Determining Extractive Summary for a Single Document Based on Collaborative Filtering Frequency Prediction and Mean Shift Clustering

Ahmed M. El-Refaiy, Ahmed R. Abas, and Ibrahim M. El-Henawy

***Abstract*—This paper presents a new unsupervised algorithm for determining extractive summary for a single document using term frequency prediction, which is obtained from memory-based collaborative filtering (CF) approach, and Mean Shift Clustering algorithm. The new algorithm uses Term-Sentence Collaborative Filtering (TSCF) for predicting term frequency. These term frequencies are used in sentence ranking according to the presence percentage of each word/term in each sentence. TSCF computes term frequencies for either terms present or missing (sparse) in a sentence via collaborative filtering prediction algorithm. The new algorithm uses Mean Shift Clustering algorithm as a final framework to group sentences according to their ranks to get more coherent summaries. Experiments show the effect of using different weighting functions including: Term Frequency (TF), Term Frequency Inverse Document Frequency (TFIDF) and binary TF. In addition, they show the effect of using different distance metrics that support sparse matrices representations including: Cosine, Euclidean and Manhattan. Experiments also, show the effect of using L1 and L2 normalization. ROUGE is used as a fully automatic metric in text summarization on DUC2002 datasets. Results show ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-SU4 average recall, precision and f-measure scores, which show the effectiveness of the new algorithm. Results show that the proposed TSCF algorithm has promising results and outperforms related baseline techniques in many ROUGE scores.**

***Index Terms*—Extractive Text Summarization, Collaborative Filtering Prediction, Term frequency, Information retrieval, Mean Shift Clustering.**

I. Introduction

AUTOMATIC text summarization aims at generating a concise piece of text from one or more documents. Text summarization is classified into two main classes: abstractive and extractive; where in abstractive class, the summarization model aims to reformulate the generated summary text; however, in extractive class, the summarizer generates summary by picking up the most prominent sentences based on ranking model. Therefore, extractive category is usually regarded as sentence-ranking model [1], [2]. Also in other research studies, extractive summarizers are constructed under the basis of a selection model which select sentences based on their prestige or saliency inside the text [3], [4]; so, it's desirable to build a good sentence ranking model first.

The earlier approach to automatically summarize text was in the late fifties [5]. Text summarization task has several forms; particularly, based on input type summarization can work on a single document or multi documents. For summarization content type, it can produce generic summary (not user specific) or query-oriented summary (based on user query). Also Summarization technique can be supervised or unsupervised. Our paper focuses on proposing an unsupervised extractive generic single document summarization approach.

Collaborative filtering (CF) presented strong promises in recommender systems for making automatic prediction or filtering about user interests (books, products, web pages, articles, etc) which mean items or information sources [6]. CF contains two types: Memory-based and Model-based CF; where in Memory-based the user rating data is used to calculate similarity between users or items via user-based approach or item-based approach using similarity matrices. On the other hand, Model-based approach uses machine learning and data mining techniques for prediction [7]. Furthermore, CF meets text summarization for personal interest summary which called personalized summarization [8], [9], [10]. For instances, Collaborative summarization approach proposed for producing personalized single-document summarization via tag recommendation with the help of affinity graph [8]; another approach using expanded social contextual information that catch user interest to give after that personalized summary [9]; personalized web news filtration approach for maintaining keywords knowledge base integrated with lexical chain technique for the summarization process [10].

In this paper, it is proposed an unsupervised algorithm for determining extractive summary for a single document. This algorithm, called Term-Sentence Collaborative Filtering (TSCF) is based on Memory-based Collaborative Filtering [11], [12], [13] and Mean Shift Clustering [14]. The proposed algorithm computes for every sentence in the document the term frequency percentage of each word/term that is founded in the document either this term is found in the sentence or missing. Afterwards, Mean Shift Clustering is applied as another sentence ranking and selection model

Ahmed Mohamed El-Refaiy is a Teaching Assistant with Department of Computer Science, Faculty of Computers and Informatics, Zagazig University, 44519, Egypt. Email: amnasser@zu.edu.eg.

Ahmed Rafat Abas is a Lecturer with Department of Computer Science, Faculty of Computers and Informatics, Zagazig University, 44519, Egypt. Email: arabas@zu.edu.eg.

Ibrahim Mahmoud El-Henawy is a Professor with Department of Computer Science, Faculty of Computers and Informatics, Zagazig University, 44519, Egypt. Email: Henawy2000@yahoo.com

to enhance summarization process. Experiments show the effect of using different representations of term weighting functions (TF, binary TF and TFIDF) and different sentence similarity metrics (Cosine, Manhattan and Euclidean). The proposed algorithm is tested with L1 and L2 normalization methods. Finally, results show that the proposed algorithm has promising results and outperforms other baseline related techniques on DUC2002 dataset.

II. Related Work

Previous techniques with comparable ideas, namely implementing sentence similarity integrated with selection model optimization or analyzing semantic orientation for representing contextual meanings and the similarity of sentences, occupied large proportion. Regarding the promising works LexRank [15] and LexPageRank [3], Both applying page ranking algorithm for computing sentence prestige and saliency after preparing cosine similarity on the basis of TFIDF matrix which helps to obtain good information coverage and gives it the ability to work with noisy data. Furthermore, another page ranking-based algorithm, TextRank [16] proposed for sentence and keyword extraction based on similarity representation which considered as language and domain independent. The overall advantages of the previous graph ranking techniques are the ability to generate topic specific summaries. But, the accuracy of such algorithms depends on the selected affinity function.

On the other hand, Latent Semantic Analysis (LSA) a semantic orientation-based approach which able to represent contextual meaning of words, was presented via several models including Gong and Liu [17], Steinberger and Jezek [18], Murray, Renals and Carletta [19] and Ozsoy [20] which meet on the first two steps and differ with each other on the final step (sentence selection). LSA is also used in many applications including information filtering as it has a promising work with CF; practically, the model-based CF can be done based on Matrix Factorization (MF) and LSA or sometimes namely Singular Value Decomposition (SVD) is a well-known MF method. LSA give summaries containing most information with least noise due to its dimensionality reduction, but it suffer from time consuming as it depends on SVD computations. Besides that, LSA still suffer from polysemy problem which means the same words with different meanings have the same concepts.

Moreover, many researches have sought to extend these traditional summarization models. For instance, LSA is integrated with Fuzzy logic where each model find its own summary and then both summaries intersected to build final one [21]. In [22], lexical association is used to find representative keywords of text topic and then calculate keywords weight by graph-based ranking algorithm to easily score sentences. This technique was able to produce coherent summary due to lexical association usage. In addition to the usage of lexical association keyword extraction strategy with graph ranking, Ravinuthala [23] proposed new aided strategy in vertices connections which increases incoming edges for topic (theme) central words.

Deep learning (DL) techniques have been used for the single document summarization task via Deep Auto-Encoder (AE) where AE attempts to learn features representation to extract high informative summaries [24]. This approach presented a good solution for sparsity problem on the basis of local term frequency usage with randomly added noise, but it suffers from training computational cost and the requirement of tuning the training hyper-parameters. Another DL technique based on recurrent neural network where summarization task is solved as sequence classification task to check availability for choosing sentence to the summary or not [25]. For the same idea of the classification task, Fuzzy inference model have been implemented over neural network (NN) framework to automatically build fuzzy rules without human experts then use this model to classify sentences [26].

