Unsupervised Document Summarization Using Clusters of Dependency Graph Nodes

Ayman El-Kilany
Faculty of Computers and Information
Cairo University
Cairo, Egypt
a.elkilany@fci-cu.edu.eg

Iman Saleh
Faculty of Computers and Information
Cairo University
Cairo, Egypt
iman.saleh@fci-cu.edu.eg

***Abstract*—In this paper, we investigate the problem of extractive single document summarization. We propose an unsupervised summarization method that is based on extracting and scoring keywords in a document and using them to find the sentences that best represent its content. Keywords are extracted and scored using clustering and dependency graphs of sentences. We test our method using different corpora including news, events and email corpora. We evaluate our method in the context of news summarization and email summarization tasks and compare the results with previously published ones.**

***Keywords*-Extractive summarization; Dependency graph; Louvain clustering; Email summarization; ROUGE;**

I. Introduction

There are two types of summaries: abstractive and extractive. Creating an abstractive summary for a document requires understanding its overall meaning then writing it in a condensed form. Extractive summaries are built by selecting important sentences or paragraphs in a document. The importance of the selected document units is determined based on some of their features. These features include the frequency of words or phrases, keywords and phrases that can determine which sentence should be extracted, as well as the position of a sentence inside text.

In this paper we propose a novel method for extractive single document summarization. Our method uses dependency graphs of sentences in addition to Louvain clustering algorithm [1] to extract keywords in a document. Keywords are used to identify important sentences. The importance of a keyword stems from the number of its dependants. This method was used in the context of sentence compression [2] and we make use of the same idea to generate extractive summaries. We used three corpora to test the performance of our method: the conversation corpus of British Colombia (BC3) [3], Document Understanding Conference corpus (DUC)[1], and the Concisus corpus of event summaries [4].

[1]http://www-nlpir.nist.gov/projects/duc/data.html

II. Proposed Approach

We start with the dependency graph of the sentence and use dependency relations to group coherent words in clusters. We assume that each document contains a set of keywords. A keyword in our approach is defined as a noun which most other words in the dependency graph depend on. This strong dependency relation is determined using a score that is assigned to all words in a cluster. The score is based on the position and the importance of a word with respect to the dependency graph of its sentence. Top scored words, keywords, should be included in the summary of the document because we assume they represent important concepts inside it. Keywords are first extracted at the level of a single sentence. Since we would like to extract keywords at the level of the whole document and not only at the level of a sentence, we increase the score of a keyword if it appears in the context of another one. Our system is illustrated in figure 1. In the next subsections we explain the four steps used to generate a summary for a document.

A. Generating Words Clusters

The goal of this step is to turn sentences in a document into clusters of words. Each cluster represents a group of coherent words. Following the footsteps of [2], the dependency graph of each sentence is generated. We use Stanford parser [5] to produce basic typed dependency graph of sentences. The Louvain clustering algorithm [1] is used over the dependency graph to generate clusters of words. This algorithm is able to find related words inside a graph of nodes. It was chosen over other clustering algorithms due to the nature of the dependency graph, where there is no distance measure between nodes and only the links between them define a relationship. Dependency relationships between words are used to decide whether they should be assigned to the same cluster or not. The Louvain algorithm creates hierarchical clustering over the dependency graph. For instance, a four level hierarchy means that 4 different levels of clusters are generated. Each group consists of a number of clusters. The first level hierarchy contains more clusters than the highest level hierarchy which contains the fewest number of clusters.

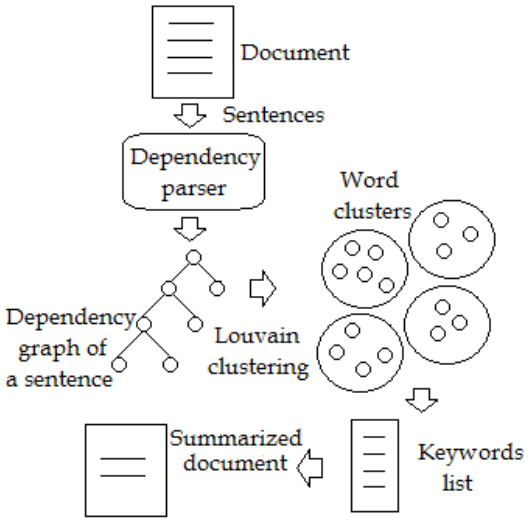

Figure 1. Illustration of our summarization system

For example, the first level contains 8 clusters while the highest level contains only 4 clusters. We choose the highest level hierarchy to obtain clusters of words. The generated clusters represent local units inside each sentence in the document.

B. Finding Keywords List

In this step, we extract keywords from clusters generated in the previous step. A keyword is the word assigned the highest score within each cluster. This score is calculated using equation 1. The dependency graph root is assigned a score of 1. The score of each word is a percentage of the score of its parent. The more dependants a word has inside the dependency graph, the more important it is. Therefore, the score of the word is based on the number of its children in the dependency graph. This score is normalized by the number of children of its siblings to decrease its importance if another sibling has more children than the word itself. The score also decreases as we descend down the dependency graph because the leaf node has no dependants unlike the root: all words in the sentence depend on it.

$$S(W) = \frac{CC(W)}{1 + CC(W) + \sum_{i=1}^{n(W)} CC(B_i)} * S(p(W)) \quad (1)$$

Where S(W) is the score of the word W, CC(W) is the number of children of the word W in the dependency graph, p(W) is the parent of the word W in the dependency graph, n(W) is the total number of siblings of word W, B_i is the sibling i of W in the dependency graph.

We limit extracted keywords to nouns as they are the best words representing the content of a document. Keywords are normalized using Porter stemmer to ensure consistency in words forms. The keywords list mainly transforms the local units of each sentence in the document into one global representation that gathers the most important words in all of the sentences inside a document.

C. Enhancing keyword score

In this step, the score of a keyword KW1 is increased by one if there is a direct edge between KW1 and any other keyword KW2 in the dependency graph of the sentence containing KW2. We perform this step to find if there is a keyword that was mentioned in the context of another keyword. Enhancing the scores of related keywords will help the algorithm select a set of coherent sentences that contain those keywords. Next, duplicate keywords are removed after their scores and counts inside the keywords list are summed and added to the score of the remaining occurrence of each keyword. Finally, scores of keywords are enhanced by adding the total frequency of each keyword in the document to its score. Score of each unique keyword is summarized in equation 2.

$$EnhancedS(UnqiueW) = \sum_{i=1}^{N} S(W_i) + C + N + M \quad (2)$$

Where EnhancedS is the enhanced word score, S is the word score within its dependency graph, W is the keyword, N is the number of occurrences of the keyword W in the keyword list, M is the number of occurrences of W inside the document, and C is the number of keywords that W appeared in their context.

D. Generating document summary

In this step, each sentence in the document is scored using the sum of keywords scores the sentence contains. Then, sentences are sorted according to their scores and the top n sentences are considered the summary of the document. Top n sentences are expected to be coherent because they contain most of the keywords assigned the highest scores.

III. Datasets

We use three different datasets in our experiments. The first one is the British Colombia Conversation corpus (BC3). The corpus is published by the University of British Colombia [3]. It consists of 40 email threads, and each thread has an average of 6 email messages. An email message contains 12 sentences on average. Three different extractive summaries are provided for each thread. The summary is created by selecting a number of sentences out of the actual sentences in a thread. The second dataset is the Document Understanding Conference (DUC) dataset for the years 2001 and 2002. Both are used for single documents summarization. Since our algorithm is unsupervised, we only use the test dataset. The test sets of DUC 2001 and 2002 consist of 308 documents and 567 documents respectively. Each document has two different summaries that consist of 100 words approximately. DUC documents are news documents extracted from newswires including Wall Street

Journal, AP newswire, and Financial Times[2]. The third dataset is the Corpus of Event Summaries [4]. It covers the following domains: aviation accidents, train accidents, earth quakes and terrorist attacks. The corpus consists of 78 documents. Each document contains an average of 43 words. The gold summaries provided in this corpus are abstractive summaries, one summary for each document.

IV. Summarizing Email Threads

Before creating the BC3 summaries we automatically preprocessed the corpus to remove sentences we think are irrelevant. These sentences include: signatures at the end of each email message, sentences containing less than 4 words, sentences containing only an email address, a phone number, a fax number or a website and sentences containing message headers. According to our approach, such sentences may have words with high scores even if they are irrelevant to the content of email thread. We also remove quoted sentences because we assume that a quoted sentence already appeared in another email message. We do not handle the problem of hidden emails, which are emails that are quoted inside another email message but do not exist as a separate email message in the corpus [6]. We expected that the number of hidden emails is not going to be large because the corpus is extracted from technical discussion forums.

In order to generate summaries, email messages in a single thread are combined in one document and then scored and ranked using our method. We found that the number of sentences is different in gold summaries created by each annotator. In order to be able to evaluate our system appropriately, we generate 3 different summaries for each email thread. The number of sentences in each summary is equivalent to the number of sentences in each of the 3 gold summaries. For instance if summary S1 created by annotator A1 consists of 3 sentences, we generate a summary that consists of the top 3 sentences in the system output.

V. Summarizing DUC and Concisus Corpus Documents

DUC summaries consist of approximately 100 words each. Therefore, we select the top n sentences with the highest score generated by our algorithm such that the total number of words in a summary is around 100. For the Concisus Corpus we also select the top n sentences such that the total number of words in a document summary is approximately equivalent to its gold summary. This method was appropriate since the number of words in each gold summary is different. The corpus has summaries consisting of words ranging between a minimum of 16 words and a maximum of 88 words.

[2]http://www-nlpir.nist.gov/projects/duc/data.html

VI. Evaluation

We used ROUGE as an evaluation measure in our experiment. ROUGE is an n-gram recall between a candidate summary and a reference summary. There are different ROUGE measures including ROUGE-N, ROUGE-L and ROOUGE-S. We report the average results of all the three previous measures between each generated summary and gold standard summary. These measures work well with single document summarization tasks according to [7] and ROUGE-1 achieved the highest results in our experiments. We used 95% confidence interval and stemmed words. Stop words were included in the evaluation. Including stop words and using stemming resulted in higher scores. For ROUGE-S, the maximum gap length between two words is 1, and unigrams are included. Recall scores for the three datasets used in our experiments are shown in table I.

Previous unsupervised email summarization methods reported in literature [8], [9], [10] used Enron data set, and the summaries of this set are not publicly available. The only previous research work that used BC3 data set, to the best of our knowledge, is reported in [11]. However, they use a supervised approach in summarization with 90% of the BC3 dataset as training set, and 10% as test set with 10 folds cross validation. They report weighted recall score about 80%. We calculated weighted recall for our summaries and it was about 55%, which is higher than the results of both MEAD [12] and CWS [8] unsupervised summarization systems on the same dataset. Both systems achieved around 40% as reported in [11].

	ROUGE-1	ROUGE-L	ROUGE-SU1
BC3	79.8	79.4	71.8
DUC 01	45.7	40.6	26.2
DUC 02	48.8	44	29.4
CONCISUS	47.7	39.1	30.6

Table I
Recall ROUGE scores

	ROUGE-1	ROUGE-L
CC method	53.12	49.78
TextRank	50.23	-
Our System	**48.8**	**45.7**
Baseline	48.7	44.4

Table II
Recall ROUGE Scores of DUC 2002 Set

In table II we report the results of our method when applied to DUC dataset compared to the state of the art results of TextRank [13], coherent chunking method (CC method) introduced by [14] and the baseline for DUC 2002 summarization task.

VII. Discussion

After we investigated the BC3 dataset, we report the following observations about our results. First, 130 sentences were removed during cleaning the data and were actually included in the gold summaries (around 4% of the total number of sentences in the corpus). 67% of these sentences are quoted sentences, and the remaining ones are either URLs, emails or attachments. We found that annotators included quoted sentences sometimes in their reference summary. We assume that quoted text is already mentioned in another email message and hence we did not include it in our summarization process. Second, the corpus contains around 9 hidden emails, and sentences extracted from 7 hidden emails were included in the gold summaries. In our work we do not handle that problem because we assumed that the number of hidden emails will be few due to the technical nature of the discussion forum from which emails were extracted. Finally, our approach is totally unsupervised where email specific features were not used in generating summaries. In addition, we use the whole dataset in evaluation and hence the size of test set in our experiment is different from the size of the test set used in experiments reported in [11]. However, our approach achieved better results than MEAD and CWS summarization systems and achieved high recall scores using ROUGE Measure.

As for the results of DUC, we expect that the main concepts in a document are not contained in few words, which might make it hard for the system to find better summaries with higher recall. Our results are comparable with baseline though.

Gold summaries of the Concisus corpus are short abstractive summaries consisting of one to two sentences while our summaries are extractive ones. Our system managed to achieve high results for very short summaries given that the original documents consist of 222 words on average while summaries consist of 43 words on average. This indicates that our summaries managed to capture the gist of a document in one or two sentences successfully, even though reference summaries are abstractive unlike our summaries.

VIII. Related Work

Generally, extractive summarization is done in 3 steps: (1) identify the boundaries of sentences (2) rank sentences according to their importance in text (3) select the top n sentences to create a summary [15]. Most systems differ in step 2. Some of the features used to identify an important sentence in a document include word frequency, cue words, the position of a sentence inside a document and its length [16], [17], [18], [19]. External sources such as Wikipedia and news search query logs can be used to aid in the process of selecting sentences to include in a summary [19]. Semantic knowledge is also employed to produce a coherent summary of a document. [20] identified strong lexical chains in text and used them to extract important sentences while [21] used textual entailment to create document summaries. [15], [12], [22], [23], [14] used graph methods in order to generate extractive summaries. The advantage of these methods is that they are unsupervised and need no training data.

The genre of summarized documents can affect the method of summarization. When summarizing emails for instance, most methods make use of email specific features such as the number of recipients, the number of responses to a message, the type of an email sentence (whether it is greeting, chit-chat or a sentence that contains a task) [24], [8], the use of speech acts and subjectivity of email sentences [11] and quotation graphs [8], [9].

IX. Conclusion

This paper presented an unsupervised method for single document summarization that employs dependency graphs of sentences to determine keywords; words on which most other words are dependent. We evaluated our method using different summarization corpora. Our results show that our method can achieve high results in email document summarization task. The system we proposed managed to create short event summaries with high recall as well. Our system proved to be suitable for multiple genres of documents. It is also language independent which makes it portable to other languages.

References

[1] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, pp. P10 008+, Oct. 2008. [Online]. Available: http://dx.doi.org/10.1088/1742-5468/2008/10/P10008

[2] A. R. El-Kilany, S. R. El-Beltagy, and M. E. El-Sharkawi, "Sentence Compression via Clustering of Dependency Graph Nodes," in *International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE'12)*, 2012.

[3] J. Ulrich, G. Murray, and G. Carenini, "A Publicly Available Annotated Corpus for Supervised Email Summarization," in *AAAI08 EMAIL Workshop*. Chicago, USA: AAAI, 2008.

[4] H. Saggion and S. Szasz, "The CONCISUS Corpus of Event Summaries," in *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, N. C. C. Chair), K. Choukri, T. Declerck, M. U. Doan, B. Maegaard, J. Mariani, J. Odijk, and S. Piperidis, Eds. Istanbul, Turkey: European Language Resources Association (ELRA), May 2012.

[5] M. Marneffe, B. Maccartney, and C. Manning, "Generating Typed Dependency Parses from Phrase Structure Parses," in *Proceedings of LREC-06*, 2006, pp. 449–454.

[6] G. Carenini, R. Ng, X. Zhou, and E. Zwart, "Discovery and regeneration of hidden emails," in *Proceedings of the 2005 ACM symposium on Applied computing*, ser. SAC '05. New York, NY, USA: ACM, 2005, pp. 503–510. [Online]. Available: http://doi.acm.org/10.1145/1066677.1066792

[7] C.-y. Lin, "Rouge: a package for automatic evaluation of summaries," in *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, 2004, pp. 25–26.

[8] G. Carenini, R. T. Ng, and X. Zhou, "Summarizing email conversations with clue words," in *Proceedings of the 16th international conference on World Wide Web*, ser. WWW '07. New York, NY, USA: ACM, 2007, pp. 91–100. [Online]. Available: http://doi.acm.org/10.1145/1242572.1242586

[9] G. Carenini, R. T. Ng, and X. Zhou, "Summarizing emails with conversational cohesion and subjectivity," in *Proceedings of ACL-08: HLT*. Columbus, Ohio: Association for Computational Linguistics, June 2008, pp. 353–361.

[10] M. Dredze, H. M. Wallach, D. Puller, and F. Pereira, "Generating summary keywords for emails using topics," in *Proceedings of the 13th international conference on Intelligent user interfaces*, ser. IUI '08. New York, NY, USA: ACM, 2008, pp. 199–206. [Online]. Available: http://dx.doi.org/10.1145/1378773.1378800

[11] G. M. R. N. Jan Ulrich Giuseppe Carenini, "Regression-Based Summarization of Email Conversations," in *3rd Int'l AAAI Conference on Weblogs and Social Media (ICWSM-09)*. San Jose, CA: AAAI, 2009.

