Graph-based Automatic Amharic Text Summarizer

Kifle Deresse Dessalegn, Martha Yifru Tachbelie

Abstract -- This paper introduces Graph-based Automatic Amharic Text Summarizer (GAATS), a generic and domain independent graph-based model for automatic single document summarization task, and shows how this model can successfully be used to generate extracts of high quality from Amharic texts. In particular, we extended the two prominent graph-based link analysis algorithms: PageRank and HITS with two sentence centrality measures: cumulative sum and discounted cumulative sum for exploiting the relation between sentences in a text and/or nodes in a graph, and shows the results of our experiments. The results demonstrated that extractive summaries of better quality can be generated when discounted cumulative sum paired with HITS. The results also revealed that our approach is domain independent and more effective than reference summarization systems.

Keywords: Amharic, graph model, centrality measures, natural language processing, position heuristic, artificial intelligence, graph-based ranking algorithms

1 Introduction

Amharic, which is the second most spoken Semitic language in the world after Arabic, has been the working language of the federal government, the military, and various institutes throughout medieval and modern times of Ethiopia [7]. This being the case, the availability of textual information written in this language specifically in news domain increases drastically. This, sheer amount of information, often hinders users(readers) to understand the material and to make sound decision within a short time. This calls for a technology known as Automatic Text Summarization, which is an artificial intelligence complete task and capable of condensing the source text into a shorter version by preserving the most salient contents and overall theme [6].

Text summarization techniques can generally be categorized into two: abstraction and extraction [8]. Abstract is often created by interpreting the information contained in the original source and generating a text that express the same information in a more concise way. Whereas extract is constructed by selecting textual units such as words, phrases, sentences, paragraphs from the original source and organizing them in a way to produce a coherent summary. Although a high-quality abstraction-based summarizer will potentially be more useful, the researches in automatic summarization, including ours, are mainly focused on extraction-based methods because they employ a more straightforward approach for constructing summaries. In line with this, recently few attempts have been made to develop text summarization systems for Amharic.

Most of these works adopt pure statistical methods, which often fail to capture the main themes of the text as they are ignorant of the relations among textual units: words, sentences, and paragraphs in the text. In this paper we introduced a rather new approach to generate extracts of good quality from text written in Amharic. First we modeled the text as an undirected weighted graph. Second we used centrality measures: cumulative sum (MI), and discounted cumulative sum (MII) along with sentence position weight to determine the significance score of sentences in the text or nodes in the graph. Then, to rank sentences based on their relative importance we run the modified PageRank and HITS al-

- *lekifle@gmail.com , School of Electrical Engineering and computing, Computer Science and Engineering Program, Adama Science and Technology University*
- *marthayifiru@yahoo.com, School of Information Science, College of Natural and Computational Sciences, Addis Ababa University*

gorithms on the graph until they converge. Finally, we extracted top N sentences to form a summary. To improve the readability of the system extracts we re-arrange the selected sentences according to their original position in the text.

To the best of our knowledge this is the first work that used the graph model and the link analysis algorithms along with sentence centrality measures for Amharic text summarization. Our approach has several benefits over previous attempts: Firstly, it is not domain dependent. Secondly, it doesn't require neither a corpus nor deep linguistic analysis of the text. Thirdly, it doesn't ignore sentences that reflects the main theme of the text.

The rest of this paper is organized as follows: section 2 presents review of related works on automatic text summarization. In section 3 we describe how we model the text into undirected weighted graph. Section 4 describes the proposed approach in detail. Section 5 deals with experiments, results and evaluations. Finally, we conclude our work with pointers of future direction in section 6.

2 Related works

Since Luhn [11], Automatic Text Summarization has been a hot research area and a lot has been done for texts written in English and good results have been achieved. In this section we review some works done for single document summarization that relied on graph model, and the different works that have been done towards developing Amharic Text Summarization Systems.