Furthermore, another related unsupervised models have been proposed. K-mean, Louvain and Agglomerative nested are the most used clustering techniques with single document summarization. For instance, “K-mean Clustering algorithm” is used in [27] as a final framework after building a document graph (nodes - edges), to group the coherent-sentences together based on correlation degree with user’s query. In [28], Louvain clustering algorithm is used to cluster words, after that each word is scored based on the summation of several scoring approaches including: word score based on dependency relations, strengthen word score if it was mentioned in another related word and term frequency score of each word. So that, it is easy to form summary by picking up top scored sentences. In [29], the hierarchical “Agglomerative nested clustering” approach was used as middle framework for single document summarization task. After document was represented by “Vector Space Modeling” with the usage of Term Frequency-Inverse Sentence Frequency (TF-ISF), sentences were clustered using the hierarchical approach based on cosine similarity. Thereafter, the final score of each sentence was formed by summing up “sentence similarity score with other sentences in the same cluster” with “sentence similarity score with document title” and then, the top two ranked sentences from each cluster were picked up to form the final summary. Fuzzy-logic algorithm is used in [30] where a combination of the fuzzy sets and roles is built to work with nine features including number of proper nouns, length, centrality and position of the sentence, … etc, to score sentences.

The proposed algorithm is based on memory-based CF technique, but it differs from other summarization techniques which integrated with CF [8], [9], [10]. As these techniques are usually built for personalized summarization task on the basis of personal interest; so, the recommender system is used as helper framework. While the proposed algorithm uses CF technique as a framework for generic extractive single document summarization task. Instead of applying filtering among users and tags or documents to know user interest, the proposed algorithm apply filtering among terms and sentences for term frequency prediction.

III. The Proposed Algorithm

The proposed approach is built on the basis of Memory-based CF approach, especially in the user-based type or sometimes called user-item filtering; where for a given user, it finds users similar to that user based on ratings similarities and then recommends and predicts items that those similar users liked. Therefore, the user-item CF approach needs to build user-item matrix MxN, where M represents the number of users and N represents the number of items. Each cell in this matrix represents user rating for each item. In the proposed summarization approach, the matrix is called term-sentence matrix, where for a particular term we find terms similar to that term based on similarity in frequencies and then, we predict frequency of this particular term for each sentence in which those similar terms appeared. Therefore, the proposed approach is called Term-Sentence Collaborative Filtering.

The proposed approach is presented in Algorithm 1.

Algorithm 1: Term-Sentence Collaborative Filtering summarization approach

<u>Summarize</u> (*d*,*wf*,*dm*,*length*);

Input: Document *d* to be summarized, *wf* is weighting function used for building term-sentence matrix (*TFIDF* is default chosen weighting function), *dm* is the distance metric used to calculate distances between terms (*Cosine* metric is default chosen distance metric) and *length* is summarization percentage to be returned from the whole sentences.

Output: A subset *c* sentences regarding to *length* from *d*, where *c* is the most salient sentences.

1- Read document sentences *N*;
2- Apply Tokenization process to get list of sentences *N*;
3- Build term-sentence matrix *MxN* using *wf* weight function, where *M* is maximum reached n-gram terms started from unigram as min;
4- Applying *L1* or *L2* normalization to remove amplitude variation and focus on the underlying distribution shape (*L2 normalization* is default chosen alternative);
5- Apply *dm* distance metric between terms *M* to build term-based similarity matrix;
6- Apply term-based CF *(7)* to predict term frequencies, so the term-sentence matrix is updated with new weights for each term;
7- Sum each sentence keyword's weights to get sentences scores;
8- Apply Mean-Shift Clustering algorithm to cluster sentences based on their scores;
9- Rank sentences in each cluster in ascending order based on the distance between sentence score and its cluster centroid;
10- Select subset *c* sentences from each cluster based on *(10)* where default summary length is 0.3;
11- Rearrange picked up sentences in the same order they appeared on the original document.

After reading the document, sentences are tokenized via unsupervised algorithm [31] to divide the text into a list of sentences (N). Also the list of terms (M) is obtained from the text and the term-sentence matrix MxN is created, where cells represent rating of words to sentences (importance of words in sentences) which can be calculated using weighting functions. The proposed approach is experimented using different weighting functions including:

1. Normal Term Frequency (TF):- which calculates the number of times that each term appear in each sentence.
2. Binary TF: - the Boolean form of term frequency is used where all non-zero counts are set to 1.
3. TFIDF:- $TF \, x \, IDF$, where TF is Normal Term Frequency and IDF is calculated via the following formula:

$$IDF(t) = \log \frac{n_d}{df(d,t)} + 1, \tag{1}$$

Where n_d represent the total number of sentences and $df(d,t)$ is the number of sentences containing term t. TFIDF is the chosen weighting function in our proposed model based on the later discussed experiments.

Afterwards, the term-sentence matrix is normalized through two different experiments including: “applying L1 normalization” or “applying L2 normalization”. Normalization is used to prune amplitude variation and focus on the underlying distribution shape. L1 and L2 normalization for x vector of covariates of n length can be calculated via:

$$\|x\|_1 = \sum_{i=1}^{n} |x_i| \tag{2}$$

$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} x^2{}_i} \tag{3}$$

Where $\|x\|_1$ is L1 normalization and $\|x\|_2$ is L2 normalization. Applying L2 normalization is the chosen alternative for our proposed model based on the later discussed experiments.

To be able to update the term-sentence matrix with predicted frequencies, it's needed to calculate similarities and create similarity matrix. Due to the usage of term-based CF, the similarity values between terms are calculated using only correlated sentences. Three different distance metrics are experimented including:

– Cosine similarity:

Use normalized dot product of terms vectors; for instance, if $T1$ and $T2$ are row vectors, their Cosine similarity $S(T1,T2)$ is defined as:

$$S(T1,T2) = \frac{T1.T2^T}{\|T1\|\|T2\|} \tag{4}$$

– Euclidean distance or also called l2 distance:
If $T1$ and $T2$ are term row vectors, their Euclidean distance $Euc(T1,T2)$ is defined as:

$$Euc(T1,T2) = \sqrt{\|T1\|^2 + \|T2\|^2 - 2T1.T2} \quad (5)$$

– Manhattan distance or L1 distance:
If $T1$ and $T2$ are term row vectors, their Manhattan distance $M(T1,T2)$ is the sum of absolute differences of their Cartesian coordinates and is defined as:

$$M(T1,T2) = \|T1 - T2\| = \sum_{i=1}^{n} |T1_i - T2_i| \quad (6)$$

Cosine distance metric is the chosen alternative for our proposed model based on the later discussed experiments. After creating the term similarity matrix, prediction is applied to balance sentences by updating existing weights (frequencies) of the term-sentence matrix and finding weights that are missing. Equation (7) used with user-based CF[13] is used to calculate the predicted weight $\hat{x}_{k,m}$ of term k for sentence m.

$$\hat{x}_{k,m} = \bar{u}_k + \frac{\sum_{u_a} s_u(u_k,u_a)(x_{a,m} - \bar{u}_a)}{\sum_{u_a} s_u(u_k,u_a)} \quad (7)$$

Where $\bar{u}_k$ and $\bar{u}_a$ represent the average weighting made by terms k and a, respectively. And $s_u(u_k, u_a)$ represents the similarity between terms k and a.

As previous equation represents, the predicted weight $\hat{x}_{k,m}$ of term k for sentence m is relied on the similarity between term k and terms a as weights that are multiplied by weights of similar terms a (corrected by average weighting value of that term) and then normalize it; so that new weights stay between min and max of weight values. And as final step, the average weights of the term k is added to the normalized values.

