SECOND LANGUAGE LEARNING FROM NEWS WEBSITES

First Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Second Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Abstract

Learning a second language is difficult and requires constant revision and immersion. Fortunately, many of us take the time to update ourselves through reading news on a daily basis. In this paper, we merge both of these goals into a Web browser extension that allows a reader to learn and master vocabulary items. We conducted a user survey to evaluate our system against user requirements collected through an earlier survey. Since we find a word's context to be useful in learning a vocabulary, we further adopt word sense disambiguation (WSD) technique to show the best translation for each word in the context. Our proposed WSD method, leveraging the extension of standard machine translation system, significantly betters baseline methods in both coverage and accuracy.

1 Introduction

People read news every day. With the increasing popularity of portable devices and computers, more and more people are reading news from news websites (**?**).

We leverage on this culture to provide users of news websites with an opportunity to learn a second language. Learning a new language from language learning websites is very time consuming. To this end, we propose a system to enable visitors to news websites to efficiently learn a new language while they are reading news. We propose to build the system as a Chrome extension which would run on the client side when readers visit news websites from preconfigured list.

Learning a new vocabulary is the most time consuming and boring part of language learning[1]. Perhaps, this justifies the poor adoption of current second language learning systems. We, therefore, focus on enabling language learners build their vocabulary efficiently while providing them with an enjoyable user experience.

2 Related Work

There are many existing language learning software, which, fall into two categories, learning by lessons and learning vocabularies. In the first category, lessons are purposefully designed to help user learn a foreign language in an easier way. Duolingo[2] is a popular websites in this category. For the second category users are guided to recite lists of words, or provided with a translation for their input word in the foreign language. Google Translate [3] stands out in this category. The service is available as desktop / mobile / web software including a chrome extension. Our chrome extension is different from all these existing tools. I mainly compare our system with the aforementioned two softwares. Each difference serves as a motivation for developing our extension.

"Duolingo is a free language-learning and crowdsourced text translation platform"[4]. Most people start to use Duolingo when they know a little or nothing about the new language. They starting from some basic lessons and improve step by step. However, our target audience include people who know nothing about the foreign language language as well as people with a who are fairly in the foreign language. We can not only help beginners learn a new language but also help them con-

[1] https://neltachoutari.wordpress.com/tag/vocabulary/
[2] https://www.duolingo.com/
[3] https://translate.google.com/
[4] http://en.wikipedia.org/wiki/Duolingo

tinue their learning by allowing them to practice their foreign language. There are also a lot articles with their translations in Duolingo, but all the articles and their translations are manually added by Duolingo or users from Duolingo. Therefore, parallel articles in Duolingo are old and limited. However, our chrome extension is always working even for those up to the minute news and our user can just practice their foreign language in their daily readings.

Google Translate: "Highlight or right-click on a section of text and click on Translate icon next to it to translate it to your language"[5]. Google Translate is a chrome extension that displays only the translation when user select a section, which can be a word, a phrase, a sentence or even a whole page. Our chrome extension will translate a single word only, and display the translation, following with the pronunciations and example sentences to help user understand and remember this word. Compared with our extension, Google Translate is more like an extension to help user understand the content of the page. Furthermore, our extension will display the most appropriate translation as it will refer to the context of the word.

As far as I know, Google Translate will not refer to the context while selecting a single word as the selected section is the only input.

Table 1: Summary of the differences

	Duolingo	Google Translate	Chrome Extension
Lessons	Yes	No	No
User's foreign language level	Low	Low-High	Low-High
Time consuming	Yes	No	No
Resource	Limited	Infinite	Infinite
Customizable	Yes	No	Yes
Link to External Dictionary	No	No	Yes

[5] http://en.wikipedia.org/wiki/Google_Chrome_Extensions

3 Algorithm

As we all know, one word may have multiple translations in another language, and our extension is expected to select the most appropriate one based on the context.We call such translation selection as cross-lingual word sense disambiguation (WSD).

WSD is an open problem in natural language processing and ontology, aiming at identifying the proper sense of a word (i.e. meaning) in a context, when the word has multiple meanings. However, the system that I have implemented is different from the traditional WSD. Normally, WSD is to identify its sense in its original language, but my system is to identify its sense in another language.

It has been proved by a lot of studies that context is the key to learning a new language. This further suggests that a good word sense disambiguation system is necessary for our Chrome Extension. It can help user remember vocabulary efficiently and is a key feature that differentiate our software from other language learning tools.

In this following, I describe four approaches that I have tried to accomplish WSD system, which is also my main progress in the second semester. The four approaches are:

- Frequency based: always selecting the most frequent translation (the baseline),
- Part-of-Speech Tag based: selecting the translation based on the Part-of-Speech Tag of the English word
- Translation based: Selecting the translation based on the result from existing Machine Translation systems
- Category based: Selecting the translation based on the category of the news article

3.1 Baseline

The simplest way to select a translation from the candidates is by random. However, the correctness of this method is very low, probably less than 20%, and is not a good baseline for other methods to compete with. Another simple idea is to always select the most commonly used translation. Luckily, when I crawled the

dictionary, Google Translate does provide usage frequency of each Chinese Translation. This turns out to be a much better result, and thus serves as a fair baseline method.

3.2 Part-of-Speech Tagger

As we all know, many English words have more than one Part-of-Speech (POS) tags and their Chinese translations in different POS may differ a lot. For example, the word "book" has two POS tags, noun and verb. If it is used as a noun, mostly it means a handwritten or printed work of fiction or nonfiction, which should be translated as "书", and mostly means to reserve if used as a verb, which should be translated as "预定". Therefore, if I can get the POS tag of the English word, it might help me identify its sense or the Chinese translation.

Table 2: Sample input/output of POSTagger

Input	Output	Tag Description
They are treating me like family now	They—PRP are—VBP treating—VBG me—PRP like—IN family—NN now—RB	IN — preposition or conjunction

Among all possible on-line sources, Stanford Log-linear Part-of-Speech Tagger (**?**) is the most stable and well performed Part-of-Speech Tagger, which is developed by The Stanford Natural Language Processing Group. Column one and column two of Table **??** is one sample input and output. However, the tag from Part-of-Speech Tagger is not the one we normally used. I need one more step to match the tag to its Part-of-Speech. Therefore, I followed the Part-of-Speech guidelines from the University of Pennsylvania (Penn) Treebank Tag-set. The last column of Table **??** is one example of the matching.

Algorithm **??** is the approach of using POSTagger. From Algorithm **??**, firstly, if the word "like" need to be translated, the algorithm will fetch all the Chinese translations as well as their Part-of-Speech tag from our dictionary. Secondly, the algorithm will send the original English sentence to Part-of-Speech Tagger, which is a Java package and has been wrapped into a server. After the client has got the output from the server, it will fetch the corresponding tag and match it to Part-of-Speech tag based on the guidelines mentioned above. Lastly, it will select the translations based on the POS.

In most cases, this algorithm is only a filter. There might be a few Part-of-Speech tags matched from the output and one POS tag might have a few corresponding Chinese translations. In this case, I will choose the translation with the highest frequency of use, which is actually a combination of POSTagger and baseline, so that this system is testable independently.

Algorithm 1 Part-of-Speech Tagger

Require: POS Tagger, Dictionary <English, Chinese, POS >, Input English
$Pairs < word, POS > \leftarrow POSTagger \leftarrow English$
if $EnglishWord \subset Dictionary.English$ **then**
  $TranslationList \leftarrow Dictionary.Chinese + Dictionary.POS$
  for $word \subset Pairs.word$ **do**
    if $word = EnglishWord$ **then**
      $POSResult \leftarrow Pairs.POS$
      $Break$
    end if
  end for
  for $POS \subset TranslationList.POS$ **do**
    if $POS = POSResult$ **then**
      $FinalResult \leftarrow TranslationList.Chinese$
      $Break$
    end if
  end for
end if
return FinalResult

3.3 Machine Translation

Since our target is to select the most appropriate translation based on the context, using existing Machine Translation (MT) systems is also a good approach, as all of them will certainly translate

Table 3: Example input/output of POSTagger

Input English	they are treating me like family now
English Word	like
Translation List	verb : 喜欢, 爱, 爱好, 待见, 好, 看上, 喜, 喜爱, 喜好 adjective : 一样, 似, 同, 相似 conjunction : 如同 noun : 类 preposition : 好像, 好比, 好以 adverb : 不啻, 若
Pairs <word,POS>	they—PRP are—VBP treating—VBG me—PRP like—IN family—NN now—RB
Final Result	好像

words based on the context.

There are mainly two kinds of MT systems. One is off-line Machine Translation systems, which are mostly not available as mostly they are build for internal usage. Luckily, NUS NLP group built one MT system before, and it has been wrapped into a server, so that I can use it as an on-line service. The other one is on-line MT system, which are wrapped as a server and open to public, such as Bing Translator or Google Translate. As only Bing Translator is free, I decide to try Bing Translator as well.

The first priority of choosing a MT system is its translation quality, if it can give me a result that nearly as good as a result from human translation, then the Chinese word that I generated from the MT system will have a high chance to be correct as well. However, after I tried both MT systems, the performance of the one from NLP group is worse than Bing Translator and the server is very unstable, I decide to use Bing Translator as my Machine Translation system.

3.3.1 Bing

Bing Translator, also called Microsoft Translator, is a on-line Machine Translation system that developed by Microsoft team with a cloud-based API that is conveniently integrated into multiple products, tools, and solutions. Table **??** is the sample input and output of Bing Translator.

Table 4: Example input/output of Bing Translator

Input	Output
including a 45-caliber pistol a pump shotgun and an ar-15 rifle	包括45 口径手枪 唧筒式猎枪和ar-15 步枪
they are asking for privacy at this time	在这个时候他们正在寻求隐私
possessing cartridges used exclusively by the military and carrying a firearm without a license	拥有只供军方使用的墨盒和携带火器的许可证

Algorithm 2 Bing Translator

Require: Bing Translator, Dictionary <English, Chinese>, Input English
$ChineseTranslation \leftarrow$
$BingTranslator \leftarrow English$
if $EnglishWord \subset Dictionary.English$ **then**
  $TranslationList \leftarrow Dictionary.Chinese$
  $MaxLength = 0$
  for $ChineseWord \subset TranslationList.Chinese$ **do**
    if $(ChineseWord \subset ChineseTranslation) \cap (MaxLength < ChineseWord.Length)$ **then**
      $FinalResult \leftarrow ChineseWord$
    end if
  end for
end if
return FinalResult

Algorithm **??** is the steps of using Bing Translator. The original English sentence is "including a 45-caliber pistol a pump shotgun and an ar-15 rifle" and "pump" is the word that we want to translate. Firstly, this algorithm will fetch all the Chinese translations from the database. Next, it will send the original English sentence to Bing Translator using the API provided by Microsoft and get the result that returned from Bing Translator. After that, for each Chinese translation, I will check

Table 5: Example input/output of Bing Translator

Input English	including a 45-caliber pistol a pump shotgun and an ar-15 rifle
English Word	pump
Translation List	verb:抽, 抽水, 打气, 唧, 唧筒, 套 noun:抽水机, 唧筒
Chinese Translation	包括45 口径手枪唧筒式猎枪和ar-15 步枪
Final Result	唧筒

whether this translation is a substring of the Bing Translator result. If there are a few translations that can match with the Bing Translator result, I will select the longest translation. If there are a few translations with the same length and all of them can match with the Bing Translator result, I will select the translation with the highest frequency of use. In this example, both "唧" and "唧筒" are the substrings of Bing Translator result. As "唧筒" have two characters and "唧" only have one character, this algorithm will take "唧筒" as the final result.

3.3.2 Bing+

Table 6: Another example of Bing Translator

Input English	as a result of the latest premier league broadcast rights deal all of its teams have made it into the worlds top 40 clubs
English Word	top
Translation List	顶部, 顶端, 顶, 颠, 盖, 极, 尖, 尖峰, 面, 上身, 头, 上面的, 最大的, 最高的, 盖
Chinese Translation	由于最新英超联赛转播的权交易所有其团队已经进入世界顶级40名俱乐部
Final Result	顶

Table **??** is another example of Bing Translator. In this example, "top" is the word that need to be translated. If we manually translated the word "top" based on the Bing Translator, I would say the correct translation should be "顶级". However, the word "顶级" is not covered by our dictionary, so the final result generated by Bing algorithm is "顶" as this is the only word that matches Bing Translator. Although the meaning of the word "顶" is almost the same as that of the word "顶级" and I, personally, would prefer to make "顶" as one of the correct result when I tried to evaluate this example, the word "顶级" is more accurate in this context. Therefore, is there any way to solve this problem and make the result more accurate? Yes, solving this problem requires our system to generate Chinese words that are not covered by our dictionary and the only way is to use Word Segmenter.

Table 7: Example of Stanford Word Segmenter

Input	Output
问题是你将要运行直赫顿到他说的第一修正案	问题, 是, 你, 将要, 运行, 直赫顿, 到, 他, 说的, 第一, 修正案
博士苏斯博士知道这将是他出版的最后一本书	博士, 苏, 斯, 博士, 知道, 这, 将, 是, 他, 出版, 的, 最后, 一, 本, 书
进入世界顶级40名俱乐部	进入, 世界, 顶级, 40, 名, 俱乐部

Text segmentation is the process of dividing written text into meaningful units, such as words, sentences, or topics. The term applies both to mental processes used by humans when reading text, and to artificial processes implemented in computers, which are the subject of natural language processing. I use Stanford Word Segmenter to do Chinese text segmentation. Stanford Word Segmenter is a open source Java package that developed by The Stanford Natural Language Processing Group. I wrapped it into a local server and Table **??** contains some example of the input and output of Stanford Word Segmenter.

Algorithm **??** is the approach of using Bing Translator together with Stanford Word Segmenter, and I would like to use Bing+ to represent this algorithm. Step one, two and three has

Algorithm 3 Bing+

Require: Bing Translator, Word Segmenter, Dictionary <English, Chinese>, Input English
$ChineseTranslation \leftarrow BingTranslator \leftarrow English$
$SegmentedChineseTranslation \leftarrow WordSegmenter \leftarrow ChineseTranslation$
if $EnglishWord \subset Dictionary.English$ **then**
  $TranslationList \leftarrow Dictionary.Chinese$
  $MaxLength = 0$
  for $ChineseWord \subset TranslationList.Chinese$ **do**
    if $(ChineseWord \subset ChineseTranslation) \cap (MaxLength < ChineseWord.Length)$ **then**
      $FinalResult \leftarrow ChineseWord$
    end if
  end for
  for $SegmentedWord \subset SegmentedChineseTranslation$ **do**
    if $FinalResult \subset SegmentedWord$ **then**
      $FinalResult \leftarrow SegmentedWord$
    end if
  end for
end if
return FinalResult

been described in the previous section as it is exactly the same as Bing approach. From Bing approach, this algorithm will generate "顶" as the result. After that, Bing+ approach will send the Chinese sentence returned from Bing Translator to Stanford Word Segmenter. Then, this algorithm will use the segmented word that contains the Bing result as a substring or equals to the Bing result as the final result. In this example, the final result of Bing+ is "顶级" which is the best result that can be generated from the result of Bing Translator and also a result that does not covered by our dictionary.

3.3.3 Bing++

Table **??** is another example of Bing+ Translator. In this example, "line" is the word that need to be translated and the correct translation based on

Table 8: Example input/output of Bing+

Input English	as a result of the latest premier league ... have made it into the worlds top 40 clubs
English Word	top
Translation List	顶部, 顶端, 顶, 颠, 盖, 极, 尖, 尖峰, 面, 上身, 头, 上面的, 最大的, 最高的, 盖
Chinese Translation	由于, 最新, 英, 超联赛, 转播, 的, 权... 进入, 世界, 顶级, 40, 名, 俱乐部
Final Result	顶级

Table 9: Another example of Bing+ Translator

Input English	but the world no 46 played one of his best matches crucially not crumbling when the finish line was in sight
English Word	line
Translation List	线, 线路, 路线, 系, 行列, 划线于, 衲, 排, 诗句, 纹, 线条, 衬, 衲
Chinese Translation	但, 世界, 没有, 46, 起, 最, 重要, 的, 是, 不, 崩溃, 在, 终点, 线上, 视线, 的, 时候, 他, 最, 好, 的, 比赛, 之一
Final Result	线上or 视线

Bing Translator should be "线上". The Bing approach will get "线" as the result. After the Bing approach got the result, the next step should be finding the segmented word that contains "线" as a substring. However, both word "线上" and word "视线" contains word "线" as a substring and, obviously, only the word "线上" is the correct result and "视线" is actually the translation for English word "sight". As both results generated by Bing+ approach are not covered by our dictionary, the algorithm cannot get the frequency of use information. As a result, Bing+ approach will randomly choose one word, so there is only 50% to get the correct result. Is there any way to solve this prob-

lem and always choose the correct result in this case? Yes, as long as the algorithm could get the word alignment information, it will know exactly how to match the Chinese words with those English words.

Table 10: Example input/output of Word Alignment

$Input_1$	dr seuss knew it would be the last book he published
$Output_1$	博士苏斯博士知道这将是他出版的最后一本书
$Output_1$	0:1—0:1 3:7—2:5 9:12—6:7 14:15—8:8 17:21—9:9 23:24—10:10 26:28—14:14 30:33—15:17 35:38—18:19 40:41—11:11 43:51—12:13
$Input_2$	the problem is youre going to run straight headon into the first amendment he said
$Output_2$	问题是你将要运行直赫顿到他说的第一修正案
$Output_2$	4:10—0:1 12:13—2:2 15:19—3:3 21:28—4:5 30:32—6:7 34:41—8:8 43:48—9:10 50:53—11:11 59:63—15:16 65:73—17:19 75:76—12:12 78:81—13:13
$Input_3$	this is something preschoolers deal with all the time
$Output_3$	这是学龄前儿童处理所有的时间
$Output_3$	0:3—0:0 5:16—1:1 18:29—2:6 31:39—7:8 41:43—9:10 45:47—11:11 49:52—12:13

Bitext word alignment or simply word alignment is the natural language processing task of identifying translation relationships among the words (or more rarely multiword units) in a bitext, resulting in a bipartite graph between the two sides of the bitext, with an arc between two words if and only if they are translations of one another. I use Bing Word Alignment API[6] developed by Microsoft team to get the word alignment from English to Chinese Simplified. Luckily, although this API only support very few sets of language pairs, English to Chinese Simplified is one of the few supported sets. Table **??** has some examples of input and output from Bing Word Alignment. The left column is the original English sentence, the column in the middle is the translated Chinese sentence and the right column is the word alignment information. For word alignment information, the colon separates start and end index, the dash separates the languages, and space separates the words. For example, in the second column, "0:1—0:1" means the word "dr" should match with word "博士" and "9:12—6:7" means the word "knew" should match with word "知道".

Table 11: Example input/output of Bing++

Input English	English Word	Dictionary Chinese	Chinese
state department spokeswoman jen psaki said that the allies had a long history of cooperation	top	...陈, 陈说, 称, 称述, 发表, 发言...	国家, 部言人, je国, 有, 作, 历史

Algorithm **??** is the approach of using Bing+ approach together with the Microsoft Bing Word Alignment. First few steps are exactly the same as Bing+ approach. In this example, "state" is the word that need to be translated. The result from Bing+ approach is "发言人", which is the translation of "spokeswoman", because the Chinese translation "发言" can be translated from both "state" and "spokeswoman". Then step five will send the original English sentence to Bing Word Alignment. Now, there will be two final results, one from Bing+ approach and the other one from Bing Word Alignment and the algorithm will choose the correct one from these two results. In this example, "state" will match with "国家" and the algorithm will choose "国家" as the final result as well.

One general question about this approach is that, since I can get the official word alignment from Bing Word Alignment approach, is Bing+ approach still useful? Table **??** contains some examples of Bing+ approach and Bing++ approach. From left to right, the four columns are original English sentence, Chinese translation, result

[6] https://msdn.microsoft.com/en-us/library/dn198370.aspx

Table 12: Some examples of Bing+ and Bing++

English	Chinese	Bing+	Bing++
oh the places youll go! rose to the bestseller list shortly after it was released in 1990 and continues to pop up there most every spring as high school and college grads transition to a new phase of life	哦你要去的地方！升至畅销书排行榜之后不久它于1990年被释放并继续弹出那里大多数每年春天为高中和大学毕业生过渡到人生的一个新阶段	春天	年春天
the darkness of the book is what makes the optimism credible nel said	这本书的黑暗是什么使乐观书可信nel说	书	本书
other seuss books have emerged since his death in 1991 but this was the last he had a hand in	自从他死后在1991年出现了其他苏斯博士的书，但这是他一只手在最后一次	最后	最后一次

from Bing+ approach and result from Bing++ approach. It is very obvious that the result from Bing+ approach is the substring of the result from Bing++ approach, but which one is better? As the purpose of our Word Sense Disambiguation system is to select the most appropriate translation based on the context, but Bing Translator is a bit too smart comparing with our purpose. Bing Translator will generate some Chinese words that cannot be translated from any of the English word but can make this sentence clear and smooth. In this case, our system will choose the short answer instead of the long answer. That's why in the Bing++ approach, I will keep the result both from Bing+ approach and Bing Word Alignment and choose the better one.

3.4 News Category

The word "interest" have two very different translations when it is used as a noun. One translation is "the feeling of a person whose attention, concern, or curiosity is particularly engaged by something", which should be translated as "兴趣". The other translation is "a share, right, or title in the ownership of property, in a commercial or financial undertaking, or the like", which should be translated as "利益". It is quite obvious that the second sense is mostly used in financial related topics. Therefore, if we can analyze the category of the original article and select the translation with the same category label, it might help disambiguate the word meaning.

Getting the category of the original news article is very simple. Most news websites have a manually assigned a category for each news article and in most cases, the category label is part of the URL. However, assigning a category for Chinese word is not simple. As we are dealing with news, it is good to obtain such information from Chinese news domain. I crawled 100 Chinese news articles in each category from Baidu News[7], making around 1000 news articles in total. After I got all the news articles, I send all the news articles to the Stanford Chinese Word Segmenter, and further calculate word document frequency under each category. For example, if word "interest" is found five times in article A and three times in article B, both article A and B are under "finance" category, then I will add two for category "finance" of word "interest" as it will be counted only once even it can be found multi times in one article. I will use "weight" to represent this value and "averageweight" is the average weight of all categories of one word. After that, I will normalize the weight and use Equation **??** and Equation **??** to assign categories for those Chinese words. Basically, the two equations means that, if this word can be found in at least ten different news articles and more than 80% of the articles are under the same category, then I will use this category for this word.

$$averageweight > 1 \tag{1}$$

$$threshold > 8 * averageweight \tag{2}$$

However, the categories in Chinese and English news are not the same. I manually aligned the categories and delete some categories when necessary. As shown in Figure **??**, on the left side, the 11 categories are from Chinese news website and on the right side, the eight categories are from English news website. During the alignment, I not only take the name of category into account, but also consider the semantics of the category.

Algorithm **??** is the steps of using news category. The English sentence is the original sentence and "picture" is the word that need to be translated. Firstly, the algorithm will fetch all the Chinese

[7] http://news.baidu.com/

wsd_4.jpg

Figure 1: Alignment between categories

translations for word “picture” and split them with comma. In this example, only the word “影片” has a category “entertainment”. Next, the algorithm will fetch the category of the English news article from the URL, which is also “entertainment”. In this case, the algorithm will use “影片” as the translation for word “picture”. If a few words shares the same category, the algorithm will choose the translation with the highest frequency of use.

4 Results

This project has two main parts, Chrome Extension and WSD system. The Chrome Extension part is a software development project and the best way to evaluate is to listen to users’ voice. The WSD system is a standard research problem and can be evaluated with ground-truth, reporting its performance by coverage and accuracy.

Table 13: Example input/output of News Cateogry

Input English	English Word	Dictionary.Chinese	C
duncan told cnns don lemon hes just painting a picture of urban street life with his lyrics	picture	... 相, 影, 影片(entertainment), 帧, 想象, 画...	en

4.1 Chrome Extension

There are a few standard aspects that can be evaluated from the Chrome Extension part, such as User Interface (UI) design, loading speed and the functionality. UI design and functionality are more related to front end, while the loading speed is highly correlated to the back end. As this project is a joint work, and I am responsible for the front end, I limit my focus to evaluate the UI design and functionality by surveying users. Also, as mentioned in the above chapters, we did a user requirement survey before we really start this project. From this survey, we roughly know our potential customers’ expectation and we need to check whether our Chrome Extension could satisfy them. I got 16 different responses, 15 of them are between 18 and 24, and 11 of them are professional in Chinese.

For the details of the survey questions and survey results, please refer to the Appendix. In this survey, I made some screen shots of our Chrome Extension and ask subjects about their opinions. Most of them think that replacing some words with their corresponding Chinese translation will not influence their normal reading, but they will feel a bit uncomfortable and prefer to read the original English articles. Based on their voice, I decide to highlight the original English words as default setting instead of replacing the English words with their Chinese Translations. Besides, most subjects think our Chrome Extension is nice and would like to try it when they are going to learn a new language.

4.2 WSD System

Our Word Sense Disambiguate System can be evaluated from two important aspects: coverage (i.e., is able to return a translation) and accuracy (i.e., the translation is proper). To this end, I manually annotate the ground truth. Each approach was evaluated right after I had implemented it, therefore, they was tested against a random but different set of recent news articles from CNN. Though the evaluation datasets are different, it is still fair to compare their results, as the size of all dataset is sufficiently large.

Firstly, we want our algorithm to return at least one result instead of blank. For POSTagger approach, if our dictionary do not cover the Part-of-Speech generated from Stanford POSTagger, the algorithm will return nothing. For News Category approach, as the algorithm will only assign categories for some of the Chinese translations and not all Chinese news categories can match with a English news category, so the algorithm sometimes will return nothing as well. For Bing+ and Bing++ approach, if none of the Chinese translations is the substring of the Bing result, the algorithm will return nothing. For Bing++ approach, if the word alignment information is phrase to phrase matching, for example, it may give a matching between "in order to" and its Chinese translation, the algorithm will return nothing. Alternatively, for all the listed algorithm listed above, they can always return the translation with the highest frequency of use, but in this case, we cannot know whether the result is generated from the algorithm itself or just the baseline. That's why I choose to return a blank instead of the translation with the highest frequency of use.

Table 14: Coverage for different approaches

	Cover	Coverage
Baseline	580/580	100%
POSTagger	510/580	88%
News Category	15/804	1.9%
Bing	801/1095	73%
Bing+	801/1095	73%
Bing++	987/1095	90%

Table **??** contains the coverage for different approaches. As the algorithm will try to translate some word only if it is covered by our dictionary, the coverage for Baseline is always 100%. The coverage for Bing, Bing+, Bing++ and POSTagger are roughly the same and all of them are acceptable. However, the coverage for News Category approach is only 1.9%. One reason is that when I set the threshold for assigning categories for Chinese word, I purposely make it very high to maximize the accuracy. If the accuracy is quite high, which means this approach is quite useful, then I will lower the threshold and find the balance point.

Secondly, we want our algorithm to be as accurate as possible, and the most ideal situation is that all the translation returned from the algorithm is the correct or the most appropriate translation in that context. When I evaluate the accuracy of these few approaches, I use a few news articles from CNN as the input data and manually select the most appropriate translation for all the output data. After that, I will compare the result from the algorithm and the result that I manually generated and get the accuracy.

Table 15: Accuracy for different approaches

	Correct	Accuracy
Baseline	400/580	69%
POSTagger	345/510	68%
News Category	5/15	30%
Bing	545/801	68%
Bing+	705/801	88%
Bing++	961/987	97%

Figure **??** contains the accuracy of all the approaches. The last column is the accuracy for News Category approach and it is only 30%. As mentioned in above Chapter, since the accuracy is very low, there is no need to lower the threshold and try to allocate more categories for Chinese words. The accuracy for Baseline is 69%, which is already a fairly hight accuracy. The accuracy for Bing and POSTagger is around 69% also, which is a bit lower than our expectation. The accuracy for Bing++ is 97% which I think is a very good result and it is already very hard to improve. Therefore, based on my test results,

Bing++ is the best approach among these five approaches.

Algorithm 4 Bing++

Require: Bing Translator, Word Segmenter, Word Alignment, Dictionary <English, Chinese>, Input English

$ChineseTranslation \leftarrow$
$BingTranslator \leftarrow English$
$SegmentedChineseTranslation \leftarrow$
$WordSegmenter \leftarrow ChineseTranslation$
$Pair < EnglishWord, ChineseWord > \leftarrow$
$WordAlignment \leftarrow English$
if $EnglishWord \subset Dictionary.English$ **then**
  $TranslationList \leftarrow Dictionary.Chinese$
  $MaxLength = 0$
  for $ChineseWord \subset TranslationList.Chinese$ **do**
    if $(ChineseWord \subset ChineseTranslation) \cap (MaxLength < ChineseWord.Length)$ **then**
      $FinalResult_1 \leftarrow ChineseWord$
    end if
  end for
  for $SegmentedWord \subset SegmentedChineseTranslation$ **do**
    if $FinalResult_1 \subset SegmentedWord$ **then**
      $FinalResult_1 \leftarrow SegmentedWord$
    end if
  end for
  for $Word \subset Pairs.EnglishWord$ **do**
    if $EnglishWord = Word$ **then**
      $FinalResult_2 \leftarrow Pairs.ChineseWord$
    end if
  end for
end if
return $FinalResult_1 \cup FinalResult_2$

Algorithm 5 News Category

Require: Dictionary <English, Chinese, category>, Input English, News URL
  if $EnglishWord \subset Dictionary.English$ **then**
    $TranslationList \leftarrow Dictionary.Chinese + Dictionary.category$
    $EnglishCategory \leftarrow URL.category$
    for $category \subset TranslationList.category$ **do**
      if $category = EnglishCategory$ **then**
        $FinalResult \leftarrow TranslationList.Chinese$
        $Break$
      end if
    end for
  end if
  return FinalResult