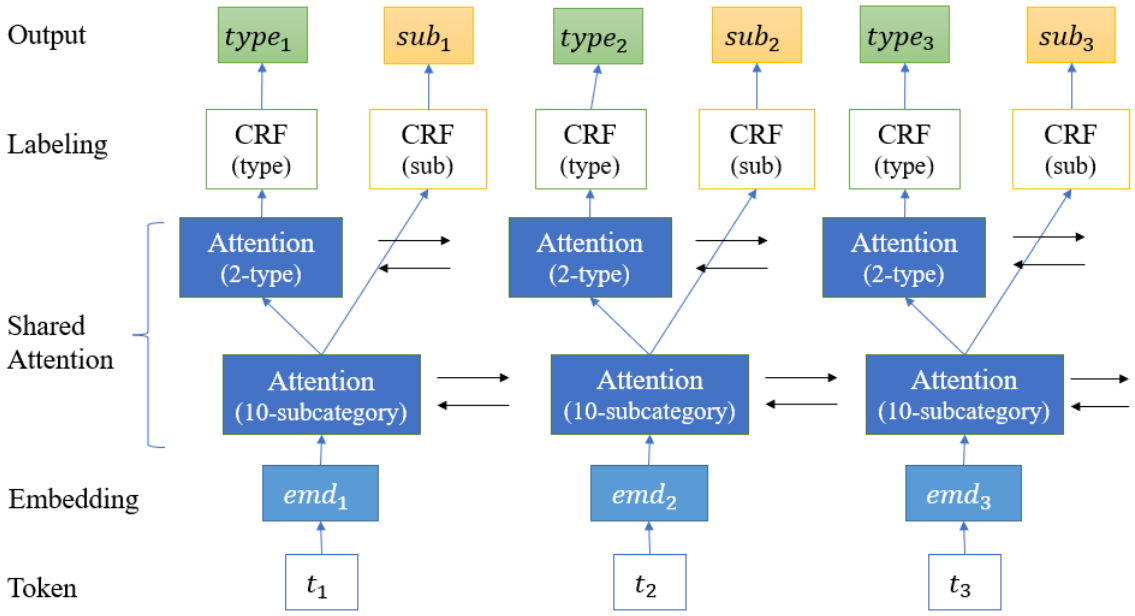

Figure 5.1: Structure of BERT for our two tasks

5.3 Evaluation

In order to verify the validity of the models, we compared the various embedding models. We set up a series of contextual encoder models. In addition, we also implemented the results of a single-task model for detecting the main types and the subcategories at the outset to determine the performance of the multitasking model in detecting both types simultaneously. The equations of the evaluation metrics are as follows:

There are four situations for the data test results:

- True positive (TP): Correctly detected the Internet slang words.
- True negative (TN): Incorrectly detected the Internet slang words.
- False positive (FP): Correctly detected the common words.
- False negative (FN): Incorrectly detected the common words.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5.1}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5.2}$$

$$\text{F1-score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5.3}$$

5.4 Implementation

Before the experiments, all the input data, are formatted in the CoNLL-2003 [37] style. Then, changing the model settings and running the model on a trial basis according to the trend of train loss and validation loss so that we could confirm the proper value to fit models. The settings of each model are shown in the Table. 5.1.

Table 5.1: Settings of each model

Model	**Embedding-level**	**Dimension**	**Single task**			**Multi-task**		
			Epoch	**Batch**	**Learning rate**	**Epoch**	**Batch**	**Learning rate**
LSTM-c	Character	200	20	16	0.01	30	16	0.01
LSTM-w	Word	200	20	16	0.01	30	16	0.01
LSTM-cw	Character + Word	200 + 200	20	16	0.01	30	16	0.01
BERT	Subword	768	20	8	2e-5	20	8	2e-05
ELMo-c	Character	1024	30	16	1e-05	50	16	1e-03
ELMo-cw	Character + Word	1024	30	16	1e-05	50	16	1e-03

5.5 Overall Results

From the results given in Table5.2, all models were better at recognizing "new blend words" (BLN) than recognizing "new semantic words" (SEM). Multi-task BERT performs the best, and our model works better than the rest of the models. In addition to the multi-task BERT model, other models, including single task BERT, are more likely to detect subcategories than main types.

From the results in Table5.3, except for LSTM-cw which performs best on "*Gairaigo*", our model reached the highest average F1-score (0.885) in detecting subcategories. In the case of shared-LM model in detecting "Japanese-English", "Dialect Borrowing", "Compound", "Derived Word", and "Abbreviation", "Compound", "Derived Word", "Abbreviation", "Rhetoric" and "Neologism" showed superiority. Moreover, unshared-LM worked best on "Acronymic", and "Pun". Compared to the results of the single task, our model was greatly improved by information transfer from two-layers annotation.

Overall, unshared-LM works better for the LSTM models, while for ELMo, shared-LM outperformed unshared-LM. Our model with multi-task results were, on average 32.5% better in F1-score than the single task. The combined results of multi-task with two-layers

annotation simultaneously showed that our proposed model with shared-LM has achieved the best performance.

Table 5.2: Results of Detecting "New Semantic Words" and "New Blend Words".

Methods \ Main.		SEM			BLN		
		F1-score	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**
Single	LSTM-c	0.303	0.381	0.281	0.485	0.482	0.487
	LSTM-w	0.094	0.357	0.054	0.200	0.404	0.134
	LSTM-cw	0.359	0.371	0.349	0.488	0.485	0.490
	BERT	0.405	0.376	0.438	0.593	0.576	0.612
	ELMo-c	0.429	0.412	0.453	0.577	0.579	0.577
	ELMo-cw	0.460	0.451	0.470	0.602	0.596	0.607
Multi	BERT	<u>0.798</u>	<u>0.855</u>	0.749	<u>0.926</u>	<u>0.946</u>	0.908
Multi shared	LSTM-c	0.420	0.520	0.651	0.590	0.685	0.851
	LSTM-w	0.351	0.393	0.396	0.495	0.491	0.578
	LSTM-cw	0.653	0.580	0.778	0.833	0.788	0.891
	ELMo-c	0.673	0.576	<u>0.839</u>	0.814	0.759	0.894
	ELMo-cw	0.757	0.705	0.825	0.915	0.881	<u>0.953</u>
Multi unshared	LSTM-c	0.453	0.490	0.572	0.658	0.687	0.695
	LSTM-w	0.546	0.449	0.713	0.565	0.570	0.676
	LSTM-cw	0.609	0.586	0.687	0.781	0.716	0.882
	ELMo-c	0.636	0.582	0.758	0.811	0.733	0.919
	ELMo-cw	0.730	0.755	0.667	0.905	0.913	0.886

5.6 Discussion

5.6.1 Analysis of the results on main type

We compared LSTM, BERT and ELMo with the single task learning models, multi-task learning models where shared-LM and unshared-LM methods were used by LSTM and ELMo, totaling 15 models. Multi-task BERT model reached the best results, it showed that the subword, which at fine granularity between character and word, can recognize semantic

Table 5.3: F1-scores for Detecting Fine-grained Subcategories. "**" denotes cases where the difference in macro average between Our Model and other methods is statistically significant for $p < 0.01$ using two-tailed paired samples t-tests.

Methods \ Subcat.		*Gairaigo*	Japanese-English	Dialect Borrowing	Compound	Derived Word	Abbreviation	Acronymic	Pun	Rhetoric	Neologism	Avg.	Significance
Single	LSTM-c	0.588	0.492	0.682	0.593	0.483	0.326	0.296	0.386	0.312	0.282	0.444	**
	LSTM-w	0.393	0.150	0.896	0.363	0.334	0.220	0.246	0.153	0.062	0.298	0.312	**
	LSTM-cw	0.762	0.596	0.898	0.596	0.486	0.325	0.374	0.416	0.361	0.267	0.508	**
	BERT	0.676	0.627	0.593	0.631	0.622	0.378	0.461	0.56	0.414	0.316	0.528	**
	ELMo-c	0.625	0.545	0.616	0.631	0.572	0.35	0.421	0.496	0.458	0.329	0.504	**
	ELMo-cw	0.639	0.600	0.596	0.627	0.598	0.368	0.486	0.532	0.486	0.355	0.529	**
Multi	BERT	0.718	0.635	0.555	0.647	0.629	0.429	0.553	0.475	0.514	0.614	0.577	**
Multi shared	LSTM-c	0.754	0.807	0.591	0.782	0.745	0.790	0.616	0.554	0.694	0.715	0.674	**
	LSTM-w	0.554	0.532	0.388	0.556	0.644	0.314	0.594	0.487	0.111	0.584	0.476	**
	LSTM-cw	0.950	0.852	0.768	0.895	0.879	0.866	0.678	0.769	0.672	0.831	0.816	
	ELMo-c	0.950	0.899	0.887	0.937	0.937	0.910	0.735	0.816	0.673	0.841	0.858	
	ELMo-cw	0.921	0.909	0.907	0.946	0.940	0.921	0.831	0.874	0.752	0.851	0.885	-
Multi unshared	LSTM-c	0.928	0.859	0.693	0.868	0.771	0.844	0.583	0.645	0.603	0.771	0.757	**
	LSTM-w	0.655	0.804	0.575	0.751	0.507	0.333	0.783	0.584	0.285	0.721	0.629	**
	LSTM-cw	0.950	0.899	0.823	0.900	0.893	0.899	0.634	0.791	0.617	0.826	0.823	
	ELMo-c	0.953	0.908	0.865	0.894	0.911	0.901	0.695	0.826	0.660	0.827	0.844	
	ELMo-cw	0.927	0.904	0.881	0.937	0.932	0.912	0.842	0.879	0.745	0.842	0.880	