After updating term-sentence matrix weights, sentences have weight vector $S_{i,tw}$ (terms' weights for each sentence):

$$S_{i,tw} = (W_{i,t1}, W_{i,t2}, \ldots, W_{i,tm}) \quad (8)$$

Where $W_{i,tj}$ represents updated weight of term j for sentence i and m is number of all terms. The score of each sentence $Score(S_i)$ is calculated by sum all weights of each sentence via:

$$Score(S_i) = \sum_{j=1}^{m} W_{i,tj} \quad (9)$$

After getting sentences scores, a cluster algorithm is used as second sentence-ranking framework to group coherent sentences together. We use Mean Shift clustering based on the algorithm discussed in [14], where it aims to discover blobs in data samples. It is a centroid-based algorithm, where the candidates for centroid points are updated to be the mean points within a given region. After that, these candidates are filtered iteratively to eliminate near duplications and the iterations stopped when the changes in centroids is small to form the final centroids. So, the algorithm automatically sets the number of clusters.

After Applying Mean Shift Clustering and getting clusters of sentences, we re-rank sentences in each cluster in an ascending order based on the distance between sentence score and its cluster centroid.

We now ready to apply the selection process to form final summary. We select a subset c sentences from each cluster which represent most important ones, based on the following equation:

$$c = \frac{Num1}{Num2} \times Len \quad (10)$$

Where *Num1* is number of sentences in each cluster, *Num2* is number of sentences of the text document and *Len* is the summary length which is calculated according to the compression rate factor, which is the ratio between summary length and original length of the document. When decreasing compression rate the summary will be short and suffer from information loss. Otherwise, the summary will be more abundant and relatively contains trivial information if the compression rate is increased. In practice, summarization quality is acceptable with 5-30% compression rate[32], [33]. The user can determine the compression rate degree or it's by default 30%.

In case of the cluster contains one sentence only, we don't apply (10) and instead, we pick up this unique sentence automatically because it may contains information which not similar to other sentences.

Finally, we rearrange the picked up sentences in the same order they appeared on the original document.

IV. Experimental Results

A. *Dataset and Setup*

For overall approach exploration, different experiments are carried out for single document summarization task on DUC2002 [34] which has standard dataset contains original documents and reference summaries. The standard officially ROUGE [35]; especially, (version 2.0) [36], [37] toolkit is used for evaluation. ROUGE measures summarization quality by counting n-gram overlapping between model summary (generated by machine) and the reference summary (generated by human). Results are shown for the average Recall, Precision and F-Measure scores obtained from ROUGE-1, which is based on unigram matching, ROUGE-2, which is based on bigram matching, ROUGE-L, which is based on Longest Common Subsequence [38] and ROUGE-SU4 which is based on the measuring the overlap of skip-bigrams between system summary and reference summary with a maximum skip distance of 4.

The compression rate used is 30%. The algorithm is implemented in Python; the weighting functions, distance metrics, normalization forms and Mean Shift algorithm are implemented via scikit-learn open source python tool [39]. Term-Sentence matrix is built using min-gram terms equal 1 to max-gram terms equal 3 (to be unigram, 2-gram and 3-gram terms). The maximum 3-gram terms is chosen to our model after different experiments as it give best evaluation results.

Before applying mean shift algorithm, we intend to choose best alternative between (weighting function choices, normalization process choices and distance metric choices). So, our TSCF model was executed with different permutations as described in **Table I** and we select the permutation having the best summary result to apply mean shift algorithm on it. In order to form summaries for each permutation, the sentences are ranked by their scores (after applying (9)) in descending order. The highly ranked sentences are selected to form the summary according to the compression rate degree of 30%.

Due to our experimental results which will be discussed next, the ("TFIDF weighting function", "L2 normalization process" and "Cosine distance metric") are chosen to be the best alternative for our proposed model as they give most prominent results in our experiments. Therefore, we apply Mean Shift Clustering algorithm on these chosen criteria.

B. *Results and Discussion*

After we got summaries for each permutation described in **Table I**, we calculate ROUGE-1 recall, precision and f-measure results for them as described in **Table II**. The results show that "TSCF-TFIDF $_{\text{cosine, l2 norm}}$" permutation give us best prominent result due to the existence of (TFIDF weighting function, cosine distance metric and L2 normalization) with values equal to 0.7213, 0.3759, and 0.4692 for Recall, Precision and F-measure respectively.

TFIDF is the chosen weighting function in our proposed model as it is the best alternative that reflect the relevant of keywords in sentences and it gives best results among other alternative in our experiments. The results reflect the importance of L2 normalization process. So, L2 normalization is the chosen normalization process rather than L1. It produces non-sparse outputs, unlike L1 normalization which produce outputs with zero or very small values and this is obvious in the results where all permutations contains L1 normalization have not good results but, all permutation contains L2 normalization have better results. Due to the non-sparsity outputs of L2 normalization, our model is handling sparsity problem. For distance metric alternatives (Cosine, Euclidean, Manhattan), the results show that Cosine is the best one as it gives us best results. Euclidean and Manhattan results are convergent with

TABLE I
TSCF model different executed permutations

Permutation Name	Used Weighting function	Used Distance metric	Used Normalization
TSCF-Frequency euclid, l2 norm	TF	Euclidean	L2
TSCF-TFIDF manh, l2 norm	TFIDF	Manhattan	L2
TSCF-TFIDF euclid, l2 norm	TFIDF	Euclidean	L2
TSCF-Frequency cosine, l2 norm	TF	Cosine	L2
TSCF-Binary manh, l2 norm	Binary TF	Manhattan	L2
TSCF-Binary euclid, l2 norm	Binary TF	Euclidean	L2
TSCF-Binary cosine, l2 norm	Binary TF	Cosine	L2
TSCF-TFIDF cosine, l2 norm	TFIDF	Cosine	L2
TSCF-Frequency manh, l2 norm	TF	Manhattan	L2
TSCF-Frequency manh, l1 norm	TF	Manhattan	L1
TSCF-Binary manh, l1 norm	Binary TF	Manhattan	L1
TSCF-TFIDF manh, l1 norm	TFIDF	Manhattan	L1
TSCF-TFIDF euclid, l1 norm	TFIDF	Euclidean	L1
TSCF-Frequency euclid, l1 norm	TF	Euclidean	L1
TSCF-Binary euclid, l1 norm	Binary TF	Euclidean	L1
TSCF-Binary cosine, l1 norm	Binary TF	Cosine	L1
TSCF-Frequency cosine, l1 norm	TF	Cosine	L1
TSCF-TFIDF cosine, l1 norm	TFIDF	Cosine	L1

Cosine results when TFIDF weighting function used with the applying of L2 normalization.

Other permutations including: "TSCF-TFIDF $_{\text{manh, l2 norm}}$", "TSCF-Frequency $_{\text{euclid, l2 norm}}$" and "TSCF-Frequency $_{\text{cosine, l2 norm}}$" give convergent results to the best selected permutation "TSCF-TFIDF $_{\text{cosine, l2 norm}}$" due to the usage of L2 normalization which play main role in enhancing results.

We select "TSCF-TFIDF $_{cosine,\ l2\ norm}$" permutation to complete Mean Shift Clustering algorithm on it.