TextRank [14] and LexRank [3] proposed graph-based model for computing relative importance of sentences and generating extracts from a single document. In both TextRank and LexRank, first the text was modeled as a graph taking sentences as vertices and the relationship between sentences as an edge. Then to rank sentences based on their relative importance LexRank used PageRank, while PageRank and HITS were employed in TextRank. [Mihalcea and Tarau, 2005] extended their previous work to create an extract from multi-document. What they did was first they created a meta-summary from each document in the cluster then combined those meta-summaries to create one summary.

Patil et al., [17] introduced SumGraph, which is designed to create extracts from newspapers articles. SumGraph relies on two ideas (1) PFnet, which uncover the conceptual organization of the sentences in the text and (2) connectedness. Thus, sentence that is highly connected in the PFnet has the highest relevance. To rank sentences SumGraph uses centrality scores with location heuristic. SumGraph generates a summary by selecting top N sentence till required summary size met.

Previous attempts on developing Amharic Text Summarization systems have used: pure statistical methods [10], [7], and [1]; machine learning [18]; and Latent Semantic Analysis (LSA)[2][13].

Kamil [10] used surface level approach that relies on features: cue-phrase, title words, header (first sentence of the document), words in the header, first sentence in the paragraph, and highly frequent words, which are statistical in nature, to generate summaries. The author didn't use a stemmer, which affects the frequency of words in a text. However, exhaustive list of common words was used to improve performance of his summarization system.

Helen [7] attempted to develop text summarization system that generates extracts from single legal judgment. Her work uses three-step process. First the text/legal judgment segmented in to five pre-defined themes: introduction, reason, fact, judicial analysis, and decision. Then statistical features: cue-phrases, and sentence position were used to determine the importance of sentences. Finally, meta-summaries from each theme with compression rate of 20% have been creates and then combined to form a single summary.

The work of Addis [1] was customizing an open source tool, Open Text Summarizer (OTS[3]), to generate extract from Amharic news texts and to test its performance. His work uses frequency of terms to determine the relative importance of a sentence in a text. As his work emphasize testing the performance of the tool, he used two experimental setups: E1, where the original Porter stemmer employed, and E2 that uses stemmer adopted from the work of [20]. After a series of experiment on each setup the results revealed that E2 outperformed E1 and performance of OTS in summarizing Amharic text was promising.

Teferi [19] investigated the applicability of Naïve Bayes classifier to generate extracts from Amharic news articles. This work used two step process: training and testing. To determine the probability of a given sentence to be included in a summary, he used features: title words, cue-words, sentence

[2] *It is an algebraic statistical method that extracts meaning of words and similarity of sentences using the information about the usage of the words in the context.*

[3] It is an open source language independent tool which uses statistical features or methods for summarizing text/s.

position, and presence of thematic words. Even though, his approach is domain dependent and requires a great deal of training corpus the finding showed that it was promising.

Melese [13] proposed, Latent Semantic Analysis (LSA) for automatic Amharic text summarization. His work integrates two techniques: TopicLSA and LSAGraph. Each text represented as a graph taking terms as nodes and semantic relationship between terms as edges. To compute the significance score of sentences the graph based algorithms were run on to the graph iteratively until they converge. What's more sentences that do not have correlation with the topic were penalized so as not to be included in a summary.

The works of [10], [7], and [1], which are based on pure-statistical methods, fail to capture the main themes of a document as their approaches are ignorant of the relations among textual units in the document. As a result, salient sentences that reflect the core concept of the document may not be included in the summary. [18]'s work is domain dependent and not easily adaptable to any language; and require a great deal of training corpora that makes it very costly in terms of resources. And [13]'s approach became inefficient when the text to be summarized gets larger, and it ignores sentences that do not win latent space or dimension even if they are suitable for a summary. Thus, this work aims to address the gaps of previous attempts on developing automatic Amharic text summarization systems.

