Measuring Contribution in Collaborative Writing: An Adaptive NMF Topic Modelling Approach

Johnny Torres, Alberto Jimenez, Sixto García, Enrique Peláez, Xavier Ochoa

Escuela Superior Politécnica del Litoral (ESPOL)
Centro de Tecnologías de Información (CTI)
Campus Gustavo Galindo Km 30.5 Vía Perimetral, P.O. Box 09-01-5863, Guayaquil, Ecuador
{johnny.torres, alberto.jimenez, sgarcia, epelaez, xavier}@cti.espol.edu.ec

***Abstract*—In universities worldwide, instructors may spend a significant amount of time reviewing homework and group projects submitted by their students. Web-based technologies, like Google Docs, have provided a platform for students to write documents collaboratively. Currently, those platforms provide limited information on the individual contribution made by each student. Previous studies have focused on the quantitative aspects of individuals' contribution in collaborative writing, while the quality aspect has received less attention. In this paper, we propose a new model to measure not only quantitative input but also the quality of the content that has been contributed to a document written collaboratively in Spanish language. Based on topics-modeling techniques, we use an adaptive non-negative matrix factorization (NMF) model to extract topics from the content of the document, and grade higher students making those contributions. Using Google documents submitted by students to the academic system of our university as part of their projects, experimental results show that compared to other baseline methods such as edits or words count, our model provide a better approximation to the scores given by human reviewers. Therefore, our model can be used as part of an automatic grading subsystem within the academic system, to provide a baseline score of students' contribution in collaborative documents. This will allow instructors to reduce their workload associated with revision and grading of documents and focus their time on more relevant tasks.**

***Index Terms*—Education technology; collaborative writing; topic modeling.**

I. Introduction

Introducing new learning paradigms in university courses is always challenging. It requires that instructors adopt a different teaching methodology, while students likewise need to learn a new approach to work. Active learning methodologies have been popular in educational models [1]. These methodologies encourage students to engage in activities, such as reading, discussion, problem solving, and writing, to motivate analysis of the class content. In the context of active learning, collaboration has been identified as an important component [2].

Collaborative work can enhance the learning process inside as well as outside the classroom [3]. It enables students to develop skills that could not be acquired working alone, such as, critical thinking and peer discussion. This methodology often leads students to complete their assignments more effectively and with higher quality [3].

978-1-5090-4830-4/17/$31.00 ©️ 2017 IEEE

One of the tasks in a collaborative environment is the creation of a document on which two or more students contribute to its content. Online platforms, (e.g. Google Docs, Office 365, or Wikipedia), have enabled students to write documents collaboratively. These platforms have solved previous technological issues that occur when working with documents, like software compatibility or sharing documents over the internet, e.g. by email. This sort of collaborative writing requires coordination and awareness of the contribution of each group member. The level of contribution in collaborative writing can be influential, although not definitive, in determining the final grade of each student.

Previous studies have proposed methodologies to tackle the problem of measuring individual contribution in collaborative documents. The approaches proposed by those studies focus on determining the authorship at level of words [4], [5], or sentences [6]. After establishing the authorship, these methods calculate the overall contribution of each author based on the sum of the number of words or sentences.

In this paper, we seek to tackle the problem of measuring the overall contribution by students to the content of a document done in a collaborative manner. This task has been considered as a cognitive process characterized by a high level of complexity [7]. Thus, we aim to provide a better approximation of each student's contribution in a collaborative document, compared to the students' relative scores given by a professor. The relative score means that if the instructor gives an overall score, let's say of $8/10$ to a document, we measure to what extent each student has contributed to that score. Therefore, it is not the objective of this paper to give the final grade of a document, but to determine the students' relative contributions.

This paper improves the state-of-the-art by making two contributions. First, we analyze how the existing measurement techniques evaluate the quantity, as well as the quality of contributions in collaborative documents. Second, we propose a novel measurement model that takes into account the quality, in addition to the quantity of the contributions. To the best of our knowledge, this paper is the first to propose measurements of the quality of contributions to the content of collaborative documents in an educational environment.

The rest of this paper is organized as follows: Section II presents previous work done to measure contributions on platforms that support collaborative writing. The proposed model is described in section III. Section IV explains the dataset used in our experiments. In Section V, we describe the results from the experiments. Finally, in section VI, we draw the conclusions and outline future lines of research.

II. Related Work

Several methods have been proposed to measure the users' contribution in collaborative writing. Most previous research has focused on Wikipedia or Git[1] due to their open source nature, whereas few studies are based on Google Docs or Office 365. Nonetheless, all these systems keep track of each revision (e.g. edit or commit action) in the documents. Therefore, the same methods can be applied to all of them to measure contributions.

In the context of software development with source code management systems such as Apache Subversion [8] or Git, contribution measurement has been analyzed in terms of code ownership. The measurement of quality of source code is still based on line-level tracking [9]. This coarse-grained level tracking is also used as a basic unit to identify contributors. In this method, line-level tracking allows identification of the user who made the last change in a specific line of code in a document, but the information about the original creator is lost. This functionality is appropriate for collaborative software development environments, because it allows detection of the user responsible for introducing defects or making changes in the code. However, this mechanism is not suitable for tracing the original contributors of the content or to detect changes at a more fine-grained level such as words or characters.

In the context of Wikipedia, several approaches have been proposed to measure users' contributions. Viegas et al. [10] attributes the content of a sentence to the user who made the last change. However, it does not recognize correctly the author of the content reintroduced after it has been deleted. Another drawback, of working at sentence level is the fact that small changes (e.g. adding words, fixing grammar, or formatting) are not tracked. Thus, this method does not correctly measure the overall contribution of users.

Ding et al. [11] developed a visualization of enterprise wikis at large scale, measuring users' contributions based on the number of edits done by each user to different pages. Hess et al. [12] has also proposed a method to measure the extent of a user's partial contribution in each version of the document. It compares the current version to the previous one, and the overall contribution is the sum of all partial contributions. Based on a similar approach, Sabel at al. [13], calculated the contributions of users to each version of the document, and then used the results as the users' rating in a reputation system.

Korfiatis et al. [7] estimated the contributions of the users to Wikipedia, based on social network analysis, specifically using metrics such as users' centrality as a proxy for their reputation. Hoisl et al. [14] implemented an add-on for wiki platforms that allows measurement of partial contribution of users based on differences between versions of the document. Then, it assigns a weight to each partial contribution based on a metric of importance (e.g., number of views or votes).

These previous studies have used different approaches to solve the problem of measuring contributions in collaborative writing. However, these approaches have several limitations: The basic measurement method is based on a simple count of edits in a document [11], therefore, it can be easily manipulated by users seeking to increase their contribution score. Approaches using social network analysis [7] are good at calculating the distribution of users' contributions across multiple wiki documents, but do not provide a good estimation of contribution level to a specific document.

Previous methods do not consider the difference between contributions that remain on the document over long periods of time, and those contributions that are quickly deleted [12], [13]. For example, a contribution that is quickly deleted receives the same score compared to a contribution that remained in a document for a long time. The higher quality contributions are likely to remain over time; therefore, algorithms should consider the duration of a contribution. In the case of Google docs, although the time span of contribution is not as large as Wiki pages, the same principle should be considered.

Additionally, algorithms to measure contribution should verify the type of contributions they are capturing (e.g. new content, formatting changes), and estimate the extent to which that type of contribution represents the users' overall contribution. Pfeil et al. [15] proposed a methodology to close this gap by utilizing a grounded theory to categorize contribution types in wikis, which was later used by Ehmann et al. [16]. This categorization takes into account contributions beyond adding new content, e.g., formatting existing information, grammatical corrections, deleting irrelevant information, or even clarifying Information.

Based on [15], [16], Arazy et al. [6] proposed a simpler categorization, by considering specific types of contribution that can be detected by algorithms, such as adding new content, formatting, internal or external linking, deleting content, and proofreading. In their algorithm, the basic unit of meaning is a sentence. This is based on the notion of sentence ownership, in which users own the sentences they created. Ownership of a sentence continues if more than 50% of the words are the same between two consecutive revisions of the document. The metrics obtained represent the number of sentences that have been added or deleted by a user. Additionally, the algorithm considers the number of links, as well as word-level changes.

Adler et al. [17] proposed *Wikitrust*, a visual tagging of trusted and untrusted sections in a wiki document. The algorithm finds the longest coincidences for all sequences of words between two consecutive revisions, but deleted word-chunks. Therefore, it can detect reintroduced words and track the authorship of the content, since performing rollbacks to previous versions is common practice in Wikipedia. However, the algorithm uses a greedy approach, and usually falls in local optima. This can lead to misinterpretations of the authorship of the words, especially when word sequences are moved instead

[1] https://git-scm.com/

of being deleted or inserted.

De Alfaro et al. [18] introduced an algorithm to measure users' contributions that requires comparing each new revision with the entire set of past revisions, considering that content can be deleted and later re-inserted. For each section of content in the last revision, the algorithm searches all previous versions if there are similar sections of content that are statistically unlikely to happen naturally, thus establishing proper authorship. Using these matches, it determines the earliest authorship of each token (e.g., words) in the new content. The algorithm's run-time is relative to the aggregation of the size of the last revision, and the total number of edits in the revision history. However, the algorithm was not tested in terms of precision.

Flöck et al. [19] presented a tree model approach to establish content authorship, and therefore measure users' contribution in a document. In this initial version, the model only considered paragraphs and sentences. This limitation produced a precision of approximately 60%, which is unsuitable for usage. An improved algorithm [4], based on a k-partite graph model, builds a fine-grained level representation of the document by considering paragraphs, sentences, and tokens. This model was more efficient and reported a 30% increase in precision.

III. MODEL

We propose an adaptive NMF-based model to measure the level of relative contribution of students in collaborative writing, considering both quantity and quality. The model consists of three steps as depicted in figure 1. First, the model establishes the authorship of each word in the document. Then, the model extracts from the document the topics and their associated words, using an adaptive NMF topic modeling. Finally, it calculates the level of contribution based on the number of word and topics contributed by each student.

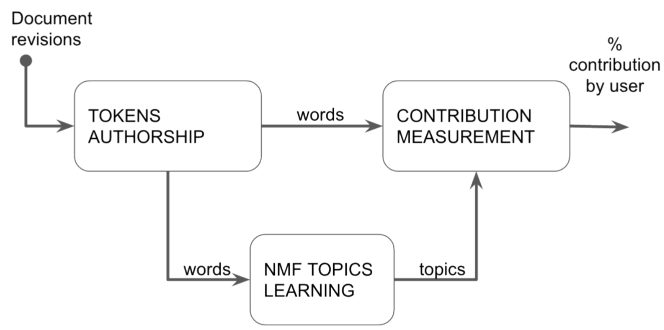

Fig. 1: Model overview.

A. Tokens authorship

In order to establish the authorship of a document's words, we extend the graph-based model proposed by Flock et al. [4]. Figure 2 illustrates an example of revisions in a collaborative document. In this model, the revisions are represented by nodes r, paragraphs by p, sentences by s, and tokens by t. The links between nodes define a container relation, and the labels over the links define the relative position of the nodes. For example, the token t_0 and t_1 are in first and second position of the sentence s_0. The second revision, r_1, has two paragraphs. Paragraph p_1 reuse sentence s_0, created in revision r_0, followed by the new sentence s_1. The third revision, $r2$, reuses paragraph p_2 from the earlier version and adds two additional paragraphs, p_3 and p_4. In another case, p_3 adds a new sentence s_3, which reuses token t_2.

Formally, the authorship model is represented by a *k*-partite graph, with $k = 4$ and $G = (V, E, \phi, \mathbb{N})$. The graph is defined as follows:

1) The set of vertices V in G consist of four pairwise disjoint subsets R, P, S, T, i.e., $V = R \cup P \cup S \cup T$. These subsets represent revisions, paragraphs, sentences, and tokens (words, characters, etc), respectively.
2) The set of links E in G is partitioned into $k-1$ cuts: $E = \langle R, P\rangle \cup \langle P, S\rangle \cup \langle S, T\rangle$. The links denote a *containment* relationship, e.g., if $p_i \in P$, $s_i \in S$, and $(p, s) \in E$, therefore paragraph p_i contains sentences s_i.
3) The relative position of a token t_i is denoted by a label mapping $\phi : E \rightarrow \mathbb{N}$ over the links in graph G. In addition, a label is used to maintain the sequence of the revisions nodes, e.g., $label(r_i) < label(r_j)$ indicates node r_i is the predecessor of node r_j.

This graph-based authorship model was implemented specifically for Wikipedia pages, but it can be generalized to Google documents or other collaborative writing systems that use a versioning control. We adapt the graph model for Google documents by mapping a document version dv to a revision node r, i.e., $DV \rightarrow R$. The rest of the hierarchy remains identical, with minor modifications, such as the addition of new properties in the nodes.

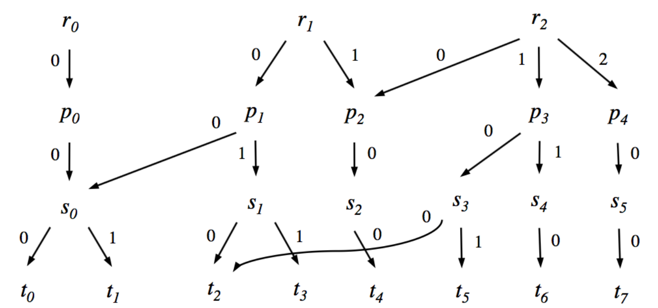

Fig. 2: Graph-based model to establish tokens' authorship in collaborative documents [4].

After the algorithm has run successfully, it labels the leaf nodes, i.e., the tokens t_i with their respective author.

As part of the optimization of the graph model, we use a word-level tokenization. A more fine-grained tokenization, e.g. to character-level, would be superfluous to the aim of measuring quality contribution, because formatting changes are irrelevant to that end. Moreover, determining the authorship of fine-grained tokens would be a more difficult task and the time complexity would increase. Thus, we do not consider formatting changes in our model.

Another consideration is detection of large changes in the document, which are labeled as vandalism in wiki pages. We maintain this functionality in our model because it allows us to detect students who are probably copying and pasting large chunks of information from external sources (e.g., web pages, pdf papers) into the document.

B. Topics learning

We associate authorship of main ideas with a high level of contribution to the document. In order to infer the main ideas, we apply topic modeling. To extract the topics t_i from the document's content, we use an adaptive non-negative matrix factorization (NMF). Figure 3 shows the general intuition for topic modeling using NMF.

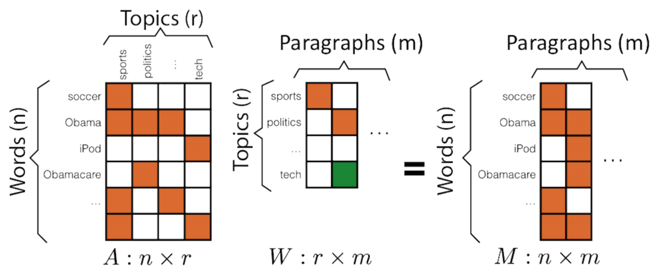

Fig. 3: NMF topic modeling.

We choose NMF-based topic modeling because it yields better results as shown in section V. The process to extract the topics from the document is described next.

First, we build the Documents-Terms matrix M using the tokens t_i generated in the authorship model. In the matrix M, the rows represent the vectors of features N and the columns represent the collection of documents D. However, we only have one input document, therefore, we adapted the model to consider each paragraph as a document to build D.

$$D = \{p_1, p_2, \dots, p_n\} \qquad \forall p_i \in P \tag{1}$$

In this configuration, each paragraph p_i is mapped to a document d_i. We enforce a minimum number of words per paragraph (wpp), that is $wpp >= 50$. If that condition is not met, several paragraphs are concatenated. The features vectors N is composed of the tokens (words), excluding stops words and consider only the stem of each word.

$$N = \{t_1, t_2, \dots, t_n\} \qquad \forall t_i \in T \tag{2}$$

In each feature vector N, we include all tokens, but only those that appear in the document p_i will have a value representing the frequency. Finally, to use TF-IDF statistic to weight how important is a word to the document collection.

Using the Documents-Terms matrix M, NMF decomposes the input matrix M into two non-negative matrices W and A, whose product approximates the non-negative matrix M. This is done by optimizing an extension of the Euclidean norm [20], known as the squared Frobenius norm:

$$\arg\min_{W,H} \frac{1}{2}||M - WA||^2_{Fro} = \frac{1}{2}\sum_{i,j}(M_{ij} - WA_{ij})^2 \tag{3}$$

NMF modeling is efficient for representing images and text, because it is based on additive models. It has been seen in [20] that, when appropriately constrained, NMF can deliver interpretable models of the data.

NMF implements the technique Non-negative Double Singular Value Decomposition (NNDSVD). NNDSVD depends on two SVD forms, one approximating the data matrix, the other approximating positive segments of the subsequent partial SVD variables using a mathematical property of unit rank matrices. The essential NNDSVD calculation is a better fit for sparse factorization, which is the case of text.

For the implementation, we use the library implemented by Pedregosa et al. [21]. In relation to the parameters used in NMF, the initialization method used has a direct impact on the performance [22]. In our experiments, we initialized using *random state* to control reproducibility.

With a specific end goal to regularize the NMF model, $L1$ and $L2$ priors can be added to the loss function. The L1 prior uses an elementwise $L1$ standard, while the $L2$ prior uses the Frobenius norm. The blend of $L1$ and $L2$ is controlled with the $l1_ratio$ (ρ) parameter, and the intensity of the regularization with the alpha (α) parameter. At that point the priors terms are:

$$\alpha\rho||W||_1 + \alpha\rho||A||_1 + \frac{\alpha(1-\rho)}{2}||W||^2_{Fro} + \frac{\alpha(1-\rho)}{2}||A||^2_{Fro} \tag{4}$$

with the regularized objective function defined as:

$$\frac{1}{2}||M - WA||^2_{Fro} + \alpha\rho||W||_1 + \alpha\rho||A||_1 + \frac{\alpha(1-\rho)}{2}||W||^2_{Fro} + \frac{\alpha(1-\rho)}{2}||A||^2_{Fro} \tag{5}$$

The number of topics is specified by the parameter k. For those k topics, we can extract the top n relevant words associated, which are used to measure the quality of contributions.

$$TW_i = \{tw_1, tw_2, \dots, tw_n\} \qquad \forall i < k \tag{6}$$

C. Contribution measurement

In order to measure the overall contribution of students in collaborative documents, our model considers both quantitative and qualitative metrics. The quantitative measurement is based on the number of words each student contributed to the document. The qualitative measurement considers to what extent the student contributes to the topics in the document. The contribution of a student is defined as follows:

$$C_s = \beta * Tc_s + (1-\beta)Wc_s \tag{7}$$

Where Tc_s, refers to the number of words associated to topics that are part of the collection of nodes that belong to each student:

$$Tc_s = \sum_{i=1}^{n} tw_i \qquad \forall tw \in G(T) \tag{8}$$

The quantitative measurement is a simple count of the words contributed by each student to the document.

$$Wc_s = \sum_{i=1}^{n} t_i \qquad \forall t \in G(T) \tag{9}$$

IV. Dataset

The dataset used consists of 39 Google documents with a total of 763 revisions, made by 92 students. The links of these documents were submitted to the academic system of our university by students as part of their homework or group projects. Using the links of the documents, we collected their revision history from Google Drive storage utilizing the API[2]. Figure 4 shows the distribution of the number of revisions by document, where, only a few documents have a significant number of revisions.

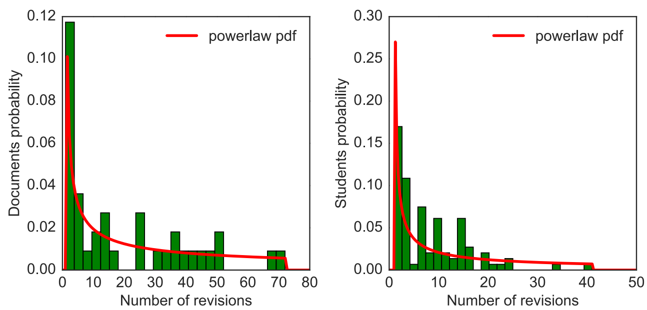

Fig. 4: Histogram of revisions by documents and students.

Also, figure 4 shows that most of the students have a small number of revisions in the documents. In both cases, the variables follow a power law distribution, as in most human activities. It was observed that the average number of students collaborating in the documents is 3 students, which is the usual number of students per project is most courses.

In figure 5, we observe that documents analyzed spans from May-2013 to January-2016. There are spikes in specific months (July, January) that concur with the end of semester, where students are busy writing reports.

We found that most students work in the documents in the afternoon. Figure 6 show the temporal distribution of the creation date and time of the revisions.

[2]https://developers.google.com/drive/v2/reference/

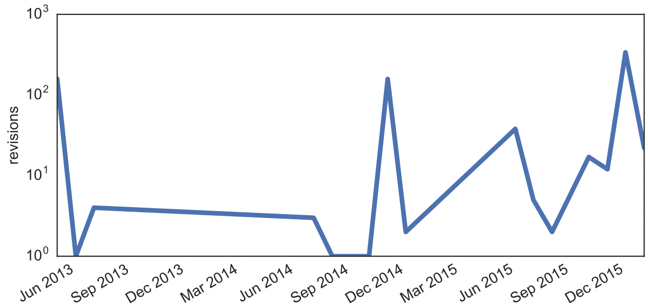

Fig. 5: Time series of revisions in documents.

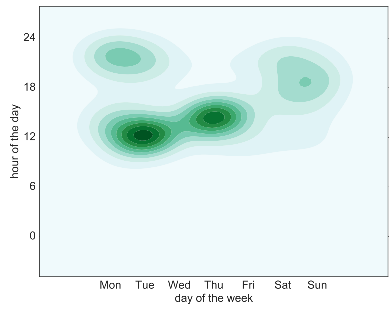

Fig. 6: Temporal distribution of documents.

In addition, we observed that there is a negative correlation between the number of words and lexical diversity in the content of the documents' revisions. High lexical diversity is related to initial revisions of the documents, whereas lower diversity is mostly related to revisions with a large number of words in the final stages prior to submission for grading.

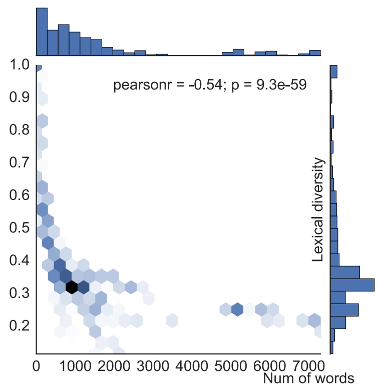

Fig. 7: Correlation between number of words and lexical diversity in documents.

A. Manual evaluations of documents

We established the ground truth of the students' contributions in the documents through manual evaluations. Because we are measuring the relative quality of the contributions, we do not use the grading of the documents given by the professors. Instead, two research assistants analyzed each document following strict guidelines to measure the quality of students' contribution. To accomplish this, the reviewers used AuthorViz tool [23]. This tool allows to visualize the contribution of each students, by highlighting their words contributed with different colors in the final version of the document. Then, the reviewers determined the percentage of contribution of students for each paragraph in the document. The assumption is that each paragraph contains a main idea, expressed in one or more sentences. Thus, the students that owns those sentences were given a higher level of contribution on a 1-5 scale. Finally, the relative contribution of each student is the average of the paragraphs scores.

V. Experiments

In order to evaluate the precision of our model in measuring the contribution of the students, we compare it against other baseline methods. For all methods, the first step was to establish the authorship of each word in the documents. Then, we apply the baseline methods, as well as our model to measure the level of students' contributions. In the next subsection, we describe the baseline methods, evaluation metric, and results obtained.

A. Baseline methods

1) Edits Count (EC): is a count of the number of students' edits in a document [11].

2) Words Count (WC): is a count of the number of words of each student contributed in a document [4].

3) Words Count Keywords (WCK): is similar to the WC method, but excluding *stop words* and extracting the stem of the remaining words.

4) Graph Similarity (GS): consists of representing each document as a graph [24]. In the document's graph, each word represents a node, and the links are established between consecutive words. Then, we create a graph for each student based on the words of his authorship. The student's contribution is the maximum common sub-graph between the document and student graphs. The graph representation captures the document's structure, but the drawback is that isolated nodes in students' graphs are not summed up in their overall contribution.

5) Graph Similarity Keywords (GSK): is similar to the GS method, and it performs data cleansing as in the WCK metric.

6) Sentences Count (SC): is a count of the number of sentences that each user owns in the document. The sentences ownership is based on the model proposed by Arazy et al. [6].

7) Topics NMF (NMF): is our proposed model using NMF to extract topics from the document.

8) Topics LDA (LDA): extends our model, but it uses latent Dirichlet allocation (LDA) instead of NMF to extract topics from documents.

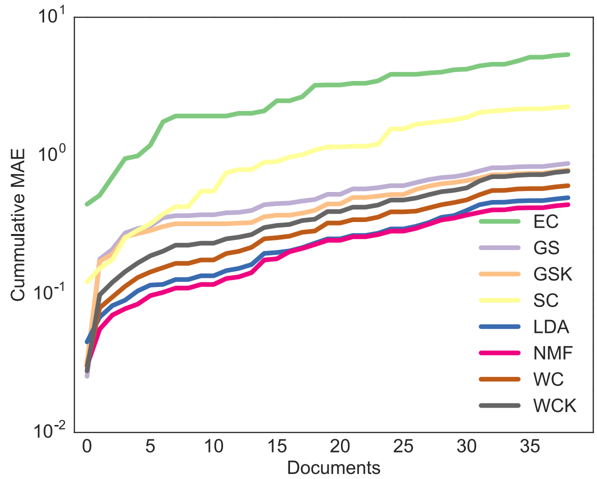

Fig. 8: Comparative analysis of precision.

B. Evaluation metric

For each document, we calculate the mean absolute error (MAE) given by

$$\mathrm{MAE}_{\mathrm{doc}} = \frac{1}{n}\sum_{i=1}^{n} |f_i - y_i| \,. \tag{10}$$

Where f_i is the calculated user's contribution and y_i is our ground truth value determined manually by human reviewers. It is important to note that we do not use MAPE because all contributions scores are in the same scale (percentages).

C. Results

We compare the methods based on the cumulative MAE for students' contribution in each document. Figure 8 shows that topics-based models (NMF, LDA) outperforms all others methods, of which NMF has lower MAE. The third lower MAE correspond to WC method, while EC is the worst.

In our proposed model, the hyper-parameter beta (β) has been evaluated to determine what value produce better precision. Figure 9 shows the cumulative MAE for different values of β in the model NMF. Higher values of β produce lower error, and we found the best setting is 0.9.

Figure 10 shows the MAE of students' contribution by methods. Our topics-based models are slightly better than other methods at calculating the contribution. NMF outperforms the WC metric by up to 3% in some documents. On average, it is approximately 1% better that WC, which has the lower MAE for simple count methods.

In relation to scalability, topics-based models are slower compared to the simple count-based baseline methods. This is due to the algorithmic complexity of NMF and LDA models. The time complexity is polynomial in NMF [25]. In LDA, the time complexity is proportional to the number of samples and iterations [26]. Figure 11 shows clearly that there are three

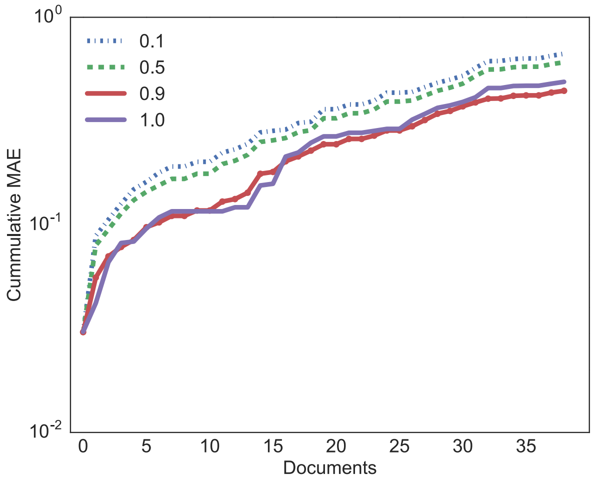

Fig. 9: Analysis of hyper-parameter β.

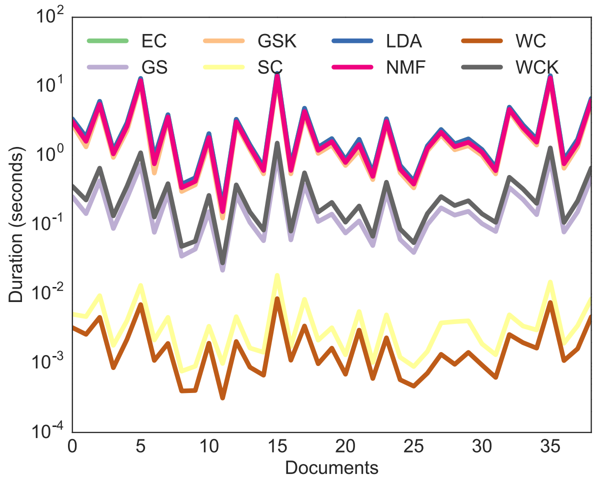

Fig. 11: Comparison of duration of algorithms.

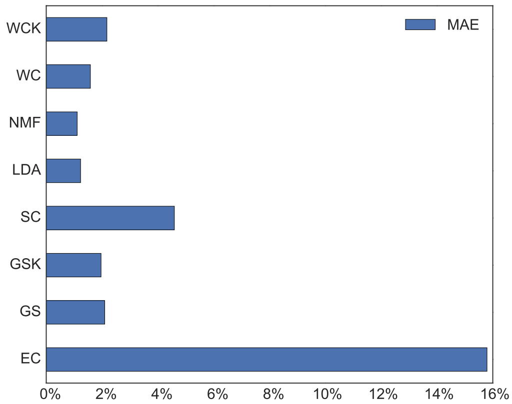

Fig. 10: MAPE comparison of algorithms.

students' participation.

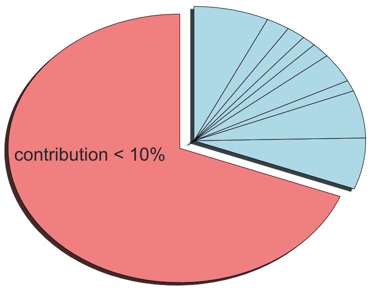

Fig. 12: Level of students' contribution in documents.

groups of algorithms according to the time complexity. The slower and more complex methods, in the upper section of the graph, are NMF, LDA, GSK, and EC. In the case of EC, although is a simple count method, it must iterate all revisions of the documents, which is a time-consuming process. The next group, regarding the time complexity, includes GS, and WCK. And, in the lower section of the graph, the fastest algorithms are simple count methods WC and SC.

Although topic-based models are slower than baseline methods, the duration depends on the size of the document and the number of revisions. Thus, the waiting time for applications using our model will be relative to these two factors [4]. In our dataset, for documents with size of $7K$ words, the duration ranges from 10 and 13 seconds.

In other aspects of the results, our model allows us to detect those students making a low contribution to the documents. Figure 12 shows that 69% of students contributed 10% or less to the documents. This insight could be used to improve

VI. Conclusions

In this work, we have evaluated documents produced in a collaborative manner by students. Current approaches focus on the quantitative aspect of the individual contributions, but this is not the only factor that instructors assess in collaborative writing. The qualitative aspect is important, because some students can write less than their peers but they may contribute with more meaningful content. In addition to the quantitative aspect, we propose a model to measure content quality contributed by students. We have shown that extracting topics embedded in paragraphs and giving a higher rating to the students that contribute with those topics yields better results. Experimental results show that our model has a better approximation to the score given by human reviewers.

We plan to do further research regarding methods to improve quality measurement through a better understanding of the documents' content, and how changes to it affect the topics extracted. By understanding the content of document and collaboration patterns, we can provide a better estimation of the effort done by students and the quality of the documents. Additionally, we will investigate visualizations of students' contribution in collaborative writing. This will complement the work done in this paper, and will provide a useful tool for instructors to grade students' projects.

REFERENCES

[1] C. C. Bonwell and J. A. Eison, *Active Learning: Creating Excitement in the Classroom. 1991 ASHE-ERIC Higher Education Reports.* ERIC, 1991.

[2] A. L. Kieser and F. O. Golden, "Using online office applications: Collaboration tools for learning," *Distance Learning*, vol. 6, no. 1, p. 41, 2009.

[3] W. Zhou, E. Simpson, and D. P. Domizi, "Google docs in an out-of-class collaborative writing activity." *International Journal of Teaching and Learning in Higher Education*, vol. 24, no. 3, pp. 359–375, 2012.

[4] F. Flöck and M. Acosta, "Wikiwho: Precise and efficient attribution of authorship of revisioned content," in *Proceedings of the 23rd international conference on World wide web*. ACM, 2014, pp. 843–854.

[5] D. Wang, J. S. Olson, J. Zhang, T. Nguyen, and G. M. Olson, "Docuviz: visualizing collaborative writing," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2015, pp. 1865–1874.

[6] O. Arazy and E. Stroulia, "A utility for estimating the relative contributions of wiki authors." in *ICWSM*, 2009.

[7] M.-A. Sicilia, N. T. Korfiatis, M. Poulos, and G. Bokos, "Evaluating authoritative sources using social networks: an insight from wikipedia," *Online Information Review*, vol. 30, no. 3, pp. 252–262, 2006.

[8] C. M. Pilato, B. Collins-Sussman, and B. W. Fitzpatrick, *Version control with subversion*. " O'Reilly Media, Inc.", 2008.

[9] R. Baggen, J. P. Correia, K. Schill, and J. Visser, "Standardized code quality benchmarking for improving software maintainability," *Software Quality Journal*, vol. 20, no. 2, pp. 287–307, 2012.

[10] F. B. Viégas, M. Wattenberg, and K. Dave, "Studying cooperation and conflict between authors with history flow visualizations," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2004, pp. 575–582.

[11] X. Ding, C. Danis, T. Erickson, and W. A. Kellogg, "Visualizing an enterprise wiki," in *CHI'07 extended abstracts on Human factors in computing systems*. ACM, 2007, pp. 2189–2194.

[12] M. Hess, B. Kerr, and L. Rickards, "Wiki user statistics for regulating behaviour," Citeseer, Tech. Rep., 2006.

[13] M. Sabel, "Structuring wiki revision history," in *Proceedings of the 2007 international symposium on Wikis*. ACM, 2007, pp. 125–130.

[14] B. Hoisl, W. Aigner, and S. Miksch, "Social rewarding in wiki systems–motivating the community," in *International Conference on Online Communities and Social Computing*. Springer, 2007, pp. 362–371.

[15] U. Pfeil, P. Zaphiris, and C. S. Ang, "Cultural differences in collaborative authoring of wikipedia," *Journal of Computer-Mediated Communication*, vol. 12, no. 1, pp. 88–113, 2006.

[16] K. Ehmann, A. Large, and J. Beheshti, "Collaboration in context: Comparing article evolution among subject disciplines in wikipedia," *First Monday*, vol. 13, no. 10, 2008.

[17] B. T. Adler, L. de Alfaro, I. Pye, and V. Raman, "Measuring author contributions to the wikipedia," in *Proceedings of the 4th International Symposium on Wikis*. ACM, 2008, p. 15.

[18] L. De Alfaro and M. Shavlovsky, "Attributing authorship of revisioned content," in *Proceedings of the 22nd international conference on World Wide Web*. ACM, 2013, pp. 343–354.

[19] F. Flöck and A. Rodchenko, "Whose article is it anyway?–detecting authorship distribution in wikipedia articles over time with wikigini," *Online proceedings of the Wikipedia Academy*, 2012.

[20] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of machine learning research*, vol. 5, no. Nov, pp. 1457–1469, 2004.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[22] C. Boutsidis and E. Gallopoulos, "Svd based initialization: A head start for nonnegative matrix factorization," *Pattern Recognition*, vol. 41, no. 4, pp. 1350–1362, 2008.

[23] D. Wang, "How people write together now: Exploring and supporting today's computer-supported collaborative writing," in *Proceedings of the 19th ACM Conference on Computer Supported Cooperative Work and Social Computing Companion*. ACM, 2016, pp. 175–179.

[24] A. Schenker, A. Kandel, H. Bunke, and M. Last, *Graph-theoretic techniques for web content mining*. World Scientific, 2005, vol. 62.

[25] A. Cichocki and P. Anh-Huy, "Fast local algorithms for large scale nonnegative matrix and tensor factorizations," *IEICE transactions on fundamentals of electronics, communications and computer sciences*, vol. 92, no. 3, pp. 708–721, 2009.

[26] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.