differences according to the context and segment OOV words correctly.

On the other hand, new blend words have better detection results than new semantic words in terms of main type. This indicates that the model's performance in identifying semantic differences is not as good as that of OOV words. New semantic words are intrinsic words, either as common usage or Internet usage, which have several corresponding collocations, which require the model to determine the current meaning based on the contextual collocations. In contrast, new blend words have no common fixed collocations and only correspond to Internet usage, so they are easier to identify because they have fewer contextual interference items with high weights when training the model.

5.6.2 Analysis of the results on subcategory

The comparison results of 15 models showed that our proposed model achieved superiority. Our model detected "Acronymic" and "Pun" most accurately with the unshared-LM methods, LSTM-cw performed best on "*Gairaigo*", and for the remaining seven subcategories, our model was most effective with the unshared-LM method. In contrast, the multi-task BERT model, which performs best on the main type, performed poorly, even inferior to LSTM-c. When we compared the results of the single task, BERT performed best on "Japanese-English", "Compound", "Derived Word", "Abbreviation", and "Pun", but when in the case of a multi-task with hierarchical annotation, the performance improvement effect of BERT is not as fast as the other models.

5.6.3 Comparison of embedding models

Both LSTM and ELMo based on the joint embedding of character and word embeddings performed better than the character or the word embeddings only. By comparing the results of LSTM networks, using only word embeddings is worse than character embeddings. Considering that the Twitter messages we collected are short, the high space complexity, which was required to compute character embeddings, did not cause excessive performance loss. Similarly, using only subword embedding was much better than word embeddings only. From these results, we conclude that it is necessary to cut words into finer-grained units when identifying new words.

ELMo and BERT generate vectors based on the current context of the token, which we call dynamic vectors. The Word2vec used by LSTM networks, on the other hand, generates fixed vectors based on pre-trained data. In other words, the representations are always same even the usage or meaning is different. Therefore, ELMo and BERT can retrain from the Internet slang corpus during fine-tuning and generate the more adapted vectors to distinguish semantic differences.

5.6.4 Comparison of single task with multi-task

The experimental results showed that the same model outperformed a single task in multi-task, whether it recognized two main types or ten subcategories, demonstrating the strong correlation between two-layers annotation. Using a two-layer model for hierarchical output helps to improve the performance of label recognition in the lower layer. Among them, the

F1-score of ELMo-cw improved by 32.5% on average compared with that of single task, and the number of subcategories with the best results increased from three to seven.

We also note that the attention-based BERT achieved optimal performance for the underlying main type recognition, but correspondingly, the performance improvement was not as good as that of the BiLSTM-based models for the subcategory recognition. Our reference BERT model was designed to improve the effect of lower labelling. Therefore, we conjectured that the information acquired by the upper attention in training the subcategories was less than the one passed to the lower main types in the process of information transfer and the difference of transmission leaded to the issue.

5.6.5 Comparison of shared-LM with unshared-LM

Based on the comparison of the shared-LM and unshared-LM methods for ELMo and LSTM, the shared-LM outperformed the unshared-LM for ELMo, while the unshared-LM outperformed the shared-LM for LSTM. As illustrated in Section 2.4.3, shared-LM shares bidirectional LSTM for both subcategory and main type labelling tasks. In other words, when using shared-LM, the hidden weights of backward LSTM and forward LSTM were both passed to the second layer. Unlike the unshared-LM, the second layer of shared-LM obtained more information. What's more, the richer hidden weights prompted the ELMo model to learn more during fine-tuning, and therefore it was easier to detect subcategories.

