Universidade FUMEC
Faculdade de Ciências Empresariais
Programa de Pós-Graduação em Sistemas de Informação e Gestão do Conhecimento

Comparison of Machine Learning Techniques for Genre Analysis of Software Engineering Research Articles

Felipe Araújo de Britto

Belo Horizonte
2020

Felipe Araújo de Britto

Comparison of Machine Learning Techniques for Genre Analysis of Software Engineering Research Articles

MSc thesis presented to the Programa de Pós-Graduação em Sistemas de Informação e Gestão do Conhecimento of FUMEC University, as partial fulfillment of the requirements for the Master's degree in Information Systems and Knowledge Management.

Research Track: Technology and Information Systems

Advisor: Prof. Dr. Fernando Silva Parreiras

Co-advisor: Prof. Dr. Thiago Castro Ferreira

Belo Horizonte

2020

Dados Internacionais de Catalogação na Publicação (CIP)

B862c Britto, Felipe Araújo de, 1981-

Comparison of machine learning techniques for genre analysis of software engineering research articles / Felipe Araújo de Britto. - Belo Horizonte, 2020.

55 f. : il. ; 29,7 cm

Orientador: Fernando Silva Parreiras

Coorientador: Thiago Castro Ferreira

Dissertação (Mestrado em Sistemas de Informação e Gestão do Conhecimento), Universidade FUMEC, Faculdade de Ciências Empresariais, Belo Horizonte, 2020.

1. Processamento de linguagem natural (Computação). 2. Aprendizado do computador. 3. Engenharia de software. I. Título. II. Parreiras, Fernando Silva. III. Ferreira, Thiago Castro. IV. Universidade FUMEC, Faculdade de Ciências Empresariais.

CDU: 681.3.72

Ficha catalográfica elaborada pela Biblioteca Universitária-FUMEC

Dissertação intitulada "**Comparison of Machine Learning Techniques for Genre Analysis of Software Engineering Research Articles**" de autoria de Felipe Araújo de Britto, aprovado pela banca examinadora constituída pelos seguintes professores:

Prof. Dr. Fernando Silva Parreiras – Universidade FUMEC
(Orientador)

Thiago Castro Ferreira

Prof. Dr. Thiago Castro Ferreira – UFMG
(Corientador)

Maia

Prof. Dr. Luiz Cláudio Gomes Maia – Universidade FUMEC
(Examinador Interno)

Adriana Pagano

Profa. Dra. Adriana Silvina Pagano – UFMG
(Examinador Externo)

Prof. Dr. Fernando Silva Parreiras
Coordenador do Programa de Pós-Graduação em Sistemas de Informação e Gestão do Conhecimento da Universidade FUMEC

Belo Horizonte, 05 de março de 2020.

Abstract

The exponential growth in the number of scientific publications is not accompanied in the same pace by the growth of people curating scientific literature. The lack of these professionals to meet all the existing demands poses a challenge to various research communities. Machine learning techniques for natural language processing produce significant results in downstream tasks and may be used to analyse linguistic elements in research articles by indicating the presence or absence of common rhetorical patterns. This study aims to compare machine learning techniques when computing genre analysis in software engineering research articles. To achieve this goal, a scientific-research sentence corpus was created and annotated in a semi-supervised fashion using SVM. Supervised and unsupervised techniques (KNN, SVM, logistic regression, DBScan and LDA) were used to perform gender analysis over the corpus. SVM performs very satisfactorily for genre analysis with an average of 84.22 on f-score when querying linguistic elements on an overall level

Keywords: Genre analysis, natural language processing, machine learning.

Resumo

O crescimento exponencial do número de publicações científicas não é acompanhado no mesmo ritmo pelo crescimento de pessoas que fazem a curadoria da literatura científica. A falta desses profissionais para atender toda a demanda existente coloca um desafio para várias comunidades de pesquisa. Técnicas de *machine learning* produzem resultados significativos em tarefas de processamento de linguagem natural e podem ser usadas para analisar elementos lingüísticos em artigos científicos, indicando a presença ou ausência de elementos retóricos comuns. Este estudo tem como objetivo comparar técnicas de *machine learning* na análise de gênero em artigos científicos de engenharia de software. Para atingir esse objetivo, um *corpus* de sentenças de artigos científicos foi criado e anotado de forma semi-supervisionada usando SVM. Técnicas supervisionadas e não supervisionadas (KNN, SVM, regressão logística, DBScan e LDA) foram utilizadas para realizar a análise de gênero no corpus. O SVM teve um desempenho satisfatório na análise de gênero científico, com uma média de 84,22 no f-score ao analisar elementos linguísticos em um nível geral.

Palavras chave: Análise de gênero, processamento de linguagem natural, machine learning.

Acknowledgements

I would like to thank all my family, especially to my beloved wife for all the support and patience.

To my advisor Dr Fernando Silva Parreiras for the teachings and instructions.

To colleagues at LAIS (Laboratory for Advanced Information Systems), particularly to Daniel Henrique Mourão Falci, Marco Antônio Calijorne Soares and Bruno Rafael de Oliveira Rodrigues for their patience and tips.

Finally, I must express my gratitude to Dr. Adriana Silvina Pagano, Dr. Leonardo Pereira Nunes and Dr. Thiago Castro Ferreira for the huge contributions in this research.

List of figures

List of tables

Acronyms

BERT Bidirectional Encoder Representations from Transformers

BoW Bag of Words

CARS Create a Research Space

KNN K-Nearest Neighbors

LDA Latent Dirichlet Allocation

NLP Natural Language Processing

SVM Support Vector Machines

Summary

1 Introduction

Written communication plays a fundamental role in scholarly development. This importance is evidenced by exponential increase in the number of scientific publications in the last few decades (Bornmann and Mutz, 2015). A single publishing company received 1,3 million manuscripts in 2015 Reller (2016). This fact impacts the scientific publishing process since it calls for the need of a greater number of reviewers. However, a lack of these professionals to meet all the existing demand Fox (2017) poses a challenge to various research communities. Furthermore, the revision of a manuscript is composed of subjective, time-consuming and complex activities. A reviewer needs to have sufficient scientific background of the evaluated discipline and a mastery of the subjacent science. Become a reviewer requires training and practice, which are not easily obtained (Hames, 2008; Voight and Hoogenboom, 2012; Provenzale and Stanley, 2006). To address this issue computational approaches to automatically analyse linguistic elements in research articles by indicating the presence or absence of common rhetorical patterns have grown necessary.

Scientific publications have multiple conventions such as preference for the passive voice, paper division in sections, the use of lexical and phrasal structures to indicate the function executed by each part of the text (Seaghdha and Teufel, 2014), to cite a few. Disseminated linguistic work on scientific publications is the CARS (Create a Research Space) model proposed by Swales (1990). CARS uses a genre analysis approach and introduces two concepts, namely *Moves* and *Steps*. While 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 used to perform the function of a *Move* (Ruiying and Allison, 2003).

Computational approaches for Natural Language Processing (NLP) produce significant results in downstream tasks such as speech recognition Graves et al. (2013) , question answering Johnson et al. (2017); Andreas et al. (2016), machine translation, summarization and language generation Liu et al. (2018). Such tasks require large linguistic datasets that are scarce when it comes to genre analysis of literature. Researches on computational approaches which could help editors, reviewers and also authors in the scientific publishing process are also scant.

Considering this scenario, this thesis evaluates machine learning techniques that analyses linguistic elements of research articles. This analysis will be carried out into the introduction sections of software engineering research articles. Software engineering was chosen due to the current paucity of studies investigating genre analysis in this field.

Furthermore, software engineering is a field that gains importance in an era where people and companies rely on software and systems. Moreover, software engineering is a multidisciplinary area that not only handles with tools used to construct and maintain software but also with the human processes surrounding them.

1.1 Research Problem

In this context, this MSC thesis addresses the following research questions: **what is the accuracy of machine learning techniques for automatic genre analysis of software engineering research articles in the English language?**

1.2 Objectives

1.2.1 Main Objective

The main objective of this thesis is to compare machine learning technique for automatic genre analysis of software engineering research articles in the English language.

