SUMMARY  Nowadays there is no way to automatically obtain the function points when using function point analysis (FPA) method, especially for the requirement documents written in Chinese language. Considering the characteristics of Chinese grammar in words segmentation, it is necessary to divide words accurately Chinese words, so that the subsequent entity recognition and disambiguation can be carried out in a smaller range, which lays a solid foundation for the efficient automatic extraction of the function points. Therefore, this paper proposed a method of K-Means clustering based on TF-IDF, and conducts experiments with 24 software requirement documents written in Chinese language. The results show that the best clustering effect is achieved when the extracted information is retained by 55% to 75% and the number of clusters takes the middle value of the total number of clusters. Not only for Chinese, this method and conclusion of this paper, but provides an important reference for automatic extraction of function points from software requirements documents written in other Oriental languages, and also fills the gaps of data preprocessing in the early stage of automatic calculation function points.

key words: Chinese, TF-IDF, K-means, clustering

1. Introduction

The software scale assessment is the first step of the software cost estimation in the early stage of the software development period. The method of Function Point Analysis (FPA), represented by IFPUG method, has become one of the most important evaluation methods in the area of software scale evaluation [1], [2].

The function point is a ‘unit of measurement’ to express the amount of business functionality an information system, often as a product, provides to a user. Function points, which are used to compute a functional size measurement (FSM) of software, were defined in 1979 in Measuring Application Development Productivity by Allan Albrecht at IBM [3]. The functional user requirements of the software are identified and each one is categorized into one of five types: outputs, inquiries, inputs, internal files, and external interfaces. Once the function is identified and categorized into a type, it is then assessed for complexity and assigned a number of function points.

As for IFPUG, by 1984, the measurement of function points had become widely used, leading to the formation of the International Function Point Users Group (IFPUG), a non-profit consortium. Today, the consortium has become the world’s largest software measurement consortium, and currently, many national software organizations have joined IFPUG. Actually, IFPUG function point analysis allows the application system to be decomposed by components, and each component is calculated with the function points defined by IFPUG as the unit of measurement, so as to get the function points that reflect the scale of the entire application system.

However, almost all the available FPA methods, including the IFPUG method, are to manually identify the data function and transaction function from the requirements documents obtained, and then calculate the function points by computer [4]. While the disadvantages of these methods are, on the one hand, they require manual mining of keywords and features in the requirements documents, which takes a lot of work and too much time, and usually leads to the increase of software cost, and at the same time, due to the particularity of manual operation and individual difference, it will inevitably bring unnecessary risks to the accuracy of software scale measurement. On the other hand, unlike western languages such as English and Spanish, software requirements documents written in Oriental languages especially written in Chinese, Japanese, etc., cannot be adapted to the western natural language processing methods due to the different syntax. For example, in western languages, the first letter of sentences, proper nouns, personal names, places and so on, are all capitalized, which has a natural advantage in classification or clustering. Although nowadays there are also methods for mining information from documents written in Oriental languages, but they all have one flaw, that is, in the early stage of mining, the transformed language feature vectors are too complicated and the formed feature matrix is too sparse, which would bring huge workload to the subsequent text disambiguation.

In actual estimates of the scale of the software, software requirements documents are often the only available reference material, hence, aiming at software requirements documents written in Chinese, this paper studies how to extract and cluster Chinese vocabulary more effectively in the early stage of software development period. The reason is
that, whether it is to calculate data function points or to calculate transaction function points, it is necessary to identify objective objects, operations and other actual information according to the words described in the software requirements document. For example, for a software requirements document of a student enrollment system, on the one hand, in order to calculate data function points, in data preprocessing, objective objects such as students’ names, genders and ages should be extracted, and then function points should be calculated according to relevant rules; on the other hand, in order to calculate transaction function points, during data preprocessing, operation terms such as input, output and deletion, should be identified and then function points should be calculated according to relevant rules. Therefore, this paper studies the only software requirement documents written in Chinese, and proposes the Method of K-Means Clustering Based on TF-IDF for Software Requirements Documents Written in Chinese Language, which automatically extracts the actual and effective information according to the text content, lays a good foundation for automatic analysis and calculation of function points, at the same time, this method fills the gaps of data preprocessing in the early stage of automatic calculation function points.

TF-IDF, short for term frequency-inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. The importance of a word increases in direct proportion to the number of times it appears in the document, but at the same time, it decreases in inverse proportion to the frequency it appears in the corpus. Actually, the main idea of TF-IDF is that if a certain word or phrase appears in one article with high frequency (TF) and rarely appears in other articles, it is considered that this word or phrase has a good ability to distinguish categories and is suitable for classification.

The first section, this paper introduces the current IFPUG method, which is mainly used to measure the software scale, but it is still not free from manual constraints. In the case of only requirement documents written in Chinese, automatic acquisition of function points is still a gap at present. Therefore, this paper proposes a method to fill the gaps of data preprocessing in the early stage of automatic calculation function points.

The second section, this paper discusses the progress of IFPUG in computing function points in recent years, and points out the disadvantages of the IFPUG method in processing requirements documents written in Chinese compared with those written in western languages.

The third section, this paper proposes a method of K-Means clustering based on TF-IDF, and explains the theoretical idea of this method in detail.

The fourth section, 24 software requirement documents written in Chinese in different fields are taken as samples. According to the theoretical ideas described in Sect. 3, experiments are carried out, and good results are obtained.

The fifth section, we analyze the threats from 4 aspects to the validity of the new method proposed in this paper.

The sixth section, this paper summarizes the conclusion that the proposed method, fills the gaps of software scale automatically measurement, which has the important practical significance.

2. Related Work and Research in Quo

With IFPUG method as the main line, NESMA, COSMIC, and Mark II methods and so on, all of them have been evolved into over the past 40 years or so. Nevertheless, the basic principles of these methods are the same.

2.1 IFPUG Function Point Method

Take the IFPUG method as an example, as the Fig. 1 shows. The IFPUG method breaks down the software or application system into components, and calculates each type of component in terms of the units measured by the function points defined by IFPUG, so as to obtain the total number of the function points reflecting the scale of the entire software or system.

Generally, the requirements are divided into two categories, one is data functional requirements and the other is transaction functional requirements. In the Fig. 1, Data functional requirements are further divided into Internal Logical Files (ILF) and External Interface Files (EIF). Transaction functional requirements are generally divided into three categories: External Inputs (EI), External Outputs (EO) and External inquiries (EQ). Finally, the total number of function points can be obtained by calculating the values of ILF, EIF, EI, EO and EQ respectively and summing them with weights.

However, both the IFPUG method and other function point methods are completely dependent on manual processing in the initial identification of valid entity information.

2.2 Research Status of K-Means Method and TF-IDF Method

In 2012, Yao et al. focused on the process of Chinese text clustering based on K-means, and found that isolated text tends to affect the center of clustering. The experiments showed that improved method averagely increased purity and F value about 10 percent over the original method [5].

In 2016, Zhang et al. proposed a new similarity measure method, took statistics information and part of speech
of feature terms into account. The experiments showed that the improved method improved cluster quality in terms of F-measure, and had a less time consumption [6].

Both literatures [5] and [6] mentioned that Chinese documents need to be segmented during data preprocessing, and make full use of k-means clustering algorithm according to specific situations. This gives us the inspiration that when we get the text to do the initial information mining when we get the text to do the initial information mining, we can also artificially segment the required text first, and then conduct data mining in a smaller range. In this way, according to the problems we face, we can make more effective use of k-means method.

Also in the same year of 2016, Y.Luo et al. applied TF-IDF method to manipulate text corpus of the commodity for getting the weight matrix of the commodity words. Then TF-IDF was combined with the Word2Vec model, in order to help to acquire a new one-dimensional vector. After experiment, the new method of identification has been improved obviously [7].

In literature [7], in order to verify the effectiveness of the proposed algorithm, 14 Chinese classics and 19 English classics were selected respectively for sample training, which covered a wide range, making the experiment representative to a certain extent. Therefore, when we choose software requirements documents, we try our best to make our samples cover a relatively broad range, and in terms of length, we also choose requirements documents of different lengths, and strive to make our experimental results has a certain representativeness.

At the same time, Y.Wang et al. improved TF-IDF method based on co-occurrence network and established feature words extraction and words sequential relations for classified incidents. They designed feature words network model for multi-documents unsafe incidents classification. After experiment, classification accuracy of improved method was verified by the experiments [8].

Literature [10] points out that Chinese word segmentation is one of the most basic tasks in data mining of Chinese text, and also a preliminary pre-processing step in downstream Chinese natural language processing. Therefore, if the automatic acquisition of function points is to be carried out in the end, the effect of the initial data pretreatment is the basis of all subsequent data mining, that is, the effective segmentation in the data pretreatment part is very important. In this paper, the data preprocessing part of software requirements document is studied.

2.3 Example Illustration

The above research results are also simple researches and expansions of K-means or TF-IDF in their respective related research fields. There into, some literatures are for documents written in Chinese. But, in fact, considering the particularity of Chinese grammar, that is, each character except the first one and last one in a Chinese sentence, can usually make up different words with the character in front of itself or behind itself. Unfortunately, both of these two situations can form sentences that conform to Chinese grammar, leading to errors in vocabulary extraction by the various methods, which will bring negative impact on the subsequent word disambiguation, especially in the field of software scale measurement. This is also because the premise of automatic calculation of data function points and transaction function points must be that the words in the text written in Chinese, can be accurately identified, clustered, and extracted, and among which, the most basic are recognition and clustering [11].

For example, we divide and extract the vocabulary of the sentence ‘小哥自已的香烟不卖给外国人’; but it turns out there are four forms of division, each of which is not only grammatically correct, but each of which has a clear, common sense meaning, as the Table 1 shows. Hence, the classification and clustering of Chinese words does not have the innate advantages of English sentences, including the initial letters of proper nouns such as people names and place names are all capitalized, and the words are clearly separated by Spaces.

From the Table 1, we can see that the same Chinese
sentence can be divided into four different divisions on the premise that it conforms to grammar and has practical meaning. However, due to the advantages of English grammar itself, its lexical segmentation is clear, so it has little negative impact on subsequent clustering.

On the contrary, in the field of software scale measurement, it is precisely because of the particularity of Chinese grammar that automatic calculation of data function points and transaction function points is not only prone to errors, but also brings problems including, but not limited to, in the early stage of mining, the transformed language feature vectors are easy to become complex, and the formed feature matrix is easy to be sparse. In general, there is very little research on automating function points based on requirements documents written in Chinese in the field of software scale measurement.

The Method of K-Means clustering based on TF-IDF for software requirements documents written in Chinese language proposed in this paper, is not a single study of K-means or TF-IDF method, but a combination of the two methods to carry out the initial information mining for the requirement document written in Chinese in the software scale measurement field. When calculating data function points and transaction function points, according to requirements documents, this method can free manpower from bondage and help to automatically calculate function points through information mining. At the same time, this method fills in the gaps of data processing in the field of software scale measurement in the early stage of research on automatic acquisition of function points based on requirements documents written in Chinese language.

3. Theoretical Idea of the Method

In the beginning, the only material we have is a requirements document written in Chinese. According to the methods of calculating function points adopted in the past, the data function points and transaction function points are obtained after analyzing the requirement documents by experts in the way of manual interpretation. The disadvantage of this method is that manual interpretation is highly subjective. Different experts often have different understandings of the same text due to different perspectives and experiences, so the final number of functional points is often not the same, and it takes longer time. The long time it takes, plus the cost of hiring or consulting experts, adds to the cost at the beginning of software development. Of course, automating software scale can’t happen overnight. This paper is to further optimize the automatic calculation of data function points and transaction function points of the early data preprocessing in the early stages of software development, when only requirements documents are available.

The method proposed in this paper is mainly divided into two modules, as the Fig. 2 shows.

3.1 Recognition Module

The first module is the recognition module. There are 3
steps, text input, text segmentation, and acquisition of feature matrix. The purpose of this module is to generate acquisition of feature matrix as the input of the second module.

The first step, text input. However, documents written in any language will inevitably appear a lot of meaningless function words, including prepositions, conjunctions, auxiliary words, interjections and so on. These function words must be removed, otherwise it will cause a waste of computer resources and result in a meaningless increase in computation.

The second step, text segmentation. In this step, the input unstructured Chinese text is artificially segmented in order to avoid the whole text as a single one-dimensional sample. This is also for the purpose of forming a Feature Matrix with appropriate dimensions in the next step. And then, by using the existing tools and methods, the multi-dimensional Chinese text that has been segmented is decomposed into multiple words conforming to Chinese grammar according to each sentence in each dimension.

In practice, it should be reasonably segmented according to the length of the whole article. Unlike Western languages such as English, Eastern languages such as Chinese can express more meaning with fewer characters. Anyway, if the number of segmentation is too large, then the number of characters in each part is too small, and the contingency of vocabulary extraction will be greatly increased, the error rate will also rise, and the effect of the experiment will be weakened. However, if the number of clustering is too small, it will lose the significance of segmentation. Because the matrix dimensions formed by the whole article extraction will be too small, so that the loss of information will be very serious.

The third step, acquisition of feature matrix. By using the method of TF-IDF, the text written in Chinese is transformed into a matrix based on the divided sentence fragments in the second step. Moreover, the transformation of the feature matrix should be carried out at the same time, because the process of transformation is to remove empty function words and characterize the remaining content words, so that we can obtain the feature matrix.

3.2 Clustering Module

The second module is the clustering module. There are 3 steps, dimensionality reduction, clustering, and effect evaluation.

The first step, Dimensionality reduction. After the acquisition of the feature matrix, it will be found that although only meaningful content words are left, the number of features formed is still large and very sparse because Chinese vocabulary can express the same entity with different words. Therefore, it is necessary to reduce the dimension of the feature matrix, in order to reduce the errors caused by redundant information, and the data can be more easily used, the accuracy of identifying can be improved.

The second step, Clustering. After dimensionality reduction, both of the complexity of matrix and the computation are reduced while the main information is retained. Since the source of the sample is the software requirements documents written in Chinese language, we need to determine the correlation of the Chinese words extracted from the documents to conduct the initial screening, and there is no prior data for reference, nor training for such identifying. Therefore, we used the clustering method, words with similar characteristics or attributes should be gathered together, dissimilar samples are grouped into different classes.

The third step, is Effect Evaluation. After all, we need to evaluate the effect of clustering. Only when the clustering effect is satisfactory or meets the requirements, can the words or information that has been clustered be used.

Through this experiment, we can verify that in the initial stage of software scale measurement, when only the requirement documents written in Chinese are available, this method can effectively preprocess the data, and lay a better foundation for subsequent automatic calculation or extraction of function points.

4. Experiments and Result Analysis

In order to verify the above method, we carried out experiments with 24 software requirement documents written in Chinese, which are all from our own software development department. The relevant information statistics for these software requirements documents are shown in the Table 2.

4.1 Preparation of Experimental Data

In order to verify the above method, we carried out experiments with 24 software requirement documents written in Chinese, which are all from our own software development department. The relevant information statistics for these software requirements documents are shown in the Table 2.

As can be seen from the Table 2, the requirement document with the research field of data administration in 2005 is the smallest, about 1900 Chinese characters, while the largest one is the requirements document on situation evaluation from 2019. The average length of the 24 documents is 5979 Chinese characters, and the scope of research is also varied. Therefore, the selection of these 24 software requirements documents, it is a certain persuasive.

4.2 Experiments

Firstly, we divide each requirements documents written in Chinese into several paragraphs. For this experiment, considering that the smallest requirement document has only 1900 Chinese characters, we artificially divide the smallest document into 8 paragraphs, each of which has an average of about 237.5 Chinese characters. In this way, the average number of Chinese characters in each paragraph can be guaranteed to exceed 230, enough to express a clear and complete meaning of a paragraph. So far, the text has been divided into 8 segments and input as 8 strings.
Table 2  Statistical table of software requirements documents written in Chinese

| Document | Year | Length (Hundred Characters) | Domain                  | Document | Year | Length (Hundred Characters) | Domain                  |
|----------|------|-----------------------------|-------------------------|----------|------|-----------------------------|-------------------------|
| 1        | 2005 | 19                          | Data Administration     | 13       | 2009 | 60                          | Vehicle Management       |
| 2        | 2008 | 28                          | Cargo logistics         | 14       | 2011 | 63                          | Personnel Management     |
| 3        | 2008 | 31                          | Personnel Management    | 15       | 2008 | 55                          | Systems Control          |
| 4        | 2006 | 34                          | Cargo logistics         | 16       | 2013 | 73                          | Data Administration      |
| 5        | 2016 | 53                          | Device Statistics       | 17       | 2015 | 75                          | Device Statistics        |
| 6        | 2018 | 77                          | Personnel Management    | 18       | 2018 | 87                          | Personnel Management     |
| 7        | 2019 | 102                         | Situation Evaluation    | 19       | 2017 | 83                          | Data Administration      |
| 8        | 2017 | 78                          | Systems Control         | 20       | 2008 | 30                          | Cargo logistics          |
| 9        | 2017 | 69                          | Vehicle Management      | 21       | 2009 | 27                          | Vehicle Management       |
| 10       | 2012 | 43                          | Data Administration     | 22       | 2017 | 82                          | Personnel Management     |
| 11       | 2015 | 54                          | Device Statistics       | 23       | 2019 | 97                          | Market Survey            |
| 12       | 2006 | 20                          | Data Administration     | 24       | 2018 | 95                          | Situation Evaluation     |

Also, for comparison purposes, all the other 23 documents that are longer than this one are also divided into eight paragraphs. Then, after removing function words, TF-IDF method is adopted to evaluate the importance of the above 8 paragraphs and to form a feature matrix.

Secondly, since the feature matrix we obtained is a sparse matrix, dimensionality reduction of this matrix is also required to simplify calculation. The dimensionality reduction method of Principal components analysis (PCA) was selected. The reason is that PCA is a technique for analyzing and simplifying data sets, and when reducing the dimensions, the features, which contribute the most to the variance value in the data sets, are retained, that is to say, the original information of the data can be retained to the maximum extent. Moreover, the running speed of PCA method is also fast.

Thirdly, On the basis of obtaining the feature matrix after dimensionality reduction above, we choose k-means clustering method, which has the advantages of simple structure and fast running speed, which is more suitable for Chinese vocabulary clustering. And, we use the Silhouette Coefficient (SC) to evaluate the clustering effect. SC combines the two factors of cohesion degree and coupling degree, to evaluate the impact of different methods or different operation modes of methods on clustering results on the basis of the same original data, in order to evaluate the effect of clustering. The value range of SC is between $[-1, 1]$. In general, positive value is better, and the closer the coefficient is to 1, the better the clustering effect is. The formula is as follows.

$$SC(i) = \frac{b(i) - a(i)}{\max[a(i), b(i)]}$$  \hspace{1cm} (1)

In the formula, $a(i)$ is the average of the degree of dissimilarity of the $i$ vector to all the other points in the same cluster; $b(i)$ is the minimum of the average dissimilarity of the $i$ vector to all the points in the other clusters, $SC(i)$ is the Silhouette Coefficient value.

Fourthly, for this sample, the percentage of retained information needs to be specified. For comparison purposes, we set the percentage successively as 95%, 90%, 85%, 80%, 75%, 70%, 65%, 60%, 55%, 50% and 45%, in order to check the clustering effect under the condition that different percentages of information are retained. The reason is that, on the one hand, at present, any method cannot achieve 100% of the data retention. In fact, theoretically, the data processing will inevitably lead to more or less data loss. On the other hand, If the percentage of retained data information is too low, no matter how good the clustering effect is, it will be meaningless.

In addition, the k-means method needs to specify the number of clusters to be divided in advance. We have divided each text into 8 segments, so the number of clustering to be divided is set as 2, 3, 4, 5, 6 and 7 respectively, and 300 iterations are carried out respectively. We will cross-compare the experimental results in each case.

We repeated the above experiment for another 23 software requirements documents from different domains, and combined all the results for comparison and analysis.

4.3 Results and Its Analysis

4.3.1 Clustering Results and Its Analysis

According to the percentage of information retained and the number of clusters, the results are shown in Table 3, where the ‘Percentage’ represents the Percentage of information retained. Each 8-bit number represents, the classification structure of the clustering of 24 different samples, according to the different percentage of retained information in the case of different number of clusters.

From the vertical perspective of each sample in Table 3, when the number of clusters is 2, no matter how much information is retained, the results of classification are very similar. In fact, we can see from the Table 3 that no matter what the number of clusters is set, the results of clustering change very slowly with the order of the percentage of retained information. That is to say, only when the change of the percentage of retained information shows a large fluctuation, the clustering results under the same number of clusters will show a significant change.

The reason is that, the samples are documents of the
| Percentage | Data Sets | Number of Clusterings | Data Sets | Number of Clusterings |
|-----------|----------|-----------------------|----------|-----------------------|
| 95%      | 1        | 11100000                | 11102222 | 00001111                | 00000000 | 12202100 | 22330100 | 43002121 | 04253131 | 04351262 |
| 90%      | 1        | 11100001                | 00011222 | 20013335                | 03311442 | 31324503 | 5016525 | 11110100 | 00011222 | 00011222 | 00011222 | 00011222 | 00011222 | 00011222 |
| 85%      | 1        | 11110000                | 11112001 | 00011352                | 01313244 | 5110342 | 3216054 | 11110100 | 00011222 | 00011222 | 00011222 | 00011222 | 00011222 | 00011222 |
| 80%      | 1        | 11110000                | 11112001 | 00011352                | 01313244 | 5110342 | 3216054 | 11110100 | 00011222 | 00011222 | 00011222 | 00011222 | 00011222 | 00011222 |
| 75%      | 1        | 11110000                | 11112001 | 00011352                | 01313244 | 5110342 | 3216054 | 11110100 | 00011222 | 00011222 | 00011222 | 00011222 | 00011222 | 00011222 |
| 70%      | 1        | 11110000                | 11112001 | 00011352                | 01313244 | 5110342 | 3216054 | 11110100 | 00011222 | 00011222 | 00011222 | 00011222 | 00011222 | 00011222 |
| 65%      | 1        | 11110000                | 11112001 | 00011352                | 01313244 | 5110342 | 3216054 | 11110100 | 00011222 | 00011222 | 00011222 | 00011222 | 00011222 | 00011222 |
| 60%      | 1        | 11110000                | 11112001 | 00011352                | 01313244 | 5110342 | 3216054 | 11110100 | 00011222 | 00011222 | 00011222 | 00011222 | 00011222 | 00011222 |
| 55%      | 1        | 11110000                | 11112001 | 00011352                | 01313244 | 5110342 | 3216054 | 11110100 | 00011222 | 00011222 | 00011222 | 00011222 | 00011222 | 00011222 |
| 50%      | 1        | 11110000                | 11112001 | 00011352                | 01313244 | 5110342 | 3216054 | 11110100 | 00011222 | 00011222 | 00011222 | 00011222 | 00011222 | 00011222 |

Table 3: Clustering results
| Percentage | Results | Data Sets | Number of Clusterings |
|-----------|---------|-----------|----------------------|
| 95%       | 00110000 | 12200110 | 13330201 | 41132200 | 41135240 | 02214365 |
| 90%       | 10001111 | 00222110 | 01333022 | 01132200 | 01145022 | 02214035 |
| 85%       | 10000111 | 12231110 | 10332210 | 11032034 | 10032315 | 05321605 |
| 80%       | 10000111 | 12231111 | 10332210 | 11032034 | 10032315 | 05321605 |
| 75%       | 10000111 | 11132022 | 11133022 | 21103224 | 41159352 | 26153024 |
| 70%       | 10000111 | 12200110 | 11330201 | 10032121 | 10032315 | 05321605 |
| 65%       | 10000111 | 12200110 | 11330201 | 10032121 | 10032315 | 05321605 |
| 60%       | 10000111 | 12200110 | 11330201 | 10032121 | 10032315 | 05321605 |
| 55%       | 10000111 | 12200110 | 11330201 | 10032121 | 10032315 | 05321605 |
| 50%       | 10000111 | 12200110 | 11330201 | 10032121 | 10032315 | 05321605 |
| 45%       | 10000111 | 12200110 | 11330201 | 10032121 | 10032315 | 05321605 |

| Table 3 (Continued) | |
|----------------------|----------------------|
| Percentage | Results | Data Sets | Number of Clusterings |
| 95%       | 00000001 | 00222211 | 01333222 | 22230401 | 05032421 | 40326383 |
| 90%       | 11100011 | 22011100 | 22033311 | 00422311 | 45320211 | 13460522 |
| 85%       | 11100011 | 22011100 | 22033311 | 00422311 | 45320211 | 13460522 |
| 80%       | 11100011 | 22011100 | 22033311 | 00422311 | 45320211 | 13460522 |
| 75%       | 11100011 | 22011100 | 22033311 | 00422311 | 45320211 | 13460522 |
| 70%       | 11100011 | 22011100 | 22033311 | 00422311 | 45320211 | 13460522 |
| 65%       | 11100011 | 22011100 | 22033311 | 00422311 | 45320211 | 13460522 |
| 60%       | 11100011 | 22011100 | 22033311 | 00422311 | 45320211 | 13460522 |
| 55%       | 11100011 | 22011100 | 22033311 | 00422311 | 45320211 | 13460522 |
| 50%       | 11100011 | 22011100 | 22033311 | 00422311 | 45320211 | 13460522 |
| 45%       | 11100011 | 22011100 | 22033311 | 00422311 | 45320211 | 13460522 |

Table 3 (Continued)
software requirements, with a limited number of characters, so the dimensions of segmentation are not large in the initial text segmentation.

Moreover, the documents of software requirements are written in Chinese language, whose characteristic is that it is a kind of ideographic character, which has certain features of transcending time and space. As a rule, compared with western languages, Chinese and other Oriental languages can express the same meaning with fewer characters. As a result, in the process of data processing, even if the percentage of retained information is low, the clustering results are not significantly different from the clustering results when the percentage of retained information is high. The results of clustering are almost the same especially when the percentage difference of their retained information is not big.

From the horizontal perspective of each data set, in the 8-bit number, regardless of the percentage of information retained, the first bit number and the last one belong to two different clusters with a high probability, and their coupling is very low. While the second bit number and the third one are divided into the same cluster with a much higher probability, indicating that the two features have a high correlation and cohesion. The reason is that the first two bit numbers are usually from the first few paragraphs of the software requirements documents which express very similar contents and meanings. In addition, the texts are written in Chinese language, so these two usually belong to the same cluster.

4.3.2 Effect Evaluation of Clustering Results and Its Analysis

We use the Silhouette Coefficient (SC) to evaluate the clustering effect. That is, the clustering SC values of these 24 different samples, which are under different number of clusters, and according to the different percentage of information retained, as the Table 4 shows.

In order to facilitate intuitive, convenient observation of SC values, based on the different number of clusters and different percentage of retained information, the SC values of overall clustering effect in the form of scatter diagram, is shown in Table 5.

At the same time, in order to facilitate the analysis, we connected the SC values in Table 5 with a plane and projected them on the interface of XOY plane, and in the form of planar diagram, to check the effect, as the Table 5 shows.

In Table 5, the Y-axis represents the ‘The Percentage of Original Information Retained’, the X-axis represents the ‘Number of Clusters’, and the Z-axis represents the ‘SC Values’. From this table, we can see that as far as the 24 samples, all of these SC values present a parabolic pattern, and the average clustering effect is better when the number of clusters is 4 or 5. The number of clustering can reach a peak at 4, and when the number exceeds 5, the SC value reflecting the clustering effect decreases rapidly.

The reason is that we divided each text into 8 parts according to the length of these texts, if the number of clusters is relatively small, many should be separated clusters were gathered together, leading to a lower cohesion, while if the number of clustering is relatively large, the clustering that should be clustered together will be separated, resulting in too high coupling and poor clustering effect.

Therefore, from the 24 samples we selected, it seems that it is the best if the average number of clusters is 4 or 5.

For the convenience of comparison, we checked the maximum, average and minimum SC values of these 24 data samples according to the different percentage of retained data, but based on the different number of clusters. As shown in the Fig. 3.

From these figures in Fig. 3, we can see that no matter what the percentage of retained data is, the trend of maximum, average and minimum values of SC is the same, which first increases and then decreases with the increase of the number of clustering. In general, when the percentage of retained information is high, the average values of SC are relatively low but relatively stable, mostly around 0.2. As the percentage of retained information decreases, the SC values are more likely to be affected by the number of clusters. When the percentage of retained information is less than 55%, the SC values reach about 0.5. In fact, when the percentage of retained information is 60% and the number of clusters is 4 or 5, the SC values are close to 0.4, which is still a not bad effect.

In addition, for the convenience of comparison, we check the maximum, average and minimum SC values of these 24 data samples according to the different number of clustering, but based on the different percentage of retained data. As shown in the Fig. 4.

From these figures in Fig. 4, on the one hand, we can see that when the number of clustering is small, the maximum, average, and minimum values of SC all show a trend of gradual increase with the decrease of the percentage of information retained; on the other hand, when the number of clusters is large, the SC values are not affected by the percentage of information retained, especially when the number of clusters is 7, the average values of SC are all below 0.2.

The reason is that when the percentage of retained information is high, some rare and some not so important single features are also retained. Each of these features often becomes a cluster by itself, and it is difficult to aggregate with each other, which violates the original intention of this method. As the amount of information retained decreases, so do the number of individual features, and the proportion of correlated features also keeps increasing, so the SC value keeps increasing as well, and the clustering effect is getting better and better.

On the other hand, although Chinese language has the characteristic of preserving the meaning of words, considering that if the percentage of retained information is too low, the loss of information will be very serious, which is undoubtedly a disaster for the discrimination and calculation of the final function points. Therefore, although the less information is retained and the better the clustering effect is, it is unfavorable to lose too much information.
| Results Percentage | Data Sets 1 | Number of Clusterings | Data Sets 2 | Number of Clusterings |
|--------------------|------------|-----------------------|------------|-----------------------|
| 95%                | 0.03959    | 0.03720               | 0.03824    | 0.03949               |
| 90%                | 0.03959    | 0.03720               | 0.03824    | 0.03949               |
| 85%                | 0.06248    | 0.05512               | 0.07174    | 0.08180               |
| 80%                | 0.06704    | 0.05512               | 0.10291    | 0.10870               |
| 75%                | 0.08472    | 0.10586               | 0.16930    | 0.22623               |
| 70%                | 0.08472    | 0.10586               | 0.16930    | 0.22623               |
| 65%                | 0.08472    | 0.09223               | 0.17469    | 0.13214               |
| 60%                | 0.15631    | 0.17335               | 0.24897    | 0.32789               |
| 55%                | 0.15631    | 0.17335               | 0.24897    | 0.32789               |
| 50%                | 0.15631    | 0.17335               | 0.24897    | 0.32789               |
| 45%                | 0.24589    | 0.31893               | 0.40018    | 0.38800               |

Table 4: SC values of clustering effect
| Percentage | Results Data Sets | Number of Clusterings |
|-----------|-----------------|----------------------|
| 95%       | 0.05716         | 0.06788              |
|           | 0.06247         | 0.05894              |
|           | 0.04931         | 0.02733              |
| 90%       | 0.07797         | 0.06574              |
|           | 0.08886         | 0.08187              |
|           | 0.09470         | 0.05542              |
| 85%       | 0.07921         | 0.10100              |
|           | 0.13156         | 0.15940              |
|           | 0.20903         | 0.10100              |
| 75%       | 0.11462         | 0.08983              |
|           | 0.16986         | 0.15940              |
|           | 0.20903         | 0.10100              |
| 65%       | 0.15951         | 0.24620              |
|           | 0.29336         | 0.36527              |
|           | 0.25892         | 0.15437              |
| 55%       | 0.16109         | 0.24620              |
|           | 0.36527         | 0.25892              |
|           | 0.15437         | 0.14282              |
| 45%       | 0.23189         | 0.35014              |
|           | 0.43840         | 0.34898              |
|           | 0.27412         | 0.12802              |
|           | 0.20496         | 0.07238              |

Table 4 (Continued)
| Data Sets | Scatter Diagram of Effect | Planar Diagram of Effect | Data Sets | Scatter Diagram of Effect | Planar Diagram of Effect |
|-----------|----------------------------|--------------------------|-----------|----------------------------|--------------------------|
| 1         | ![Image](49x741)           | ![Image](535x733)        | 7         | ![Image](49x741)           | ![Image](535x733)        |
| 2         | ![Image](49x741)           | ![Image](535x733)        | 8         | ![Image](49x741)           | ![Image](535x733)        |
| 3         | ![Image](49x741)           | ![Image](535x733)        | 9         | ![Image](49x741)           | ![Image](535x733)        |
| 4         | ![Image](49x741)           | ![Image](535x733)        | 10        | ![Image](49x741)           | ![Image](535x733)        |
| 5         | ![Image](49x741)           | ![Image](535x733)        | 11        | ![Image](49x741)           | ![Image](535x733)        |
| 6         | ![Image](49x741)           | ![Image](535x733)        | 12        | ![Image](49x741)           | ![Image](535x733)        |

Table 5  SC values of clustering effect
Table 5 (Continued)

| Data Sets | Scatter Diagram of Effect | Planar Diagram of Effect | Data Sets | Scatter Diagram of Effect | Planar Diagram of Effect |
|-----------|---------------------------|--------------------------|-----------|---------------------------|--------------------------|
| 13        | ![Image](image1.png)      | ![Image](image2.png)     | 19        | ![Image](image3.png)      | ![Image](image4.png)     |
| 14        | ![Image](image5.png)      | ![Image](image6.png)     | 20        | ![Image](image7.png)      | ![Image](image8.png)     |
| 15        | ![Image](image9.png)      | ![Image](image10.png)    | 21        | ![Image](image11.png)     | ![Image](image12.png)    |
| 16        | ![Image](image13.png)     | ![Image](image14.png)    | 22        | ![Image](image15.png)     | ![Image](image16.png)    |
| 17        | ![Image](image17.png)     | ![Image](image18.png)    | 23        | ![Image](image19.png)     | ![Image](image20.png)    |
| 18        | ![Image](image21.png)     | ![Image](image22.png)    | 24        | ![Image](image23.png)     | ![Image](image24.png)    |
From these figures in the Fig. 4, as far as the software requirements documents are concerned, the percentage of retained information should be around 55%~75%. At this time, if the number of clusters is 4 or 5, then the clustering effect can reach the best under the condition that sufficient or required information is retained.
4.4 Examples Analysis

4.4.1 The Results of the Test-Set Examples

For comparison purposes, we collected another 9 software requirement documents as samples. The function points of these documents have been calculated by the respective development teams. We also use the new method proposed in the paper to process these 9 samples separately, and then compare the values of function points with those calculated by the development team. The results are shown in the Table 6 below.

As can be seen from Table 6, the length of these 9 software requirements documents is different, with an average of 7022 Chinese characters. For these 9 samples of different lengths, the development team calculated the function points by completely manual, the maximum value is 116, the minimum is 33, the maximum time consuming is 3 working days, the minimum is 1 working day, the average time consuming is 2.33 working days. Using the new method proposed in this paper, the maximum value of func-
Table 7  Comparison of FP of digital pipeline management subsystem

| Number | Module                  | Name of Count Item for FP | Category | Results of the Team | Results of the New Method |
|--------|-------------------------|---------------------------|----------|---------------------|---------------------------|
| 1      | Head-Head               | ILF                       | 10       | 10                  | 10                        |
| 2      | Head-Count              | ILF                       | 10       | 10                  | 10                        |
| 3      | Count-Head              | ILF                       | 10       | 10                  | 10                        |
| 4      | Head-Integrated         | ILF                       | 10       | 2 Workdays          | 4 Hours                   |
| 5      | Integrated-Head         | ILF                       | 10       | 10                  | 4 Hours                   |
| 6      | Head-Storing            | ILF                       | 10       | 10                  | 10                        |
| 7      | Storing-Head            | ILF                       | 10       | 10                  | 10                        |
| 8      | Head-Turning            | ILF                       | 10       | 9                   | 9                         |
| 9      | Turning-Head            | ILF                       | 10       | 9                   | 9                         |
| 10     | Planning Basic Information | ILF                      | 10       | 1 Workday           | 2.5 Hours                 |
| 11     | Add                     | EI                        | 4        | 4                   | 4                         |
| 12     | Modify                  | EI                        | 4        | 4                   | 4                         |
| 13     | Delete                  | EI                        | 4        | 4                   | 4                         |
| 14     | Query                   | EQ                        | 4        | 4                   | 4                         |

| Total  | 116                      | 3 Workdays                | 113      | 6.5 Hours           |                           |

Accuracy of FP: 97.41%  Difference of FP: 3  Difference Percentage: 2.59%

In order to better clarify our conclusions, we use the first 2 of the 9 examples for specific explanation.

(1) Digital Pipeline Management Subsystem

This example comes from a software system to be developed by a subsidiary of a group company. The main business of the subsidiary includes, but not limited to, providing information system technology development and consulting services for the group, focusing on the research, design and development of digital pipeline management system, engineering design integration and other application software. They have a research and develop team of more than 100 people.

Let’s take one of the subsystems, the digital pipeline management system, as an example. The software requirements document for this subsystem is written in Chinese language, about 9200 words. We used the method of K-Means clustering based on TF-IDF proposed in the paper to preprocess the data in the requirement document, and then extracted the function points according to the results of clustering. Finally, we compared the results of this extraction with the function points calculated by their development team, as shown in Table 7.

From the Table 7, we can see that the subsystem has two modules that measures its data function and transaction function respectively. The development team spent 3 workdays to calculate the function points of the subsystem as 116, and we obtained the function points of the subsystem as 113 after processing the software requirements document by using the method proposed in this paper. Assuming that the function points calculated by the development team were standard values, our team only took 7 hours to obtain the function points of the subsystem with an accuracy rate of 97.41% and an error rate of only 2.59%.

(2) Equipment Warehouse Logistics Management Subsystem

This example comes from an equipment management system construction project. The project also has multiple subsystems, focusing on the development, management, technical support and assurance of the system, and has an inde-
Table 8  Comparison of FP of equipment warehouse logistics management subsystem

| Number | Module                  | Name of Count Item for FP | Category | Results of the Team | Results of the New Method |
|--------|-------------------------|---------------------------|----------|---------------------|----------------------------|
| 1      | User List               | User Information          | ILF      | 10                  | 10                         |
| 2      | Add                     | EI                        | 4        | 4                   | 4                          |
| 3      | Delete                  | EI                        | 4        | 4                   | 4                          |
| 4      | Modify                  | EI                        | 4        | 4                   | 4                          |
| 5      | Monitoring Rules        | Monitoring Information    | ILF      | Half of 1 Workday   | 10                         |
| 6      | Details of Transactions | Information on Transaction Details | ELF | Half of 1 Workday | 8                           |
| 7      | Submitted to the Head Office | Submission Behavior | EO       | 5                   | 5                          |
| Total  |                         |                           |          | 44                  | 45                         |

| Accuracy of FP | Difference of FP | Difference Percentage |
|----------------|------------------|------------------------|
| 97.73%         | 1                | 2.27%                  |

Let’s take one of the subsystems, the equipment warehouse logistics management subsystem, as an example. The software requirements document for this subsystem is written in Chinese language, about 6600 words. We used the method of K-Means clustering based on TF-IDF proposed in the paper to preprocess the data in the requirement document, and then extracted the function points according to the results of clustering. Finally, we compared the results of this extraction with the function points calculated by their development team. As shown in Table 8.

From the Table 8, we can see that the subsystem has four modules. The development team spent 2.5 workdays to calculate the function points of the subsystem as 44, while we obtained the function points of the subsystem as 45 after processing the software requirement documents by using the method proposed in this paper. Assuming that the function points calculated by the development team were standard values, our team only took 5.5 hours to obtain the function points of the subsystem with an accuracy rate of 97.73% and an error rate of only 2.27%.

Through the above two examples, it can be seen that if the function points calculated by their team are assumed to be completely accurate, then adopting the new method proposed in this paper does have a higher time efficiency in obtaining the values of function points, and its accuracy is also better. Therefore, for software requirements documents written in Chinese language, the new method proposed in this paper has practical significance in the research field of automatic calculation of data function points and transaction function points.

5. Validity Threats Analysis

In general, there are four main aspects to be analyzed in the consideration of validity threat. They are conclusion validity threat, internal validity threat, construct validity threat, and external validity threat [12].

As for conclusion validity threat, in the experiment conducted in this paper, we selected 24 software requirements documents as samples which covers a wide range of research fields, and after the experiments of each sample, there were obvious similarities in the results and almost the same conclusions. Therefore, we believe that our experiment has a statistically significant impact on the results, and its conclusion validity threat is low.

As for Internal validity threat, for comparison, we divided all documents of different lengths into the same sections based on the documents of the smallest length, and conducted experiments on the number of clustering from 2 to 7 respectively. It can also be seen from the experimental results that the different number of clustering does have an impact on the quality of the experimental results. In addition, we can also see from the experimental results that the difference in the percentage of retained information will also have an impact on the final experimental results.

Fortunately, in order to alleviate this threat, according to the observed results in the experimental results, when the extracted information is retained by 55% to 75% and the number of clusters is in the middle of the number of segments, not only is sufficient information retained, but the SC value of its conclusion can be obtained satisfactorily. This is a very important reference for the parameter setting of the new algorithm in document processing.

As for Construct validity threat, from the perspective of structure, the improved methods proposed in this paper are composed of recognition module and clustering module respectively. Its core algorithms are TF-IDF method and K-means method respectively. In the identification module, TF-IDF is used to obtain feature matrix; in the clustering module, K-means is used to obtain clustering results, meanwhile, SC value is used to carry out effect evaluation. We choose to apply the algorithms or theories mentioned above precisely for the purpose of obtaining the results that these algorithms should bring. Through the experiments, we did get these experimental results in line with expectations.

As for External validity threat, the new method proposed in this paper is aimed at software requirement documents written in Chinese language. Considering the similarity of East Asian languages, this method also has practical reference value for software requirement documents written in other East Asian languages. Meanwhile, in order to further reduce external validity threat, in the future, we will collect more software requirement documents in more other fields, including those written in other languages, in order to make our research results more universal.
6. Conclusion

Although at present, it is not possible to extract data function points automatically by using FPA method for software scale measurement, but in this paper, a method of K-Means clustering based on TF-IDF for software requirements documents written in Chinese language is proposed, which can be used for clustering before Chinese words classification and disambiguation. The result of clustering can be used as the input of subsequent entity recognition and disambiguation, which can improve the efficiency of entity determination, and also make the entity disambiguation be carried out over a smaller range. At the same time, this method further reduces the calculation amount of subsequent automatic calculation data function points and transaction function points, and clears the early technical obstacles for the final automatic calculation software scale.

In the experiments, we took 24 software requirements documents written in Chinese language as samples. The results show that the best clustering effect whose average SC value is even up to about nearly 0.4, is achieved when the extracted information is retained by 55% to 75% and the number of clusters is in the middle, that is, 4 to 5 clusters in those cases above, at the same time, much of the information is retained and has practical significance.

Not only for software requirement documents written in Chinese, but also for software requirement documents written in Oriental languages represented by Chinese language, the new improved method proposed in this paper, has practical reference significance in the research field of automatic calculation of data function points and transaction function points, which also fills the gaps in the field of automated software size measurement based on requirements documents.

Acknowledgments

This work is supported by the ‘National Key R&D Program of China’ (No. 2018YFB1403400), Natural Science Foundation of China (No. 61702544), Natural Science Foundation of Jiangsu Province, China (No.BK20160769, BK20141072), China Postdoctoral Science Foundation (No. 2016M603031).

References

[1] A.Z. Abualkishik and L. Lavazza, “IFPUG Function Points to COSMIC Function Points Convertibility: a Fine-Grained Statistical Approach,” Information and Software Technology, vol.97, pp.179–191, 2018.
[2] M. Ochodek and B. Ozgok, “Functional and Non-functional Size Measurement with IFPUG FPA and SNAP — Case Study,” Software Engineering in Intelligent Systems, vol.349, pp.19–33, 2015.
[3] A.J. Albrecht, “Measuring application development productivity,” Joint Share, Guide, & Ibm Application Development Symposium, 1979.
[4] O. Demirors and C. Gencel, “Conceptual Association of Functional Size Measurement Methods,” IEEE Software, vol.26, no.3, pp.71–78, 2009.
[5] M. Yao, D. Pi, and X. Cong, “Chinese Text Clustering Algorithm Based k-means,” Physics Procedia, vol.33, 2012.
[6] S. Zhang, W. Zhong, and H. Wei, “An improved Similarity Measure For Chinese Text Clustering,” DEStech Transactions on Engineering and Technology Research, 2016.
[7] Y. Luo, S. Zhao, L.I. Xiaochao, Y. Han, and Y. Ding, “Text keyword extraction method based on word frequency statistics,” 2016.
[8] Y. Wang, Z. Zhang, and W. Huo, “Research on aviation unsafe incidents classification with improved TF-IDF algorithm,” Modern Physics Letters B, vol.30, no.12, p.1650184, 2016.
[9] D. Kim, D. Seo, S. Cho, and P. Kang, “Multi-co-training for document classification using various document representations: TF–IDF, LDA, and Doc2Vec,” Information Sciences, vol.477, pp.15–29, 2018.
[10] H. Yan, X. Qiu, and X. Huang, “A Graph-based Model for Joint Chinese Word Segmentation and Dependency Parsing,” Transactions of the Association for Computational Linguistics, vol.8, no.2, pp.78–92, 2020.
[11] H.-T. Chang, S.-W. Liu, and N. Mishra, “A Tracking and Summarization System for Online Chinese News Topics,” Aslib Journal of Information Management, vol.67, no.6, pp.687–699, 2014.
[12] F. Robert and A. Magazinius, “Validity threats in empirical software engineering research an initial survey,” Seke, pp.374–379, 2010.

Jing Zhu received the B.E. degree from Jiangsu University of Science and Technology in 2006, and the M.E. degree from Navy Command College in 2011. He is a Ph.D. candidate in Software Engineering at Army Engineering University of PLA. He joined the Navy Command College as a lecturer in 2013, and his research interests include software engineering, pattern recognition and intelligent systems.

Song Huang received the B.S., M.S. and Ph.D. degrees from the PLA University of Science and Technology. He is a professor of software engineering at Software Testing and Evaluation Center from Army Engineering University of PLA. He is a member of the advisory boards of Journal of Systems and Software, IEEE Transactions on Reliability. His research interests include software testing and empirical software engineering.

Yaqing Shi received the B.S. degree from Xi’an Technological University in 2004, and the M.S. and the Ph.D. degrees from Hehai University in 2007 and 2015, respectively. She is an associate professor in Army Engineering University of PLA, and her research interests include spatiotemporal indexing and searching methods and ITS. Her awards and honors include the General Financial Grant (No.2016M603030).
**Kaishun Wu**  received the B.S. degree from Zhejiang University in 2015, and the M.S. degree from the Army Engineering University of PLA in 2018. He has been a Ph.D. candidate in Software Engineering at Army Engineering University of PLA since 2018. His research interests include software testing, software quality evaluation, and crowdsourcing testing.

**Yanqiu Wang**  received the B.E. degree in 1982. She had been a lecturer in the institute before retiring, teaching Information Fusion, C++, Machine Learning, and she also has been a private teacher who teaches relevant courses in another private institution. She has now been rehired to continue her research, which includes fuzzy reasoning, Asian language studies and so on.