Hierarchical summarization of financial reports with RUNNER

Marina Litvak
Shamoon College of Engineering (SCE)
Beer-Sheva
Israel
marinal@ac.sce.ac.il

Natalia Vanetik
Shamoon College of Engineering (SCE)
Beer-Sheva
Israel
natalyav@sce.ac.il

Tzvi Puchinsky
Shamoon College of Engineering (SCE)
Beer-Sheva
Israel
tzvipu@ac.sce.ac.il

Abstract

With the constantly growing amount of information, the need arises to automatically summarize this written information. One of the challenges in the summary is that it's difficult to generalize. For example, summarizing a news article is very different from summarizing a financial earnings report. This paper reports an approach for summarizing financial texts, which are different from the documents from other domains at least in three parameters: length, structure, and format. Our approach considers these parameters, it is adapted to hierarchical structure of sections, document length, and special "language". The approach builds a hierarchical summary, visualized as a tree with summaries under different discourse topics. The approach was evaluated using extrinsic and intrinsic automated evaluations, which are reported in this paper. As all participants of the Financial Narrative Summarisation (FNS 2020) shared task, we used FNS2020 dataset for evaluations.

1 Introduction

The area of text summarization exists for several decades, since the first work of Luhn (Luhn, 1958). Since then, the summarization approaches evolved from simple and straightforward extractive unsupervised approaches to abstractive supervised methods, using deep learning language models (Liu, 2019). However, the most advanced seq2seq models (transformers) are very limited in input size and, therefore, are inapplicable to long texts. Also, only few of state-of-the-art summarizers consider hierarchical structure of the input documents (Yang and Wang, 2008; Zhang et al., 2019), their key concepts (Ouyang et al., 2009) or topics (Wang et al., 2013; Akhtar, 2017) and build a hierarchical summary (Christensen et al., 2014; Akhtar et al., 2019). Usually, hierarchical summary is built per document collection. The top level of hierarchy provides a general overview and users can navigate the hierarchy to drill down for more details on topics of interest.

There is a growing interest in the application of automatic and computer-aided approaches for extracting, summarising, and analysing both qualitative and quantitative financial data, as a series of FNP and related workshops (El-Haj, 2019; El-Haj et al., 2018) recently demonstrates. However, summarization of documents in financial domain is usually limited to summarization of financial news (Filippova et al., 2009; Yang and Wang, 2003; de Oliveira et al., 2002; Baralis et al., 2016; Zhang et al., 2018) which are not very different from the general news in length and format. Only few attempts were made to summarize financial reports (Isonuma et al., 2017), which are different from the news articles in at least four parameters: length, structure, format, and lexicon.

This paper reports an approach for hierarchical summarization of financial reports. Financial annual reports in the data of Financial Narrative Summarisation (FNS 2020) shared task[1](El-Haj et al., 2020) are long, have many sections, and are written in "financial" language using many special terms, numerical data, and tables. Our system for hie**R**archical s**U**mmarization fi**N**a**N**cial r**E**po**R**ts (shortly RUNNER) considers discourse and topic hierarchical structure and builds a hierarchical view of the summarized

This work is licensed under a Creative Commons Attribution 4.0 International License.
License details: http://creativecommons.org/licenses/by/4.0/.

[1] http://wp.lancs.ac.uk/cfie/fns2020/

Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation, pages 213–225
Barcelona, Spain (Online), December 12, 2020.

report with interactive user interface. In contrast with the previous works on hierarchical summarization, our approach considers the internal hierarchical structure of a document and its topics instead mapping it to a global hierarchy of entire corpus.

2 Hierarchical Summarization with RUNNER

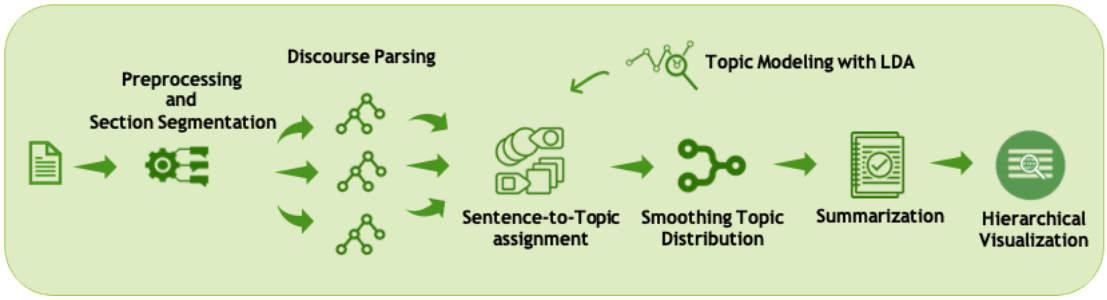

Figure 1: RUNNER pipeline

RUNNER utilizes two main methods: topic modeling (TM) and discourse parsing (DP). The pipeline of the proposed methodology is depicted in Figure 1 and includes the following steps:

Text preprocessing, that includes text cleaning, sentence splitting and tokenization. We developed our own tool that cleaned text before segmenting it to sentences and tokens. Financial reports usually contain a lot of sections, figures, and tables. Because the text files in the FNS-2020 dataset were obtained by converting pdf files to plain texts, these texts contain a lot of "noise" left from broken tables and meta-data such as section and page numbers. We cleaned the noise by measuring the ratio between text and numbers and ratio between number of words and whitespaces. Lines with ratio less than 0.4 were removed. Then, regular expressions were applied to find and mark such entities as URL, phone number, date, time, email. Finally, non-Unicode characters were filtered out. Figure 2 demonstrates the example of text before and after preprocessing.

