Karlsruhe Institute of Technology

Audience-specific Explanations for Machine Translation

Master's Thesis of

Renhan Lou

at the Department of Informatics
Institute for Anthropomatics and Robotics (IAR)
Artificial Intelligence for Language Technologies Lab (AI4LT)

Reviewer:	Prof. Dr. Jan Niehues
Second reviewer:	Prof. Dr. Alexander Waibel

September 30, 2022 – March 30, 2023

Karlsruher Institut für Technologie
Fakultät für Informatik
Postfach 6980
76128 Karlsruhe

I declare that I have developed and written the enclosed thesis completely by myself, and have not used sources or means without declaration in the text.

PLACE, DATE

. . . .

(Renhan Lou)

Abstract

In machine translation (MT), a common problem is that the translation results of certain words can cause incomprehension of the audience in the target language. Because these words are common in the source language, but not common in the target language. If these words are simply translated from the source language to the corresponding words in the target language, or just copy them into the translation result, this will make the audience in the target language unable to understand the meaning of the translation when they see the translation results.

To solve the problem of how to eliminate the incomprehension of the target language audience during the translation process, human translation provides a solution. That is to add explanations to the translation results for these words that will cause incomprehension to the target language audience. These additional information can well eliminate the incomprehension of the target language audience.

Therefore, the purpose of our research work is to explore whether it is possible to find a suitable model that can accurately predict which words need to be explained when performing machine translation tasks. The sparsity of sentences containing words that need to be explained makes building the training dataset extremely difficult. We therefore propose a heuristic method for finding sentences with words that need to be explained. With the help of a series of tools including named entity recognition, Wikipedia, etc., the method we propose can greatly reduce the final manual selection work, and at the same time, it can stably and efficiently find the target sentence in the last remaining sentences.

We conducted experiments on English→German, English→French and English→Chinese language pairs. The experimental results show that when the input is five million sentence pairs, our proposed method can reduce the number of remaining sentence pairs that may contain the target sentence pair to a very low number, and in the last remaining sentence pairs, a certain proportion of target sentence pairs with explanations can be found stably. In the last remaining sentence pairs, more than 10% of the target sentence pairs can be found for English→German, more than 7% target sentence pairs can be found for English→Chinese, and more than 5% target sentence pairs can be found for English→French.

The experimental results show that our proposed method of finding sentences containing words that need to be explained is positive and robust. This reduces the difficulty of building a training dataset and also facilitates the training of models in the future.

Zusammenfassung

Bei der maschinellen Übersetzung besteht ein häufiges Problem darin, dass die Übersetzungsergebnisse bestimmter Wörter zu Unverständnis des Publikums in der Zielsprache führen können. Weil diese Wörter in der Ausgangssprache häufig vorkommend sind, aber nicht in der Zielsprache. Wenn diese Wörter einfach aus der Ausgangssprache in die entsprechenden Wörter in der Zielsprache übersetzt oder nur in das Übersetzungsergebnis kopiert werden, dann kann das Publikum in der Zielsprache die Bedeutung der Übersetzung nicht verstehen.

Um das Problem zu lösen, wie das Unverständnis des zielsprachlichen Publikums während der Übersetzung beseitigt werden kann, bietet die menschliche Übersetzung eine Lösung. Das heißt, den Übersetzungsergebnissen für diese Wörter Erklärungen hinzuzufügen, die beim Publikum in der Zielsprache zu Unverständnis führen. Diese zusätzlichen Informationen können das Unverständnis des zielsprachlichen Publikums gut beseitigen.

Das Ziel unserer Forschungsarbeit ist es, zu untersuchen, ob es möglich ist, ein geeignetes Modell zu finden, das genau vorhersagen kann, welche Wörter erklärt werden sollen, wenn maschinelle Übersetzungsaufgaben ausgeführt werden. Aber die geringe Menge an Sätzen, die Wörter enthalten, die erklärt werden sollen, macht den Aufbau des Trainingsdatensatzes extrem schwierig. Deswegen schlagen wir ein heuristisches Verfahren vor, um Sätze mit Wörtern zu finden, die erklärt werden sollen. Mit Hilfe einer Reihe von Tools, einschließlich der Erkennung von benannten Entitäten, Wikipedia usw., kann das von uns vorgeschlagenes Verfahren die endgültige manuelle Auswahlarbeit erheblich reduzieren und gleichzeitig den Zielsatz in den letzten verbleibenden Sätzen stabil und effizient finden.

Wir haben Experimente mit den Sprachpaaren Englisch→Deutsch, Englisch→Französisch und Englisch→Chinesisch durchgeführt. Die experimentellen Ergebnisse zeigen, dass unser vorgeschlagenes Verfahren bei einer Eingabe von fünf Millionen Satzpaaren die Anzahl der verbleibenden Satzpaare, die das Zielsatzpaar enthalten können, auf eine sehr geringe Anzahl reduzieren kann. Und in den letzten verbleibenden Satzpaaren kann ein gewisser Anteil an Zielsatzpaaren mit Erklärungen stabil gefunden werden. In den letzten verbleibenden Satzpaaren können mehr als 10% der Zielsatzpaare für Englisch→Deutsch gefunden werden, mehr als 7% Zielsatzpaare können für Englisch→Chinesisch gefunden werden, und mehr als 5% Zielsatzpaare können für Englisch→Französisch gefunden werden.

Die experimentellen Ergebnisse zeigen, dass unser vorgeschlagenes Verfahren zum Finden von Sätzen, die Wörter enthalten, die erklärt werden sollen, positiv und robust ist. Dies verringert die Schwierigkeit beim Aufbau eines Trainingsdatensatzes und erleichtert auch das Training von Modellen in der Zukunft.

Contents

List of Figures

List of Tables

1. Introduction

1.1. Motivation

As one of the most classic fields of natural language processing (NLP), machine translation has a long history. In order to enable machines to achieve better performance in translation tasks, many outstanding methods and models are proposed. For example, in the early 1990s, IBM Model is proposed by Brown et al. [7, 8], which becomes one of the most classic models in machine translation.

Compared with statistical machine translation (SMT), neural machine translation (NMT) is much better. Especially as different neural network models are proposed, neural machine translation is now becoming the dominant approach in machine translation. Some new neural network models, such as Transformer [49], improve significantly the performance of machines in translation tasks.

Although many machine translation models can perform well, they cannot achieve the same high quality as human translation. There are still many problems in translation work that machine translation cannot solve. One of the most common problems is that the translation of certain words can cause incomprehension of the audience in the target language. Because some words are common in the source language, but not common in the target language. This leads to a fact that when the audience of the target language sees the translation of these words, they cannot understand the meaning of the translation.

A simple example is ***the Super Bowl***, the annual championship game of the National Football League in the United States. The Super Bowl is one of the most famous games in the United States, however, in some countries in Europe and Asia, only a few people who like American football know about it. Therefore, when the Super Bowl is translated into another language, such as German or Chinese, the audience in the target language will simply understand it as a kind of tableware according to the literal meaning of the translation.

The reason for this problem is that people who use different languages have different cultural backgrounds and living environments. Some words that are common in one language may be not common in another. Thus, when these words are translated into another language, how to eliminate the incomprehension of the target language audience on the translation of these words is a problem that cannot be ignored in machine translation. The work of this thesis is to develop a model that can judge which words will cause incomprehension to the target language audience during translation, and then give more understandable translation results.

1.2. Problem Statement and Research Questions

To solve the problem of how to eliminate the incomprehension of the target language audience during the translation, we can learn from human translation. In human translation, a simple solution to solve this problem is to add explanations when translating these words that are common in the source language but not common in the target language. With the help of this additional information, the incomprehension of these words by the target language audience can be well eliminated.

With the help of the human translation solution, the problem of how to eliminate the target language audience's incomprehension during the translation can be transformed into another more specific problem, that is, how to add additional information to the appropriate position of the translation. Therefore, the **problem statement** of this thesis now can be clearly given: "How can we model audiences' specific needs for additional information during translation?"

In order to train the model so that the model has high performance and accuracy, the first step is to establish a high quality training dataset. However, sentences containing words that need to be explained are extremely uncommon. The sparsity of the target sentence makes it difficult to establish the training dataset. Inspired by this, we formulate the first research question:

- **Research Question 1:** How to build a training dataset containing translation examples with explanation?

The next step is to train the model by using the data found in the first step so that the model can accurately predict which words need to be explained during the translation, and add additional information (i.e., the explanation) for the words that need to be explained.

1.3. Thesis Outline

The rest of the work is structured as follows. Chapter 2 introduces the necessary background information for our work. In addition, some related work is also in Chapter 2. Chapter 3 explains how we build the training dataset. Chapter 4 presents the experiments and the results. Finally, we give our conclusion and discussion in Chapter 5.

2. Background and Related Work

The concepts and foundations covered in this thesis are given in this chapter. For some of the techniques involved in this thesis, some related work is introduced. Section 2.1 introduces the parallel corpus, the underlying data required for all machine translation tasks. In Section 2.2, we introduced word alignment, which is a preprocessing step for text. Then we present named entity recognition in the Section 2.3, which is an important technique used in this thesis. The stemming algorithm is in Section 2.4. The evaluation metrics used in this thesis is given in Section 2.5. Finally, in Section 2.6 ,we explain how Wikipedia can be used in NLP tasks.

2.1. Parallel Corpus

Many natural language processing tasks are data-driven tasks. Especially for machine translation, the translation model needs to be trained by using text as input data. However, the task of machine translation is that the text in one natural language is translated to text in another natural language by the translation model. This means that it is not enough for the input data to contain text in only one language, so a parallel corpus is essential for the machine translation.

A parallel corpus contains text paired with its translation in another language [22]. If the task of machine translation is not limited to bilingual, then a multilingual parallel corpus is needed. Parallel corpora can be obtained in some ways. One way is that some corpora can be downloaded directly online, such as Europarl Parallel Corpus [21] and United Nations Parallel Corpus [51], both of them are available for free. Another way is by crawling the websites, a good source is the BBC News website, where news is translated into different languages. Table 2.1 gives some examples of English-German sentence pairs from Europarl Parallel Corpus [21].

English	German
Resumption of the session	Wiederaufnahme der Sitzungsperiode
Madam President, on a point of order.	Frau Präsidentin, zur Geschäftsordnung.
It is the case of Alexander Nikitin.	Das ist der Fall von Alexander Nikitin.

Table 2.1.: Some English-German sentences in Europarl Parallel Corpus

However, obtaining the corpus by crawling the website requires some technology. In addition, how to extract text from the results of website crawling and align the text is also a problem that must be considered. On the other hand, for some corpora that can be downloaded directly online, the topics are limited, such as Europarl Parallel Corpus

[21], which is politically related. Some corpora were created earlier, and the data size is not large. For example, in Europarl Parallel Corpus [21], there are only about 1.92 million sentence pairs in English and German.

Recently, some new corpora are released, such as ParaCrawl [4] and CCMatrix [46, 13], which overcome these shortcomings. CCMatrix [46, 13] initially contains 4.5 billion parallel sentences in 38 languages, of which 661 million are aligned to English. The latest version contains a total of 10.8 billion parallel sentences in 80 languages, of which 2.9 billion are aligned with English. Table 2.2 gives some language pairs and number of sentence pairs contained in CCMatrix [46, 13].

Language pairs	**Sentences Number (M : Million)**
French-English	328.6 M
German-English	247.5 M
Spanish-English	409.1 M

Table 2.2.: Some CCMatrix Statistics

Most parallel corpora available online, including CCMatrix [46, 13], can be downloaded directly from the website OPUS [48]. OPUS Project [48] collect freely accessible parallel corpora, until now it is still growing. Table 2.3 lists some corpora that can be downloaded directly from OPUS.

Corpus name
CCMatrix
ParaCrawl
MT560
WikiMatrix

Table 2.3.: Some available corpora in OPUS

2.2. Word Alignment

Word alignment is one of the necessary preprocessing for machine translation. The quality of word alignment also affects the quality of the machine translation results. Word alignment also plays a crucial role in this thesis. When building a training dataset, most of the work is done on the basis of word alignment.

The idea of word alignment was first proposed by Brown et al. in the IBM model [7, 8]. For each words in the target language sentence, a word alignment indicates the origin of it in the source language sentence [7]. Word alignment can be clearly shown on the figure with the help of alignment matrix [22]. Figure 2.1 from [22] shows an example of the word alignment between an English sentence and a German sentence.

In addition to one-to-one alignment, there are also other situations such as one-to-many alignment and Insertion. In the caption 2.1, the alignment between the English word

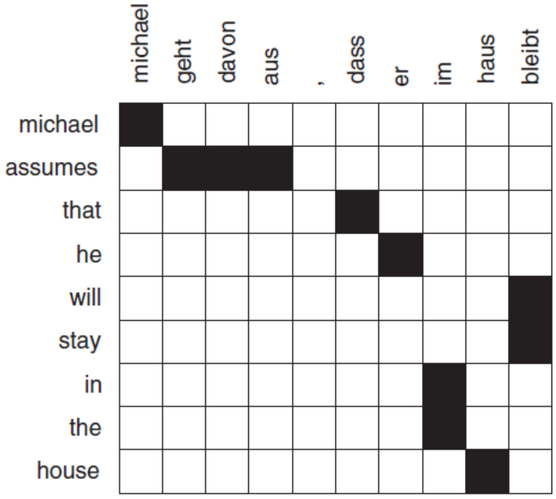


Figure 2.1.: An example of the word alignment [22]

that and the German word ***dass*** is a one-to-one alignment, while the alignment between the English word ***assumes*** and the three German words ***geht davon aus*** is a one-to-many alignment. The insertion means that for a word in the target sentence, there is no alignment for it in the source sentence, such as the ***comma*** in the German sentence, it has no corresponding word in the English sentence.

Different word alignment situations make understanding the concept of word alignment more complex and difficult. Therefore, Och and Ney give a more general definition of alignment [37]: An alignment between a source string $f_1^J = f_1, \dots, f_j, \dots, f_J$ and a target string $e_1^I = e_1, \dots, e_i, \dots, e_I$ is defined as the set $\mathcal{A}$ (2.1), it is a subset of the Cartesian product of the word positions.

$$\mathcal{A} \subseteq \{(j, i) : j = 1 \dots J; i = 1 \dots I\} \tag{2.1}$$

Many tools can extract word alignment, such as GIZA++ [37] or fast-align [12]. They are all based on the IBM model [8] in the statistical machine translation (SMT) category. At the same time, some other tools such as SimAlign [20] and awesome-align [11] use neural machine translation (NMT) to achieve the function of extracting word alignment. These tools are all efficient and perform well in extracting word alignments.

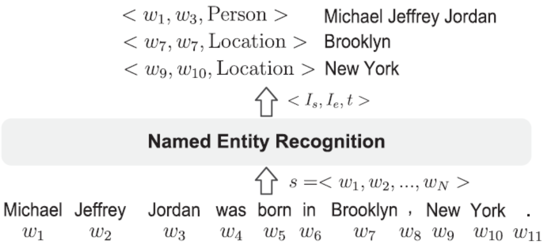
SimAlign [20] benefits from contextualized and static multilingual word embeddings. It chooses fastText [6] to train static monolingual embedding, meanwhile, it applys multilingual BERT model (mBERT) [10] to obtain contextualized monolingual embedding, then uses them to achieve word alignment. On the other side, awesome-align [11] also uses a contextualized word embedding model (mBERT) [10] to extract word alignments, but it fine-tunes on the pretrained mBERT model for better results.

Since awesome-align completely covers the language pairs experimented in this thesis, it is chosen as the tool for extracting word alignment in this thesis.

2.3. Named Entity Recognition

The term "Named Entity" first appeared in the Sixth Message Understanding Conference (MUC-6) [15]. The word "named" in the expression "named entity" restricts the entity to only those entities whose referents are represented by one or more rigid designators [31]. Rigid designator is defined by S. Kripke [23], both proper nouns and certain natural kind terms are considered rigid designators. Besides, Petasis et al. [39] also give a simpler definition of named entities: named entity is a proper noun used as the name of something or someone. The following words and phrases are examples of named entities: WHO, George Washington, German national football team. They are the names of institutions, famous people or places.

Named entity recognition (NER) is a technology that can identify named entities. It is one of essential and major tasks of natural language processing. The definition of NER is given by Li et al. [25]: NER is a method that can locate and classify named entities in text into predefined entity categories.



$< w_1, w_3, \text{Person} >$ Michael Jeffrey Jordan
$< w_7, w_7, \text{Location} >$ Brooklyn
$< w_9, w_{10}, \text{Location} >$ New York
$< I_s, I_e, t >$
Named Entity Recognition
$s =< w_1, w_2, ..., w_N >$
Michael w_1
Jeffrey w_2
Jordan w_3
was w_4
born w_5
in w_6
Brooklyn w_7
, w_8
New w_9
York w_{10}
. w_{11}

Figure 2.2.: An example of the NER process [25]

Figure 2.2 from [25] clearly indicates an example of the NER process. The input s is a sequence of tokens, consisting of N tokens. After NER, the output is several tuples with the form $\langle I_s, I_e, t \rangle$. Each tuple is a named entity recognized from input s. Here, I_s and I_e are the start and end indexes of named entities in the input, which is in the range $[1, N]$. The t is the entity type in the predefined entity categories.

