Universiteit Leiden

Master Computer Science

Automatic Named Entity Recognition
for ASR output in the Political Domain

Name:	Jiakun Sun
Student ID:	s2485524
Date:	July 14, 2021
Specialisation:	Artificial Intelligence
1st supervisor:	Suzan Verberne
2nd supervisor:	Marco Spruit

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

Named Entity Recognition (NER) is the basis of many tasks in the field of Natural Language Processing. The traditional machine-learning-based NER methods like CRF require a large amount of annotated training data, which is impracticable in practice, especially for applications in a brand new contextual environment. On the other hand, the meetings held by European parliaments are of great value to the research of EU politics. However, these meetings can cover a large range of topics with only audio/video records available, and therefore, the traditional NER methods are no longer suitable for such a tough scenario. To this end, we propose an applied framework for the NER applications in the EU political domain based on the AutoNER, which no longer requires precise pair-wise annotations for training, but only asks for the domain-specific dictionary instead, saving lots of effort from the tedious pre-processing tasks. Specifically, we first convert the audio records into text with Automatic Speech Recognition (ASR), and then build our own political-related core dictionary with specific entity categories via the mapping of external dictionaries and the full dictionary of high-frequency domain vocabulary; Finally, we adopt AutoNER [41] to complete the remaining job. Nevertheless, the text converting from ASR usually contains a large number of errors, obstructing the proper functioning of NER. In this work, we also study how would the noisy data affect the NER performance by comparisons of the AutoNER performance on the datasets with three different levels of noise.

Key Words: Named Entity Recognition, European Union Political Domain, AutoNER, Automatic Speech Recognition output, No manual annotations, Natural Language Processing

Acknowledgements

First of all, I would like to extend my highest gratitude to my first supervisor Suzan Verberne and my second supervisor Marco Spruit. In the process of doing the whole project, because there are very few materials for reference in the field of inquiry, and there is no high-quality data that can be used directly, in the early stage of constructing the data set and other stages, it was the supervisors who led the way and encourage me when I was in trouble. At the same time, I would also thank senior Hugo de Vos for his professional guidance on the political domain, which laid a solid foundation for my completion.

Secondly, I would like to thank my classmate Jingsen Chen, who helped me complete the human annotation process of the second person so that I can scientifically consider the results of my annotations during post-analysis.

Finally, I want to thank my parents, brother, and boyfriend for their support, so that I can devote myself to the exploration of the graduation project without any worries.

The COVID-19 pandemic disrupted the original rhythm of the world. A series of measures such as isolation, wearing masks, and keeping distance did not isolate the connection between people. This different study abroad experience will make me unforgettable for a lifetime.

Contents

1 Introduction

EU Parliament Committee Meetings are a few of the most important meetings in Europe, therefore, the speech and discussion records of the meetings are of great value to political researchers. However, due to the intense discussions and the variety of topics, records of these meetings are mainly reserved in video or audio formats only, which obstructs further analysis.

Named Entity Recognition (NER) is a technology that can identify certain entities from the corpus [29]. With techniques like Automatic Speech Recognition (ASR), audio records can be transformed into text seamlessly, after which the NER can be applied to extract key information for the subsequent analysis, e.g., political stance analysis and summarization. However, due to the noisy nature of the ASR output, achieving satisfactory results is still challenging.

Currently, the existing NER methods can be roughly classified into three categories, namely rule-based methods, traditional machine learning-based methods, and deep learning-based methods. The rule-based methods rely on the dictionary matching with manually pre-defined rules, such as LaSIE-II [17] and NetOwl [21]. These methods are easy for deployment without the requirement on annotations, however, the accuracy always remains an issue and usually leads to high migration costs.

On the other hand, machine learning-based NER methods are also proposed to save people from tedious yet ineffective rules generation tasks. Unsupervised machine learning NER methods, e.g., Clustering, may waive the prerequisite of annotations, but their applications are usually limited [29], therefore, the mainstream still focuses on supervised machine learning. Given features yielding from the word-level features or document and corpus level features, machine learning models like Hidden Markov Model (HMM), MaximumEntropy Model (ME), Conditional Random Fields (CRF) can be utilized to return much better results compared with rule-based and unsupervised machine learning-based methods [29].

Recently, deep learning-based NER methods like Transformers [25] have over-performed the conventional ones in both feature extraction depth and quality. Because of the high capacity enabled by the millions of learnable parameters in a deep learning model, the deep learning-based methods can learn complex feature representations, and no longer require the feature engineering tasks as traditional machine learning methods. In other words, deep learning-based models convey an end-to-end solution to the NER problem, cutting the pre-processing and post-processing efforts by a large margin with even better performance.

But as it is known to all, the supervised deep learning-based methods are data-hungry and usually need a large amount of labeled data to train the models, which is intractable for our case. To resolve the aforementioned problems, in this work, we propose the application of AutoNER [42], which only requires the domain-specific dictionary instead of the one-to-one data-label pairs to tune the models.

Concretely, we first convert the audio data into text with automatic speech recognition (ASR) with Punctuator [48], NLTK tokenization [4] and NeuSpell [18] for better noise filtering. Next, in order to provide the domain-specific dictionary needed by AutoNER, we build a political-related core dictionary with specific entity categories through the mapping of external dictionaries and the full dictionary of high-frequency domain vocabulary by crawling data from EU political-related websites as supplementary materials. Finally, we adopt AutoNER [41] to complete the remaining NER task.

However, the text data converting from the ASR usually contains a large amount of errors, and to further reveal how would the noisy input affect the performance of our method, we conduct experiments based on three datasets with increasing levels of noise: one is a relatively clean and manual annotated benchmark political dataset, and the second one is the unlabeled EU written data generated through the written record of the European Parliament debate. The third is the ASR output data of the audio record of the European Parliament debate that has been transformed by ASR.

To conclude, the main contributions of this work can be summarised as followings:

- Based on AutoNER, we propose a framework for effective NER application, which no longer requires precise word-wise annotations but only asks for the domain-specific dictionary, saving great efforts from data collection and preprocessing.
- We study the performance of AutoNER under the various quality of input data and propose guidelines for its further improvement.
- We compile three datasets with different degrees of noise in the European political domain and extract the corresponding domain-specific dictionaries, which can benefit other researchers or similar applications.
- We tried different combinations of preprocessing for EU political corpus with different noise levels, and also proposed a working way on the real-world corpus NER problem.

This paper will be organized as the following: background of NER tasks and AutoNER methods are discussed in Section 2. Section 3 introduces the processing framework and main methods of this scientific problem. The experiments are given in Section 4, with a special address on the influence of noisy datasets. Some discussion and future work based on the whole experimental results are shown in Section 5. Last but not the least, the conclusion of this works is given in Section 6.

2 Background

In this chapter, we will introduce in details NER tasks, mainstream solutions to the NER tasks, and their applications in the Political Domain.

2.1 Named Entity Recognition

Sequence Labeling is an important technique applied in many Natural Language Processing (NLP) tasks, such as Speech Tagging, Named Entity Recognition and Relationship Extraction, etc. Generally speaking, sequence labeling refers to the process of marking each element in a sequence, with a special focus on the learning of relationships among sequence elements. Specifically, it is to combine the relationship between the tokens in the sequence to make each token in the given sequence corresponding a label [10].

Named entity recognition(NER) belongs to the sequence labeling problem, it identifies various pre-defined entity categories from a text sequence [29], such as person names, place names, and organization names, etc. On the other hand, NER tasks are also crucial to many other NLP tasks, including but not limited to relation extraction, question answering systems, etc. And the important concepts of NER will be given in the following sub-sections.

2.1.1 Tagging

Given the aforementioned NER is also a process of tagging tokens. The commonly used tagging schemas in NER tasks are **IO**, **BIO** and **BIOES** [34]. The main meanings and differences of these three are introduced as follows:

- IO: **I** means **I**nside the entity while **O** means **O**utside the entity, however, this schema will lead to mistakes when two entities are neighbouring.
- BIO: **B, I, O** represents the **B**eginning of the entity, the **I**nside of the entity, and the **O**utside of the entity, respectively.
- BIOES: The letter **B, I** and **O** share the same definition as the BIO schema, and in BIOES schema, **E** stands for **E**nd of entity and **S** stands for the **S**ingle word entity.

Among them, the IO schema is the simplest. It can clearly distinguish between entities and non-entities, but it cannot clearly distinguish the boundaries of entities. The BIO schema can solve this problem. However, BIOES schema provides additional information about **E**nd and **S**-tags for a single term, which provides more information and therefore more accurate, but as it has more tags to make predictions (more **E** and **S**), its efficiency may be affected [36].

2.1.2 Metrics

The performance of NER systems is usually evaluated with **Precision**, **Recall** and **F1-score** metrics, which are calculated by comparisons between the entities and outside entities that NER systems identified with the human annotations [25]. To reveal the calculation of Precision, Recall, and F1-score, we first introduce several important concepts as follows.

For classification problems, the confusion matrix is usually adopted to measure the performance of a classifier, as shown in Table 9 [11].

Table 1: Confusion Matrix

Confusion Matrix		**True Label**	
		Positive	**Negative**
Predict Label	**Positive**	TP	FP
	Negative	FN	TN

- **True Positive(TP)**: Entities correctly identified and labeled by NER.
- **False Positive(FP)**: Entities wrongly identified or labeled by NER.
- **False Negative(FN)**:Entities wrongly missed by NER.
- **True Negative(TN)**: Non-entities correctly identified by NER.

Hence, the Precision, Recall and F1-score can be calculated [25]:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

As shown above, F1-score is a harmonic average of Precision and Recall and it is more commonly used as the evaluation standard. However, since most of the NER tasks distinguish more than one entity type, how to fairly evaluate the overall performance for all entity categories becomes an issue. Therefore, apart from the F1-sore, the following two metrics are also widely adopted [25]:

- **Macro-averaged F1-score**: Calculate F1-score for each entity type separately, and then calculate the average F1-score value, that is, treat each type as the same weight.
- **Micro-averaged F-score**: Treat each entity as equal, without distinguishing entity types, and calculate the overall F1-score.

2.2 Mainstream NER methods

In general, the development of the NER method has gone through 4 stages, as shown in Figure 1, from the early matching based on some rules and dictionaries to the method based on traditional machine learning, and then to the application of deep learning to the NER task to achieve an end-to-end solution. Nowadays, Attention, Transfer Learning and Semi-supervised Learning are the mainstream research directions [25].

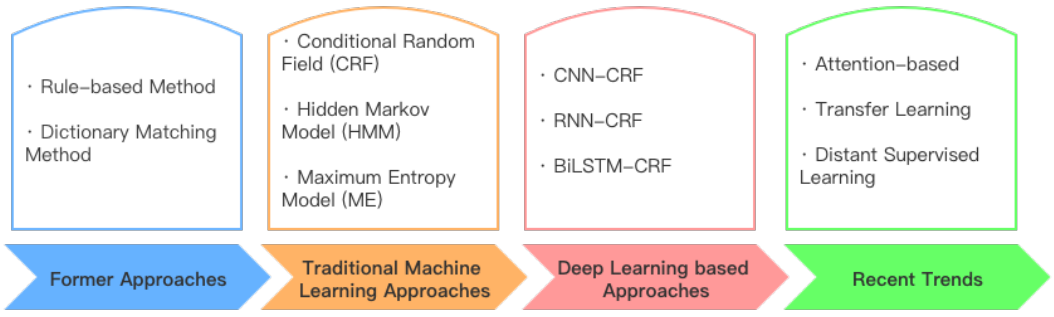

Figure 1: The develop trends of NER approaches

2.2.1 Dictionary and Rule-based method

The early method is usually to build the NER system through dictionary matching, manual rule-making, etc. To be specific, the rules mean that manually specify what words/categories meet what criteria. Well-known ones such as NetOwl [21] of ISO-Quest, LaSIE-II [17] of the University of Sheffield, are the rule-based NER solutions constructed by domain-specific dictionaries, syntactic vocabulary templates, and regular expressions, etc. These rules might be This kind of method usually does not need to annotate data, however, they are domain-specific that cumbersome to make rules and maintain, and with high migration costs.

2.2.2 Traditional Machine Learning methods

For the Machine Learning methods, some Unsupervised Learning algorithms such as Cluster can also be used to solve this problem where no annotation data is required, but the accuracy of this kind of method is generally limited [29]. Some Supervised Machine Learning methods can achieve better results than the unsupervised learning ones, which rely on the quality of feature engineering. These feature-based approaches need well-annotated data which is typically combined with well-designed features. Commonly used models such as Hidden Markov Model (HMM), Maximum Entropy (ME), Conditional Random Fields (CRF), etc. Commonly used features include word-level features (capitalization, word morphology, part of speech), document and corpus level features, etc.