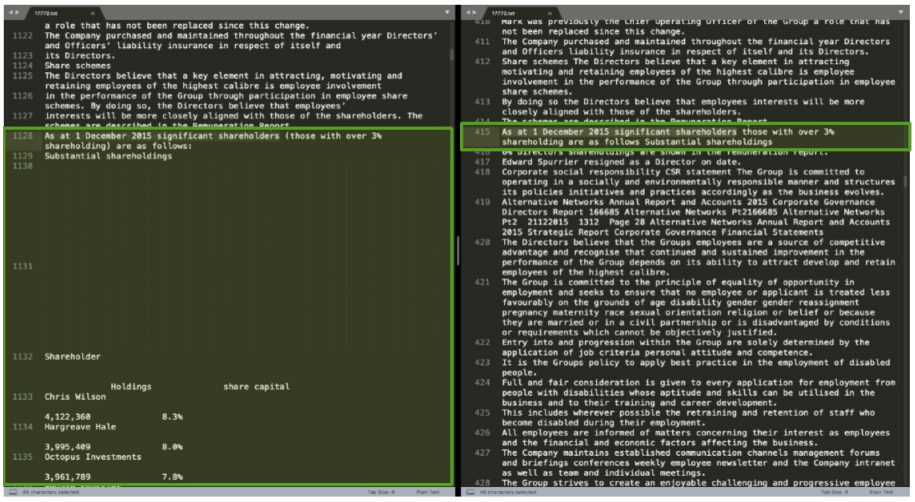

Figure 2: Text before and after preprocessing.

Section segmentation, where section headers are identified and a document is segmented into sections. The section titles were extracted following the heuristic rules saying that (1) each title appears in a separate line, (2) does not end with period mark, and (3) contains only few (up to 5) words with (4) each word either starting with capital case letter or containing only upper case letters. The extracted

candidates were then compared against the list of 13 manually edited titles[2]. The candidate that obtained Jaccard similarity above 0.4 to one of the titles from the list was extracted as a title. The text body between two consequent titles was marked as a section.

Discourse parsing of each section. For discourse parsing we used the CODRA parser (Joty et al., 2015). CODRA parser performs two-part process: (1) a discourse segmenter creates a segmentation analysis on the sentence level and EDU's for the discourse parsing process and (2) a discourse parser parses the text on sentence level and document level to identify relations between parts of sentences and sentences in the document. Figure 3 shows an example discourse tree. Leaf node stands for a sentence or a part of a sentence. The rhetorical analysis of the parser starts from a breaking a text into Elementary Discourse Units (EDUs). Because EDUs do not span across multiple sentences, this segmentation task finds EDUs inside the sentence boundaries. As a result, some sentences (actually, most, according to our observations) are split into EDUs. Every EDU is marked as a *nucleus* (an essence part) or a *satellite* (a complementary part of the related nucleus), based on the relation that they are connected to. Internal (relation) nodes represent different inter-sentence relations: elaboration, same-unit, etc.

Figure 3: Discourse tree for sentences from "remuneration report" section of document 17941.txt in FNS-2020 dataset.

Topic modeling. For topic modeling we applied Latent Dirichlet Allocation (LDA) model (Blei et al., 2003). It was applied on all files in the FNS-2020 dataset with predefined number of topics[3].

Topic-to-text assignment, where each sentence (or sentence part) represented by a leaf node of the discourse tree, is assigned to one of the topics obtained by LDA. We refer topic probabilities $p(t|w)$ for all sentence S words $w \in S$ as their topic-related importance scores. Therefore, we extract a dominant topic ($t \in T$) for each sentence S, as a topic with the maximal normalized sum of topic probabilities for all sentence words $w \in S$: $\max_{t \in T} \frac{\sum_{w \in S} p(t|w)}{\sum_{w \in S} 1}$. Figure 4 shows an example of a topic-to-sentence assignment.

Topic distribution smoothing. We noticed that after single text nodes (that stand for sentences or

[2]Titles that appear in almost every report in FNS-2020 dataset, such as: 'chairman statement', 'chief executive officer CEO review', 'chief executive officer CEO report', 'governance statement', 'remuneration report', 'business review', 'financial review', 'operating review', 'highlights', 'auditors report', 'risk management', 'chairman governance introduction', 'corporate social responsibility CSR disclosures'.

[3]We experimented with 4, 6, and 10 topics, and finally decided to keep 10 topics as best performing value. After reviewing word clusters representing topics, we found that they most probably represent key information from the different sections of the financial report

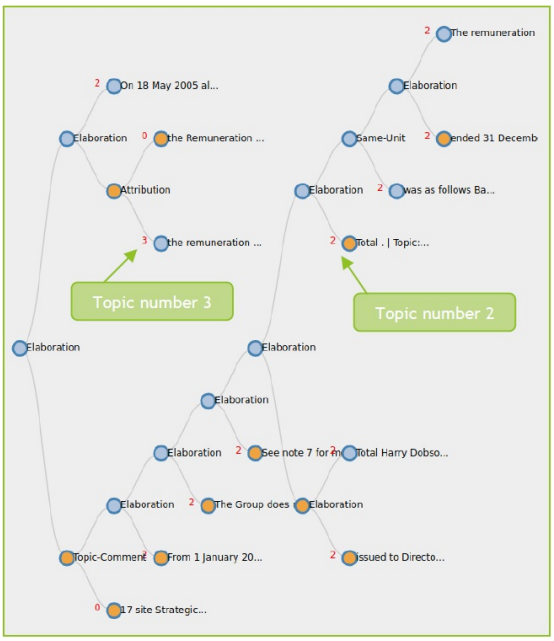

Figure 4: Topic-to-text assignment.

sentence parts) are assigned to topics, we can get unexpected topic distribution where two parts of the same sentence or two adjacent sentences inside the same paragraph and/or belonging to the same discourse relation are assigned to different topics, and transition from one topic to another is not coherent.[4] We decided to smooth topic distribution by extrapolating one dominant topic on entire block of adjacent sentences and sentence parts, connected by a direct discourse relation. We denote nodes with at least one leaf node as "simple" (see Figure 5) and all leaves in its sub-tree are finally assigned to one dominant topic, so that a "random" noise is left out. The implement this approach as follows. We know that all leaf nodes are arranged in the natural sequential order of their texts from right-to-left (top-down) in a discourse tree. We assume that the important information usually comes first (important part of a sentence usually precedes its complementary part, and a sentence stating some fact usually precedes a sentence that elaborates more about this fact) and, therefore, upper right nodes and nucleuses should propagate their topics on their siblings. According to this assumption and our empirical observations on each parameter's influence, the final impact factor NI of node n is calculated as follows. $NI(n) = \sum_{i=1}^{3} w_i \times f_i(n)$, where:

- f_1 is a relative depth feature $rd(n) = \frac{h(t)+1-d(n)}{h(t)+1}$, $h(t)$ is a tree height, $d(n)$ is n's depth
- f_2 is a position feature $pos(n) = \begin{cases} 1, \text{if } n \text{ is on right} \\ 0, \text{else} \end{cases}$
- f_3 is a discourse label feature $l(n) = \begin{cases} 1, \text{if } n \text{ is nucleus} \\ 0, \text{else} \end{cases}$
- $w_1 = 0.5$, $w_2 = 0.3$, and $w_3 = 0.2$

Then, the final dominant topic for a "simple" sub-tree is calculated as follows: $\max_{t \in T} \{\sum_{n \in leaves} NI(n) * score_{t,n}\}$. After topic-to-sentence assignment (at previous stage), every leaf node has non-zero value for only one dominant topic, other topics have $score_{t,n} = 0$.

We also experimented with the *second strategy* of assignment topics to sentences, where we do not assign a dominant topic to each leaf but operate their vectors of topic weights $\vec{v_n} = (w_{t_1}, w_{t_2}, \ldots, w_{t_{|T|}})$,

[4]We assume that in a natural topic distribution, that is usually observed in general domains, topics must flow from one paragraph (or sections or cluster of sentences) to another, without mix of topics inside clusters.

where w_{t_i} is a normalized sum of topic t_i probabilities for all sentence words, as calculated in previous stage. The dominant topic is assigned to entire sub-tree (under the "simple" node) by summing the topic distribution vectors multiplied by the importance score of their nodes and choosing a topic with a maximal score. Formally, the dominant topic is assigned as follows: $\max_{t \in T} \sum_{n \in leaves} \vec{v_n} \times NI(n)$.

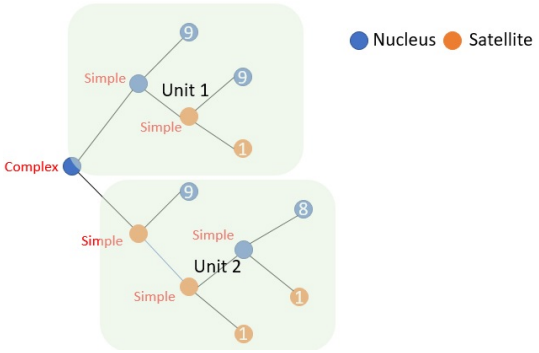

Figure 5: Example of "simple" sub-trees with initial LDA-based sentence-over-topics distribution. Based on this distribution and discourse structure, one dominant topic is finally calculated for each "simple" sub-tree. Given the initial topic assignment to leaves, the dominant topic of the first (upper) sub-tree is finally assigned to 9.

Summarization of entire report (regardless visualization) and of each section (for visualization needs) was performed by two different greedy strategies:

(1) All topics t are ranked by their importance $TI(t)$ (normalized sum of their probabilities for all document/section words). Then, summaries are created by extraction of nucleuses from each topic, in the topics' importance order, until the maximum length limit is reached. As for entire report a summary should not exceed 1000 words according to the shared task instructions, we limit a section summary to 100 words.

(2) All nucleuses are ranked by their importance. An importance score of nucleus m, represented by a node n in the discourse tree, is calculated as $NI(n) \times TI(dt)$, where dt is a dominant topic assigned to m. Then, in a greedy manner, summaries are created by extraction of nucleuses in their importance order, until the maximum length limit is reached.

We report the results for both strategies in the Experiments section.

Hierarchical visualization. At this stage RUNNER creates an interactive html file with the data from all the stages for a user to browse. The file contains the following sections: (1) original text; (2) processed XML text after cleaning and section segmentation; (3) discourse trees for all the sections; (4) sentences (nodes) with assigned topics after smoothing; (5) the final hierarchical tree with the section summaries, and (6) a general report summary. For visualization and interactive user's navigation, the following tree structure of a document is built and present to a user: root represents an entire document and points to its sections, each section is split to major topics inside this section after smoothing, and each topic points to a summary of this particular section focused on the chosen topic. Visualization is performed in interactive manner, upon a user's request. Figure 6 shows an example of such a tree. Demo video[5] demonstrates all interactive options provided by the system.

3 Experiments

We performed two types of evaluation for our summarization method[6]: extrinsic and intrinsic. Extrinsic evaluation can help judge the quality of the summaries based on how they affect the completion of specific task, while intrinsic evaluation estimates the quality of the generated output directly, usually by comparing it to the human-generated content.

[5] https://drive.google.com/file/d/14qMRUhZIwaVoSltaLPSiH6NZx13M_9ue/view

[6] Only general summaries were evaluated, due to limitations of gold standard summaries provided with a test set.

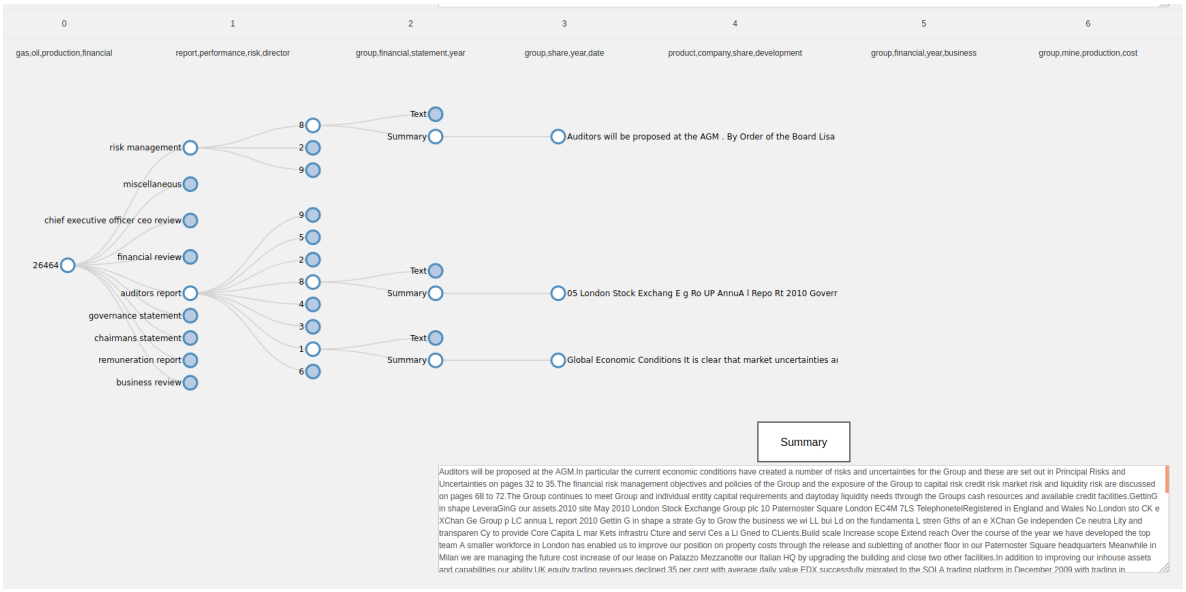

Figure 6: Visualization summary tree.

3.1 Dataset

The Financial Narrative Summarisation (FNS 2020) shared task aims to demonstrate the value and challenges of applying automatic text summarisation to financial text written in English, usually referred to as financial narrative disclosures. The task dataset has been extracted from UK annual reports published in PDF file format. UK annual reports are lengthy documents with around 80 pages on average, some annual reports could span over more than 250 pages, while the summary length should not exceed 1000 words. The training set includes 3,000 annual reports, with 3-4 human-generated summaries as gold standard. For the evaluation process the test set of 500 files were provided. To address the time limitations and processing long files[7] the project reduced the length of the original files (to 15000 characters) to be able to process in feasible time limit (20 minutes per file at most). Table 1 contains the dataset statistics. Please note that our method is unsupervised and does not require a training set. Therefore, we calculated average statistics only for the documents of the test set.

# documents	avg words	avg sentences	avg sections	avg words/section
500	63583.7	40	5.3	13452.6

Table 1: FNS 2020 dataset statstics. Test set.

3.2 Evaluated methods

We evaluate four variations of our approach (denoted by RUNNER$_{ij}$), which are combinations of two strategies for node importance calculation ($i \in \{1, 2\}$) and two strategies of summarizaion ($j \in \{1, 2\}$), and compare their results with two baseline methods—MUSE (Litvak et al., 2010) and POLY (Litvak and Vanetik, 2013). MUSE is a supervised approach based on a genetic algorithm, it was trained on 30 randomly selected gold standard summaries provided with FNS-2020 dataset. POLY is unsupervised approach based on linear programming, it was applied with Maximal Weighted Term Sum (OBJ1 in (Litvak and Vanetik, 2013)) objective function.

3.3 Extrinsic evaluation

We decided to utilize clustering as an evaluated task and see how similar the clustering of summaries is to that of the original reports. For this purpose, K-means (Lloyd, 1982) was applied on original reports and

[7] mostly, due to a very time-consuming discourse parsing

then on their summaries. Different clustering quality metrics were calculated on both clustering results and compared. As preprocessing before K-Means application, we performed corpus vectorization with tf-idf and Principal component analysis (Pearson, 1901) to reduce dimensionality. Number of clusters was set to three. In order to compare between clustering results we used the following metrics: Davies-Bouldin index (DBI) (Davies and Bouldin, 1979), Dunn index (DI) (Dunn, 1974), Silhouette coefficient (SC) (Rousseeuw, 1987), inter-cluster distance (inter-CD—sum of the square distance between each cluster centroid), intra cluster distance (intra-CD—sum of the square distance from the items of each cluster to its centroid), maximum radius (MR—largest distance from an instance to its cluster centroid), and average radius (AR—sum of the largest distances in each cluster divided by the number of clusters). Since algorithms that produce clusters with high intra-cluster similarity and low inter-cluster similarity will have a low DBI, the clustering algorithm that produces a collection of clusters with the smallest DBI is considered the best algorithm. Dunn index is defined as the ratio between the minimal inter-cluster distance to maximal intra-cluster distance. Therefore, algorithms that produce clusters with high DI are more desirable. Silhouette coefficient contrasts the average distance to elements in the same cluster with the average distance to elements in other clusters. Objects with a high SC value are considered well clustered, objects with a low value may be outliers. We also measured Precision, Recall, and Purity for all clusters of summaries, assuming that clusters of reports are ground truth.

Table 2 shows the comparative results. The best scores are marked in bold and the second best are marked by grey background. It can be seen that clustering of reports gains better scores than clustering of summaries in most metrics. However, smaller intra-cluster similarity and radius mean that clusters of summaries are smaller and more distant from each other. Also, as MUSE SC score shows, the summaries clusters may contain less outsiders. As it can be seen, RUNNER produces summaries with second best scores for sever (out of ten) metrics, meaning that it succeeds to keep the most representative information and filter out the redundant one in its summaries. The most important, that despite close results, clustering of summaries took much less time (16 times faster) than clustering of entire reports—2 versus 33 seconds for entire test set of 500 documents.

corpus	DBI	DI	SC	inter-CD	intra-CD	MR	AR	P	R	Purity
Reports	**0.702**	**0.558**	0.498	**0.331**	0.112	0.370	0.316	**1.000**	**1.000**	**1.000**
MUSE	0.673	0.456	**0.542**	0.278	0.091	0.253	**0.196**	0.407	0.492	0.602
POLY	0.857	0.318	0.414	0.206	**0.080**	0.307	0.277	0.396	0.369	0.504
RUNNER_{11}	0.909	0.321	0.386	0.223	0.097	0.281	0.252	0.303	0.284	0.682
RUNNER_{12}	0.847	0.461	0.403	0.218	0.089	0.231	0.200	0.388	0.337	0.412
RUNNER_{21}	0.884	0.335	0.383	0.223	0.090	0.322	0.273	0.599	0.569	0.636
RUNNER_{22}	0.834	0.469	0.408	0.219	0.089	**0.224**	0.204	0.430	0.377	0.420

Table 2: Clustering results.

3.4 Intrinsic evaluation

Intrinsic evaluation was performed using ROUGE metrics (Lin, 2004) which work by comparing an automatically produced summary against a set of reference summaries (typically human-produced). We applied three ROUGE metrics—ROUGE-1, ROUGE-2, and ROUGE-L. Table 3 show the results, with recall, precision, and F-measure for each metric. It can be seen that RUNNER performs better than POLY (both are unsupervised), and even outperforms MUSE (which is supervised) in one metric (ROUGE-L, Precision), meaning that its summaries are less "scattered" and more coherent (and therefore probably more readable) then other summaries. The comparative results with other systems participating in the FNS 2020 shared task can be seen in Appendix, Tables 4-7.[8]

3.5 Tools and runtime environment

For LDA, we used the Python gensim4 package. Corpus tf-idf vectorization and K-means clustering were performed by the Python sklearn package. For running Rouge, we used ROUGE 2.05 java pack-

[8]Only one variation of RUNNER was submitted to the task evaluations, which is the closest to RUNNER_{11}. However, since then RUNNER's code was significantly updated, therefore the scores are not the same.

system	R-1 R	R-1 P	R-1 F	R-2 R	R-2 P	R-2 F	R-L R	R-L P	R-L F
MUSE	**0.483**	**0.413**	**0.433**	**0.311**	**0.198**	**0.234**	**0.486**	0.381	**0.419**
POLY	0.324	0.253	0.274	0.147	0.088	0.105	0.270	0.182	0.212
RUNNER_{11}	0.290	0.396	0.324	0.150	0.153	0.144	0.290	**0.396**	0.324
RUNNER_{12}	0.358	0.337	0.337	0.181	0.127	0.144	0.331	0.285	0.299
RUNNER_{21}	0.293	0.392	0.324	0.151	0.151	0.144	0.294	0.325	0.300
RUNNER_{22}	0.358	0.337	0.337	0.181	0.127	0.144	0.331	0.285	0.299

Table 3: Rouge results.

age (Ganesan, 2018). Our approach was implemented in Python and run on Intel Pentium Gold G5400 with 16GB memory server with 40GB swap file configured.

4 Conclusions and Future Work

This paper describes a new method for hierarchical summarization of financial reports, based on integrating the discourse structure and topic modeling. In future, we intend to apply this method and its extension to educational materials, which also have highly hierarchical structure and an evolving flow of topics in a discourse. Hierarchical summarization can help to organize those materials in a hierarchical structure and provide users with interactive navigation to the topics of interest. RUNNER's source code is available[9] and can be run using the provided instructions[10].

Acknowledgements

We would like to thank Alla Kitaeva for supporting this project in a scope of the final undergraduate project.

References

Nadeem Akhtar, Hira Javed, and Tameem Ahmad. 2019. Hierarchical summarization of text documents using topic modeling and formal concept analysis. In *Data Management, Analytics and Innovation*, pages 21–33. Springer.

Nadeem Akhtar. 2017. Hierarchical summarization of news tweets with twitter-lda. In *Applications of Soft Computing for the Web*, pages 83–98. Springer.

Elena Baralis, Luca Cagliero, and Tania Cerquitelli. 2016. Supporting stock trading in multiple foreign markets: a multilingual news summarization approach. In *Proceedings of the Second International Workshop on Data Science for Macro-Modeling*, pages 1–6.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Janara Christensen, Stephen Soderland, Gagan Bansal, et al. 2014. Hierarchical summarization: Scaling up multi-document summarization. In *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 902–912.

David L Davies and Donald W Bouldin. 1979. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227.

Paulo Cesar Fernandes de Oliveira, Khurshid Ahmad, and Lee Gillam. 2002. A financial news summarization system based on lexical cohesion. In *Proceedings of the International Conference on Terminology and Knowledge Engineering, Nancy, France*.

Joseph C Dunn. 1974. Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, 4(1):95–104.

Mahmoud El-Haj, Paul Rayson, and Andrew Moore. 2018. The first financial narrative processing workshop (fnp 2018). In *Proceedings of the LREC 2018 Workshop*.

[9] https://github.com/Tzvi23/Hierarchical-Summarization-Part1

[10] https://drive.google.com/drive/folders/1YxnNQ-9ebPX1Gtd6Dmr0to6UIBgudflC

Mahmoud El-Haj, Ahmed AbuRa'ed, Nikiforos Pittaras, and George Giannakopoulos. 2020. The Financial Narrative Summarisation Shared Task (FNS 2020). In *The 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation (FNP-FNS 2020*, Barcelona, Spain.

Mahmoud El-Haj. 2019. Multiling 2019: Financial narrative summarisation. In *Proceedings of the Workshop MultiLing 2019: Summarization Across Languages, Genres and Sources*, pages 6–10.

Katja Filippova, Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. 2009. Company-oriented extractive summarization of financial news. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 246–254.

Kavita Ganesan. 2018. Rouge 2.0: Updated and improved measures for evaluation of summarization tasks. *arXiv preprint arXiv:1803.01937*.

Masaru Isonuma, Toru Fujino, Junichiro Mori, Yutaka Matsuo, and Ichiro Sakata. 2017. Extractive summarization using multi-task learning with document classification. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2101–2110.

Shafiq Joty, Giuseppe Carenini, and Raymond T Ng. 2015. Codra: A novel discriminative framework for rhetorical analysis. *Computational Linguistics*, 41(3):385–435.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Marina Litvak and Natalia Vanetik. 2013. Mining the gaps: Towards polynomial summarization. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 655–660.

Marina Litvak, Mark Last, and Menahem Friedman. 2010. A new approach to improving multilingual summarization using a genetic algorithm. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 927–936.

Yang Liu. 2019. Fine-tune bert for extractive summarization. *arXiv preprint arXiv:1903.10318*.

Stuart Lloyd. 1982. Least squares quantization in PCM. *IEEE transactions on information theory*, 28(2):129–137.

Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165.

You Ouyang, Wenjie Li, and Qin Lu. 2009. An integrated multi-document summarization approach based on word hierarchical representation. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 113–116.

Karl Pearson. 1901. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.

Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

Chi Wang, Xiao Yu, Yanen Li, Chengxiang Zhai, and Jiawei Han. 2013. Content coverage maximization on word networks for hierarchical topic summarization. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 249–258.

Christopher C Yang and Fu Lee Wang. 2003. Automatic summarization for financial news delivery on mobile devices. In *WWW (Posters)*.

Christopher C Yang and Fu Lee Wang. 2008. Hierarchical summarization of large documents. *Journal of the American Society for Information Science and Technology*, 59(6):887–902.

Yong Zhang, Erdan Chen, and Weidong Xiao. 2018. Extractive-abstractive summarization with pointer and coverage mechanism. In *Proceedings of 2018 International Conference on Big Data Technologies*, pages 69–74.

Xingxing Zhang, Furu Wei, and Ming Zhou. 2019. Hibert: Document level pre-training of hierarchical bidirectional transformers for document summarization. *arXiv preprint arXiv:1905.06566*.

Appendix

system	R-1 R	R-1 P	R-1 F
SRIB2020-SYSTEM3	0.612	0.393	0.466
SRIB2020-SYSTEM2	0.611	0.392	0.465
SUMSUM-BASE	0.494	0.481	0.462
SUMSUM-BERT	0.450	0.530	0.460
KG-SUMMAR-NN	0.568	0.381	0.445
SUMSUM-01	0.447	0.511	0.442
HULAT-1	0.536	0.393	0.441
KG-SUMMAR-SVM	0.495	0.416	0.438
KG-SUMMAR-S-LSTM	0.506	0.406	0.438
MUSE	0.483	0.413	0.433
CIST-BUPT-RUN3	0.434	0.449	0.428
SUMTO-SUMMARY-3PE	0.447	0.426	0.424
SUMTO-SUMMARY-2PE	0.441	0.427	0.422
SUMTO-SUMMARY-1PE	0.431	0.437	0.421
CIST-BUPT-RUN2	0.418	0.440	0.416
AMEX-ENSEMBLE	0.442	0.408	0.412
AMEX-BILSTM	0.436	0.406	0.409
FORTIA-SYSTEM1	0.428	0.431	0.407
HULAT-2	0.503	0.352	0.402
CIST-BUPT-RUN1	0.405	0.423	0.401
FORTIA-SYSTEM2	0.394	0.410	0.384
FORTIA-SYSTEM3	0.365	0.370	0.352
UOB-NLP-SECOND-SUMMARIES	0.307	0.315	0.301
SCE-SUMMARY (RUNNER)	0.288	0.399	0.297
UOB-NLP-THIRD-SUMMARIES	0.296	0.316	0.296
AMEX-TEXTRANK	0.353	0.271	0.295
UOB-NLP-FIRST-SUMMARIES	0.296	0.315	0.295
SRIB2020-SYSTEM1	0.241	0.378	0.283
POLY	0.324	0.253	0.274
LEXRANK-SUMMARY	0.337	0.269	0.264
TEXTRANK-SUMMARY	0.414	0.118	0.172

Table 4: Comparative results. Rouge-1.

system	R-2 R	R-2 P	R-2 F
SUMSUM-BERT	0.365	0.295	0.306
SUMSUM-BASE	0.398	0.259	0.294
SRIB2020-SYSTEM3	0.451	0.222	0.289
SRIB2020-SYSTEM2	0.448	0.220	0.288
SUMSUM-01	0.358	0.277	0.286
FORTIA-SYSTEM1	0.299	0.282	0.274
HULAT-1	0.412	0.200	0.261
SUMTO-SUMMARY-3PE	0.296	0.228	0.249
CIST-BUPT-RUN3	0.288	0.233	0.248
KG-SUMMAR-SVM	0.357	0.199	0.247
KG-SUMMAR-NN	0.402	0.184	0.246
KG-SUMMAR-S-LSTM	0.360	0.193	0.243
FORTIA-SYSTEM2	0.247	0.263	0.241
SUMTO-SUMMARY-1PE	0.270	0.225	0.237
CIST-BUPT-RUN2	0.272	0.224	0.237
SUMTO-SUMMARY-2PE	0.276	0.217	0.235
MUSE	0.311	0.198	0.234
HULAT-2	0.375	0.177	0.233
CIST-BUPT-RUN1	0.258	0.206	0.220
AMEX-ENSEMBLE	0.264	0.192	0.214
AMEX-BILSTM	0.260	0.190	0.211
FORTIA-SYSTEM3	0.207	0.222	0.202
SCE-SUMMARY (RUNNER)	0.159	0.157	0.138
UOB-NLP-SECOND-SUMMARIES	0.149	0.110	0.121
LEXRANK-SUMMARY	0.193	0.107	0.120
AMEX-TEXTRANK	0.184	0.097	0.120
SRIB2020-SYSTEM1	0.114	0.138	0.118
UOB-NLP-THIRD-SUMMARIES	0.140	0.108	0.117
UOB-NLP-FIRST-SUMMARIES	0.140	0.107	0.116
POLY	0.147	0.088	0.105
TEXTRANK-SUMMARY	0.229	0.044	0.070

Table 5: Comparative results. Rouge-2.

system	R-L R	R-L P	R-L F
SRIB2020-SYSTEM3	0.605	0.376	0.456
SRIB2020-SYSTEM2	0.603	0.377	0.455
MUSE	0.470	0.370	0.407
SUMTO-SUMMARY-3PE	0.410	0.395	0.394
SUMTO-SUMMARY-1PE	0.406	0.385	0.387
HULAT-1	0.444	0.357	0.386
SUMTO-SUMMARY-2PE	0.403	0.382	0.385
FORTIA-SYSTEM1	0.397	0.397	0.381
AMEX-ENSEMBLE	0.408	0.365	0.378
AMEX-BILSTM	0.402	0.360	0.372
HULAT-2	0.392	0.358	0.364
FORTIA-SYSTEM2	0.374	0.373	0.362
FORTIA-SYSTEM3	0.341	0.339	0.331
CIST-BUPT-RUN3	0.324	0.348	0.329
SUMSUM-BASE	0.332	0.350	0.324
CIST-BUPT-RUN2	0.311	0.352	0.324
SUMSUM-BERT	0.304	0.389	0.322
KG-SUMMAR-NN	0.389	0.278	0.318
KG-SUMMAR-S-LSTM	0.344	0.307	0.317
CIST-BUPT-RUN1	0.294	0.361	0.317
SUMSUM-01	0.304	0.375	0.313
KG-SUMMAR-SVM	0.340	0.303	0.312
AMEX-TEXTRANK	0.246	0.245	0.237
SRIB2020-SYSTEM1	0.213	0.254	0.225
SCE-SUMMARY (RUNNER)	0.225	0.286	0.223
LEXRANK-SUMMARY	0.210	0.263	0.218
UOB-NLP-THIRD-SUMMARIES	0.227	0.202	0.208
UOB-NLP-FIRST-SUMMARIES	0.226	0.203	0.208
UOB-NLP-SECOND-SUMMARIES	0.223	0.204	0.207
TEXTRANK-SUMMARY	0.235	0.197	0.206
POLY	0.260	0.177	0.205

Table 6: Comparative results. Rouge-L.

system	R-SU4 R	R-SU4 P	R-SU4 F
FORTIA-SYSTEM1	0.344	0.332	0.318
SUMSUM-BERT	0.406	0.268	0.302
FORTIA-SYSTEM2	0.299	0.313	0.290
SUMSUM-BASE	0.442	0.236	0.288
SRIB2020-SYSTEM3	0.508	0.209	0.288
SRIB2020-SYSTEM2	0.506	0.208	0.286
SUMSUM-01	0.398	0.253	0.282
HULAT-1	0.464	0.193	0.264
SUMTO-SUMMARY-3PE	0.353	0.223	0.264
SUMTO-SUMMARY-1PE	0.332	0.218	0.254
MUSE	0.375	0.201	0.253
FORTIA-SYSTEM3	0.263	0.271	0.253
SUMTO-SUMMARY-2PE	0.340	0.211	0.252
CIST-BUPT-RUN3	0.346	0.209	0.251
KG-SUMMAR-SVM	0.411	0.188	0.248
KG-SUMMAR-S-LSTM	0.417	0.182	0.245
CIST-BUPT-RUN2	0.330	0.204	0.243
KG-SUMMAR-NN	0.464	0.170	0.242
HULAT-2	0.430	0.173	0.239
AMEX-ENSEMBLE	0.328	0.194	0.235
AMEX-BILSTM	0.325	0.192	0.232
CIST-BUPT-RUN1	0.315	0.190	0.228
SCE-SUMMARY (RUNNER)	0.208	0.164	0.158
UOB-NLP-SECOND-SUMMARIES	0.214	0.123	0.150
SRIB2020-SYSTEM1	0.165	0.149	0.149
UOB-NLP-THIRD-SUMMARIES	0.203	0.122	0.146
UOB-NLP-FIRST-SUMMARIES	0.203	0.121	0.145
AMEX-TEXTRANK	0.250	0.108	0.144
LEXRANK-SUMMARY	0.253	0.117	0.140
POLY	0.213	0.105	0.135
TEXTRANK-SUMMARY	0.302	0.048	0.079

Table 7: Comparative results. Rouge-SU4.