Quote Detection: A New Task and Dataset for NLP

Selma Tekir and **Aybüke Güzel** and **Samet Tenekeci** and **Bekir Ufuk Haman**
Izmir Institute of Technology
Dept. of Computer Engineering
35430 Izmir, Turkey
{selmatekir, aybukeguzel, samettenekeci, bekirhaman}@iyte.edu.tr

Abstract

Quotes are universally appealing. Humans recognize good quotes and save them for later reference. However, it may pose a challenge for machines. In this work, we build a new corpus of quotes and propose a new task, quote detection, as a type of span detection. We retrieve the quote set from Goodreads and collect the spans through a custom search on the Gutenberg Book Corpus. We run two types of baselines for quote detection: Conditional random field (CRF) and summarization with pointer-generator networks and Bidirectional and Auto-Regressive Transformers (BART). The results show that the neural sequence-to-sequence models perform substantially better than CRF. From the viewpoint of neural extractive summarization, quote detection seems easier than news summarization. Moreover, model fine-tuning on our corpus and the Cornell Movie-Quotes Corpus introduces incremental performance boosts. Finally, we provide a qualitative analysis to gain insight into the performance.

1 Introduction

Human beings have aesthetic appeal. They create and enjoy different works of art. Among these, literary works contain the highest form of bookish experience. People enjoy reading novels and highlighting textual segments that are distinctive and memorable, which we can term quotes. Humans can readily recognize good quotes and save them for later reference. However, it may pose a challenge for machines since the quote detection task relies mostly on semantic features such as memorability and distinctiveness.

The Goodreads website[1] stores a collection of quotes that are extracted from different resources to meet users' expectations. This community-wide interest has led us to propose a work in this context.

This paper proposes a new NLP task, quote detection, as a variant of span detection, and releases a benchmark quotes dataset. In the literature, there is a movie quotes corpus for binary quote classification (Danescu-Niculescu-Mizil et al., 2012). There is also a similar task of quotation detection and classification (Pareti et al., 2013, Papay and Padó, 2020, Vaucher et al., 2021) where the aim is to extract/identify direct, indirect, or mixed speech parts from the text. Quote detection is different and unique in that spans represent the free-standing textual segments that are distinctive and favorable for later reference (Table 1). A similar trend has been in the Viral Texts Project, which interrogates the qualities that cause literary texts to go viral by their reprints in newspapers (Cordell and Smith, 2022). Furthermore, quotes are different from subtexts as subtexts underlie a new meaning connected with a speaker's motive in particular.

To challenge the problem, we first formulate it as a sequence tagging problem and work with a statistical baseline, conditional random field (CRF). Secondly, we regard it as a type of summarization. To have a baseline performance, we experiment with two neural sequence-to-sequence models; the pointer-generator network (Vinyals et al., 2015) and Bidirectional and Auto-Regressive Transformers (BART) (Lewis et al., 2020), respectively. The solutions' performances confirm that the task is relatively easier than the existing summarization problems but is a difficult sequence tagging problem.

The main contributions of this paper are: (a) a corpus of 5015 quotes with their 10 sentence-length left and 10 sentence-length right contexts; (b) a distinctiveness analysis based on language model log-likelihoods and comparison against movie quotes (the Cornell Movie-Quotes Corpus); (c) experimental results from summarization and sequence tagging methods; (d) a qualitative analysis to give insight on errors (whether they are mainly precision

[1] goodreads.com/quotes

Proceedings of the 7th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature (LaTeCH-CLfL2023), pages 21–27
May 5, 2023 ©2023 Association for Computational Linguistics

or recall-based).

2 A Corpus of Quotes

To construct the quote dataset, we rely on two primary resources. The first one is the Goodreads platform which shares a voted collection of quotes. The collection consists of $348,085$ instances, each with Quote, Title, Author, Likes, and Tags columns. We download this collection from Kaggle[2]. As humans recognize good quotes and user rating is an indicator for recognition, we exclude the rows with ≤ 10 likes from the dataset. Another filtering criterion is the language of quotes. We detect the language with the help of Python's NLTK library and remove the non-English quotes. After these two filtering steps, the quote dataset includes $100,837$ rows.