TABLE II
ROUGE-1 scores of the proposed approach with different permutations

Name	Avg. Recall	Avg. Precision	Avg. F-measure
TSCF-Frequency euclid, l2 norm	**0.7148**	**0.3723**	**0.4654**
TSCF-TFIDF manh, l2 norm	**0.7164**	**0.3750**	**0.4683**
TSCF-TFIDF euclid, l2 norm	0.6910	0.3587	0.4467
TSCF-Frequency cosine, l2 norm	**0.7060**	**0.3725**	**0.4624**
TSCF-Binary manh, l2 norm	0.6727	0.3661	0.4520
TSCF-Binary euclid, l2 norm	0.6945	0.3585	0.4495
TSCF-Binary cosine, l2 norm	0.7073	0.3643	0.4563
TSCF-TFIDF cosine, l2 norm	**0.7213**	**0.3759**	**0.4692**
TSCF-Frequency manh, l2 norm	0.6359	0.3637	0.4337
TSCF-Frequency manh, l1 norm	0.5218	0.4305	0.4500
TSCF-Binary manh, l1 norm	0.4245	0.4953	0.4332
TSCF-TFIDF manh, l1 norm	0.4460	0.4798	0.4404
TSCF-TFIDF euclid, l1 norm	0.4091	0.5032	0.4387
TSCF-Frequency euclid, l1 norm	0.4183	0.4373	0.4082
TSCF-Binary euclid, l1 norm	0.3913	0.5001	0.4194
TSCF-Binary cosine, l1 norm	0.3630	0.4180	0.3739
TSCF-Frequency cosine, l1 norm	0.3643	0.4186	0.3728
TSCF-TFIDF cosine, l1 norm	0.3477	0.4394	0.3740

In this table, bold numbers are the best convergent results

C. *Comparison with Other Algorithms*

Results are shown for the proposed algorithm ("TSCF-TFIDF $_{cosine,\ l2\ norm}$" only) which called "**TSCF only**" and ("TSCF-TFIDF $_{cosine,\ l2\ norm}$" with Mean Shift clustering using unigram terms only) which called "**TSCF-Mean Shift unigram**" and ("TSCF-TFIDF $_{cosine,\ l2\ norm}$" with Mean Shift clustering using unigram, 2-gram and 3-gram terms) which called "**TSCF-Mean Shift 1:3-gram**" compared with – re-implemented – related baseline works including (LSA [17], [18], [20], TextRank [16] and LexRank [15]) and also other related techniques that reported for DUC2002 dataset including (Fuzzy-logic [30], Louvain clustering with dependency graph [28], Graph ranking + lexical association [22], SummaRuNNer [25], Summarization system based on Vertex in-degree strength as KS-KWIS [23] and UniformLink+bern +neB [40]).

Table III presents average ROUGE-1 recall, precision and f-measure for the proposed approach with the related techniques. Results show that the proposed "TSCF-Mean Shift 1:3-gram" approach outperforms all other techniques in recall scores with value equal to 0.7536. **Fig. 1(a)** shows the average ROUGE-1 recall for all techniques. The proposed approach "TSCF only" gives a good result in precision scores with value equal to 0.3758 and the KS-KWIS model outperforms our models with value equal to 0.5143. **Fig. 1(b)** shows the average ROUGE-1 precision for all techniques. For f-measure scores, KS-KWIS and Fuzzy-logic model outperform us with values equal to 0.5605 and 0.4702 respectively; but, the proposed "TSCF only" approach still outperforms all other remaining techniques with value equal to 0.4692. **Fig. 1(c)** shows the average ROUGE-1 f-measure for all techniques.

TABLE III
ROUGE-1 scores of the proposed approach with other related techniques

Name	Avg. Recall	Avg. Precision	Avg. F-measure
TSCF-Mean Shift 1:3-gram	**0.7536**	0.3264	0.4651
TSCF-Mean Shift unigram	0.6867	0.3061	0.4332
TSCF only	**0.7213**	0.3758	**0.4692**
LSA2001	0.6726	0.3849	0.4621
LSA2004	0.6620	0.3834	0.4578
LSA2011	0.6242	0.3657	0.4346
TextRank	**0.7252**	0.3464	0.4470
LexRank	0.5186	**0.4395**	0.4396
Fuzzy-Logic	0.4666	**0.4759**	**0.4702**
Louvain clustering with dependency graph	0.488	____	____
Graph ranking + lexical association	0.4865	____	____
SummaRuNNer	0.466	____	____
UniformLink + bern + neB	0.4643	____	____
KS-KWIS	0.6164	**0.5143**	**0.5605**

In this table, bold numbers are the top three best results.

Table IV shows average ROUGE-2 recall, precision and f-measure for the proposed approach with the related techniques. Results show that the proposed approach "TSCF-Mean Shift 1:3-gram" outperforms all other techniques in recall scores with value equal to 0.5220. **Fig. 2(a)** shows the average ROUGE-2 recall for all techniques. The proposed approach "TSCF-Mean Shift 1:3-gram" outperforms all other techniques in precision scores with value equal to 0.2347 and just the KS-KWIS model outperforms the proposed approach with value equal to 0.4032. **Fig. 2(b)** shows the average ROUGE-2 precision for all techniques. For f-measure scores, KS-KWIS model outperform us with values equal to 0.4398; but, our "TSCF-Mean Shift 1:3-gram" approach still outperforms all other remaining techniques with value equal to 0.3301. **Fig. 2(c)**

shows the average ROUGE-2 f-measure for all techniques.

Also our proposed approach “TSCF-Mean Shift 1:3-gram” outperforms results of “TSCF only” and this reflect the importance of Mean Shift Clustering framework and number of grams used. “TSCF-Mean Shift 1:3-gram” increase recall, precision and f-measure results by 20%, 4% and 9% respectively. The usage of Mean Shift algorithms helps to get more coherent summaries. Also Usage of 2-gram terms and 3-gram terms with unigram increase coherency.

TABLE IV
ROUGE-2 scores of the proposed approach with other related techniques

Name	Avg. Recall	Avg. Precision	Avg. F-measure
TSCF-Mean Shift 1:3-gram	**0.5220**	**0.2347**	**0.3301**
TSCF-Mean Shift unigram	**0.4900**	**0.2272**	**0.3173**
TSCF only	0.3243	0.1945	0.2449
LSA2001	0.2474	0.1680	0.1984
LSA2004	0.3022	0.2016	0.2369
LSA2011	0.2544	0.1633	0.1963
TextRank	0.3120	0.1673	0.2210
LexRank	0.2452	0.2255	0.2226
Graph ranking + lexical association	0.3993	____	____
SummaRuNNer	0.2310	____	____
UniformLink + bern + neB	0.2070	____	____
KS-KWIS	**0.4841**	**0.4032**	**0.4398**

In this table, bold numbers are the top three best results.

Table V shows average ROUGE-L recall, precision and f-measure for the proposed approach with the related techniques. The results show that the proposed “TSCF-Mean Shift 1:3-gram” model outperforms all other techniques in recall scores with value equal to 0.6701. **Fig. 3(a)** presents the average ROUGE-L recall for all techniques. The proposed approach “TSCF-Mean Shift 1:3-gram” outperforms all other techniques in precision scores with value equal to 0.3287 and only LexRank model outperforms us with convergent value equal to 0.3383. **Fig. 3(b)** shows the average ROUGE-L precision for all techniques. The proposed approach “TSCF-Mean Shift 1:3-gram” outperforms all other techniques in f-measure scores with value equal to 0.4480. **Fig. 3(c)** shows the average ROUGE-L f-measure for all techniques.

Our “TSCF-Mean Shift 1:3-gram” model compared with “TSCF only”, increase recall, precision and f-measure results by 14%, 2% and 6% respectively.

Table VI shows average ROUGE-SU4 recall, precision and f-measure for the proposed approach with the related techniques. The results show that the proposed “TSCF-Mean Shift 1:3-gram” model outperforms all other techniques in recall scores with value equal to 0.5528. **Fig. 4(a)** presents the average ROUGE-SU4 recall for all techniques. The proposed approach “TSCF-Mean Shift 1:3-gram” outperforms all other techniques in precision scores with value equal to 0.2419 and only LexRank outperform us with convergent value equal to 0.2502. **Fig. 4(b)** shows the average ROUGE-SU4 precision for all techniques. The proposed “TSCF-Mean Shift 1:3-gram” approach outperforms all other techniques in f-measure scores with value equal to 0.3432. **Fig. 4(c)** shows the average ROUGE-SU4 f-measure for all techniques.