3 Modeling Text as Graph

Graph is capable of representing different phenomena where relations between objects are important, and one of such phenomena is text summarization [18]. In text summarization a given document D is represented as graph G = (V, E), where the graph G= (V, E) is undirected weighted graph that represents document D with set of vertices V and edge between vertices E. In this work, to model text as graph we used sentences as vertices and the degree of similarity between two vertices V_i and V_j as an edge.

4 The Proposed Approach

We developed a generic and domain independent extractive text summarization system for Amharic texts called GAATS. Figure 1 shows the overview of our summarization model, which consists of four modules: preprocessing, graph construction, sentence ranking, and sentence extraction and re-ordering. The input to the model is a text written in Amharic. Firstly, the input text is pre-processed. Then undirected weighted graph is constructed with sentences as vertices and relationship or similarity between two vertices as edges. Thereafter, the modified link analysis algorithm run on the graph iteratively until it convergence to get salient score of each sentences. The sentences are raked based on their degree of centrality. Top N sentences are selected to form a summary. Finally, selected sentences are re-arranged according to their original sequence in the input text to make the extract as coherent as possible. Description of each module is given in the following sections.

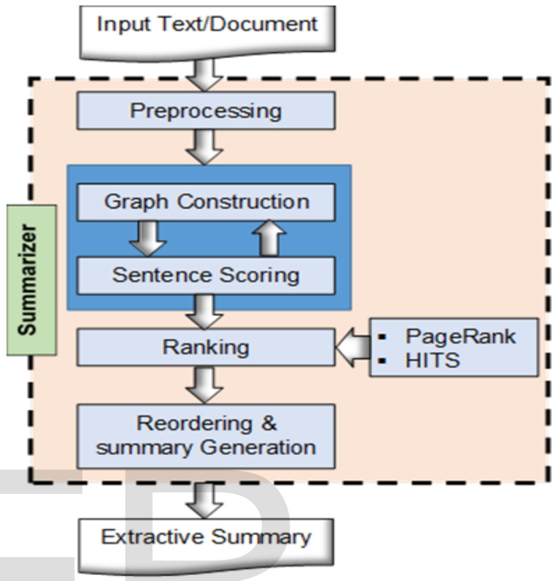

Fig 1: architecture of the proposed model

4.1 Pre-Processing

preprocessing is the most important and language dependent task that prepare the input text into a format that is suitable to the text summarization process. This module incorporates five interdependent sub tasks: Firstly, we eliminate short sentences, which contain no information. Then, we slice text into smallest linguistically significant and methodologically useful units and clean non-language specific characters. Secondly, we have filtered and removed words that do not contain any particular information from the text. Thirdly, since Amharic is a morphologically rich language, a number of words can be inflected from the given word by adding affixes, which makes the task of stemming inevitable. For example, the following inflectional forms ቤቱ/bet-u/'his-house',ቤቷ/betwä/'her-house',ቤቴ/bet-e/'myhouse',ቤታቸው/betäčew/'their-house',ቤታችን/betäčn/'our-house',ቤታችሁ/betäčhu/'your-house',ቤቶቻቸው/betočäčew/'their-houses',ቤቶቻችሁ/betočäčhu/'your-houses',ቤቶቻችን/betočäčn/'our-houses',ቤቶች/betoč/ 'houses' etc., all can be reduced to the root form ቤት/bet/ 'house' [14]. Thus in this work we use affix removal technique of stemming to reduce inflectional words to their root. Fourthly, we have conflated characters that have same sound but different forms in writing to a single representative form. For instance, in Amharic one may write the word "prayer" as ጸሎት/'tselot' or ፀሎት/'tselot'. However, such kind of variations negatively affects the summarization algorithm as it weakens the strength of relation between sentences. Thus the task of normalization is unavoidable. Finally, we have extracted tokens, which are capable of representing the content of the text that meet the thresholds. In this work, to extract the tokens, we have empirically set a threshold values of 2 and 8 for lower and upper cut-off points respectively.