[12] D. Radev, T. Allison, S. Blair-Goldensohn, J. Blitzer, A. Çelebi, S. Dimitrov, E. Drabek, A. Hakim, W. Lam, D. Liu, J. Otterbacher, H. Qi, H. Saggion, S. Teufel, M. Topper, A. Winkel, and Z. Zhang, "{MEAD} — {A} platform for multidocument multilingual text summarization," in *Conference on Language Resources and Evaluation (LREC)*, Lisbon, Portugal, May 2004.

[13] R. Mihalcea, "Graph-based ranking algorithms for sentence extraction, applied to text summarization," in *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, ser. ACLdemo '04. Stroudsburg, PA, USA: Association for Computational Linguistics, 2004. [Online]. Available: http://dx.doi.org/10.3115/1219044.1219064

[14] S. K and S. L, "An Approach to Text Summarization." in *Proceedings of the Third International Workshop on Cross Lingual Information Access: Addressing the Information Need of Multilingual Societies (CLIAWS3)*. Boulder, Colorado: Association for Computational Linguistics, Jun. 2009, pp. 53–60. [Online]. Available: http://www.aclweb.org/anthology/W09-1608

[15] K. Patil and P. Brazdil, "SUMGRAPH: TEXT SUMMARIZATION USING CENTRALITY IN THE PATHFINDER NETWORK," *IADIS International Journal on Computer Science and Information System*, vol. 2, pp. 18–32, 2007.

[16] J. L. Neto, A. A. Freitas, and C. A. A. Kaestner, "Automatic Text Summarization Using a Machine Learning Approach," in *Proceedings of the 16th Brazilian Symposium on Artificial Intelligence: Advances in Artificial Intelligence*, ser. SBIA '02. London, UK, UK: Springer-Verlag, 2002, pp. 205–215. [Online]. Available: http://dl.acm.org/citation.cfm?id=645853.669480

[17] J. M. Conroy and D. P. O'leary, "Text summarization via hidden Markov models," in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, ser. SIGIR '01. New York, NY, USA: ACM, 2001, pp. 406–407. [Online]. Available: http://doi.acm.org/10.1145/383952.384042

[18] D. Shen, J.-T. Sun, H. Li, Q. Yang, and Z. Chen, "Document summarization using conditional random fields," in *Proceedings of the 20th international joint conference on Artifical intelligence*, ser. IJCAI'07. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007, pp. 2862–2867. [Online]. Available: http://dl.acm.org/citation.cfm?id=1625275.1625736

[19] K. M. Svore, L. Vanderwende, and C. J. C. Burges, "Enhancing single-document summarization by combining ranknet and third-party sources." in *EMNLP-CoNLL*. ACL, 2007, pp. 448–457.

[20] R. Barzilay and M. Elhadad, "Using Lexical Chains for Text Summarization," in *Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*. Madrid, Spain, 1997, pp. 10–17.

[21] E. Lloret, O. Ferrández, R. Muñoz, and M. Palomar, "A Text Summarization Approach under the Influence of Textual Entailment." in *NLPCS*, B. Sharp and M. Zock, Eds. INSTICC Press, 2008, pp. 22–31. [Online]. Available: http://dblp.uni-trier.de/db/conf/nlucs/nlpcs2008.html#LloretFMP08

[22] R. Mihalcea and P. Tarau, "Textrank: Bringing order into texts," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Barcelona, Spain, 2004. [Online]. Available: http://acl.ldc.upenn.edu/acl2004/emnlp/pdf/Mihalcea.pdf

[23] G. Erkan and D. R. Radev, "{LexRank}: Graph-based Lexical Centrality as Salience in Text Summarization," *Journal of Artificial Intelligence Research*, vol. 22, 2004. [Online]. Available: http://www.sadl.uleth.ca/nz/cgi-bin/library?e=q-000-00—0jair–00-0-0–0prompt-10—4—-dcr–0-1l–1-en-50—20-about-erkan–00031-001-1-0utfZz-8-00&a=d&c=jair&cl=search&d=erkan04a

[24] S. Corston-Oliver, E. Ringger, M. Gamon, and R. Campbell, "Task-Focused Summarization of Email," in *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, S. S. Marie-Francine Moens, Ed. Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 43–50.