Our “TSCF-Mean Shift 1:3-gram” model compared with “TSCF only”, increase recall, precision and f-measure results by 18%, 3% and 7% respectively.

TABLE V
ROUGE-L scores of the proposed approach with other related techniques

Name	Avg. Recall	Avg. Precision	Avg. F-measure
TSCF-Mean Shift 1:3-gram	**0.6701**	**0.3287**	**0.4480**
TSCF-Mean Shift unigram	**0.5978**	0.2911	**0.4006**
TSCF only	**0.5264**	0.3019	**0.3860**
LSA2001	0.4911	0.2849	0.3617
LSA2004	0.4568	**0.3061**	0.3579
LSA2011	0.4037	0.2527	0.3095
TextRank	0.5164	0.2841	0.3676
LexRank	0.4168	**0.3383**	0.3608
SummaRuNNer	0.4303	____	____
Louvain clustering with dependency graph	0.44	____	____

In this table, bold numbers are the top three best results.

TABLE VI
ROUGE-SU4 scores of the proposed approach with other related techniques

Name	Avg. Recall	Avg. Precision	Avg. F-measure
TSCF-Mean Shift 1:3-gram	**0.5528**	**0.2419**	**0.3432**
TSCF-Mean Shift unigram	**0.5075**	**0.2272**	**0.3212**
TSCF only	**0.3756**	0.2096	**0.2711**
LSA2001	0.2825	0.1793	0.2151
LSA2004	0.3465	0.2235	0.2669
LSA2011	0.2961	0.1771	0.2187
TextRank	0.3632	0.1807	0.2451
LexRank	0.2747	**0.2502**	0.2469

In this table, bold numbers are the top three best results.

V. Conclusion and Future Work

This paper presents the Term-Sentence Collaborative filtering (TSCF) unsupervised algorithm via term-based approach similar to user-based collaborative filtering to solve extractive single document summarization task as sentences ranking model. TSCF aims to balance all sentences by updating existing terms' weights and predict other missing ones. Also, the proposed algorithm uses Mean Shift clustering algorithm to enhance the obtained summary, reduce redundancy and get more coherent sentences.

The proposed algorithm is fast and easy and don't suffer from time consuming problem like other related algorithms. The usage of L2 normalization improves results due to its non-sparsity outputs and the usage of Mean Shift clustering as second sentence ranking model improves the coherence. The disadvantage of the proposed algorithm is that it has, similar to all LSA techniques, the polysemy problem. Polysemy means the same words with different meanings have the same concepts. Compared to other algorithms, the proposed algorithm gives promising results and outperforms related baseline algorithms. It produces results equal to 75%, 37%, and 46% for ROUGE-1 Recall, Precision and F-measure respectively. It also produces results equal to 52%, 23% and 33% for ROUGE-2 Recall, Precision and F-measure respectively. In addition, it produces results equal to 67%, 32% and 44% for ROUGE-L Recall, Precision and F-measure respectively. Finally, it produces results equal to 55%, 23% and 34% for ROUGE-SU4 Recall, Precision and F-measure respectively.

In the future, the proposed algorithm could be extended by applying different clustering algorithms or dynamic programming algorithms as final selection stage to obtain better information coverage with least noise in order to improve results. In addition, lexical association could be used to build coherent summaries and to solve polysemy problem. Also, the proposed algorithm could be evaluated on different domains. Finally, the proposed algorithm could be used with multi-document summarization

References

[1] H. P. Edmundson, "New methods in automatic extracting," J. ACM, vol. 16, no. 2, pp. 264–285, 1969.

[2] A. Siddharthan, "Inderjeet Mani and Mark T. Maybury (eds). Advances in Automatic Text Summarization. MIT Press, 1999. ISBN 0-262-13359-8. 442 pp. $47.95/£ 32.95 (paperback).," Nat. Lang. Eng., vol. 7, no. 3, p. 271, 2001.

[3] G. Erkan and D. R. Radev, "Lexpagerank: Prestige in multi-document text summarization," in Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, 2004.

[4] Y. Pei and W. Yin, "Generic multi-document summarization using topic-oriented information," in Pacific Rim International Conference on Artificial Intelligence, 2012, pp. 435–446.

[5] H. P. Luhn, "The automatic creation of literature abstracts," IBM J. Res. Dev., vol. 2, no. 2, pp. 159–165, 1958.

[6] L. Terveen and W. Hill, "Beyond recommender systems: Helping people help each other," HCI New Millenn., vol. 1, no. 2001, pp. 487–509, 2001.

[7] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," Adv. Artif. Intell., vol. 2009, p. 4, 2009.

[8] Y. Qu and Q. Chen, "Collaborative summarization: when collaborative filtering meets document summarization," in Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation, Volume 2, 2009, vol. 2.

[9] P. Hu, D. Ji, C. Teng, and Y. Guo, "Context-enhanced personalized social summarization," Proc. COLING 2012, pp. 1223–1238, 2012.

[10] G. Mani, "Customized News Filtering and Summarization System Based on Personal Interest," Procedia Eng., vol. 38, pp. 2214–2221, 2012.

[11] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: an open architecture for collaborative filtering of netnews," in Proceedings of the 1994 ACM conference on Computer supported cooperative work, 1994, pp. 175–186.

[12] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence, 1998, pp. 43–52.

[13] J. Wang, A. P. De Vries, and M. J. T. Reinders, "Unifying user-based and item-based collaborative filtering approaches by similarity fusion," in Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, 2006, pp. 501–508.

[14] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 5, pp. 603–619, 2002.

[15] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," J. Artif. Intell. Res., vol. 22, pp. 457–479, 2004.

[16] R. Mihalcea and P. Tarau, "Textrank: Bringing order into text," in Proceedings of the 2004 conference on empirical methods in natural language processing, 2004.

[17] Y. Gong and X. Liu, "Generic text summarization using relevance measure and latent semantic analysis," in Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, 2001, pp. 19–25.

[18] J. Steinberger and K. Jezek, "Using latent semantic analysis in text summarization and summary evaluation," Proc. ISIM, vol. 4, pp. 93–100, 2004.

[19] G. Murray, S. Renals, and J. Carletta, "Extractive summarization of meeting recordings.," 2005.

[20] M. G. Ozsoy, I. Cicekli, and F. N. Alpaslan, "Text summarization of turkish texts using latent semantic analysis," in Proceedings of the 23rd international conference on computational linguistics, 2010, pp. 869–876.

[21] S. A. Babar and P. D. Patil, "Improving performance of text summarization," Procedia Comput. Sci., vol. 46, pp. 354–363, 2015.

[22] R. V. V. M. Krishna and C. S. Reddy, "Extractive Text Summarization Using Lexical Association and Graph Based Text Analysis," in Computational Intelligence in Data Mining—Volume 1, Springer, 2016, pp. 261–272.

[23] V. V. M. K. Ravinuthala and S. R. Chinnam, "A Keyword Extraction Approach for Single Document Extractive Summarization Based on Topic Centrality."

[24] M. Yousefi-Azar and L. Hamey, "Text summarization using unsupervised deep learning," Expert Syst. Appl., vol. 68, pp. 93–105, 2017.

[25] R. Nallapati, F. Zhai, and B. Zhou, "SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents.," in AAAI, 2017, pp. 3075–3081.