4.2 Graph Construction

This module transforms the preprocessed input text into undirected weighted graph G= (V, E) using sentence connectivity matrix, where each row and columns in the matrix correspond to a particular sentence that represent node in the graph and each cell values represent similarity between the corresponding sentence pair that establish an edge between nodes in the graph. There are different methods: Jaccard, Dice, Word-Overlap, and Cosine, that can be applied in Amharic text to compute similarity between two sentences. In this work, we used cosine similarity as:

$$sim(s_i, s_j) = \frac{\sum_{w \in s1,s2} tf_{w,i} * tf_{w,j} (isf_w)^2}{\sqrt{\sum_{i_k \in i} (tf_{i_k,i} * isf_{i_k})^2} \sqrt{\sum_{j_k \in j} (tf_{j_k,j} * isf_{j_k})^2}} \qquad (1)$$

Where $tf_{w,i}$, and $tf_{w,j}$ are the frequency of the word w in the sentence s_i and s_j respectively, and isf is the inverse sentence frequency. Table 1 shows the similarity matrix for sample Amharic news article used in the dataset.

Table 1: intra-sentence similarity for sample Amharic news article.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	1.000	0.103	0.000	0.175	0.085	0.101	0..065	0.130	0.109	0.071
S2		1.000	0.051	0.084	0.122	0.145	0.000	0.093	0.079	0.103
S3			1.000	0.000	0.085	0.101	0.065	0.000	0.000	0.000
S4				1.000	0.069	0.082	0.106	0.106	0.134	0.117
S5					1.000	0.120	0.077	0.230	0.065	0.085
S6						1.000	0.000	0.092	0.077	0.101
S7							1.000	0.059	0.000	0.000
S8								1.000	0.099	0.065
S9									1.000	0.218
S10										1.000

4.2.1 Sentence Scoring

To compute the score of each node on the graph G or sentence in text we use two centrality measures: MI and MII. MI computes sentence's centrality using the mean of link weights of the sentence with others considering links whose weights is greater than or equal to specified threshold. MII follows the same principle as MI to compute the centrality of a sentence. Thereafter, set the corresponding row and column values of the matrix related to that sentence to zero, and compute the centrality of the next sentence based on the contributions made by the remaining 'n-1' sentences… etc., this process iterates until the centrality scores of all sentences are obtained. Thus, centrality score of each node on the graph G is computed as:

$$\sigma_i = \left(\frac{1}{N-1} \sum_{i=1}^{k} w_{ij}\right) \qquad (2)$$

Where σ_i denotes the centrality of sentence i, w_{ij} is the cosine similarity of sentences s_i and s_j, and N is the number of sentences in the text or nodes in the graph.

Furthermore, since position of a sentence plays a significant role in determining its importance in a text we used it as an added feature. There are three methods: linear, hyperbolic, and quadratic for computing sentence position weight. In this paper we used the third method, which gives better estimation of sentence position weight as discussed in [2]:

$$\mathrm{P_s} = \begin{cases} 1 - \left\{\frac{2}{\mathrm{n}-1} * (\mathrm{i}-1)\right\}, \mathrm{i} < \frac{\mathrm{n}+1}{2} \\ 1 - \left\{\frac{2}{\mathrm{n}-1} * (\mathrm{n}-\mathrm{i})\right\}, \mathrm{i} > \frac{\mathrm{n}+1}{2} \\ 0.1, \mathrm{i} = \frac{\mathrm{n}+1}{2} \end{cases} \qquad (3)$$

Where Pₛ is the position weight assigned to the sentence s, i is the sentence location in the text, and n is the total number of sentences in the text.

To determine the significance score of a node on the graph G or sentence in the text, we linearly combined the results of equation 2 and 3 as:

$$S_i = \sigma_w(v) + P_s \quad (4)$$

Where S_i denotes significance score of a sentence, $\sigma_w(v)$ is the centrality score of a sentence and P_s is the position score of the sentence in a text.

