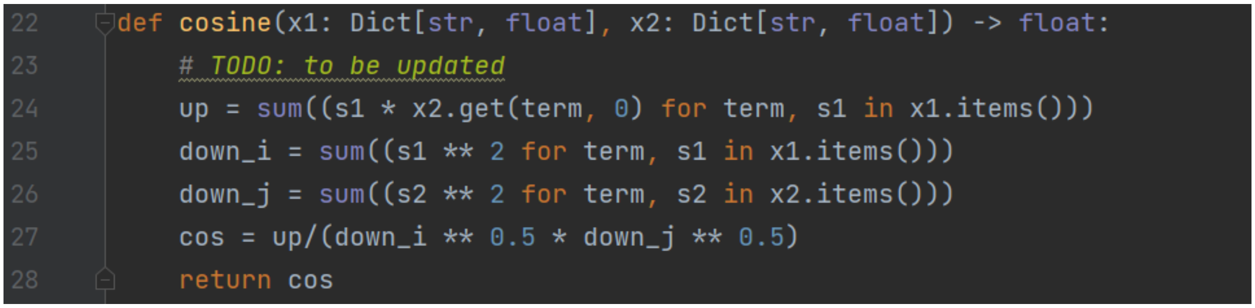
Quiz 2

Yulin Hu

Task 1:



```python
def cosine(x1: Dict[str, float], x2: Dict[str, float]) -> float:
    # TODO: to be updated
    up = sum((s1 * x2.get(term, 0) for term, s1 in x1.items()))
    down_i = sum((s1 ** 2 for term, s1 in x1.items()))
    down_j = sum((s2 ** 2 for term, s2 in x2.items()))
    cos = up/(down_i ** 0.5 * down_j ** 0.5)
    return cos
```

For Task 1, I created 3 for-loops (up, down_i, down_j) each denoting:

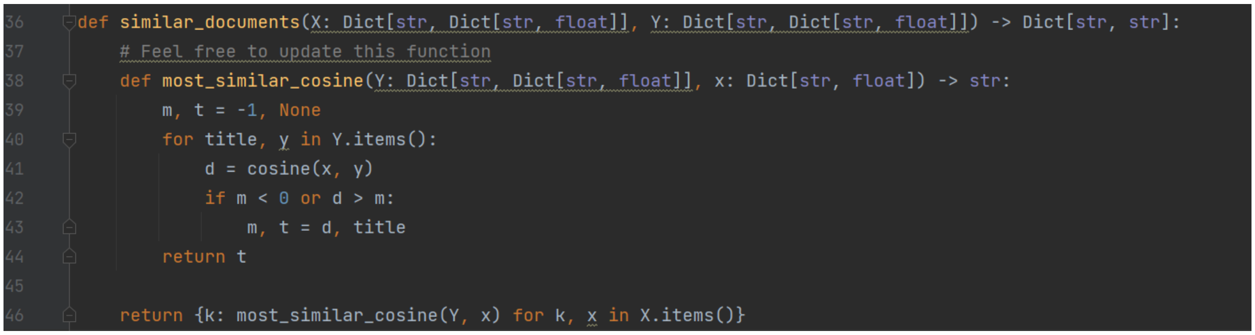
up: $\sum_{\forall k}(x_{ik} \cdot x_{jk})$

down_i: $\sqrt{\sum_{\forall k}(x_{ik})^2}$

down_j: $\sqrt{\sum_{\forall k}(x_{jk})^2}$

The function cos = up/(down_i ** 0.5 * down_j ** 0.5) then returns the result of cosine similarity.

Task 2:



```python
def similar_documents(X: Dict[str, Dict[str, float]], Y: Dict[str, Dict[str, float]]) -> Dict[str, str]:
    # Feel free to update this function
    def most_similar_cosine(Y: Dict[str, Dict[str, float]], x: Dict[str, float]) -> str:
        m, t = -1, None
        for title, y in Y.items():
            d = cosine(x, y)
            if m < 0 or d > m:
                m, t = d, title
        return t

    return {k: most_similar_cosine(Y, x) for k, x in X.items()}
```

For Task 2, I only modified the similar_documents function to only use cosine similarity instead of Euclidean distance. In fact, I implemented the most_similar_cosine function within the similar_documents function for convenience. The reason why I chose to use cosine similarity is because that cosine similarity measures the vectors with respect to the origin.

In VSM, the two vector spaces are both in high dimensions. The property of cosine similarity makes it only subject to the common terms between two vectors, whereas Euclidean distance performs poorly on high dimensional data.

In the vectors we are comparing, they both have high dimensions and mainly are the same stories reconstructed in different story-telling orders. Therefore, it is the common terms between vectors that matter more than other factors, in which case cosine similarity is known to have far better performance than Euclidean distance.