[26] Y. J. Kumar, F. J. Kang, O. S. Goh, and A. Khan, "Text summarization based on classification using ANFIS," in Asian Conference on Intelligent Information and Database Systems, 2017, pp. 405–417.

[27] H. J. Jain, M. S. Bewoor, and S. H. Patil, "Context Sensitive Text Summarization Using K Means Clustering Algorithm," Int. J. Soft Comput. Eng., vol. 2, no. 2, 2012.

[28] A. El-Kilany and I. Saleh, "Unsupervised document summarization using clusters of dependency graph nodes," in Intelligent Systems Design and Applications (ISDA), 2012 12th International Conference on, 2012, pp. 557–561.

[29] A. Sharaff, H. Shrawgi, P. Arora, and A. Verma, "Document Summarization by Agglomerative nested clustering approach," in Advances in Electronics, Communication and Computer Technology (ICAECCT), 2016 IEEE International Conference on, 2016, pp. 187–191.

[30] L. Suanmali, M. S. Binwahlan, and N. Salim, "Sentence features fusion for text summarization using fuzzy logic," in Hybrid Intelligent Systems, 2009. HIS'09. Ninth International Conference on, 2009, vol. 1, pp. 142–146.

[31] T. Kiss and J. Strunk, "Unsupervised multilingual sentence boundary detection," Comput. Linguist., vol. 32, no. 4, pp. 485–525, 2006.

[32] U. Hahn and I. Mani, “The challenges of automatic summarization,” Computer (Long. Beach. Calif)., vol. 33, no. 11, pp. 29–36, 2000.

[33] J. Kupiec, J. Pedersen, and F. Chen, “A trainable document summarizer,” in Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval, 1995, pp. 68–73.

[34] “Document Understanding Conferences - Past Data.” [Online]. Available: https://www-nlpir.nist.gov/projects/duc/data/2002_data.html. [Accessed: 13-Jul-2018].

[35] C.-Y. Lin, “Rouge: A package for automatic evaluation of summaries,” Text Summ. Branches Out, 2004.

[36] K. Ganesan, “ROUGE 2.0: Updated and Improved Measures for Evaluation of Summarization Tasks,” 2015.

[37] “ROUGE 2.0.” [Online]. Available: https://rxnlp.github.io/ROUGE-2.0/. [Accessed: 13-Jul-2018].

[38] C.-Y. Lin and F. J. Och, “Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics,” in Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, 2004, p. 605.

[39] “scikit-learn: machine learning in Python.” [Online]. Available: http://scikit-learn.org/stable/index.html. [Accessed: 13-Jul-2018].

[40] P. Goyal, L. Behera, and T. M. McGinnity, “A context-based word indexing model for document summarization,” IEEE Trans. Knowl. Data Eng., vol. 25, no. 8, pp. 1693–1705, 2013.

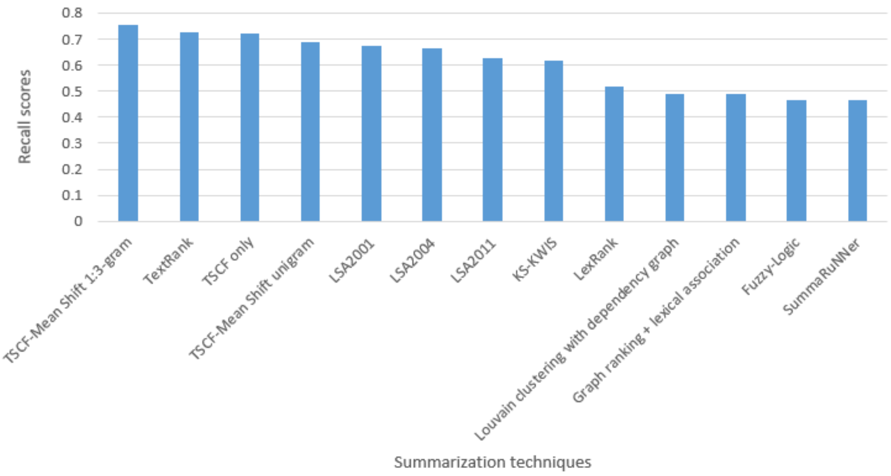

Fig. 1(a). Average ROUGE-1 recall values for our models (TSCF only and TSCF – Mean Shift) with the related techniques.

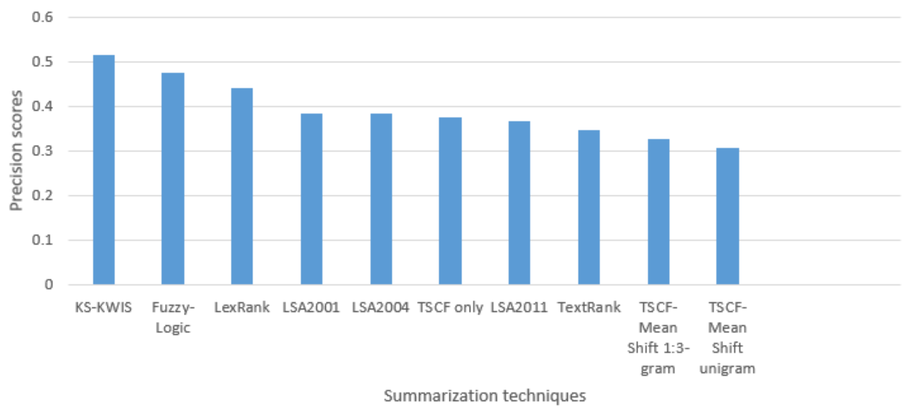

Fig. 1(b). Average ROUGE-1 precision values for our models (TSCF only and TSCF – Mean Shift) with the related techniques.

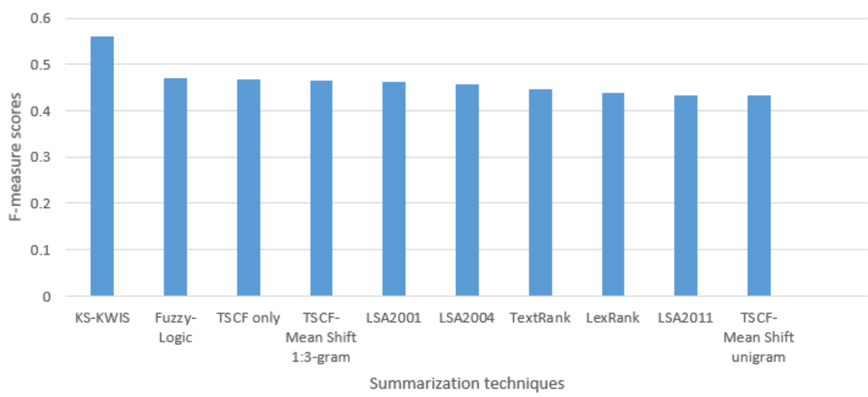

Fig. 1(c). Average ROUGE-1 f-measure values for our models (TSCF only and TSCF – Mean Shift) with the related techniques.