There are many various approaches to achieve NER. The approaches to NER are divided into four main types [25]: (1) Rule-based approaches; (2) Unsupervised learning approaches; (3) Feature-based supervised learning approaches; (4) Deep-learning based approaches. Rule-based NER system depends on hand-crafted rules. Unsupervised learning NER system is based on unsupervised learning methods such as clustering [31]. The Feature-based supervised learning NER system applies some supervised learning methods to the NER system, such as support vector machines (SVM) [18] and conditional random fields (CRF) [24]. The Deep-learning based NER system benefits from some deep learning models, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU).

Some NLP tools and frameworks, such as Stanza [42] from Stanford and spaCy [19]from Explosion company, provide the NER function, so that NER can be easily integrated into various projects. Here is an example of running NER using spaCy:

- **The World Health Organization** (WHO) is a specialized agency of **the United Nations** responsible for international public health.

This sentence is selected randomly from Wikipedia. Phrases ***The World Health Organization*** and ***the United Nations*** are identified named entities. The complete output is as follows, ***ORG*** means that the entity type is an organization:

- **The World Health Organization**: $\langle 0, 29, ORG \rangle$

- **the United Nations**: $\langle 63, 81, ORG \rangle$

In this thesis, NER is an efficient and helpful tool. Through NER, it is possible to accurately locate and identify which words in the input sentence need to be explained during translation.

2.4. Stemming

Give some words: ***plays***, ***played*** and ***playing***. The common feature of these words is that they have the same root ***play***. In other words, these words are different forms of the root word ***play*** by adding different affixes (inflectional affix or derivational affix).

Stemming is a technique for treating words. It can remove derivational and inflectional suffixes from each word, so that all words with the same root can be reduced to a common form [26]. Stemming can eliminate the interference caused by different forms of a word, and it can add convenience to some processing and analysis steps in fields such as information retrieval.

The first stemming algorithm appeared in 1968 and was proposed by Lovins [26]. Lovins implemented his stemming algorithm using a list of common endings (for example, ***-fully***, ***-ing***, etc.). Each ending corresponds to a predetermined condition. Stemming a word can be accomplished with the help of these conditions. After Lovins' algorithm, more stemming algorithms are proposed, such as porter stemming [40], snowball stemming [41] and lancaster stemming [38].

Porter stemming [40] is one of the most classic algorithms for stemming, and it is still a popular stemmer until now. In the porter stemming algorithm, a word is regarded as a combination of vowels and consonants. Here, vowels refer to the letters ***A***, ***E***, ***I***, ***O***, ***U***, and the letter ***Y*** after the consonants. Consonants refer to the letters except ***A***, ***E***, ***I***, ***O***, ***U*** and ***Y*** after the vowel. In addition, some rules are also given. The rules are related to the form of words (the combination of vowels and consonants). By iteratively applying the rules to a word, the stem of the word is finally obtained.

The snowball stemming algorithm [41] is the successor of the porter algorithm, and the effect of the snowball stemmer is also better than that of the porter stemmer. So the snowball stemmer is also the main stemming algorithm used in this thesis. In this thesis, there are some operations such as comparison between words, but these operations will cause a problem, that is, some words have different forms, such as plural forms of nouns. this problem brings additional difficulties to the operations. With the help of word stemming algorithm, these difficulties can be easily overcome.

2.5. Evaluation Metric

In machine learning, how to evaluate the quality of the trained model or the implemented algorithm is one of the necessary steps. Therefore, in the evaluation phase, some evaluation metrics are needed as the criterion for evaluating the quality of the model or algorithm. Commonly used evaluation metrics include precision, recall, and F1-score.

First, for a typical binary classification problem, the confusion matrix in table 2.4 can be obtained according to the actual class and the predicted class of a sample. The confusion matrix shows the possible outcomes of each sample. There are four results:

- True positive ($\boldsymbol{TP}$): The sampling is actually a positive class, and the model predicts sampling is also a positive class.
- False Negative ($\boldsymbol{FN}$): The sampling is actually a positive class, but the model predicts sampling is a negative class.
- False Positive ($\boldsymbol{FP}$): The sampling is actually a negative class, but the model predicts sampling is a positive class.
- True Negative($\boldsymbol{TN}$): The sampling is actually a negative class, and the model predicts sampling is also a negative class.

If now there are a total of $\boldsymbol{N}$ samples, $\boldsymbol{TP}$, $\boldsymbol{FP}$, $\boldsymbol{TN}$ and $\boldsymbol{FN}$ can be simply used to represent the sample numbers of the respective results. Based on these sample numbers, the definitions for Precision, Recall and F1-score can be given.

		Predicted Result	
		Positive	Negative
Actual Result	Positive	True positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative(TN)

Table 2.4.: Confusion Matrix

2.5.1. Precision

Precision is defined as the equation 2.2. Precision gives how many samples are actually positive among all the samples predicted to be positive, i.e., the proportion of samples that are actually positive in all samples that are predicted to be positive.

$$Precision = \frac{TP}{TP + FP} \tag{2.2}$$

2.5.2. Recall

Recall is defined as the equation 2.3. Recall gives how many samples are correctly predicted to be positive among all actually positive samples, i.e., the proportion of samples that are predicted to be positive among all samples that are actually positive.

$$Recall = \frac{TP}{TP + FN} \tag{2.3}$$

2.5.3. F1-score

F1-score is defined as the equation 2.4. F1-score is the harmonic mean of precision and recall.

It is contradictory to get precision and recall as high as possible at the same time. If we want get the model with a higher recall, we can just let the model predict as many samples as possible, but at the same time the model is more likely to predict errors, then the precision will be lower. If we want to get a model with a higher precision, we can just let the model predict its most certain samples, then the precision will be high, but the recall will be lower.

Therefore, it is necessary to use F1-score as an evaluation metric, and using F1-score is also better than using precision and recall. F1-score is also the main metric used in this thesis to evaluate the performance of the model.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{2.4}$$

2.6. Wikipedia

Wikipedia is a multilingual online encyclopedia operated by the Wikimedia Foundation. It was created and launched by Jimmy Wales and Larry Sanger in 2001. Wikipedia is an open collaborative project, in other words, everyone can create and maintain entries and articles on Wikipedia. There are now more than 250 language articles on Wikipedia, and as of early 2021, there are 55 million Wikipedia entries in all languages.

2.6.1. Structure of Wikipedia

Wikipedia also has the structural features of a traditional paper encyclopedia, including articles, internal cross-references to other articles, and external references to the academic literature [29]. Besides, some new structural features are also added to Wikipedia. Figure 2.3 shows a typical Wikipedia page. This Wikipedia page is used to introduce Wikipedia.

This page clearly shows the structure of Wikipedia. First and foremost is the article. Articles are texts that contain information about the concepts being introduced. For example, the article in the figure 2.3 is about Wikipedia, starting with "Wikipedia[note 3] is a multilingual free online encyclopedia written and maintained by a community of volunteers, ...". An article has a title. The bold word "Wikipedia" at the top of the figure is the title. The title is also included on the URL of the Wikipedia article.

Another important component of Wikipedia is language links. In the upper right corner of the figure, the word "languages" is the language link. Language links are used to switch to other language versions of the same article. In addition to these components, Wikipedia also has components such as Disambiguation pages and Hyperlinks.

2.6.2. Usage of Wikipedia

Wikipedia has a large number of multilingual articles, and these articles cover a wide range of topics. These properties make Wikipedia an extremely useful language resource

Wikipedia

From Wikipedia, the free encyclopedia

This article is about the online encyclopedia. For Wikipedia's home page, see Main Page. For the primary English-language Wikipedia, see English Wikipedia. For other uses, see Wikipedia (disambiguation).
"The Free Encyclopedia" redirects here. For the concept of a free encyclopedia, see Encyclopedia § Free encyclopedias.

Wikipedia[note 3] is a multilingual free online encyclopedia written and maintained by a community of volunteers, known as Wikipedians, through open collaboration and using a wiki-based editing system called MediaWiki. Wikipedia is the largest and most-read reference work in history.[3] It is consistently one of the 10 most popular websites ranked by Similarweb and formerly Alexa; as of 2022, Wikipedia was ranked the 5th most popular site in the world.[4] It is hosted by the Wikimedia Foundation, an American non-profit organization funded mainly through donations.[5]

Wikipedia was launched by Jimmy Wales and Larry Sanger on January 15, 2001. Sanger coined its name as a blend of *wiki* and *encyclopedia*.[6][7] Wales was influenced by the "spontaneous order" ideas associated with Friedrich Hayek and the Austrian School of economics after being exposed to these ideas by the libertarian economist Mark Thornton.[8] Initially available only in English, versions in

Wikipedia

The logo of Wikipedia, a globe featuring glyphs from various writing systems

Figure 2.3.: A Wikipedia example

in natural language processing. Wikipedia is now widely used in various directions for natural language processing.

2.6.2.1. Using Wikipedia to create a corpus

Because Wikipedia is a huge multilingual language resource, it is very suitable as a corpus to train high-quality language models.

Margaretha and Lüngen [28] built a German monolingual corpus using Wikipedia articles and talk pages. They converted the Wikipedia content into a corpus in XML format, and then converted the XML corpus into I5 format so that the obtained Wikipedia corpus can be integrated into the German Reference Corpus.

Denoyer and Gallinari [9] also used Wikipedia to create an XML corpus. The corpus they created contains files in 8 languages. In addition to the common English, German and French, there are also Arabic and Japanese among the 8 languages.

There are also many more multilingual corpora built on the basis of Wikipedia. For example, Reese et al. [43] proposed an automatically sense tagged corpus that support three languages of Catalan, Spanish, English. In addition to this trilingual Corpus, WikiAtomicEdits built by Faruqui et al. [14] is also a multilingual corpus containing 8 languages. Unlike the above corpus, WikiAtomicEdits is created using only Wikipedia edit history.

The use of Wikipedia in creating a corpus reminds us that perhaps we can also use Wikipedia to find and determine which words need to be explained in our work.

2.6.2.2. Use Wikipedia to improve the effect of NER

In addition to being used to create a corpus, Wikipedia is also very useful and helpful in named entity recognition (NER).

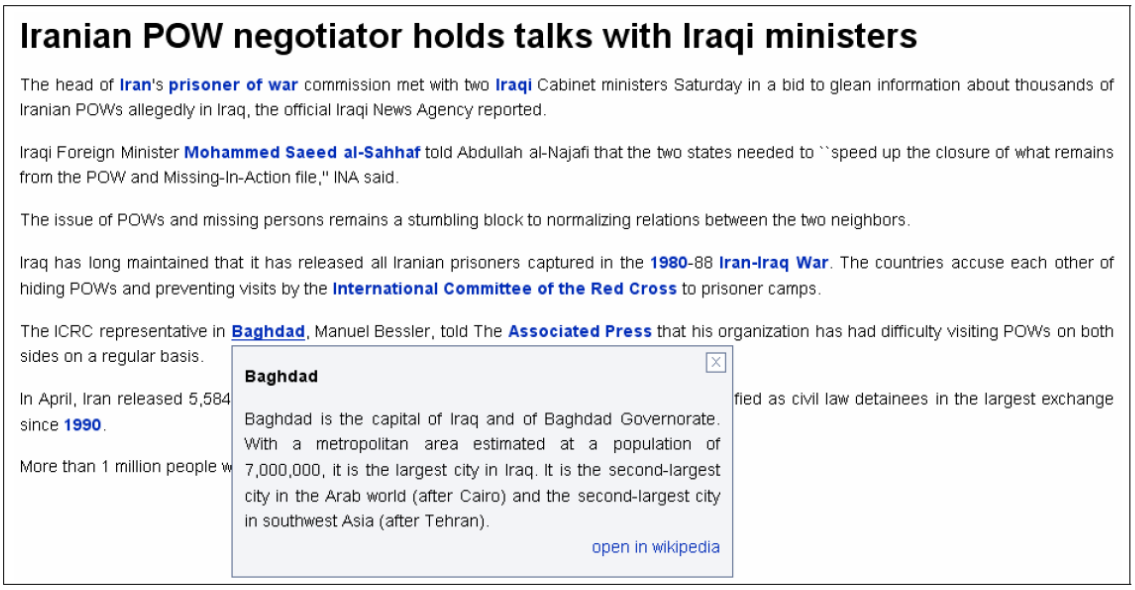
Nothman et al. [34] proposed a method to build named entity data with the help of Wikipedia. This data can be used to train NER models. The idea of their method is to extract named entities from links between Wikipedia articles. Based on this method, Nothman et al. proposed a new method of using Wikipedia to create NER training data [36]. In the new method, after extracting named entities through links between Wikipedia articles, additional links and Wikipedia corpus are used for fine-tuning to achieve higher accuracy. The NER training data built with the new method is multilingual, supporting nine languages.

Nothman et al. also compared and evaluated the NER training data created by using Wikipedia with other existing NER training data [35]. The results show that the Wikipedia training data has better performance than other existing training data, such as MUC, CoNLL, etc.

The work of Nothman et al. also brought a new idea to our work, that is, we can combine NER and Wikipedia to identify words and phrases that need to be explained.

2.6.2.3. Wikification

The wikification task aims at automatically recognizing concept mentions appearing in a text document and link them to concept references in Wikipedia [45]. Figure 2.4 from [30] shows an example of wikification. The word **Baghdad** in the figure is linked to its corresponding Wikipedia article.



Iranian POW negotiator holds talks with Iraqi ministers

The head of **Iran**'s **prisoner of war** commission met with two **Iraqi** Cabinet ministers Saturday in a bid to glean information about thousands of Iranian POWs allegedly in Iraq, the official Iraqi News Agency reported.

Iraqi Foreign Minister **Mohammed Saeed al-Sahhaf** told Abdullah al-Najafi that the two states needed to ``speed up the closure of what remains from the POW and Missing-In-Action file,'' INA said.

The issue of POWs and missing persons remains a stumbling block to normalizing relations between the two neighbors.

Iraq has long maintained that it has released all Iranian prisoners captured in the **1980**-88 **Iran-Iraq War**. The countries accuse each other of hiding POWs and preventing visits by the **International Committee of the Red Cross** to prisoner camps.

The ICRC representative in **Baghdad**, Manuel Bessler, told The **Associated Press** that his organization has had difficulty visiting POWs on both sides on a regular basis.

In April, Iran released 5,584 fied as civil law detainees in the largest exchange since **1990**.

More than 1 million people w

Baghdad

Baghdad is the capital of Iraq and of Baghdad Governorate. With a metropolitan area estimated at a population of 7,000,000, it is the largest city in Iraq. It is the second-largest city in the Arab world (after Cairo) and the second-largest city in southwest Asia (after Tehran).

Figure 2.4.: A wikification example [30]

The purpose of our work is to find a suitable model that can predict which words and phrases need to be explained. On the basis of our work, a task that can be continued in the future is to add explanations to words and phrases that require explanations. The wikification provides a feasible solution for the future work.

Shnayderman et al. proposed RedW [47], an efficient end-to-end wikification solution based on Wikipedia redirects. RedW uses Wikipedia redirects to realize the function of linking an entity to the corresponding Wikipedia page. Another tool that can be used for wikification is BLINK [50]. BLINK uses BERT transformer to complete the task of entity linking and achieves state-of-the-art results.

3. Terminologie Explanation

This chapter describes how to build a training dataset containing sentences with words that need to be explained. In Section 3.1 we present our proposed method for identifying and finding sentences with words that need to be explained.

3.1. Build Dataset

In order to accurately predict the words that need to be explained when doing machine translation tasks, it is necessary to build a dataset for training and evaluation. In this thesis, the required dataset contains sentences with words that require additional explanation.

However, there are several problems when building the dataset. The first problem is which sentences contain words that require additional explanation. More precisely, how to distinguish sentences containing words that need to be explained from other sentences. If we can't find a good way to distinguish the target sentences from other sentences, it will bring a huge manual workload to build the training dataset.

In addition, in order for the trained model to have high quality, the training dataset must contain a sufficiently large number of target sentences. However, target sentences are not common, which makes how to find a sufficient number of target sentences become another problem that must be solved.

Both of these problems require us to propose a sufficiently ideal method, which can find as many target sentences as possible while reducing the manual workload as much as possible. It also makes building a training dataset a difficult challenge.

3.1.1. Definition of target sentence pair

Bilingual sentence pairs are used to find target sentences containing words that need to be explained. Therefore, before building a dataset, one thing that must be clarified is what kind of sentence pair is our target sentence pair, more specifically, what is the sentence pair with explanation.

Here are a few examples of sentence pairs (English-German) found that contain an explanation for a word:

1. **En**: He is replaced by Mike **Ahern** , the only premier never to contest an election as premier .
   De: Er wird von Mike **Ahern** (Michael Ahern (australischer Politiker)) , der einzige Premier ersetzt , um um eine Wahl als Premier nie zu kämpfen .

2. **En**: China has been accused of artificial devaluation of its currency to prop up exports , while the **ECB** 's policy has had an opposite effect for the economy of

France and some South European countries : the euro has become too strong ; whereas for Germany 's it has become too weak .
De: China wurde der künstlichen Wertminderung seiner eigenen Währung zum Abfangen von Exportartikeln beschuldigt , während die Police der **EZB** (Europäische Zentralbank) eine gegensätzliche Wirkung auf die Wirtschaft Frankreichs und einige südeuropäische Länder hatte : der Euro ist inzwischen zu stark ; indessen ist er für die deutsche Wirtschaft zu schwach geworden .

