1. **DEFINITION OF CLUSTER ANALYSIS:**

- Cluster Analysis (Clustering) is a method of arranging items (e.g: features, posts…) so that similar items are in one cluster and dissimilar items are in different ones. To achieve this, we must transform the text (in this case, the requirements of the feature) into meaningful values, so we can feed them into our machine-learning algorithms such as clustering.

1. **CALCULATE SIMILARITY BETWEEN FEATURES USING CLUSTER ANALYSIS:**

- One of the method to calculate the text similarity measure is to use Levenshtein distance (edit distance). The Levenshtein distance between two words is the minimum number of single character edits (e.g: insertions, deletions or substitutions) required to change one word into the other.

For example: We have two words: ‘screen’ and ‘seen’. The minimum number of single character edits (edit distance) would be 2, we need to delete ‘c’ and ‘r’ to turn ‘screen’ into ‘seen’. This algorithm is, howerver, quite costly, as it is bound by the product of lengths of the first and second words.

- Another method is using bag-of-words model approach. It calculates the text similarity measure using simple words counts. For each word in the requirements of the feature, its occurrence is counted and noted in a vector.

For example:

+ Feature A contains requirement 1: When DVR Service is not authorized, click on view playlist will display OSD 117

+ Feature B contains requirement 2: User can access the recordings by clicking on playlist

The occurrences of words are below:

|  |  |  |
| --- | --- | --- |
| **Word** | **Occurrences in Feature A** | **Occurrences in Feature B** |
| when | 1 | 0 |
| playlist | 1 | 1 |
| recordings | 0 | 1 |
| authorized | 1 | 0 |
| … | … | … |

We can use two columns Feature A and Feature B to make a simple vectors. Then, we can calculate the Euclidean distance between the vectors of all features. Using this way, we can view the impact when adding a new feature comparing to existing features by the distance between them.

1. **IMPLEMENT CLUSTER ANALYSIS USING BAG-OF-WORDS METHOD:**

Output of the program when comparing with existing feature E

Feature A with similarity measure = 5

Feature B with similarity measure = 3

Feature C with similarity measure = 1

Feature D with similarity measure = 2

From above output, we can conclude that Feature C is impacted by feature E the most.

1. **FURTHER OPTIMIZE BAG-OF-WORDS METHOD :**

* Removing less important words in requirements: some of the words in the requirements appear very often in all sorts of different contexts (e.g: a, the, about, along…). To improve the output from cluster analysis, we shouldn’t use these words when calculating the text similarity measure.

In Python, we can use parameter stop\_words = ‘english’ to filter out these redundant words. By adding this parameter, we will use a set of 318 common English stop words.

For example:

*from sklearn.feature\_extraction.txt import CountVectorizer*

*vectorizer = CountVectorizer(min\_df=1, stop\_words=’english’)*

*print(vectorizer.get\_stop\_words())*

*>>> frozenset(['all', 'six', 'less', 'being', 'indeed', 'over', 'move', 'anyway', 'fifty', 'four', 'not', 'own'….])*

Or we can define our own stop\_words

*stop\_words = frozenset(["word1", "word2","word3"])*

* Stemming: When using bag-of-words model approach with simple words counts method, we usually encounter many similar words which are different variants but have the same root. For example: ‘clicking’, ‘clicks’, ‘click’, they are pretty much the same. We should count these words as one instead of three different words. To reduce words to their specific stem by removing ed, ing, ly… (stripping ‘clicking’, ‘clicks’ into one word, ‘click’), we can use package NLTK in Python. In below example, we select SnowballStemmer for stemming ‘clicking’ and ‘clicks’

For example:

*import nltk.stem*

*s = nltk.stem.SnowballStemmer('english')*

*print(s.stem("clicking"))*

*print(s.stem("clicks"))*

*>>> click*

*>>> click*

Based on above methods, we can split requirements processing phase into below steps:

+ Tokenizing the requirements

+ Removing stop words

+ Reducing words to their word stem

+ Counting the remaining words

1. **DRAWBACK OF BAG-OF-WORDS METHOD:**

* It does not cover word relations. e.g: The text ‘change channel’ and ‘channel change’ will both have the same vector.
* It does not capture negations correctly. e.g: The text "DVR is authorized" and "DVR is not authorized" will look very similar in vectors, although they contain quite the opposite meaning. This problem can be solved by two or three words at the same time instead of counting only one word.
* It can not calculate misspelled words correctly. e.g: ‘tune to channel’ and ‘tune to channej’, ‘tune to channej’ is clearly a misspelled words and we should count them as one. But bag-of-words will treat them as totally different words.