In our two-layer system, the output of the main type layer is affected by token embeddings, the hidden state obtained from the subcategory layer, and the cell state of the current layer. We concatenate the forward and backward hidden vectors (and also for the forward and backward cell vectors). Then, the memory cells transmit useful information and forget useless information according to the concatenated hidden state. For shared-LM, due to the fixed token embedding and shared hidden states from the two layers, the memory of similar information is transmitted to the deeper layer when calculating the hidden state of the second layer, which misses the useful information. Thanks to the independent layers in BiLSTM, unshared-LM will not face the interference of repeated memory.

Chapter 6

Conclusion

6.1 Summary

In this paper, we constructed a new 10,000-sample Internet slang corpus, in which 100 Internet slang words have both common usage and Internet usage, and proposed a two-layers annotation, main types and subcategories, by analyzing popular Internet slang words in recent years based on their definitions and word-formation features. Also, a joint embedding based on character and word embeddings was proposed, and the ELMo multi-task learning method was applied for detecting two-layers annotation. Our experiments showed that our proposed model with shared-LM was superior to LSTMs in detecting the main types of Internet slangs and performed best when detecting subcategories, even outperformed multi-task BERT.

The main contributions of this paper are as follows:

1. By analyzing the popular Internet slang in recent years, we proposed a new Internet slang corpus with 10,000 samples and designed two-layers annotation in terms of definition and word-formation. One is the main type, including new semantic word and new blend word. The other is fine-grained subcategories, including "*Gairaigo*", "Japanese-English", "Dialect Borrowing", "Compound", "Derived Word", "Abbreviation", "Abbreviation", "Acronymic", "Pun" and "Rhetoric".

2. We proposed a joint embedding model based on character embedding and word embedding, which can learn the internal structure of words and understand the relationship between words according to the context, and worked much more effectively than character embedding only or word embedding only.

3. Our proposed ELMo multi-task learning method (shared-LM) is the best in detecting subcategories, and it is also better than other LSTM models in detecting the main type.

6.2 Future Work

After analyzing Chinese Internet slang words, we found that the definitions and word formation methods are also applicable to the rules set in this experiment. Therefore, we expect to collect Chinese Internet slang words and create a Chinese Internet slang corpus. Then, to figure out whether our proposed model is also superior in detecting Chinese Internet slang. Our model still needs to be improved in the detection of the main type, and we also found that the research on ELMo model based on subword embeddings is still rare, especially for non-English language processing, so we will try to use subword embeddings with our ELMo multi-task learning method to see how it works.

Acknowledgements

I am sincerely thankful to my supervisor, Associate Professor Yohei Seki, for his patience, enthusiasm, professionalism and advice on how to improve my work. I was impressed by his knowledgeable and progressive spirit and his serious and rigorous approach to research. I would also like to thank Assistant Professor Hai-Tao Yu for his support and encouragement as an associate supervisor.

Besides, I would also like to thank Associate Professor Wakako Kashino from the National Institute for Japanese Language and Linguistics for her professional advice in creating the Internet slang corpus, especially the definition of subcategories and the classification of Internet slangs. It is her professionalism that I could create a rigorous and dedicated corpus due to f her professionalism and excellent analytical skills.

I am grateful to my two tutors, Masaki Oguni and Tetsuya Ishida, who helped me with my studies, research and campus life as an international student, their enthusiasm and responsibility made me feel warm in a foreign country. In addition to them, other members of our lab, Ko Senoo, Seiji Nakayama and Mei Taniguchi, also helped me to classify and annotate Japanese Internet slang words and explain the knowledge to me patiently and carefully. Also Kangkang Zhao, who help me implement and understand the techniques of models.

I would also like to thank my schoolmates, Lirong Zhang and Jujie Xu, for studying, sharing learning skills and experiencing the campus life together. During the COVID-19 period, studying abroad became different and challenging, so last my thank would go to my parents for their support and my pets for being with my parents instead of me, so that I could focus on my research without worrying about my family. Best wished to all the teachers and staffs of University of Tsukuba for their assistance and help in my student life, and I would also like to thank my friends and schoolmates for their concern and encouragement.

References

[1] Johannes Bjerva. Will my auxiliary tagging task help? estimating auxiliary tasks effectivity in multi-task learning. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 216–220, 2017.