1.2.2 Specific Objectives

The specific objectives of this MSC thesis are:

- **Obj1**: Create a sentence corpus of software engineering research papers for genre analysis in the English language.
- **Obj2**: Expand the sentence corpus using a semi-supervised approach for annotation.
- **Obj3**: Compare the accuracy of supervised machine learning techniques for automatic genre analysis of software engineering research articles in the English language.

1.3 Adherence to FUMEC's Graduate Program

The FUMEC's graduate program in Information Systems and Knowledge Management is focused on conducting applied and practices research in managerial and technological areas. The program has a multidisciplinary approach in Technology and Information Systems and Information and Knowledge Management lines of research.

This project proposes the application of natural language processing and machine learning techniques for genre analysis of software engineering research articles. This analysis may support editors, reviewers and also authors to accelerate reviewing and proofreading processes. This thesis focus is under the Technology and Information Systems field and into Cognition, Machine Learning and Information Retrieval subfield in compliance with FUMEC's graduate program.

The multidisciplinary character comes from the purpose of the research that may enable future applications in several disciplines.

1.4 Communications of this Thesis

The research presented in this thesis will be communicated through proceedings. In the following, the publication is mentioned according to the chapters covering the respective contribution.

- **Chapter 2 - A Gold Standard Corpus for Genre Analysis in Software Engineering Research Articles**: 28th International Conference on Computational Linguistics (COLING'2020).
- **Chapter 3 - Comparison of Supervised Machine Learning Techniques for Genre Analysis of Software Engineering Research Articles**: 28th International Conference on Computational Linguistics (COLING'2020).

1.5 Document Structure

This MSc thesis is structured in 4 chapters. Chapter 1 presented the Introduction. Chapter 2 presents the first paper to be submitted to publication and is composed of the process details to create a gold standard corpus for genre analysis. Chapter 3 presents the second paper to be submitted to publication, describing the supervised machine learning techniques employed to analyse the results of automatic genre analysis. Finally, Chapter 4 outline the conclusions about the current research.

2 A Gold Standard Corpus for Genre Analysis in Software Engineering Research Articles

2.1 Introduction

In the last decades, there has been a global exponential increase in the number of scientific publications (Bornmann and Mutz, 2015; Ware and Mabe, 2015); A single publishing company received 1,3 million manuscripts in 2015 (Reller, 2016). This fact impacts the scientific publishing process since it calls for the need of a greater number of editors and reviewers. However, increasing the number of professionals curating scientific literature is a challenge as the revision of a manuscript entails subjective, time-consuming and complex activities. A reviewer needs to master the conventions of academic discourse or genre employed in the discipline under evaluation. They also need to have sufficient scientific domain knowledge and to grasp various theoretical underpinnings. Furthermore, becoming a qualified reviewer is not easily achievable, as it requires extensive training and practice (Hames, 2008; Voight and Hoogenboom, 2012; Provenzale and Stanley, 2006).

While the number of scientific publications is in steady growth, research on computational approaches which could help editors, reviewers and also authors in the scientific publishing process remains scant. Rather, manual analysis to investigate genre and communicative events employed in papers is more frequently reported in the literature (Ruiying and Allison, 2003; Kanoksilapatham, 2005; Basturkmen, 2012; Maswana et al., 2015). In light of this context, an automatic approach which analyses linguistic elements in research articles or papers by indicating the presence or absence of common rhetorical patterns may lead to improvements to the scientific publishing process.

Computational approaches to Natural Language Processing (NLP) produce significant results in downstream tasks such as speech recognition (Graves et al., 2013) , question answering (Johnson et al., 2017; Andreas et al., 2016), machine translation, summarisation and language generation (Liu et al., 2018). Such tasks require large linguistic datasets that are scarce in the literature when it comes to genre analysis. Considering this scenario, this paper presents SciSents[1], a newly created gold standard corpus to bridge this gap. The corpus is based on 9,193 articles from 10 highly-cited software engineering journals and proceedings, comprising 322,630 sentences from the Introduction sections. In this respect, the results of initial experiments of automatic clustering of sentences from the corpus using 3 machine learning techniques (K-Nearest Neighbors (KNN), Latent Dirichlet Allocation(LDA), and DBScan) are presented. These resources aim to contribute to

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

research by supporting the development of approaches for genre analysis and by providing a comparative analysis of the techniques hereby used.

This paper is organized as follows: Section 2 reviews the theoretical foundations used, Section 3 presents related works, Section 4 describes the process of creating, cleaning and annotating the corpus, Section 5 details the experiments executed to explore the corpus, Section 6 presents the experiment results, and Section 7 summarizes the conclusions of this study and outlines future works.

2.2 Foundations

2.2.1 Genre Analysis

Genre is a fuzzy concept frequently employed to refer to categories of real-world entities. In the language field, genre may be erroneously characterized as a mere mechanism or may be associated as a formulaic way of constructing texts when it is in fact a matter of choice mainly influenced by the research community in which the author belongs (Swales, 1990). Each community uses specific language to target a specific audience. Thus, genre analysis is employed as an attempt to provide a grounded explanation of language use in scholarly or professional settings (Bhatia, 2014).

In the academic context, researchers use scientific publications to report the results from their studies. The communication form used in this type of publication is quite specific to the academic environment and even more so when considering disciplines individually. In this context, English for Academic Purposes (EAP), encompassing both English for Specific Academic Purposes (ESAP) and English for General Academic Purposes (EGAP) (Jordan, 1997), enables the identification of more in-depth descriptions of academic language and the understanding of its communicative nature through text comprehension and restrictions of the scholarly context (Hyland, 2006). ESAP mainly investigates practices within a certain academic discipline, such as medicine, physics and software engineering. The genre analysis approach used by Swales (1990) is a disseminated linguistic-pedagogical research in ESAP.

In his work, Swales states that "a genre comprises a class of communicative events, the members of which share some set of communicative purposes"(Swales, 1990, p. 58). These communicative events are combined according to their structure, content, intended audience and style to reach purposes of overall communication. Representative samples of a genre may be seen as a prototypical of parent discourse community and can be recognized by the community expert. The genre is justified by those communicative purposes that impact the choice of content and style forming the schematic arrangement of the discourse. Swales' definition of genre can be best explicated as a sociocognitive construct which particular speech and writing communities utilize to keep discoursal forms and promote recurring rhetorical purpose to establish recurring practices to communicate with one another (Lenart and Berdanier, 2017). Although genres are typically correlated with repetitive rhetorical environments, discoursal forms and lexico-grammatical constraints that allow us to identify a shared set of communicative purposes, they are not static (Bhatia, 2014).

Swales (1990) proposes the Create a Research Space (CARS) model which primarily focuses on the Introduction section and is made up of *Moves* and *Steps*. According to (Ruiying and Allison, 2003, p. 370) "the concept of Move captures the function and purpose of a segment of text at a more general level, while Step spells out more specifically the rhetorical means of realizing the function of Move". A *Move* may be achieved with one or more *Steps*. A combination of *Steps* for a *Move* is the collection of rhetorical selections usually available to authors to meet a specific purpose. The sequence that a *Step* appears in each *Move* is not a prototypical feature of the genre itself, but rather the author's preferred order (Ruiying and Allison, 2003).

Bhatia (1993), however, advocates that the genre determines the *Move*'s specific characteristics, and that authors of a particular scientific genre tend to use certain patterns of rhetorical moves. Thus, the CARS method is commonly employed to identify these patterns in each section of a paper. The analysis of combined sections can indicate the structural pattern of the whole text and, as a result, provide a comprehensive understanding of a specific genre.

2.2.1.1 Academic Phrasebank

Academic Phrasebank is a general resource for academic writers compiling a database of English sentences selected from scholarly sources (Morley, 2018). It purports to provide phraseological examples for academic writers inspired by Swales' genre analysis approach (Swales, 1981, 1990).

Phrasebank sentences were initially derived from 100 postgraduate dissertations written at the University of Manchester, UK. Phrases retrieved from research articles of different disciplines were, and remain to be, included [2]. The phrases are divided into two distinct sets: one with sentences categorized by the main sections of an academic article or dissertation and the other with sentences classified according to their general communicative functions.

The Academic Phrasebank database is composed of thousands of phrases totalling 182 *Steps*. There is no explicit grouping into *Moves*. To enable the sentences to be used in any discipline, the Academic Phrasebank extraction process selected phrases that had non-

[2] A reduced version of *Phrasebank* can be accessed at `http://www.phrasebank.manchester.ac.uk/`

domain specific content. In addition, sentences were simplified and cleaned of their specific domain language content. The Academic Phrasebank Introduction Section is composed of 2925 sentences and 23 *Steps* (Morley, 2018).

2.2.2 Natural Language Processing

For a human to understand another human there is a somewhat easy task. However, this is a rather complex endeavour when it comes to computers. Natural Language Processing (NLP) was created with the aim to explore how computers can be used to learn, understand, and produce content in human language (Chowdhury, 2003; Hirschberg and Manning, 2015). NLP foundations lie in a number of disciplines such as computer and information sciences, linguistics, mathematics, electrical and electronic engineering, and psychology (Chowdhury, 2003). NLP goals are varied and can use a set of techniques, such as counting word frequencies to compare different writing styles or employ deep learning models to understand and provide answers for human questions.

The last two decades have seen a growth of NLP into both scientific research and practical technology. This growth occurred thanks to the rise in computing power, the availability of large quantities of linguistic data, the development and sharing of machine learning algorithms, and deep comprehension of human language structure. Current NLP approaches are employed in downstream tasks such as speech recognition (Graves et al., 2013), question answering (Andreas et al., 2016), machine translation (Johnson et al., 2017), and multi-document summarization (Liu et al., 2018).

A straightforward approach towards genre analysis is to cluster or classify sentences according to their similarity aiming to identify common rhetorical features. Sentences that a human consider having related meanings tend to be grouped or classified with the same label. During the last decade techniques for text clustering and classification such as LDA, KNN, and DBScan have proven to be powerful tools (Rus et al., 2013; Wang and Goutte, 2018; Tan, 2005; Schubert et al., 2017; Ng et al., 2002).

Bag Of Words (BoW) is a technique that allows text comparison and may be used with machine learning techniques such as LDA and achieve strong results (Moghaddam and Ester, 2012) with lower computational cost. For a vocabulary of words, BoW counts the occurrences of words within a document, but the Word order or structure in the document are not considered, therefore the name *bag*. Recent approaches to text comparison employ sentence similarity to compute how close in meaning two phrases are. This is not a trivial task, due to the large number of possible sentences and the ambiguity of words. Current approaches that tackle sentence similarity problem (Kiros et al., 2015; Hill et al., 2016; Devlin et al., 2019) represent sentences as vectors so that operations in the Euclidean Space as the cosine similarity can be undertaken.

Measuring sentence similarity is closely related to similarity of words which also require computational representation. Modern approaches to learning word and sentence representation use Deep Learning models (Peters et al., 2018). Learning is performed by using a vast amount of raw data in an unsupervised fashion in order to generate vector representations considering the context in which the word is set.

Word representation models are based on the Distributional Semantics theory, which claims that words occurring in similar contexts are semantically alike. Distributional Semantics has multiple theoretical origins such as psychology, structuralist linguistics, and lexicography (Bruni et al., 2014). It can be traced back to Harris (1954) proposal of distributional analysis. The author states that semantic similarity between two words is a function of the similarity degree of the contexts in which they occur. As a consequence, their meanings rely mainly on their environments of use (Lenci, 2008; Harris, 1954).

Current sentence representation approaches, in turn, are based on *The Principle of Compositionality Semantic* (Pelletier, 1994). According to the author, the meaning of a whole is merely a function of the meanings of its components and the way in which these components are combined. The words and rules that connect the words are therefore used to measure the similarity between two sentences.

Measuring the similarity of words and sentences is commonly undertaken operations in a vector space such as cosine similarity. Within cosine similarity, the inner product of two vectors is calculated to measure the cosine of the angle between them. Considering X = vector (”King”) - vector (”Man”) + vector (”Woman”), the cosine distance may be employed to search, in the vector space, for the word closest in meaning to X. The expected result for X is vector ("Queen") (Mikolov et al., 2013).

2.3 Related Work

As previously stated, Swales (1990) introduces the CARS model for genre analysis. This framework entails three obligatory *Moves* for research article introductions: Move 1 - Establishing a territory, Move 2 -Establishing a niche, and Move 3 - Occupying the niche. Swales (2004) states that the type or nature of a move may, sometimes, be indicated by grammatical features. For instance, topic generalization in *Move* 1 can be perceived by the use of present continuous tenses; indicating a gap in *Move* 2 may be identified by negative or quasi-negative elements; and the beginning of Move 3 may be characterized by the use of deictics, personal pronouns or the presence of “was to” in the text. Moreover, lexical signals are also presented by Swales (2004), ranging from the most clear ones such as “The main methods used were"to more subtle instances as concluding phrases indicating the end of a *Move*. Despite the fact that Swales’ work has defined and explained the notions of *Moves* and *Steps* in an academic text (and also provided ways to recognize them), their identification in scientific texts is not as transparent as it may seem (Cotos, 2011).

A similar scientific research-article sentence corpus to the current research was proposed by Fisas et al. (2015). They presented an annotated corpus of scientific discourse in the domain of Computer Graphics comprising 40 documents and 10,780 sentences. Each sentence was labelled as belonging to one of 5 categories (Challenge, Background, Approach, Outcome and Future Works) based on *Argumentative Zoning* theory. *Argumentative Zoning* states that a research article can be studied according to its zones (text blocks), which share a rhetorical function and are related to the discourse aim of the overall paper (Teufel, 1999; Liakata et al., 2010). The annotations by the Fisas et al. (2015) corpus was extended by Fisas et al. (2016) adding the purpose for a citation and the relevance of a sentence for a summary. Lauscher et al. (2018) uses Fisas et al. (2016) to study the effect of argumentative components on rhetorical analysis tasks. A neural multi-task learning architecture combined with argument extraction with a set of rhetorical classification was employed. The results showed that rhetorical analysis tasks are positively impacted by argumentative components. Nevertheless, their analysis does not produce any insights that could guide reviewers, editors, and authors paper examination, despite the fact that, compared to the Fisas et al. (2016) corpus in of Computer Graphics (and restricted to 40 documents), more sentences had been annotated. This numbers may still not be enough for ML algorithms.

Another automatic approach that uses NLP and explores the structure of research articles was proposed by Seaghdha and Teufel (2014). It was also based on *Argumentative Zoning* theory and proposes the hypothesis that "the generality of rhetorical language allows the construction of models that can separate out topical and rhetorical language use". By using an unsupervised topic model architecture, they analyzed 129,595 abstracts taken from open-access journal articles. The argumentative zones and most probable words for each zone were identified. The authors stated their study is the first step into inducing templates that could be used by writers. In an analogous way, it would be like the automatic assembly of an article model from sentences of the *Academic Phrase Bank* considering the main rhetorical functions found in the scientific environment.

Yang et al. (2018) propose an automatic academic paper rating that trains an attention-based Convolution Neural Network (CNN) by using accepted and rejected papers from Arxiv[3]. The model aims to generate a high-level representation of papers by dividing them into sections and by using the attention mechanism to aggregate the representations of each section. Paper information such as title, authors, and abstract was included as features for the neural network. The proposed approach may contribute to a better paper examination, yet uses rejected papers that are not commonly found. Moreover, textual patterns frequently used in papers are not explicitly identified; focus is on

[3] https://arxiv.org/

an analysis of the sections that have a greater influence on the acceptance or rejection of papers.

2.4 Dataset

Due to the inexistence of a corpus for genre analysis in software engineering, we compiled a scientific-research sentence (SciSents) corpus. This was conducted in two stages: dataset collection and manual annotation. These stages are detailed in the following subsections.

2.4.1 Dataset Collection

2.4.1.1 Article selection and download

Software engineering was chosen due to the current paucity of studies investigating rhetorical moves in this field. Papers from journals and proceedings listed in the top 13 Google Scholar citation ranking, category Engineering & Computer Science, subcategory Software System[4] were selected to compose the dataset. Publications from 3 journals could not be downloaded so they were not included. Papers from the journals listed in Table 1 were downloaded. The criteria for article inclusion were: English language, publications between the years 2000 and 2018, the presence of a section named Introduction. Any paper which did not meet the following criteria was removed from corpus if it had one page only, if it did not have a Document Object Identifier (DOI), or if it was an editorial, a letter, an erratum, or a Note. By the end of this process, the dataset comprised 9,193 papers (Table 1).

2.4.1.2 Sentence extraction and cleaning

After the research papers were downloaded their contents were extracted and cleaned so as to be used as input in the experiment. Content extraction was made through an algorithm named Science Parse[5]. The algorithm parses research articles (in PDF format) and returns them in a structured way (in JSON format). Its main advantage over an algorithm which simply transforms PDF into text is that it seeks to identify the sections of an article and adds them separately to a JSON file.

Following the extraction process, the transformation process started with a sentence tokenizer that generated 403,895 phrases. Split sentences were analyzed and those that Science Parse misidentified as part of the Introduction section (but in fact, were footnotes incorrectly included) were removed from the corpus. The number of words per

[4] https://scholar.google.com.bitations?view_op=top_venues&hl=EN&vq=eng_softwaresystems (accessed on 2018/09/14)

[5] Science-parse can be accessed in https://github.com/allenai/science-parse/

sentence was also checked and phrases with up to 3 words or with more than 49 words per sentence (up outliers in boxplot) removed. By the end of this process the corpus amounted to 322,630 sentences (Table 1).

Journals and Conference Proceedings	**Papers**	**Sents**
ACM/IEEE International Conference on Software Engineering (ICSE)	2,224	54,300
IEEE Transactions on Software Engineering (TSE)	1,128	46,654
Journal of Systems and Software (JSS)	1,627	64,738
ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE)	464	15,553
ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)	635	26,402
Information and Software Technology (IST)	1,343	46,779
Mining Software Repositories (MSR)	250	6,236
Empirical Software Engineering (ESE)	544	24,347
IEEE/ACM International Conference on Automated Software Engineering (ASE)	424	13,100
Software & Systems Modeling (SoSyM)	554	24,521
Total	**9,193**	**322,630**

Table 1 – Number of papers and sentences per Journal and Conference.

2.4.2 Manual Annotation Procedures

In an attempt to accelerate the manual annotation process, facilitating the find of sentences with the same rhetorical moves, sentences from the corpus were normalized and compared with the academic phrasebank sentences that also were normalized. The normalization was carried out with spaCy library[6], which tokenized, down-cased, and removed stopwords from sentences.The *Moves* and *Steps* categories were defined before the identification of rhetorical moves in the corpus. Academic phrasebank introduction sentences were grouped into 23 sub-classifications. For the purpose of the current experiment, they were deemed as belonging to the *Steps* categories (Swales, 1990).

Manual analysis in the phrasebank normalized sentences showed that short phrases with one or two words were wrongly segmented. These were removed, as they alone did not form complete sentences. Regular expressions were included in the phrasebank sentences so as to increase the probability of finding similar sentences in the corpus. These expressions were considered instead of sentence similarity techniques due to the fact that phrasebank sentences are generic and often incomplete, as some contain letters such as 'X' or 'Y' replacing specific content words. Moreover, sentences with typical terms and expressions within the realm of Software Engineering field could generate noise in a sentence similarity analysis, which could lead to such sentences not being found even if they presented the same rhetorical moves.

[6] https://github.com/explosion/spaCy

Next, a regular expression search in corpus sentences with expressions created from academic phrasebank sentences was undertaken. 2,760 sentences were found, but none corresponded to 7 out of the overall 23 phrasebank *Steps*. Manual analysis was once again performed to ensure the output phrases fell within the same rhetorical moves from all respective academic phrasebank sentences. This validation was carried out by two linguistic experts who were assigned to validate 10 sentences for each *Step*. No sentence could be validated for Step *Outlining the structure of a thesis or dissertation*. Also, it was not possible to validate a minimum of 10 sentences for 8 out of the overall phrasebank *Steps*. This could only be achieved for a total of 7 *Steps*. These results indicated that regular expression search was not able to find all proposed *Steps*, showing that it was inefficient with 30% of the *Steps* as to reaching a minimum of 10 validated sentences.

As som *Steps* categories are not commonly used in papers (e.g. *Outlining the structure of a thesis or dissertation*), and given subtle semantic differences between several *Steps* (with no substantial impact to genre analysis in article Introductions), the 23 phrasebank sub-classifications were narrowed down to 13 *Steps*. This was made with the aid of a linguistic expert who merged, classified and placed these categories across the 3 aforementioned *Moves* proposed by Swales (1990): *Establishing the territory*, *Establishing a niche* and *Occupying the niche* (see Table 4).

Finally, the last procedure was performed with the aid of Prodigy[7], a scriptable annotation tool. During this process, the annotator selected the corresponding *Step* indicating the rhetorical move performed by the sentence. In total, 595 sentences were manually annotated by the linguistic expert. Table 4 presents the number of sentences annotated for each *Move* and *Step*.

2.4.2.1 Corpus Statistics

This section presents some statistics regarding SciSents. As mentioned previously, the corpus was crawled from 10 highly-cited journals and conferences, and has 322,630 sentences from 9,193 papers (see Table 1). Figure 1 presents the distribution of papers by year of publication.

2.5 Experiments

2.5.1 Machine learning techniques

In order to query the corpus and to establish a gold-standard for comparing the results, an experiment was carried out for sentence classification within *Moves* and *Steps*. The following techniques were employed: KNN, LDA, and DBScan. The techniques utili-

[7] https://prodi.gy/

Move	Step	Qty.
Establishing the territory		**187**
	Establishing the importance of the topic for the discipline	37
	Establishing the importance of the topic for the world or society	45
	Establishing the importance of the topic as a problem to be addressed	45
	Referring to previous work to establish what is already known	60
Establishing a niche		**98**
	Identifying and highlighting inadequacies, weaknesses, controversies and negative outcomes within the field of study	45
	Identifying a knowledge gap, a lack of or paucity of previous research in the field of study	53
Occupying the niche		**310**
	Stating the focus, aim, purpose or argument of the current research	44
	Setting out the research questions or hypotheses	36
	Describing the research design and the methods used	47
	Explaining the significance or give reasons for personal interest in the current study	33
	Describing the limitations of the current study	31
	Outlining the structure of a chapter, paper, thesis or dissertation	80
	Explaining Keywords (also refer to Defining Terms)	39
Total		**595**

Table 2 – Number of annotated sentences by *Move* and *Step*

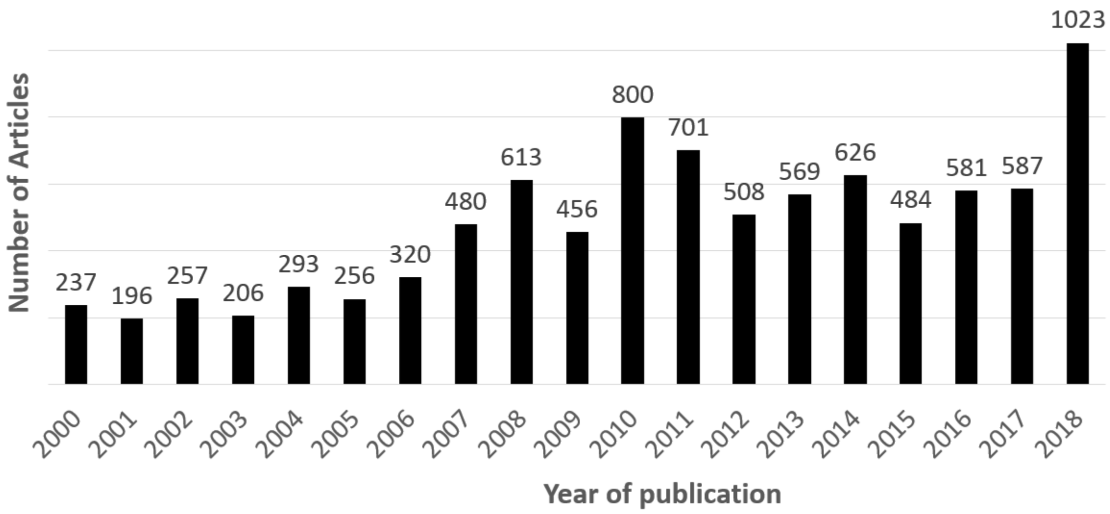

Figure 1 – Articles distribution by year of publication

zed were fully implemented with the use of Python language. KNN implemantation was based on Cover et al. (1967), LDA used the gensim library (Řehůřek and Sojka, 2010), and DBScan used sklearn (Buitinck et al., 2013).