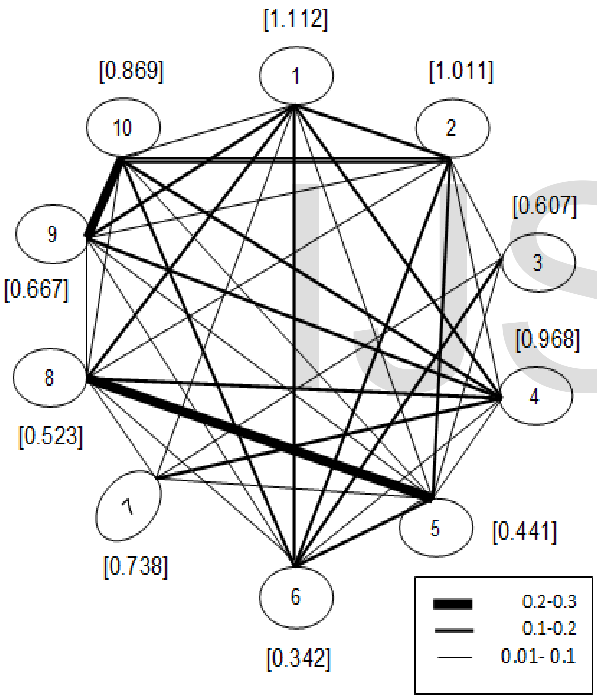

Fig 2: Sample graph build for text representation.

4.3 Ranking Sentences

Once graph G= (V, E), is built, we computed salient score for each node on G using PageRank and/or HITS algorithm, and then rank them according to their degree of importance. PageRank and HITS are the most popular link analysis algorithms which are useful to determine the importance of a node within a graph, based on information drawn from the graph structure [3] [11]. The later one involves computation of two scores for each node in the graph: hub and authority score. The hub score is a measure of the outgoing links of a node whereas the authority score measures the scores of incoming links of a node. These algorithms originally assumed unweighted directed graphs. However, the graph that has been constructed in the previous module was undirected weighted graph, and using the original PageRank and HITS algorithm on this type of graph results in identical In-degree and Out-degree scores for a node in PageRank; and identical hub and authority scores for a node in HITS.

Thus, in this work, we adapt these two iterative link analysis algorithms using sentence centrality measures as discussed in the work of [6]:

$$\mathrm{PR^w}(\mathrm{v_i}) = \frac{(1-\mathrm{d})}{\mathrm{N}} + \mathrm{d} * \sum_{\mathrm{v_j} \in \mathrm{In}(\mathrm{v_i})} \mathrm{w_{ij}} \frac{\mathrm{PR^w}(\mathrm{v_j}) * \mathrm{c_m}}{\sum_{\mathrm{v_k} \in \mathrm{Out}(\mathrm{v_j})} \mathrm{w_{kj}}} \quad (5)$$

$$\mathrm{HITS_A}(\mathrm{v_i}) = \sum_{\mathrm{v_j} \in \mathrm{In}(\mathrm{v_i})} \frac{\mathrm{w_{ij}} * \mathrm{c_m}}{\sum_{\mathrm{v_k} \in \mathrm{Out}(\mathrm{v_j})} \mathrm{w_{kj}}} \mathrm{HITS_H}(\mathrm{v_j}) \quad (6a)$$

$$\mathrm{HITS_H}(\mathrm{v_i}) = \sum_{\mathrm{v_j} \in \mathrm{Out}(\mathrm{v_i})} \frac{\mathrm{w_{jk}} * \mathrm{c_m}}{\sum_{\mathrm{v_k} \in \mathrm{In}(\mathrm{v_j})} \mathrm{w_{ij}}} \mathrm{HITS_A}(\mathrm{v_j}) \quad (6b)$$

Where $PR^w(v_i)$ is weight of a node, which represents salient score of a sentence in the text, $HITS_A(v_i)$ is the authority score of node V_i, $HITS_H(v_i)$ is the hub score of a node v_i, $In(v_i)$ is set of nodes that points to vi, $Out(v_j)$ is set of node to which node v_j points, w_{ij} is the weight of the edge directing from node v_i to node v_j, C_m is the centrality score of a sentence, d is a damping factor which is typically chosen in the Interval [0.1, 1] [Brin and Page, 1998]. For this work, d is set to be 0.15 empirically.

4.4 Extraction and re-ordering

After the ranking algorithms converged, and the sentences are sorted according to their salient scores. The next task is selecting top ranked N-sentences to generate a summary. GAATS extracts the most salient sentences from the original document sequentially until the desired length of the summary is reached. Then these sentences are re-arranged according to their original order in the source text to make the summary as sensible and coherent as possible for readers.

5 Experiment, Evaluations, Results, and Discussion

In this section, we present: the data sets that have been used, experiments that have been conducted, the results obtained and the evaluations to test the performance of our approach, and discussion of the findings.

5.1 Dataset

The experiments conducted on 30 news articles of which 8 are on economic, 4 are on politics, 14 are on society, and 4 are on sports. These articles are collected from the Amharic version of Addis Admas [4]and Ethiopian Reporter [5]websites. We used a threshold of 20 (article that has more than 20 sentences) to select the article as a dataset.

5.2 Experiments

For each article in the data sets, 12 experiments have been conducted by pairing the link analysis algorithms with sentence centrality measures discussed earlier as E1, E2, E3, and E4 to generate system summaries with compression rates of 10%, 20%, and 30%. E1 and E3 paired PageRank with MI and MII respectively, whereas E2 and E4 paired MI and MII with HITS respectively. All the experiments considered a sentence cutoff point of 10, i.e., sentences that contain less than 10 words were excluded from being part of summary, and that was decided empirically on the subset of our dataset. Moreover, we have conducted series of experiments based on articles' domain with those three compression rates.

5.3 Evaluations

Evaluation is one of the notoriously challenging task in natural language processing researches, specifically in text summarization as there is no golden standard. In this study, to evaluate the performance of GAATS, three reference summaries for each news article on the dataset with compression rates of 10%, 20%, and 30% have been prepared by experts. Then, for each experiments, the summaries generated by GAATS were compared against the human crafted summaries for their linguistic quality and content overlap. To measure the linguistic quality and content overlap of the summaries we used co-selection measure, which is intrinsic by its nature [9] [16], that uses common information retrieval metrics such as precision, recall, and f-measure that are depicted by equations 7, 8, and 9 as follows:

$$P = \frac{|\text{System and Human choice overlap}|}{|\text{sentences choosen by the System}|} \qquad (7)$$

$$R = \frac{|\text{System and Human choice overlap}|}{|\text{sentences choosen by Human}|} \qquad (8)$$

$$F = \frac{2*P*R}{P+R} \qquad (9)$$

As can be seen from the above equations precision (P) and recall (R) are antagonistic to one other as a system that strives for coverage will get lower precision and a system that strives for precision will get lower recall. Thus, in this study we used an f-measure (F) that tradeoffs between P and R to measure linguistic quality and content overlap of summaries.

5.4 Results and Discussion

As discussed earlier the performance is expressed in terms of f-measure. Table 2, Table 3, Table 4, Table 5, and Table 6 show the results obtained from our experiments and evaluations.

