



(AHP)	”	”	(SP)	(GC)	(EF)	(TA)
Google	DeepL	Google	0.7574	(0.8695)	(1.0000)	
		0.6947				
Google			0.8035	0.7706	(0.5643)	(0.4436)

(1)

(2)

[itemindent=2em]

- 1
- 2
- 3
- 4
- 5
- 6

1. (AHP)

(Google Translate) DeepL

2.

jieba

**3.**

1.

$$\begin{pmatrix} \text{SP} \\ \text{GC} \end{pmatrix}$$

DejaVu Sans Mono

```
1 class TranslationEvaluator:
2     """
3
4     def __init__(self):
5         """
6         # (AHP)
7         # (SP) (GC) (EF) (TA)
8         self.weights = {
9             "SP": 0.4, #
10            "GC": 0.25, #
11            "EF": 0.2, #
12            "TA": 0.15, #
13        }
14
15        #
16        self.stopwords = set(["", " ", "a", "an", "and", "are", "as", "at", "be", "but", "by", "can", "could", "do", "does", "for", "from", "has", "have", "he", "her", "his", "in", "is", "it", "of", "on", "or", "over", "she", "that", "the", "there", "they", "to", "was", "were", "with", "would", "you", "your"])
17
```

DejaVu Sans Mono

```
1 def calculate_semantic_preservation(self, original_text, translated_text):
2     """
3
4
5     :
6     original_text (str):
7     translated_text (str):
8
9     :
10    float: [0,1]
11    """
12    #
13    if not original_text.strip() or not translated_text.strip():
14        return 0.5
15
16    try:
17        #
18        cosine_sim = self.calculate_cosine_similarity(original_text, translated_text)
19        edit_sim = self.calculate_edit_distance(original_text, translated_text)
20
21        #
22        semantic_score = 0.6 * cosine_sim + 0.4 * edit_sim
23
24        #
25        if semantic_score < 0.2:
26            semantic_score = 0.2
27
28        return semantic_score
29    except Exception as e:
30        print(f" : {e}")
31        return 0.5 #
```

DejaVu Sans Mono

```
1 def calculate_grammar_correctness(self, text):
2     """
3
4
5     :
6     text (str):
7
```