Translated Word Similarities

Claire Wang
zw1511@nyu.edu

Kevinray Lu
zl1829@nyu.edu

Abstract

Translation inevitably adds more burden to similarity computing since meanings of words could be divided into chunks or be represented by punctuated expressions. Hence, simply comparing source words would not be enough for translated confusion term extractions. In order to provide an efficient way to deal with this problem, we propose a method to output similar Chinese translations of a given word within a word list when human resource is limited for checking.

1 Introduction

Computing translated word similarities should be divided into several phases. This paper will introduce that sentence parsing, chunk similarity computing, and aggregating are all needed after translating targeted English word into Chinese. Due to the complexity of Chinese language structure, two approaches will be taken to experiment for quality checking. The system takes an English word as input and output a list of similar words in terms of their Chinese meanings. Our goal is regarding to confusion terms extraction, which leads to various applications such as multiple choice question designs in vocabulary builder software.

To translate words from English to Chinese may be a risk if using translation software like Google Translation. From previous related work we knew that Google Translation may lose accuracy when coming to specific materials in terms of BLEU scores (Maxim Khalilov, 2012). Unfortunately, BLEU itself takes a "quantity leads to quality" approach, leading to possible misses of chunk translation accuracy in compensation (Kishore Papineni and Zhu, 2002). Therefore it is still challenging and time consuming to evaluate translation output. In order to ensure accuracy, we will take use of published dictionaries with a less amount of testing in return (about 2000 words). This will ensure the translation process does not add to uncertainties for next steps in similarity computations.

After translation, one simple word may be divided or represented by a list of, in our case, Chinese words. Those words could be of the same meanings, or partial representations of the original word. Mature word similarity computation libraries already exist, leading by packages like NLTK and Synonyms. However, even multilingual versions of these libraries are not able to accurately calculate similarities directly from a translation without chunking. We hence need to perform sentence parsing with great care since there are no natural spaces between characters in Chinese text (Rui Wang, 2015).

Computing chunk similarities in Chinese involves two strategies: word level comparison and character level comparison (Maoxi Li, 2011). Although the previous research mainly talked about machine translation comparisons, the same notations can be applied to our topic treating one of the chunks as reference. Character-level comparison yields higher matching scores, but it would cause bias introduced in Section 4. We will utilize Cilin Structure to compute exactly the distance from one word to another and calculated their similarities using weights provided by the structure assigned to word bag.

In Section 5 we are implementing two algorithms to get a final score for translation similarities. One of the approach includes a weighted sum of chunk similarities, with the other being instinctive but not always reliable. When the score passes a threshold, the two words are considered similar in Chinese. For evaluation, we are using mainly artificial way of checking and compare the two outputs we have to a well established English similarity checking library.

2 Problem Examples

The main idea of our work is to find out similar Chinese translations given an English word from a word list. For example, given a three-word-list translated by Oxford Dictionary:

word1: happy, Chinese: 感到（或显得）快乐的；高兴的

word2: cheerful, Chinese: 快乐的；高兴的；兴高采烈的

word3: gigantic, Chinese: 巨大的；庞大的

Happy and cheerful are synonyms in English, but after translation the two words are divided into different lengths of chunks, with the Chinese character “的" as a symbol for indicating adjectives. With common sub-strings “快乐" and “高兴" in Chinese translation, it is obvious that we should categorize those two expressions as confusion terms. It is however not possible for word similarity tools to find them out since neither of the two translations are considered “words". If we parse the translations for chunk comparisons, there will be chunks like “感到” in word1 and “兴高采烈" in word2 that have similarities very low which weakens overall similarity score, and “的" in all three words to unnecessarily increase similarities by providing no semantic meanings. The latter generates a risk of counting gigantic as a confusion term of happy or cheerful, which makes no sense.

Consequently, our algorithms introduced in Section 4 and Section 5 will include a stop words list, an optimal library for parsing, an algorithm to compare chunk similarities, and a strategy to combine chunk similarities by adding more weight on high-chunk-similarity scores. We will also talk about why we did not choose the highly preferred cosine similarity algorithm to achieve our goal.

3 Machine Translation vs. Dictionaries

In order to translate words from English to Chinese, there are two options. One is Machine Translations, and the second one is to directly cite authoritative dictionaries to map English words into Chinese explanations.

Machine Translation is a powerful tool in segment translations (usually sentences), the primary complication of which involves around choosing the suitable translation when a word can have more than one meaning and the correct orderings of the translated words to form a coherent sentence. For our purposes, we are seeking to translate separate English words, rather than segments of words. In this way, common Machine Translation tools, such as Google Translate, usually give a primary word translation, following with several high frequency translations appearing in public documents.

To use Machine Translations in our system, proper evaluation method on the translated result is necessary. However, evaluating the output of Chinese translation has always been a complicated topic, especially centering around whether to use a word-level evaluation or a character-level evaluation. The International Workshop on Spoken Language Translation in 2005 (IWSLT’2005) used the word-level BLEU metric (Kishore Papineni and Zhu, 2002), while IWSLT’08 and NIST’08 adopted character-level evaluation metrics to rank the submitted systems (Maoxi Li, 2011).

However, neither the word-level or the character-level BLEU metric applies very well to our purposes. The primary programming task for a BLEU implementer is to compare n-grams of the candidate with the n-grams of the reference translation and count the number of matches. These matches are position independent. The more the matches, the better the candidate translation is(Kishore Papineni and Zhu, 2002). However, when doing single word translation, the number of matches does not necessarily indicate the accuracy of translation. Namely, a small number of matches does not necessarily indicate low accuracy. For instance,

Word: cheerful

Reference: 快乐的；高兴的；兴高采烈的

Translation: 快乐，愉快，快活

where the reference is the Chinese definition for the word "cheerful" in Oxford Dictionary, and the translation is the output of Google Translate. Based on character-level BLEU evaluation, the Modified Unigram Precision = 1/3. Only the character " 快" and “乐” in the translation match those in reference. However, in terms of semantics, the word “愉快” is not that much different from “高兴”, and " 快活" has a similar meaning to " 兴高采烈", even they have completely different characters and appears to be completely different words. The Google Translate output captures the meaning of "cheerful" accurately, despite using different words and characters from the reference. This is because when we are translating a single word, there is no need to choose the correct translation from various meanings. Rather, we need to output all possible meanings. Without the restriction of a sentence's context, the wordings of the Chinese translations can vary, but with very similar meanings.

Because of the complexity inherent in Machine Translation evaluation, as well as the particularity of our system, we choose not to use Machine Translation to deal with the multilingual section, and to take use of Oxford Dictionaries for English word translation with about 2000 common English words.

4 Translations Parsing and Chunk Similarity Computation

4.1 Regexp jieba.lcut()

Our approach on tokenizing translations start with getting rid of all punctuation within the expression. We use the command as followed to extract only Chinese characters:

”.join(re.findall(r’[4̆e00-9̆fa5]’, targeted-translation))

After cleaning, Jieba library is used to tokenize the Chinese string. Jieba is a library dedicated on sentence parsing and the name means “saying sentence like a stammer" in Chinese. This library provides an efficient way to parse sentences into meaningful chunks for the following similarity computation. For the “happy" translation we have shown in Section 2, jieba.lcut() function will return, after cleaning:

[感到，或，显得，快乐，的，高兴，的]

This seems long with meaningless terms like “的” as a chunk, which adds work to erase them later. But it is necessary and more preferred to parsing other dividing strategies. If we divide the sentence according to punctuation, making the result [感到或显得快乐的, 高兴的], we would encounter the problem of doing virtually nothing since multiple chunks would still be present under one punctuation-stop.

4.2 Stop List

Because of the output from Jieba library, we need to create a stop list to extract possible meaningless words before computing similarities. Our selected list are shown below:

[‘的’, ‘；’, ‘地’, ‘…’, ‘...’, ‘或’, ‘;’ , ‘为’, ‘上’, ‘(’, ‘)’, ‘使’, ‘在’, ‘有’, ‘把’, ‘不’, ‘adv’, ‘等’, ‘adj’, ‘.’ , ‘（’, ‘）’, ‘vi’, ‘vt’, ‘phr’, ‘某人’, ‘’]

This list could be refined for general translations since our list is targeted to Oxford Dictionary.

Note: The reason of including punctuation is to ensure that no punctuation is missed from the previous step since Chinese punctuation take another form of expression.

4.3 Chunk Similarities

4.3.1 Cilin Structure

《同义词词林》(Tongyici Cilin) is a synonym dictionary publish in 1983 by Mei (1983). This work is highly respected in China on developing algorithms regarding similarity computations. Cilin has a unique structure containing words grouped together by a reference code:

> Ac03C05= 结巴，结子，咬舌儿，大舌头
>
> Ac03C06# 斜眼，少白头
>
> Ac03D01@ 俊男靓女
>
> Ac03E01@ 卡通人
>
> Ac03E02# 唐老鸭，灰姑娘，白雪公主，狮子王

Above shows five sample lines of Cilin entries. In front of every bag of words there exists a series of alphabets and numbers indicating where they are at. The reference code can be divided into five levels：

Level 1: index 1, weight = 0.65
Level 2: index 2-3, weight = 0.8
Level 3: index 4, weight = 0.9
Level 4: index 5, weight = 0.96
Level 5: index 6-7

Level 5 does not have a weight since if all 7 prior numbers correspond for two words, they belong to the same cluster but are not necessarily similar. Looking at the 8th index we see three possibilities: =, @, #. “=" indicates equality, meaning words in this group have similarity of 1; “@” implies there only exists one word in this group with no synonyms; “#" shows parallel existences of different words, such as Donald Duck and Cinderella. The last category is hard to determine similarities, therefore we followed a previous research and assign a score of 0.5 (Jiule Tian, 2010). Knowing the structure of Cilin helps us to calculate the “distance" among various word groups and compare similarities of selected word chunks accordingly.

4.3.2 Cosine Similarities

we did not choose to use cosine similarities to find similar chunks in different words. Cosine Similarities are good for character-wise computation, and it is proved in previous research that character-wise translations are likely to have higher matching rates (Maoxi Li, 2011). The same ideology can be applied to our research. For example, “银行” (bank) and “行走" (walk) have no overlapping meanings whatsoever, but character-wise, they do have a common sub-string. This will cause discrepancies due to the lack of information that Chinese characters could provide compare to words.

If we were to compare word similarities using cosine similarities algorithm, we will need to utilize TF-IDF function for calculation using a gigantic data set to train on. Although we would be able to resolve the character overweighting problem, we still have a complicated issue to resolve. Words like “开心” and “高兴” both translate to “happy", but they look really different and cosine similarities would not detect their confusion on surface level (as far as we understand the algorithm). Therefore we chose to make use of Cilin instead of this approach for our experiments.

5 Combining Chunk Similarities and Finding Confusion Terms

Now we propose two ways to calculate the overall translation similarities. The first approach includes a formula to weight chunk similarities and normalize the score, and the second takes the maximum of chunk similarities.

5.1 Weighted function Approach

$$\frac{\sum I(chunks! \in stopwords) * \{I(sim[chunk_i, chunk_k] = 1) * 2 + I(sim[chunk_i, chunk_k] \neq 1) * sim[chunk_i, chunk_k]\}}{num\_comparisons}$$

Note: "I" represents the Indicator function.

The above formula is implemented for a weighted similarity score. It may seem complicated but really straightforward to understand. The nominator is a weighted sum of chunk similarities. Essentially, the first indicator function checks if the chunks were inside the stop word list, if so we do not even need to compare their score (multiply by zero). Then we check if the two chunks match with a really high similarity score and if so, we multiply the weight by 2. The reason behind the weight comes from several previous conducted experiments, and by artificial checking we picked the most optimal number. We also want to compare the result using our formula with established tools in evaluation process. At last we divide the sum by the number of comparison excluding stop word pairs.

5.2 Maximum Approach

$$max(sim(chunk_i, chunk_k))$$

This approach is simpler and more instinctive compare to the prior one. We determine the final score of the two words as the maximum of their chunk similarity scores. Since our final goal is confusion term extraction, it is reasonable to determine two words confusing if one or more chunks of their translations match. This method should be efficient when comparing words like “happy" and “cheerful" in terms of their Chinese translations shown in Section 2.

6 Evaluation

6.1 Data Set

As mentioned before, our testing data set is the 2000 common English words and their Chinese definition from Oxford Dictionary.

6.2 Base Line

We first take English similarity comparison as the Base Line of our system. In order to extract the confusion terms of an English word, it is natural to assume that the synonyms of the word is confusing in terms of its Chinese translation. Therefore we take use of an established library en_core_web_lg to find the synonyms of English words. For any input word, we use en_core_web_lg to compute the similarity between that and all other words from our test set. We consider any words with a similarity larger than 0.5 to be the synonyms of the input word.

However, due to the complexity of Chinese language and the variation of wordings, the English Synonyms do not directly result in translated confusion terms. For instance, for the word "propensity," the synonym output by en_core_web_lg is shown in Table 1. We can see that even the most similar word "predilection" has a different Chinese definition than that of "propensity," which cannot be considered as a confusion term in Chinese. As for the other synonyms, only "predisposition" is somewhat similar to "propensity" in terms of Chinese translation, both expressing " 倾

Synonyms	**Similarity**	**Chinese Translation**
propensity	1.0	（行为方面的）倾向；习性
predilection	0.693	喜爱; 偏爱; 钟爱
predisposition	0.679	倾向；癖性；（易患某种病的）体质
aversion	0.581	厌恶；憎恶
prone	0.575	易于遭受；有做（坏事）的倾向
innate	0.573	（品质、感情等）天生的；先天的；与生俱来的
invariably	0.518	始终如一地；一贯地
disregard	0.518	不理会；不顾；漠视
compulsion	0.518	强迫；强制
averse	0.511	不喜欢；不想做；反对做

Table 1: Synonyms for the word "propensity"

Confusion Terms	Translation
propensity	（行为方面的）倾向；习性
complexion	（事物的）性质，特性
endorsement	支持
mannerism	习性，言谈举止
disposition	倾向，意见

Table 2: Weighted Function's output for "propensity"

Confusion Terms	Translation
propensity	（行为方面的）倾向；习性
nudge	（朝某方向）轻推，渐渐推动
precept	（思想、行为的）准则，规范
turf	（自己的）地盘，势力范围
appreciable	可觉察的；相当大的；明显的
...	

Table 3: Maximum Function's output for "propensity"

向". Moreover, the word "prone" is relatively low in its similarity to "propensity," however, it also has the word " 倾向" in its Chinese translation, yielding a similar Chinese meaning to "propensity." The English word similarity does not directly relate to similarity in Chinese translation.

6.3 Evaluation based on Human Checking

The difficulty of evaluating confusion terms is similar to that of evaluating Machine Translation output mentioned in section 3. Any current system or metrics can only focus the evaluation on the number of matches, regardless of the meaning.

Therefore, we run our algorithm on 20 words from the data set, and calculate the precision of our algorithms' outputs by manually checking if they are confusion terms to the input words in terms of Chinese translation.

6.3.1 Precision of Weighted Function Algorithm

Again, we take the word "propensity" as an example to show the performance of our algorithm. The output of our algorithm is shown in Table 2. We can see that besides "endorsement," all terms in the output have a confusion Chinese Translation similar to that of "propensity," both in terms of character matches and semantics. The precision of the Weighted Function Algorithm on this word is 0.8.

6.3.2 Precision of Maximum Approach

We use a similar method to measure the precision of the Maximum Algorithm on "propensity". The output of the Maximum Algorithm is shown in Table 3. The algorithm outputs 112 terms, most of which are not at all similar to propensity at all. The Maximum Approach takes the maximum chunk similarity between two words as their final similarity. There might be chunks in the translations that are very similar, but don't determine the actual meaning of the phrase. And these words would be counted as confusion terms even though they are not similar in meanings. For instance, the word "nudge" classified as a synonym to "propensity," because it has the Chinese word " 方向" in its translation, which is very similar to " 方面" in the translation of "propensity." However, both of these two chunks are in parentheses, meaning that they serve only to qualify the explanations outside the parentheses to a certain aspect, and they do not determine the actual meaning of the translations. Therefore the Maximum Approach would yield a excessively high false positive rate and low precision of 0.071.

Word	Weighted	Max	Encorewb
colossal	0.874	0.062	0.378
monsoon	0.783	0.092	0.271
irretrievable	0.902	0.104	0.140
negligence	0.856	0.118	0.447
rake	0.824	0.083	0.302
tumult	0.851	0.086	0.358
spur	0.924	0.137	0.284
tweak	0.752	0.079	0.296
aspirational	0.874	0.091	0.285
priest	0.750	0.239	0.656
sift	0.928	0.108	0.163
upstanding	0.934	0.049	0.230
intervene	0.906	0.061	0.135
incriminate	0.800	0.158	0.280
feline	0.667	0.181	0.372
propensity	0.80	0.071	0.300
aesthetic	0.857	0.068	0.125
accrue	0.963	0.121	0.222
arrogate	0.892	0.203	0.272
compliance	0.700	0.190	0.313
AVERAGE	0.806	0.115	0.291

Table 4: Precision for the 3 algorithms for 20 words

6.4 Evaluation Result

The precision of Weighted Approach, Maximum Approach, and English Similarity using En_core_web of the 20 test are shown in Table 4 words that we run. We can see that the Weighted Algorithm has an Average Precision = 0.806, and the Max Algorithm 0.115, and the English Synonyms 0.291. The Weighted Algorithm yields a precision that excessively surpass the other two.

7 Conclusion and future work

The has not been any libraries of existing methods to compare the similarity between translated words. In this paper, we provide the base line systems and two algorithms in solving this problem, which are the English Synonyms, the Weighted Approach and the Maximum Approach. Our evaluation process shows that the Weighted Approach best incorporates the meaning and variations in Chinese expressions, and gives the best precision measure.

We would like to expand our research possibly in the future start by finding a more reliable stop word list. We are also in need for a more concrete, objective, and time-saving evaluation corpus for us to utilize since a sample of 20 words does not necessarily reflect the finalized precision of the three methods.

References

Wei Zhao Jiule Tian. 2010. 基于同义词词林的词语相似度计算方法 (word resemblance computation based on tongyi cilin dictionary). *Journal of Jilin University*, 28(6).

Todd Ward Kishore Papineni, Salim Roukos and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318.

Hwee Tou Ng. Maoxi Li, Chengqing Zong. 2011. Automatic evaluation of chinese translation output: Word-level or character-level? In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:shortpapers*, pages 159–164.

Rahzeb Choudhury Maxim Khalilov. 2012. Building english-chinese and chinese-english mt engines for the computer software domain. In *Proceedings of the 16th EAMT Conference, 28-30 May 2012, Trento, Italy*, pages 33–40.

Jiaju Mei. 1983. *Tongyici Cilin*, volume 1. Harbin Institute of Technology, Harbin, China.

Bao-liang Lu Rui Wang, Zhao Hai. 2015. English to chinese translation:how chinese character matters? In *29th Pacific Asia Conference on Language, Information and Computation*, pages 270–280.