Table 2: Results obtained from content overlap evaluation

Experiments	Average F-Score		
	10%	20%	30%
E1 (PageRank + MI)	0.5556	0.6433	0.7131
E2 (HITS + MI)	0.5609	0.6604	0.7144
E3 (PageRank + MII)	0.5836	0.6968	0.7570
E4 (HITS + MII)	0.6590	0.7708	0.8323

Table 3: Results obtained from linguistic quality evaluation

Experiments	Average F-Score		
	10%	20%	30%
E1 (PageRank + MI)	0.4543	0.4900	0.5708
E3 (PageRank + MII)	0.4731	0.5273	0.5865
E2 (HITS + MI)	0.5168	0.5812	0.6556
E4 (HITS + MII)	0.5735	0.6937	0.7522

Table 2 and Table 3 show the average f-score for linguistic quality and content overlap of summaries generated by GAATS with compression rate of 10%, 20%, and 30%. As illustrated by those tables, the quality as well as informativeness of the system generated summaries directly correlated with the compression rate, this is expected as the larger the compression rates the more the number of sentences to be included in the summary and the higher the probability of a sentence generated by GAATS to match with a sentence in the reference

[4] http://addisadmasnews.com

[5] http://ethiopianreporter.com

summary. Moreover, the results depicted on these tables show that best results were obtained when GAATS paired MII with both ranking algorithms to generate summaries for all the three compression rates.

Table 4: Results obtained from content overlap evaluation based on domain

Domains	Average F-Measures		
	10%	20%	30%
Economic	0.6409	0.6596	**0.7482**
Politics	0.5711	0.684	0.7096
Society	0.5538	**0.6847**	0.7343
Sport	**0.6703**	0.6452	0.7352

Table 5: Results obtained from linguistic quality evaluation based on domain

Domains	Average F-Measures		
	10%	20%	30%
Economic	**0.5442**	0.5548	0.6234
Politics	0.4969	0.5356	0.6327
Society	0.5189	**0.5681**	**0.6558**
Sport	0.4989	0.5554	0.6501

Table 4 and Table 5 show the average f-measure for linguistic quality and informativeness of summaries generated by GAATS with compression rate of 10%, 20%, and 30% based on domains of the datasets. Like results in Table 2 and Table 3, F-score is directly correlated with compression rates for all domains. Best content overlap results are obtained for summaries generated from sport, society, and economic domains for compression rates of 10%, 20%, and 30% respectively. As for linguistic quality, summaries generated from economic domain registered best results for extraction rate of 10% while summaries from society domain give better results for compression rates of 20% and 30%.

Table 6: Comparison between our approach and other summarization system for Amharic

Experiments	Average F-Score	
	20%	30%
Baseline	0.4915	0.5839
LSA-Graph	0.3650	0.4600
GAATS	**0.6329**	**0.6977**

Table 6 reports the results obtained by three text summarization systems: Baseline, LSA-Graph, and GAATS. The baseline is a system that uses the classical link analysis algorithms and doesn't consider features like sentence centrality, position heuristic, tuning etc., whereas LSA-Graph is a system that incorporates LSA-based text summarization approaches and graph-based ranking algorithms [13].

Like the previous tables, F-score is directly correlate with the compression rate and f-score for all systems are significantly higher with compression rate of 30% for the obvious reason; and our system outperformed the baseline as well as the LSA-Graph. We can see that it registered 14.14% and 11.38% improvements over the baseline, and 26.79% and 23.77% improvement over the LSA-Graph for compression rates of 20% and 30% respectively. Here, we would like to state that the results obtained comparing GAATS with LSA-Graph are inconclusive. This is because: the datasets used in this study and LSA-Graph are different, and due to the absence of benchmark for Amharic like that of DUC.

6 Conclusion and Future work

In this paper, we introduced GAATS, a generic and robust graph-based automatic single document summarization system for Amharic. It is designed by extending the legacy link analysis algorithms: PageRank and HITS with two sentence centrality measures: cumulative sum, and discounted cumulative sum. An interesting finding of our experiments was that pairing the later centrality measure with both graph-based ranking algorithms gives significantly better results for all compression rates than its counterpart, cumulative sum, this is mainly because computing sentence centrality using discounted cumulative sum measure minimizes, if not totally avoided, the chance of information repetition in a summary. The result of our experiments also showed that our system outperformed other graph-based text summarization systems designed for Amharic. An important aspect of GAATS is that it's unsupervised approach and doesn't require deep linguistic analysis of the text, and doesn't ignore sentences that reflects the main theme of the text. One of the challenges we faced during this study was the prevalence of dangling reference in summaries generated by our system. Therefore, finding a method to resolve this situation is one of the main points we target in our future work. And we also would like to test our system using standardized corpora like that of DUC.