3. **En**: The European Union is funding the MOON project (multimodal optical diagnostics for age-related diseases of the eye and central nervous system) as part of the Horizon2020 program with around 3.7 million euros as an initiative of the Photonics **Public-Private-Partnership** Photonics21 .
De: Die Europäische Union fördert das Projekt „ Moon " (multimodale optische Diagnostik für altersbedingte Erkrankungen des Auges und des Zentralnervensystems) im Rahmen des Horizon2020-Programms mit rund 3,7 Millionen Euro als Initiative der Photonics **Public-Private-Partnership** (Öffentlich-Private-Partnerschaft) Photonics21.

In the first example, the red part *Mike Ahern* is the name of a person, and in the target language sentence, the blue part is the explanation of this person, that is, *Mike Ahern is an Australian politician*. In the second example, the word *ECB* in the red part is an abbreviation. In the sentence of the target language, the blue part gives the full meaning of the abbreviation, that is, the *European Central Bank*.

These two examples are very interesting, both can be considered as sentence pairs with the explanation, and the explanation part can be clearly identified. But at the same time, there are other examples where the explanation part is not clearly identifiable. In the third example, the blue part of the sentence in the target language is strictly a translation of the word in the red part. But considering that for some words, it may be better to keep the original words when translating them. In this case, the additional translation of these words can also be seen as an explanation.

Therefore, in this thesis, there are two categories of target sentence pairs with explanations. One is a sentence pair with extremely clear explanation, the explanation part may be an explanation for a person, a thing, or an abbreviation. The other category is sentence pairs whose explanation part cannot be clearly distinguished, and the explanation part is the translation of the explained words in the target language.

3.1.2. How to find sentence pairs with an explanation for a word

To build the dataset, some parallel corpus containing parallel sentences in different language pairs can be used to find the sentences needed. However, the number of sentence pairs contained in a parallel corpus is extremely large, and the target sentence pairs with explanations are very uncommon. The sparsity of target sentence pairs makes it impossible to precisely and easily distinguish target sentence pairs from other sentence pairs. So how to find the needed sentence pairs among tens of millions or even hundreds of millions of sentence pairs is a difficult problem. Therefore, it is necessary to summarize the characteristics of the sentence pair that contains an explanation for a word, and then filter out the needed sentence pairs from the corpus based on the summarized characteristics.

Here are a few more examples of sentence pairs (English-German) found that contain an explanation for a word:

1. **En**: John **Bunyan** said , “ He who runs from God in the morning will scarcely find Him the rest of the day . ”
   De: John **Bunyan** , der Autor der bekannten Pilgerreise , hat einmal gesagt : „ Wer morgens vor Gott wegläuft , wird Ihn den Rest des Tages kaum noch finden . “

2. **En**: You can say things like **Canadian-Americans** – like Jim Carrey who has dual citizenship .
   De: Man kann Sachen sagen wie **Canadian-Americans** (Amerikaner kanadischer Herkunft) - wie Jim Carrey , der eine doppelte Staatsbürgerschaft besitzt .

3. **En**: However , when you get infected with the likes of **BS2005** , it might take a while before you even realize that something is wrong with your system .
   De: Jedoch , wenn Sie infiziert mit den gleichen von **BS2005** (auch bekannt als BS2005 Virus) , es könnte eine Weile dauern , bevor Sie überhaupt erkennen , dass etwas falsch mit Ihrem system .

Because the words being explained or some words in the phrase being explained are generally common in the source language but not common in the target language, the first characteristic is: **The word being explained or the word in the phrase being explained is rare in the target language**. In the three examples above, the red bold words, namely *Bunyan*, *Canadian-Americans* and *BS2005*, are all rare in the German.

In the three examples, the blue parts of the German sentences are the explanations for the red words and phrases. For example, in the first example, the blue part is used to explain the red phrase John Bunyan is a writer. Comparing the English sentence with the German sentence, it can be found that the part of explanation only exists in the German sentence. If the explanation is deleted, there is no loss in the translation result. Therefore, it is also an important characteristic that **the explanation is a redundant part of the sentence in the target language**.

Besides, it can be observed that in the three examples, the blue part is always after the red part, and it is immediately followed by the red part. This means that **the explanation follows the word or phrase being explained**, which is also a characteristic.

If only the explanation part is considered, it can be observed that in addition to the text, **the explanation part also contains some punctuations**. These punctuations can be used to identify the explanation. In the first example the comma is used, in the other two examples parentheses are used. Meanwhile, **for the text part in the explanation, it contains words that are different from the word or phrase being explained**. Especially in the first and second examples, the explanation uses words that are completely different from the words and phrases being explained.

On the other hand, if the words and phrases being explained are considered, it can be found that **the word or phrase being explained is more likely to be a proper noun**. For example, *John Bunyan* in the first example is the name of a person, and *Canadian-Americans* in the second example is the name of a nation or an ethnic group. Furthermore, if Wikipedia is used to search for these two words and phrases, the corresponding articles will be found. This implies that **the information about words or phrases that need to be explained can be found using Wikipedia**.

Finally, we summarize and collect these found characteristics. These characteristics are listed below:

1. The word being explained or the word in the phrase being explained is rare in the target language
2. The explanation is a redundant part of the sentence in the target language
3. The explanation follows the word or phrase being explained
4. The explanation contains punctuation
5. Words that differ from the word or phrase being explained are also included in the explanation
6. The word or phrase being explained is more likely to be a proper noun, such as the name of a person, thing, institution, or place
7. Information about words or phrases that need to be explained can be found using Wikipedia

These characteristics can be considered in two aspects: the first aspect is the characteristics of the word being explained, and the second aspect is the characteristics of the explanation part.

Based on these characteristics, in this thesis, a heuristic method for searching target sentence pairs is proposed. Considering the sparsity of the target sentence pairs that need to be found, the goal of this method is to find as many target sentence pairs as possible while minimizing the number of non-target sentence pairs.

This heuristic method is divided into three processes. The first process is to identify and find candidates that may contain the target sentence pair by using the characteristics of the sentence pair with explanation (Characteristics 1-5). The second process is to use the NER model to identify target sentence pairs based on the obtained candidates (Characteristic 6). The last process is to exploit Wikipedia to more accurately identify target sentence pairs (Characteristic 7).

3.1.2.1. Preprocessing

For machine translation tasks, preprocessing is the first step in subsequent work. In this process, the word tokenization of the sentence and the word alignment extraction of the sentence pair will be completed.

Word tokenization can be done with several tools, such as NLTK [5], spaCy [19], and Stanza [42]. Although these tools all support multilingual processing, due to the difference between Chinese and other languages, some of them cannot perfectly support Chinese word tokenization. Therefore, some other tools specially used for Chinese word tokenization can be selected. For example, pkuseg [27], jieba and hanlp [17] can all perform Chinese word tokenization very well and get more accurate results.

There are also many tools that can be used to extract word alignment, such as fast-align [12] and pialign [32, 33]. But on the one hand, they all appeared very early, and have not been updated for a long time. On the other hand, for some language pairs, these tools cannot perform very well. Meanwhile, a new tool for extracting word alignment called awesome-align [11] is published. This tool can well support all the language pairs required for the experiment in this thesis, and for each language pair, the performance of extracting word alignment is excellent. So, this tool will be used to extract word alignment in this thesis.

It must be noted that for Chinese, there is another step in the preprocessing before starting word tokenization and alignment. Chinese text is divided into Simplified Chinese and Traditional Chinese, which are completely different. For unification, it is necessary to convert Traditional Chinese to Simplified Chinese. This step can be done simply with the tool OpenCC [16].

3.1.2.2. Find sentence pairs candidates that may contain explanations

After obtaining the word alignment of the sentence pairs, the sentence pair candidates that may contain explanations can be found by using the previously summarized characteristics of the target examples.

This process also includes several steps. The first thing to determine is which words in a sentence may need to be explained. Intuitively, when translating, if a word is rare in the target language, it is more likely to be explained than other words. This is also the first characteristic. In order to decide which words are rare, the word count within a certain range can be used. A word can be considered rare if its count is below a certain threshold. For the purpose of finding as many rare words as possible, the word count in all Wikipedia articles is used to check whether a word is rare. However, if only the uncommon words in the target language are considered, it is found that many non-candidates are introduced in the experiment, so not only uncommon words in the target language but also uncommon words in the source language must be considered.

The next thing to determine is which sentence pairs may contain explanations. According to the second and third characteristics summarized, it is easy to find sentence pairs that may contain explanations. With the help of the word alignment of the determined rare word and the word following it, the corresponding words in the target language sentence and their position in the sentence can be found. If there is a redundant part between the corresponding words in the target language sentence, it can be considered that there may be an explanation for the rare word in the redundant part. What needs to be decided here is the length of the redundant part. If the length is too long, many possible examples will be missed, but if the length is too short, many non-candidates will be added.

The final step is to determine whether the redundant part contains an explanation. With the help of the fourth and fifth characteristics summarized, specifically, the characteristics of the explanation part, it is possible to find the sentence pairs that actually contain explanations. The explanation part is often accompanied by punctuation marks, such as commas and parentheses. This means that it is possible to determine whether a redundant part contains an explanation by checking for possible punctuation in the redundant part. In addition, the explanation should contain other words besides the explained word, so it can be judged whether there is an explanation in the redundant part by checking the words in the redundant part and their word alignment. If the redundant part also contains words other than the explained word, and none of the words in the redundant part have a word alignment, the redundant part can be considered as likely to contain the true explanation.

Figure 3.1 shows an ideal candidate sentence pair. Given the source language sentence $\boldsymbol{a}$ and the corresponding target language translation sentence $\boldsymbol{b}$. The length of sentence $\boldsymbol{a}$ is $\boldsymbol{N}$, which means it is composed of $\boldsymbol{N}$ tokens. Similarly, the length of sentence $\boldsymbol{b}$ is $\boldsymbol{M}$. In sentence $\boldsymbol{a}$, the kth token $\boldsymbol{token_k}$ is an uncommon word, meanwhile, The mth token $\boldsymbol{token_m}$ aligned with $\boldsymbol{token_k}$ in sentence $\boldsymbol{b}$ is also a rare word. The next token $\boldsymbol{token_{k+1}}$ of token $\boldsymbol{k}$ in sentence $\boldsymbol{a}$ is aligned with token $\boldsymbol{token_{m+n+1}}$ in sentence $\boldsymbol{b}$. And in sentence $\boldsymbol{b}$ $\boldsymbol{token_{m+n+1}}$ is not the next token of $\boldsymbol{token_m}$, which means that there is a redundant part of length $\boldsymbol{n}$ after $\boldsymbol{token_m}$ in sentence $\boldsymbol{b}$. In the redundant part from $\boldsymbol{token_{m+1}}$ to $\boldsymbol{token_{m+n}}$, there are punctuation, such as $\boldsymbol{token_{m+1}}$ is likely to be a comma, or a parenthesis. All tokens in the redundant part should have no word alignment results, and should contain other words except $\boldsymbol{token_m}$.

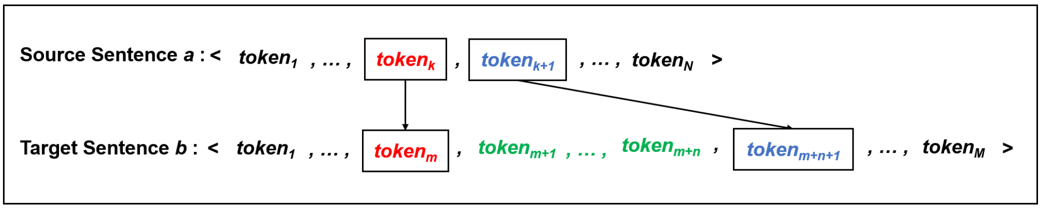

Source Sentence a : < token1 , … , tokenk , tokenk+1 , … , tokenN >
Target Sentence b : < token1 , … , tokenm , tokenm+1 , … , tokenm+n , tokenm+n+1 , … , tokenM >

Figure 3.1.: Candidate sentence pair

3.1.2.3. Using NER

The sixth characteristic summarized is that the words and phrases being explained are more likely to be proper nouns. This characteristic provides another way to determine candidate sentence pairs. If the proper nouns in a sentence, such as person names, place names or organization names, can be identified and located, then the range of candidate sentences can be narrowed down better. Named entity recognition (NER) is an effective tool for identifying proper nouns in a sentence. NER can identify the proper nouns that exist in a sentence, that is, named entities, and can also give the location of each named entity.

So based on the sixth characteristic summarized, NER should further identify possible candidates while also reducing the number of non-candidates. Many natural language processing tools support NER, such as spaCy [19], Stanza [42] and flair [1]. But not all libraries support all the languages involved in the experiments, for example, flair [1] does not support Chinese. So for different languages, different NER tools are tried and compared, and finally, the best tool for each language is selected based on performance.

After using NER to recognize all named entities, the words or phrases that need to be explained are likely to be in these named entities. Besides, the word being explained is either itself a named entity, or it is part of a named entity. So the candidate sentence pairs can be further determined by comparing the named entities with the previously confirmed words that may be explained.

In the experiments, another problem was found, that is, the redundant part of the sentence in the target language does not contain an explanation of the word or phrase, but the word or phrase itself. This problem can also be easily solved by using NER. After using NER, as long as the identified named entity is compared with the entity in the redundant part, the sentence pair that contains only the named entity itself in the redundant part can be identified.

3.1.2.4. Using Wikipedia

All the named entities in a sentence can be recognized after NER. Based on named entities, another method that can further to identify target sentence pairs with explanations is using Wikipedia.

The titles of Wikipedia articles can be used to determine target sentence pairs. If a source language named entity is a title of a Wikipedia article, then it is likely a candidate that needs to be explained. However, this only considers the aspect of the source language, if the consideration for the target language is added, candidates can be further identified. So if a source language named entity is the title of a Wikipedia article, and the corresponding target language named entity is not the title of a Wikipedia article, then the named entity is more likely to be a good candidate that needs to be explained.

On the other hand, in addition to the title of the Wikipedia article can be used to determine candidates, the Wikipedia article itself can also be used to determine candidates. If both the named entity in the source language and the corresponding named entity in the target language are the titles of Wikipedia articles, then the articles corresponding to the titles can be compared. More precisely, candidates can be determined by comparing the size of Wikipedia articles. If the size of the Wikipedia article in the source language is larger than the size of Wikipedia article in the target language, then the title of the source language Wikipedia article might be a good candidate that needs to be explained.

Figure 3.2 shows the process of using Wikipedia to identify candidates.

All Wikipedia data can be downloaded directly from the Wikimedia website, including Wikipedia articles and titles. Some tools can extract Wikipedia articles and titles directly from downloaded files. For example, gensim [44] and wikiextractor [3] are tools used to extract Wikipedia articles. In addition, wikipedia-parallel-titles [2] can extract parallel article titles across languages in Wikipedia.

3.1.2.5. Summary

Finally, we can give the detailed heuristic method. All steps are covered in the Table 3.1. For different language pairs, these steps can be modified according to the experimental results.

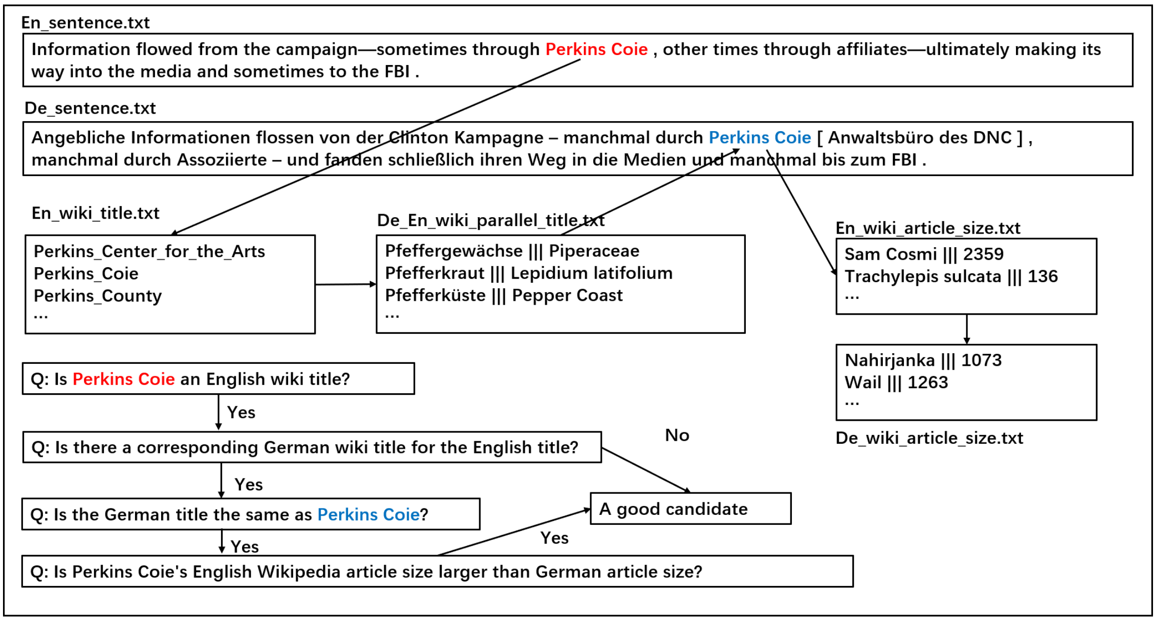

En_sentence.txt
Information flowed from the campaign—sometimes through Perkins Coie , other times through affiliates—ultimately making its way into the media and sometimes to the FBI .
De_sentence.txt
Angebliche Informationen flossen von der Clinton Kampagne – manchmal durch Perkins Coie [Anwaltsbüro des DNC] , manchmal durch Assoziierte – und fanden schließlich ihren Weg in die Medien und manchmal bis zum FBI .
En_wiki_title.txt
Perkins_Center_for_the_Arts
Perkins_Coie
Perkins_County
…
De_En_wiki_parallel_title.txt
Pfeffergewächse ||| Piperaceae
Pfefferkraut ||| Lepidium latifolium
Pfefferküste ||| Pepper Coast
…
En_wiki_article_size.txt
Sam Cosmi ||| 2359
Trachylepis sulcata ||| 136
…
Nahirjanka ||| 1073
Wail ||| 1263
…
De_wiki_article_size.txt
Q: Is Perkins Coie an English wiki title?
Yes
Q: Is there a corresponding German wiki title for the English title?
No
Yes
Q: Is the German title the same as Perkins Coie?
A good candidate
Yes
Yes
Q: Is Perkins Coie's English Wikipedia article size larger than German article size?

Figure 3.2.: An example of identifying candidates using Wikipedia

Steps 0, 1 and 2 are preprocessing. Steps 3-9 are based on the summarized characteristics to find candidate sentence pairs that may contain explanations. Steps 10 and 11 is using NER. Steps 12-14 are further to find candidates based on NER results with the help of Wikipedia.

Besides, changes in hyperparameters in some steps can affect the performance of the method, so these hyperparameters need to be determined. Hyperparameters that need to be determined are in Table 3.2. The details of parameter determination are introduced in Section 3.1.3.

3.1.3. Determination of parameters and some step details

In our proposed method for finding sentence pairs with explanations, if the parameters in some steps are changed, the final result will be affected, therefore, these parameters should be determined to make the method achieve better results. In this section, we will discuss the influence of these parameters on the experimental results and try to find an optimal combination. In addition to this, the details of some steps also need to be determined.

Some parameters can be easily determined. For example, punctuation contained in the redundant part of sentences in the target language that may contain explanations. The determination of these parameters will be discussed first. The determination of some other parameters is complicated, such as the threshold for distinguishing whether a word is common or uncommon. The determination of these parameters requires some experiments, so the determination of these parameters will be discussed in detail later.

0	Chinese Convert (only for Chinese)
1	Sentence tokenization
2	Extract word alignments of sentence pairs
3	Check if a word in a source sentence is rare in the Wikipedia articles
4	Check if the word has only one to one alignment
5	Check if there is a redundant part in the target sentence
6	Check if the words in the redundant part have alignment
7	Check if the corresponding target word is rare in the Wikipedia articles
8	Check if there are punctuations in the redundant part
9	Check if there are other words in the redundant part except the explained word
10	Named Entity Recognition (NER)
11	Check for duplicates of named entities
12	Check the Wikipedia article title in the source language
13	Check the Wikipedia article title in the target language
14	Check the Wikipedia article size
15	Manually select sentence pairs with explanations

Table 3.1.: Steps to finding a candidate

1. Parameter	Step 3: Source language word count threshold
2. Parameter	Step 5: Redundant part length
3. Parameter	Step 7: Target language word count threshold
4. Parameter	Step 8: Punctuation type

Table 3.2.: List of hyperparameters to be determined

3.1.3.1. Determination of punctuation

In the target language sentence, the redundant part with explanations contains punctuation, this characteristic is observed and summarized from examples. So which punctuation is contained in the explanation part can also be determined by example. Review the few examples mentioned in the section 3.1:

1. **En**: John **Bunyan** said , “ He who runs from God in the morning will scarcely find Him the rest of the day . ”
   De: John **Bunyan** , der Autor der bekannten Pilgerreise , hat einmal gesagt : „ Wer morgens vor Gott wegläuft , wird Ihn den Rest des Tages kaum noch finden . “

2. **En**: You can say things like **Canadian-Americans** – like Jim Carrey who has dual citizenship .
   De: Man kann Sachen sagen wie **Canadian-Americans** (Amerikaner kanadischer Herkunft) - wie Jim Carrey , der eine doppelte Staatsbürgerschaft besitzt .

From these two examples, it can be observed that the punctuation generally contained in the explanation part are commas and parentheses. Besides that, there are some other examples:

1. **En**: Josh missed seven weeks with an **MCL** .
   De: Josh verpasste sieben Wochen mit einem **MCL** [Verstauchung] .

2. **En**:The tool also assesses the CCF of course (= Common Cause **Failure**) .
   De: Natürlich bewertet das Tool auch die CCF (= Common Cause **Failure** : Ausfall in Folge gemeinsamer Ursache) .

These two new examples show that, in addition to commas and parentheses, other punctuation such as brackets and colons can also appear in the explanation part. Therefore, in order to find as many sentence pairs with explanations as possible, in addition to the above punctuation, more punctuation will be checked. The punctuation considered to be checked in the experiment are given in Table 3.3. These punctuation marks are suitable for checking sentences in English, German, and French.

Name	**Punctuation**
Parentheses	()
Square brackets	[]
Curly brackets	{}
Angle brackets	<>
Comma	,
Colon	:
Dash	–
Equals sign	=
Quotation mark	”

Table 3.3.: Checked punctuation for En, De and Fr

However, the punctuation marks used in Chinese are not the same as those used in English, German and French. The punctuation used in Chinese is full-width, while the punctuation used in English, German and French is half-width. Therefore, for Chinese sentences, the full-width version of the punctuation marks in Table 3.3 will be checked in the experiment. In addition, the proper name mark will also be checked.

3.1.3.2. Determination of the length of the explanation part

Except for punctuation, the length of redundant part containing explanation in the target language sentence is also an important parameter that can affect the experimental results.

A common way to determine the length of an explanation part is to set a minimum length and a maximum length. However, determining a maximum length is difficult because the length of the explanation is not fixed. Therefore, only the minimum length is set in our experiments, if the length of the redundant part in the target language sentence is greater than or equal to the minimum length, the sentence will be considered as possibly containing an explanation.

Consider the following example:

1. **En**: Foundations and **NGOs** like Hivos and Access Now
   De: Stiftungen und **NGOs** (Nichtregierungsorganisationen) wie Hivos und Access Now

2. **En**: The European Aviation Safety Agency (**EASA**) has also approved AerSafe on Airbus 321 aircraft (10065226) as a Flammability Reduction System (FRS) . **De**: Die European Aviation Safety Agency (**EASA** , Europäische Luftfahrtaufsichtsbehörde) hat ebenfalls AerSafe in Airbus 321 Flugzeugen (10065226) als System zur Reduzierung der Entflammbarkeit (Flammability Reduction System , FRS) zugelassen .

It can be observed from these two examples that the explanation part can consist of parentheses and a word, or it can also consist of a comma and two words. Their explanation part length is 3, so in our experiment, the minimum length will be set to 3. If a sentence in the target language contains a redundant part with a length greater than or equal to 3, it will be considered as possibly containing an explanation.

3.1.3.3. Word alignment in explanation part

The explanation part is a redundant part of the target language sentence, so the corresponding word cannot be found in the source language sentence, which means that the word alignment result of the word in the explanation part does not exist. But this is not absolute. The following example can be observed:

1. **En**: You know , my own universe might be a book of **haiku** .
   De: Wisst ihr , mein eigenes Universum könnte auch ein **Haiku-Buch** sein [japanische Kurzgedichte] .

In this example, the blue part of the target sentence is redundant part with an explanation for German word *Haiku-Buch*, however the German word *sein* is aligned with the word *be* in the English sentence. Considering this situation, we relax the word alignment results in the explanation part in the experiment. The relaxation setting for the word alignment of the explanation part are listed in Table 3.4.

Length	Number
0 ∼ 3	0
4 ∼ 6	1
≥ 7	2

Table 3.4.: relaxation setting for the word alignment of the explanation

If the length of the redundant part is within 3, all the words in the redundant part cannot have word alignment results. If there are 4 to 6 tokens in the redundant part, at most one token in the redundant part can have a word alignment result. If the number of tokens in the redundant part is greater than or equal to 7, there can be at most two tokens that can have word alignment results.

3.1.3.4. Distinguish between common and uncommon words

In this thesis, the basis for finding sentence pairs with explanations is to find those words that are common in the source language but uncommon in the target language. For a word, its word count in all Wikipedia articles in the corresponding language is used to distinguish whether it is an uncommon word. A word can be considered an uncommon word if its word count is below a certain threshold.

This threshold is also one of the parameters that has a big impact on the method of finding target sentence pairs with explanations. Because if the threshold is set too low, many examples will be missed, but if the threshold is set too high, then many sentence pairs that do not actually have explanations will be selected, which will also greatly increase the final manual selected workload.

Therefore, finding an appropriate threshold that can reduce the workload of final manual selection while retaining as many target sentence pairs as possible is an important step in building the dataset.

In order to quickly find a suitable threshold, a relatively large value can be set to a start point, then select several other different smaller values for experiments, and determine the best value for threshold by comparing the experimental results.

3.1.3.5. Selection of NER tools and models

Many tools support named entity recognition (NER), and some tools also provide different NER models for the same language. And NER also affects the final result of the method of finding sentence pairs with explanations.

If a NER tool is used to identify named entities, and then the NER results contain as many named entities as possible that need to be explained, then this NER tool may provide better results. So different NER tools and models need to be tested to select the best one.

When carrying out the step of NER, one more thing to note is that in order to recognize as many target sentence pairs as possible, we will perform NER on both the source language and the target language sentences. As long as the named entity that needs to be explained is identified in one of the sentences, then this sentence pair will be considered as a candidate sentence pair.

3.1.3.6. Handling duplicate named-entities

In the experiment, the final experimental result contains a large number of sentence pairs, and the redundant part of the sentence in the target language does not contain the explanation of the word or phrase, but the word or phrase itself. If these sentence pairs can be removed, the final manual workload can be greatly reduced without reducing the number of target sentence pairs.

Observe the following examples:

1. **En**: Chevrolet also is the sole Engine supplier for the Formula Rolon single seater series in India .
   De: Chevrolet ist auch der alleinige Motorlieferant für die Formel Rolon (Formel Rolon) einzelne seater Reihe in Indien .

2. **En**: From 1967 to 1991 Johnson collaborated with John Burgee .
   De: Von 1967 bis 1991 arbeitete Johnson mit John Burgee (John Burgee) zusammen .

3. **En**: It is notable that John Wildman , resisted religious language , arguing that the Bible produced no model for civil government and that reason should be the basis of any future settlement .
   De: Es ist bemerkenswert , dass John Wildman (John Wildman) religiöser Sprache widerstand , behauptend , dass die Bibel kein Modell für die Zivilregierung erzeugte , und dass Grund die Basis jeder zukünftigen Ansiedlung sein sollte .

For these three sentence pairs, the blue part of the sentence in the target language is the redundant part, but the redundant part contains the red named entity itself. These sentence pairs should all be removed.

According to the token contained in the redundant part, it can be found that there are three situations for sentence pairs containing repeated named entities. The first is that the redundant part consists only of parentheses and the named entity itself (1. case). The second is that the redundant part contains other words besides parentheses and the named entity itself (2. case). And the third is that the redundant part only contains the left parenthesis and the named entity itself, but the right parenthesis is missing (3. case).

According to these three situations in Table 3.5, the content in the redundant part can be simply extracted and compare them with the results of NER. Using this method, a sentence contains repeated named entities can be easily determined.

Case	Redundant part
1	Named-Entity (Named-Entity)
2	Named-Entity (Named-Entity) Other-words
3	Named-Entity (Named-Entity

Table 3.5.: Three cases of duplicate named entities

3.1.3.7. Comparison of named entities and Wikipedia titles

After NER, the identified named entities will be compared with the titles of Wikipedia articles to further find and determine which words and phrases need to be explained.

However, sometimes, even though the identified named entity is the object referred to by the title of a Wikipedia article, the comparison result between the named entity and the title shows that they are not the same. This is because the identified named entity is in a different form than the Wikipedia article title. For example, if a named entity ends in *-s* to indicate plural, and the corresponding Wikipedia article title is singular, then although the two refer to the same object, the comparison results show that they are different. The following example clearly illustrates this situation:

1. **En**: The Vibhutis of the Lord , Siddhis and Riddhis are theirs though they do not want them .

> **De**: Die Vibhutis (Gnade) des Herrn , Siddhis und Riddhis (acht / vier okkulte Kräfte) gehören ihnen , obwohl sie sie nicht begehren .

The word **Vibhutis** marked in red is the identified named entity, and its corresponding Wikipedia article title is **Vibhuti**. It means that **Vibhutis** is the plural form of **Vibhuti**.

Possible formal inconsistencies between identified named entities and their corresponding Wikipedia article titles can affect the performance of the method for finding target sentence pairs. To solve this problem, before comparing the recognized named entity with the Wikipedia article title, the stemming technique is used on the named entity to eliminate the interference caused by the inconsistency of the form as much as possible.

4. Evaluation

In this chapter, we present the experiments and the results obtained. In Section 4.1, the experimental setup will be introduced. After this, in Section 4.2, the evaluation metrics used in the experiments are introduced. Finally, in Section 4.3 we show the performance of our proposed method for identifying and finding sentences containing words need to be explained and discuss the results obtained.

4.1. Experimental Setup

4.1.1. Datasets

Our experiments are conducted on three language pairs: English-German (En-De), English-Chinese (En-Zh), English-French (En-Fr). In all language pairs, English is the source language. In other words, our experiments are only performed on the three language pairs: **En**→ **De**, **En** → **Zh** and **En** → **Fr**. In order to ensure that as many target sentence pairs as possible can be found, the number of sentence pairs for each language pair in the corpus must be sufficiently large. Therefore, we choose CCMatrix [46, 13] as the corpus required for the experiment. Table 4.1 gives the statistics of the corpus CCMatrix. All CCMatrix data are downloaded from OPUS [48].

Language pairs	**Sentences Number (M : Million)**
English-German	247.5 M
English-Chinese	71.4 M
English-French	328.6 M

Table 4.1.: CCMatrix Statistics

In addition, Wikipedia titles and articles are also required for experiments. All articles and titles can be obtained from Wikipedia dumps. The Wikipedia data used in the experiments are in Table 4.2.

We use the Wikipedia article files (files ending in *-pages-articles.xml.bz2*) to obtain word counts used to determine whether a word is rare in a language. At the same time, article files are also used to count the size of Wikipedia articles. Wikipedia titles are contained in files ending in *-all-titles-in-ns0.gz*. Base per-page data (files ending in *-page.sql.gz*) and Wiki interlanguage link records(files ending in *-langlinks.sql.gz*) are used to create Wikipedia parallel titles corpus. This corpus contains titles that co-exist in Wikipedia article titles in two selected languages.

Language	File Name
English	enwiki-20221101-pages-articles.xml.bz2
English	enwiki-20221101-all-titles-in-ns0.gz
German	dewiki-20221101-pages-articles.xml.bz2
German	dewiki-20221101-all-titles-in-ns0.gz
German	dewiki-20221101-page.sql.gz
German	dewiki-20221101-langlinks.sql.gz
Chinese	zhwiki-20221101-pages-articles.xml.bz2
Chinese	zhwiki-20221101-all-titles-in-ns0.gz
Chinese	zhwiki-20221101-page.sql.gz
Chinese	zhwiki-20221101-langlinks.sql.gz
French	frwiki-20221101-pages-articles.xml.bz2
French	frwiki-20221101-all-titles-in-ns0.gz
French	frwiki-20221101-page.sql.gz
French	frwiki-20221101-langlinks.sql.gz

Table 4.2.: Wikipedia Data

4.1.2. Tools and Models

4.1.2.1. Preprocessing

For each language, there are several options for word tokenization. Table 4.3 lists all the tools we found that can complete word tokenization tasks and the languages they support.

Among these tools, spaCy, Stanza and HanLP can support all languages involved in experiments. Compared with Stanza and HanLP, spaCy has the fastest word tokenization speed. In addition, spaCy can also support two other word tokenization tools pkuseg and jieba for Chinese. Therefore, we choose spaCy as the word tokenization tool. For Chinese, we use pkuseg under the framework of spaCy for word tokenization.

Tool	Supported Language
spaCy	English, German, Chinese, French
NLTK	English, German, French
Stanza	English, German, Chinese, French
HanLP	English, German, Chinese, French
pkuseg	Chinese
jieba	Chinese

Table 4.3.: Word tokenization tool

In preprocessing, the task that needs to be completed after word tokenization is to extract word alignment between sentence pairs. We use awesome-align [11] to extract word alignment results. This tool provides pre-trained word alignment models, as well as the performance of each model for different language pairs (Table 4.4 [11]). The alignment error rates (AERs) are used as performance scores. For each language pair we choose the best performing model to extract word alignments.

Model	**Language pairs**		
	De-En	Fr-En	Zh-En
Ours (w/o fine-tuning, softmax)	17.4	5.6	18.1
Ours (multilingually fine-tuned w/o –train_co, softmax)	15.2	**4.1**	**13.4**
Ours (multilingually fine-tuned w/ –train_co, softmax)	**15.1**	4.5	14.5

Table 4.4.: Model performance of awesome-align [11]

4.1.2.2. NER

There are several tools that can perform named entity recognition (NER), and the languages they support are listed in Table 4.5. We will test all these tools in the experiment, and then select a tool with the best performance for subsequent experimental steps.

For simplicity, in the same language pair, the same tool is chosen to perform the NER task in the source and target languages. For each NER tool, if multiple NER models are provided, then the available NER model with the highest accuracy is chosen. For example, spaCy provides four English NER models, among which the transformer version model is the most accurate, but it cannot run on our machine, so we finally choose the large version of the NER model. All NER models selected for the experiment are listed in Table 4.6.

Tool	**Supported Language**
spaCy	English, German, Chinese, French
Stanza	English, German, Chinese, French
Flair	English, German, French
HanLP	English, Chinese

Table 4.5.: NER tool

4.1.2.3. Processing of Wikipedia data

When the Wikipedia article from a Wikipedia database backup dump is downloaded, it cannot be read and used directly. We need to find a tool to extract the article from the file. We use *wikiextractor* [3] to extract Wikipedia articles. Meanwhile, we use the tool *wikipedia-parallel-titles* [2] to create the Wikipedia parallel titles corpus.

4.1.2.4. Stemming

In the experiment, it is necessary to use the stemming algorithm to process the found words or phrases, in order to better compare the words or phrases with Wikipedia titles. NLTK [5] implements stemming algorithms, including Porter algorithm and Snowball algorithm. Because the Snowball algorithm supports multiple languages, and it works better than the Porter algorithm, we choose the Snowball algorithm to perform stemming.

Language	Tool	Model name
English	spaCy	en_core_web_lg
English	Stanza	en model
English	Fliar	ner-english-large
English	HanLP	CONLL03_NER_BERT_BASE_CASED_EN
German	spaCy	de_core_news_lg
German	Stanza	de model
German	Fliar	ner-german-large
French	spaCy	fr_core_news_lg
French	Stanza	fr model
French	Fliar	ner-french
Chinese	spaCy	zh_core_web_lg
Chinese	Stanza	zh model
Chinese	HanLP	MSRA_NER_ALBERT_BASE_ZH

Table 4.6.: Selected NER models

4.2. Evaluation Metric

Filter the target sentences from the corpus to construct the training dataset and train the model to predict which words need to be explained. These problems are all classification problems. Therefore, the metric BLEU used to evaluate machine translation models does not work here. For a classification problem, F1-score is a good evaluation metric. The calculation of the F1-score requires the number of positive examples and the number of negative examples. However, in our experiments, the evaluation of the method for finding target sentences only involves target sentences, i.e. only the number of positive examples is considered. This means that we just selected a subset of F1-score as the evaluation metric for our experiments.

4.3. Performance of the method to build training dataset

4.3.1. Find candidates

In order to compare and evaluate the subsequent steps of our proposed method, we first run our method to the step before NER, which is the ninth step, Check if there are other words in the redundant part except the explained word.

First take the first 1 million sentence pairs from the CCMatrix corpus as input, and the word count thresholds for both the source and target languages are set to 15,000. The statistical results are in Table 4.7.

It can be observed from the table that the results of the three language pairs are very similar, and there are more than 1000 sentence pairs left before NER. The En→Fr language pair has the least number of remaining sentence pairs, it has 1228 sentence pairs. The En→Zh language pair has the largest number of remaining sentence pairs with 2274, while the En→De language pair has 1701 sentence pairs remaining.

	En→De	En→Zh	En→Fr
Total	1000000	1000000	1000000
1. Check sou. word count (15000)	696070	670186	677946
2. Sou. word has one alignment	669131	603481	657832
3. Exists a redundant part	57115	91043	35844
4. word in redundant part no align.	2656	6735	3638
5. Check tar. word count (15000)	2079	5068	2498
6. Redundant part has punctuation	1717	2279	1231
7. Explained word not in redundant part	1701/21	2274/54	1228/20

Table 4.7.: The statistical results of the first 1 million sentence pairs

On the basis of the remaining sentence pairs, manual work is performed to select out the sentence pairs that contain explanations. The results are in the the last line (7. step) of Table 4.7. Finally, 21 sentence pairs with explanations are found out of 1701 En→De sentence pairs. 54 sentence pairs with explanations are found in 2274 En→Zh sentence pairs. Out of 1228 En→Fr sentence pairs, 20 sentence pairs with explanations are found.

In order to reduce the possible deviations caused by the input data, the input is expanded to the first five million sentence pairs of the corpus, and the same experimental steps are executed. The results are in Table 4.8.

	En→De	En→Zh	En→Fr
Total	5000000	5000000	5000000
1. Check sou. word count (15000)	3694372	3668594	3598430
2. Sou. word has one alignment	3556339	3329814	3495833
3. Exists a redundant part	345466	615601	230934
4. word in redundant part no align.	14813	42078	23466
5. Check tar. word count (15000)	11811	32062	16089
6. Redundant part has punctuation	9197	13561	6988
7. Explained word not in redundant part	8977/173	13541/402	6982/122

Table 4.8.: The statistical results of the first 5 million sentence pairs

The same manual work is done for the input sentence pairs of 5 million to select the sentence pairs containing the explanation. The statistical results are shown in Table 4.9. The corresponding F1-score for each language pair is also calculated. Based on the results in Table 4.9, the subsequent experiments can be performed to compare the effects of different NER tools.

Step	En→De	En→Zh	En→Fr
7. Explained word not in redundant part	8977/173	13541/402	6982/122
7. F1-score	0.0378	0.0577	0.0343

Table 4.9.: The statistical results of the first 5 million sentence pairs

4.3.2. Comparison of NER tools

The next step is to use NER to identify as many words and phrases as possible that need to be explained. Different NER tools have different effects and can also affect the results of subsequent steps. Therefore, it is necessary to compare the effects of different NER tools.

Based on the results in Table 4.9, experiments on NER are conducted. In other words, the input data is the first five million sentence pairs for each language pair in the CCMatrix corpus. The word count thresholds for both the source and target languages are 15000.

The results of NER for the En→De language pair are in Table 4.10. To compare the performance of different NER tools, the F1-score of each NER tool is also calculated. From the F1-score of each NER tool, it can be seen that NER model of flair has the best effect. After using NER from flair, there are 1391 sentence pairs left, of which there are 126 sentence pairs that contain explanations. Its F1-score, 0.1611, is also the highest among these NER tools.

Step	**Numbers**	**F1-Score**
7. Explained word not in redundant part	8977/173	0.0378
8. NER (Flair)	**1391/126**	**0.1611**
8. NER (Stanza)	1488/118	0.1421
8. NER (spaCy)	2223/132	0.1102

Table 4.10.: NER result for En→De (first 5 million)

Furthermore, the NER results for the En→Fr language pair are in Table 4.11, and the NER results for the En→Zh language pair are in Table 4.12. The NER results for the En→Fr language pair and the En→Zh language pair are similar to the NER results for the En→De language pair. NER can save as many target sentence pairs with explanations as possible while removing a large number of non-candidate sentence pairs.

Step	**Numbers**	**F1-Score**
7. Explained word not in redundant part	6982/122	0.0343
8. NER (Flair)	2148/98	0.0863
8. NER (Stanza)	**2152/99**	**0.0871**
8. NER (spaCy)	2196/96	0.0819

Table 4.11.: NER result for En→Fr (first 5 million)

Step	**Numbers**	**F1-Score**
7. Explained word not in redundant part	13541/402	0.0577
8. NER (Hanlp)	**3897/274**	**0.1275**
8. NER (Stanza)	4419/277	0.1149
8. NER (spaCy)	4511/282	0.1148

Table 4.12.: NER result for En→Zh (first 5 million)

But for each language pair, the most suitable NER tool (i.e. has the highest F1-score) is different. For the En→Fr language pair, Stanza's NER model has the highest F1-score. The F1 score is 0.0871 after using Stanza's NER model. However, the F1 scores of the three NER tools for the En→Fr language pair are all extremely close, which is different from the other two language pairs. For the En→Zh language pair, Hanlp's NER model achieved the highest F1-score of 0.1275.

From the results of NER, it can be observed that for the three language pairs, NER can greatly reduce the amount of manual work required in subsequent steps to improve the efficiency of finding target sentence pairs. On the other hand, by comparing the F1-scores of different NER tools, it can be found that for En→De language pair, different tools have different effects. The most suitable NER tool can be found quickly. But for the En→Fr and En→Zh language pairs, the F1-scores of each NER tool are very similar, which shows that there is not much difference between these NER tools.

For each language pair, the result of the NER tool with the highest F1-score (Table 4.13) will be selected for subsequent experiments. More precisely, the bold row in the table of each NER result is the baseline for subsequent experiments.

Step	**En→De**	**En→Zh**	**En→Fr**
7. Explained word not in redundant part	8977/173	13541/402	6982/122
7. F1-Score	0.0378	0.0577	0.0343
8. NER	1391/126	3897/274	2152/99
8. F1-Score	0.1611	0.1275	0.0871

Table 4.13.: NER result for all language pairs (First 5 million)

4.3.3. Performance using Wikipedia

After using NER, the named entities in each sentence will be recognized. On the basis of the identified named entities, Wikipedia can be used to further select target sentence pairs. Before using Wikipedia, the first thing to do is to remove sentence pairs containing duplicate named entities in the redundant part.

On the basis of the results in Table 4.13, the step of removing duplicate named entities is performed. The results are in Table 4.14 and 4.15.

Step	**En→De**	**En→Zh**	**En→Fr**
7. Explained word not in redundant part	8977/173	13541/402	6982/122
8. NER	1391/126	3897/274	2152/99
9. Remove duplicate named entities	1243/126	3897/274	2151/99

Table 4.14.: Result after removing duplicate named entities for all language pairs (First 5 million)

The step of removing duplicate named entities works well for the En→De language pair, it saves all target sentence pairs with explanations while reducing the number of

Step	En→De	En→Zh	En→Fr
7. Explained word not in redundant part	0.0378	0.0577	0.0343
8. NER	0.1611	0.1275	0.0871
9. Remove duplicate named entities	0.1780	0.1275	0.0871

Table 4.15.: F1-score after removing duplicate named entities for all language pairs (First 5 million)

non-candidate sentence pairs. Its corresponding F1-score is also significantly improved. However, this step does not work for the En→Fr and En→Zh language pairs, the F1-scores are the same as before.

After removing the sentence pairs containing repeated named entities, Wikipedia can be used to continue to identify and select the target sentence pairs in the next step. First, the identified named entities in the source language sentences are checked for consistency with Wikipedia titles. The comparison results of the identified named entities and Wikipedia titles are in Table 4.16.

Step	En→De	En→Zh	En→Fr
7. Explained word not in redundant part	8977/173	13541/402	6982/122
9. Remove duplicate named entities	1243/126	3897/274	2151/99
10. Check sou. wiki title	869/90	2783/209	1047/38

Table 4.16.: The comparison results of the entities and Wikipedia titles for all language pairs (First 5 million)

Step	En→De	En→Zh	En→Fr
7. Explained word not in redundant part	0.0378	0.0577	0.0343
9. Remove duplicate named entities	0.1780	0.1275	0.0871
10. Check sou. wiki title	0.1727	0.1312	0.0650

Table 4.17.: F1-scores after the comparison of the entities and Wikipedia titles for all language pairs (First 5 million)

The results in Table 4.16 show that after comparing the identified named entities with Wikipedia titles, there is a significant reduction in the number of remaining sentence pairs, but at the same time the number of target sentence pairs containing explanations is also reduced. In order to compare and evaluate the performance of this step more clearly, the F1-scores are calculated and the results are presented in Table 4.17. According to the results in Table 4.17, after comparing named entities and Wikipedia titles, only the F1-score of the En→Fr language pair has an observable change, It decreased from 0.0871 to 0.0650. Meanwhile, the F1 scores of the other two language pairs are very similar, and there are no obvious changes.

On the other hand, after comparing named entities and Wikipedia titles, the number of remaining sentence pairs still does not reach an ideal value. For instance, after this step, there are 2783 sentence pairs left in the En→Zh language pair. It is still a challenge to manually find out the target sentence pairs from these sentence pairs, because the manual workload is still huge. This shows that it is not enough to compare the identified named entities and Wikipedia titles in the source language sentences, and it is necessary to further identify and compare the remaining sentences with the help of Wikipedia.

After comparing the identified named entities in the source language sentences with Wikipedia titles, the next step is to check whether these Wikipedia titles also have corresponding titles in the target language. In other words, this step is to check whether the identified named entity is in the parallel Wikipedia titles. The results of this step are in Table 4.18.

Step	**En→De**	**En→Zh**	**En→Fr**
7. Explained word not in redundant part	8977/173	13541/402	6982/122
10. Check sou. wiki title	869/90	2783/209	1047/38
11.1 Not in the parallel Wikipedia title	313/43	1459/119	348/16
11.2 In the parallel Wikipedia title	557/48	1333/93	701/22

Table 4.18.: The comparison results of the entities and Wikipedia parallel titles for all language pairs (First 5 million)

After comparing the named entity with the Wikipedia parallel title, the result consists of two parts, one is that the named entity is not in the parallel title (Step 11.1 in Table 4.18), i.e., the named entity is a Wikipedia title in the source language, and there is no corresponding Wikipedia title in the target language. The other part is that the named entity is in the parallel title (Step 11.2 in Table 4.18).

Since a named entity not in a parallel Wikipedia title is a good candidate to be explained, we can focus on the results of step 11.1 in Table 4.18. Comparing the F1-scores in Table 4.19, it can be found that the F1-scores of the step 11.1 are very similar to the F1-scores of the previous steps, although the F1-scores of the En→De and En→Fr language pairs have slightly improved.

Step	**En→De**	**En→Zh**	**En→Fr**
7. Explained word not in redundant part	0.0378	0.0577	0.0343
10. Check Wiki title	0.1727	0.1312	0.0650
11.1 Not in the parallel Wikipedia title	0.1770	0.1279	0.0681
11.2 In the parallel Wikipedia title	0.1315	0.1072	0.0535

Table 4.19.: F1-scores after the comparison of the entities and Wikipedia parallel titles for all language pairs (First 5 million)

At the same time, it can be found that there are also many target sentence pairs with explanations in the part where the named entity is a parallel Wikipedia title (Step 11.2 in Table 4.18). For example, for the En→De language pair, after comparing named entities and parallel Wikipedia titles, there are a total of 557 sentence pairs in which the named entity is a parallel Wikipedia title, of which 48 sentence pairs are with explanations, while only 43 sentence pairs with explanations are in another part (Step 11.1). This means that if only Step 11.1 is considered, then half of the target sentence pairs with explanations will be excluded. Therefore, based on the results of step 11.2 in Table 4.18, these sentence pairs in step 11.2 can be further identified and selected by using Wikipedia.

If the identified named entity in the source language sentence is a Wikipedia title in the source language, and there is a corresponding Wikipedia title in the target language, then the next step is to compare the Wikipedia titles in the target language with the identified named entities in the target language sentences to see if they are consistent. In other words, check the consistency of the identified named entities with Wikipedia titles in the target language sentences.

The comparison results of the identified named entities in target language sentences and Wikipedia titles are in Table 4.20. Surprisingly, for the En→Zh language pair, there are no sentence pairs with explanations among the remaining sentence pairs after the comparison.

In the NER step (Step 8), NER is performed on both source and target language sentences. When comparing identified named entities with Wikipedia parallel titles (Step 11.2), only the named entities in source language sentences are considered, and named entities in target language sentences are ignored. For En→Zh language pair, among the remaining 1333 sentence pairs after step 11.2, there are a large number of sentence pairs whose named entities in the target language sentences are not recognized. These sentence pairs will be removed when comparing named entities in the target sentence with Wikipedia titles (Step 12). However, all target sentence pairs containing explanations are also in these removed sentence pairs. This leads to the fact that after step 12, there are no sentence pairs with explanations among the remaining sentence pairs.

If we want to improve the experimental results of this step, we should find and test more NER tools and models that support Chinese, and these NER models should have higher accuracy. However, this is not an easy task.

Step	**En→De**	**En→Zh**	**En→Fr**
7. Explained word not in redundant part	8977/173	13541/402	6982/122
11.2 In the parallel Wikipedia title	557/48	1333/93	701/22
12. Check target wiki title	285/30	103/0	421/12

Table 4.20.: The comparison results of the entities in target language and Wikipedia titles for all language pairs (First 5 million)

After comparing the identified named entities in the target language sentences with the Wikipedia titles, If the named entity in the target language sentence is the same as the Wikipedia title, then the last thing to do is to compare the size of the Wikipedia article corresponding to the Wikipedia title in the source language and the target language. If the size of the Wikipedia article corresponding to the source language title is larger, then the named entity corresponding to the source language title is a good candidate to be explained.

The comparison results of the sizes of Wikipedia articles are in Table 4.21. It can be seen from the results that the comparison of the size of Wikipedia articles can continue to reduce the number of remaining sentence pairs on the basis of the previous step, but at the same time the number of target sentence pairs with explanations is also reduced.

Step	En→De	En→Zh	En→Fr
7. Explained word not in redundant part	8977/173	13541/402	6982/122
12. Check target wiki title	285/30	103/0	421/12
13. Check wiki article size	155/13	98/0	248/4

Table 4.21.: The comparison results of the wiki article size for all language pairs (First 5 million)

Finally we combine the results of each step of using Wikipedia to identify and select target sentence pairs. This allows for a clearer evaluation of Wikipedia's performance in identifying and selecting sentence pairs with explanations.

Here three cases are considered. The first case is when the recognized named entity in the source language sentence is not a parallel Wikipedia title (Step 11.1). In the second case it consists of two parts, the first part is the first case (Step 11.1), and the second part is that the recognized named entity in the source language is a parallel Wikipedia title, and the Wikipedia title in the target language is consistent with the recognized named entity in the target language sentence (Step 12). The third case also contains two parts, the first part is still the first case (Step 11.1), the second part is based on the second part of the second case and then continues to compare the size of Wikipedia articles (Step 13).

The final overall results are in Table 4.22 and Table 4.23.

Step	En→De	En→Zh	En→Fr
7. Explained word not in redundant part	8977/173	13541/402	6982/122
8. NER	1391/126	3897/274	2152/99
9. Remove duplicate named entities	1243/126	3897/274	2151/99
10. Check Wiki title	869/90	2783/209	1047/38
11.1. Not in para. Wiki. title	313/43	1459/119	348/16
11.1.+12. Not in para. Wiki title+Check tar. wiki title	598/73	1562/119	769/28
11.1.+13. Not in para. Wiki title+Check wiki article size	468/56	1557/119	596/20

Table 4.22.: Result after using Wikipedia for all language pairs (First 5 million)

From the results in Table 4.22, it can be found that for the three language pairs, using Wikipedia to identify and select target sentence pairs with explanations can significantly reduce the final manual workload. For example, for the En→De language pair, after removing sentence pairs with duplicate named entities (Step 9), there are 1243 sentence pairs left, while after using Wikipedia, only 313 sentence pairs remain after step 11.1. This is an extremely large reduction in manual work.

However, as the number of remaining sentence pairs decreases, the number of target sentence pairs with explanations included in it also decreases. For the En→De language pair, after step 9, in the remaining 1243 sentence pairs there are 126 sentence pairs with explanations, while after step 11.1, only 43 sentence pairs with explanations can be found out of the remaining 313 sentence pairs.

Step	**En→De**	**En→Zh**	**En→Fr**
7. Explained word not in redundant part	0.0378	0.0577	0.0343
8. NER	0.1611	0.1275	0.0871
9. Remove duplicate named entities	0.1780	0.1275	0.0871
10. Check Wiki title	0.1727	**0.1312**	0.0650
11.1 Not in the parallel Wikipedia title	0.1770	0.1279	**0.0681**
11.1.+12. Not in para. Wiki title+Check tar. wiki title	**0.1894**	0.1212	0.0629
11.1.+13. Not in para. Wiki title+Check wiki article size	0.1747	0.1214	0.0557

Table 4.23.: F1-scores after using Wikipedia for all language pairs (First 5 million)

On the other hand, if we look at the results of the F1-scores in Table 4.23, then the situation is different. For each language pair, the steps to get the highest F1-score after using Wikipedia are different. For the En→De language pair, steps 11.1 + 12 have the highest F1-score of 0.1894. For the En→Zh language pair, step 10 has the highest F1-score of 0.1312. While for the En→Fr language pair, step 11.1 has the highest F1-score, 0.0681.

For the same language pair, the F1-scores at different steps after using Wikipedia are similar. For both En→Fr and En→Zh language pairs, the F1-scores for the three cases we consider are very close. For the En→De language pair, although the F1-score of step 11.1+12 is higher than the F1-scores of other steps, considering the number of remaining sentences, it can be thought that the effect is not much different.

If we compare the highest F1-score after using Wikipedia for each language pair with the highest F1-score before using Wikipedia, We can find that Wikipedia's performance in identifying and selecting target sentence pairs is not the same for different language pairs. For En→Fr language pair, the use of Wikipedia leads to a decrease in F1-score. This shows that for this language pair, the use of Wikipedia will remove many target sentences with explanations while reducing the amount of manual work. For the En→De language pair, the use of Wikipedia can improve the performance of identifying and selecting target sentence pairs on the basis of NER. But for En→Zh language pair, although F1-score has improved after using Wikipedia, the improvement is very small, the results are very similar.

4.3.4. Comparison of different thresholds for word counts

The word count threshold for deciding whether a word is an uncommon word is an important parameter. This parameter affects the performance of methods for finding target sentence pairs. Therefore, for each language pair, finding the most appropriate word count threshold is an important task.

We need to determine not only word count thresholds for source language words, but also word count thresholds for target language words. If try both are not the same combination, We can get more accurate results. But this will make the search extremely slow and take a huge amount of time. Therefore, for simplicity, we consider setting the thresholds of the source and target language words to be the same value. For each language pair, we selected 5 different threshold pairs starting from 15000 for experiments. They are: 15000, 10000, 5000, 1000 and 100.

In order to test the effects of different word count thresholds on different NER tools, we first run the method to the step before NER. For each language pair, the input is the first five million sentence pair of the corpus.

The results of En→De language pairs are in Table 4.24. Similarly, based on the experimental results obtained by step 7 with the threshold value of 15000, the F1-scores of different thresholds can be calculated. The results of the F1-scores are in Table 4.25.

From the results of Table 4.24 and 4.25, we can see that as the threshold becomes smaller, the number of remaining sentence pairs and the number of target sentence pairs with explanation are decreasing, and the F1-score is increasing, from 0.0378 to 0.1333.

	15000	**10000**	**5000**	**1000**	**100**
Total	5000000	5000000	5000000	5000000	5000000
1. Check sou. word count (Para. 1)	3694372	3266672	2587424	1450466	724707
2. Sou. word has one alignment	3556339	3130441	2462009	1369452	681064
3. Exists a redundant part	345466	278532	196021	99449	56797
4. Word in redund. part no align.	14813	12436	5883	2780	1288
5. Check tar. word count (Para. 2)	11811	8166	4405	1817	641
6. Redund. part has punctuation	9197	6094	3253	1356	480
7. Explained word not in redund. part	8977/173	5901/159	3102/134	1262/90	442/41

Table 4.24.: The results of different threshold pairs in En→De language pair

	15000	**10000**	**5000**	**1000**	**100**
7. Explained word not in redund. part	0.0378	0.0524	0.0818	0.1254	0.1333

Table 4.25.: F1-scores of different threshold pairs in En→De language pair

The same experiment is carried out for En→Fr and En→Zh language pairs, the results for En→Fr are in Tables 4.26 and 4.27, and the results for En→Zh are in Tables 4.28 and 4.29.

	15000	**10000**	**5000**	**1000**	**100**
Total	5000000	5000000	5000000	5000000	5000000
1. Check sou. word count (Para. 1)	3598430	3141224	2426412	1290786	625966
2. Sou. word has one alignment	3495833	3041323	2337267	1233728	595298
3. Exists a redundant part	230934	179546	120709	61192	38059
4. word in redund. part no align.	23466	17954	11257	4876	2412
5. Check tar. word count (Para. 2)	16089	11749	6882	2501	1035
6. Redund. part has punctuation	6988	5406	3355	1390	640
7. Explained word not in redund. part	6982/122	5400/112	3350/95	1388/64	639/37

Table 4.26.: The results of different threshold pairs in En→Fr language pair

	15000	10000	5000	1000	100
7. Explained word not in redund. part	0.0343	0.0405	0.0547	0.0848	0.0972

Table 4.27.: F1-scores of different threshold pairs in En→Fr language pair

	15000	10000	5000	1000	100
Total	5000000	5000000	5000000	5000000	5000000
1. Check sou. word count (Para. 1)	3668594	3241378	2550061	1382301	658471
2. Sou. word has one alignment	3329814	2918308	2270959	1215187	577915
3. Exists a redundant part	615601	509745	374455	196774	106865
4. Word in redund. part no align.	42078	34698	24762	11995	5982
5. Check tar. word count (Para. 2)	32062	25083	16399	6247	2358
6. Redund. part has punctuation	13561	11000	7373	3044	1205
7. Explained word not in redund. part	13541/402	10983/365	7360/302	3038/177	1203/82

Table 4.28.: The results of different threshold pairs in En→Zh language pair

For En→Fr language pair, a similar conclusion can be obtained from its results, that is, as the threshold decreases, the number of remaining sentence pairs and the number of target sentence pairs with explanation are decreasing, and the F1-score is increasing, from 0.0343 to 0.0972. However, the results for En→Zh language pair are somewhat different. For En→Zh language pair, as the threshold decreases, the number of remaining sentence pairs and the number of target sentence pairs with explanations also decrease, but the F1 score does not always rise. When the threshold is 1000, the F1-score is the highest, which is 0.1029. When the threshold is 100, the F1-score is 0.1022, which is extremely close to the F1-score when the threshold is 1000, but still lower than it.

4.3.4.1. Effect of thresholds on NER steps

Based on the results obtained before, we can continue to test the impact of different thresholds on NER tools.

In Table 4.30 is the result of En→De language pair. In Table 4.31 is the F1-score of each NER tool under different thresholds.

We can find some interesting things from the F1-scores in Table 4.31. First of all, for each NER tool, as the threshold decreases, the F1-score is not always increased. For example, for the NER models from Stanza and spaCy, when the threshold is 1000, their F1-scores are the highest. But when the threshold is set to 100, their F1-scores are reduced. For the NER model from flair, when the threshold is 5000, its F1-score is the highest, which is 0.1735.

	15000	10000	5000	1000	100
7. Explained word not in redund. part	0.0577	0.0641	0.0778	0.1029	0.1022

Table 4.29.: F1-scores of different threshold pairs in En→Zh language pair

	15000	**10000**	**5000**	**1000**	**100**
7. Expl. word not in redund. part	8977/173	5901/159	3102/134	1262/90	442/41
8. NER (flair)	1391/126	1163/115	899/93	544/62	252/31
8. NER (Stanza)	1488/118	1245/107	941/86	558/57	255/26
8. NER (spaCy)	2223/132	1905/123	1254/100	724/66	327/30

Table 4.30.: The result of the NER under different thresholds of En→De language pair

	15000	**10000**	**5000**	**1000**	**100**
7. Expl. word not in redund. part	0.0378	0.0524	0.0818	0.1254	0.1333
8. **NER (flair)**	**0.1611**	**0.1722**	**0.1735**	**0.1729**	**0.1459**
8. NER (Stanza)	0.1421	0.1509	0.1544	0.1560	0.1215
8. NER (spaCy)	0.1102	0.1184	0.1402	0.1472	0.12

Table 4.31.: The F1-scores of the NER under different thresholds of En→De language pair

The second thing is that for each threshold, the F1-score of NER model from flair is always higher than the F1-score of other NER models.

Therefore, according to the results of the F1-score, for En→De language pair, the threshold is 5000, and when using the NER model from flair, the best results can be obtained.

For En→Zh and En→Fr language pairs, different NER tools are also tested at each threshold. The results for the En→Zh language pair are in Tables 4.32 and 4.33, and the results for the En→Fr language pair are in Tables 4.34 and 4.35.

	15000	**10000**	**5000**	**1000**	**100**
7. Expl. word not in redund. part	13541/402	10983/365	7360/302	3038/177	1203/82
8. NER (HanLP)	3897/274	3300/245	2557/194	1410/103	621/43
8. NER (Stanza)	4419/277	3831/252	2985/207	1615/113	769/53
8. NER (spaCy)	4511/282	3950/255	3124/209	1702/119	821/55

Table 4.32.: The result of the NER under different thresholds of En→Zh language pair

For the En→Fr language pair, the results are similar to the results of the En→De language pair. For each NER model, the F1-score does not consistently increase with decreasing threshold. For each threshold, the F1-score of NER model from stanza is consistently higher than that of NER models from other tools. The results of the NER step for the En→Fr language pair are optimal when the threshold is 1000 and the Stanza NER model is used.

For the En→Zh language pair, the situation is slightly different. Similarly, for each NER model, the F1-score does not consistently increase as the threshold is lowered. However, for each threshold, there is no single NER model that consistently has the highest F1-score. When the threshold is 100, the NER model of stanza has the highest F1 score, while for other thresholds, the NER model of HanLP has the highest F1-score. The results of the

	15000	**10000**	**5000**	**1000**	**100**
7. Expl. word not in redund. part	0.0577	0.0641	0.0778	0.1029	0.1022
8. **NER (HanLP)**	**0.1275**	**0.1324**	**0.1311**	**0.1137**	0.0841
8. NER (Stanza)	0.1149	0.1191	0.1222	0.1120	**0.0905**
8. NER (spaCy)	0.1148	0.1172	0.1185	0.1131	0.0899

Table 4.33.: F1-scores of the NER under different thresholds of En→Zh language pair

	15000	**10000**	**5000**	**1000**	**100**
7. Expl. word not in redund. part	6982/122	5400/112	3350/95	1388/64	639/37
8. NER (flair)	2148/98	1817/89	1359/75	861/48	464/25
8. NER (Stanza)	2152/99	1822/90	1333/76	790/48	404/27
8. NER (spaCy)	2196/96	1875/88	1453/75	920/50	494/26

Table 4.34.: The result of the NER under different thresholds of En→Fr language pair

NER step for the En→Zh language pair are optimal when the threshold is 10000 and the HanLP NER model is used.

After comparing the effects of different thresholds on the NER steps, based on the experimental results obtained, the most suitable NER model in each language pair is determined. The result is listed in Table 4.36. For the En→De language pair, the most suitable NER model is flair's model. For the En→Fr language pair, the most suitable NER model is Stanza's model. And HanLP's model is the most suitable NER model for the En→Zh language pair. We also give the threshold for obtaining the highest F1-score under each most suitable NER model. But this is not the final decision, because different thresholds also have an impact on the results of the steps using Wikipedia. We will combine all the experimental results to give the final decision about the threshold.

4.3.4.2. Effect of thresholds on Wikipedia usage steps

After confirming the most suitable NER model for each language pair, the effect of different thresholds on the use of Wikipedia can be tested.

For En→De language pair, the NER model from flair is used. The result is in Table 4.37 and Table 4.38.

From the results of the F1-score, it can be found that the use of Wikipedia can not always improve the performance for each threshold. In the case where the threshold is

	15000	**10000**	**5000**	**1000**	**100**
7. Expl. word not in redund. part	0.0343	0.0405	0.0547	0.0848	0.0972
8. NER (flair)	0.0863	0.0918	0.1013	0.0977	0.0853
8. **NER (Stanza)**	**0.0871**	**0.0926**	**0.1045**	**0.1053**	**0.1027**
8. NER (spaCy)	0.0828	0.0881	0.0952	0.0960	0.0844

Table 4.35.: F1-scores of the NER under different thresholds of En→Fr language pair

Language pair	NER tool	Threshold for NER
En→De	flair	5000
En→Fr	Stanza	1000
En→Zh	HanLP	10000

Table 4.36.: The most suitable NER tool and threshold combination of each language pair

	15000	10000	5000	1000	100
7. Expl. word not in redund. part	8977/173	5901/159	3102/134	1262/90	442/41
8. NER (flair)	1391/126	1163/115	899/93	544/62	252/31
9. Remove duplicate named entity	1243/126	1031/115	791/93	483/62	228/31
10. Check wiki title	869/90	691/80	506/68	290/42	92/17
11.1. Not in parallel wiki title	313/43	284/37	247/33	165/30	69/15
11.1. + 12. Check tar. wiki title	598/73	486/66	395/57	232/37	83/17
11.1. + 13. Check wiki article size	468/56	373/50	323/44	203/33	75/15

Table 4.37.: The results of the use of Wikipedia under different thresholds of En→De language pair

	15000	10000	5000	1000	100
7. Expl. word not in redund. part	0.0378	0.0524	0.0818	0.1254	0.1333
8. NER (flair)	0.1611	0.1722	0.1735	0.1729	0.1459
9. Remove duplicate named entity	0.1780	0.1910	0.1929	0.1890	0.1546
10. Check wiki title	0.1727	0.1852	0.2003	0.1814	0.1283
11.1. Not parallel wiki title	0.1770	0.1619	0.1571	0.1775	0.1240
11.1. + 12. Check tar. wiki title	0.1894	0.2003	**0.2007**	0.1827	0.1328
11.1. + 13. Check wiki article size	0.1747	0.1832	0.1774	0.1755	0.1210

Table 4.38.: F1-scores of the use of Wikipedia under different thresholds of En→De language pair

1000 and 100, the F1-score after using Wikipedia is decreased. Among all the steps using Wikipedia, the results of steps 11.1 and 12 are the best at each threshold. Among all the results of steps 11.1 and 12, when the threshold value is 5000, the F1-score is the highest, which is 0.2007.

The results of En→Zh language pairs are in Table 4.39 and 4.40, and the results of En→Fr language pairs are in Table 4.41 and 4.42.

	15000	**10000**	**5000**	**1000**	**100**
7. Expl. word not in redund. part	13541/402	10983/365	7360/302	3038/177	1203/82
8. NER (HanLP)	3897/274	3300/245	2557/194	1410/103	621/43
9. Remove duplicate named entity	3897/274	3300/245	2557/194	1410/103	621/43
10. Check wiki title	2783/209	2302/183	1736/142	852/67	282/18
11.1. Not in parallel wiki title	1459/119	1284/103	1049/87	621/45	221/15
11.1. + 12. Check tar. wiki title	1562/119	1362/103	1086/87	628/45	222/15
11.1. + 13. Check wiki article size	1557/119	1358/103	1083/87	626/45	222/15

Table 4.39.: The results of the use of Wikipedia under different thresholds of En→Zh language pair

	15000	**10000**	**5000**	**1000**	**100**
7. Expl. word not in redund. part	0.0577	0.0641	0.0778	0.1029	0.1022
8. NER (HanLP)	0.1275	0.1324	0.1311	0.1137	0.0841
9. Remove duplicate named entity	0.1275	0.1324	0.1311	0.1137	0.0841
10. Check wiki title	0.1312	**0.1354**	0.1328	0.1069	0.0526
11.1. Not in parallel wiki title	0.1279	0.1222	0.1199	0.0880	0.0482
11.1. + 12. Check tar. wiki title	0.1212	0.1168	0.1169	0.0874	0.0481
11.1. + 13. Check wiki article size	0.1215	0.1170	0.1172	0.0875	0.0481

Table 4.40.: F1-scores of the use of Wikipedia under different thresholds of En→Zh language pair

For the En→Zh language pair, from the results of the F1-score we can find that using Wikipedia also cannot always improve the performance of the method for finding the target sentence pairs. For the threshold of 1000 and 100, the use of Wikipedia even reduces the F1-score. For other thresholds, although the F1-score is improved after comparing the identified named entities and Wikipedia titles in the source language sentences (Step 10), the improvement is limited. For thresholds 15000, 10000 and 5000, there is almost no difference between the F1 score after step 10 and the F1-score after step 9. In addition to step 10, other steps using Wikipedia have lower F1-scores than step 10. Among all the steps using Wikipedia, the results of steps 10 is the best at each threshold. Among all the results of step 10, when the threshold value is 10000, the F1-score is the highest, which is 0.1354.

For En→Fr language pair, the results are completely different. From the F1-scores, it can be found that the use of Wikipedia has reduced the F1-score for each threshold. For

	15000	**10000**	**5000**	**1000**	**100**
7. Expl. word not in redund. part	6982/122	5400/112	3350/95	1388/64	639/37
8. NER (Stanza)	2152/99	1822/90	1333/76	790/48	404/27
9. Remove duplicate named entity	2151/99	1821/90	1332/76	790/48	404/27
10. Check wiki title	1047/38	870/35	628/29	316/16	93/6
11.1. Not in parallel wiki title	348/16	319/15	266/14	172/9	67/6
11.1. + 12. Check tar. wiki title	769/28	645/26	482/25	257/14	83/6
11.1. + 13. Check wiki article size	596/20	505/19	395/18	230/12	77/6

Table 4.41.: The results of the use of Wikipedia under different thresholds of En→Fr language pair

	15000	**10000**	**5000**	**1000**	**100**
7. Expl. word not in redund. part	0.0343	0.0405	0.0547	0.0848	0.0972
8. NER (Stanza)	0.0871	0.0926	0.1045	0.1053	0.1027
9. Remove duplicate named entity	0.0871	0.0926	0.1045	0.1053	0.1027
10. Check wiki title	0.0650	0.0706	0.0773	0.0731	0.0558
11.1. Not in parallel wiki title	0.0681	0.0680	0.0722	0.0612	0.0635
11.1. + 12. Check tar. wiki title	0.0629	0.0678	**0.0828**	0.0739	0.0585
11.1. + 13. Check wiki article size	0.0557	0.0606	0.0696	0.0682	0.0603

Table 4.42.: F1-scores of the use of Wikipedia under different thresholds of En→Fr language pair

each step using Wikipedia, as the threshold decreases, the F1-score first increases and then decreases. When the threshold is 5000, the F1-score of each step is the highest. When the threshold is 5000, step 11.1 and 12 have the highest F1-score, which is 0.0828.

After comparing the effects of different thresholds on steps using Wikipedia, based on the experimental results obtained, the most suitable threshold for each language pair is determined. The result is listed in Table 4.43.

Language pair	Threshold
En→De	5000
En→Fr	5000
En→Zh	10000

Table 4.43.: The most suitable threshold for the steps using Wikipedia of each language pair

4.3.4.3. Summary

Combining the above experimental results, an appropriate threshold can be determined for each language pair.

By testing the effect of different thresholds on the NER step, we can determine the most suitable NER model for each language pair, and give the most suitable threshold for the NER model. And by testing the effect of different thresholds on Wikipedia usage steps, we also determine the most appropriate threshold for each language pair. The results about the appropriate thresholds for the NER step are somewhat different from the results about the appropriate thresholds for the step Wikipedia usage, so the results of both need to be considered together to determine the final thresholds.

For some language pairs, the use of Wikipedia does not improve the performance of the method for finding target sentence pairs, but the use of Wikipedia can significantly reduce the workload of the final manual work. Therefore, on the basis of NER, Wikipedia should continue to be used to identify and select target sentence pairs.

For the En→De language pair, when the threshold is 5000 and after using Wikipedia, the highest F1-score is 0.2007. This F1-score is the highest among all experimental results, so 5000 is the most appropriate threshold for the En→De language pair.

For the En→Zh language pair, when the threshold is 10000 and Wikipedia is used, the highest F1-score is 0.1354. Although this F1-score is also the highest score among all experimental results, the corresponding number of remaining sentence pairs is 2302, which is still a huge challenge for the final manual work, so we choose 5000 as the most suitable threshold for En→Zh language pair.

For the En→Fr language pair, when the threshold is 5000, the highest F1-score after using Wikipedia is 0.0828. So for the En→Fr language pair, 5000 is considered the most appropriate threshold.

Therefore the final results of the threshold selection for each language pair are in Table 4.44. Based on the chosen threshold, the complete experimental results for each language pair can be given.

Language pair	Threshold
En→De	5000
En→Fr	5000
En→Zh	5000

Table 4.44.: The most suitable threshold of each language pair

Table 4.45 is the result for the En→De language pair. Table 4.46 is the result for the En→Zh language pair. The result for the En→Fr language pair is in Table 4.47. Based on the final results for each language pair, for all steps using Wikipedia, we also calculate the proportion of target sentence pairs among all remaining sentence pairs. The proportion results are in Table 4.48.

From the results in Table 4.48, we can find that the percentage results of each step using Wikipedia are not very different. Therefore, for each language pair, we take the average of the results of all steps using Wikipedia as the final proportion result of the target sentence pairs in the remaining sentence pairs.

After using Wikipedia (Steps 10-13), 13.71% (about 14%) of target sentence pairs with explanations can be found among the remaining sentence pairs for the En→De language pair. 4.91% (about 5%) of target sentence pairs with explanations can be found among the remaining sentence pairs for the En→Fr language pair. 8.28% (about 8%) of target sentence pairs with explanations can be found among the remaining sentence pairs for the En→Zh language pair.

Step	Numbers
Total	5000000
1. Check sou. word count (5000)	2587427
2. Sou. word has one alignment	2462009
3. Exists a redundant part	196021
4. word in redundant part no align.	5883
5. Check tar. word count (5000)	4405
6. Redundant part has punctuation	3253
7. Explained word not in redundant part	3102/134
8. NER (flair)	899/93
9. Remove duplicate named entity	791/93
10. Check wiki title	506/68
11.1. Not in parallel wiki title	247/33
11.1 + 12. Check tar. wiki title	395/57
11.1 + 13. Check wiki article size	323/44

Table 4.45.: The final results of En→De language pair

Step	Numbers
Total	5000000
1. Check sou. word count (5000)	2550061
2. Sou. word has one alignment	2270959
3. Exists a redundant part	374455
4. word in redundant part no align.	24762
5. Check tar. word count (5000)	16399
6. Redundant part has punctuation	7373
7. Explained word not in redundant part	7360/302
8. NER (HanLP)	2557/194
9. Remove duplicate named entity	2557/194
10. Check wiki title	1736/142
11.1. Not in parallel wiki title	1049/87
11.1 + 12. Check tar. wiki title	1086/87
11.1 + 13. Check wiki article size	1083/87

Table 4.46.: The final results of En→Zh language pair

Step	Numbers
Total	5000000
1. Check sou. word count (5000)	2426412
2. Sou. word has one alignment	2337267
3. Exists a redundant part	120709
4. word in redundant part no align.	11257
5. Check tar. word count (5000)	6882
6. Redundant part has punctuation	3355
7. Explained word not in redundant part	3350/95
8. NER (Stanza)	1333/76
9. Remove duplicate named entity	1332/76
10. Check wiki title	628/29
11.1. Not in parallel wiki title	266/14
11.1 + 12. Check tar. wiki title	482/25
11.1 + 13. Check wiki article size	395/18

Table 4.47.: The final results of En→Fr language pair

	En→De	En→Fr	En→Zh
10. Check wiki title	13.44%	4.62%	8.18%
11.1. Not in parallel wiki title	13.36%	5.26%	7.95%
11.1 + 12. Check tar. wiki title	14.43%	5.19%	9.01%
11.1 + 13. Check wiki article size	13.62%	4.56%	7.96%
Average	**13.71%**	**4.91%**	**8.28%**

Table 4.48.: The percentage results of steps using Wikipedia for each language pair

4.3.5. Exploration on named entities that need to be explained

Our experimental results show that, for each language pair, a certain number of target sentence pairs with explanations are found in the first five million sentence pairs of the corpus. For target sentence pairs found, the named entities in sentence pairs that need to be explained are continued to be explored. Our goal is to check whether each named entity that is explained in the found target sentence pairs also always needs to be explained in other sentences.

	En→De	**En→Fr**	**En→Zh**
11.1 + 13. Check wiki article size	323/44	395/18	1083/87

Table 4.49.: The result of the last step for each language pair

Experiments are based on the results of the step 11.1+13 for each language pair (Table 4.49). For each named entity found in step 11.1+13 that needs to be explained, all sentence pairs containing this named entity will be found in the first five million sentence pairs of the corpus, each sentence pair is then checked to see if it contains an explanation for the named entity. The results in Table 4.50.

	En→De	**En→Fr**	**En→Zh**
The number of named entities that are always explained	42/15	18/1	87/15
Proportion	35.71%	5.56%	17.24%

Table 4.50.: The result of named entities that need to be explained for each language pair

After removing duplicate named entities, for the En→De language pair, among the remaining 42 explained named entities, 15 named entities are found, which are also always explained in other sentence pairs. For the En→Fr language pair, of the remaining 18 interpreted named entities, only 1 named entity is found that always needs to be explained in other sentence pairs. For the En→Zh language pair, 15 of the remaining 87 explained named entities are found always to be explained in other sentence pairs.

We also consider another case, where a found explained named entity can be considered to be in need of explanation with a high probability if more than half (also including half) of the sentence pairs containing it have an explanation for it.

The results for the En→De language pair are in Figure 4.1. The results for the En→Fr language pair are in Figure 4.2. And the results for the En→Zh language pair are in Figure 4.3. For the En→De language pair, 20 named entities out of 42 have a high probability of being explained, accounting for 48%. For the En→Fr language pair, only 3 named entities out of 18 are found with a high probability of needing explanation, accounting for 17%. For the En→Zh language pair, 25 named entities out of a total of 78 named entities are found with a high probability of needing explanation, accounting for 32%.

The exploration results for named entities that need to be explained show that not all found named entities always require explanation. This also brings inspiration and help for subsequent method optimization and model training.

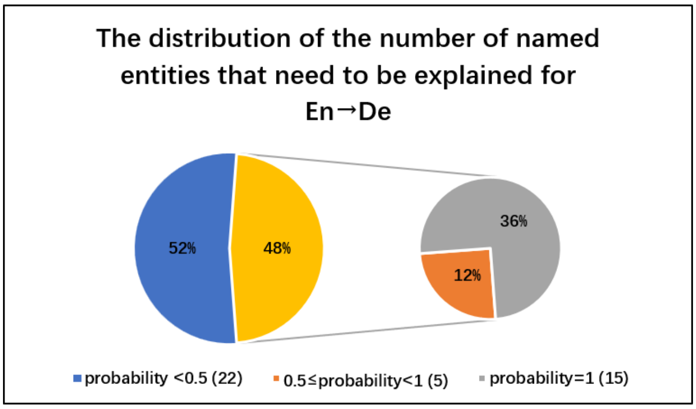


Figure 4.1.: The distribution Results of high-probability named entities requiring explanation for the En⟶De

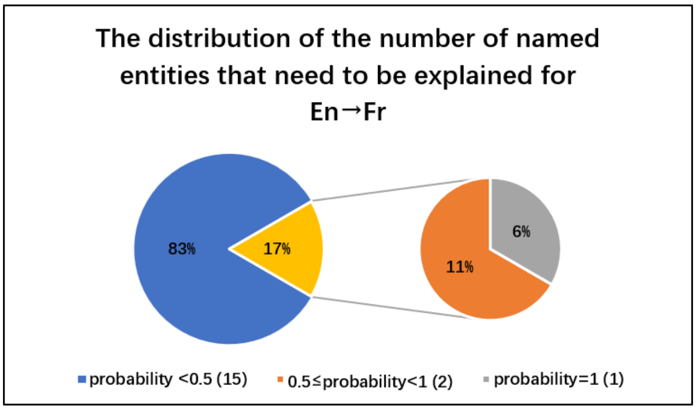


Figure 4.2.: The distribution Results of high-probability named entities requiring explanation for the En⟶Fr

4.3.6. Test

Our experiments are all done on the first 5 million sentence pairs of the corpus. When the first five million sentence pairs of the corpus are used as input, our proposed method of identifying and finding target sentence pairs with explanations can reduce the number of sentence pairs that need to be manually selected to a small value. On the En⟶De language pair, our method achieves the best performance. For simplicity, we can just check the result of the last step of the method (Steps 11.1+13). In the last step, only 323 sentence pairs are left for the En⟶De language pair, which is the smallest among the three language pairs. Among the remaining 323 sentence pairs, 44 target sentence pairs with explanations can be found, accounting for 13.62%, which is also the highest among the three language pairs.

The experimental results on the first five million sentence pairs of the corpus prove that our proposed method can greatly improve the efficiency of finding target sentence pairs. In order to verify the general effectiveness of our method, what we need to do is

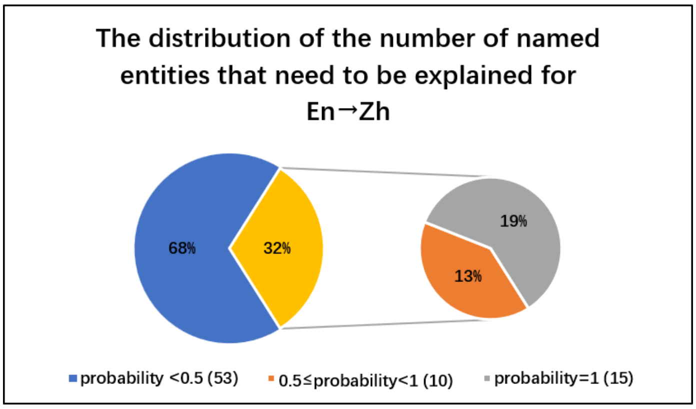


Figure 4.3.: The distribution Results of high-probability named entities requiring explanation for the En→Zh

to test the effect of our method on other inputs. Therefore, we will randomly select five million sentence pairs from all remaining sentence pairs in the corpus except for the first 5 million sentence pairs. Then, these randomly selected sentence pairs are used as input, and another experiment for testing is performed using the optimal settings and parameters (Table 4.51) obtained before. In order to avoid accidental errors, we will conduct five experiments for testing for each language pair.

Setting	**En→De**	**En→Fr**	**En→Zh**
Source word count threshold	5000	5000	5000
Target word count threshold	5000	5000	5000
NER tool	flair	Stanza	HanLP

Table 4.51.: Optimal settings and parameters for all language pairs

The results for the En→De language pair are in Table 4.52. The results in Table 4.45 are used as the baseline. From Table 4.52, it can be found that the results of the five experiments are very similar. But if we compare the experimental results in Table 4.52 with the baseline, we can find that starting from the NER step, the value of each experiment in Table 4.52 is about ten times the value of the baseline. For example, the number of remaining sentence pairs after step NER in the baseline is 899, while the number of sentence pairs after step NER in the first new experiment is 8844. After using Wikipedia, the number of remaining sentence pairs in step 11.1 of the baseline is 247, while the corresponding number in the first new experiment is 2049, which is 8 times that of the baseline.

The results for the En→Fr language pair are in Table 4.53, the baseline is the results from Table 4.47. And the results for the En→Zh language pair are in Table 4.54, the baseline is the results from Table 4.46.