[2] Michal Bojkovskỳ and Matúš Pikuliak. STUFIIT at SemEval-2019 Task 5: Multilingual hate speech detection on twitter with MUSE and ELMo embeddings. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 464–468, 2019.

[3] Xinxiong Chen, Lei Xu, Zhiyuan Liu, Maosong Sun, and Huanbo Luan. Joint Learning of Character and Word Embeddings. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, page 1236–1242, 2015.

[4] Jason PC Chiu and Eric Nichols. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4:357–370, 2016.

[5] Jishnu Ray Chowdhury, Cornelia Caragea, and Doina Caragea. On Identifying Hashtags in Disaster Twitter Data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):498–506, 2020.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota, June 2019.

[7] Chen Gong, Zhenghua Li, Qingrong Xia, Wenliang Chen, and Min Zhang. Hierarchical LSTM with char-subword-word tree-structure representation for Chinese named entity recognition. *Science China Information Sciences*, 63(10):1–15, 2020.

[8] Guoliang Guan and Min Zhu. New Research on Transfer Learning Model of Named Entity Recognition. *Journal of Physics: Conference Series*, 1267(1):012017, 2019.

[9] Jiahao Hu, Yuanxin Ouyang, Chen Li, Chuanrui Wang, Wenge Rong, and Zhang Xiong. Hierarchical Lexicon Embedding Architecture for Chinese Named Entity Recognition. In *International Conference on Artificial Neural Networks*, pages 345–356. Springer, 2021.

[10] Zhenfei Ju, Jian Wang, and Fei Zhu. Named entity recognition from biomedical text using SVM. In *2011 5th international conference on bioinformatics and biomedical engineering*, pages 1–4. IEEE, 2011.

[11] Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. Character-Aware Neural Language Models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, page 2741–2749, 2016.

[12] Taku Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. *arXiv preprint arXiv:1804.10959*, 2018.

[13] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*, 2018.

[14] Fazal Masud Kundi, Shakeel Ahmad, Aurangzeb Khan, and Muhammad Zubair Asghar. Detection and scoring of Internet slangs for sentiment analysis using SentiWordNet. *Life Science Journal*, 11(9):66–72, 2014.

[15] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural Architectures for Named Entity Recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California, June 2016.

[16] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[17] Xiaoya Li, Yuxian Meng, Xiaofei Sun, Qinghong Han, Arianna Yuan, and Jiwei Li. Is Word Segmentation Necessary for Deep Learning of Chinese Representations? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, July 2019.

[18] Nut Limsopatham and Nigel Collier. Bidirectional LSTM for named entity recognition in Twitter messages. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 145—-152, 2016.

[19] Zihan Liu, Feijun Jiang, Yuxiang Hu, Chen Shi, and Pascale Fung. NER-BERT: A Pre-trained Model for Low-Resource Entity Tagging. *arXiv preprint arXiv:2112.00405*, 2021.

[20] Ying Luo, Fengshun Xiao, and Hai Zhao. Hierarchical Contextualized Representation for Named Entity Recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8441–8448, 2020.

[21] Wentao Ma, Yiming Cui, Chenglei Si, Ting Liu, Shijin Wang, and Guoping Hu. CharBERT: Character-aware Pre-trained Language Model. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING '20)*, pages 39–50, Barcelona, Spain (Online), December 2020.

[22] Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. *arXiv preprint arXiv:1603.01354*, 2016.

[23] G Maragatham and Shobana Devi. LSTM model for prediction of heart failure in big data. *Journal of medical systems*, 43(5):1–13, 2019.

[24] Chris McCormick. Word2Vec Tutorial Part 2 - Negative Sampling, 2017.

[25] Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 746–751, 2013.

[26] Sudha Morwal, Nusrat Jahan, and Deepti Chopra. Named entity recognition using hidden Markov model (HMM). *International Journal on Natural Language Computing (IJNLC) Vol*, 1, 2012.

[27] Linh The Nguyen and Dat Quoc Nguyen. PhoNLP: A joint multi-task learning model for Vietnamese part-of-speech tagging, named entity recognition and dependency parsing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 1–7, 2021.

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

[29] Thai-Hoang Pham, Khai Mai, Nguyen Minh Trung, Nguyen Tuan Duc, Danushka Bolegala, Ryohei Sasano, and Satoshi Sekine. Multi-task learning with contextualized word representations for extented named entity recognition. *arXiv preprint arXiv:1902.10118*, 2019.