The second resource is the free eBook library Project Gutenberg[3]. We download books in plain text format to check whether a quote appears in the referenced book. For this purpose, we search the title and author of the relevant books in the search section of the Project Gutenberg site and collect the book's plain text links. Then, we scrape plain texts using plain text links by the BeautifulSoup library of Python. We remove the quotes that do not comply with the UTF-8 standard and that give a page not found error (404 error). We also exclude song lyrics and philosopher speeches as we cannot extract the contexts they appear from Project Gutenberg. Finally, we discard quotes from some books of contemporary literature that are not accessible. After filtering, the total number of rows is reduced to 8670.

To search for a quote in the plain text of the target book, we first trim the standard book header and footer using regular expressions. Then, we have a custom search based on the F1 score. We compute the F1 score based on overlap-based precision and recall definitions to determine the best possible match. In our context, precision is the ratio of the number of shared words to the total number of words in the target span, and recall is defined as the ratio of the number of shared words to the total number of words in the ground truth (quote). We also consider the lengths of quotes in this procedure, having faced the fact that a quote can be a phrase within a sentence, a single sentence, or a text made up of a group of sentences. Therefore, we process a sliding window of quote length when searching for the closest sentence or sentences in the book text. For example, if the quote consists of three sentences, we calculate the F1 score by sliding over three sentences in the text and return the three-sentence span with the highest score as the most similar context for the quote.

The next step is the validation of the returned spans. We arrange quotes in different bins based on F1 score thresholds to decide whether each span corresponds to the wanted quote. The match becomes better as the threshold increases, but the dataset size shrinks. We observe that F1 scores increase as the quote lengths decrease. On the other hand, there is no noticeable difference in the quotes' lengths in each bin. We choose the optimum F1 score threshold as 50%, with an average 2.40 sentence count, 22.97 word count per sentence, and 5015 quotes in total.

In the construction of the final collection (T50[4]), each quote is enclosed by 10 left and 10 right sentences. The appendix A.1 includes an example instance.

2.1 Analysis on Distinctiveness of Quotes

Quotes are known to use distinctive vocabulary (Danescu-Niculescu-Mizil et al., 2012). To check the distinctiveness of our quotes dataset, we compare quotes and non-quotes contexts in terms of language use. In particular, we calculate negative log-likelihoods based on a state-of-the-art language model (GPT-2) (Radford et al., 2019) to measure their unique vocabulary use. We rely on the Mann-Whitney U non-parametric test of the null hypothesis that there is no difference between the negative log-likelihoods of quotes and non-quotes in our dataset to test the statistical difference. The test returns a p-value of $P < .001$, which confirms that we can reject the null hypothesis in favor of the alternative. Moreover, the negative log-likelihoods for quotes are higher than their non-quote counterparts, which means that the vocabulary choice in quotes is more discrete.

To further test the language characteristics of our quotes dataset, we run the analysis of variance (Table 2) where the groups are the Cornell Movie-Quotes Corpus quotes (Mov^{+}), their negative samples (Mov^{-}), our dataset's quotes (T50^{+}), and our dataset's non-quote contexts (T50^{-}). We first test the group null hypothesis and get a p-value

[2] kaggle.com/datasets/faellielupe/goodreads-quotes

[3] gutenberg.org

[4] https://cloud.iyte.edu.tr/index.php/s/YO407M8uAglLIY3

Task	Main Source / Structure of Input	Indicators	Examples
Quote Detection	Free-form literary texts (books, poems, lyrics)	Semantic features and distinctive vocabulary	- There is always something left to love. - No medicine cures what happiness cannot.
Quotation Detection	Excerpts from direct or indirect speech (news, political speech, dialog)	Quotation marks and speech verbs	- “I’m in love with you,” he said quietly. - Authorities say that the risk still remains.

Table 1: Quote vs Quotation Detection

of $P < .001$ to reject it safely. When we consider pairwise differences, the results confirm a statistical difference between T50$^-$ and T50$^+$ and Mov$^-$ and Mov$^+$. On the other hand, the test reveals no difference between T50$^+$ and Mov$^+$, which is another piece of evidence for quote recognition. The negative mean differences in the μ_d column in each row indicate that Group 1 has a lower negative log-likelihood than Group 2, which again shows that Group 1 has a higher probability of occurrence based on the language model.

Group 1	Group 2	μ_d	p-value	Reject
T50$^-$	**T50$^+$**	-43.98	0.001	**True**
T50$^-$	Mov$^-$	-31.91	0.001	True
T50$^-$	Mov$^+$	-65.38	0.001	True
T50$^+$	Mov$^-$	$+12.06$	0.507	False
T50$^+$	**Mov$^+$**	-21.39	0.063	**False**
Mov$^-$	**Mov$^+$**	-33.46	0.022	**True**

Table 2: ANOVA on negative log likelihoods. μ_d: mean difference, $^+$: quote, $^-$: non-quote

3 Experiments

3.1 Datasets

We experiment with two datasets as part of the evaluation. The first is the proposed corpus (T50), and the second is an adapted version of the Cornell Movie-Quotes Corpus (Danescu-Niculescu-Mizil et al., 2012). Although both datasets are similar in nature, they are in different domains; the former is on books while the latter is on movies. We briefly describe the latter in the following subsection.

3.1.1 Cornell Movie-Quotes Corpus

Cornell Movie-Quotes (Danescu-Niculescu-Mizil et al., 2012), is a dataset[5] of movie scripts with memorability annotations. It contains a total of 2197 memorable and non-memorable short text pairs. The dataset also includes 6282 movie quotes (IMDB memorable quotes), each linked to a movie script line.

As the proposed task is quote detection rather than quote classification, we need extended spans of quotes. Since the dataset includes the full movie scripts where the quotes appear, we expand each quote with its left and right contexts, which are 4 script lines each, creating a total length of 9.

3.2 Baselines

3.2.1 Conditional Random Fields (CRF)

As the first baseline, we utilize conditional random fields (CRF) (Lafferty et al., 2001) to catch the span of quotes. Accordingly, each training sample includes 10-length left and right contexts of the quote and the quote itself. CRF computes a feature vector for each word in the training instance and maximizes the likelihood of the output label given the feature vector. The feature vector consists of whether the current word is in the upper or title case or a digit, its first bi-gram and tri-gram, the part-of-speech (POS) tag, the left and right neighbors’ case, and digit information with their POS tags. The motivation is that the model captures distinctive vocabulary by its character n-grams. Alternatively, we ran CRF with a feature vector of the word, word level bi-gram, word level tri-gram, their POS tags, 3rd person pronoun (indicator for generality), and the indefinite article (indicator for generality) because Danescu-Niculescu-Mizil et al. (Danescu-Niculescu-Mizil et al., 2012) worked with these features to quantify the level of distinctiveness of a quote. However, our experiments prove that character-level features perform better than their word-level counterparts for CRF. Thus, distinctive vocabulary plays a vital role in the discrimination of quotes. We label each token as previous (P), quote (Q), or next (N). We execute CRF for 500 iterations on T50 and movie datasets and evaluate the model’s performance using ROUGE scores.

3.2.2 Pointer Generator Networks

The second baseline is a pointer generator network (See et al., 2017) for text summarization. It com-

[5] cs.cornell.edu/~cristian/memorability.html

bines an LSTM-based sequence-to-sequence model with a pointer network (Vinyals et al., 2015) to summarize news articles and can specify the weight of abstractive/extractive summarization as a variable. As the quote detection task is extractive in nature, we fine-tune and evaluate the model in a fully extractive form. The base model is pre-trained on CNN (Hermann et al., 2015) data without coverage loss (the coverage loss is responsible for making the output more abstractive). We fine-tune this model with the train partition of the T50 data for 5000 steps in batches of 16.

3.2.3 Bidirectional and Auto-Regressive Transformers (BART)

The last baseline is BART (Lewis et al., 2020). BART is a neural sequence-to-sequence model that aims to improve the masked language model and next-sentence prediction objectives within the end-to-end transformer architecture by shuffling the order of sentences and allowing longer sequences to be masked. The model is capable of identifying different types of transformations to the input and making predictions about overall sentence length.

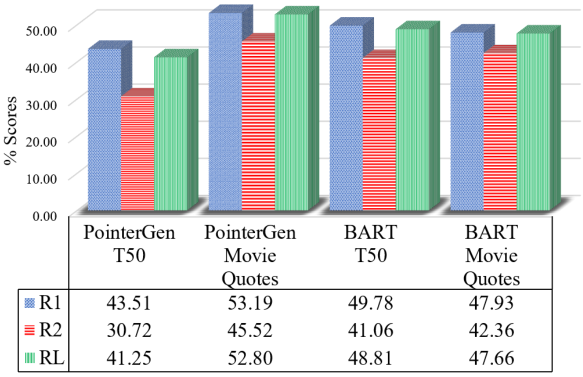
3.3 Evaluation Metrics

Using a train-validation-test split of 0.7-0.1-0.2, sequence-to-sequence models are evaluated using recall-oriented overlap-based ROUGE (Lin, 2004) metrics. For the formal definitions of the evaluation metrics, see Appendix A.2.

3.4 Results

In our experiments with 5-fold cross-validation, the CRF baseline achieves average R1 scores of 20.28 ± 2.99% and 26.42 ± 0.13% on T50 and movie datasets, respectively. We report the detailed results, including the R2 and RL scores, in our code repository[6].

Given a test instance in T50, the T50 fine-tuned pointer generator network predicts the ground-truth quote with an R1 score of 43.51% (Figure 1 the leftmost diagram). When we apply the same fine-tuning to the Cornell Movie-Quotes data, we obtain an R1 score of 53.19% on movie quotes. Compared to the CNN pre-trained model result (39.53%) in news summarization, we observe performance improvements using task-specific fine-tuning with T50 and Cornell Movie Quotes.



	PointerGen T50	PointerGen Movie Quotes	BART T50	BART Movie Quotes
🟦 R1	43.51	53.19	49.78	47.93
🟥 R2	30.72	45.52	41.06	42.36
🟩 RL	41.25	52.80	48.81	47.66

Figure 1: ROUGE Scores

We perform the same fine-tuning operations with BART on both datasets, resulting in R1 scores of 49.78% and 47.93% (Figure 1 rightmost). The result with the T50 dataset mirrors BART's improvement over the pointer generator network on the summarization benchmarks. However, BART falls behind the pointer generator network on Movie Quotes, which can be attributed to the domain and average length differences.

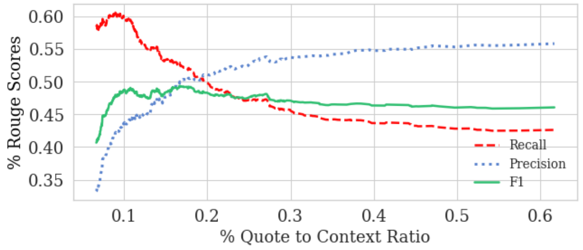


Figure 2: T50 Point-Gen Scores by the relative length

An essential factor for the model performance is the length of quotes. Figure 2 depicts the relationship of the obtained rouge scores with the quote-to-context length ratio. As can be seen from the plot, when the quote length gets higher relative to that of the context, precision increases as a word's probability of being inside the quote gets higher. On the other hand, it becomes difficult to pick all the words in the ground-truth quote correctly, which results in a fall in the recall. Moreover, the challenge remains in the recall, as can be observed by the parallel convergence of recall and F1 curves.

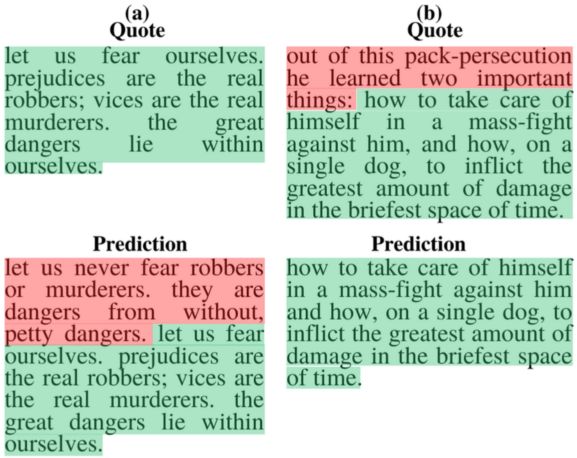
In general, longer quotes favor precision, while shorter ones favor recall. Given similar context lengths, a T50 quote (47 words on average) is almost twice as long as a movie quote (22 words on average). Length statistics for the T50 and Cornell Movie-Quotes datasets in all train, test and validation partitions are given in Table 3.

[6] https://github.com/Darg-Iztech/pointer_summarizer
https://github.com/Darg-Iztech/quote-detection-crf

	T50				Mov			
	Context Lengths		Quote Lengths		Context Lengths		Quote Lengths	
	mean	std.	mean	std.	mean	std.	mean	std.
Train	590	266	47	49	107	62	21	24
Test	606	241	48	44	108	58	22	26
Val	593	263	46	46	109	70	21	24

Table 3: T50 and Movie Quotes Word Count Statistics

3.5 Qualitative Analysis


(a)
Quote
let us fear ourselves. prejudices are the real robbers; vices are the real murderers. the great dangers lie within ourselves.

Prediction
let us never fear robbers or murderers. they are dangers from without, petty dangers. let us fear ourselves. prejudices are the real robbers; vices are the real murderers. the great dangers lie within ourselves.

(b)
Quote
out of this pack-persecution he learned two important things: how to take care of himself in a mass-fight against him, and how, on a single dog, to inflict the greatest amount of damage in the briefest space of time.

Prediction
how to take care of himself in a mass-fight against him and how, on a single dog, to inflict the greatest amount of damage in the briefest space of time.

Figure 3: Quote prediction examples

Quantitative results prove that finding out quotes in endless contexts poses a difficulty in precision (e.g., 0.1 summary to context ratio in Figure 2), but while the quote to context ratio grows, recall becomes the determining factor.

We perform a qualitative analysis to observe what kind of errors is common in our experiments. We depict two cases (Figure 3) where the model is inclined to overshoot (a) and undershoot (b) the ground truth quotes. In the usual case, it extends the prediction from the beginning (a) or from the end where recall is perfect, but precision is low. Less often, the model undershoots the actual quote as in example (b) of the figure, yielding a perfect precision score and a low recall.

What we can reflect from these examples is that, generally, longer quotes favor precision, while shorter ones favor recall. When the context length is considered, recall increases as the quote-to-context length ratio decreases, and precision follows the opposite pattern. Thus, one can manipulate the context length to steer the recall-precision balance for the model training.

4 Conclusion & Future Work

What makes a sequence of words a quote? Although this question is hard to answer, we empirically show that it has a distinctive vocabulary using language model log-likelihoods on T50. This phenomenon was also confirmed by (Danescu-Niculescu-Mizil et al., 2012) on movie quotes. Moreover, the selected baselines show that it is possible to recognize a quote within its context.

Ultimately, this paper presents the quote detection task by releasing a new dataset with baseline performances. Our results state that quote detection is easier than news summarization using neural summarization. As for sequence tagging, detecting quotes by classifying the beginning and end tokens seems relatively more complicated. Thus, there is much room for improvement over mentioned baselines. We hope this task leads to the development of new methods and data sharing.

5 Limitations

The paper proposes a new task on quote detection and releases a dataset, and provides baselines to meet the purpose. The dataset includes the quotes that appear in books. Although we find similar patterns in movie quotes, the task's difficulty may differ for quotes in other contexts, e.g., lyrics and poems.

Moreover, the provided summarization and sequence tagging baselines give an idea about the difficulty level of the proposed task. They are in no way the best systems to solve the problem.

Finally, in constructing the dataset, each quote is enclosed by 10 left and 10 right sentences. This choice can be considered subjective, knowing that the quote lengths, context lengths, and their ratio have a role in the performance. Accordingly, we provide comments on this behavior in our quantitative and qualitative analyses.

Acknowledgements

The NVIDIA Corporation donated the Titan V used for this research.

References

Ryan Cordell and David Smith. 2022. Viral texts: Mapping networks of reprinting in 19th-century newspapers and magazines. http://viraltexts.org. Accessed: 2023-03-27.

Cristian Danescu-Niculescu-Mizil, Justin Cheng, Jon Kleinberg, and Lillian Lee. 2012. You had me at hello: How phrasing affects memorability. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 892–901, Jeju Island, Korea. Association for Computational Linguistics.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1693–1701, Cambridge, MA, USA. MIT Press.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Sean Papay and Sebastian Padó. 2020. RiQuA: A corpus of rich quotation annotation for English literary text. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 835–841, Marseille, France. European Language Resources Association.

Silvia Pareti, Tim O'Keefe, Ioannis Konstas, James R. Curran, and Irena Koprinska. 2013. Automatically detecting and attributing indirect quotations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 989–999, Seattle, Washington, USA. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1073–1083. Association for Computational Linguistics.

Timoté Vaucher, Andreas Spitz, Michele Catasta, and Robert West. 2021. Quotebank: A corpus of quotations from a decade of news. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, WSDM '21, page 328–336, New York, NY, USA. Association for Computing Machinery.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

A Appendix

A.1 Example: A Quote with its Span

Are you not happy in your? Naughty darling. At Dolphin's barn charades in Luke Doyle's house. Mat Dillon and his bevy of daughters: Tiny, Atty, Floey, Maimy, Louy, Hetty. Molly too. Eighty-seven that was. Year before we. And the old major, partial to his drop of spirits. Curious she an only child, I an only child. So it returns. ***Think you're escaping and run into yourself. Longest way round is the shortest way home.*** And just when he and she. Circus horse walking in a ring. Rip van Winkle we played. Rip: tear in Henny Doyle's overcoat. Van: breadvan delivering. Winkle: cockles and periwinkles. Then I did Rip van Winkle coming back. She leaned on the sideboard watching. Moorish eyes. Twenty years asleep in Sleepy Hollow.
@highlight
Think you're escaping and run into yourself. Longest way round is the shortest way home.

A.2 Evaluation Metrics

Given an n-gram length N, the ROUGE-N metric between a candidate document D_C and a reference document D_R is given by:

$$\text{ROUGE-N}(D_C, D_R) = \frac{\sum\limits_{r_i \in D_R} \sum\limits_{\omega \in r_i} T(\omega, D_C)}{\sum\limits_{r_i \in D_R} T(r_i)} \tag{1}$$

where r_i are the sentences in the reference document D_R, $T(\omega, D_C)$ is the number of times the specified n-gram ω occurs in the candidate document D_C, and $T(r_i)$ is the number of all n-grams in the specified reference sentence r_i.

To calculate ROUGE-L, we first calculate the recall (R_{lcs}) and precision (P_{lcs}) scores based on the longest common subsequences in the reference (D_R) and candidate (D_C) documents by:

$$R_{lcs}(D_C, D_R) = \frac{\sum\limits_{r_i \in D_R} |\text{LCS}_{\cup}(D_C, r_i)|}{L(D_R)} \tag{2}$$

$$P_{lcs}(D_C, D_R) = \frac{\sum\limits_{r_i \in D_R} |\text{LCS}_{\cup}(D_C, r_i)|}{L(D_C)} \tag{3}$$

where $L(D_R)$ is the number of words in D_R, $L(D_C)$ is the number of words in D_C, and $\text{LCS}_{\cup}(D_C, r_i)$ is the union of the longest common subsequences in D_R and D_C, which is given by:

$$\begin{aligned} &\text{LCS}_{\cup}(D_C, D_R) = \\ &\qquad \cup_{r_i \in D_R} \{w | w \in \text{LCS}(D_C, r_i)\} \end{aligned} \tag{4}$$

where $\text{LCS}(D_C, r_i)$ is the set of longest common subsequences in the candidate document D_C and sentence r_i from reference document D_R. Using R_{lcs} and P_{lcs}, ROUGE-L can be defined as:

$$\begin{aligned} &\text{ROUGE-L}(C, R) = \\ &\qquad \frac{(1 + \beta^2) R_{lcs}(C, R) P_{lcs}(C, R)}{R_{lcs}(C, R) + \beta^2 P_{lcs}(C, R)} \end{aligned} \tag{5}$$

where the parameter β controls the relative importance of the precision and recall. Because the ROUGE score favors recall, β is typically set to a high value.