In this paragraph, the most commonly used traditional approach **CRF** will be introduced in detail. Conditional Random Field (CRF) was first introduced in 2001 by Laffery et al. [22]. As we mentioned in the Section 2.1, we can think of NER as a sequence labelling problem, receiving an input sequence (Observations) such as $X = (X_1, X_2, ..., X_n)$, and outputting the target sequence (States) as $Y = (Y_1, Y_2, ..., Y_n)$, hence, the architecture of Linear-CRF shows in Figure 2.

Suppose $P(Y|X)$ is a Linear-CRF, then under the condition that the random variables X and Y are x and y respectively, the conditional probability is [45]:

$$P(y|x) = \frac{1}{Z(x)} \exp(\sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, x, i) + \sum_{i,l} u_l s_l(y_i, x, i)) \tag{1}$$

where

$$Z(x) = \sum_{y} \exp(\sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, x, i) + \sum_{i,l} u_l s_l(y_i, x, i))$$

t_k, s_l: Characteristic Function
λ_k, u_l: Corresponding Weights

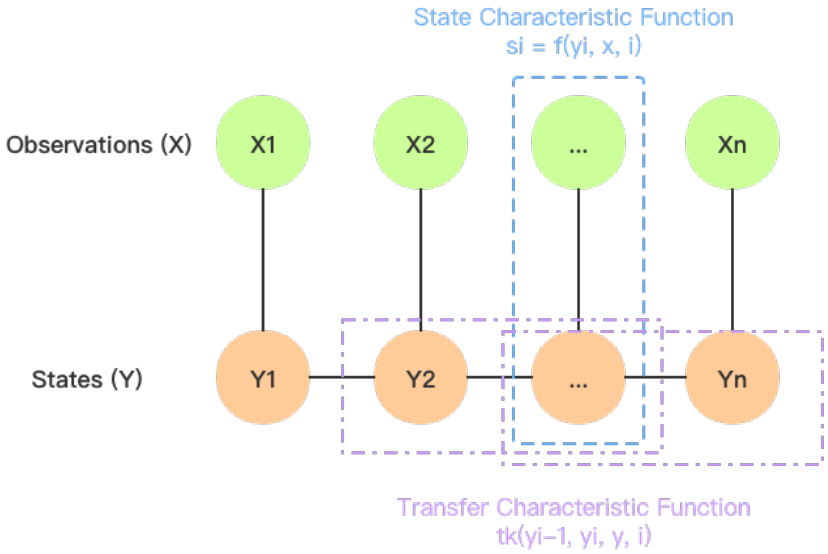

Figure 2: The architecture of Linear-CRF

$Z(x)$: Normalization Factor

To simplify, the transition features and state features and their weights are represented by uniform symbols. Hence, the simplified formula of the conditional random field is as follows [45]:

$$P(y|x) = \frac{1}{Z(x)} \exp \sum_{k=1}^{K} w_k f_k(y, x) \tag{2}$$

where

$$Z(x) = \sum_{y} \exp(\sum_{k=1}^{K} w_k f_k(y, x))$$

Therefore, the CRF-based named entity recognition process is mainly divided into 3 steps:

1. Tokenize the sentence and label the entity type for each token.
2. Determine the feature template, generally use the features of n-gram before and after.
3. Training the CRF model for the parameter w_k in Equation (2).

Here, using the BIO schema mentioned in section 2.1.1 as an example, the structure diagram of CRF as Named Entity Recognition is shown as Figure 3.

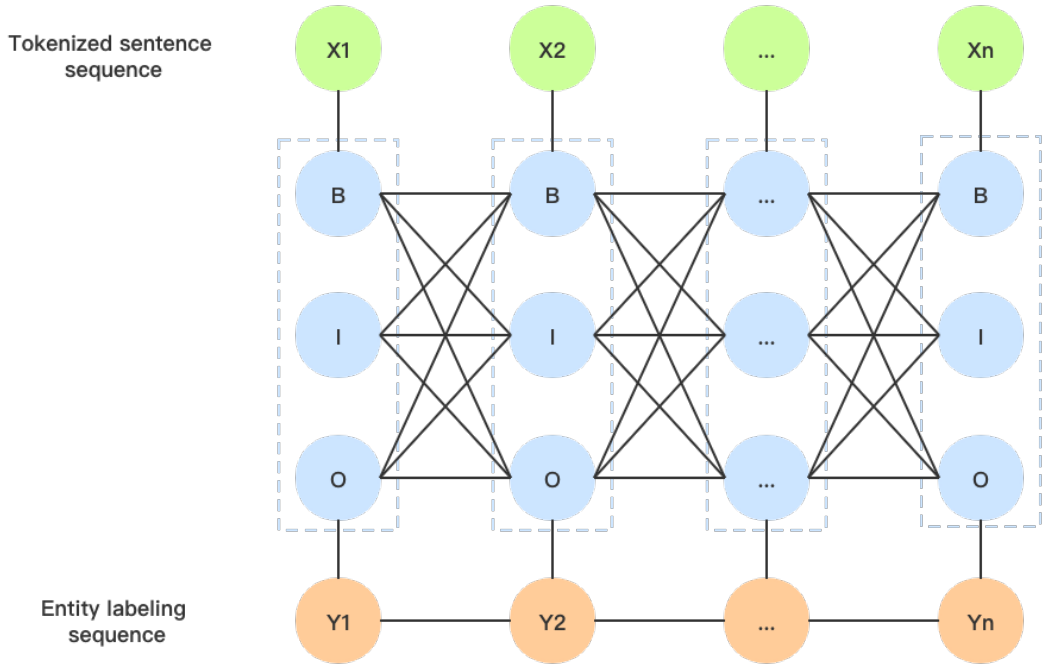

Figure 3: The structure of CRF for Named Entity Recognition

2.2.3 Deep Learning based methods

Another state-of-art NER approaches are Deep Learning-based. Thanks to the highly non-linear nature of deep learning, it is conducive to the learning of complex features. At the same time, deep learning can automatically learn features that are beneficial to the model. The complicated feature engineering required by traditional machine learning methods can be simplified. Besides, deep learning can help us build models end-to-end. Therefore, deep learning models combined with some rules are currently the more commonly used NER methods.

One early-stage representation of DL-based NER is that Collobert et al. [8] propose two network structures to perform NER: window-based and sentence-based. To be specific, it will only use the words around within a definite window to assign the label in the window-based approach. While sentence one will consider the whole sentence with the help of the convolutional layer. Since then, drawing on the same idea, a series of NER work using the RNN structure combined with the CRF layer has proposed. One of the most used methods is BiLSTM-CRF [16].

The NN/CNN/RNN-CRF model, which combines the neural network with the CRF

model, has become the mainstream model in the Neural Network field of NER. The NER method based on the neural network structure inherits the advantages of the Deep Learning method and does not require a large number of handcraft features. Only word vectors and character vectors can reach the mainstream level, and adding high-quality gazetteer features can further enhance the effect. However, as we mentioned before, both Deep Learning and Machine Learning methods usually require large amounts of manually annotated training data, especially Deep Learning based methods which need a large amount of training data, how to relieve the workload of human annotation is also a problem [25].

2.2.4 Recent Trends

Introducing Attention is one of the trends of recent years in the NER field. Rei et al. [35] is still based on the structure of the RNN-CRF model, focusing on improving the splicing of word vectors and character vectors. Using the attention mechanism to improve the original character vector and word vector splicing, to sum up, the weights, two layers of traditional neural network hidden layers are used to learn the weight of the attention so that the model can dynamically use the word vector and character vector information. The experimental results show that it is better than the original splicing method. Bharadwaj et al. [3] use the original BiLSTM-CRF model, phonological features are added, and the attention mechanism is used on the character vector to learn to pay attention to more effective characters.

For Deep Learning methods, a large amount of annotated data is generally required, but there is not a large amount of annotated data in some fields. Therefore, how to use a small amount of labeled data for NER in the neural network structure method is also the focus of recent research. These include Transfer Learning and semi-supervised learning [25]. Among them, the most famous application of Transfer Learning is BERT (Bidirectional Encoder Representations from Transformers) [19], which is widely used in the field of Nature Language Processing not limited to NER. At the same time, Sarhan et al. surveys the transfer from open information extraction (OIE) to other NLP tasks and realize the cross-domain learning, which can successfully address the lack of annotations to some extend [40].

At the same time, some computational linguistic approaches represented by Spacy [15] are also commonly used. With the help of this framework, the entity ruler can be customized and assembled into Spacy to make the NER research and comparison process more convenient.

2.3 NER in the Political Domain

The task of Named Entity Recognition in a specific field usually has certain challenges. The reason is that the mainstream knowledge-based methods may require some detailed feature engineering defined by some domain experts [47]. At the same time, the political domain still has limited data and a lack of gold annotated data. Leitner et al. [23] generate data through manual annotation, then, by applying CRF and Bi-LSTM to coarse-grained annotated and fine-grained annotated data, different NER results are obtained, and a German legal domain NER system was constructed. Besides, to solve the aforementioned limited data problem, Leitner et al. [24] also constructed a German political field dataset. Revenko et al. [37] is working on how to automatically convert coarse-grained entity classification to political domain fine-grained entity classification. Similarly, it also focuses on German legal files and builds sufficient annotated datasets through manual annotations. Cardellino et al. [6] propose a system that can identify entities in political text and automatically link them with LKIF ontology and so on. Like in the previously mentioned papers, a lot of manual annotation works are combined in the construction of the data set. As we mentioned earlier, political domain NER problem research is not common, and studies on pure English are also very rare. Therefore, we refer to other languages which also solve the NER problem in the political domain. Although the grammar and system of each language are different, the entity types concerned on the political corpus have a high degree of overlap, so it is beneficial as well.

2.4 NER for ASR output

Most of the NER tasks for ASR output are doing ASR to convert audio into text data and then doing the NER process. Ghannay et al. proposed a novel neural network framework that can directly extract the entities from the speech data [12]. Zhai et al. uses the ASR output data generated from a Chinese news corpus through the BBN LVCSR system and then use the maximum entropy model to perform the corresponding NER task. At the same time, they used a re-segmenting method to improve the performance of NER on this high-noise corpus [53]. Hatmi et al. use the labeled pronunciation dictionary to introduce the NER process into the ASR system to directly generate translated text data labeled with entities [13]. These methods all introduce manually annotated data in the training process. The difference is that Zhai et al. use the manually annotated dataset by Beijing University to train the NER model and apply it to the text data and ASR data respectively for comparison. While Ghannay et al. and Hatmi et al. are combined the NER process into the ASR framework, also trained by the existing manual annotated data. In this article, we choose to use the ASR method to convert the record into the transcript and then use the AutoNER

method to complete the task of named entity recognition. Unlike the previous work, AutoNER does not require a large amount of manually annotated training data. It only needs a domain-specific dictionary to automatically complete the NER process [41].

2.5 AutoNER

The problem with the latest methods and research trends mentioned above in the EU political domain is that there is not enough corpus, especially not enough annotation data. At the same time, there are fewer applications for such problems. In this research, we use ASR to construct a dataset from audio records and use a fully automatic NER method to solve the above-mentioned problems.

As we know, supervised learning has obtained high-quality results in the NER tasks, but in the absence of manually annotated data, it is difficult to apply in the industrial community. A common practice in the industry is to use some distant supervision NER methods, however, the results might be not clean and difficult for the following using. AutoNER accepted the noisy annotated data generated by distant supervision and proposed a new NN model combing with a novel tagging schema to address this kind of insufficient annotated data problems successfully [42].

To be specific, AutoNER deals with the problem mentioned before with the idea of distant-supervision, the domain dictionary is used to generate the tag data. But for this kind of method, two main problems exist:

1. The entities in the dictionary are limited, there might be some entities that can not be recognized beyond the dictionary.
2. The same entity might belong to several categories, but it can only be assigned to one label according to the dictionary matching algorithm.

To address these problems, the author introduced the **Full dictionary** and **Tie or Break**.

Full dictionary To solve the above-mentioned problem that entities outside the dictionary will not be marked by distant supervision, the author introduced a full dictionary in AutoNER. Specifically, the author divided the domain-specific dictionary into two parts. The core dictionary is similar to the entity dictionary we commonly use, marking the entity token and its specific classification. The full dictionary is a part of supplementary information composed of high-frequency domain-specific vocabulary.

These words will be labeled as 'unknown' in the matching process. This allows the model to automatically pay more attention to these high-frequency vocabularies that may become entities during the learning process, and at the same time, it is not necessary to accurately distinguish their specific labels.

Tie or Break Unlike the commonly used 'BIO' and 'BIOES' schema, which will annotate the token itself. The 'Tie or Break' annotation schema is more concerned with whether the current token and the previous token are in the same entity: If they are inside the same entity, it's labeled *Tie*; If the neighbor words have at least one in the full dictionary then assign it as *Unknown*, mark it as *Unknown*; If they are not in one entity then mark it as *Break*. An example of the 'Tie or Break' schema can be found in Fig 4. And the optimization process is separating entity identification from entity type determination. As described in the original paper, the entity identification was done first, and the two breaks were used as a span, and then the entity type was determined. Then, in entity recognition, for the output of the category between the current word and the previous word, a binary loss is made for *Tie* and *Break*. If the category is an *Unknown* category, the loss is skipped directly and no loss is calculated. The advantage of this annotation schema is that even if the boundary of the entity is incorrectly recognized by the model, most of the labels in the middle part of it are still correct, which can be more universal.

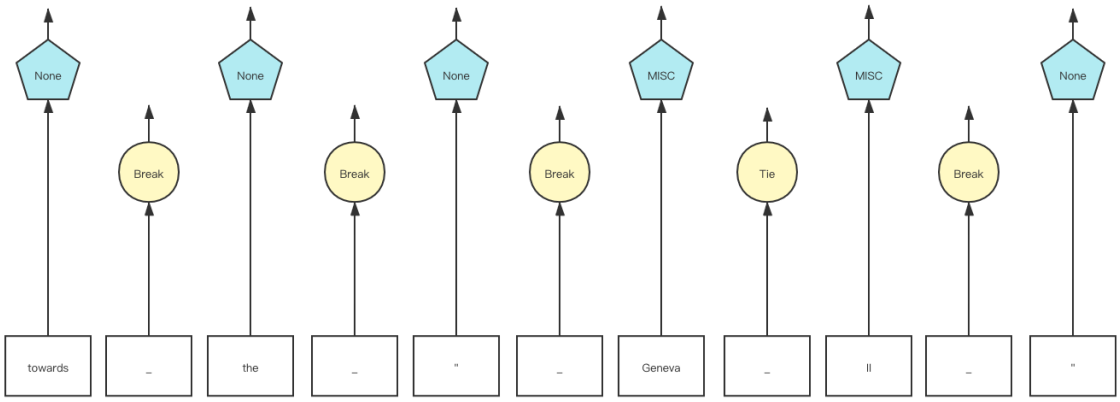

Figure 4: The 'Tie or Break' schema of AutoNER

After introducing some prior knowledge. We can look into the whole working framework of AutoNER. AutoNER can simply seem like a dictionary matching combined with a neural network classification, as shown in Fig 5. The training inputs are only the raw data and the domain-specific dictionary. Then it will use the distant supervision, specifically, the dictionary matching method to generate some raw annotated

data. Then it will use the neural network to learn from such noisy annotation data. After training, we can just input the raw test data into the trained model to get the annotated result. It needs to know that to deal with the multiple label problem we mentioned, it will annotate all the possible labels of each token. Hence, the final CE loss function was carried out with the soft labels, and no hard label was used.

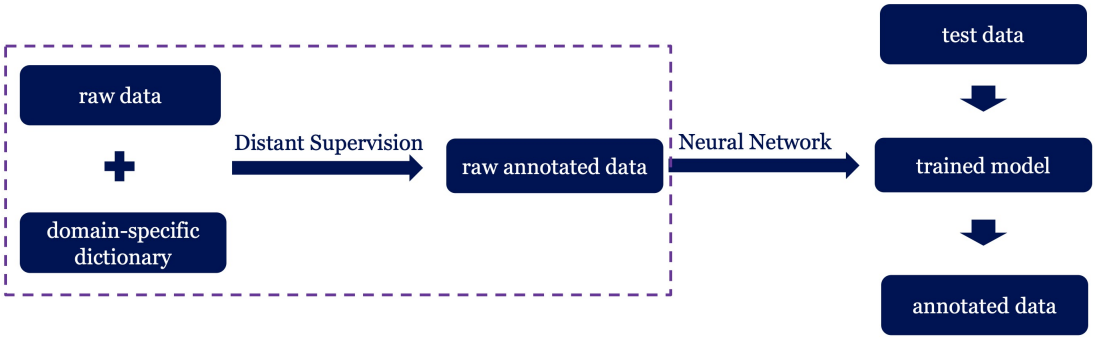

Figure 5: The main framework of AutoNER

3 Methodology

In this section, we mainly introduce the specific operations of data acquisition, data preprocessing, construction of domain-specific dictionaries, and how to evaluate the results. We mainly use two methods to solve the task of named entity recognition. The traditional CRF method needs to be provided with labeled data as training and test data, and NER is performed by token and context features; while for the AutoNER method, no labeled training data is needed. But we need to provide domain-specific dictionaries, which include a well-categorized entity dictionary that plays the main matching role and high-frequency vocabulary of the field corpus that does not require entity categories as supplementary dictionaries. At the same time, because of the need to evaluate AutoNER's named entity recognition. The test data also needs to be annotated. And the whole method framework of this paper is shown in Figure 6.

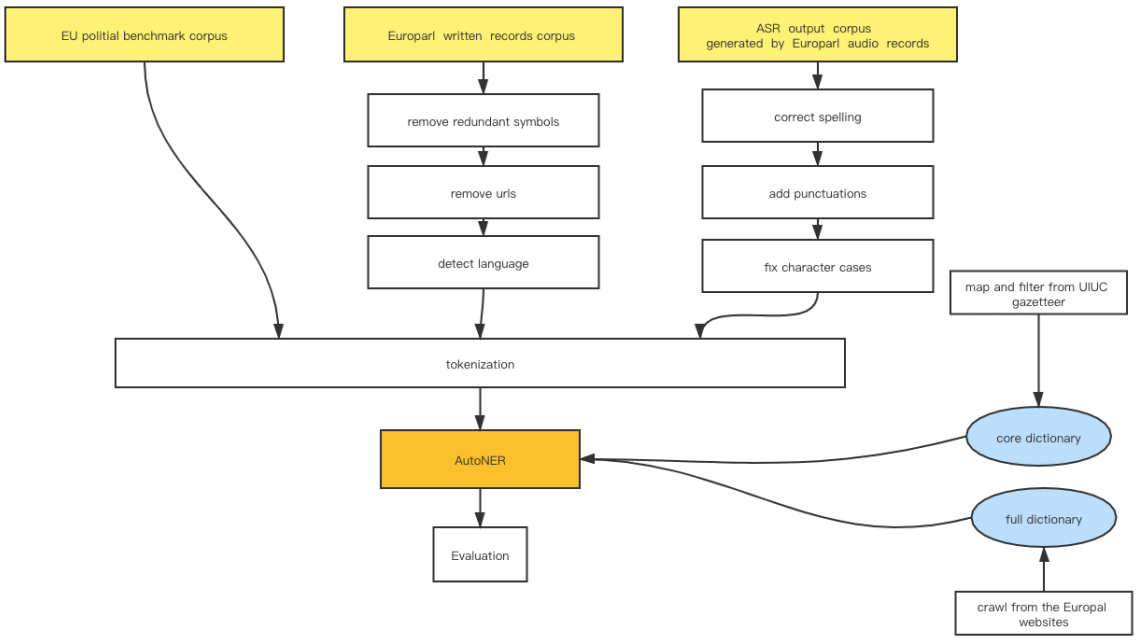

Figure 6: The whole method framework

3.1 Datasets Building

As the European Political domain is with a limited high-quality dataset for NER, we establish 3 datasets in this domain with different levels of noise to explore the per-

formance of AutoNER and CRF on such real-life problems. The specific construction method will be introduced in the following subsections.

3.1.1 Benchmark dataset

For the benchmark data experiment, we used publicly available data, which was obtained from the content of political speeches in the UN General Assembly from 1993 to 2016[1]. The data has been manually labeled by the author Leslie Huang, the test set and the training set are split, but articles related to this data set have not been published. The entities are annotated by 4 entity categories, as shown in Table 2, which is the same as CoNLL-2003 [39].

Table 2: The entity types of CoNLL-2003

Entity Type	**Explaniation**
I-LOC	Locations
I-PER	Persons
I-ORG	Organizations
I-MISC	Miscellaneous

Since the original data is the data of the UN General Assembly, and our main research domain is the EU political domain, we first need to filter out the text of EU countries to get a new dataset. To be specific, we first get the list of EU members[2]. Since the data is divided into the training set and test set, it contains 53 and 23 files named by 'IDxxx_year_country.txt' respectively. Therefore, we only need to judge whether the country in the file name is in the list of EU member states to divide all the corpus into EU and non-EU parts. The filtered EU-related data has a total of 716 sentences, 15,012 tokens, while not EU-related ones are 4,944 sentences and 134,834 tokens. The number of entities contained in the dataset is shown in Table 3.

Table 3: The details of original benchmark dataset

	Sentences	**Tokens**	**LOC**	**PER**	**ORG**	**MISC**
EU Counts	716	#15,012	286	25	257	141
Not-EU Counts	4,944	#134,834	2,391	199	1,446	1,203

[1] https://github.com/leslie-huang/UN-named-entity-recognition

[2] https://europa.eu/european-union/about-eu/countries_en/tab-0-0

Due to the limited amount of data, in order to prevent possible over-fitting problems, we will expand the data set. To be specific, we split the filtered data into 3 equal parts randomly, the training set, the development set, and the test set. In this way is because the counts of not EU sentence is very rare, and we want not only test set and validation set has the EU sentences. It should also be added into the training corpus, so we divided it into 3 parts. This division is based on the subjective reasons for the lacking EU political data. Then we add non-EU-related data to the training set to achieve data set expansion. Because the problem we need to study is for the EU political domain, therefore, whether our NER method works or not needs to be measured on the EU test set. In the training process, we can use similar political domain or EU-related corpus to expand the training data to a certain extent (we use political domain corpus expansion here), while the test data only use EU political domain to verify whether it can be a good solution for our specific problem. Table 4 shows the final counts of our training, development, and test set. After our statistics, we found the entities are not equally distributed over the 5,183 sentences, there are 2,304 sentences without entities in the training set.

Table 4: The details of the splitted benchmark dataset

	Sentences	**Tokens**	**LOC**	**PER**	**ORG**	**MISC**
Training set	5,183	#140,612	2,498	204	1,529	1,243
Development set	239	#5,789	91	8	78	51
Test set	239	#6,079	86	12	94	49

It is worth mentioning that since the original dataset is only annotated according to the IO schema, automatic conversion is required, supplemented by a manual check, to convert it into the result of BIO tagging. We adopted the conversion from BIO schema to BIOES schema in NCRF++ [52] to get the final BIOES tagging results.

For traditional supervised machine learning methods such as CRF, the training set and test set are in the similar format, as in the example in Table 5, each row has two columns of data, separated by Tab, the first column is the token, the second column is the BIO token label, each row is a token, with a blank line between different sentences. The only difference is that there is a Part-of-Speech (POS) tag annotation between the token and entity type of the training data as shown on the left. The POS tags are obtained with NLTK [4].

However, for the AutoNER method, since the training data does not need to be annotated, for this part, the BIO label is directly removed. Similarly, each line is

Table 5: The data format example of CRF method

Training			**Test**	
Denmark	NN	B-LOC	Denmark	B-LOC
continues	VBZ	O	continues	O
to	TO	O	to	O
strongly	RB	O	strongly	O
support	NN	O	support	O
the	DT	O	the	O
efforts	NNS	O	efforts	O
towards	NNS	O	towards	O
the	DT	O	the	O
“	NN	O	“	O
Geneva	NN	B-MISC	Geneva	B-MISC
II	NN	E-MISC	II	E-MISC
”	NN	O	”	O
conference	NN	O	conference	O
.	.	O	.	O

a token, and there is a blank line between different sentences. At the same time, AutoNER's test data has a slightly different format, and the start and end of each line are marked with '<s>' and '<eof>' respectively. The conversion for BIO tagging and Break or Tie tagging is based on the corresponding method mentioned in 2.5 above, which is converted to generate data as shown in the example on the right below. The 'Break' is represented by 'I' and 'Tie' is 'O'. Similarly, each line is a token, its tag, and its entity type, separated by spaces, and blank lines indicate different sentences. And the sentence example of the training and test set of the AutoNER method can be found below in Table 6.

3.1.2 Europarl written records dataset

The EU written data consists of the transcripts of the European Parliament Plenary debates from 1999 to 2014 [14, 49]. This part of the data is noisy and required substantial pre-processing. To be specific, there are plenty of redundant symbols, the URLs of record, the record of report time, and speaking person. For example:

<http://purl.org/linkedpolitics/eu/plenary/1999-07-20_Speech_19>
lpv:text "<0xa0><0xa0> –(PT) Thank you very much, Madam President. Please allow me to use my first words to congratulate you. You have conducted

Table 6: The data format example of AutoNER method

Training	**Test**		
	<s>	O	None
Denmark	Denmark	I	LOC
continues	continues	I	None
to	to	I	None
strongly	strongly	I	None
support	support	I	None
the	the	I	None
efforts	efforts	I	None
towards	towards	I	None
the	the	I	None
“	“	I	None
Geneva	Geneva	I	MISC
II	II	O	MISC
”	”	I	None
conference	conference	I	None
.	.	I	None
	<eof>	I	None

a fine campaign, you have acted with great dignity and have always behaved impeccably in all the dealings that I have had with you. I would therefore like to congratulate you most sincerely and to wish you an excellent Presidency, which I am sure it will be. I would also like to thank, if you donẗ mind, everyone who voted for me. Through my nomination, we have unleashed a new dynamism, and I would like to thank not only those belonging to my own group, but also those from other groups who voted for me too.(Applause)"@en ;

lpv:unclassifiedMetadata ""@en ,

"."@en ,

"Soares (PSE)"@en .

We can clearly find that there are some URLs, symbols, classification information, etc. that we don't need to build the NER dataset. Therefore, we use the regular expression matching method to filter the text that is meaningless for this task. At the same time, there are some non-English data in this part of the data, so we also need to use langdetect [30] for language recognition, based on our target needs, filter out the non-English data. So that we can transform it into relatively clean data with only the specific content of the debate, and save one line for each sentence. Similarly, in

order to use the AutoNER method, it is necessary to tokenize the generated sentence data, for which we use NLTK's tokenization method [4]. The details of the processed Europarl written records dataset are shown in Table 7.

Table 7: The details of the Europarl written records dataset

	Sentences	**Tokens**
Europarl written records dataset	2,198,573	#61,181,361

3.1.3 ASR output dataset generated by Europarl audio records

The ASR output dataset is obtained from the audio records of the European Parliament meetings[3] from July 2014 to June 2016. Then, text data is generated through ASR technology, Wav2Vec2[4] [2].

For the need of ASR to process data, the audio is divided into 15-30 seconds of small fragments, sorted by time, and the final text data is also in the same format as small pieces. It is worth noting that this part of the data is all uppercase, and there are no punctuation marks. And one example of the output utterances is:

SO A LEX DOID BY ACCLAMATION CONGRATULATION STERMORISE I DECLARE UNTE CANDIDATE ELACTED BY ACCLAMATION AND ASKED ATHE NEW CHAIR TO TAKE SID

To process the data, first, we splice the utterances generated by several pieces of the same day and convert them to lowercase. Then, use Punctuator [48] to automatically punctuate each paragraph of text. Here, we use the pre-trained model they provide. Through processing, we get the text data marked with punctuation. We split the sentence of the text according to the punctuation and capitalized the first letter of each sentence. Finally, we apply the NeuSpell [18] automatic spelling correction tool on them. Hence, we get an example of the utterances after preprocessing shown as:

Before applying Punctuator:
so a lex doid by acclamation congratulation stermorise i declare unte candidate elacted by acclamation and asked athe new chair to take sid

[3] `https://www.europarl.europa.eu/committees/en/libe/meetings/webstreaming`
[4] `https://huggingface.co/transformers/model_doc/wav2vec2.html`

After using Punctuator, it is split into two sentences, like:
..., a lex doid by acclamation, congratulation stermorise. I declare unte candidate elacted by acclamation and asked athe new chair to take sidthank.

After applying NeuSpell:
. . . , a lex did by acclamation , congratulations stermorise . I declare until candidate elected by acclamation and asked the new chair to take sidthank .

At the same time, because people may also have repetitions when speaking, ASR has a certain error rate when processing, even though we applied the automatic spelling corrections. Besides, we will also have some classification errors when generating punctuation. Therefore, we found that some of the generated sentences are too long or too short. In response to this situation, we counted the composition of sentences after splitting into sentences and filtered the sentences shorter than 5 words or longer than 30 words as the '5-30 words sentences' dataset. And the overall data situation is displayed in Table 8. Then we applied the NLTK tokenization [4] to generate the tokenized data for the requirement of AutoNER.

Table 8: The details of the ASR output dataset

	All sentences	**Number of words in one sentence**		
		<5 words	5-30 words	>30 words
Counts	137,052	11,970	77,497	47,585

Word Embedding The word embedding is a matrix of numerical representation of words in a multi-dimensional space. AutoNER uses word embedding to encode the words into real numbers for the following processes. The GloVe[5] pre-trained word embedding was adopted, it trained from Wikipedia 2014 + Gigaword 5, and the 200-dimension ones is chosen [33].

Stopwords The frequency of stop words in English is very high, such as words 'a, an, and, or, of, at, the, etc'. The substantive information of these words is extremely limited, and their appearance will affect sentence length and weight distribution. Therefore, removing stop words is necessary to improve the effect of text processing [43] in the dictionary matching step. Actually, an interface is also provided in AutoNER. By specifying a stop word list, stop words can be automatically removed during an operation in the initial dictionary matching process. Then when adding the

[5] `https://nlp.stanford.edu/projects/glove/`

matching results into the tokens to generate a training set, it uses the raw text data without stopwords filtering for the reason the sequence labeling needs contextual information. Additionally, it is also a problem of studying English corpus, we use the same stopwords provided by AutoNER [42], a total of 57 commonly used stopword lists.

3.2 Domain Specific Dictionary

For the AutoNER method, whether it can automatically mark entities and achieve acceptable results, the most important thing is the domain-specific dictionary. There are two dictionaries mainly used. One is called **Core Dictionary**, which records some known entities in the field and plays the main matching role; the other is a **Full Dictionary** composed of high-frequency vocabulary in the field. To supplement phrases that may be entities but unknown types, to alleviate the situation that a large number of entities may be lost if the dictionary is not comprehensive only by dictionary matching. Below, in this section, we will describe in detail the process of how to build two dictionaries.

3.2.1 Core Dictionary

When building the core dictionary, we followed the entity category mentions before as shown in Table 2, we generated the corresponding entities for each type within the EU political field to form a domain-specific dictionary. Then the processing of LOC, PER, ORG, MISC into the 4 types of EU political domain knowledge will be introduced separately in the following paragraph.

LOC The main consideration for Location is the country members of EU[6] to fill in the location information in the dictionary.

PER We added the previous Members of the European Parliament to the Person category to implement political domain knowledge extension. The data is obtained through Wikidata [51], the date as of January 2021.

ORG The organization mainly considers EU's Political Parties[7] and EU Agencies[8]. Obtain these data through Web Crawler and Wikidata query [51] which is updated

[6] https://europa.eu/european-union/about-eu/countries_en
[7] https://europa.eu/european-union/about-eu/institutions-bodies_en
[8] https://europa.eu/european-union/contact/institutions-bodies_en

until January 2021. At the same time, because some organizations have short names[9], the corresponding short name also defined as a new entity entry together with the full name.

MISC It is worth noting that to make the dictionary applicable to the political domain, pieces of legislation are also included as part of the entity and marked as the MISCELLANEOUS category. The pieces of legislation information come from the EUR-Lex website[10], it contains a total of 32 categories of topics, we use the summary in each small topic in each category as pieces of legislation. For example, *Agricultural and food supply chain — unfair business-to-business trading practices* and *Future of the common agricultural policy (CAP)* are taken out as entities from the topic *Agriculture/Common Agricultural Policy (CAP)*[11]. At the same time, we noticed that there are some years, short names, etc. in the summary. For these, we have done the following:

1. Identify the parentheses. If there is a short name in the parentheses, use the short name and full name as two entities respectively.

2. Identify the year, remove the year and the summary with the year as two entities respectively.

After the above processing and supplemented by manual check, we have realized the legal domain knowledge of adding the political domain to the MISC category of the dictionary.

Moreover, to supply a more sufficient dictionary, we also adopted an external Gazetteer to extend our dictionary. To be specific, it is based on the data publicly available in the Gazetteer provided by UIUC NER system [9], which contains 79 fine-grained entity types, about 1.5 million entities in total. In order to map this fine-grained classification to the 4 entity types of CONLL-2003 [39], as shown in Table 2, we mainly use the mapping rules in Liu et al. [27] combined with the annotation guideline[12] of CONLL-2003, and the specific mapping rules are displayed in Table 9. After mapping and tailoring, about 1.3 Million entities are obtained.

[9] `https://en.wikipedia.org/wiki/European_political_party`
[10] `https://eur-lex.europa.eu/browse/summaries.html`
[11] `https://eur-lex.europa.eu/summary/chapter/0306.html`
[12] `https://www.clips.uantwerpen.be/conll2003/ner/annotation.txt`

Table 9: The mapping rules between Gazetteer and Core Dictionary types (* Means any subsequent characters)

Target	Gazetteer tag	Num
LOC	Locations* Parks Paths	166,546
PER	People*	873,849
ORG	Organizations* PoliticalParities* Government*	243,243
MISC	Languages Nationalities CriminalActs Films TV.Programs Weapons*	90,525

3.2.2 Full Dictionary

For the full dictionary, as we mentioned in Section 2.5 before, the domain-specific dictionary is supplemented with domain high-frequency vocabulary. We use the data mining method of AutoPhrase [26, 41], also used in the original AutoNER paper, to obtain this high-frequency vocabulary list. For the text data of the political domain, we chose the written data introduced in section 3.1.

After all the above processing, cleaning, and deduplicating the generated dictionary, we finally get the core dictionary and full dictionary details as shown in Table 10.

Table 10: The entity counts of Core Dictionary and Full Dictionary

	Category	Num
Core Dictionary	LOC	166,596
	PER	877,665
	ORG	243,437
	MISC	97,064
Full Dictionary		42,112

3.3 Evaluation

Since we use written data and ASR output data are raw data sets that are not labeled, the AutoNER method is used to solve the problem of entity recognition without human annotation. However, whether the results of this part of the data are acceptable, still needs to be evaluated. For this, we mainly thought of two methods. First, use the method of manually labeling part of the data, and then use this part of the manually labeled data as a test set to detect the effect of the model. Second, through the posterior analysis of the annotated text, the effect of AutoNER on solving a large number of named entity recognition problems without annotated data is detected from an intuitive perspective.

For manual labeling, we randomly select 200 sentences as the test set from each dataset. Then, two annotators will make an annotation separately to achieve the relative credibility of the result. All of our experiments are based on the labeling results of one of the annotators, and the annotating of the same data set by the other annotator will be used to test the credibility of the first annotator's work. The guidelines for manual annotation are formulated by referring to the mapping rules of the external source dictionary introduced in section 3.2 and the annotation guideline of CONLL-2003[13]. The tool we use for manual annotation is doccano [31].

[13] https://www.clips.uantwerpen.be/conll2003/ner/annotation.txt

4 Experiments and Evaluation

In this chapter, we introduce specific experiments and their results. We mainly introduced three kinds of datasets in Section 3, and conduct experiments respectively. Among them, for the benchmark dataset with gold annotation, we used the traditional supervised learning method CRF and AutoNER through corresponding preprocessing and compared the performance of the two methods on the same NER problem. For the Europarl written reEuroparl written records dataset composed of the EU Parliament's debate transcripts, we only use AutoNER, which does not require a large amount of manual annotation in advance, but evaluates the effect of NER by the randomly sampled 200 sentences manual annotated data. Then for the ASR output data experiment, we perform certain preprocessing on the text data generated by the European Parliament video processed by Wav2Vec2 [2], and only use AutoNER, which does not require a large amount of manual annotation training corpus, for named entity recognition and explore its performance on noise data.

4.1 Benchmark Experiments

For the benchmark dataset, as we mentioned in Section 3.1.3, we use the 200-dimension pre-trained Glove word embedding of the same dimensionality as the AutoNER BC5CDR demo experiment, and we first use the same default settings for its hyperparameters [42]. By using the AutoNER method on the processed data set, we found that both the development set and the test set can only achieve an F1-score of about 0.2. For AutoNER, the training set is first generated through distant supervision of dictionary matching, and then soft classification is performed based on the training set obtained by this distant supervision process. Therefore, it is expected that AutoNER does not perform as well as the fully supervised CRF model on the benchmark dataset. It may be because the quality of the annotated dataset produced by the first step of the distant supervision is not high, or it may be that the neural network learning part of the second step is not properly set, resulting in insufficient learning. As a result, in response to this conjecture, we first cleaned up the dictionary, including deduplication, language filtering, and so on. But this series of actions did not bring significant benefits.