[30] Yuval Pinter, Robert Guthrie, and Jacob Eisenstein. Mimicking word embeddings using subword rnns. *arXiv preprint arXiv:1707.06961*, 2017.

[31] Yuval Pinter, Cassandra L. Jacobs, and Max Bittker. NYTWIT: A Dataset of Novel Words in the New York Times. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6509–6515, Barcelona, Spain (Online), December 2020.

[32] Yuval Pinter, Marc Marone, and Jacob Eisenstein. Character Eyes: Seeing Language through Character-Level Taggers. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 95–102, Florence, Italy, August 2019.

[33] Marek Rei. Semi-supervised Multitask Learning for Sequence Labeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada, July 2017.

[34] Marek Rei and Helen Yannakoudakis. Auxiliary Objectives for Neural Error Detection Models. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 33–43, 2017.

[35] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[36] Kalyan Sundar Samanta and Durga Sankar Rath. Social Tags Versus LCSH Descriptors: A Comparative Metadata Analysis in the Field of Economics. *Journal of Library & Information Technology*, 39(4):145–151, July 2019.

[37] Erik F Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*, 2003.

[38] Mike Schuster and Kaisuke Nakajima. Japanese and Korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152. IEEE, 2012.

[39] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany, August 2016.

[40] Jivitesh Sharma, Charul Giri, Ole-Christoffer Granmo, and Morten Goodwin. Multi-layer intrusion detection system with ExtraTrees feature selection, extreme learning machine ensemble, and softmax aggregation. *EURASIP Journal on Information Security*, 2019(1):1–16, 2019.

[41] Cong Sun and Zhihao Yang. Transfer learning in biomedical named entity recognition: An evaluation of BERT in the PharmaCoNER task. In *Proceedings of the 5th Workshop on BioNLP Open Shared Tasks*, pages 100–104, 2019.

[42] Yaming Sun, Lei Lin, Nan Yang, Zhenzhou Ji, and Xiaolong Wang. Radical-enhanced Chinese character embedding. In *International Conference on Neural Information Processing*, pages 279–286. Springer, 2014.

[43] Matej Ulčar and Marko Robnik-Šikonja. High Quality ELMo Embeddings for Seven Less-Resourced Languages. In *Proceedings of the 12th Language Resources and Evaluation Conference (LREC '20)*, pages 4731–4738, Marseille, France, May 2020. European Language Resources Association.

[44] Yining Wang, Long Zhou, Jiajun Zhang, and Chengqing Zong. Word, subword or character? an empirical study of granularity in Chinese-English NMT. In *China Workshop on Machine Translation*, pages 30–42. Springer, 2017.

[45] John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. Charagram: Embedding Words and Sentences via Character n-grams. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1504–1515, Austin, Texas, November 2016.

[46] Hang Yan, Bocao Deng, Xiaonan Li, and Xipeng Qiu. TENER: adapting transformer encoder for named entity recognition. *arXiv preprint arXiv:1911.04474*, 2019.

[47] Xinyi Zhao, Masafumi Hamamoto, and Hiromasa Fujihara. Laboro BERT Japanese: Japanese BERT Pre-Trained With Web-Corpus. `https://github.com/laboroai/Laboro-BERT-Japanese`, 2020.

[48] Qile Zhu, Xiaolin Li, Ana Conesa, and Cécile Pereira. GRAM-CNN: a deep learning approach with local context for named entity recognition in biomedical text. *Bioinformatics*, 34(9):1547–1554, 2018.

Publications

Peer-Reviewed International Conference

- Yihong Liu and Yohei Seki. Joint Model Using Character and Word Embeddings for Detecting Internet Slang Words. Proceedings of the 23rd International Conference on Asia-Pacific Digital Libraries (ICADL 2021). pp.18–33. 2021-12. (full paper, acceptance rate 19%)

Book Chapter (the same paper described above)

- Yihong Liu and Yohei Seki. Joint Model Using Character and Word Embeddings for Detecting Internet Slang Words. In: Ke HR., Lee C.S., Sugiyama K. (eds) Towards Open and Trustworthy Digital Societies. ICADL 2021. Lecture Notes in Computer Science, vol 13133. Springer, Cham. https://doi.org/10.1007/978-3-030-91669-5_2

Domestic Conference

- Yihong Liu and Yohei Seki. Detection of Lexical Semantic Changes in Twitter Using Character and Word Embeddings. Proceedings of the 27th Nature Language Processing. pp.476-480, 2021-03