7 Acknowledgment

The authors would like to thank the anonymous reviewers for their unreserved and invaluable comments, and those who helped us during this study.

8 References

[1] A. Addis, “Automatic Summarization for Amharic Text Using Open Text Summarizer”. Thesis, Department of Information Science, Addis Ababa University, 2013.

[2] P. Baxendale, “Machine-Made Index for Technical Literature: An Experiment,” IBM Journal of Research and Development, pp. 354-361, 1958.

[3] S. Brin & L. Page,” The Anatomy of a Large-Scale Hyper Textual Web Search Engine,” Computer Networks and ISDN Systems, pp. 107-117, 1998.

[4] G. Erkan & D. Radev, “LexRank: Graph-based Lexical Centrality as Salience in Text Summarization,” Artificial Intelligence Research, 457-479, 2004.

[5] M. Gasser, “HornMorpho: A System for Morphological Processing of Amharic, Oromo, and Tigrinya,” Conference on Human Language Technology for Development (CHLTD), Alexandria, Egypt, 2011.

[6] S. HariHaran, & R. Srinivasan, “Studies on Graph Based Approaches for Single and Multi-Document Summarizations,” International Journal of Computer Theory and Engineering, Vol.1, Issue No, 2009.

[7] A. Helen, “Text Summarization on Amharic Legal Judgements”. Thesis, Department of Information Science, Addis Ababa University, 2006.

[8] E. Hovy, C. Lin, L. Zhou, & J. Fukumoto, “Automated Summarization Evaluation with Basic Elements,” In Preceding of the 5th Conference on Language Resources and Evaluation (LREC), pp. 899-902, 2006.

[9] K. Ježek, & J Steinberger, “Automatic Text Summarization: State of the art and challenges.,” Znalosti, pp. 1-12, 2008.

[10] N. Kamil, “Automatic Amharic News Text Summarizer”, Thesis, Department of Information Science, Addis Ababa University, 2004.

[11] J. Kleinberg, “Authoritative Sources in a Hyperlinked Environment,” Journal of the ACM (JACM), 604-632, 1999.

[12] P. Lhun, “A Statistical Approach to Mechanized Encoding and Searching of Literary Information,” IBM Journal, 159-168,1958.

[13] T. Melese, “Automatic Amharic Text Summarization using Latent Semantic Analysis,” Thesis, Department of Computer Science, Addis Ababa University,2009.

[14] S. Meron, “Concept-based Automatic Amharic Text Categorization,” Thesis, Department of Information Science, Addis Ababa University,2009.

[15] R. Mihalcea & P. Tarau, “TextRank: Bringing order into texts”, Empirical Methods in Natural Language Processing (EMNLP), pp.404-411, Barcelona, Spain, 2004.

[16] A. Nenkova, “Evaluating Content Selection in Summarization: The Pyramid Method,” In DUC, Vancouver, Canada, 2006.

[17] K. Patil & P. Brazdil, “SumGraph: Text Summarization using Centrality in the Pathfinder Network,” International Journal on computer Science and Information Systems, pp.18-32, 2007

[18] G. Saeedeh, AS. Mohsen, & G. Bahareh, “A Comprehensive Survey on Text Summarization Systems,” CSA 2:462-468, 2009.

[19] A. Teferi, “The Application of Machine Learning Technique for Automatic Text Summarization: The Case of Amharic News Texts,” Thesis, Department of Information Science, Addis Ababa University, 2005.

[20] M. Tesema, “Design and Implementation of Amharic Search Engine”. Thesis, Department of Information Science, Addis Ababa University, 2007.