To generate a sentence representation for LDA the BoW model was applied to the corpus sentences. For the other two techniques, sentences were represented as a 1024-position vector using BERT as a Service (Xiao, 2018) based on BERT (Devlin et al., 2019). In these techniques, the cosine similarity was used for sentence comparison.

2.5.2 Settings and Metrics

KNN training was carried out separately for *Moves* and *Steps*. For each case, a subset of annotated sentences was created so that the number of sentences in each *Move* and *Step* was balanced, i.e. the maximum number of elements in a *Move* or *Step* could not be greater than the minimum number of sentences in the *Move* or *Step* set with the least number of items. Thus, the *Move* training subset amounted to 294 sentences, since the one with the fewer number of elements was *Establishing the niche*, totalling 98 sentences. The subset of *Steps* had 403 sentences, since the one with the least amount of elements was *Describing the limitations of the current study*, with 31 sentences. Three training epochs were carried out and in each round subsets were divided into balanced test and training sets in the proportion of 33%/67%, 50%/50%, and 67%/33%. In cases where the division was not exact, rounding was performed. The K value used in KNN varied from 3 to 98 for the *Move* and from 3 to 31 for the *Step* training.

The training of LDA and DBScan ran through all corpus sentences. For LDA, sentences were tokenized and down-cased, stopwords were removed with spaCy library[8], and the BoW model was applied. As the number of *Steps* found could be higher than the 13 *Steps* devised for our framework shown in Table 1, a range from 13 to 26 clusters were tried for LDA training. An additional training with 3 clusters was performed to analyze the *Moves*. For all topics quantity 10 iterations were utilized. For DBScan, an epsilon ranging from 0.022 to 0.072 and 23, 46 and 92 minumum points were tried. With the use of cosine similarity the DBScan clustered sentences according to their semantic proximity.

There is no direct evaluation metric to determine whether an unlabelled sentence belongs to a specific category. Thus, to analyze technique performance the purity of labels considering the *Steps* and the *Moves* was measured. Purity retrieves the frequency of the most prevalent category into each cluster. The larger the purity value, the more productive the clustering solution (Zhao and Karypis, 2001)

2.6 Results

Purity calculation was based on manually annotated sentences, but except for KNN all other techniques grouped the sentences based on entire corpus calculus. KNN in turn classified sentences based on training set sentences. The most productive results obtained for each technique are presented in Table 3.

[8] https://github.com/explosion/spaCy

Technique	**Step Purity**	**Move Purity**
DBScan	0.1635	0.4714
KNN	0.5000	0.7879
LDA	0.2758	0.4945

Table 3 – Purity reached by each technique

The experimental results show significant difference between the three techniques regarding *Step* purity and *Move* purity. Results point that KNN achieved best score in both analyses. In *Steps* the purity was 0,5000 for a k of 16, whereas in *Moves* purity was 0.7879 for a k of 17. This sheer discrepancy between *Step* and *Move* results may be explained by the fact that the concept of *Move* is at an overall level divided into only 3 main categories. Conversely, the concept of *Step* goes deeper into a more fine-grained explanation as to how the rhetorical means are used to execute the *Move* function. This is the reason why they were classified into 13 distinct categories.

When looking only into clustering techniques we found that even LDA using BoW did not take the order or structure of words into account, and for this reason it presented better results than DBScan using BERT (Devlin et al., 2019), which is a state-of-the art model. An important difference between LDA and DBScan is that in the former the number of clusters is predetermined whereas in the latter the calculation is done by the algorithm itself. LDA *Step* purity reached 0.2758 for 14 clusters while DBScan reached a purity of 0.1635 for 7 clusters. LDA *Move* purity achieved 0.4945 for 3 clusters whereas DBScan reached 0.4714 for 5 clusters.

2.7 Conclusion and Future Work

This paper presented a gold standard corpus for genre analysis in software engineering research articles. The corpus was compiled from 10 highly cited journals and conference proceedings comprising 322,630 sentences and 565 annotated sentences. Training was performed by means of three machine learning techniques using sentence similarity, namely K-Nearest Neighbors, Latent Dirichlet Allocation and DBScan. KNN presented the highest purity scores for both *Moves* and *Steps*. Regarding future work, supervised machine learning techniques may be used to detect rhetorical patterns within a larger number of annotated sentences.

3 Comparison of Supervised Machine Learning Techniques for Genre Analysis of Software Engineering Research Articles

3.1 Introduction

Written communication plays a fundamental role in scholarly development. This importance is evidenced by the large number of estimated publications and journals (Larsen and von Ins, 2010; Björk et al., 2008; Mabe, 2003). To ensure this great volume of scientific publications is academically sound, a sufficient number of reviewers becomes crucial. However, the lack of such professionals to meet all the existing demand (Fox, 2017) poses a challenge to various research communities. To address this issue, computational approaches have been purportedly implemented to automatically query linguistic segments in research articles in order to indicate the presence or absence of commonly used rhetorical patterns.

Scientific publications have multiple and somewhat standardised conventions such as preference for the passive voice, paper division in sections, and use of lexical and phrasal structures to indicate the function executed by each part of the text (Seaghdha and Teufel, 2014), to cite a few. Disseminated linguistic work on scientific publications is the CARS (Create a Research Space) model proposed by Swales (1990). CARS approaches genre analysis by introducing two concepts, namely *Moves* and *Steps*. While 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 used to perform the function of a *Move* (Ruiying and Allison, 2003).

Automatic approaches such as Support Vector Machines (SVM) (Bennett and Demiriz, 1999; Tang et al., 2007) may be employed to compute 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, and the existing ones have limited amount of data (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 problem, SVM can be used as a semi-supervised approach, in which considerable amounts of unlabeled data are utilized with labeled data to form more solid classifiers (Zhu, 2005).

This work aims to evaluate supervised and semi-supervised machine learning techniques that automatically retrieve rhetorical patterns within Swales' CARS genre analysis model in research articles. This investigation was carried out into SciSents[1] 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 the F-Scores generated by supervised approaches for genre analysis in research articles. In this respect, the results of experiments of SVM and logistic regression techniques are presented.

The paper proceeds as follows. First, the corpus and the semi-supervised annotation procedures are presented. Next, the techniques and their features are discussed in a comparative fashion. Following the experiments, included implementation details are addressed and, finally, the results are described.

3.2 Data

A genre analysis experiment was conducted in SciSents, a dataset of research article sentences. This data resource is based on 9,193 software engineering articles published between 2000 and 2018 in highly-cited journals and conference proceedings. The corpus consists of 322,630 sentences from the Introduction sections, in addition to 595 annotated sentences in 13 *Steps* and 3 *Moves* (see Table 4) in SciSents and based on Swales' CARS model (Swales, 1990).

To increase the number of annotated sentences in SciSents, a semi supervised strategy with SVM was employed. 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 3.1.

Two different embeddings were generated. The first consisted of generating corpus sentence representation individually and the second a representation in conjunction with the previous sentence. The purpose of this second approach was to investigate whether the previous sentence had an influence on the genre analysis of the subsequent one. In the cases where the sentence was not preceded by any other, the representation was calculated with that sentence solely. It is important to highlight that the previous sentences were not necessarily the immediately preceding ones since some invalid sentences were removed from SciSents during the preprocessing stage.