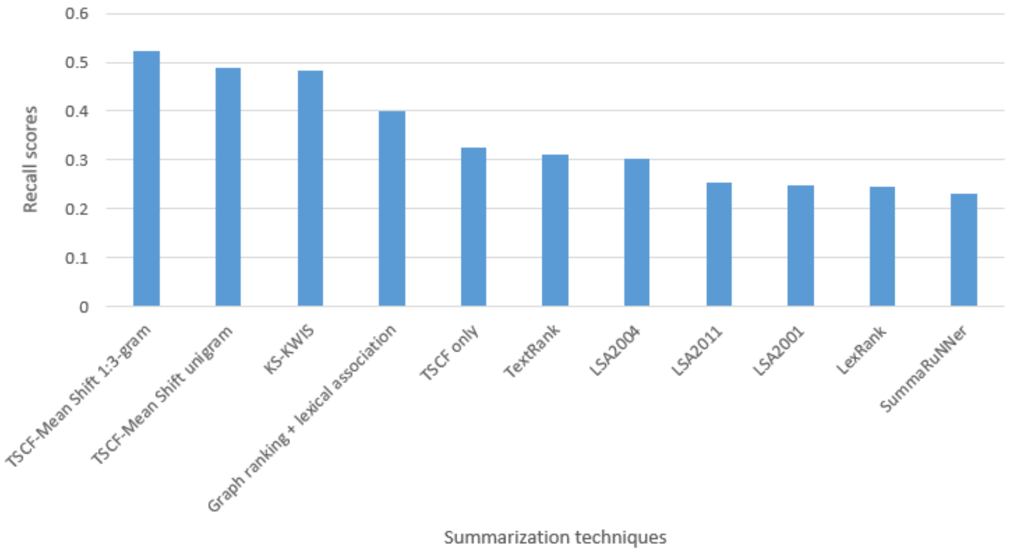

Fig. 2(a). Average ROUGE-2 recall values for our models (TSCF only and TSCF – Mean Shift) with the related techniques.

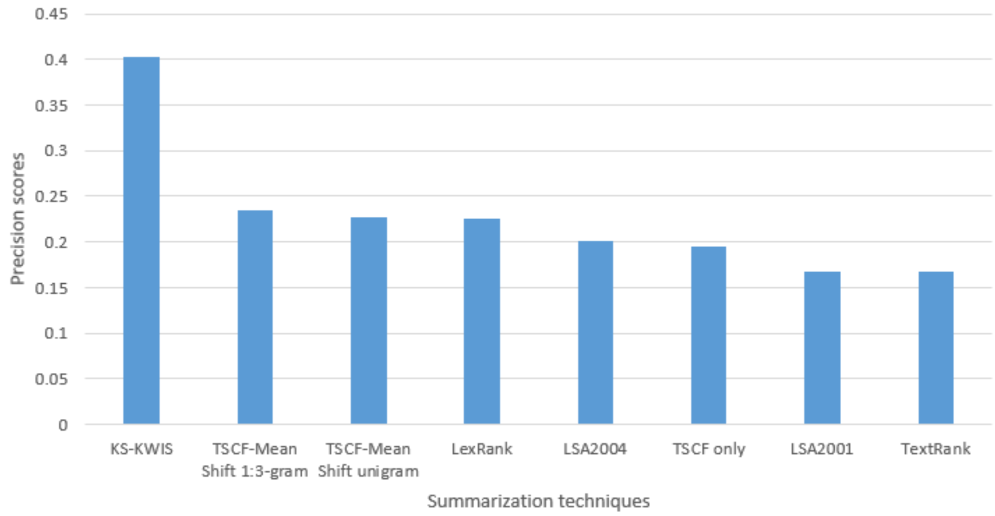

Fig. 2(b). Average ROUGE-2 precision values for our models (TSCF only and TSCF – Mean Shift) with the related techniques.

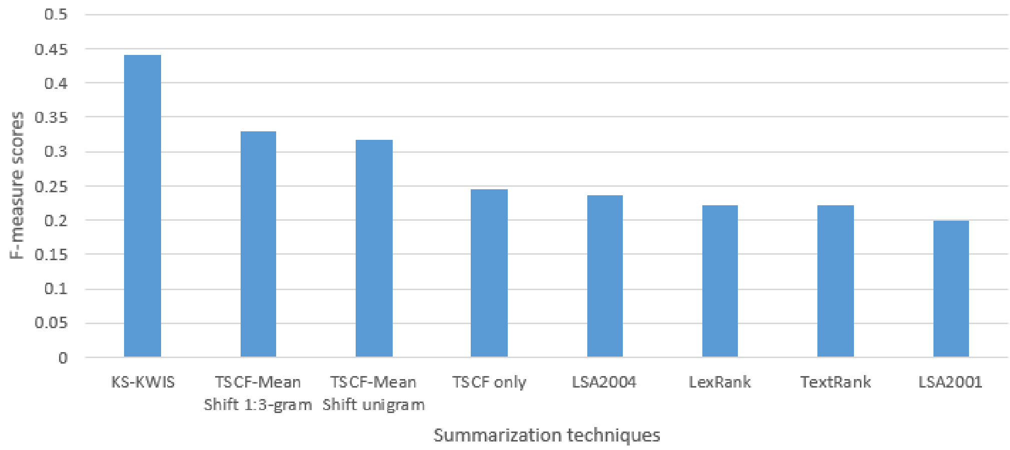

Fig. 2(c). Average ROUGE-2 f-measure values for our models (TSCF only and TSCF – Mean Shift) with the related techniques.

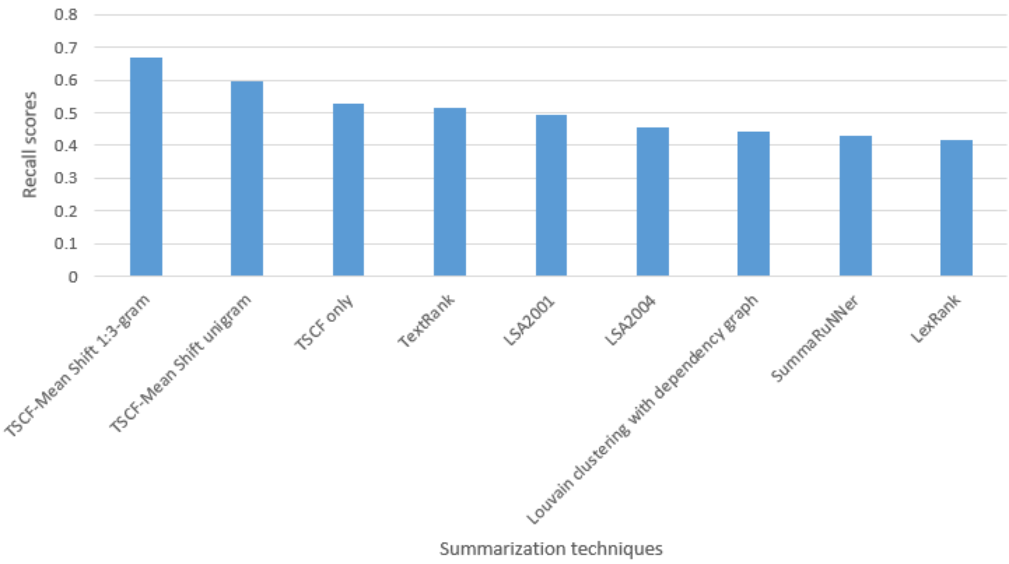

Fig. 3(a). Average ROUGE-L recall values for our models (TSCF only and TSCF – Mean Shift) with the related techniques.

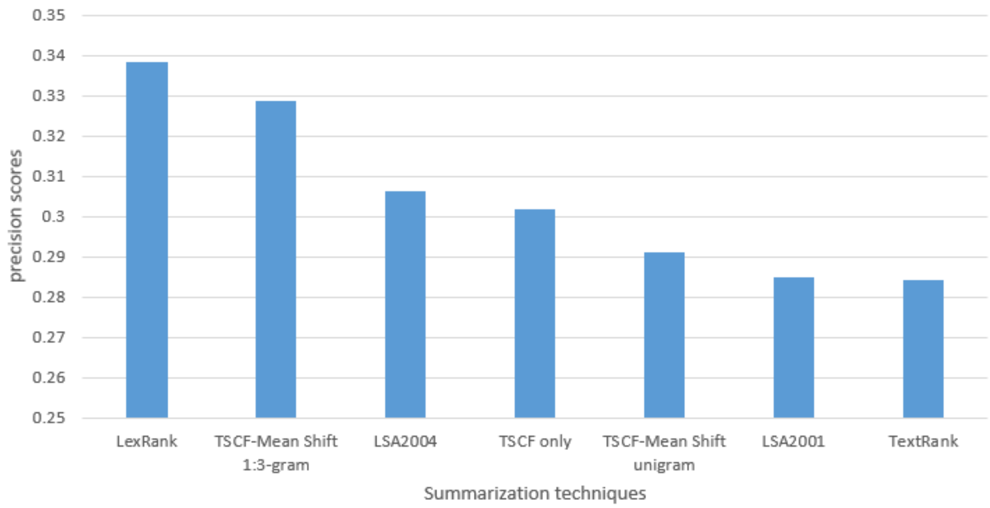

Fig. 3(b). Average ROUGE-L precision values for our models (TSCF only and TSCF – Mean Shift) with the related techniques.

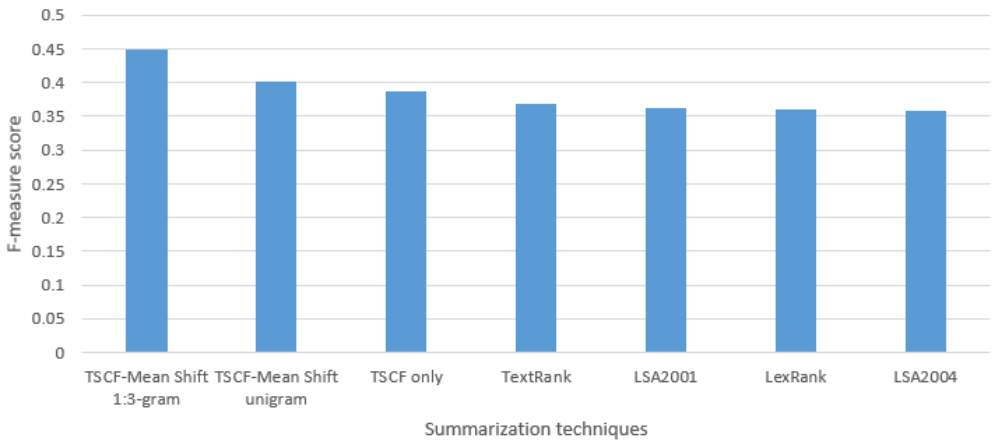

Fig. 3(c). Average ROUGE-L f-measure values for our models (TSCF only and TSCF – Mean Shift) with the related techniques.

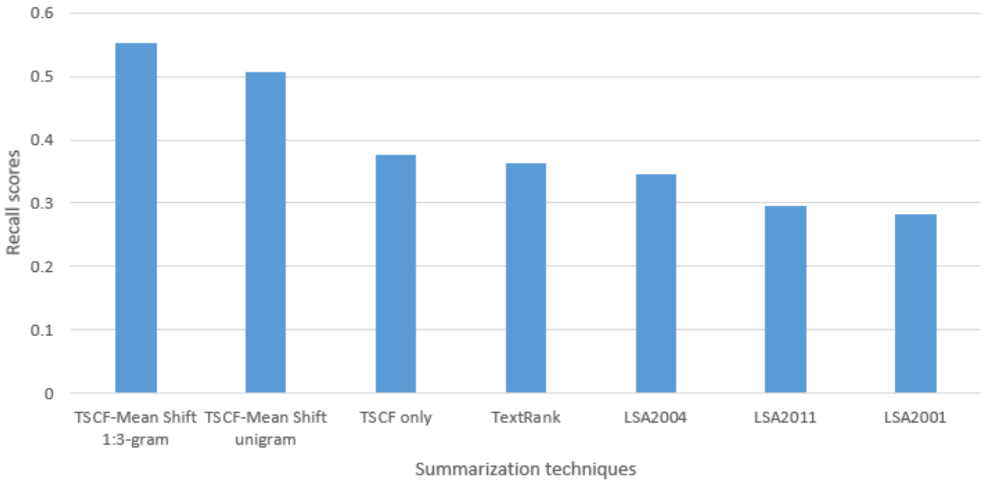

Fig. 4(a). Average ROUGE-SU4 recall values for our models (TSCF only and TSCF – Mean Shift) with the related techniques.

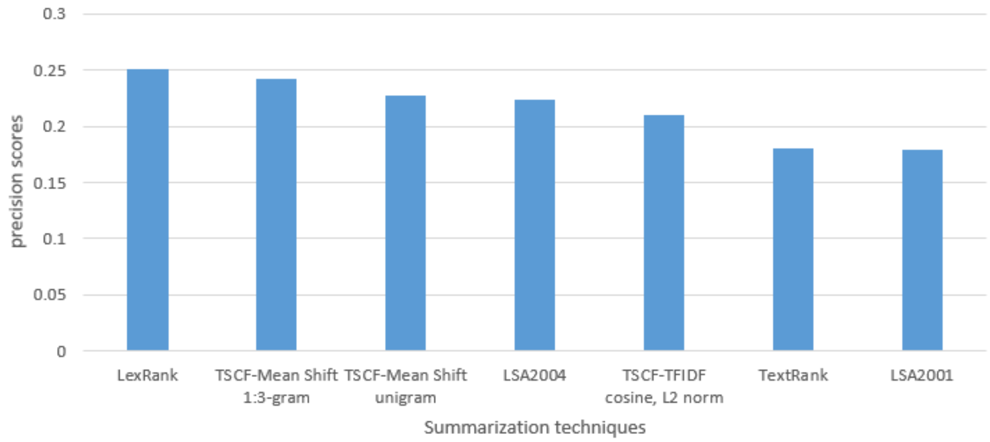

Fig. 4(b). Average ROUGE-SU4 precision values for our models (TSCF only and TSCF – Mean Shift) with the related techniques.

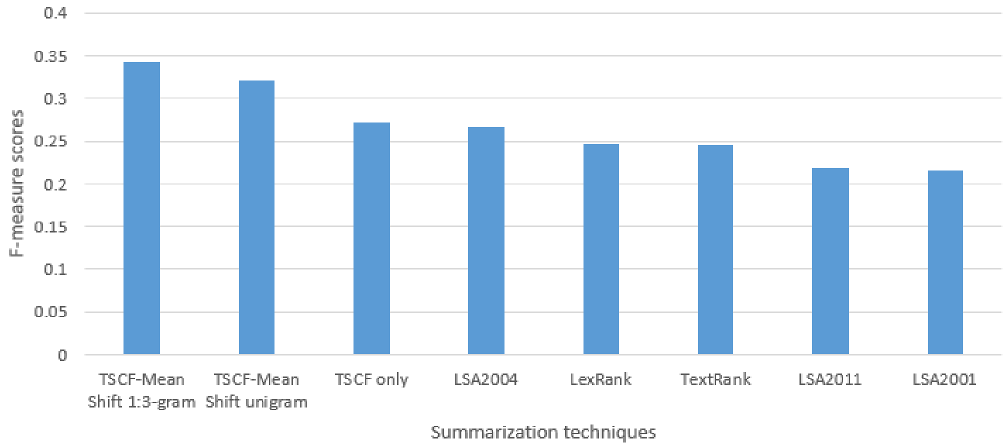

Fig. 4(c). Average ROUGE-SU4 f-measure values for our models (TSCF only and TSCF – Mean Shift) with the related techniques.