In order to improve the quality of AutoNER, we made the following adaptations to the resources and process.

Dictionaries Through careful study of the distant supervision process, we found that noise in the full dictionary may be a major factor affecting the results. As Shang

mentioned in [42], the full dictionary can be obtained in two ways: Mining on domain-specific corpus through AutoPharse [41]; or obtaining it from the domain-specific dictionary.

In the experiment, we first tried to empty the full dictionary and found that the F1-score result was doubled compared to the initial F1-score 0.2. Therefore, the noise of the full dictionary was the main reason for the large deviation between the distant supervision result and the gold annotation. Since the previous full dictionary was excavated from the EU parliament debate report through AutoPhrase [41], however, in the excavation process, all the entities that appear frequently on the wiki are considered as a piece of knowledge in the excavation process. So, among the generated high-frequency vocabulary, there are many entities that have appeared in the wiki, but they do not necessarily appear in our corpus. This part of the data appears as a dictionary of distant supervision, which will introduce a large number of false positives, which makes the precision very low. In addition, this part of the dictionary data that is easy to cause misunderstanding of the model will also cause confusion in the results of word chunking, thereby affecting the overall effect of the model.

So we started trying to build a cleaner and more efficient full dictionary for better performance. In our solution, we decompose the constructed dictionary introduced in section 3.2. First of all, take this part of the vocabulary obtained from the mapping of a known foreign dictionary as the core dictionary, and the data we have excavated from EU parliament-related websites, etc. as the full dictionary. The reason for this classification is that the known exogenous dictionary is fine-grained, and we map it to a more coarse-grained classification, which can enrich our vocabulary while strictly controlling the internal inclusion of each entity category, and get a more general political domain dictionary.

The entities constructed by our mining are actually obtained by the specified classification, and there may be a certain deviation in the accuracy of the classification. In addition, this part of the information is mainly taken from EU-related websites, which is a limited knowledge data for the EU. Therefore, using this part of the data as a full dictionary can supplement EU-related knowledge without specifically subdividing it into corresponding entity types to achieve the main purpose of EU political domain research. Therefore, after our processing, the specific situation of the obtained dictionary is shown in Table 11.

Case-sensitive filtering As mentioned earlier, we found that the results of distant supervision have a great impact on the performance of AutoNER. In the dictionary string matching process of the AutoNER method, the case is not distinguished.

Table 11: The generated core dictionary and full dictionary for benchmark experiment

	LOC	PER	ORG	MISC
Core Dictionary	166,546	873,849	243,243	90,525
Full Dictionary		10,550		

In our case-sensitive corpus and dictionary, a large amount of false-positive data will be introduced, which will affect the final result. Therefore, for this, we will add case-sensitive filtering to the process of distant supervision. To be specific, for the entities in the core dictionary, only their own and uppercase forms are added to the matching list, so that case sensitivity can also be used as the main factor when matching.

Through the above process, we can get the performance of the AutoNER method and supervised learning method CRF on the same benchmark dataset as shown in Table 12. And the detailed results for each entity type are in Table 13. What's more, among them, for CRF, because the output report is for BIOES to obtain precision, recall, and f1-score in each entity category, for a better horizontal comparison to AutoNER, we will compare the average precision, recall and f1-score within each entity category respectively. For the traditional CRF method, we mainly use the crfsuite method of scikit-learn [32]. For each token's capitalization, Part-Of-Speech and related features of its adjacent words as the main features of CRF.

Table 12: The Precision, Recall and F1-score of AutoNER and CRF on the benchmark dataset

	Development set			**Test set**		
	Precision	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
AutoNER	0.577	0.689	0.628	0.460	0.589	0.516
CRF	0.847	0.790	**0.814**	0.874	0.795	**0.829**

Through the results, we can find that the quality that the AutoNER method can achieve is lower than that of the CRF method. This is because the supervised method itself knows the result of the annotation. For the model, as long as the known type of sequence information and features are learned, the result of a given pattern in a given sentence can be classified more accurately. As for AutoNER's method, it only has two domain-specific dictionaries, and all labeling results are generated by distant supervision. As we all know, it is difficult to achieve satisfactory results with only

Table 13: The Precision, Recall and F1-score of AutoNER and CRF on the test set for each entity type

	AutoNER			CRF		
Entity Type	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
LOC	0.582	0.907	0.710	0.886	0.845	0.864
PER	0.167	0.083	0.111	0.907	0.569	0.702
ORG	0.442	0.564	0.495	0.842	0.697	0.755
MISC	0.204	0.204	0.204	0.755	0.650	0.711

dictionary matching methods [25]. The AutoNER method achieves acceptable results to a certain extent by combining dictionary matching and deep learning methods.

For the results of each category in Table 13, we can find that AutoNER has a better recognition effect on the two categories of Location and Organization, while relatively, it has a poor recognition effect on Miscellaneous and Person. It may be related to the dictionary we built. In the core dictionary, each record of MISC is a longer sentence, while for LOC, there is more than one word and one entity sample. After our aforementioned adjustments, the current full dictionary is obtained by entering 4 entity types from the mapping that we have filtered from the external fine-grained dictionary. It is worth noting that, as shown in Table 9, the number of PER in the full dictionary is the largest, reaching 5 times that of ORG and LOC. Through manual checks, we found that in these examples, there are some non-English names, and some abbreviations, these messy and unclean results may be a major reason for the poor recognition of PER.

Since the AutoNER hyperparameter scheme, we initially adopted was consistent with the original experiment, but during the training process, it was found that the parameter combination reached the optimal result when half of the setting of the training epoch was completed, and even dropped afterward. Therefore, it is suspected that there is a certain over-fitting problem, and how to choose the optimal hyperparameter combination for our experiment has become a new problem. In response to this problem, we use the grid search method [7]. At the same time, for a fair comparison, we have also optimized the hyperparameters of the CRF method, specifically searching for the coefficient of regulation on a separate validation set. For the specific parameter space, please refer to Table 15.

By agreeing on the parameter space of AutoNER as shown in table 14, searching, and applying random searching [7] for CRF , then we obtained new results as shown in Table 16 and Table 17. The parameters we used to achieve this result are as follows:

Table 14: The AutoNER hyperparameter space of the optimization of benchmark experiment

Hyperparameter	**Optimization Space**
optimization	Adam, SGD, Adadelta, Adagrad
learning rate	SGD: [0.01, 0.5], step = 0.05 others: [0.001, 0.01], step = 0.001
layer number	2,3,4
rnn unit	rnn, lstm, gru
batch size	1000, 2000, 3000, 5000

Table 15: The CRF hyperparameter space of the optimization of benchmark experiment($PDF(x) = \lambda * e^{(-\lambda * x)}$ and the scale for exponential function is $1/\lambda$)

Hyperparameter	**Optimization Space**
L1 regulation	scale=0.5
L2 regulation	scale=0.05

For AutoNER: Optimization: SGD, Learning rate: 0.1, Layer number: 2, RNN unit: rnn, Batch size: 5000. For CRF: L1 regulation: 0.068, L2 regulation: 0.093.

Table 16: Comparison of the results of AutoNER and CRF on the benchmark dataset before and after hyperparameter optimization

	AutoNER			**CRF**		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Before	0.460	0.589	0.516	0.874	0.795	0.829
After	0.487	0.602	0.538	0.888	0.804	0.840

We can find that through hyperparameter optimization, the performance of AutoNER in the benchmark dataset can be improved to a certain extent, and precision, recall, and f1-score are all improved.

Entity Type	AutoNER			CRF		
	Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
LOC	0.589	0.884	0.707	0.886	0.845	0.864
PER	0.667	0.333	0.444	0.958	0.569	0.702
ORG	0.450	0.574	0.505	0.868	0.7301	0.788
MISC	0.256	0.224	0.239	0.775	0.672	0.719

Table 17: The Precison, Recall and F1-score of AutoNER compared to CRF on the benchmark dataset after hyperparameter optimization for each entity type

4.2 Europarl Written Records Data Experiments

In the written data experiments, for the dictionary, we used the same dictionary optimized by the benchmark experiment. We firstly use the randomly selected 200 sentences as the test set and other sentences as the training set. And the distribution of our manual annotated test set is shown in Figure 7. In the manually annotated test set, we annotated ORG, LOC, PER, MISC for 32, 31, 18, and 75 times respectively.

Inspired by the benchmark data experiments, to compare its performance on the corpus of different noise levels, the size of the dataset should be consistent. We randomly sampled 5,000 sentences and 10,000 sentences respectively for the experiments. The experimental results of the two training sets are shown in Table 18 and Table 19.

Table 18: The AutoNER performance on the test set of 5,000 sentences training experiment and filtered 5,000 sentences training experiment

	5,000 sentences			filtered 5,000 sentences		
	Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
Overall	0.321	0.234	**0.271**	0.316	0.234	0.269
LOC	0.375	0.670	0.488	0.385	0.667	0.488
PER	0.560	0.167	0.261	0.500	0.167	0.250
ORG	0.360	0.281	0.316	0.333	0.313	0.323
MISC	0.115	0.041	0.060	0.115	0.041	0.060

The Europarl written records experimental data repeats the basic conclusions of the benchmark dataset experiments. The difference is that the Europarl written records

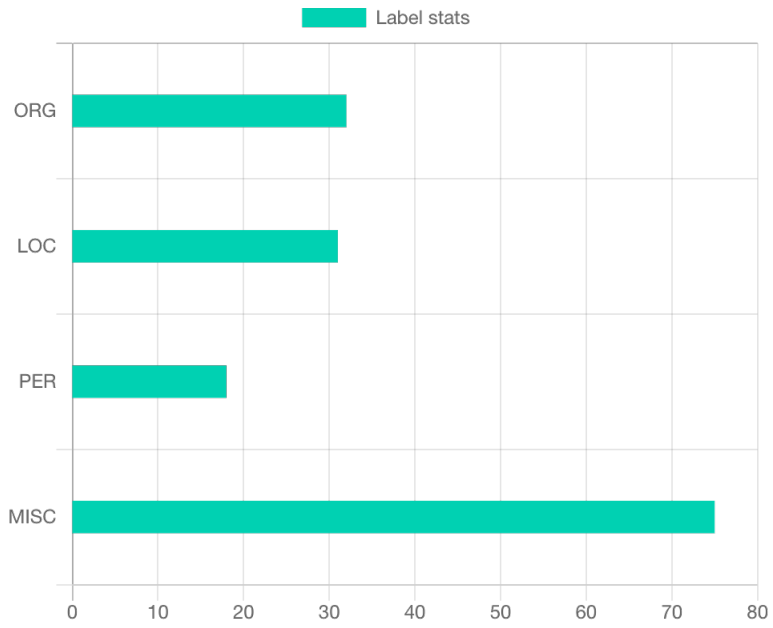

Figure 7: The manual annotated label distribution of 200 sentences of Europarl written records dataset

dataset set is closer to the actual situation of the algorithm application object of our subject request than the benchmark. But this is not the whole truth, we through controlling the training corpus from two aspects to explore the entry point of improving the performance of AutoNER method.

First of all, We noticed that there are a large number of sentences in the training corpus, all tokens in one sentence of which are all 'None' after the dictionary matching in the initial stage of AutoNER. This part of the sentence accounts for nearly half of all generated initial training corpus (2.4k/5k, about 50%). The AutoNER method is still a sequence labeling model, and the essence of the sequence labeling model is to classify each token in the sequence. Therefore, the sequence labeling model is essentially a multi-classification model applied to the sequence. And the minimum granularity used for learning in the training corpus is the label of each token. For those training sentences with not all 'None' labels, the entity label in all tokens is a minority, and for those sentences with all 'None', the proportion of the entity label is 0. From this point of view, it is unreasonable that the number of 'None' in the training corpus accounts for an overwhelming majority. Based on such a natural idea, we designed a comparative experiment. The specific method is that we first filter the annotation results obtained by the dictionary matching of the original corpus in the early stage of

AutoNER, and filter out the sentences that do not contain entities. In this way, the comparison results are also shown in Table 18.

In the comparative experiment, all sentences that were only 'None' were eliminated, which means that about 50% of sentences in the training set are removed. It can be seen from the performance in the table that the indicators after excluding all none sentences have not changed significantly from before, and the indicators of each category are the same. It can be concluded that the sequence labeling model of a deep neural network based on RNN is insensitive to the proportion of positive and negative examples of the training corpus of the input model, at least for the overwhelming majority of a certain classification label (None). The current proportion of the number of entity tags is sufficient for model learning, and changing the type and composition of tags will not improve the model's improvement effect. On the contrary, the composition of the training corpus deviates from the test set and the training set, resulting in a slight decrease in the effect. Therefore, we infer that the main direction to further improve the quality of the training data set should be to improve the efficiency of automatic labeling, rather than adjusting the proportion of each label type under the condition of reproducing the labeling quality.

In addition, we found that in the 5,000 sentence experiment, the effects of the PER and MISC categories were poor, so we tried to strengthen the training set of the PER category of the 10,000 sentence training set to verify whether it would influence the final effect. The specific method is that we used the search idea of trie tree [20] to perform simple and fast dictionary matching. We scanned the sentences containing the PER category words in the core dictionary in the full training set, and randomly selected 5,000 sentences from them, add to the 10,000 sentences training set. It is worth noting that when we filter the sentences containing the PER category in the core dictionary, we have not excluded the 10,000 sentence training set that has been selected. In the process of model training, the training set data is too small, and targeted oversampling is also a way to expand and strengthen the data set. After the aforementioned processing, we got a new training set of 15,000 sentences. The comparison between it and the original training set of 10,000 sentences can also be seen in table 19.

We can find that the precision of the PER category has improved after we strengthened, while the recall remains unchanged. But at the same time, the recall rate and accuracy rate of the MISC category are very low.

And Table 20 shows the performance of AutoNER on different sizes of the training set. We can find that the performance of the models trained for training sets of different sizes is slightly different. When we expand the training corpus from 5,000 sentences to

Table 19: The AutoNER performance on the test set of 10,000 sentences training experiment and 15,000 sentences(10,000 + 5,000 person sentences) training experiment

	10,000 sentences training experiment			15,000 sentences training experiment		
	Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
Overall	0.293	0.221	0.252	0.304	0.221	**0.256**
LOC	0.328	0.700	0.447	0.368	0.700	0.483
PER	0.500	0.111	0.182	0.667	0.111	0.190
ORG	0.320	0.250	0.281	0.286	0.250	0.267
MISC	0.130	0.041	0.062	0.125	0.041	0.061

10,000 sentences, to avoid over-fitting the model, the learning rate is adjusted slightly. This point can probably explain why the expansion of the training data set did not substantially improve the final result. However, for the 1,0000 and 15,000 experiments, it is not the increase in F1-score brought by the expansion of the training data of 5,000 sentences, but we targeted the poorly performing PER categories in the training data, using a simple dictionary matching over-sampling in the raw corpus.

Table 20: The overall performance on AutoNER of different size training corpus

the size of training set	**Precision**	**Recall**	**F1-score**
5,000 sentences	**0.321**	**0.234**	**0.271**
10,000 sentences	0.293	0.221	0.252
15,000 sentences	0.304	0.221	0.256

On another hand, we also used the traditional supervised learning method CRF model trained on the EU political-related corpus used in the benchmark experiment in Section 4.1 and applied it to the task of the test dataset of Europarl written records dataset. The results obtained are shown in Table 21, here, AutoNER uses the best-performing model trained on the 5000-sentence training set for comparison.

From the results, we can find that on the real-world corpus NER tasks, the pre-trained supervised learning method CRF still cannot handle it well. And the performance of each entity type is quite similar to the AutoNER, it can identify the LOC category much easier and much more difficult in PER and MISC. This may have a certain relationship with our data distribution. In the actual corpus, most of the LOCs are

Table 21: Comparison of the performance of AutoNER and CRF on the Europarl written records test dataset

	AutoNER			CRF		
	Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
Overall	0.321	0.234	**0.271**	0.648	0.148	0.191
LOC	0.375	0.670	0.488	0.804	0.514	0.623
PER	0.560	0.167	0.261	0.750	0.113	0.194
ORG	0.360	0.281	0.316	0.396	0.250	0.294
MISC	0.115	0.041	0.060	0.553	0.083	0.105

some geographical names and country names, which are better to identify. PER is less distributed in the test corpus, which may have a certain impact on the results. The boundary of the MISC category itself is more ambiguous, so this task may be more difficult than other categories. For both AutoNER and CRF, it is more difficult to achieve better results.

4.3 ASR Output Data Experiments

Similarly, for the ASR output data experiment, we use the best dictionary constructed by the benchmark experiment. Similarly, we randomly choose 200 sentences then manually annotate these sentences for the evaluation. The final annotated label distribution can be found in Fig 8. In the 200 sentences of test data of ASR output, we annotated 28 entities for ORG, LOC 14 entities, PER 20 entities, and MISC 25 entities.

Firstly, we use the similar strategy of benchmark data experiments and Europarl written records dataset experiments, we randomly sampled from the 77,497 5-30 words sentences to get the 5,000 sentences training set. Then we use the dataset combined with the manual annotated 200 evaluation sentence for the AutoNER experiments. However, in the trying process, we found that the f1-score of the initial AutoNER experiment remains 0.02 all the time. So we pay more attention to the dataset, we find that the poor performance might result in 4 aspects: ASR errors, Punctuations and capitalization, Entity distributions, and Entity types. To make AutoNER work for such a difficult task, we take a much deeper analysis of them and make some specific handle methods. The specific analysis is as follows:

ASR Errors As we know, automatic speech recognizes might introduce some errors, such as spelling errors, wrong words, etc. What's more, when a person is speaking,

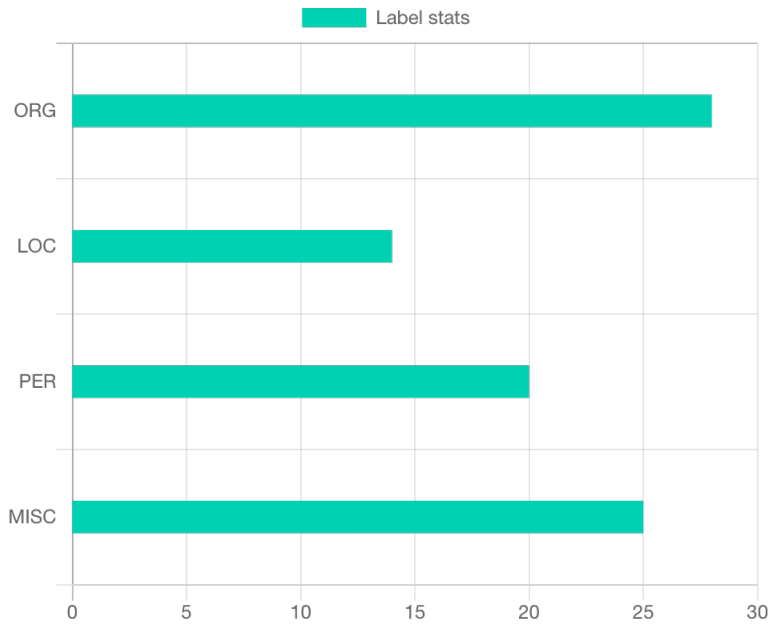

Figure 8: The manual annotated label distribution of 200 sentences of ASR output data

he might repeat the same pieces of words, and ASR cannot recognize this situation. For example, here is one sentence of the ASR output dataset:

Thank you very much i also want to join the voices which shuws tressing their the urgency of their rigts for tisdisabled people which just last week had in talen.

It can be found that there are many spelling errors in the sentence. Therefore, in response to this situation, we adopted NeuSpell [18] automatic error correction method to correct spelling errors. In actual operation, we use the BERT-based pre-trained model provided in the toolkit. Specifically, automatic error correction is performed for the divided sentences in the previous preprocessing stage, so as to correct spelling errors to a certain extent. The following shows the result of the sentence in the above example after NeuSpell:

Thank you very much I also want to join the voices which shows stressing there the urgency of their rights for tisdisabled people which just last week had in taken .

We can clearly see that for spelling errors such as 'rigts', 'shuws', 'talen', NeuSpell can successfully correct them to the correct format. However, for errors such as 'tis-

disabled', which may be caused by too close to 'disabled' or repeated pieces of words in the speech, it cannot be effectively corrected. Taking into account that it may be because NeuSpell is a correction for spelling errors, but 'tisdisabled' is no spelling error in its opinion.

In fact, for such non-spelling errors that NeuSpell cannot correct, there are also some grammar problems, which are common in ASR output data. This kind of problem can only be solved by making the ASR process better understand the content, or by applying error correction tools that understand and correct the content. However, due to the quality of ASR output and the lack of reliable error correction tools for understanding content, it is currently difficult to solve. And this defect may become an aspect that will limit our AutoNER results in the future.

Punctuation and Capitalization As we mentioned in section 3.1, the original output of ASR is in all capitals and there is no punctuation. Although in the preprocessing stage, we used Punctuator [48] to automatically punctuate the output text. As mentioned in this tool, it was developed on the Europarl data set, and the overall F1-score of this pre-trained model reported was 74.8 [48] not 100% truly. And our corpus is mainly the text converted from the voice record of the European Parliament debate, which is not the same as their trained benchmark Europarl data set. Therefore, in the process of punctuation, a certain error rate will also be introduced. Wrong punctuation marks will lead to certain mistakes in sentence splitting and word segmentation.

On the other hand, the original ASR output data is all uppercase and is not case sensitive. In the process of processing, it is currently all converted to lower case and then capitalizes the first letter of each sentence, the proper nouns, I, etc. the capitalization in the middle of the sentence of the title is not considered. Due to the lack of a reliable capitalization tool, the processing of this part is relatively rough. At the same time, the current scheme is based on the correct result of the clause. Therefore, this part has a certain amount of noise introduced.

But at the same time, we noticed that AutoNER's dictionary matching process has lowercase filtering for entities in the core dictionary. Specifically, only the entities in the dictionary and their all-capital form are added to the matching list to prevent the introduction of noise while ensuring that as many entities as possible are recognized. Our current situation is a lack of specific capitalization. Therefore, through the transformation of AutoNER's first step of remote matching, entities in the dictionary that do not contain stopwords are also added to the matching list to adapt to our current corpus. Among them, the filtering of stopwords is to prevent the mislabeling of

stop words in lower case. Facts have proved that through such adaptation, the entity recognition performance of ASR output data has been improved a lot, as shown in Table 22.

Entity distributions Through our statistics, we found that in the manually annotated test dataset, only 25% of the sentences with entities, and most of the other sentences are negative samples that do not contain entities. Since the annotation data of these 200 sentences are randomly selected from the original corpus, it is inferred that the entire sentence does not have an entity and also exists in the training corpus.

For the sentences with 5-30 words, we want to explore the influence of the percentage of entities as well as avoiding over-fitting by just simply copying. Therefore, given the extremely uneven distribution of entities, we want to increase the distribution of the entity in the training set. The specific method is to filter out all sentences marked as None in the remote annotation, and only keep sentences that contain at least one entity, here, we mainly use the results of dictionary matching to filter. Then by over-sampling the remaining sentences after filtering, copy 5 times of each sentence to generate a new training set. We also did a comparative experiment of filtering then over-sample and only filtering negative samples without over-sample.

Entity types In the entire data set and dictionary construction process, we have preset 4 entity types (LOC, PER, MISC, ORG) that need to be recognized. However, as we mentioned earlier, ASR errors, punctuation errors, indistinguishable capitalization in sentences, imperfect automatic error correction, and repetition of human speech and expression errors, etc., will introduce a certain amount of noise to the ASR output dataset. And these noises further aggravate the difficulty of our named entity recognition task. Moreover, a major improvement of AutoNER is that it can reasonably solve the multi-label problem in traditional dictionary matching. Through the remote labeling stage, the categories that may belong to are temporarily labeled. This means that we have 4 types of labeling tasks, and it is possible to generate 4*4 types of label combinations. In this way, the difficulty of classification for subsequent classifiers is 4 times.

Therefore, in response to this problem, we used not to measure how many entities were identified by each category, but to measure how many entities AutoNER can correctly identify, thereby reducing the difficulty of identifying a large amount of noisy data. Specifically, in the implementation, the four categories are not distinguished, and all the entity types in the dictionary and the manually annotated test dataset are unified into the same. In this way, for the entire named entity recognition task, it becomes

just another type of entity task.

At the same time, we also applied the supervised learning NER method CRF trained by the benchmark in Section 4.1 to the ASR output test dataset. Similarly, the previous 4 named entity categories are unified into one, so that it is easy to compare the number of entities that can be accurately identified on the noisy data set by these two different methods. Through the above scheme, we have carried out several experiments, and the specific experimental results are shown in Table 22. The precision, recall, and f1-score are all calculated based on the unified entity category.

Table 22: The experimental results of ASR output dataset

Experiments		**Precision**	**Recall**	**F1-score**
AutoNER	baseline	0.250	0.012	0.022
	lowercase	0.170	0.388	0.237
	lowercase + filter negative samples	0.154	0.400	0.222
	lowercase + filter negative sample+ enlarge 5 times	0.176	0.424	0.249
	lowercase + NeuSpell	0.189	0.376	0.252
	lowercase+ NeuSpell + filter negative samples + enlarge 5 times	0.304	0.247	**0.273**
CRF		0.236	0.014	0.027

Through the experimental results, we can find that even after simplifying the NER problem to only counting the number of entities that are correctly recognized, the ability of the distance supervision method AutoNER and the traditional supervised learning method CRF to solve the named entity recognition problem of the ASR output dataset is both limited. By observing the experimental results, we found that the recall rate of the two methods(CRF and the AutoNER without external preprocessing) in this part of the dataset is very low, which may be due to the ASR conversion error, the inaccuracy of human speech, the error of the punctuation clause, and the capitalization, etc. we mentioned earlier, makes the noise of this part of the data very large. It is worth noting that we used the benchmark experiment pre-trained

model for the CRF model here, which was also trained on the benchmark corpus with less noise and case sensitivity. By comparing with the AutoNER method without lowercase adaptation, it is found that the performance is very similar. Capitalization might be the main factor restricting CRF. We have passed a series of preprocessing methods to enable AutoNER to achieve an overall F1-score of **0.273** in this such a high-noise situation, which may indicate that AutoNER is more adaptable to corpus than CRF.

4.4 Post Analysis

As we mentioned in Section 3, for these large amounts of unlabeled EU parliament debates record related corpus, to evaluate the performance of AutoNER in such situation, we first need to manually annotate the selected test set of the corpus. For Europarl written records dataset experiment and ASR output data experiment, we randomly selected 200 sentences from the processed corpus, respectively, and then manually labeled them as the test set. Although manual annotation is the default ground truth in the experiments the quality of this process involving human behavior requires an intuitive statistical evaluation.

Because raw ASR output corpus has certain spelling errors, no punctuation, and no case distinction. It adds a series of difficulties to our manual annotation work after adding punctuations and sentence tokenizations. At the same time, a series of preprocessing such as punctuation, spelling correction, and case conversion may introduce a certain amount of noise. Therefore, we choose the relatively clean Europarl written records dataset constructed by the European parliament debates report to make the post-analysis of the quality of manual labeling. Specifically, in the experiment, we all use the data annotated by the first annotator, and in the process of post analysis, let the second annotator follow the same annotation guideline to annotate the same Europarl written records dataset test set. Then we use the inter-annotator agreement (IAA) to measure the consistency of the annotated labels between two annotators on the same corpus [1].

The labeling of named entities has a large number of 'O' labels (non-entity) which leads to a strong label imbalance in the data. Therefore, ordinary Kappa calculations for inter-rater agreement may not be able to give an objective evaluation of multi-annotator consistency measurement. Therefore, in addition to the ordinary Kappa value calculation for the 200-sentence Europarl written records dataset test corpus, we also adopted the Kappa calculation mentioned by Brandsen et al. in [5] for only labeled tokens to exclude the impact of a large number of 'O'. It is worth noting that

as long as one of the two annotators has annotated for this token, it is considered to be an annotated token and is included in the scope of the Kappa calculation. In addition, Brandsen et al. also mentioned another method to measure the consistency of manual annotations of named entities. That is, for two annotators, consider the result of one person as a true label and the other person as a predicted label, and then calculate the Precision, Recall, and F1-score of the two. After the calculation is completed, exchange the true and predict positioning, recalculate, and then take the average F1-score to measure the consistency of the two annotators [5]. The results of the above calculations are displayed in Table 23.

Table 23: The IAA of two annotators on the Europarl written records dataset test set

the Cohen's Kappa of all tokens	0.790
the Cohen's Kappa of annotated tokens	0.502
average F1-score	0.970

In addition, for the above-mentioned calculation of the quasi-call rate by treating the labeled tags as true and predict labels respectively, we display the confusion matrix in Figure 9.

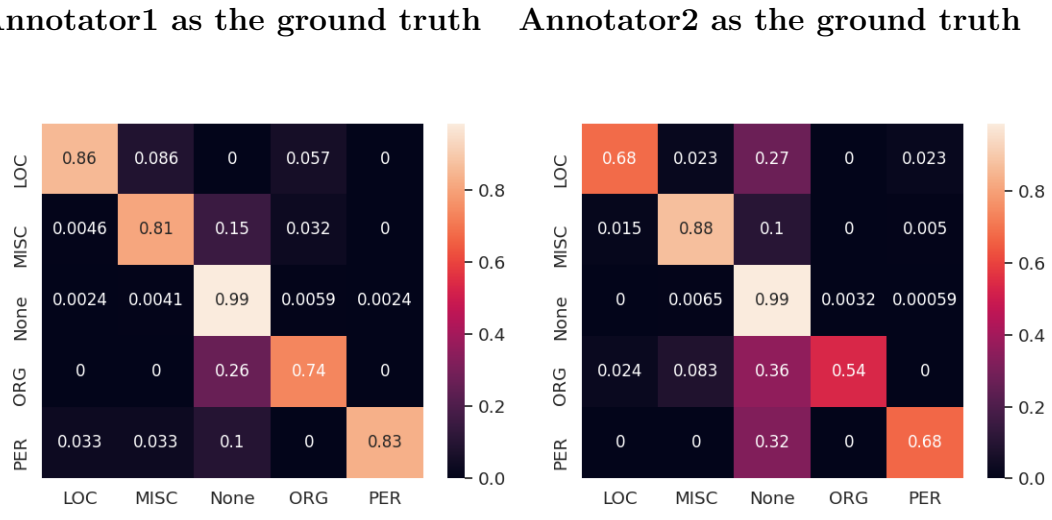

Figure 9: The confusion matrix between the annotations of two annotators

Through the results, it is not difficult to find that for all tokens, Kappa shows that the two annotators are marked as '**Substantial agreement**', and only at the level of the annotated tokens, the two annotators are marked as '**Moderate agreement**' [50]. At the same time, by observing the confusion matrix of the two, it is found that the consistency of the annotation results is still acceptable, but there may be some disputes about the labeling of the 'ORG' category. Later, through the check specific labeling results, it was found that there was controversy as to whether words such as 'Committee' belong to the entity category of 'ORG', which is the main reason for the difference in manual annotations.

5 Discussion and Future works

From the analysis of the experimental results of benchmark experiments showed in Section 4.1, it can be concluded that although automatic annotation solves the problem of missing annotated data very well, its performance is still largely restricted by the quality of training corpus and dictionary matching, and the main manifestations and reasons of the constraints are as follows in my point of view:

1. From the training corpus itself, there is a problem that the entity types under the same category entity are too concentrated. Analyzing the performance of AutoNER on the test set (a total of 239 sentences, 6,079 tokens), we find that the predictions given by the model tend to focus on certain combinations of certain words, and these combinations are the combinations that appear most frequently in the training set. For example, the top three entity words appearing in the ORG category in the test set are United Nations (36 times), Security Council (19 times), European Union (11 times), they accounted for a total of 55% dictionary matching result of the training set of ORG category. But at the same time, assuming that the results of manual annotation in our test set are correct, the sum of the three entities accounts for 75% of all the manual annotated ORG labels in the test set (United Nations 38 times, Security Council 19 times, European Union 14 times). This result is much higher than the ratio predicted by the model at the end. For further analysis, we counted the label entities that were automatically annotated in the training set using dictionary matching in the first stage of AutoNER. It was found that these three accounted for 22.18%, 5.98%, and 1.34% of all automatically labeled ORG entities (United Nations 679 times, Security Council 183 times, European Union 41 times, total 3061 times).

   It is worth mentioning that although the test cases in the test set are randomly selected sentences that are homogenous with the training set, the annotations are performed using a manual method with relatively high quality. The three examples we mentioned in the automatically annotated training corpus accounted for less than 30% of the total automatic tags, but in the manual tags, they accounted for three-quarters, indicating that there are a large number of invalid tags in the automatic tagging internal results. As we all know, the introduction of too many invalid annotations increases the noise of the training corpus. The excessive concentration of effective entities in the corpus itself is an important reason for this phenomenon because the diversity of effective entities is too small to make the effective value in the vocabulary sufficient hit the training corpus, it is impossible to extend the gap with the invalid value in the vocabulary. And

this shortcoming is unavoidable in automatic annotation to some extend.

2. In terms of the automatic annotation mechanism, the algorithm used for dictionary matching is also too rough. The basic tagging algorithms are all based on a limited list of entity vocabulary. There is basically no possibility of generalization of the patterns that may appear in certain categories. It adopts a more tolerant attitude to the coverage of the vocabulary and the rate of mislabeling. Although this reduces the degree of manual work and improves automation efficiency, it also introduces too much noise and labeling errors, which confuses the model learning. In addition, because the effective words in the vocabulary used for tagging—that is, the words in the vocabulary that are actually used to tag and accurately hit the taggable words in the training set are very few and concentrated compared to the vocabulary itself, This is equivalent to producing such an effect: on the one hand, the accuracy of the label itself is very low, on the other hand, the accurate part of the label is too concentrated. According to the GIGO principle [38], when the training examples that can provide accurate generalization ability are too scarce in the training corpus, the entire AutoNER is naturally unable to achieve satisfactory neither the prediction accuracy nor the generalization ability.

3. In the process of dictionary cleaning, we also try more preprocess and use a larger and more diverse training corpus to help solve the above problems. However, it is limited by the difficulty of obtaining experimental corpus, the labor cost that may be consumed to clean the vocabulary, and the risks of related information security laws and regulations. At present, it is difficult to carry out further experiments. Adopting a more complete labeling strategy, such as providing some more flexible pattern auxiliary labeling, etc., can also achieve the purpose of improving the effectiveness, but this problem is currently beyond the scope of this topic and will be left for future research.

When it comes to the Europarl Written Records Data Experiments introduced in Section 4.2, after analyzing the training corpus and labeling standards, we analyze the possible reasons for two points:

1. The definition of the MISC category itself is blurred. Compared with the other categories (location, person name, organization name) in this task, the MISC category is not so much a separate category as it is the bin of some meaningful unclassified entity. In this case, the concept The boundaries of the category must be blurred. The core idea of AutoNER to realize 'Auto' is automatic labeling. The automatic labeling system based on this experiment is realized by vocabulary matching. This technology is very working when the category boundary is

clear and the context pattern is typical and clear. But if the category boundary is not clear, it is difficult to locate the typical context pattern in the labeled data, then the model can learn from the training corpus is naturally very limited.

Corroborating evidence for this conjecture is that the organization name and location have clear conceptual boundaries, but the organization name is divided randomly. For example, the European Commission for Foreign Affairs recognizes it as "EU" when it appears in the corpus. Whether the two organizations of the "Foreign Affairs Committee" is a separate organization of the "EU Foreign Affairs Committee" is debatable. This is particularly difficult for the automatic labeling system implemented by this table matching. Similarly, place-name category entities have the same problem. But relatively, the boundary of the named entity is very clear, and we rarely encounter situations where there are multiple solutions to the boundary of the named entity. In fact, judging from the accuracy of AutoNER, its quality is arranged in the order of person name $>$location name, organization name $>$miscellaneous.

2. Another possible reason is that the MISC category itself appears too infrequently in the training corpus, which can explain why the category has a low recall rate when the accuracy rate is low. According to my statistics, without considering interventions on the frequency of occurrence of each category entity, the MISC category only accounts for 18 percent of the training expectations. Combined with the fact that the category has diversified connotations, the model cannot effectively or more accurately learn the entity's pattern and recurrence conditions. Therefore, the prediction of this category is significantly less, resulting in an insufficient recall.

As a classical NLP task, NER plays an important role in various text analysis tasks, especially in domain-specific ones, which provide important basic functions. In particular, the use of AutoNER technology that does not rely on tedious manual annotation can provide an efficient and cost-acceptable method when the complexity of the document to be parsed is high and the manual annotation cost is high. In the political field studied in this article, the application of AutoNER is promising. However, we found through experiments when the text itself is noisy, the performance of the AutoNER system will be strongly affected, and its performance needs to be improved by the continuous update and optimization of automatic annotation methods and algorithms.

This paper studies the performance of AutoNER on three different text corpora, namely the filtered benchmark EU political text corpus, the unscreened EU political meeting written record corpus, and the raw political text corpus recognized by ASR. The noise

content of these three corpora increases in sequence, and the richness of their corpus composition also increases. Among the three corpora, the closest and most likely application scenarios are actually the latter two, so we pay more attention to the performance of AutoNER on the latter two. Therefore, in the case that the AutoNER method does not perform well on the latter two raw data, we have adopted a series of optimization and preprocess methods so that it can solve the problem of Named Entity Recognition of a large number of the unlabeled corpus of the political domain to a certain extent. Experiments show that the performance of AutoNER on the latter two is greatly affected by the noise content of the corpus itself. This means that AutoNER is not competent for corpora with too poor a signal-to-noise ratio. On the EU political written corpus that has no spelling errors and recognition errors, the screened corpus has better performance than the unscreened corpus. The difference between the two lies in the degree of conformity between the corpus used and the domain of interest and the degree of standardization of sentence expression. It can be inferred that in addition to the signal-to-noise ratio, the selection of the corpus itself will also significantly affect the performance of AutoNER.

Based on the above conclusions, we draw further inferences. That is to say, no matter what kind of corpus or domain it is applied to, it is necessary for AutoNER to filter, correct and supplement the corpus and related information and prior knowledge before application. The effect of this kind of leading and processing work directly affects the final effectiveness of the entire AutoNER system has become an important factor added to the entire system. Through the research of this article, it is found that increasing the proportion of entities, selecting appropriate domain-specific dictionaries, improving the efficiency of entity information annotation, updating the algorithm logic of the dictionary matching algorithm, setting algorithm rules, cleaning the corpus and dictionaries to improve the corpus signal-to-noise ratio, supplementing the prior annotation knowledge, etc. The means are all effective means to improve the performance of AutoNER. If the combination and granularity of preprocessing can be adjusted for different domains and different application scenarios, the overall performance can be greatly improved. It is worth noting that not all preprocessing for the improvement of training corpus-related indicators is work. Even though we have proposed one working way in this thesis, there still some other conventional preprocessing methods that will be more effective for specific scenarios. How to choose the right combination of preprocessing is a topic still worth studying.

Except for the part of automatic annotation through domain-specific dictionary matching at the beginning, the deep model used in the entities recognition part of AutoNER follows the general NLP sequence labeling model paradigm. Adjusting various hyperparameters through the grid search method can affect the performance of the model

to a certain extent, but its impact is not decisive. By trying to adjust the hyperparameters of the model and the structure of the model, we can draw the inference that under the existing thinking framework of sequence annotation to solve the NER problem, the performance of machine learning models such as deep models or CRFs is not the bottleneck of the overall performance improvement. On the other hand, if automatic annotation and entities recognition can be integrated into one model to realize the automatic seq2seq [44] annotation pipeline, instead of separating the two parts like the AutoNER method studied in this article, then it is possible to realize a breakthrough in the overall performance of the system by improving the performance of the model itself. This issue needs to be studied in the future. At the same time, in the whole experiments, in order to realize an 'automatic' pipeline, we used the same public pre-trained word embedding-Glove [33]. Glove is not trained based on EU political domain expectations and in the real-word corpus, they are with plenty of noise. Therefore, there may be some domain-related vocabulary or noise tokens that cannot be covered in the vocabulary. This Out-Of-Vocabulary (OOV) problem will also affect the results [28]. Additionally, the quality of the dictionary can also be improved. Although in the experiment, since the biggest feature of the entire AutoNER-based system is 'auto', we used the automatic way to process in the dictionary construction without too much manual intervention. The quality might be limited, but the quality of this part of the dictionary can also be improved by some advanced dictionary construction methods such as bootstrapping [46].

To sum up, the AutoNER system studied in this paper can realize the automatic recognition of named entities under a specific domain (political) at the expense of certain acceptable performance. Its performance is mainly affected by the applied corpus and the quality of provided prior knowledge and the correction and effective combination of preprocessing can greatly improve the overall performance. Although we have done a lot of work when constructing the domain-specific dictionary and the corpus preprocessing process, this part of the work can be reused. At the same time, because it skips the tedious and expensive manual annotations and preparation of the corpus, AutoNER is more economical than other NER solutions which rely on heavy manual annotations, have a wider range of application scenarios, and better adaptability. Especially for the scenes of online learning in the industry or somewhere that needs to update the model daily, the framework based on AutoNER is obviously more economical than the daily manual annotations. If appropriate transformation and migration are made, it may be possible not to be limited to NER tasks, capable of more diverse tasks. It is a very promising framework for NLP processing ideas.

6 Conclusion

In this work, a new framework for NER in the EU politics domain is proposed. With the help of the AutoNER technique, the time-consuming and labor-intensive manual annotation work can be waived in the proposed method. Experiments on several evaluation datasets suggest that our method is better at dealing with the NER task in specific domains with acceptable performance and adaptability, especially when compared with the traditional machine learning-based methods, where tedious manual annotations and re-training are still required. Meanwhile, to further investigate the performance of AutoNER in the real-world scenario, we also evaluate AutoNER on three different datasets with increasing ratios of noise, to simulate the variant of input data quality. We notice that the AutoNER is quite sensitive to the quality of input data, and therefore, to take the most advantage of AutoNER, special cautions need to be paid to the data pre-processing. Last but not the least, we hope our work can be inspiring, and the domain-specific dictionaries we compiled in this work are also helpful to further studies in the related areas.

References

[1] Ron Artstein. Inter-annotator agreement. In *Handbook of linguistic annotation*, pages 297–313. Springer, 2017.

[2] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33, 2020.

[3] Akash Bharadwaj, David R Mortensen, Chris Dyer, and Jaime G Carbonell. Phonologically aware neural model for named entity recognition in low resource transfer settings. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1462–1472, 2016.

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

[5] Alex Brandsen, Suzan Verberne, Milco Wansleeben, and Karsten Lambers. Creating a dataset for named entity recognition in the archaeology domain. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4573–4577, 2020.

[6] Cristian Cardellino, Milagro Teruel, Laura Alonso Alemany, and Serena Villata. A low-cost, high-coverage legal named entity recognizer, classifier and linker. In *Proceedings of the 16th edition of the International Conference on Articial Intelligence and Law*, pages 9–18, 2017.

[7] Marc Claesen and Bart De Moor. Hyperparameter search in machine learning.

[8] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE):2493–2537, 2011.

[9] Ben Zhou Tom Redman Christos Christodoulopoulos Vivek Srikumar Nicholas Rizzolo Lev Ratinov Guanheng Luo Quang Do Chen-Tse Tsai Subhro Roy Stephen Mayhew Zhili Feng John Wieting Xiaodong Yu Yangqiu Song Shashank Gupta Shyam Upadhyay Naveen Arivazhagan Qiang Ning Shaoshi Ling Dan Roth Daniel Khashabi, Mark Sammons. Cogcompnlp: Your swiss army knife for nlp. In *11th Language Resources and Evaluation Conference*, 2018.

[10] Hakan Erdogan. Sequence labeling: Generative and discriminative approaches. ICMLA, 2010.

[11] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[12] Sahar Ghannay, Antoine Caubrière, Yannick Estève, Nathalie Camelin, Edwin Simonnet, Antoine Laurent, and Emmanuel Morin. End-to-end named entity and semantic concept extraction from speech. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 692–699. IEEE, 2018.

[13] Mohamed Hatmi, Christine Jacquin, Emmanuel Morin, and Sylvain Meigner. Incorporating named entity recognition into the speech transcription process. In *Proceedings of the 14th Annual Conference of the International Speech Communication Association (Interspeech'13)*, pages 3732–3736, 2013.

[14] Laura Hollink, Astrid van Aggelen, Henri Beunders, Martijn Kleppe, and Max Kemman. *Jacco van Ossenbruggen (2017)*. Talk of Europe - The debates of the European Parliament as Linked Open Data. DANS.

[15] Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

[16] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging.

[17] Kevin Humphreys, Robert Gaizauskas, Saliha Azzam, Christian Huyck, Brian Mitchell, Hamish Cunningham, and Yorick Wilks. University of sheffield: Description of the lasie-ii system as used for muc-7. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29-May 1, 1998*, 1998.

[18] Sai Muralidhar Jayanthi, Danish Pruthi, and Graham Neubig. NeuSpell: A neural spelling correction toolkit. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 158–164, Online, October 2020. Association for Computational Linguistics.

[19] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding.

[20] DE Knuth. The art of computer programming, vol 3: sorting and searching 2nd ed, 1998.

[21] George Krupka and Kevin Hausman. Isoquest inc.: Description of the netowl™ extractor system as used for muc-7. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29-May 1, 1998*, 1998.

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

[23] Elena Leitner, Georg Rehm, and Julian Moreno-Schneider. Fine-grained named entity recognition in legal documents. In *International Conference on Semantic Systems*, pages 272–287. Springer, 2019.

[24] Elena Leitner, Georg Rehm, and Julian Moreno Schneider. A dataset of german legal documents for named entity recognition. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4478–4485, 2020.

[25] 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.

[26] Jialu Liu, Jingbo Shang, Chi Wang, Xiang Ren, and Jiawei Han. Mining quality phrases from massive text corpora. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1729–1744, 2015.

[27] Tianyu Liu, Jin-Ge Yao, and Chin-Yew Lin. Towards improving neural named entity recognition with gazetteers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5301–5307, Florence, Italy, July 2019. Association for Computational Linguistics.

[28] Shubhanshu Mishra and Jana Diesner. Semi-supervised named entity recognition in noisy-text. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 203–212, 2016.

[29] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[30] Shuyo Nakatani. Language detection library for java, 2010.

[31] Hiroki Nakayama, Takahiro Kubo, Junya Kamura, Yasufumi Taniguchi, and Xu Liang. doccano: Text annotation tool for human, 2018. Software available from https://github.com/doccano/doccano.

[32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[33] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[34] Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155, 2009.

[35] Marek Rei, Gamal Crichton, and Sampo Pyysalo. Attending to characters in neural sequence labeling models. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 309–318, 2016.

[36] Nils Reimers and Iryna Gurevych. Optimal hyperparameters for deep lstm-networks for sequence labeling tasks.

[37] Artem Revenko and Georg Rehm. Automatic induction of named entity classes from legal text corpora. 2020.

[38] L Todd Rose and Kurt W Fischer. Garbage in, garbage out: Having useful data is everything. *Measurement: Interdisciplinary Research & Perspective*, 9(4):222–226, 2011.

[39] Erik Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, 2003.

[40] Injy Sarhan and Marco Spruit. Can we survive without labelled data in nlp? transfer learning for open information extraction. *Applied Sciences*, 10(17), 2020.

[41] Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R Voss, and Jiawei Han. Automated phrase mining from massive text corpora. *IEEE Transactions on Knowledge and Data Engineering*, 30(10):1825–1837, 2018.

[42] Jingbo Shang, Liyuan Liu, Xiaotao Gu, Xiang Ren, Teng Ren, and Jiawei Han. Learning named entity tagger using domain-specific dictionary. In *EMNLP*, 2018.

[43] Catarina Silva and Bernardete Ribeiro. The importance of stop word removal on recall values in text categorization. In *Proceedings of the International Joint Conference on Neural Networks, 2003.*, volume 3, pages 1661–1666. IEEE, 2003.

[44] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, 27:3104–3112, 2014.

[45] Charles Sutton and Andrew McCallum. An introduction to conditional random fields. *Machine Learning*, 4(4):267–373, 2011.

[46] Shaheen Syed, Marco Spruit, and Melania Borit. Bootstrapping a semantic lexicon on verb similarities. In *Proceedings of the 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, volume 1, pages 189–196. SciTePress, 2016.

[47] Anu Thomas and S Sangeetha. Performance analysis of the state-of-the-art neural named entity recognition model on judicial domain. In *Soft Computing: Theories and Applications*, pages 147–154. Springer, 2020.

[48] Ottokar Tilk and Tanel Alumäe. Bidirectional recurrent neural network with attention mechanism for punctuation restoration. In *Interspeech 2016*, 2016.

[49] Astrid Van Aggelen, Laura Hollink, Max Kemman, Martijn Kleppe, and Henri Beunders. The debates of the european parliament as linked open data. *Semantic Web*, 8(2):271–281, 2017.

[50] Anthony J Viera, Joanne M Garrett, et al. Understanding interobserver agreement: the kappa statistic. *Fam med*, 37(5):360–363, 2005.

[51] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

[52] Jie Yang and Yue Zhang. Ncrf++: An open-source neural sequence labeling toolkit. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 2018.

[53] Lufeng Zhai, Pascale Fung, Richard Schwartz, Marine Carpuat, and Dekai Wu. Using n-best lists for named entity recognition from chinese speech. In *Proceedings of HLT-NAACL 2004: Short Papers*, pages 37–40, 2004.