The embedded sentences were the input of training and phrase labels were utilized to perform a 5-fold cross-validation. To evaluate the performance of the semi-supervised process three measures were employed: Precision, Recall, and F-score. Precision measures the proportion of correct classified sentences out of the total number of annotated sentences, while Recall estimates the proportion of correctly annotated sentences out of the

[1] Avaliable on: ANONYMOUS

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 4 – Manual annotated sentences number by *Move* and by *Step* in SciSents (SS), semi supervised Round 1 (R1) and Round 2 (R2).

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). The results of F-Score for each *Step* with the two different embedding representations are presented in Tables 6 and 7.

Following the annotation stage, the probability of the corpus sentences belonging to each one of the 13 *Steps* in SciSents was computed using the sentence embedding with one sentence solely. The 20 most likely sentences for each *Step* (260 in total) were manually checked by a linguistic expert with considerable knowledge about Swales' CARS model. Through this analysis we identified that 228 sentences were correctly classified whereas 5 sentences were incorrectly classified. 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 5 incorrect classified ones that were corrected, which amounted to a total of 828 manually annotated sentences. A new SVM 5-fold cross validation was administered with this annotated set. The results of F-Score are also presented in Tables 6 and 7.

A second round of semi-supervised annotation followed. This time the linguistic expert analysed random sentences with different probabilities for *Steps* calculated by a trained SVM considering embeddings generated from corpus sentence individually. Table 5 shows in each column the number of sentences with a higher probability than that indicated in the column title, with a difference of 10% in each column.

A total of 481 sentences were manually checked, of which 308 were considered as being correctly classified and 173 were deemed as being 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 as shown in Table 4. Once again a SVM 5-fold cross validation was performed. The results of F-Score are also presented in Tables 6 and 7.

Step	>90%	>80%	>70%	>60%	>50%	>40%	>30%	>20%	>10%
M1-S01	2	69	384	1149	2781	5850	10511	15545	16781
M1-S02	6	162	660	1692	3675	7076	11882	16741	17773
M1-S03	269	1428	3387	6217	10326	16158	23687	31631	33384
M1-S04	480	2432	5819	10895	18403	29812	47161	66291	70053
M2-S05	6	145	771	2238	5094	9953	16808	23865	25322
M2-S06	68	188	337	527	797	1214	1910	2831	3098
M3-S07	38	364	1031	2172	3906	6488	10091	13477	14153
M3-S08	60	357	813	1442	2379	4006	6989	11142	12294
M3-S09	101	884	2837	5943	11254	19772	31806	43718	46109
M3-S10	10	155	708	2078	4539	9262	17364	26874	28863
M3-S11	1	27	70	128	239	474	977	1770	2064
M3-S12	20486	24488	27306	29714	32091	34818	37955	40876	41519
M3-S13	167	453	861	1442	2367	3817	6328	9239	9908
Total	21694	31152	44984	65637	97851	148700	223469	304000	321321

Table 5 – Number of predicted sentences per probability set calculated considering embeddings generated from individual corpus sentences

3.3 Machine Learning Techniques

We have recognized the genre analysis issue as a supervised problem using SVM, logistic regression techniques and BERT (Devlin et al., 2019), and Universal Sentence Encoder (Cer et al., 2018a) sentence embedding techniques. Each pair of technique-embedding type ran over every annotated set: SciSents, semi-supervised round 1 and semi-supervised round 2. The pair SVM-BERT was employed as the basis for a semi-supervised annotation and used for comparison with the rest of the experiment.

3.3.0.0.1 SVM:

Support Vector Machines are non-parametric and deterministic algorithms based on statistical learning. They have been broadly used in several fields, 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 creating a boundary between distinct classes (Hearst et al., 1998; Joachims, 1998).

3.3.0.0.2 Logistic Regression:

Logistic regression is a statistical technique for binary classification which can also be used to multi-class classification by treating these as several binary classification problems (Ifrim et al., 2008). It computes classes probabilities using a logistic function and constructs a linear hyperplane separating the classes.

3.3.1 Features

The two different vector features representations used to train the techniques are described next:

3.3.1.0.1 Universal Sentence Encoder:

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

3.3.1.0.2 BERT:

Bidirectional Encoder Representations from Transformers or BERT (Devlin et al., 2019) is a masked-language model to text representation. It is composed of multi-layer bidirectional transformer encoder that pre-trains in a large unlabeled text corpus and has two objectives: masked language modeling and next sentence prediction. A random sample of the tokens is masked (replaced with the special token), the next sentence is predicted and BERT continues with the training and optimization until it obtains satisfactory results (Liu et al., 2019).

3.4 Experiment

3.4.1 Implementation

The techniques utilized were fully implemented with the use of the Python language. Each technique was trained using 5 fold cross-validation and averages across F-Score results on test folds were reported. SVM and logistic regression uses sklearn library (Buitinck et al., 2013). SVM uses linear kernel and C = 1.

For each technique two different combinations of sentence embedding features were explored: Universal Sentence Encoder and BERT. As to the former, a TensorFlow implementation[2] (Cer et al., 2018b) was used and generated a 512-dimensional sentence embedding vector. Concerning the latter, BERT as a Service (Xiao, 2018) was employed and a 1024-position vector was generated.

3.4.2 Results

We report the F-score averaged over the folds of our techniques in Tables 6, 7, 8 and 9. Each table column shows the result of an experiment type that is composed of 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 6 summarizes the results of the experiments on *Steps* when using one sentence solely to generate the embeddings. With all but 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 better scores. In 6 times of these the highest results were achieved in the semi-supervised annotation round 2 (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, and M3-S10-Explaining the significance or give reasons for personal interest in the current study). In the remaining 5

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

Steps (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, and 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. All results in Table 6 for M3-S12 were higher than 0.84. This result can be explained by the fact that sentences within this *Step* are very prototypical as *"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 6 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 more detailed information juxtaposed in a focus-plus-context oriented format."*, and *"The backward analysis computes an over approximation of all possible inputs that can generate those attack strings."*. Throughout annotations rounds, M3-S10 improved its results reaching a still low performance of 0.4092. The best performance in Table 6 for M3-S10 scored 0.5081 when using the pair SVM-BERT.

Table 7 shows the performance of the experiments on *Steps* when using both actual and previous sentences to generate the vector representation. The pair logistic regression with BERT beats 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 produced in M3-S10.

We notice that results shown in Table 6 are more productive than results in Table 7

Step	**SVM-BERT**			**SVM-USE**			**LR-BERT**			**LR-USE**		
	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

Table 6 – 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 the vector representation (LR = Logistic Regression; USE = Universal Sentence Encoder). The strongest F-score in each row is in bold. (Acho que esta ultima informação deve estar numa nota de rodapé)

in 44 (or 56.41%) out of 78 when considering the experiments that used BERT in isolation. When analysing only the best scores for each *Step* in both tables, Table 1 has best results in 8 (61.53%), while Table 2 has the best results in 4 (30.77%) out of (quantos?) cases. There was a draw in one case. Surprisingly, the performance when using universal sentence encoding was the same in both tables.

Step	**SVM-BERT**			**SVM-USE**			**LR-BERT**			**LR-USE**		
	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

Table 7 – Experiment results by *Steps* on SciSents (SS), semi supervised Round 1 (R1) and Round 2 (R2) annotated sentences sets using the sentence in conjunction with the previous sentence to create the vector representation (LR = Logistic Regression; USE = Universal Sentence Encoder). The strongest F-score in each row is in bold.(Acho que esta ultima informação deve estar numa nota de rodapé)

Table 8 summarizes the results of experiments on *Moves* when using one sentence solely to generate the embeddings. 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 in 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 0.0187 between *Moves* M1, M2 and M3 respectively.