The test results for these two language pairs are similar to those for the En→De language pair. Comparing the experimental results with the baseline, the same problem can be found, that is, in some steps, there is a non-negligible gap between the experimental results and the baseline results. For example, for the En→Fr language pair, the number of sentence

Step	**1. Exp.**	**2. Exp.**	**3. Exp.**	**4. Exp.**	**5. Exp.**
Total	5000000	5000000	5000000	5000000	5000000
1. Check sou. word count (5000)	2857954	2859559	2859992	2858722	2858396
2. Sou. word has one alignment	2614173	2615316	2616458	2613206	2613989
3. Exists a redundant part	310971	309736	309113	310419	310749
4. word in redundant part no align.	33250	33270	33267	33270	33089
5. Check tar. word count (5000)	26323	26246	26314	26392	26152
6. Redundant part has punctuation	18603	18629	18452	18619	18310
7. Explained word not in redun. part	17542	17560	17370	17548	17271
8. NER (flair)	8844	8958	9000	8947	8936
9. Remove duplicate named entity	7163	7200	7269	7166	7176
10. Check wiki title	4631	4670	4658	4625	4619
11.1. Not in parallel wiki title	2049	2056	2076	1956	2053
11.1 + 12. Check tar. wiki title	3538	3588	3620	3524	3543
11.1 + 13. Check wiki article size	2836	2858	2882	2801	2832

Table 4.52.: The testing results of En→De language pair

Step	**1. Exp.**	**2. Exp.**	**3. Exp.**	**4. Exp.**	**5. Exp.**
Total	5000000	5000000	5000000	5000000	5000000
1. Check sou. word count (5000)	2950000	2947466	2948355	2950066	2948501
2. Sou. word has one alignment	2709596	2706253	2707623	2709966	2708019
3. Exists a redundant part	344527	343891	343903	344371	346172
4. word in redundant part no align.	67247	67507	67482	67264	67845
5. Check tar. word count (5000)	41851	41855	41901	41917	42300
6. Redundant part has punctuation	22550	22565	22688	22691	22708
7. Explained word not in redun. part	22525	22536	22664	22662	22679
8. NER (Stanza)	12419	12370	12477	12581	12423
9. Remove duplicate named entity	12412	12364	12468	12574	12420
10. Check wiki title	6457	6397	6395	6511	6371
11.1. Not in parallel wiki title	2655	2700	2683	2693	2715
11.1 + 12. Check tar. wiki title	4905	4965	4870	5006	4890
11.1 + 13. Check wiki article size	4049	4104	4051	4142	4051

Table 4.53.: The testing results of En→Fr language pair

Step	1. Exp.	2. Exp.	3. Exp.	4. Exp.	5. Exp.
Total	5000000	5000000	5000000	5000000	5000000
1. Check sou. word count (5000)	2609791	2608485	2609799	2608767	2609678
2. Sou. word has one alignment	2237130	2236433	2236878	2235205	2237337
3. Exists a redundant part	467770	466875	467348	466226	466101
4. word in redundant part no align.	48174	48248	48237	48128	48131
5. Check tar. word count (5000)	33117	32961	32967	33073	33003
6. Redundant part has punctuation	16253	16003	16215	16184	16303
7. Explained word not in redun. part	16218	15972	16192	16148	16267
8. NER (HanLP)	7374	7350	7418	7379	7451
9. Remove duplicate named entity	7372	7350	7418	7379	7451
10. Check wiki title	5113	5079	5083	5156	5143
11.1. Not in parallel wiki title	2986	2990	3007	3023	3019
11.1 + 12. Check tar. wiki title	3120	3114	3120	3154	3155
11.1 + 13. Check wiki article size	3110	3105	3113	3143	3149

Table 4.54.: The testing results of En→Zh language pair

pairs remaining after the NER step of the baseline is 1333, while the number of sentence pairs remaining after the NER step of the first experiment is 12419, which is about 10 times the baseline result. If we consider the steps of using Wikipedia, there is also a 10 times gap between the experimental results and the baseline results. The same gap can also be observed in steps using wikipedia. For example, the first experimental result of steps 11.1 and 13 is 4049, which is about ten times the baseline result of 395.

For the En→Zh language pair, the observed gap between the experimental results and the baseline results is not so large. However, this gap cannot be ignored also. After the NER step, the number of remaining sentence pairs in the first experiment is 7374, which is about three times the baseline result of 2557. If considering the results of steps 11.1 and 13, The result of the first experiment is 3110, which is also about three times the baseline result of 1083.

In the results of all the three language pairs, for steps using NER and using Wikipedia, the gap in the number of remaining sentence pairs between the baseline results and experimental results is too large to ignore. This problem will introduce a huge challenge to the final manual selection work for all language pairs.

In order to check the proportion of target sentence pairs with explanations in the remaining sentence pairs, We also check the number of target sentence pairs among the last remaining sentence pairs for each language pair. We select the results of the last step (i.e. steps 11.1 and 13) of the fifth experiment for each language pair for validation. The results are in Table 4.55.

From the results in the Table 4.55, it can be found that for the En→De language pair, the number of sentence pairs left in step 11.1+13 is 2832. A total of 294 sentence pairs can be found in these 2832 sentence pairs that contain explanations. For the En→Fr language pair, 334 sentence pairs with explanations can be found among the remaining 4051 sentence

	En→De	**En→Fr**	**En→Zh**
11.1 + 13. Check wiki article size	2832/294	4051/334	3149/233
Proportion	10.38%	8.24%	7.40%
Baseline:			
Proportion_Exp_Step 11.1+13.	13.62%	4.56%	7.96%
Proportion_Exp_Aver.	13.71%	4.91%	8.28%

Table 4.55.: The percentage results of testing for each language pair

pairs. For the En→Zh language pair, there are a total of 3149 remaining sentence pairs, and 233 sentence pairs with explanations can be found from the remaining sentence pairs.

We calculate the proportion of the target sentence pair in the remaining sentence pairs for each language pair, and compare the calculated results with the baseline results (Table 4.48). For the En→De language pair, among the remaining sentence pairs, 10.38% target sentence pairs can be found. This is lower than the baseline results. In the baseline results, the result of the same step is 13.62%, which is about 3.2% higher than the experimental result. And the average result of baseline is 13.71%, which is about 3.3% higher than the experimental result. Although this percentage result of the experiment is lower than the baseline result, it is also higher than 10%, which is an acceptable result for us.

For the En→Fr language pair, among the remaining sentence pairs, 8.24% target sentence pairs can be found. Surprisingly, the experimental result is much higher than the baseline results. In the baseline results, the result of the same step is 4.56%, which is about 3.7% lower than the experimental result. And the average result of baseline is 4.91%, which is about 3.3% lower than the experimental result. This shows that our method can also find target sentence pairs efficiently on the En→Fr language pair. Among the last remaining sentence pairs, more than 5% of the target sentence pairs can be found.

For the En→Zh language pair, among the remaining sentence pairs, 7.40% target sentence pairs can be found. The experimental result is very similar to the baseline results. In the baseline results, the result of the same step is 7.96%, which is very close to the experimental result. And the average result of baseline is 8.28%, which is only 0.88% higher than the experimental result. This shows that for the En→Zh language pair, the experimental result is consistent with the baseline results, and our method can also efficiently find the target sentence pair for the En→Zh language pair.

The results for the proportion of target sentence pairs for all language pairs show that our method is robust. Although there is a gap in the number of remaining sentence pairs between the baseline results and the experimental results, this gap will bring some troubles to finding the target sentence pairs, but the robustness of our method can ensure that we can find a sufficient number of target sentence pairs from the last remaining sentence pairs, and then build the training dataset.

5. Conclusion

In this chapter, we summarize our work and present the conclusions. In addition, we also discuss future research possibilities. In Section 5.1 we first answer the research question, and then in Section 5.2 we discuss possible improvements and extensions of this thesis.

5.1. Answers to Research Questions

A research questions is proposed in Section 1.2. Based on the experimental results we can answer this research question as follows.

- **Research Question 1:** How to build a training dataset containing translation examples with explanation?

We propose a heuristic method to find target sentence pairs with explanations. In this method, three tools and data are used: word alignment, named entity recognition, and Wikipedia. We conduct experiments on three language pairs: English→German, English→French and English→Chinese. For all three language pairs, the source language is English. The experimental results show that for each language pair, our proposed method can reduce the number of remaining sentence pairs that may contain explanations to an extremely low number. Moreover, our method is robust, among the remaining sentence pairs, a certain proportion of target sentence pairs can always be found for each language pair.

In our experiment, the total number of input sentence pairs is five million. For the English→German and English→French language pairs, the number of the last remaining sentence pairs can be controlled within 500, while for the English→Chinese language pair, the number of the last remaining sentence pairs is around 1000. For the English→German language pair, among the last remaining sentence pairs we can find about 14% of the target sentence pairs with explanations. For English→French, about 5% of the target sentence pairs can be found in the remaining sentence pairs. And for English→Chinese, about 7% of the target sentence pairs can be found in the remaining sentence pairs. This shows that our proposed method can greatly reduce the manual work of finding target sentence pairs while also effectively improving the efficiency of finding target sentence pairs.

We also test our proposed method on the input of 5 million random sentence pairs. The results of the test show that for the English→German language pair, we can find more than 10% of the target sentence pairs in the last remaining sentence pairs. And for the English→French language pair, about 8% of the target sentence pairs can be found in the remaining sentence pairs. Meanwhile, for the English→Chinese language pair, about 7% of the target sentence pairs can be found in the remaining sentence pairs.

Based on the results about the proportion of target sentence pairs, we can conclude that our proposed method is robust. Among the remaining sentence pairs, more than 10% of the target sentence pairs can be found for the English→German language pair, more than 7% of the target sentence pairs can be found for the English→Chinese language pair, and for the English→French language pair, more than 5% of the target sentence pairs can be found.

Although there is a huge gap between the test and experimental results about the number of the remaining sentence pairs. This gap will bring some challenges to the building of the training dataset. But the robustness of our method can well overcome these challenges. Therefore, we can find a sufficient number of target sentence pairs to build a training dataset.

5.2. Future Work

The first thing to do is to use our method to find a sufficient number of target sentence pairs so that we can build a training dataset. If the training dataset can be successfully built, then we can find and train a suitable model that can accurately predict which words need to be explained during the translation process.

Moreover, in our experiments, the source language of all the language pairs is English. We did not try another direction, i.e. the target language is English. Trying more language pairs and extending the experiment to bidirectional within each language pair will also bring greater improvement to our work.

The last thing we can do is to use the wikification tool to link the words and phrases that need to be explained to the corresponding Wikipedia pages. In our current work, we want to find a model that can predict which words need to be explained during translation. It would make our work even better if words or phrases that need to be explained are found during the translation process and their explanations could be added into the translation results.

Bibliography

[1] Alan Akbik et al. "FLAIR: An easy-to-use framework for state-of-the-art NLP". In: *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*. 2019, pp. 54–59.

[2] Waleed Ammar. *wikipedia-parallel-titles*. `https://github.com/clab/wikipedia-parallel-titles.git`. 2015.

[3] Giusepppe Attardi. *WikiExtractor*. `https://github.com/attardi/wikiextractor`. 2015.

[4] Marta Bañón et al. "ParaCrawl: Web-scale acquisition of parallel corpora". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2020, pp. 4555–4567.

[5] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, Inc., 2009.

[6] Piotr Bojanowski et al. "Enriching word vectors with subword information". In: *Transactions of the association for computational linguistics* 5 (2017), pp. 135–146.

[7] Peter F Brown et al. "A statistical approach to machine translation". In: *Computational linguistics* 16.2 (1990), pp. 79–85.

[8] Peter F Brown et al. "The mathematics of statistical machine translation: Parameter estimation". In: (1993).

[9] Ludovic Denoyer and Patrick Gallinari. "The wikipedia xml corpus". In: *ACM SIGIR Forum*. Vol. 40. 1. ACM New York, NY, USA. 2006, pp. 64–69.

[10] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[11] Zi-Yi Dou and Graham Neubig. "Word alignment by fine-tuning embeddings on parallel corpora". In: *arXiv preprint arXiv:2101.08231* (2021).

[12] Chris Dyer, Victor Chahuneau, and Noah A Smith. "A simple, fast, and effective reparameterization of IBM model 2". In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2013, pp. 644–648.

[13] Angela Fan et al. "Beyond english-centric multilingual machine translation". In: *The Journal of Machine Learning Research* 22.1 (2021), pp. 4839–4886.

[14] Manaal Faruqui et al. "WikiAtomicEdits: A multilingual corpus of Wikipedia edits for modeling language and discourse". In: *arXiv preprint arXiv:1808.09422* (2018).

[15] Ralph Grishman and Beth M Sundheim. “Message understanding conference-6: A brief history”. In: *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*. 1996.

[16] Jiabao Guo. *Open Chinese Convert (OpenCC)*. `https://github.com/BYVoid/OpenCC.git`. 2022.

[17] Han He and Jinho D. Choi. “The Stem Cell Hypothesis: Dilemma behind Multi-Task Learning with Transformer Encoders”. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 5555–5577. URL: `https://aclanthology.org/2021.emnlp-main.451`.

[18] Marti A. Hearst et al. “Support vector machines”. In: *IEEE Intelligent Systems and their applications* 13.4 (1998), pp. 18–28.

[19] Matthew Honnibal et al. “spaCy: Industrial-strength Natural Language Processing in Python”. In: (2020). DOI: `10.5281/zenodo.1212303`.

[20] Masoud Jalili Sabet et al. “SimAlign: High Quality Word Alignments without Parallel Training Data using Static and Contextualized Embeddings”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*. Online: Association for Computational Linguistics, Nov. 2020, pp. 1627–1643. URL: `https://www.aclweb.org/anthology/2020.findings-emnlp.147`.

[21] Philipp Koehn. “Europarl: A parallel corpus for statistical machine translation”. In: *Proceedings of machine translation summit x: papers*. 2005, pp. 79–86.

[22] Philipp Koehn. *Statistical machine translation*. Cambridge University Press, 2009.

[23] Saul A Kripke. *Naming and necessity*. Harvard University Press, 1980.

[24] John Lafferty, Andrew McCallum, and Fernando CN Pereira. “Conditional random fields: Probabilistic models for segmenting and labeling sequence data”. In: (2001).

[25] Jing Li et al. “A survey on deep learning for named entity recognition”. In: *IEEE Transactions on Knowledge and Data Engineering* 34.1 (2020), pp. 50–70.

[26] Julie Beth Lovins. “Development of a stemming algorithm.” In: *Mech. Transl. Comput. Linguistics* 11.1-2 (1968), pp. 22–31.

[27] Ruixuan Luo et al. “PKUSEG: A Toolkit for Multi-Domain Chinese Word Segmentation.” In: *CoRR* abs/1906.11455 (2019). URL: `https://arxiv.org/abs/1906.11455`.

[28] Eliza Margaretha and Harald Lüngen. “Building linguistic corpora from Wikipedia articles and discussions”. In: *Journal for Language Technology and Computational Linguistics* 29.2 (2014), pp. 59–82.

[29] Olena Medelyan et al. “Mining meaning from Wikipedia”. In: *International Journal of Human-Computer Studies* 67.9 (2009), pp. 716–754.

[30] David Milne and Ian H Witten. “Learning to link with wikipedia”. In: *Proceedings of the 17th ACM conference on Information and knowledge management*. 2008, pp. 509–518.

[31] David Nadeau and Satoshi Sekine. “A survey of named entity recognition and classification”. In: *Lingvisticae Investigationes* 30.1 (2007), pp. 3–26.

[32] Graham Neubig et al. “An unsupervised model for joint phrase alignment and extraction”. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. 2011, pp. 632–641.

[33] Graham Neubig et al. “Machine translation without words through substring alignment”. In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2012, pp. 165–174.

[34] Joel Nothman, James R Curran, and Tara Murphy. “Transforming Wikipedia into named entity training data”. In: *Proceedings of the Australasian Language Technology Association Workshop 2008*. 2008, pp. 124–132.

[35] Joel Nothman, Tara Murphy, and James R Curran. “Analysing Wikipedia and gold-standard corpora for NER training”. In: *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*. 2009, pp. 612–620.

[36] Joel Nothman et al. “Learning multilingual named entity recognition from Wikipedia”. In: *Artificial Intelligence* 194 (2013), pp. 151–175.

[37] Franz Josef Och and Hermann Ney. “A systematic comparison of various statistical alignment models”. In: *Computational linguistics* 29.1 (2003), pp. 19–51.

[38] Chris D Paice. “Another stemmer”. In: *ACM Sigir Forum*. Vol. 24. 3. ACM New York, NY, USA. 1990, pp. 56–61.

[39] Georgios Petasis et al. “Automatic adaptation of Proper Noun Dictionaries through cooperation of machine learning and probabilistic methods”. In: *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*. 2000, pp. 128–135.

[40] Martin F Porter. “An algorithm for suffix stripping”. In: *Program* 14.3 (1980), pp. 130–137.

[41] Martin F Porter. *Snowball: A language for stemming algorithms*. 2001.

[42] Peng Qi et al. “Stanza: A Python natural language processing toolkit for many human languages”. In: *arXiv preprint arXiv:2003.07082* (2020).

[43] Samuel Reese et al. “Wikicorpus: A word-sense disambiguated multilingual wikipedia corpus”. In: *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC’10)*. 2010.

[44] Radim Řehůřek and Petr Sojka. “Software Framework for Topic Modelling with Large Corpora”. English. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. `http://is.muni.cz/publication/884893/en`. Valletta, Malta: ELRA, May 2010, pp. 45–50.

[45] Dan Roth et al. “Wikification and Beyond: The Challenges of Entity and Concept Grounding.” In: *ACL (Tutorial Abstracts)* 7 (2014).

[46] Holger Schwenk et al. “Ccmatrix: Mining billions of high-quality parallel sentences on the web”. In: *arXiv preprint arXiv:1911.04944* (2019).

[47] Ilya Shnayderman et al. “Fast end-to-end wikification”. In: *arXiv preprint arXiv:1908.06785* (2019).

[48] Jörg Tiedemann. “Parallel data, tools and interfaces in OPUS.” In: *Lrec*. Vol. 2012. Citeseer. 2012, pp. 2214–2218.

[49] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).

[50] Ledell Wu et al. “Scalable zero-shot entity linking with dense entity retrieval”. In: *arXiv preprint arXiv:1911.03814* (2019).

[51] Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. “The united nations parallel corpus v1. 0”. In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*. 2016, pp. 3530–3534.

A. Appendix