Evaluation Metrics for Depth and Flow of Knowledge in Non-fiction Narrative Texts

Sachin Pawar, Girish K. Palshikar, Ankita Jain, Mahesh Singh
Mahesh Rangarajan, Aman Agarwal, Vishal Kumar[*]**, Karan Singh**[*]
TCS Research, Tata Consultancy Services Limited, India.
{sachin7.p,gk.palshikar,ankita7.j,mahesh.psingh,mahesh.rangarajan,aman.agarwal6}@tcs.com
{vtkumar022,karanfru627}@gmail.com

Abstract

In this paper, we describe the problem of automatically evaluating quality of knowledge expressed in a non-fiction narrative text. We focus on a specific type of documents where each document describes a certain technical problem and its solution. The goal is not only to evaluate the quality of knowledge in such a document, but also to automatically suggest possible improvements to the writer so that a better knowledge-rich document is produced. We propose new evaluation metrics to evaluate quality of knowledge contents as well as flow of different types of sentences. The suggestions for improvement are generated based on these metrics. The proposed metrics are completely unsupervised in nature and they are derived from a set of simple corpus statistics. We demonstrate the effectiveness of the proposed metrics as compared to other existing baseline metrics in our experiments.

1 Introduction

Documents containing non-fiction narrative text occur in many practical applications; e.g., essays, news, emails, safety or security incident reports, insurance claims, medico-legal reports, troubleshooting guides, user manuals etc. It is important to ensure that each such document is of high quality, for which purpose we need metrics that measure their quality. While metrics for readability (or comprehensibility) are obviously usable, we need specialized metrics that attempt to measure quality of non-fiction narrative text in terms of the specific characteristics. Fictional narratives are characterized in terms of structural elements such as conflicts, plot points, dialogues, characters, character arcs, focus, etc.; there is extensive literature about their linguistic analysis. However, non-fiction narrative texts are comparatively less studied in linguistics; e.g., (Sorock et al., 1996; Bunn et al., 2008; McKenzie et al., 2010; PBG, 2014). In this paper, we identify following characteristics of non-fiction narrative texts: (i) depth and variety of factual and conceptual knowledge elements present; (ii) distribution of different classes of sentences that represent essential aspects of information content; and (iii) flow and coherence of different types of sentences. We also propose novel quantitative metrics for measuring the quality of non-fiction narrative texts in terms of these characteristics.

In this paper, we focus on a specific type of non-fiction narrative text documents – *Contextual Master (CM) stories*. Contextual Master^{TM} is a registered trademark of TCS[1], which refers to an associate who has over the time gained a significant contextual knowledge or understanding of a business domain or a particular client's business. An *CM story* is a short narrative text that a CM writes to describe a particular instance where he/she has used the expert-level knowledge to solve a specific problem or to address a specific challenge. Each such CM story generally consists of 25-30 sentences (details in Section 7.1). A typical process of writing these stories is that a CM first writes some initial version which is reviewed by reviewers for knowledge contents, readability, narration flow and other aspects like grammar. Over a few iterations of incorporating reviewers' suggestions, a story is accepted to be published internally and for marketing purposes. In this paper, our goal is to develop a system for – (i) automatic evaluation of a CM story for its knowledge contents and narration flow quality, and (ii) automatic generation of suggestions for improvement so that the time needed to produce a publishable final version of a story from its initial version is reduced. The main motivations for building this system are as follows:

- Because of the automatically generated suggestions, a CM can produce a better initial

[*] Work done while working at TCS Research

[1] https://www.tcs.com/tcs-way/contextual-knowledge-mastery-tcs-client-growth

Proceedings of the The 5th Workshop on Narrative Understanding, pages 16–24
July 14, 2023 ©2023 Association for Computational Linguistics

version of a story, requiring lesser time to be invested by human reviewers. This would lead to faster publication of more such stories.

- Because of the automatic evaluation, the existing CM stories can be compared with each other or ranked as per the quality of their knowledge contents. This would be helpful to search, analyze, or refer to a few top quality CM stories in a particular business area of interest.