Move	**SVM-BERT**			**SVM-USE**			**LR-BERT**			**LR-USE**		
	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

Table 8 – 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 the vector representation (LR = Logistic Regression; USE = Universal Sentence Encoder). The strongest F-score in each row is in bold. (Acho que esta ultima informação deve estar numa nota de rodapé)

Table 9 presents experiments results on *Moves* when the vector representation is created using the actual sentence in conjunction with the previous sentence. As in 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 8 with those from Table 9 we can notice that the figures on the former surpass all respective results on the latter when considering BERT alone. Once again, when Universal Sentence Encoder was used there was no difference between the embedding from one sentence alone and from a sentence in conjunction with its previous one.

Move	**SVM-BERT**			**SVM-USE**			**LR-BERT**			**LR-USE**		
	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

Table 9 – Experiment results by *Move* on SciSents (SS), semi supervised Round 1 (R1) and Round 2 (R2) annotated sentences sets using the sentence in conjunction with the previous sentence to create the vector representation (LR = Logistic Regression; USE = Universal Sentence Encoder). The strongest F-score in each row is in bold.(Acho que esta ultima informação deve estar numa nota de rodapé)

When analysing results evolution throughout the annotation process within each experiment type we can notice that they did not always improve accordingly. In Table 6, when we compare SciSents with annotations from Round 1 only in one situation (1,92% of the total), there was no improvement in the F-score. 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 that the techniques already knew were included, whereas in Round 2 sentences which were different from those the techniques knew (but still belonged in that *Step*) were included.

When approaching annotation evolution throughout Table 7 we observe that experiments within Round 1 annotations outperformed experiments in SciSents for 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 8 and 9 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 8), but some improvement in 33,33% of the overall cases, as we see in Table 9. These results indicate that the second round of annotation may have included sentence types unknown to the technique.

3.5 Discussion

The present study was designed to augment the number of annotations in SciSents corpus and to compare the results of supervised machine learning techniques for genre analysis in software engineering research articles. The number of annotated sentences was increased from the 595 in SciSents to 1309 through two semi-supervised rounds using SVM. Although most of the best results for *Steps* were achieved with the second round annotation set, this did not always happened, thus indicating the need for more annotations for different probabilities which potentially fall within the *Steps* categories. These annotations may also contribute to genre analysis regarding *Moves*, whose results have barely evolved with the second round of annotation.

One interesting finding in the experiments in supervised machine learning techniques is that the use of sentence embedding generated from the sentence alone was better, in most cases, than that with the use of the actual sentence together with its preceding one. Another important finding is that the vector representation provided by BERT delivered better results than Universal Sentence Encoder into the tested sets. Finally, logistic regression mostly provided better F-score than SVM into the tested sets.

3.6 Related Work

Anthony and Lashkia (2003) work was based on Swales' theory and proposed a computer software tool which automatically identifies the structure of a research article. The tool is named *Mover* and aims to present to learners a panorama of the move structure utilized in the RA. It was tested in 100 information technology articles abstracts with 692 sentences. The abstracts were manually annotated based on a Modified Create a Research Space (CARS) model proposed by Anthony (1999). The model is composed of Swales' (1990) 3 *Moves*, as well as 12 *Steps*. As if is a general model created for introductions and due to the small size of the base, not all steps appeared in the dataset. A modified bag of words was utilized to represent the text in a way that 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. Because of that, they named the model *Bag of Clusters*. As well as allowing the system to identify steps which are only possible to classify if the preceding or later *Steps* are known, an additional "location" feature was added to the bag of clusters model. The model's output feeds a Naive Bayes classifier that performed consistently with an average accuracy of 68%. The accuracy varied by *Step* ranging from 17% (Indicate gap) to 92% (Announce research). The authors justified the poor result of some classes by the scare training items from these *Steps*. Through error analysis, the authors observed that when the software presented flaws, the *Step* incorrectly categorised tended to fall within the same *Move*. To improve the accuracy of the system the two most probable classifications were used in a second experiment. In this case, the user had to select the most appropriate option. With this procedure, accuracy achieved 86%. Despite the productive results, the reduced number of articles and sentences was a hindrance for deeper analyses.

Pendar and Cotos (2008) proposed the development of a pedagogical tool which automates discourse evaluation. The tool's goal is to appraise academic writing drafts in agreement with an adapted model based on Swales, to compare it with other papers in the same discipline and to provide feedback to the student. To develop such a tool, a text-categorization approach using Suport Vector Machine (SVM) to classify sentences from research article introductions drawn on Swales' rhetorical moves was employed. To check the tool's performance, an experiment with a corpus called Intelligent Academic Discourse Evaluator (IADE) composed of published research articles from 20 academic disciplines was executed. The dataset consisted of 401 Introduction sections and 11,149 sentences. These sentences were manually annotated according to the CARS model's *Moves*. The *Steps* were suggested by the authors. Due to the sparseness of data, the study did not attempt to automatically classify the *Steps* within the *Moves*. To execute the classification in SVM, sentences were stemmed and represented in a n-dimensional vector (word unigrams, bigrams and trigrams). Experiment results were very encouraging (with an accuracy above 70%), but the dataset was relatively small.

Cotos and Pendar (2016) made progress on their own 2008 work by increasing IADE's size to 1,020 research articles across 51 disciplines. Sentences were also annotated according to CARS model, but this time including both *Moves* and *Steps*. SVM was also employed and sentences were represented as an n-dimensional vector of word unigrams and trigrams. The classifier 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 as an automated approach to examine rhetorical structure in student writing. The architecture comprised an embedding layer, a sentence-level recurrent layer, and a document-level recurrent layer. 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 a better word representation. Each word embedding generated by this layer was fed to sentence-level layer made up of two separate LSTMs: one forwards and one backwards. The result of each LSTM was later concatenated to generate the full sentence embedding and serve as input for the document-level layer. This one, in turn, was also composed of forward and backward LSTMs generating new sentence embeddings which encoded not only sentence level features but also inter-sentence information. CRF layer, in turn, used sentence embeddings to retrieve the influence of past and future tags in order to measure transition probabilities for each tag individually and for each tag in combination with another. Two datasets were employed to test the model: IADE (Pendar and Cotos, 2008; Cotos and Pendar, 2016) and Research Writing Tutor (RWT) comprised of 900 full research articles (and not only Introduction sections) across 30 academic disciplines. RWT was manually annotated and a sentence could be labeled with several steps if it had one communicative goal and more than one functional strategy; 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 a recall of 77%, and an F1-score of 76% for the classification task in RWT dataset.

3.7 Conclusion

The present study was designed to compare supervised machine learning techniques which automatically retrieved linguistic segments from research articles. Firstly, a semi-supervised approach was used to increase the number of annotated sentences in SciSents corpus. Next, two supervised techniques and two sentence embedding techniques were employed to execute genre analysis on the dataset. Our results suggest that an approach based on logistic regression and BERT can perform very satisfactorily for genre analysis. In addition, the semi-supervised annotation process has proven to contribute to the annotation process, but needs to include elements with random probabilities so as to consistently improve the technique.

As future work, the semi-supervised annotation process and the techniques hereby described could be used in other sections of software engineering research articles. In addition, these same analyses could be performed on articles from other disciplines to compare the differences between these fields within the scientific genre.

4 Conclusions

The objective of this thesis was to compare the F-score of supervised machine learning techniques for automatic genre analysis of software engineering research articles in the English language. Two different machine learning models were compared: SVM and logistic regression. Each model was training with two different sentence embedding techniques: BERT and universal sentence encoder. The first specific objective of this work was "create a sentence corpus of software engineering research papers for genre analysis in the English language"and was achieved with the creation of SciSents (chapter 2) that is composed of 322,630 sentences. The second specific objective was "expand the sentence corpus using a semi-supervised approach for annotaion"and has been achieved using SVM and with the manually checking of linguistic expert. 1,309 sentences were annotated and the probabilities of other corpus sentences belonging to the annotated labels was calculated. The third and final specific objective was "compare the F-score of supervised machine learning techniques for automatic genre analysis of software engineering research articles in the English language"and was achieved with the results section in chapter 3.

Comparison among techniques shows that logistic regression with BERT can perform very well in genre analysis outperforming other techniques with an average of 84.22 on f-score when querying linguistic elements on an overall level. In addition, the semi-supervised annotation process has proven to contribute to the annotation process.

