# Learning Second Languages from News Websites


## Abstract

Learning a second language is difficult and requires constant revision and immersion. Fortunately, many of us read news online everyday. In this paper, we propose a web browser extension that allows readers to learn a second language vocabulary while reading news online. To this end, we propose algorithms to disambiguate word sense and translate the words in new articles properly to target langauge. We find a machine translation based method significantly betters baselines in both coverage and accuracy. We also propose techniques for generating appropriate distractors for multiple-choice word mastery quizzes for assessing language learners. We conducted a user survey to evaluate our system.


## 1 Introduction

Learning a new language from language learning websites is time consuming. Research shows that regular practice, guessing, memorization (**?**) as well as immersion into real scenarios (**?**) hastens language learning process. Therefore, to make second language learning attractive and efficient we seek to interleave language leaning with a popular daily activity: online news reading. Further recent increase in the popularity of portable devices has made online news reading popular (Kathryn Zickuhr and Brenner, 2012). We leverage on this culture to provide users of news websites with an opportunity to learn a second language.

It is observed that language development in children begins with vocabulary acquisition before syntax and other aspects are learnt. One may extrapolate by assumption that vocabulary acquisition is perhaps the first step in language learning, in general. Research also shows that extensive reading builds up vocabulary. This, however, could make learning time consuming and boring, if the learner is uninterested in the reading material. We, therefore, seek to enable learners build their vocabulary efficiently with an enjoyable user experience.

Existing language learning software are either instruction driven or user driven. Duolingo[1] is a popular instruction driven system that teaches through structured lessons. Google Translate[2], on the other hand, allows you to learn by translating your inputs. Instruction driven systems demand dedicated learner time on a daily basis and are difficult to customise according to the learner's skill level. User driven systems, however, lack rigor since the system never tests learners periodically to hone their skills. We seek to seamlessly merge learning and assessment into the same user session. Our system is also able to adapt to the learner's skill during assessment. We propose a system to enable online news readers to efficiently learn a new language, Chinese (English), while they are reading English (Chinese) news. Learners are trained by providing translations to randomly chosen open domain words on the English (Chinese) news page. Learners are also assessed by replacing English (Chinese) words in the news article with their Chinese (English) translations and asked to translate them back given a choice of possible translations. In this paper, we propose algorithms: **(i)** for translating English (Chinese) words to Chinese (English) from news articles, **(ii)** for generating distractor translations for learner as-

[1] `https://www.duolingo.com/`

[2] `https://translate.google.com/`

sessment. The system, deployed as a Chrome extension[3] is triggered when readers visit a preconfigured list of news websites.

## 2 Methods

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 users easily learn a foreign language. Duolingo[4] is a popular websites in this category. For the second categoryusers are guided to recite lists of words, or provided with a translation for their input word in the foreign language.Google Translate [5] stands out in this category. The service is available as desktop / mobile / web software including a chrome extension. We mainly compare our system with the aforementioned two softwares. Table 1 summarises important differences between our system and all these existing tools. Each difference serves as a motivation for developing our extension.
"Duolingo is a free language-learning and crowdsourced text translation platform"[6]. 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 is a mix novice and intermediate level learners of the foreign language. We can not only help beginners learn a new language but also help them continue 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"[7]. 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.

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 |

### 2.1 Software Design

Based on the user study, we divided the extension into three components: translating, learning and testing. After user opened a news website, some words in the main content will be replaced by their translation from user's preferred foreign language, and this is what our translating component is doing. If user want to know more about the replaced word, he can simply move his mouse over the translation and a window will pop over to help user learn this word, and this is learning component. If user have encountered some word for a few times, we will generate some quiz for him and this is testing component.

#### 2.1.1 Translating

After the original web page, our chrome extension will fetch the content of the news and pass

[3]a software extension to the Chrome web browser
[4]`https://www.duolingo.com/`
[5]`https://translate.google.com/`
[6]`http://en.wikipedia.org/wiki/Duolingo`
[7]`http://en.wikipedia.org/wiki/Google_Chrome_Extensions`

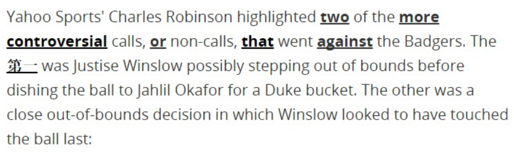


Figure 1: Screen shot of Translating Component

them to the server paragraph by paragraph. After receiving the content, server will compare every word in the paragraph with the words in our vocabulary. If there are some matches, which simply means there are some words that need to be replaced. As every English word might have a few Chinese meanings, our server must select the most appropriate translation among all the meanings. The way that we are trying to solve this problem so far is to compare all the Chinese meanings with the translation of the whole sentence from Bing Translate. If any of the Chinese meanings is the substring of the translation of the sentence, our server will choose that meaning (This is not a proper and accurate way to solve this problem, but it is much better than randomly choose one Chinese meanings. Also, this would be my main research problem that I need to solve next semester). Then, server will pass a JSON string that contains all the words that need to be replaced, their Chinese meanings as well as their pronunciations back to front end. Then, front end will replace the content of the news paragraph by paragraph, in which some words have been replaced. Figure 1 is the screen shot of this component.

### 2.1.2 Learning


Figure 2: Screen shot of popover with highlighted English word

By moving mouse on the Chinese word for one second, a window with its English meaning and


Figure 3: Screen shot of popover with highlighted Chinese word

pronunciation will pop over. Figure 2 is the screen shot of the pop over without its example sentence. If user want to know how to use this word, he can just click the button next to the pronunciation to get the example sentences of this word. After user click the button to get example sentence, our extension will send a request to server and wait for server' response. There is another way of doing this, which is simply get the example sentences together with the words in the Translating component. However, the example sentences contains much more characters comparing with the pronunciation. We want to maximize the loading speed and minimize the data transferred between front end and server, so we decided to split the pop over content into two request. Figure 3 is the screen shot of the pop over with its example sentences.

### 2.1.3 Testing


Figure 4: Screenshot of English test popover

When user has encountered the same word for a few times, our system will generate a quiz about this word for him. If user move mouse over this replaced word, a window with a quiz will pop over. After user select one option, this window will tell


Figure 5: Screen shot of Chinese test popover

user the correct answer and sent whether the answer is correct to server. Figure 4 is the screen shot of our testing popover in English. Figure 5 is the screen shot of our testing popover in Chinese.

## 3 Word Sense Disambiguation System

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).

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, getting the POS tag of the English word might help us identify its sense or the Chinese translation. We decide use Stanford Log-linear Part-of-Speech Tagger (Toutanova et al., 2003).

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.

### 3.3 News Category

The word “interest” have two very different translations when it is used as a noun. One translation is about “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 about “a share, right, or title in the ownership of property”, which should be translated as “利息”. It is quite obvious that the second sense is mostly used in financial related topics. Therefore, analysing the category of the original article and selecting the translation with the same category label might help disambiguate the word meaning.

In Table 2, word “picture” is the word that need to be translated. Firstly, the algorithm will fetch all the Chinese translations for word “picture” and 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

Table 2: Example input/output of WSD

| English Sentence | Word | Dictionary | Baseline | Category | Bing | Bing+ | Bing++ |
|---|---|---|---|---|---|---|---|
| ... treating me like family ... | like | verb : 喜欢, 爱... ... preposition : 好像, 好比 ... | 喜欢 | 好像 | | | |
| ... painting a picture of urban street life ... | picture | ... 相, 影, 影片(entertainment), 帧, 想象, 画 ... | | 影片 | | | |
| ... pistol a pump shotgun ... | pump | verb:抽, 抽水, 打气, 唧, 唧筒, 套 noun:抽水机, 唧筒 | | | 唧筒 | | |
| ... have made it into the worlds top 40 clubs ... | top | 顶部, 顶端, 顶, 颠, 盖, 极 ... | 顶部 | | 顶 | 顶级 | |
| state department spokeswoman ... | state | ...陈, 陈说, 称, 称述, 发表, 发言... | | | 发言 | 发言人 | 国家 |
| ... ... | | ... ... | | | | | |

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.

### 3.4 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 words based on the context. After I tried a few on-line or off-line MT systems, We decide to use Bing Translator as our Machine Translation system.

#### 3.4.1 Bing

In the example, 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 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.

#### 3.4.2 Bing+

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 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.

#### 3.4.3 Bing++

The Bing++ algorithm is basically 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.

### 3.5 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.

Table 3: Coverage for different approaches

| | Cover | Coverage |
|---|---|---|
| Baseline | 707/707 | 100% |
| POSTagger | 668/707 | 94.5% |
| News Category | 14/707 | 2.0% |
| Bing | 555/707 | 78.5% |
| Bing+ | 535/707 | 75.7% |
| Bing++ | 544/707 | 76.9% |

Table 8 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.

Table 4: Accuracy for different approaches

| | Correct | Accuracy |
|---|---|---|
| Baseline | 405/707 | 57.3% |
| POSTagger | 369/668 | 55.2% |
| News Category | 1/14 | 7.1% |
| Bing | 443/555 | 79.8% |
| Bing+ | 433/535 | 80.9% |
| Bing++ | 530/544 | 97.4% |

Figure 9 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.

## 4 Distractors Generation Algorithm

The key research topic here is to investigate a way to automatically generate suitable distractors for a certain vocabulary test. The distractors are generated in English form.

### 4.1 Collecting category-related words

To generate good category-related distractors, it is essential to gather enough words that are more related in a certain category to serve as distractors candidates.

To find good "category-related" words, it is essential to get the words from those already classified news articles. The process involves 3 steps, crawling news contents from popular news website, preprocessing, and classifying word category.

#### 4.1.1 Crawling news content

Several web crawlers are designed to get news content from different popular news websites.

The crawler will detect URLs from each news website's main page as and its sub-category pages. For example, there are sub-categories like "football", "basketball" under main category "Sports", and the crawler is able to crawl URLs from "football" page and "basketball" page as well.
After detailed comparison of most news websites, I divided news articles into seven categories, namely "World", "Technology", "Sports", "Entertainment", "Finance", "Health" and "Travel". Most news articles can be classified into one of the seven categories. The web crawler will store all paragraph tags from each websites and store them as one file under one category.
In this experiment, around 1400 news articles, i.e. 200 news articles from each category are crawled and stored locally. Post natural language processing is then applied to this category corpus.

#### 4.1.2 Preprocessing

After storing all the news articles document into each category, the server uses Natural Language Tool Kit (Edward, 2009) for word tokenizing and POS Tagging. The system will store the POS tag of each word. After elimination of all non-English words and those words that contain special symbols, like "O'Real", "S$40", all words that contains only alphabetic letters are conserved. All stop words are also eliminated as well. They are stored as lower case for the ease of future process.

#### 4.1.3 Classification

In statistical analysis step, the server counts the document frequency of each word in all those stored news articles, i.e. if word "scored" appeared 4 times in one article, it will only be counted as once. By following this approach we can successfully reduce the bias of some words only appear a lot of times in one article while don't appear often in other article. As we are storing similar number of articles in each category, this approach will provide a fair comparison of each word's popularity among different categories. After this step we will know the document frequency count of each word in different category. Assume C is the list of category names, and f(w, C(i))=m means word w appeared in category C(i) for m times, then the sum weight of word w as sw(w) is calculated in Equation 1:

$$sw(w) = \sum_{i=1}^{n} f(w, C(i)) \tag{1}$$

The average weight of word w as aw(w) is calculated in Equation 2::

$$aw(w) = sw(w)/n \tag{2}$$

A word w is classified into category C(i) if it satisfies Equation 3::

$$f(w, C(i)) - aw(w) >= \delta \tag{3}$$

The confidence factor $\delta$ can be a positive integer between 0 and the average number of articles in each category. It means on average, the word w must appear in a specific category C(i) $\delta$ times more than it appear in other category before it can be classified into category C(i).

In the example below in Figure 18, frequency counts for word "investment" in each category are displayed. It is obvious that sw("investment") = 2 + 1 + 2 + 10 + 3 + 2 + 1 = 21, thus aw("investment") = sw("investment")/7 = 3, in this case if we choose a confidence factor $\delta = 3$, word "investment" will be classified into category "Finance", as 10–3>$\delta$ = 3. However, it we choose a very big $\delta$, for example $\delta = 8$, then word "investment" will not be classified into any category.

Table 5: Example of classification word into category

| Category | Investment |
|---|---|
| Technology | 2 |
| World | 1 |
| Sports | 2 |
| Finance | 10 |
| Entertainment | 3 |
| Health | 2 |
| Travel | 1 |

It is obvious that a higher confidence factor value will result in less number words get classified, but it will result in getting words that are more accurate. A lower confidence factor value will result in more number of words get classified, but less accurate in each category. In this experiment, after several round of tests and analysis, we chose a confidence factor value of 10, which is capable of producing enough number of classified words while maintaining the accuracy.

### 4.2 Generating distractors

My selection strategy in choosing distractors takes following parameters:

- News website URL
- News sentence
- Word to test
- User's knowledge level of the word

#### 4.2.1 Detect news category

After getting the news URL, our system needs to determine the category of the news. Based on the analysis from most popular news URLs, there is a set of common identifiers that can identify the category of the news article. For example, technology news URL often contains "/tech", "/science", and if we find these strings in news URL, we will classify this news URL into "Technology" category. The algorithm will go through all category identifier in the list, and will return the category name the moment it finds a match. The current list of category provides reasonable accuracy for the purpose of detecting news category.

#### 4.2.2 Detect Part-Of-Speech Tag

Given the target word and the target sentence, it is easy to run the NLTK POS tagger to get the correct POS tag of this word. This step is essential to help select distractors with similar forms, i.e. if the target word is adjective, it will be appropriate to choose three other adjectives, not verbs, as distractors.

#### 4.2.3 Semantic Distance

Before we go to explain the next step, it is essential to introduce the semantic distance calculator we used in the server implementation.

The perspective of semantic relatedness or its inverse, semantic distance, is a concept that indicates the likeness of two words. It is more general than the concept of similarity as stated in WordNet's synset relation. Similar entities in WordNet are classified into same synset based on their similarity. However, dissimilar entries may also have a close semantic connection by lexical relationships such as meronymy (car-wheel) and antonymy (hot-cold), or just by any kind of functional relationship or frequent association(pencil-paper, penguin-Antarctica) (Alexander, 2001). Semantic distance calculator aims to calculate the semantic relatedness score between two words.

There are many approaches to calculate semantic relatedness score. In this application, we are using Lin Distance (Lin, 1998) to calculate the semantic distance between two concepts. The detail of Lin Distance methodology is explained as follows.

Lin attempted to define a measure of semantic similarity that would be both universal and theoretically justified. There are three intuitions that he used as a basis:

- The similarity between arbitrary objects A and B is related to their commonality; the more commonality they share, the more similar they are;
- The similarity between A and B is related to the differences between them; the more differences they have, the less similar they are.
- The maximum similarity between A and B is reached when A and B are identical, no matter how much commonality they share.

Based on the intuition above, Lin proposed his approach in measuring similarity between two concepts c1, c2 in Equation 4:

$$sim(c1, c2) = \frac{2 * log_p(lso(c1, c2))}{log_p(c1) + log_p(c2)} \tag{4}$$

where p(c) denotes the probability of encountering concept c, and lso(c1,c2) denotes the lowest common subsumer, which is the lowest node in WordNet hierarchy that is a hypernym of c1 and c2.

The distance calculator will return a score from 0 to 1, as can be easily seen from the formula above. If the score is closer to 1, it means the two words are closer in semantic sense. This distance calculator will play an important role in the following algorithm.

#### 4.2.4 Distractors Selection Algorithm

Based on the input parameters, at this stage the server has already got the current category of the news article and the correct POS tag of the target word to test. The server is going to generate distractors based on user's knowledge level of the target word to test.

Knowledge level is 1: This indicates that the user has just learnt this word. The algorithm will randomly select three words from current category's word list. The reason for using randomization is to avoid the situation that similar distractors are generated every time.

Knowledge level is 2: This indicates that the user has known this word for some times. The algorithm will randomly select two words from the current category's word list as two distractors. Then the algorithm will randomly select word from the current category's word list and calculated the semantic distance between the selected word and the target word, once the score is above certain threshold, the selected word will be chose as the third distractor. The selection of threshold value will have a direct effect on the speed of distractors generation process. As a very high threshold value will result in more rounds of calculation in semantic distance calculator, and it will take a long time before the distractors are returned to the front end. After several rounds of analysis of each category's words and the results returned from semantic distance calculator, the threshold value of 0.1 is selected.

Knowledge level is 3: This indicates that the user has a good understanding of the word already; the algorithm will choose distractors solely based on results returned from semantic distance calculator. Similar to the approach when knowledge level is 2, the algorithm will randomly select word from current category's word list and calculate the semantic distance between the selected word and the target word. If the score is above certain threshold, the selected word is chosen as one of the distractors. The process is continued until the server can find three distractors.

## 5 Evaluation

This project has two main research parts, Distractors Generation algorithm and WSD system. To evaluate the distractors selection strategy, 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.

### 5.1 Distractors Generation Algorithm

To evaluate the distractors selection strategy as described in this report, we chose the knowledge-based approach used by many other language learning systems, which is to utilize the WordNet data and selection distractors based on synonyms of synonyms. WordGap system uses this approach to generate vocabulary test for its android application.
In our implementation of the baseline algorithm, we will choose the most frequent used word w1 from the target word's synonym set, and select the most frequent used word w2 from word w1's synonym set. The selection process is continued until we can find 3 distractors to form a vocabulary test. However, if the number of valid result we can get is less than 3, we will choose the word that shares the same antonym with the target word.

#### 5.1.1 Designing Survey

To compare the two approaches in generating distractors, we designed several survey sets to ask users to compare the plausibility of distractors. We randomly selected 50 sentences from recent news articles and choose one noun or adjective inside the sentence as the target word to test. In the survey, participants are required to answer each question and rank the plausibility of all distractors from 1 to 7. The correct answer will be ranked as 1, and the least plausible distractor will be ranked as 7. A screenshot of one sample question is shown in Figure 6.

There are two evaluations to be done as follows:

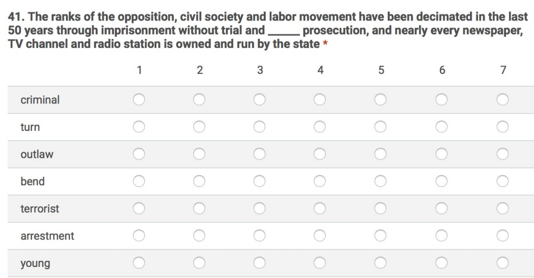


Figure 6: A sample survey question

1. Compare Baseline with Knowledge Level 1 Algorithm

. Compare Baseline with Knowledge Level 3 Algorithm For each comparison, three distractors are generated from the baseline algorithm; three distractors are generated from the stated algorithm in this report. With the first comparison we will be able to see if the category information will help in selecting more suitable distractors. By comparing the results from the both evaluation, we will be able to see if semantic distance and category information will help improve the suitability of distractors.

#### 5.1.2 Results

The evaluation contains 100 questions and is separated into 4 surveys, with each survey containing 25 questions. Each participant is free to choose one or more than one surveys. The purpose is to reduce the workload in each survey to get better responses. The surveys are sent to Year 1 students from School of Computing, National University of Singapore. There are 15 valid responses with each participant ranking each distractor with a different weight from 1 to 7. Half of the participants are native English speakers.

Each participant's rank will be the weight of the particular distractor in that question, i.e. if the user rank one distractor as rank "5", the weight of this distractor in this user's response will be 5. For each distractor of each question, the ranks of all users' responses are summed. As the more plausible the distractor is, the higher rank it will have, thus if the sum is higher, the approach is not as plausible as the other from user's point of view.

Table 6: Comparison 1 Baseline vs. Knowledge level 1 Algorithm

| | Number of winning questions | Average score |
|---|---|---|
| Baseline | 27 | 3.84 |
| Level 1 Algorithm | 23 | 4.10 |

Table 6 and Table 7 showed the detailed result of each comparison. If for any question, the sum of weight from all participants for one approach is bigger than the other, then this approach is considered to have won this question. The "average score" is the average sum of weight from each approach for all questions. The lower the average score is, the better performance this approach has gained.

Table 7: Comparison 2 Baseline vs. Knowledge level 3 Algorithm

| | Number of winning questions | Average score |
|---|---|---|
| Baseline | 21 | 4.16 |
| Level 3 Algorithm | 29 | 3.49 |

From Figure 6 we can see that in the first comparison, the baseline algorithm actually outscored the knowledge level 1 generation algorithm by 4 questions, with a sum of weight lower than 0.26. From Table 6 we can see that in the second comparison, the knowledge level 3 generation algorithm surpassed the baseline algorithm by 8 questions, with the average weight of 3.49 vs 4.16.

#### 5.1.3 Analysis

In knowledge level 1 generation algorithm, there is no semantic distance calculation involved. If the target word to test has no strong category indication, for example, words like "venue", "week", it is possible that the knowledge level 1 algorithm will select some distractors that are not as plausible as those coming from the target word's synonym of synonym.

However, this problem is solved with the help of semantic distance calculator. In the knowledge level 3 generation algorithm, the distractors chosen are both semantic close and also category-related, which produced a relatively better experiment result.

Also in the baseline algorithm, it is possible that it will select words that are very rare in real life (Susanne, 2013), which may also have influence in the result.

### 5.2 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 8: Coverage for different approaches

| | Cover | Coverage |
|---|---|---|
| Baseline | 707/707 | 100% |
| POSTagger | 668/707 | 94.5% |
| News Category | 14/707 | 2.0% |
| Bing | 555/707 | 78.5% |
| Bing+ | 535/707 | 75.7% |
| Bing++ | 544/707 | 76.9% |

Table 8 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 9: Accuracy for different approaches

| | Correct | Accuracy |
|---|---|---|
| Baseline | 405/707 | 57.3% |
| POSTagger | 369/668 | 55.2% |
| News Category | 1/14 | 7.1% |
| Bing | 443/555 | 79.8% |
| Bing+ | 433/535 | 80.9% |
| Bing++ | 530/544 | 97.4% |

Figure 9 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.

## 6 Conclusion

## References

Hirst Alexander, Budanitsky; Graeme. 2001. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures.

Jordan L Boyd-Graber, David M Blei, and Xiaojin Zhu. 2007. A topic model for word sense disambiguation. In *EMNLP-CoNLL*, pages 1024–1033.

Klein Edward, Loper; Ewan. 2009. Natural language tool kit.

Kristen Purcell Mary Madden Kathryn Zickuhr, Lee Rainie and Joanna Brenner. 2012. Younger americans' reading and library habits. *Pew Internet*.

Dekang Lin. 1998. An information-theoretic definition of similarity.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys*, 41(2):10:1–10:69, February.

Knoop; Sabrina Wilske Susanne. 2013. Wordgap - automatic generation of gap-filling vocabulary exercises for mobile learning.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 173–180, Stroudsburg, PA, USA. Association for Computational Linguistics.