Automatic essay scoring or grading (Ke and Ng, 2019) is a related problem but it differs from our problem in some key aspects. Essay grading is a task of automatically scoring essays based on multiple dimensions like grammar, word usage, style, relevance to the essay topic (prompt), cohesion, coherence, persuasiveness etc. On the other hand, evaluation of non-fiction narrative texts like CM stories emphasizes more on the depth of the knowledge contents which are often not explicitly evaluated by the most essay grading techniques. To some extent, *cohesion* and *coherence* are common desirable aspects for essays as well as non-fiction narrative texts like CM stories. However, cohesion and coherence of *ideas* or *topics* is expected in essays whereas in CM stories, cohesion and coherence of certain *types of sentences* is expected. Therefore, in this paper, we propose new metrics to specifically evaluate the knowledge depth and the narration quality in terms of flow of sentence types. Here, it is important to note that we refer to *knowledge* as a more conceptual and abstract notion as compared to factual and data-oriented *information*. For example, we consider *task* as one of the knowledge markers (Section 3) which is defined as a volitional activity which needs expert knowledge to carry out (Pawar et al., 2021). A task such as "`analysed the configuration of the security protocol`" clearly represents an aspect of knowledge of a CM rather than mere factual information. Similarly, we consider specialized sentence categories (such as Solution, Benefit) introduced in Section 4 as another aspects of knowledge and hence considered as part of knowledge quality metrics.

All the proposed metrics are unsupervised in nature, i.e., they do not need any set of stories which are explicitly annotated for knowledge quality by human reviewers. The specific contributions of this paper are:

- Identifying knowledge markers (Section 3) & sentence categories (Section 4)
- Evaluation metrics for knowledge quality (Section 5) and narration flow quality (Section 6)
- Statistical analysis of effectiveness of the evaluation metrics (Section 7)

2 Problem Definition

Our goal is to determine the quality of *knowledge* and *narration flow* of a CM story with respect to a set of *knowledge quality* and *narration flow quality metrics*. Each metric is designed to capture and evaluate a certain aspect of the story, as described in detail in later sections. The problem can be specifically defined in terms of input, output and training requirements as follows:

• **Input**: A text document describing a CM story s

• **Output**: (i) An evaluation score for each of the knowledge and flow metrics for the CM story s and an aggregated score combining the individual scores. (ii) A set of suggestions for improving the CM story s.

• **Training Regime**: We assume that a set D^{train} of *final* CM stories is available which have been revised and improved by taking into consideration the suggestions from human reviewers.

Summary of the Proposed Solution: We propose a two-phase solution to this problem which is depicted in Figure 1.

• **Learning Phase**: In this phase, we use the set of *final* CM stories (D^{train}) to calculate certain corpus statistics of the proposed knowledge and flow quality metrics. As this set consists of all the stories which are already revised and improved as per human reviewers' suggestions, we assume that the corpus statistics learned from this set characterize a set of *ideal* values for these metrics.

• **Operating Phase**: In this phase, given a new CM story, we evaluate its knowledge and flow metrics with respect to the corpus statistics learned using D^{train}. We also generate a set of specific suggestions for improvement.

3 Knowledge Markers

We hypothesize that the knowledge needed for solving a particular domain or technical problem is expressed in terms of certain *knowledge markers*. These knowledge markers are mentions of some key entity types as follows:

• Skills: Names of tools, technologies, or technical

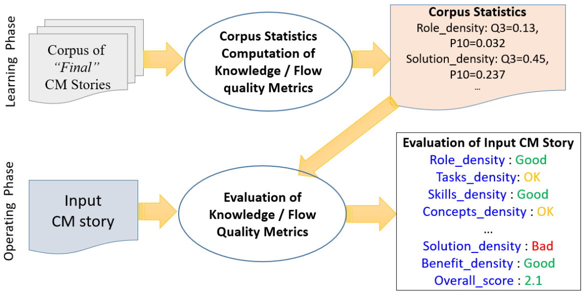
Learning Phase
Corpus of "Final" CM Stories
Corpus Statistics Computation of Knowledge / Flow quality Metrics
Corpus Statistics
Role_density: Q3=0.13, P10=0.032
Solution_density: Q3=0.45, P10=0.237
…
Operating Phase
Input CM story
Evaluation of Knowledge / Flow Quality Metrics
Evaluation of Input CM Story
Role_density : Good
Tasks_density: OK
Skills_density : Good
Concepts_density : OK
…
Solution_density : Bad
Benefit_density : Good
Overall_score : 2.1

Figure 1: Architecture of the proposed solution

concepts such as SAP S4 HANA, shell scripting, data warehousing, SolarWinds.

- Tasks: A task is a volitional and knowledge-based activity carried out by a person, a group of persons, or a system (Pawar et al., 2021). Some examples of Tasks are as follows: analysed the configuration of the security protocol, integrated SolarWinds with XYZ tool, development of several innovative solutions using S4 HANA processes.
- Roles: A specific role performed by any human expert such as IT Manager, Manufacturing Solution Architect.
- Concepts: Key noun phrases corresponding to certain domain-specific *concepts*. E.g., plastic manufacturing industry, legacy BI servers, unsupervised learning.

Entity Extraction Techniques: We use different techniques for the extraction of mentions of different entity types depending on their nature. For extraction of mentions of Skill, we use a large gazette of known skill names and simply look up in this gazette for identifying skill mentions. This gazette is created semi-automatically by combining several existing resources (like DBPedia) and a list created by a semi-supervised iterative algorithm similar to the one described in Pawar et al. (Pawar et al., 2012). Task mentions are extracted using the linguistic rules described in Pawar et al. (Pawar et al., 2021). For extracting Role mentions, we adopt a gazette lookup-based strategy similar to Skill. For identification of domain-specific Concepts, we compute domain relevance scores for all the noun phrases and select only those which are above a certain threshold. We follow the domain relevance calculation as proposed by Navigli and Velardi (Navigli and Velardi, 2004).

4 Sentence Categories

In addition to the knowledge markers, an ideal CM story should describe all the aspects of a certain problem being solved such as a brief background of the problem, the problem itself, the solution that was provided, and finally what were the benefits that were achieved. Therefore, it is important to identify presence of these aspects in a given story. We propose to identify these aspects in the form of the following sentence categories:

- Background: Sentences describing some background for the client for which a problem is being solved. E.g., The client is a European healthcare organization which offers a platform to manage user manuals and operator documents.
- Problem: Sentences describing the actual problem or challenge that is being addressed in the CM story. E.g., The users were not able to search for the mortgage related documents for some of the indexed mortgage deals.
- Expert_Knowledge: Sentences describing specific technical or domain knowledge of the CM in the context of the problem being solved. E.g., He has brought 25 years of a strong domain knowledge in supply chain area.
- Solution: Sentences describing the proposed solution, analysis, or actual implementation or execution of the solution. E.g., Agile approach was adopted to develop the planned functionalities in multiple sprints.
- Benefit: Sentences describing the benefits achieved from the implemented solution. E.g., Also, manufacturing solution enabled to bring the legacy system into SAP resulting into dropping additional manpower requirement.
- Client_Appreciation: Sentences describing the positive feedback or appreciations received from the client. E.g., The client was highly impressed with the reusability of the new automated solution.

We modelled the problem of identifying appropriate sentence categories as a multi-label, multi-class sentence classification problem. We used a multi-label setting because in some cases, a sentence may have more than one valid category. For example, the following sentence belongs to Solution as well as Benefit – He used his understanding of the client's applications and restructured the database accordingly to reduce recurring issues, which resulted in reduction in incidents by 70%.

We use a sentence classification model which is based on DistilBERT (Sanh et al., 2019), a lighter version of BERT (Devlin et al., 2018). DistilBERT model is 40% smaller than BERT while retaining its 97% language understanding capabilities. DistilBERT[2] is capable of producing semantically rich representations for any input text and the individual words in it. These representations are 768 dimensional dense vectors of real numbers ($\mathbb{R}^{768}$). We use these representations for building our classifier to predict appropriate sentence categories for a sentence in a CM story.

We now explain the model architecture in detail. Let the input sentence be S which is first passed through the pre-trained DistilBERT model to obtain – (i) [CLS] token encoding which provides the representation of the entire input text S, and (ii) the representations for each word in S.

$$\mathbf{x_{CLS}}, X = DistilBERT(S) \tag{1}$$

Here, $\mathbf{x_{CLS}} \in \mathbb{R}^{768}$ and $X \in \mathbb{R}^{L\times768}$ where L is the maximum number of words in any input sentence (we use $L = 128$). Let $X_i \in \mathbb{R}^{768}$ be the representation for the i^{th} word in S. We use attention mechanism so that the contribution of each word in S is determined based on its importance for prediction of each of the sentence categories. We use 6 attention layers corresponding to the 6 sentence categories. Each attention layer is similar to the one described in Basiri et al. (2021).

$$a_i^c = \mathbf{w_a^c}^T \cdot X_i + b^c \tag{2}$$

Here, $\mathbf{w_a^c} \in \mathbb{R}^{768}$ and $b^c \in \mathbb{R}$ are the weight vector and the bias of the attention layer for category c, respectively. $a_i^c \in \mathbb{R}$ is the score for the i^{th} word as computed by the attention layer for category c. These scores are normalized across all the words in S to obtain final attention weights (α_i^c's) which are used to obtain a weighted average of word representations.

$$\alpha_i^c = \frac{exp(a_i^c)}{\sum_{j=1}^{L} exp(a_j^c)} \; ; \; \mathbf{x_w^c} = \sum_{i=1}^{L} \alpha_i^c \cdot X_i \tag{3}$$

Finally, the overall representation ($\mathbf{x_{final}^c} \in \mathbb{R}^{1536}$) of the input sentence is obtained by concatenating representations obtained in Equations 1 and 3.

$$\mathbf{x_{final}} = [\mathbf{x_{CLS}}; \mathbf{x_w^c}] \tag{4}$$

This final representation is then passed through a linear transformation layer to obtain a hidden representation.

$$\mathbf{x_h^c} = ReLU(W_h \cdot \mathbf{x_{final}^c} + \mathbf{b_h}) \tag{5}$$

Sentence Category	**Precision**	**Recall**	**F1**
Background	0.787	0.808	0.797
Expert_Knowledge	0.817	0.870	0.843
Problem	0.762	0.701	0.730
Solution	0.803	0.704	0.750
Benefit	0.782	0.806	0.794
Client_Appreciation	0.875	0.854	0.864
Overall (micro avg)	0.794	0.766	**0.780**
Overall (macro avg)	0.804	0.791	**0.796**

Table 1: Sentence classifier evaluation results

Here, $W_h \in \mathbb{R}^{H\times1536}$ and $\mathbf{b_h} \in \mathbb{R}^H$ are the weight matrix and the bias vector of the hidden layer, where H is the number of units in the hidden layer (we use $H = 500$). Finally, each sentence category has its different output layer to predict a probability distribution over two labels – c and Not-c.

$$y_{pred}^c = Softmax(W_o^c \cdot \mathbf{x_h^c} + \mathbf{b_o^c}) \tag{6}$$

$$loss_c = CrossEntropyLoss(y_{gold}^c, y_{pred}^c) \tag{7}$$

$$loss = \sum_c loss_c \tag{8}$$

Here, $W_o^c \in \mathbb{R}^{2\times H}$ and $\mathbf{b_o^c} \in \mathbb{R}^2$ are the weight matrix and the bias vector of the output layer corresponding to the sentence category c. Cross entropy loss is computed using the predicted and the gold-standard label distributions which is summed over all categories to get the overall loss. The model is then trained to minimize this loss over the labelled training data. We used a training set of 1618 sentences which were labelled manually using a few active learning iterations. We evaluated the trained sentence classification model on a held out evaluation dataset of 636 sentences. Table 1 shows the classification performance of this model where the F1-score of around 80% was achieved.

5 Knowledge Quality Metrics

In this section, we describe our proposed *knowledge quality metrics* based on the knowledge markers and the sentence categories described in the previous sections. For a CM story s, for each knowledge marker and sentence category, we compute a metric which measures its density within the story as follows:

$$\text{Skills\_density}(s) = \frac{No.\ of\ Skill\ entity\ mentions\ in\ s}{No.\ of\ sentences\ in\ s}$$

$$\text{Solution\_density}(s) = \frac{No.\ of\ Solution\ sentences\ in\ s}{No.\ of\ sentences\ in\ s}$$

Here, the division by the number of sentences in s offsets the effect of the length of

[2] We preferred DistilBERT due to its better efficiency within constraints of our deployment environment. However, without loss of generality, the proposed technique can be used with any of the encoder models from the BERT family given sufficient compute resources.

the story. We similarly compute such metrics for all knowledge markers as well as sentence categories – Skills_density, Tasks_density, Roles_density, Concepts_density (*based on knowledge markers*), Background_density, Problem_density, Expert_Knowledge_density, Solution_density, Benefit_density, and Client_Appreciation_density (*based on sentence categories*).

One limitation of these knowledge quality metrics is that the metrics are dependent on the density of multiple knowledge markers but do not explicitly check whether multiple such markers are relevant or pertinent to each other. We plan to handle this as a future work and currently assume that there is no malicious intent in writing the document (e.g., by adding multiple irrelevant entities in text to artificially boost the quality score).

5.1 Learning Phase

As described in Figure 1, in the learning phase, we consider a corpus of *final* accepted CM stories. As these stories have been revised in several iterations to incorporate human reviewers' suggestions, we can assume that these are *ideal* from the point of view of knowledge quality. Therefore, we compute some useful corpus statistics of the knowledge quality metrics defined above. We calculate these metrics for all the CM stories in the training corpus and then we calculate the following corpus statistics for each metric m:

- Mean and Standard Deviation (μ_m and σ_m)
- Quartiles ($q1_m$: 25^{th} percentile, $q2_m$: 50^{th} percentile, i.e., *median*, and $q3_m$: 75^{th} percentile)
- Percentile ($p10_m$: 10^{th} percentile)

We have overall 10 knowledge quality metrics – based on 4 knowledge markers and 6 sentence categories. In order to capture the inter-dependence among these metrics, we also estimate the covariance matrix Σ (of size 10×10) from the same corpus. Table 2 shows the estimated corpus statistics of the proposed knowledge quality metrics.

5.2 Operating Phase

As described in Figure 1, in this phase, a given story is evaluated with respect to the knowledge quality metrics using the corpus statistics generated from the training corpus.

Evaluation of Knowledge Quality Metrics: We evaluate each knowledge quality metric m for the given CM story s as *Good*, *OK*, or *Bad* as follows. Let v_{ms} be the value of the metric m computed for the story s.

$$\textit{Good}\ (v_{ms} \geq q3_m);\ \textit{OK}\ (q3_m > v_m \geq p10_m);\\ \textit{Bad}\ (v_{ms} < p10_m)$$

Generating Suggestions for Improvement: For any of the above metrics, if a given story has a value lower than $p10_m$, a corresponding suggestion for improvement is shown to the user so that the story can be revised accordingly. For example, if Benefit_density of a story has a very low value, the corresponding suggestion would be – *Please add more details about the specific benefits achieved because of your solution.* If the metric Skills_density has a very low value, the corresponding suggestion would be – *Please mention the names of some specific tools or technologies which were employed to solve the problem.*

Aggregated Knowledge Quality Metrics: We explored the following two ways to get a single aggregate metric which captures the overall knowledge quality of a CM story by combining the individual knowledge quality metrics.

• **Distance from the mean vector** ($Dist_{mean}$): This metric is based on the mean vector ($\vec{\mu} \in \mathbb{R}^{10}$) and the co-variance matrix ($\Sigma \in \mathbb{R}^{10\times10}$) learned from the corpus of *final* accepted stories as described above. For a new story s, let $\vec{v_s}$ ($\in \mathbb{R}^{10}$) be the vector representing values of all the 10 knowledge quality metrics. Then the metric is computed as the Mahalanobis distance of $\vec{v_s}$ from $\vec{\mu}$.

$$Dist_{mean}(s) = \sqrt{(\vec{v_s} - \vec{\mu})^T \Sigma^{-1} (\vec{v_s} - \vec{\mu})} \quad (9)$$

Lower the value of $Dist_{mean}(s)$, better is the knowledge quality of s because the lower value indicates that the story s is more similar to the *ideal* stories.

• **Sum of the scaled metrics** (Z_{sum}): This metric is computed as the sum of scaled values of all the 10 knowledge quality metrics. For a new story s, let v_{ms} ($\in \mathbb{R}$) be the value of the knowledge quality metric m. This value is scaled using the mean (μ_m) and standard deviation (σ_m) of m estimated from the corpus of *final* accepted stories as described above. The metric is computed as follows:

$$Z_{sum}(s) = \sum_m \frac{v_{ms} - \mu_m}{\sigma_m} \quad (10)$$

Here, the higher values of Z_{sum} indicate better knowledge quality.

6 Narration Flow Quality Metrics

In addition to the knowledge content, it is also important to evaluate the narration quality of any narrative text such that it measures how well-structured the flow of narration is. In this section, we describe our proposed metric to evaluate the flow of different sentence categories in a CM story.

Sentence Categories Flow Metric: A good *flow* of sentence categories is that sequence of sentence categories which is generally used to describe an *ideal* story. For example, generally any story begins with some background of the problem followed by the description of the problem itself. Then the contextual knowledge of the CM is discussed followed by the proposed or implemented solution. Finally, the story concludes by discussing the benefits that were achieved by the solution and whether any appreciations were received for it. Though it is not mandatory to strictly follow this flow of narration and some sentences can be out of place, the good stories are generally structured in this way. Moreover, a good cohesive story will contain all the sentences describing a certain aspect (say Problem) in close proximity of each other and also at a proper relative position within the entire story. Hence, we propose a new metric – SCF (Sentence Categories Flow) which tries to capture these aspects of an ideal flow of sentence categories in a CM story.

First, a relative position of each sentence within the CM story is determined as follows. For any i^{th} sentence in a CM story consisting of n sentences, the relative position is $\frac{i}{n}$. For a particular sentence category (say Solution), we create a sample of relative positions of all sentences belonging to that category from all the stories in our training corpus. We compute mean (μ_{RP}) and standard deviation (σ_{RP}) of this sample (e.g., for Solution, $\mu_{RP} = 0.6$ and $\sigma_{RP} = 0.22$; this means that normally the Solution sentences occur in a story after 60% of the overall sentences are written). Now, given any new story s, the metric $SCF_{Solution}(s)$ is computed as the number of sentences of category Solution in s whose relative position is more than one standard deviation away from the mean, i.e., relative position outside the range $[\mu_{RP} - \sigma_{RP}, \mu_{RP} + \sigma_{RP}]$. Similar metrics are computed for other sentence categories in the same way (*note that μ_{RP} and σ_{RP} are specific to each sentence category*). Lower the value of this SCF metric, better is the narration flow quality, because it simply counts the number of sentences of a particular sentence category which are at *unusual* relative positions within a story. Based on this metric, suggestions for improvement are generated for those sentences in a CM story for which the relative position is outside the expected range. E.g., *Please consider re-positioning the Solution sentence [x] which is appearing too early (or late) in your story.* We also compute a single aggregate metric to combine the SCF metrics for individual sentence categories: $SCF_{all} = \sum_c SCF_c$.

7 Experimental Analysis

In this section, we describe our experiments in terms of datasets, baselines, and the evaluation strategy.

7.1 Datasets

We use the following two datasets[3] of CM stories.

- **Training corpus** (D^{train}): It is a large corpus of $53,675$ CM stories consisting of 1.4 million sentences and 28.8 million words. The median length of these CM stories is 23 sentences. This corpus contains all the *final* CM stories which have been reviewed by human reviewers and revised multiple times by the story writers (CMs) to incorporate the reviewers' suggestions. Hence, we consider D_{train} to be a set of *ideal* stories and use it to learn corpus statistics (see Table 2) of the knowledge quality metrics and flow quality metrics.
- **Evaluation dataset** (D_i^{eval}, D_f^{eval}): It consists of 67 CM stories where for each story two versions are available – (i) *initial* version ($\in D_i^{eval}$) which was written by the story writer (CM), and (ii) the corresponding *final* version ($\in D_f^{eval}$) which was prepared after a few iterations of incorporating suggestions for improvement by human reviewers. Both D_i^{eval} and D_f^{eval} consist of *paired* initial and final versions of 67 CM stories where the number of sentences are 2517 and 2010, respectively. The median lengths of these CM stories are 33 and 29 sentences for D_i^{eval} and D_f^{eval}, respectively.

7.2 Baselines

We explored 3 baseline metrics.

- **Readability Score**: We used Flesch reading-ease score (FRES) which was proposed by Flesch

[3] The datasets can not be made available publicly as they contain private and confidential information about our organization as well as its customers.

Metric	p10	q1	q2	q3	mean (μ)	st. dev. (σ)
Skills_density	0.000	0.048	0.103	0.174	0.125	0.103
Tasks_density	0.300	0.387	0.500	0.615	0.509	0.178
Roles_density	0.037	0.067	0.100	0.148	0.111	0.066
Concepts_density	0.000	0.875	1.333	1.681	1.193	0.746
Background_density	0.053	0.091	0.136	0.188	0.143	0.073
Problem_density	0.091	0.143	0.200	0.269	0.209	0.097
Expert_Knowledge_density	0.043	0.074	0.107	0.143	0.113	0.056
Solution_density	0.192	0.250	0.320	0.400	0.327	0.108
Benefit_density	0.050	0.091	0.138	0.190	0.143	0.073
Client_Appreciation_density	0.000	0.037	0.061	0.091	0.066	0.042

Table 2: Corpus statistics of the proposed knowledge quality metrics estimated from the training corpus D^{train}

(1979). It is calculated as follows:

$$FRES(s) = 206.835 - 1.015 \times \frac{\#words\ in\ s}{\#sentences\ in\ s} \\ -84.6 \times \frac{\#syllables\ in\ s}{\#words\ in\ s}$$

The higher values of FRES score indicate better readability. If any story has lower readability than a threshold, then a few longest sentences (in terms of #words) and a few longest words (in terms of #syllables) are suggested for potential simplification. For D^{train}, the mean FRES score is observed to be 40.2 with standard deviation of 8.4, so the threshold used is 31.8 (mean - st.dev.).

• **Perplexity**: It is generally used for evaluating the quality of language model (Jurafsky and Martin, 2021). Here, we borrow this metric to evaluate a specific sequence of sentence categories appearing in a CM story. A language model (using bigrams and trigrams of sentence categories) is learned over the sequences of sentence categories appearing in D^{train} and is used to compute perplexity of the sequences of sentence categories in D_i^{eval} and D_f^{eval}. Hence, a lower perplexity value indicates more similarity with the sequences of sentence categories observed in D^{train}.

• **Essay Grading** (EG): We trained the hierarchical neural network based model proposed by Zhang and Litman (2018) using their code[4] on the ASAP3 dataset[5] and evaluated on our datasets D_i^{eval} and D_f^{eval}.

7.3 Evaluation Strategy

We compute each evaluation metric (the proposed knowledge quality and narration flow quality metrics as well as the baseline metrics) for both the datasets – D_i^{eval} and D_f^{eval}. Next, for each metric, we determine whether it is consistently assigning a better score for a *final* version of a story as compared to its corresponding *initial* version. For this purpose, we use one-sided, two-samples, paired t-test to check whether the scores for *final* stories are significantly better than those of *initial* stories, using a specific metric. Here, the intuition behind this evaluation is – each story in D_i^{eval} is revised as per the suggestions of human reviewers to obtain the corresponding story in D_f^{eval}. If our metric consistently assigns a better value for a *final* version of a story as compared to its *initial* version, then it can be said that the metric is able to capture the same aspects of the story which human reviewers also think are important. Moreover, because the automatically generated suggestions for improvement are based on the same metrics, this evaluation strategy also implicitly measures the effectiveness of those suggestions.

We now describe the one-sided, two-samples, paired t-test for a metric m in detail. We compute the values of metric m for all 67 stories in D_i^{eval} as well as D_f^{eval}, so that we get two paired samples of size 67 each – S_i^{eval} and S_f^{eval}. The null and alternate hypotheses are as follows:

H_0: Mean of S_i^{eval} = Mean of S_f^{eval}

H_1: Mean of S_i^{eval} < Mean of S_f^{eval} (*if the metric m is such that higher values indicate better quality*); **OR**

H_1: Mean of S_i^{eval} > Mean of S_f^{eval} (*if the metric m is such that lower values indicate better quality*)

7.4 Analysis of Results

Table 3 shows the evaluation results for – (i) our proposed aggregated knowledge quality metrics (D_{mean} and Z_{sum}) and the flow quality metric (SCF_{all}), and (ii) the baseline metrics ($FRES$,

[4] https://github.com/Rokeer/co-attention

[5] https://www.kaggle.com/c/asap-aes

Metric	Mean(S_i^{eval})	Mean(S_f^{eval})	p-value
$Dist_{mean} \downarrow$	3.064	2.544	**0.00001**
$Z_{sum} \uparrow$	-0.053	0.089	**0.00043**
$SCF_{all} \downarrow$	10.443	8.015	**0.02974**
$FRES \uparrow$	36.147	35.111	0.89956
$Perplexity \downarrow$	6.486	6.178	0.08759
$EG \uparrow$	0.654	0.613	0.98258

Table 3: Evaluation results for aggregated knowledge quality metrics and narration flow quality metrics using the evaluation datasets D_i^{eval} and D_f^{eval}. (Arrows besides a metric indicate its nature - ↑ indicates higher the better and ↓ indicates lower the better; **Bold** p-values indicate the statistically significant result with $\alpha = 0.05$)

$Perplexity$, and EG). The aggregated metrics D_{mean} and Z_{sum} capture the combined effect of the proposed 10 knowledge quality metrics and both these metrics are showing statistically significant difference between S_i^{eval} and S_f^{eval}. Another proposed metric SCF_{all} for evaluating the sentence categories flow quality is also showing a statistically significant difference between S_i^{eval} and S_f^{eval}. However, for the baseline metric $Perplexity$, no statistically significant difference is observed at $\alpha = 0.05$. The other two baseline metrics $FRES$ and EG, assign better scores for initial versions as compared to the final versions, which is against our expectation that final versions should be relatively better than the corresponding initial versions.

$FRES$ is designed to measure *ease of reading* and although it is an important aspect of a narrative text, in case of CM stories, more emphasis is given to produce knowledge-rich text. Such knowledge-dense documents may become little less readable which can be observed in our experiments where the average readability of the final CM stories is little less than the initial versions. Similarly, EG is assigning higher scores for initial versions of the CM stories as compared to the final versions. This shows that the essay grading techniques give more importance to other aspects than those measuring the knowledge and flow quality in non-fiction documents like CM stories. For computing $Perplexity$, we are considering bigrams and trigrams of sentence categories. Hence, it tends to focus on small local window (of 2-3 sentences) and may not capture overall order of sentence categories in an entire CM story. On the other hand, our proposed metric SCF is able to evaluate flow of sentence categories in a better way as it is not limited within a small local window of sentences. Rather, it focuses on identifying sentences whose relative placement in a CM story is quite unusual.

7.5 Deployment

The system based on the proposed techniques is deployed for evaluating CM stories as well as for automatically generating suggestions for improvement. The initial feedback of the system is positive and we are planning to conduct detailed user-studies as a future work.

8 Conclusions and Future Work

We proposed a set of novel evaluation metrics for depth and flow of knowledge in non-fiction narrative texts that are unsupervised as well as interpretable. We focused on a specific type of documents identified as CM stories. Two different types of evaluation metrics were proposed: (i) for measuring the quality of the knowledge contents in a CM story, and (ii) for evaluating flow of different categories of sentences in a CM story. We demonstrated the effectiveness of the proposed metrics as compared to the existing metrics like perplexity, readability, and essay grading.

In future, we plan to explore how the proposed metrics can be adapted to other types of non-fiction narrative texts such as security incident reports. One interesting research direction is whether we can discover the key sentence categories automatically for a new type of documents. We also plan to develop some new narration flow quality metrics such as a metric based on sequence entropy.

9 Acknowledgements

We would like to acknowledge the support of Haridas Menon and Karthik Parvathi from TCS HR for this work. We would also like to acknowledge the contributions of TCS' Technical Communication Group for helping us identify a proper set of sentence categories and also for providing an initial set of annotated sentences.

References

2014. Understanding the social context of fatal road traffic collisions among young people: a qualitative analysis of narrative text in coroners' records. *BMC Public Health*, 14.

Mohammad Ehsan Basiri, Shahla Nemati, Moloud Abdar, Erik Cambria, and U Rajendra Acharya. 2021. Abcdm: An attention-based bidirectional cnn-rnn deep model for sentiment analysis. *Future Generation Computer Systems*, 115:279–294.

Terry L. Bunn, Svetla Slavova, and Laura Hall. 2008. Narrative text analysis of kentucky tractor fatality reports. *Accident Analysis and Prevention*, 40(2):419–425.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Rudolf Flesch. 1979. How to write plain english. *University of Canterbury. Available at http://www. mang. canterbury. ac. nz/writing_guide/writing/flesch. shtml.[Retrieved 5 February 2016]*.

Dan Jurafsky and James H Martin. 2021. Speech and language processing (3rd edition). `https://web.stanford.edu/~jurafsky/slp3/`.

Zixuan Ke and Vincent Ng. 2019. Automated essay scoring: A survey of the state of the art. In *IJCAI*, volume 19, pages 6300–6308.

Kirsten McKenzie, Deborah Anne Scott, Margaret Ann Campbell, and Roderick John McClure. 2010. The use of narrative text for injury surveillance research: A systematic review. *Accident Analysis and Prevention*, 42(2):354–363.

Roberto Navigli and Paola Velardi. 2004. Learning domain ontologies from document warehouses and dedicated web sites. *Computational Linguistics*, 30(2):151–179.

Sachin Pawar, Girish Palshikar, and Anindita Sinha Banerjee. 2021. Weakly supervised extraction of tasks from text. In *Proceedings of the 18th International Conference on Natural Language Processing (ICON)*.

Sachin Pawar, Rajiv Srivastava, and Girish Keshav Palshikar. 2012. Automatic gazette creation for named entity recognition and application to resume processing. In *5th ACM COMPUTE Conference: Intelligent & scalable system technologies*, pages 1–7.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Gary S. Sorock, Thomas A. Ranney, and Mark R. Lehto. 1996. Motor vehicle crashes in roadway construction workzones: An analysis using narrative text from insurance claims. *Accident Analysis and Prevention*, 28(1):131–138.

Haoran Zhang and Diane Litman. 2018. Co-attention based neural network for source-dependent essay scoring. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 399–409.