The SciSent corpus was made publicly available on the internet and may be used by future investigations focused on genre analysis for the English language. We believe that the current research may provide improvements to the scientific publishing process helping editors, reviewers and also authors when writing or analysing research articles.

References

Andreas, J., Rohrbach, M., Darrell, T., and Klein, D. (2016). Learning to Compose Neural Networks for Question Answering. In *Proceedings of NAACL-HLT*, pages 1545–1554.

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

Anthony, L. and Lashkia, G. V. (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.

Basturkmen, H. (2012). A genre-based investigation of discussion sections of research articles in Dentistry and disciplinary variation. *Journal of English for Academic Purposes*, 11(2):134–144.

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

Bhatia, V. (2014). *Worlds of Written Discourse: A Genre-Based View.* Bloomsbury Academic, London, reprint edition edition.

Bhatia, V. K. (1993). Analysing Genre: language use in professional settings.

Björk, B.-C., Roosr, A., and Lauri, M. (2008). Global Annual Volume of Peer Reviewed Scholarly Articles and the Share Available Via Different Open Access Options. page 9.

Bornmann, L. and Mutz, R. (2015). Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references: Growth Rates of Modern Science: A Bibliometric Analysis Based on the Number of Publications and Cited References. *Journal of the Association for Information Science and Technology*, 66(11):2215–2222.

Bruni, E., Tran, N.-K., and Baroni, M. (2014). Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49:1–47.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., and Varoquaux, G. (2013). API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122.

Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., and Tar, C. (2018a). Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., St. John, R., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Strope, B., and Kurzweil, R. (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.

Chowdhury, G. G. (2003). Natural language processing. *Annual review of information science and technology*, 37(1):51–89.

Cotos, E. (2011). Potential of automated writing evaluation feedback. *Calico Journal*, 28(2):420–459.

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

Cover, T. M., Hart, P. E., and others (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (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.

Fernández-Delgado, M., Cernadas, E., Barro, S., and Amorim, D. (2014). Do we Need Hundreds of Classifiers to Solve Real World Classification Problems? *Journal of Machine Learning Research*, 15:3133–3181.

Fiacco, J., Cotos, E., and Rosé, C. (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.

Fisas, B., Ronzano, F., and Saggion, H. (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.

Fisas, B., Saggion, H., and Ronzano, F. (2015). On the discoursive structure of computer graphics research papers. In *Proceedings of the 9th linguistic annotation workshop*, pages 42–51.

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

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

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

Graves, A., Mohamed, A.-r., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE.

Hames, I. (2008). *Peer review and manuscript management in scientific journals: guidelines for good practice*. John Wiley & Sons.

Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3):146–162.

Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., and Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28.

Hill, F., Cho, K., and Korhonen, A. (2016). Learning Distributed Representations of Sentences from Unlabelled Data. In *Proceedings of NAACL-HLT*, pages 1367–1377.

Hirschberg, J. and Manning, C. D. (2015). Advances in natural language processing. *Science*, 349(6245):261–266.

Horn, C., Manduca, C., and Kauchak, D. (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.

Hyland, K. (2006). *English for academic purposes: An advanced resource book*. Routledge.

Ifrim, G., Bakir, G., and Weikum, G. (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.

Iyyer, M., Manjunatha, V., Boyd-Graber, J., and Daumé III, H. (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.

Joachims, T. (1998). Text categorization with Support Vector Machines: Learning with many relevant features. In Nédellec, C. and Rouveirol, C., editors, *Machine Learning: ECML-98*, Lecture Notes in Computer Science, pages 137–142, Berlin, Heidelberg. Springer.

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., and Corrado, G. (2017). Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Jordan, R. R. (1997). EAP and study skills: definitions and scope. In *English for Academic Purposes: A Guide and Resource Book for Teachers*, Cambridge Language Teaching Library, pages 1–19. Cambridge University Press.

Kanoksilapatham, B. (2005). Rhetorical structure of biochemistry research articles. *English for Specific Purposes*, 24(3):269–292.

Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.

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

Lauscher, A., Glavaš, G., Ponzetto, S. P., and Eckert, K. (2018). Investigating the Role of Argumentation in the Rhetorical Analysis of Scientific Publications with Neural Multi-Task Learning Models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3326–3338, Brussels, Belgium. Association for Computational Linguistics.

Lenart, J. B. and Berdanier, C. G. (2017). Optimizing a genre analysis framework to investigate engineering literature reviews. In *2017 IEEE International Professional Communication Conference (ProComm)*, pages 1–6. IEEE.

Lenci, A. (2008). Distributional semantics in linguistic and cognitive research. *Italian journal of linguistics*, 20(1):1–31.

Liakata, M., Teufel, S., Siddharthan, A., and Batchelor, C. (2010). Corpora for the conceptualisation and zoning of scientific papers. *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10).*

Liu, P. J., Saleh, M., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L., and Shazeer, N. (2018). Generating Wikipedia by Summarizing Long Sequences. *International Conference on Learning Representations.*

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

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

Maswana, S., Kanamaru, T., and Tajino, A. (2015). Move analysis of research articles across five engineering fields: What they share and what they do not. *Ampersand*, 2:1–11.

Mikolov, T., Yih, W.-t., and Zweig, G. (2013). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751.

Moghaddam, S. and Ester, M. (2012). On the design of LDA models for aspect-based opinion mining. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 803–812.

Morley, J. (2018). Academic phrasebank. Technical report, University of Manchester.

Ng, A. Y., Jordan, M. I., and Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856.

Pelletier, F. J. (1994). The principle of semantic compositionality. *Topoi*, 13(1):11–24.

Pendar, N. and Cotos, E. (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.

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

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237.

Provenzale, J. M. and Stanley, R. J. (2006). A systematic guide to reviewing a manuscript. *Journal of nuclear medicine technology*, 34(2):92–99.

Reller, T. (2016). Elsevier publishing - a look at the numbers, and more. *Key journal*.

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

Rus, V., Niraula, N., and Banjade, R. (2013). Similarity Measures Based on Latent Dirichlet Allocation. In Gelbukh, A., editor, *Computational Linguistics and Intelligent Text Processing*, Lecture Notes in Computer Science, pages 459–470, Berlin, Heidelberg. Springer.

Schubert, E., Sander, J., Ester, M., Kriegel, H. P., and Xu, X. (2017). DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems (TODS)*, 42(3):19.

Seaghdha, D. O. and Teufel, S. (2014). Unsupervised learning of rhetorical structure with un-topic models. page 12.

Swales, J. (1990). *Genre analysis: English in academic and research settings.* Cambridge University Press.

Swales, J. M. (1981). *Aspects of article introductions.* Number 1. University of Aston.

Swales, J. M. (2004). *Research genres: Explorations and applications.* Cambridge University Press.

Tan, S. (2005). Neighbor-weighted K-nearest neighbor for unbalanced text corpus. *Expert Systems with Applications*, 28(4):667–671.

Tang, F., Brennan, S., Zhao, Q., and Tao, H. (2007). Co-tracking using semi-supervised support vector machines. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE.

Teufel, S. (1999). *Argumentative zoning: Information extraction from scientific text.* PhD Thesis, Citeseer.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, \\., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Voight, M. L. and Hoogenboom, B. J. (2012). Publishing your work in a journal: understanding the peer review process. *International journal of sports physical therapy*, 7(5):452.

Wang, Y. and Goutte, C. (2018). Real-time Change Point Detection using On-line Topic Models. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2505–2515.

Ware, M. and Mabe, M. (2015). The STM report: An overview of scientific and scholarly journal publishing.

Xiao, H. (2018). bert-as-service.

Yang, P., Sun, X., Li, W., and Ma, S. (2018). Automatic Academic Paper Rating Based on Modularized Hierarchical Convolutional Neural Network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 496–502, Melbourne, Australia. Association for Computational Linguistics.

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

Zhao, Y. and Karypis, G. (2001). Criterion functions for document clustering: Experiments and analysis. Technical Report, Department of Computer Science and Engineering University of Minnesota.

Zhu, X. J. (2005). Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences.

Řehůřek, R. and Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA.