Comparing Supervised Machine Learning Techniques for Genre Analysis in Software Engineering Research Articles

Felipe A. de Britto[1], **Thiago C. Ferreira**[2], **Leonardo P. Nunes**[2], **and Fernando S. Parreiras**[1]
FUMEC University[1]
Federal University of Minas Gerais[2]
felipebritto@gmail.com, thiagocf05@ufmg.br,
leopereiranunes@ufmg.br, fernando.parreiras@fumec.br

Abstract

Written communication is of utmost importance to the progress of scientific research. The speed of such development, however, may be affected by the scarcity of reviewers to referee the quality of research articles. In this context, automatic approaches that are able to query linguistic segments in written contributions by detecting the presence or absence of common rhetorical patterns have become a necessity in the refereeing process. This paper aims to compare supervised machine learning techniques tested to accomplish genre analysis in Introduction sections of software engineering articles. A semi-supervised approach to augment the number of annotated sentences in SciSents[1] was performed. Two supervised approaches using SVM and logistic regression to assess the F-score for genre analysis in the corpus were undertaken. A technique based on logistic regression and BERT has been found to perform genre analysis highly satisfactorily with an average of 88.25 on F-score when retrieving patterns at an overall level.

1 Introduction

Written communication plays a fundamental role in scholarly development. Evidence for this is the high number of estimated publications and journals (Larsen and von Ins, 2010; Björk et al., 2008; Mabe, 2003). In this scenario, the reviewing process is a crucial pathway for improving publication quality, as it acts as a filter through which suitable research papers are selected for publication (Ware and Mabe, 2015). In principle, although academic gatekeeping does not entail rigid language rubrics, scientific publications on the whole present standardised conventions such as preference for passive constructions, high nominal style, paper division in sections, and use of lexical and phrasal structures to indicate the function and purpose of each text portion (Seaghdha and Teufel, 2014). Disseminated linguistic work aiming to systematically describe writing organization with a focus on the Introduction section is the CARS (Create a Research Space) model (Swales, 1990). CARS approaches genre analysis by introducing two concepts, namely *Moves* and *Steps*. Whereas a *Move* represents the objectives and functions of a text segment at an overall level, a *Step* further elaborates on explaining how the rhetorical means are specifically used to perform the function of a *Move* (Ruiying and Allison, 2003) (see examples in Table 1). Despite models serving as a basis for the reviewing process (e.g. CARS), the availability of reviewers to evaluate scientific publications does not keep pace with the ever-growing number of papers which require gatekeeping (Fox, 2017), therefore making computational techniques necessary.

Computational approaches may be implemented to query linguistic segments automatically in research articles by indicating the presence or absence of commonly used rhetorical patterns. Automatic approaches such as Support Vector Machines (SVM) (Bennett and Demiriz, 1999; Tang et al., 2007) can be employed to perform genre analysis due to its productive results regarding textual issues (Horn et al., 2014; Fernández-Delgado et al., 2014). Nonetheless, they require annotated data, which are scant in the literature and not easily obtained, with the existing ones having limited amount of input (Fisas et al., 2015, 2016; Seaghdha and Teufel, 2014; Anthony and Lashkia, 2003; Pendar and Cotos, 2008; Cotos and Pendar, 2016; Fiacco et al., 2019). Manual annotation is an arduous, expensive and time-consuming task as it requires expert human annotators. To tackle this issue, SVM may be used as a semi-supervised approach, in which considerable amounts of labeled and unlabeled data are

[1] Avaliable on: https://github.com/coling2020-lais/SciSents

Proceedings of Recent Advances in Natural Language Processing, pages 63–72
Sep 1–3, 2021.
https://doi.org/10.26615/978-954-452-072-4_008

utilized together to form more solid classifiers (Zhu, 2005). In this context, this work aims to evaluate supervised and semi-supervised machine learning techniques for automatic retrieval of rhetorical patterns within Swales' CARS genre analysis schema in research articles. This investigation was carried out into SciSents[2] corpus and, for this reason, is restricted to the Introduction section of software engineering articles. This paper has two main objectives: the first is to augment the number of annotations in SciSents corpus, and the second is to compare and assess F-Scores generated by supervised approaches for genre analysis. For such, we performed a comparison between SVMs and logistic regression techniques for the classification task. For sentence encoding, we evaluated novel approaches such as Universal Sentence Encoder (Cer et al., 2018a) and BERT (Devlin et al., 2019).

This paper proceeds as follows: firstly, we review state-of-the-art works on genre analysis automation. Next, we present and describe the corpus and the semi-supervised annotation procedures. Also, we comparatively discuss the main features of the techniques employed in the experiments. We then address the included implementation details and, finally, report the results.

2 Related Work

One relevant reference for genre analysis automation is the work of Anthony and Lashkia (2003), who proposed a computer software tool for outlining a research article's structure. Based on Swales' schema, the tool (named *Mover*) aimed at presenting to learners a panorama of the move structure utilized in RA. The tool scanned 100 information technology articles abstracts comprising 692 sentences. The abstracts were manually annotated on the grounds of a Modified Create a Research Space (CARS) model proposed by Anthony (1999). The model includes Swales' (1990) 3 *Moves*, as well as 12 *Steps*. Since this is a general and small-sized model designed for the Introduction section, not all *Steps* appeared in the dataset. A modified bag of words was utilized to represent the text so it could be machine manipulated. In a traditional bag of words, dataset sentences would be tokenized in single words. However, the authors added clusters of sequential words in order to allow the system to operate at the discourse level, therefore naming the model as *Bag of Clusters*. As well as allowing the system to identify steps only possible to classify if preceding or subsequent *Steps* are known, an additional "location" feature was added to the bag of clusters model. The model's output fed a Naive Bayes classifier which performed consistently with an average *Step* accuracy rate of 68% (ranging from 17% - Indicate gap - to 92% - Announce research). The authors justified the poor results by the scarce training items from these *Steps*. Through error analysis, they observed that when the software presented flaws, the incorrectly categorised *Step* tended to fall within the same *Move*. In order to improve accuracy the two most probable classifications were used in a second experiment. In this turn, the user had to select the most appropriate option. With this procedure, accuracy achieved 86%. However, despite the productive result, the reduced number of articles and sentences was a hindrance for further validation.

Pendar and Cotos (2008) attempted to devise a pedagogical tool for automating discourse evaluation. The purpose was to appraise academic writing drafts in agreement with an adapted model based on CARS, to compare it with other papers from the same discipline and to provide feedback to the student. To develop such a tool, a text-categorization approach using Suport Vector Machine (SVM) for sentence classification in research article introductions drawn on Swales' rhetorical moves was employed. An experiment was conducted with a corpus named Intelligent Academic Discourse Evaluator (IADE) consisting of 11,149 sentences from 401 Introduction sections in 20 academic disciplines. Each sentence was manually annotated within the *Moves* from CARS schema. To execute the classification, sentences were stemmed and represented in an n-dimensional vector (up to word trigrams). Experiment results were encouraging (with an accuracy above 70%), but the dataset was relatively small and did not take *Steps* into account.

Cotos and Pendar (2016) made progress in their own 2008 work by increasing IADE's size to 1,020 research articles across 51 disciplines. Sentences were also annotated according to the CARS model, but this time including both *Moves* and *Steps*. An SVM classifier with the previous settings achieved a *Move* accuracy of 72.6% and a *Step* accuracy of 72.9%.

Fiacco et al. (2019) presented a neural network architecture composed of a Bi-LSTM with CRF

[2] Avaliable on: https://github.com/coling2020-lais/SciSents

as an automated approach to examine rhetorical structure in student writing. The embedding layer was initialised with a pre-trained representation of GloVe (Pennington et al., 2014) and was fine-tuned to the dataset to produce more accurate word representation. Two datasets were used to test the model: IADE (Pendar and Cotos, 2008; Cotos and Pendar, 2016) and Research Writing Tutor (RWT) comprising 900 full research articles (not only Introduction sections) across 30 academic disciplines. RWT was manually annotated and sentences with one communicative goal and more than one functional strategy could be labeled with several steps; a sentence could be assigned with a secondary *Move/Step* tag if it had more than one communicative goal. Experiments results achieved a precision and recall of 77%, and an F1-score of 76% for the classification task in RWT dataset.

Due to the data paucity problem present in the aforementioned works, this study proposes a semi-supervised approach as a contribution towards genre analysis automation as far as the CARS framework is concerned. A detailed explanation of the procedures can be found in the following sections.

3 Semi-Supervised Approach

3.1 Data

We used SciSents, a dataset of software engineering research article sentences. This data resource is based on 9,193 software engineering articles published between the years 2000 and 2018 in highly-cited journals and conference proceedings. The corpus consists of 322,630 sentences from Introduction sections. From this amount we randomly extracted 595 sentences as our dataset, which was then manually annotated across 13 *Steps* within 3 *Moves* as shown in Table 1.

3.2 Models

We automatically performed the genre analysis classification task comparing SVMs and logistic regression as classifiers as well as BERT (Devlin et al., 2019) and Universal Sentence Encoder (Cer et al., 2018a) as sentence embeddings.

3.2.1 Classifiers

SVM: Support Vector Machines are non-parametric and deterministic algorithms based on statistical learning. They have been used specially in NLP (Joachims, 1998; Yang, 1999; Goudjil et al., 2018). SVM builds a hyperplane in a multi-dimensional space with the aim of training a set of labeled instances which create a boundary between distinct classes (Hearst et al., 1998; Joachims, 1998).

Logistic Regression: Logistic regression is a statistical technique for binary classification that can also be applied to multi-class classification by treating genre analysis issues as a binary classification problem (Ifrim et al., 2008). It computes probabilities of classes using a logistic function and then constructs a linear hyperplane separating those classes.

3.2.2 Features

Universal Sentence Encoder: Universal sentence encoding (Cer et al., 2018a) generates embedding vectors by encoding greater-than-word length text using two models: transformer architecture (Vaswani et al., 2017) and Deep Averaging Network (DAN) (Iyyer et al., 2015). Transformer architecture encoder consumes substantial resources and imposes complexity to the model aiming at high accuracy. It is context-aware and takes into account the ordering and the identity of all words in context. It also uses attention to compute the representations of words in a sentence. The second encoding model (i.e. DAN) assumes lightly reduced accuracy aiming at efficient inference. It receives embeddings for words and bi-grams as input, computes its average and inserts it into a feedforward Deep Neural Network (DNN) to create sentence embeddings. The output of both models is a 512-dimensional sentence embedding.

BERT: Bidirectional Encoder Representations from Transformers or BERT (Devlin et al., 2019) is a masked-language model for representing text and comprises a multi-layered bidirectional transformer encoder used for pre-training on a large unlabeled text corpus. It aims at modelling masked-language as well as predicting the next sentence. A random sample of the tokens is masked (replaced with the special token), the next sentence is predicted and BERT proceeds with training and optimization until it obtains satisfactory results (Liu et al., 2019).

3.3 Method

3.3.1 Semi Supervision

To increase the number of annotated sentences in SciSents, we employed a semi supervised strategy, training an SVM in the labeled part of corpus to

Move Step	SS	R1	R2
Establishing the territory	**187**	**257**	**444**
M1-S01 - Establishing the importance of the topic for the discipline	37	57	94
M1-S02 - Establishing the importance of the topic for the world or society	45	65	98
M1-S03 - Establishing the importance of the topic as a problem to be addressed	45	63	116
M1-S04 - Referring to previous work to establish what is already known	60	72	136
Establishing a niche	**98**	**136**	**199**
M2-S05 - Identifying and highlighting inadequacies, weaknesses, controversies and negative outcomes within the field of study	45	63	113
M2-S06 - Identifying a knowledge gap, a lack of or paucity of previous research in the field of study	53	73	86
Occupying the niche	**310**	**435**	**666**
M3-S07 - Stating the focus, aim, purpose or argument of the current research	44	64	97
M3-S08 - Setting out the research questions or hypotheses	36	56	73
M3-S09 - Describing the research design and the methods used	47	67	122
M3-S10 - Explaining the significance or give reasons for personal interest in the current study	33	42	100
M3-S11 - Describing the limitations of the current study	31	50	72
M3-S12 - Outlining the structure of a chapter, paper, thesis or dissertation	80	99	117
M3-S13 - Explaining Keywords (also refer to Defining Terms)	39	57	85
TOTAL	**595**	**828**	**1,309**

Table 1: Number of manually annotated sentences by *Move* and by *Step* in SciSents (SS), in semi supervised Round 1 (R1) and Round 2 (R2).

classify the unlabeled part. For such, the corpus phrases were represented in a 1024-position vector using BERT (Devlin et al., 2019), following the implementation of Xiao (2018) as described in section 3.2.2.

Following the annotation stage, the probability of the corpus sentences falling into each of the 13 *Steps* in SciSents was computed. The 20 most likely sentences for each *Step* (260 in total) were manually checked by a human linguistic expert with considerable knowledge on Swales' CARS model. Through this analysis we identified 228 correctly classified sentences against 5 wrongly classified ones. 27 sentences could not be categorized because of a few tokenization glitches. At the end of this stage, 233 sentences were added to the annotated set (including the former 5 incorrectly classified ones which were later corrected), amounting to a total of 828 manually annotated sentences. A new SVM training was then administered with this annotated set.

A second round of semi-supervised annotation followed, in which the linguistic expert analysed random sentences with different probabilities for *Steps* calculated by the second SVM training. A total of 481 random sentences were manually checked, of which 308 were marked as correctly classified and 173 marked as incorrectly classified. The misclassified sentences were manually reclassified so they could be added to the correct ones within the annotated set. Sentences with tokenization problems were discarded. At the end of this stage, 1,309 sentences were part of the manually annotated set (see Table 1). This corpus was used in the experiments, which are presented in the following section.

3.3.2 Evaluation

We used three measures to assess model performance: Precision, Recall, and F-score. Precision measures the proportion of correctly classified sentences out of the total number of annotated sentences, while Recall estimates the proportion of correctly annotated sentences out of the incorrectly predicted sentences plus the correctly classified sentences. F-Score in turn is the harmonic mean of both Precision and Recall (Goutte and Gaussier, 2005). Each technique was trained using 5 fold cross-validation and averages across F-Score results on test folds were reported.

Two embeddings were generated for the experiments. The first consisted of generating corpus sentence representation individually and the second

a representation in co-occurrence with the previous sentence. The purpose of this second approach was to investigate whether the previous sentence had an influence on the subsequent one in terms of genre analysis. In cases where the sentence was not preceded by any other, representation was calculated with that sentence solely. We highlight that the previous sentences were not necessarily the immediately preceding ones since invalid sentences were removed from SciSents during the preprocessing stage. The embedded sentences and phrase labels were the input for SVM training.

3.4 Baselines

The SVM-BERT pair, availed as the basis for the semi-supervised annotation (considering the embeddings generated from individual corpus sentences), was used for comparison with the rest of the experiment. For each technique, we explored two combinations of sentence embedding features: Universal Sentence Encoder and BERT. As to the former, a TensorFlow implementation[3] (Cer et al., 2018b) was used and generated a 512-dimensional sentence embedding vector. Regarding the latter, BERT as a Service (Xiao, 2018) was employed and generated a 1024-position vector.

4 Results

We report the F-score averaged over the folds of our techniques in Tables 2, 3, 4 and 5. Each table column shows the result of an experiment type comprising a technique (SVM or logistic regression), a sentence embedding technique (BERT or universal sentence encoder), and an annotated set (SciSents, semi-supervised Round 1 and semi-supervised Round 2).

Table 2 summarizes the results of experiments on the *Steps* categories when using one sentence alone to generate the embeddings[4]. Except for 2 *Steps* (M1-S03- Establishing the importance of the topic as a problem to be addressed and M3-S11-Describing the limitations of the current study), logistic regression technique with BERT presented higher scores overall. In 6 times out of these the highest results in the semi-supervised annotation Round 2 were achieved for the following *Steps*: M1-S01-Establishing the importance of the topic for the discipline; M1-S02-Establishing the importance of the topic for the world or society; M1-S04-Referring to previous work to establish what is already known; M2-S05-Identifying and highlighting inadequacies, weaknesses, controversies and negative outcomes within the field of study; M3-S09-Describing the research design and the methods used; M3-S10-Explaining the significance or give reasons for personal interest in the current study. In the remaining 5 *Steps* (i.e. M2-S06-Identifying a knowledge gap, a lack of or paucity of previous research in the field of study; M3-S07-Stating the focus, aim, purpose or argument of the current research; M3-S08-Setting out the research questions or hypotheses; M3-S12-Outlining the structure of a chapter, paper, thesis or dissertation; M3-S13-Explaining Keywords (also refer to Defining Terms)), better scores were obtained in the semi-supervised annotation Round 1.

The best performance among all results was achieved for *Step* M3-S12 (Outlining the structure of a chapter, paper, thesis or dissertation) in semi-supervised annotation Round 2 using logistic regression and BERT, which showed a 0.8856 F-Score. The results in Table 2 for M3-S12 were higher than 0.84. This result can be explained by the fact that sentences within this *Step* are prototypical (e.g. *"The paper is structured as follows"*,*"Finally, Section 6 concludes the paper and discusses its implications."*, and *"The remainder of this paper begins with a comparison to related work (Section 2), followed by an overview of the approach used to create a corpus, perform change classification, and evaluate its performance (Section 3)."*).

The worst performance among all results in Table 2 was a 0.1152 F-Score produced in M3-S10 (Explaining the significance or give reasons for personal interest in the current study) when using logistic regression and universal sentence encoder in SciSents annotated sentences. One possible explanation for this low performance is that the number of annotations is one of the smallest among all *Steps* (33 sentences). In addition, this result can be justified by the fact that sentence type used in this *Step* is quite varied such as *"Our experiments, backed by a human study, suggest DeltaDoc could replace over 89% of human-generated What log messages."*, *"This combines visualizations, providing a high level overview, and wiki pages, providing detailed information juxtaposed in a focus-plus-context oriented format."*, and *"The backward analysis computes an over approxima-*

[3] https://tfhub.dev/google/universal-sentence-encoder/4

[4] The strongest F-score in each row is in bold.

tion of all possible inputs that can generate those attack strings.”. Throughout annotations rounds, M3-S10 improved its results and reached a performance of 0.4092. The best performance in Table 2 for M3-S10 scored 0.5081 when using the SVM-BERT pair.

Table 3 shows the performance of the experiments on *Steps* when using both actual and previous sentences to generate vector representation[5]. The pair logistic regression with BERT surpassed other pairs in 7 (M1-S02, M1-S03, M2-S05, M2-S06, M3-S08, M3-S09, and M3-S10) out of the 13 *Steps*. As to the results regarding sole sentence embedding, the best performance among all was achieved in M3-S12 but this time in SciSents annotations using SVM and BERT with a 0.8932 F-Score. One of the reasons that may have contributed to this result even before semi-supervised rounds is the annotated sentence number (80) being the highest among all *Steps*. The worst performance in this type of experiment was a 0.1152 F-Score output for M3-S10.

We notice that results shown in Table 2 are more productive than the ones from Table 3 in 44 (or 56.41%) out of 78 when considering experiments that used BERT in isolation. When analysing only the best scores for each *Step*, Table 2 presents best results in 8 (61.53%), whereas Table 3 shows the most productive scores in 4 (30.77%) out of 13 cases. There was a draw in one case. Performance with universal sentence encoding was the same on both tables.

Table 4 summarizes the results of experiments on *Moves* when using one sentence solely to generate the embeddings[6]. The best F-score for each *Move* was achieved with logistic regression and BERT in semi-supervised annotation Round 1 with an average of 0.8569 against an average of 0.8422 for Round 2. The lowest score in Round 2 was 0.7867 for M1 (Establishing the territory) whereas M2 (Establishing a niche) scored 0.8564. M3 (Occupying the niche) outperformed all other results with a score of 0.9275. When we compare these results with their respective scores in semi-supervised annotation (Round 2), there is a difference of 0.0126, 0.0129, and of 0.0187 between *Moves* M1, M2 and M3 respectively.

Table 5 presents results on *Moves* when the vector representation is created using the actual sentence in conjunction with the previous sentence [7]. Similar to the technique with sole sentence embeddings for *Moves*, the best F-Score was reached with logistic regression and BERT. But this time M1 and M2 were reached in semi-supervised annotation in Round 2 and M3 in semi-supervised annotation in Round 1. When we compare scores from Table 4 with those from Table 5 we can notice that the figures on the former surpass all respective results on the latter when considering BERT alone. Again, when Universal Sentence Encoder was used there was no difference between the embedding from one sentence alone and from a sentence co-occurring with its previous one. [8]

5 Discussion

The present study was designed to augment the number of annotations in SciSents corpus and to compare results in supervised machine learning techniques for genre analysis in software engineering research articles. The number of annotated sentences was increased from the 595 ones in SciSents to 1309 through two semi-supervised rounds using SVM.

SVM versus Logistic Regression: Logistic regression produced higher outcomes than SVM in 64% of the experiments. When associated with BERT, logistic regression beats SVM in 85% of cases, but when in conjunction with USE, SVM outperformed logistic regression in 57% of experiments.

Universal Sentence Encoder versus BERT: Vector representation provided by BERT delivered higher scores than Universal Sentence Encoder did in 75,5% of the tested sets. When BERT was employed with logistic regression, the results overcame other experiments in 81% of cases. Thus, from the pairs of techniques tested, the indicated one for genre analysis is logistic regression with BERT.

Vector representation - sentence alone versus co-occurring sentences: One finding in the experiments in supervised machine learning techniques is that, in most cases, the use of sentence embedding generated from the sentence alone provided more productive results than those with the use of the actual sentence together with its preceding one.

[5]The strongest F-score in each row is in bold.

[6]The strongest F-score in each row is in bold.

[8]The strongest F-score in each row is in bold.

	SVM-BERT			SVM-USE			LR-BERT			LR-USE		
Step	**SS**	**R1**	**R2**	**SS**	**R1**	**R2**	**SS**	**R1**	**R2**	**SS**	**R1**	**R2**
M1-S01	34.00	53.28	65.05	34.5	54.02	65.71	39.52	59.24	**65.80**	35.82	61.59	62.56
M1-S02	31.87	52.10	61.56	47.33	58.46	56.31	38.52	53.05	**64.32**	40.24	60.76	55.31
M1-S03	47.51	62.80	**63.54**	49.44	55.11	58.73	45.97	63.26	62.91	52.02	51.91	55.18
M1-S04	33.14	44.57	54.47	32.10	49.12	50.44	41.26	51.93	**60.98**	34.92	47.55	51.00
M2-S05	41.8	50.43	55.88	40.84	53.07	51.24	37.99	51.6	**58.42**	35.65	53.97	52.17
M2-S06	69.09	80.46	78.00	62.54	74.45	70.45	73.33	**81.60**	79.81	55.59	69.72	65.92
M3-S07	50.75	68.65	70.30	49.99	68.97	64.18	58.18	**77.04**	74.03	53.60	67.57	66.18
M3-S08	58.56	75.33	63.19	62.83	66.64	53.22	60.66	**76.03**	69.59	60.11	61.44	55.22
M3-S09	27.92	46.02	63.38	29.86	42.09	55.07	24.03	51.20	**64.58**	26.28	43.74	51.58
M3-S10	31.98	34.30	50.81	19.50	25.76	42.98	36.79	38.29	**56.82**	11.52	18.43	40.92
M3-S11	78.97	86.12	76.75	75.84	78.79	75.26	81.38	86.92	80.44	83.70	**87.66**	75.55
M3-S12	82.94	85.71	85.37	75.45	81.32	74.64	86.43	**88.55**	84.74	71.47	81.01	74.90
M3-S13	81.69	85.42	81.45	69.94	74.65	73.68	84.29	**88.13**	79.72	71.12	75.07	69.46
Overall	53.93	65.84	66.71	52.27	62.46	60.68	56.54	68.44	**69.16**	50.14	61.85	59.54

Table 2: Experiment results per *Step* on SciSents (SS), semi supervised Round 1 (R1) and Round 2 (R2) annotated sentences sets using one sentence solely to create vector representation (LR = Logistic Regression; USE = Universal Sentence Encoder).

	SVM-BERT			SVM-USE			LR-BERT			LR-USE		
Step	**SS**	**R1**	**R2**	**SS**	**R1**	**R2**	**SS**	**R1**	**R2**	**SS**	**R1**	**R2**
M1-S01	27.22	47.55	56.63	34.5	54.02	**65.71**	32.34	52.67	62.11	35.82	61.59	62.56
M1-S02	60.5	64.50	70.50	47.33	58.46	56.31	58.99	67.35	**75.92**	40.24	60.76	55.31
M1-S03	41.16	53.64	58.58	49.44	55.11	58.73	48.25	54.81	**58.79**	52.02	51.91	55.18
M1-S04	44.42	52.09	**60.84**	32.10	49.12	50.44	50.14	56.79	58.09	34.92	47.55	51.00
M2-S05	47.59	52.68	61.10	40.84	53.07	51.24	48.60	54.60	**62.42**	35.65	53.97	52.17
M2-S06	56.02	74.03	71.44	62.54	74.45	70.45	62.12	**79.07**	75.12	55.59	69.72	65.92
M3-S07	29.63	55.37	53.37	49.99	**68.97**	64.18	33.65	54.50	59.49	53.60	67.57	66.18
M3-S08	47.48	64.74	52.68	62.83	66.64	53.22	57.30	**68.57**	58.79	60.11	61.44	55.22
M3-S09	40.29	52.12	61.83	29.86	42.09	55.07	41.19	59.27	**66.49**	26.28	43.74	51.58
M3-S10	50.5	39.77	58.66	19.50	25.76	42.98	55.03	55.37	**62.91**	11.52	18.43	40.92
M3-S11	69.51	72.71	71.02	75.84	78.79	75.26	68.43	76.55	70.77	83.70	**87.66**	75.55
M3-S12	**89.31**	87.84	84.65	75.45	81.32	74.64	87.10	86.59	83.27	71.47	81.01	74.90
M3-S13	78.60	77.56	**82.51**	69.94	74.65	73.68	79.55	81.23	80.61	71.12	75.07	69.46
Overall	55.37	63.72	65.37	52.27	62.46	60.68	58.74	66.50	**67.50**	50.14	61.85	59.54

Table 3: Experiment results by *Steps* on SciSents (SS), semi supervised Round 1 (R1) and Round 2 (R2) annotated sentences sets using a sentence in co-occurrence with its immediately preceding one to create vector representation (LR = Logistic Regression; USE = Universal Sentence Encoder).

	SVM-BERT			SVM-USE			LR-BERT			LR-USE		
Move	**SS**	**R1**	**R2**	**SS**	**R1**	**R2**	**SS**	**R1**	**R2**	**SS**	**R1**	**R2**
M1	72.22	75.91	72.68	60.98	69.65	63.53	72.20	**78.67**	77.41	46.79	65.79	61.70
M2	79.30	84.01	79.71	75.70	79.70	78.58	80.39	**85.64**	84.35	78.48	80.84	78.70
M3	88.17	91.99	88.17	85.61	88.54	85.35	89.23	**92.74**	90.87	84.87	88.48	85.86
Overall	82.78	86.86	82.98	78.57	82.68	79.81	83.72	**88.25**	86.66	76.87	82.40	79.88

Table 4: Experiment results by *Moves* on SciSents (SS), semi supervised Round 1 (R1) and Round 2 (R2) annotated sentences sets using one sentence solely to create vector representation (LR = Logistic Regression; USE = Universal Sentence Encoder).

	SVM-BERT			SVM-USE			LR-BERT			LR-USE		
Move	**SS**	**R1**	**R2**	**SS**	**R1**	**R2**	**SS**	**R1**	**R2**	**SS**	**R1**	**R2**
M1	63.42	65.99	69.06	60.98	69.65	63.53	64.70	72.01	**73.91**	46.79	65.79	61.70
M2	76.88	77.80	78.19	75.70	79.70	78.58	77.78	80.93	**82.45**	78.48	80.84	78.70
M3	86.18	88.60	86.69	85.61	88.54	85.35	87.09	**90.47**	89.4	84.87	88.48	85.86
Overall	79.74	81.53	81.20	78.57	82.68	79.81	80.81	84.51	**84.70**	76.87	82.40	79.88

Table 5: Experiment results by *Move* on SciSents (SS), semi supervised Round 1 (R1) and Round 2 (R2) annotated sentences sets using the sentence co-occurring with its previous one to create the vector representation (LR = Logistic Regression; USE = Universal Sentence Encoder).

Semi-Supervised Approach: When analysing the evolution of results throughout the annotation process within each experiment type we can notice that they did not always improve accordingly. In Table 2, when we compare SciSents with annotations from Round 1, there was no increase in the F-score (despite one situation only representing 1.92% out of the total). When comparing annotations from Round 1 with Round 2, the latter outperformed the former in 46.15% of the cases. A possible explanation is that in Round 1 highest-ranked sentences by SVM were annotated while in Round 2 sentences with random probabilities were annotated. Thus, in Round 1, similar sentences to those the techniques already knew were included, whereas in Round 2 sentences which were different from those the techniques knew (but still fell into that *Step*) were included.

When approaching annotation evolution throughout Table 3 we observe that experiments within Round 1 annotations outperformed experiments in SciSents in 90.38% of the results. When comparing experiments in annotations between Round 1 and Round 2 there is a 50% (26 times) draw in which Round 2 showed better results than Round 1. The same analysis in Tables 4 and 5 shows that experiments in Round 1 annotations outperformed experiments in SciSents. When comparing annotations between Round 1 and Round 2 we observe no improvement in the latter (as shown in Table 4), but some improvement in 33,33% of the overall cases, as we see in Table 5. These results indicate that the second round of annotation may have included sentence types unknown to the technique.

Although most of the best results for *Steps* were achieved with a second round annotation set, setbacks were also present, thus indicating the need for more annotations for probabilities potentially corresponding to the categories set for *Steps*. These annotations may also contribute to genre analysis regarding *Moves*, despite results presenting insignificant improvements with a second round of annotations.

6 Conclusion

The present study compared supervised machine learning techniques which automatically retrieved linguistic segments from research articles. Firstly, we used a semi-supervised approach to increase the number of annotated sentences in SciSents corpus. Next, we used two supervised and two sentence embedding techniques to carry out genre analysis on the dataset. The results suggest that an approach based on logistic regression and BERT presents higher scores for genre analysis. In addition, although a semi-supervised annotation process has proven to contribute to the overall procedure, it lacks elements with random probabilities for substantial improvement.

As future work, the semi-supervised annotation process and the techniques hereby described can be used for annotating other sections of software engineering research articles. Also, the same analyses could be applied to articles from other domains so that cross-disciplinary rhetorical differences could be identified.

References

Laurence Anthony. 1999. Writing research article introductions in software engineering: How accurate is a standard model? *IEEE transactions on Professional Communication*, 42(1):38–46.

Laurence Anthony and George V. Lashkia. 2003. Mover: a machine learning tool to assist in the reading and writing of technical papers. *IEEE Transactions on Professional Communication*, 46(3):185–193.

Kristin P Bennett and Ayhan Demiriz. 1999. Semi-supervised support vector machines. In *Advances in Neural Information processing systems*, pages 368–374.

Bo-Christer Björk, Annikki Roosr, and Mari Lauri. 2008. Global Annual Volume of Peer Reviewed Scholarly Articles and the Share Available Via Different Open Access Options. page 9.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, and Chris Tar. 2018a. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018b. Universal Sentence Encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

Elena Cotos and Nick Pendar. 2016. Discourse classification into rhetorical functions for AWE feedback. *Calico Journal*, 33(1):92–116.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of

Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. 2014. Do we Need Hundreds of Classifiers to Solve Real World Classification Problems? *Journal of Machine Learning Research*, 15:3133–3181.

James Fiacco, Elena Cotos, and Carolyn Rosé. 2019. Towards enabling feedback on rhetorical structure with neural sequence models. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, pages 310–319. ACM.

Beatriz Fisas, Francesco Ronzano, and Horacio Saggion. 2016. A multi-layered annotated corpus of scientific papers. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3081–3088.

Beatriz Fisas, Horacio Saggion, and Francesco Ronzano. 2015. On the discoursive structure of computer graphics research papers. In *Proceedings of the 9th linguistic annotation workshop*, pages 42–51.

Charles W Fox. 2017. Difficulty of recruiting reviewers predicts review scores and editorial decisions at six journals of ecology and evolution. *Scientometrics*, 113(1):465–477.

Mohamed Goudjil, Mouloud Koudil, Mouldi Bedda, and Noureddine Ghoggali. 2018. A Novel Active Learning Method Using SVM for Text Classification. *International Journal of Automation and Computing*, 15(3):290–298.

Cyril Goutte and Eric Gaussier. 2005. A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. In *European Conference on Information Retrieval*, pages 345–359. Springer.

Marti A. Hearst, Susan T. Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. 1998. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28.

Colby Horn, Cathryn Manduca, and David Kauchak. 2014. Learning a lexical simplifier using Wikipedia. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 458–463.

Georgiana Ifrim, Gökhan Bakir, and Gerhard Weikum. 2008. Fast logistic regression for text categorization with variable-length n-grams. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 354–362.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1681–1691.

Thorsten Joachims. 1998. Text categorization with Support Vector Machines: Learning with many relevant features. In *Machine Learning: ECML-98*, Lecture Notes in Computer Science, pages 137–142, Berlin, Heidelberg. Springer.

Peder Olesen Larsen and Markus von Ins. 2010. The rate of growth in scientific publication and the decline in coverage provided by Science Citation Index. *Scientometrics*, 84(3):575–603.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Michael Mabe. 2003. The growth and number of journals. *Serials*, 16(2):191–198.

Nick Pendar and Elena Cotos. 2008. Automatic identification of discourse moves in scientific article introductions. In *Proceedings of the Third Workshop on Innovative Use of NLP for Building Educational Applications*, pages 62–70. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Yang Ruiying and Desmond Allison. 2003. Research articles in applied linguistics: moving from results to conclusions. *English for Specific Purposes*, 22(4):365–385.

Diarmuid O Seaghdha and Simone Teufel. 2014. Unsupervised learning of rhetorical structure with untopic models. page 12.

John Swales. 1990. *Genre analysis: English in academic and research settings*. Cambridge University Press.

Feng Tang, Shane Brennan, Qi Zhao, and Hai Tao. 2007. Co-tracking using semi-supervised support vector machines. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, \Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Mark Ware and Michael Mabe. 2015. The STM report: An overview of scientific and scholarly journal publishing. Technical report, International Association of Scientific, Technical and Medical Publishers, University of Nebraska-Lincoln.

Han Xiao. 2018. bert-as-service.

Yiming Yang. 1999. An Evaluation of Statistical Approaches to Text Categorization. *Information Retrieval*, 1(1):69–90.

Xiaojin Jerry Zhu. 2005. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences.