UNIVERSITY COLLEGE ROOSEVELT
Lange Noordstraat 1, Middelburg, The Netherlands

ACADEMIC INTERNSHIP REPORT

SCICAPS250

Computational Text Analysis using R

Author:
Ivo Verhoeven
A Science Major completing tracks in Computer Science, Mathematics, and Statistics.

Student Number:
6130615

July 9, 2019

Structured Abstract

Background

Reproducibility is by necessity a central tenet of science [38]. Despite this, the importance of replication is remains a point of debate, prominent in the behavioural sciences [18], but also in computer science [13] [41]. As part of an academic internship, this project aims to replicate a prominent paper in an effort to contribute to a developing field, while providing an excellent didactic exercise.

Aims and Objectives

The aim of this project was to replicate the seminal 2011 paper by Athar [5]. Additionally, this work would be extended by including a second feature set, developed by Abu-Jbara, Ezra and Radev and Jha et al. [2][28]. Lastly, all analyses, data and code would be made publicly available to support replication and the development of the field.

Methods and Deliverables

In order to fulfil the described project aims, several weeks of training had to be completed. These were related to developing a working understanding of R, common NLP techniques and supervised machine learning architecture. The source code and data set was obtained from Athar's Github and through contact with Rahul Jha. The source code was analysed, emulated in R using a variety of packages, and then compared.

Results and Interpretation

The processing pipeline as described by the authors was recreated as faithfully as possible before re-running a Naive Bayes and Support Vector Machine model. Using the macro-F1 evaluation metric, these replication models came close to those initially reported. In the case of the Naive Bayes model, significant improvement was obtained over those used by Athar. The extension of the feature set did not generate the desired improvement.

Conclusions and Future Work

This report is the first formal and first successful informal replication of Athar's work. Furthermore, this work has taken the code from Java and managed to emulate its functionality in R. Further work should aim to include citation context or apply more sophisticated models to the detection of sentiment.

[1]

Declaration

I, Ivo Verhoeven, declare the following:

This report is my own work. I performed the work as reported. Quotations are properly attributed. The work reported on was performed to fulfil the requirements of an academic internship at the University College Roosevelt.

[1] This is a standard declaration format.

Acknowledgements

I would like to thank my internship supervisors Dr. Andrew Brooks and Dr. Ir. Richard van den Doel for their guidance and support in this project.

Furthermore, I like to thank Dick, Eveline and Laney for their unfaltering support during this project, and all the times outside of it. I would like to congratulate Vera, and hope she discovers the same curiosity and eagerness as I have enjoyed during my tertiary education. I would like to thank Luigi, for being a steadfast companion in the long nights before deadlines.

Lastly, I would like to thank Rahul Jha and Amjad Abu-Jbara for taking the time out of their day and sharing with me the resources I needed to continue this project.

Contents

List of Tables

1 Introduction

1.1 Project Aim

The primary aim of this project was to replicate a study within computational text analysis using R. To achieve this, a number of weeks needed to be dedicated to training and developing a working understanding of natural language processing in general, and the specific R packages that facilitated analyses within the domain.

Eventually it was chosen to replicate a paper that applied sentiment analysis to scientific citations; one in a series leading to a final Ph.D. dissertation regarding the application of NLP to bibliometrics [6]. This replication involved reusing a Java program and attempting to emulate the broad functionality of the WEKA package within R. Additionally, a second paper was used in an attempt to extend the original analysis [2]. The addition of a second feature set, those not discussed in the originally replicated study, did not have a significant impact on model predictive power.

1.2 Project Plan and Changes

Originally, the project proposal divided this internship into three phases: a) becoming familiar with the R language and the Rstudio environment (˜1 week), b) becoming familiar with common NLP tasks and methods, with particular focus on R implementations of those (˜1 week) and c) finding and replicating the findings of published research after which one more additional analyses needed to be undertaken (˜3 weeks). Due to familiarity with R from a previous course, the first phase could be completed within 2 days as opposed to a whole week. The chosen text processing package `quanteda` required significant amounts of training, given the general incompatibility of it and other necessary packages. Furthermore, one of the project stakeholders gave a series of training briefings on software quality assurance - see Section 3.4. Therefore, while the plan deviated slightly, I remained on track for the replication study.

1.3 Technology Platform

The specification of the computer used for emulating and testing the methods described by Athar [6] and Jha et al. [28] were as follows: **operating system** 64-bit Windows 7 Home, **main memory** 8 GB, **hard drive** 256 GB (Solid State Drive), **processor** Intel Core i7-8550U CPU @ 1.80 GHz, and **number of cores** 8.

2 Work Setting and Project Stakeholders

2.1 Work Setting

This academic internship was conducted at the University College Roosevelt, an honours college of University Utrecht in Middelburg, the Netherlands. While accredited as a course in the 2019 spring semester, the actual work was conducted from 27/05/2019 to 05/07/2019. Work was full-time, 8 hours each week day, with the exception of two days of national holidays (Ascension Day and Whit Monday). An individual office was not provided and interns were accommodated in a college classroom in the ground floor of the Eleanor building. Interns were required to use their personal laptop computers, though the college was ready to provide additional laptop computers if required.

Mission statement
University College Roosevelt is a residential honours college of University Utrecht that emphasises the students' academic excellence, personal growth and civic responsibility. Classes are kept small, allowing for intensive contact between the professors and their pupils. No mandatory curriculum is set, besides a handful of requisite courses, but students are held responsible for their own learning and are expected to excel in all their chosen courses. The diverse array of fields offered is matched by the diverse collection of nationalities and cultures that come to the small, medieval town of Middelburg.

Brief History
Previously called Roosevelt Academy or RA, it was founded by Hans Adriaanssens in 2004. Prof. Adriaanssens also served as founding dean to our sister institution, the University College Utrecht. At the time, a general dissatisfaction existed towards standard tertiary education, where programs were often too impersonal, and did not provide the challenge students required. The new Liberal Arts and Sciences programs would embody the principles of being small-scale, intensive undergraduate program, that encouraged the *Artes Liberales*. The College is named after the Roosevelt family, prominent still in Zeeland [39].

While all the courses necessary for a true liberal arts education are already present under the three departments, namely the arts and humanities, social sciences and sciences, within the coming years an engineering department is expected to be set up as well. In continuously adapting, taking advantage of its unique location and upholding the precedent of high quality teaching, University College Roosevelt hopes to remain at the forefront of Liberal Arts and Sciences education within both the Netherlands and Europe.

For more information regarding the college, please consult the website [40].

Size, Organization and Funding
As of 2016, UCR receives 6.7 M from a mixture of government loans and tuition fees. There currently exists a capacity for 600 students, who are distributed over 214 courses. Since 2004, almost 2000 students have graduated. Currently 60.5 FTE are employed, 41.5 of whom are faculty.

2.2 Project Stakeholders

Note: The material of this sub-section is reproduced by permission in full from a template provided by the project stakeholders.

The primary supervisor was Dr. Andrew Brooks. Daily meetings took place at which Dr. Brooks advised on the next steps to take. Dr. Brooks also helped trouble-shoot technical difficulties. Dr.

Brooks was responsible for identifying additional tasks. The secondary supervisor was Dr. Ir. Richard van den Doel. Both supervisors recognise the importance of replications. Dr. Brooks first involved himself in replication work in the 1990s. The profiles of these two supervisors, taken from the college website, are as follows:

> Andrew Brooks is an associate professor of computer science at University College Roosevelt, Middelburg. He has BSc (astrophysics) and MPhil (astronomy) degrees from the University of Edinburgh (UK), and a PhD degree in computer science from the University of Strathclyde (UK). His doctoral thesis showed how data mining techniques can be used to analyze data from human-computer interaction experiments, resulting in a better understanding of the nature of the results. His current research interests are in experimental computer science and astronomical data processing. He is a member of the ACM, IEEE, and BCS.
>
> He (Richard van den Doel) got his MSc degree in Applied Physics from Delft University of Technology in 1997. He got his PhD degree in 2002 from the same university. His PhD focused on the monitoring of dynamic processes in microarrays using quantitative microscopic techniques. He continued to work in the same research group as a PostDoc from 2002 to 2004. This project involved a collaboration with the Unilever Research Laboratory in Vlaardingen, the Netherlands. In 2004 he started working at UCR as lecturer in Mathematics and Physics and as a tutor. His research interests are in the field of quantum mechanics, computer vision and image processing, dynamic systems and Bayesian statistics.

3 Training

3.1 Online Courses

Prior to selecting an application domain and replication study, a strong background in statistical computing using the R software needed to be developed. Luckily, online courses teaching R are plentiful. I chose the 'Mastering Software Development in R Specialisation' Coursera track, offered by John Hopkins and taught by Robert Peng. While not all courses were immediately relevant, they provide a broad introduction to the R environment and highlight some of its strengths. Especially the sections 'R Programming Environment', 'R Programming', 'Advanced R Programming' and 'Building Data Visualization Tools' provided the knowledge of R necessary for this internship. With some experience in programming and R specifically, these can be completed within two days. This course is recommended as a broad overview of R and its capabilities, but with the amount of courses on the market, one should look for a more specific course first.

Before consulting more rigorous works on NLP, several online courses were audited without completion. These were Natural Language Processing' by the Higher School Economics and 'CS224n: Natural Language Processing with Deep Learning'. Neither course was standalone, with a series of machine and deep learning discussions before these courses. For someone without a background in statistical learning, the black-box presentation of models used was incredibly frustrating and required reading outside video lectures. Despite this, the broad overview and informal normative comments from instructors helped create an overview of commonly used techniques within the field. Should one wish to complete the course, expect at least 3 days of work. These courses were not aimed at students without extensive background in machine learning, and as such are not recommended for anything beyond picking up important topics and common methodologies. The books described below served me much better.

3.2 Online Tutorials

As mentioned before, the `quanteda` package for text processing and NLP methods required extensive training. The format used, while incredibly powerful and fast, was not always compatible with vanilla R packages. Besides a stand-alone and comprehensive reference site and introductory article [12], Kohei Watanabe and Stefan Müller also created a tutorial page [45] with examples. Following along, step by step, greatly helped in learning to use `quanteda` as effectively as any other text processing package. Furthermore, Ken Benoit and the authors frequently respond to queries regarding the use of their package, on relevant sites. The cited tutorial takes a few hours to complete fully, but serves as an excellent resource for help with errors and clunky code.

3.3 Articles and Book Chapters

The field of natural language processing (NLP) draws upon a vast domain of knowledge, bringing together theory from statistics, computing and linguistics while contributing significantly to each. Yet despite this great variation in available literature, one book stands out as a truly seminal didactic resource: 'Speech and Language Processing' by Jurafsky and Martin [34]. The breadth of topics discussed does not come at the cost of depth of information provided, allowing it to serve as a gateway into more domain specific techniques required by certain applications. While this book does require an introductory course in mathematical statistics and experience with statistical computing, the text remains clear throughout. While the last published version [34] might be outdated, the authors plan to release a third edition in late 2019, with a draft version publicly available as of 2018 [35]. For the sake of this internship, the chapter corresponding to regular expression (chp. 2), n-gram language models (chp. 3), vector representations of text (chp. 6), sentiment classification (chp. 4), the Naive Bayes model (chp. 4), part-of-speech tagging (chp. 8), dependency parsing (chp. 13) and hidden Markov models (appendix A) were crucial reads. All in all, this book comes highly recommended and might serve as an excellent basis for a 200 or 300 level course in computational text analysis.

While the vast majority of text processing tasks were already discussed by Jurafsky & Martin, this internship took a more computer oriented approach to analysis. Questions regarding computational efficiency and complexity were answered by Manning, Raghavan and Schütze [33] in *Introduction to Information Retrieval*. This specific sub-domain of NLP might not have been directly related to sentiment analysis within scientific documents, the chapters on document classification are extensive and repeated (in relation to earlier mentioned sources) topics still provided very relevant information. This text proved much harder to read, being far more equation and algorithm dense, than Speech and Language Processing, but took a computer science centric approach that proved necessary. Especially the chapters regarding Naive Bayes (chp. 13) and Support Vector Machine (chp. 15) document classification were useful. Overall, these chapters and the book as a whole come highly recommended.

In case of any questions regarding machine learning methods, especially the supervised ones used for this report, any student should try the excellent 'An Introduction to Statistical Learning' [27]. Many of the chapters deal with extremely relevant topics, although of special importance were those on resampling methods (chp. 5) and Support Vector Machines (chp. 9). For this reason, and the use as general reference manual for later studies, this book comes highly recommended.

3.4 In-house Training

One of the project supervisors, Dr. Brooks, provided a series of lectures regarding software quality assurance, critical appraisal of abstracts and the stages within a successful replication. The scope of this individual project saw this training occur a week before delving into replication and coding, and served as a good springboard for the work to come. Especially the software quality assurance methods have been applied extensively to functions created, although reported in minimum. It is the hope that with these metrics, replication of sentiment analysis for scientific citation texts will be easier to pick up.

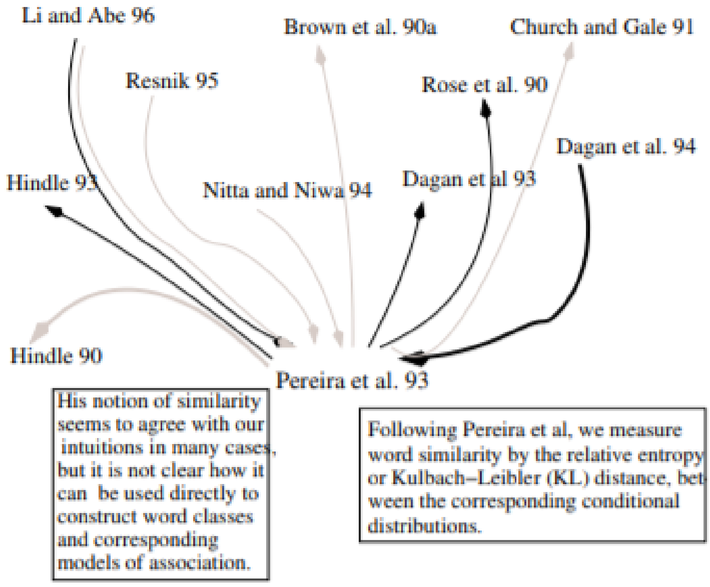
4 Articles and Choice(s)

4.1 Articles

The field of sentiment analysis, with scientific citations (SAoSC) as a subject, is a burgeoning one. The initiation of the field can be attributed to the work of Teufel et al.'s use of machine learning techniques to identify the function of citations [43]. These citations link authors, such as in figure 1, carrying both a purpose for establishing this link and a sentiment of the citing author to the cited. While their annotation scheme focused primarily on rhetorical function, these were immediately linked to sentiments [42]. Using various lists of cue phrases, POS patterns, verb tense and modality, the citation location and the presence of self-citations, a comprehensive feature set was created. Feeding 2829 citation sentences in a K-nearest neighbours variant, initial sentiment classification results were encouraging (macro-F1[2]: 0.71, Kappa: 0.58).

It would not be until 2011, however, until the next attempt at SAoSC was made with *Sentiment Analysis of Scientific Citations using Sentence Structure-Based Features* [5]. Athar opens the search for reliable sentiment classifiers by discussing the successes of state-of-the-art systems in corpora belonging to other genres. Despite this vote of confidence, it was already expected that scientific citations would provide more difficult. Not only are there innate differences between citation sentences and more traditional sentiment carrying texts, the politeness of academia towards to each others' works presents another hurdle. The main challenges highlighted include: obscured sentiment clues out of politeness; objective citations outnumber sentiment laden ones due to use outside of critical appraisal; the presence of jargon; varying scopes of citations, from a single clause to entire paragraphs (for example, the latter applies here).

[2] For explanation of evaluation metrics, see appendix B



His notion of similarity seems to agree with our intuitions in many cases, but it is not clear how it can be used directly to construct word classes and corresponding models of association.

Following Pereira et al, we measure word similarity by the relative entropy or Kulbach–Leibler (KL) distance, between the corresponding conditional distributions.

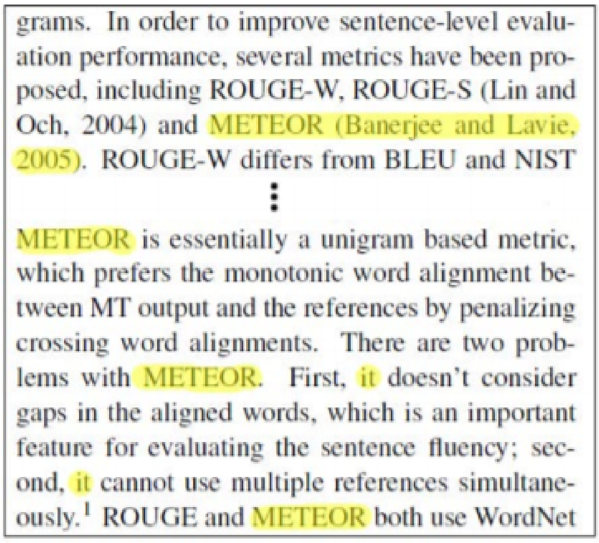
Figure 1: A citation network, showing interconnected publications and the relevant citation sentence in establishing the connection. Note the difference in tone between the two highlighted citation sentences. Retrieved from: [43].

To overcome these hurdles, three categories of features were considered. The first was limited to the words present within the citation sentence. Using an approach standard to computational text analysis, tokens were extracted as 1,2 and 3 grams; not just words, but words in specific sequences of length 1,2 and 3. The second category included several lexicon based approaches, wherein the citation context was tested for the presence of sentiment clues. The third category aimed to capture more complex relationships within text by incorporating structure within the sentence. This include adding dependency structures that capture the relationship of words among each other. For example, in the sentence below [5], the nominal subject[3] relationship between competitive and results is captured by the token *nsubj_results_competitive*. The subject of the sentence below are the results, obtained by an author (denoted by CIT to avoid bias), which are competitive. Another included sentence based feature were negation windows: words that come after prominent negation cues (see appendix A) gained a neg tag to their token.

> <CIT> showed that the results for French-English were competitive to state-of-the-art alignment systems.

While only the results using a SVM with a radial basis function were reported in the 2011 paper, a baseline Naive Bayes method was considered in the resulting technical report [6]. The most important results are summarised in table 2. All in all, the 1-3 grams with dependency features and negation windows proved most effective (macro-F1: 0.764) although this was not significantly better than using only the 1-3 grams with dependency features (macro-F1: 0.760). Further lexicon

[3] A noun phrase that is also the subject of the sentence or clause it resides in [19].



grams. In order to improve sentence-level evaluation performance, several metrics have been proposed, including ROUGE-W, ROUGE-S (Lin and Och, 2004) and METEOR (Banerjee and Lavie, 2005). ROUGE-W differs from BLEU and NIST

⋮

METEOR is essentially a unigram based metric, which prefers the monotonic word alignment between MT output and the references by penalizing crossing word alignments. There are two problems with METEOR. First, it doesn't consider gaps in the aligned words, which is an important feature for evaluating the sentence fluency; second, it cannot use multiple references simultaneously.[1] ROUGE and METEOR both use WordNet

Figure 2: A citation with extensive context. While the author's paper is only cited once, repeated references to it are made, greatly expanding the citation length. Retrieved from: [9].

based approaches only reduced predictive power.

Two papers followed by Athar and Teufel, prior to the release of Athar's technical report [6], attempting to include citation context as a feature for prediction. Often, sentiment of citations is found only after a mention of the citation [8]. An example of such a citation where context is important is figure 2. In their first, a re-annotated version of the data set from 2011 [5] was used, now taking into account the sentences preceding and following a citation clause. Experimentation followed, wherein the length of the citation context considered (again using 1-3 grams and dependency triplets as features) was increased. Surprisingly, there existed a negative relationship between context length considered and F1 scores [8]. The best achieved model when using variable citation sentence length was with an extended window of 0 sentences to the left and right (macro-F1:0.768); increased context only added noise to the training data. Their second paper attempted to remediate this by automatically detecting implicit citations. These different from explicit citations in the sense that no (Name, Year) token (or some variant) is present. Instead, papers and authors are receive implicit citations through repeated mentions. Athar and Teufel estimate that 60% of sentiments within their data set only come to light within this extended context, and that these sentiments are predominantly negative. Using a small subset of their data, a major improvement was shown when including automatically detected implicit citations (macro-F1:0.465 v. 0.687).

The above discussed works were finally synthesised in a technical report by 2014 [6]. One possible application of SAoSC is discussed, namely as a tool within bibliometrics. This opinion is also shared by Abu-Jbara, Ezra and Radev [2] and Jha et al. [28] in their work regarding sentiment analysis within scientific texts. Building off the work by Athar and Teufel, a methodology was constructed for first detecting citation context, followed by purpose and polarity classification.

While the task definition is very similar to earlier discussed works, their feature set differed. A number of lexica were applied, including ones similar to [5], but information regarding the citation was also considered (amount within sentence, grouped or separate, whether self citation, section of citation sentence), along with dependency relations. New to this approach was limiting the inclusion of certain tokens to those *closest* to the citation; these included the nearest verb, adverb, adjective and subjectivity cue. Another difference is the use of a linear kernel (as opposed to a radial one) and splitting the classifier into two binary SVMs; first subjectivity was classified (objective v. the rest) before polarity (positive v. negative). Results were comparable to those achieved by Athar, with a macro-F1 of 0.621 as baseline and 0.742 when considering human annotated citation contexts.

Several different authors have attempted to apply their methods to similar tasks. By 2017, Yousif et al. [54] identified 7 additional studies within the same sub-domain, and at least 4 additional data sets created specifically for this task. The highest reported macro-F1 is by Hernandez-Alvarez Gomez (macro-F1: 0.929) [24]. However, it should be noted that theirs used an ad-hoc data set that has seen no additional study applied to it, with far greater balance than those employed by Athar and Abu-Jbara et al. Recent publications show a shift towards non-supervised/neural classification schemes [30] [52] [53], but keep the original application of bibliometrics in mind [51].

4.2 Choice(s)

I chose to replicate the study by Athar, described first in 2011 and then with more detail in 2014, due to the relative importance it has had in the founding of a new field: sentiment analysis for bibliometrics. Today, works are being published, applying a variety of techniques to similar data sets. Furthermore, two informal Python replications had already been attempted, with lacklustre results [36][21]. These were most likely based on the publicly available code and data published by Athar themself [7].

5 Replication Work

5.1 Data sets

In this replication work, a large corpus of scientific citations were used corresponding to the 2011 Athar paper [5], and includes 8736 citation sentences that were manually annotated with polarity labels by the author and colleagues. The annotators achieved an inter-rater agreement, measured using Cohen's Kappa [17], of $K = 0.675$ or within acceptable limits. The corpus contained references to 194 papers from the ACL Anthology network.

Sentiment was classified using a 3 label scheme, with formal citations being either positive, negative or objective. For a positive label, the citation had to mention a specific positive attribute of the target paper, or an advantage/improvement of the target paper over another reference. Negative labels required mentioned attribute to be negative, or the comparison to prefer another referenced paper over the target reference. Objective labels were used when no explicit sentiment was detected. As is common with scientific texts, the majority of citations were found to be of objective sentiment (87.3%), while polar (non-neutral) citations were more commonly positive (9.6%) than negative (3.1%).

Package	Description	Version	Doc.
readtext	A collection of functions for importing text in various formats. Maintained by Ken Benoit and is directly compatible with Quanteda text formats. Used for importing text file	0.74	[29]
Quanteda	A fast and comprehensive text processing library. Maintained by Ken Benoit. Used for text processing	1.4.3	[11]
tidyverse	A self described ecosystem of R packages with a shared data processing philosophy, including *iterators, forcats, stringr, dplyr, purrr, readr, tidyr, tibble* and *ggplot2*. Used for general utilities and graphs	1.2.1	[46]
e1071	An R implementation of the C/C++ LibSVM package, along with additional machine learning tools. Used for SVM classification	1.7-2	[20]
foreach	Includes a loop function for parallel processing	1.4.4	[14]
doSNOW	Backend for parallel processing	1.0.16	[15]
Matrix	A hierarhy of different matrix classes, including sparse		[32]
cleanNLP	A set of tools for conversion of textual tools to tidy formats	2.3.0	[4]
udpipe	Natural language processing tools for POS tagging, lemmatization and dependency parsing	0.8.3	[47]

Table 1: The R packages used in replication, with details regarding their usage and included functions.

While the whole annotated data set is available, Athar has made available the test/train subset (7264 sentences) with dependency triplet tags appended on Github [7]. The remaining features were used as a development set and should not be used for model training and evaluation. The data was saved as a series of tuples consisting of 5 attributes:

1. Class: sentiment label determined by the author; factor: {o,n,p}
2. ID: a unique sentence identifier; integer: {736,7999}
3. Sentence: the citation sentence with the target reference replaced by a <CIT> token, and all other references replaced by <OTH>; string
4. Author: the name of the citing paper's author; string
5. Dependencies: dependency triplets generated by Athar, stored as *relation(governor,dependent)* triplet [6]; string

5.2 Process

A large part of this project involved replicating Java and Python code, software languages traditionally used for computation text analysis, in R. Athar's sentiment classifier made use of the WEKA package for Java [22], which includes a vast array of tools and methods for text classification specifically. While a similar R package does exist, this did not include all functionality necessary for replication. Instead, the following R packages were used throughout this project, labeled and described in Table 1.

The provided arff file, compatible with the WEKA software package, was converted into a CSV file using Microsoft Excel's `Convert Text to Table` function, such that each attribute inhabits a single column. The data file was then loaded into R using the `readtext` package, such that the resulting data frame contains 1 doc_id column (ID), 1 text column (Sentence) and 3 additional docvar columns (Class,Author,Dependencies). The `readtext` is another package created by Ken Benoit, and is directly compatible for use with his `quanteda` package.

The features described in 6.2.1 were generated for the Athar data set and attached to FS2. These were used as input for a radial basis, three class SVM (c=100) and a multinomial Naive Bayes model with a document-class proportion prior. To combat overfitting, all models were 10-fold cross validated, without stratification.

5.2.1 Features

1. **Token Extraction**: the texts were imported into R using the `readtext` package, and converted into a data frame with one text per row. For SVM models to be ran, a document feature matrix (DFM) needed to be constructed, with one token or word type per column, and each row containing the weighted count of that token within the target text. Initially white space tokens (separated by the spaces, tabs and new line characters) were extracted. Afters this, Quanteda's standard tokenizer was ran over the texts. Where required, the tokenized texts were clumped together as n-grams. The resulting DFM was TF-IDF weighted, defined in line with the book 'Information Retrieval' [33] below as,

   $$\mathrm{TF} \cdot \mathrm{IDF} = (1 + \log_{10}(\mathrm{count}\{t, d\})) \cdot \log_{10}(\frac{N}{\mathrm{count}\{d, t\}}) \tag{1}$$

   Otherwise, consider TF-IDF the product of the augmented log-transform of the terms given a document, with the log-transformed proportion of documents given a term. Intuitively this diminishes terms that occur frequently within a single sentence (TF), and further diminishes terms that occur frequently throughout the entire corpus (IDF).

2. **Dependency Features**: the dependency triplets had already been provided by Athar, incorporated as an additional column in the data set. These were extracted using a simple white space tokenizer and added to the DFM at a later stage. As with the n-gram features, all feature counts were TF-IDF transformed to suppress high frequency but noise inducing features.

3. **Negation Windows**: using the unprocessed texts, all words coming after the selected negation words (see appendix A) within a specified window were extracted from the text. The window was set to 15, following Athar's recommendation [6]. These word were then appended with a '_neg' tag and put back into their respective sentences. The example provided by Athar [6] shows a w=2 negation window with tag words appearing after a negation cue.

   > Turney's method did not work_neg well_neg although they reported 80% accuracy in CIT .

   The task of detecting negation with text, both its presence and scope, is non-trivial. The paper from which Athar derives their negation cues reported a macro-f1 score between 0.576 and

0.405 using state-of-the-art systems [37]. Therefore, for simplicity's sake, negation windows were used instead. The logic of such a system is displayed in the above example sentence, with presumably positive words having their meaning inverted since they come after a 'not'. Athar predicts that their location provides more information that negation cues alone would provide.

5.2.2 Support Vector Machine Implementations

Initial replications of the sentiment classifiers, with feature sets similar to those described by Athar [5][6], proved to be disappointing. This is largely due to a lack of reporting of methodology, despite the extensive related publications by the author. An expected reason for this was too great reliance on WEKA's black-box data processing function. This section is an exhaustive summary of the steps taken in reproducing the classification pipeline in R.

Classification results were far poorer than expected and further decrease with increasing complexity. The jump from Macro-F1 of 0.581 to 0.760 (see Table 2) by adding dependency triplets was simply not present, with the extremely large DFM only adding more noise into the model and reducing the number of correctly classified instances. The size of the data set additionally made processing and classification a painstakingly slow process, requiring 50 minutes for a single run of the 10-fold cross validation model evaluation.

Attempts to rectify this situation by adapting the processing of features (i.e. word tokenization, removal of English stop words, spelling corrections, removal of non-alphabetic characters) proved unsuccessful and diverged from the methodology described. It was therefore concluded that the discrepancy between results was due to an underlying issue with the R-based SVM models, and an investigation into these packages followed.

Both e1071 (R-package, see Table 1) and WEKA use a variant of the C/C++ LibSVM package [1]. Despite this, there are some features include in the `e1071` SVM command that are not present in other implementations. These include support for sparse matrices, formula descriptions using R objects and automatic scaling of training/test data. Using the rJava [44] and RWEKA [25] packages, the LibSVM module was ran on the iris data set through a WEKA interface. For the same parameters, there was virtually[4] no difference existed. This was in line with expectations, with both SVM implementations using parameters closely similar to the initial software package [16].

Chang, Lin and Hsu [26] have published guides with best practises for using LibSVM in different situations. While they consider the radial basis function as kernel the most general, it is expected that the linear kernel should achieve the same results in the case of large numbers of both data attributes and instances. The linear kernel is much less computationally intense, allowing for a factor of 10 reduction in cross-validation time. In the case of the Athar data set, switching to the `LiblineaR` package [23] a this reduction factor reached a factor of 30, however with the same disappointing results.

A big gain in efficiency was made through switching to parallel processing. By making use of the

[4] Due to random sampling during the cross-validation process, results were not identical every time

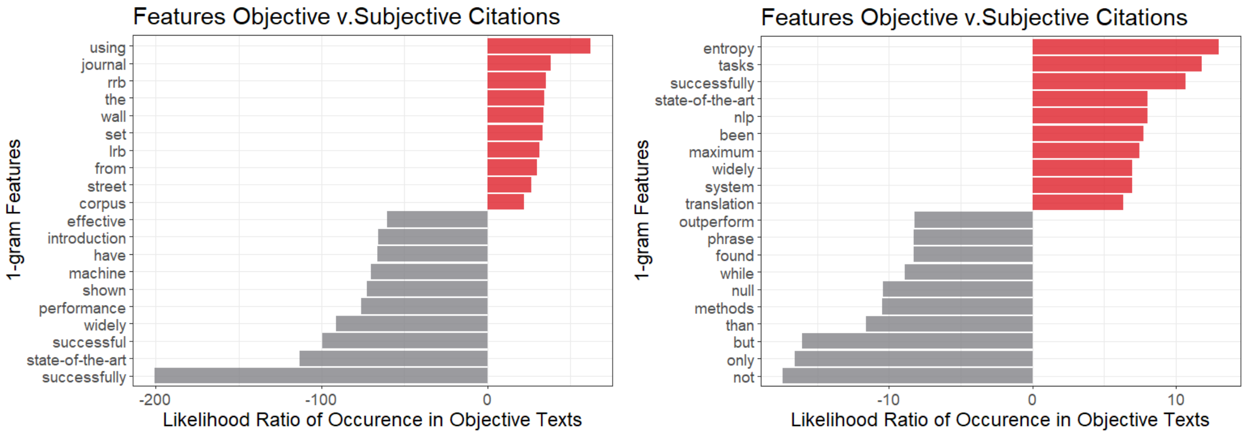


Figure 3: Likelihood ratio of words occurring per class. The target document class is highlighted in red. While certain discriminatory terms appear (e.g. not, but, than, outperform for the negative sentiment citations), many terms include merely represent noise.

8 cores present on the PC used for replication, through the `foreach` and `DoSNOW` packages, tasks repetitive in nature can be made significantly faster.

Various other best practises were attempted (i.e. forcing more iterations by lowering the stopping parameter drastically, weighting the c-parameter with the inverse class frequency to enforce 'stricter' classification for classes with more data, toggling the probabilistic vote counting for class assignment, applying scaling of train and test data outside of the SVM function) with marginal gain in predictive power. Using Athar's publicly available Java code [7] the same parameter values could be used (c=1000 for FS1), however it was suggested repeatedly [16][26] to tune the SVM model for adequate classification. Making use of parallel processing and sparse matrices, model training and testing times were drastically reduced for the larger data sets common in document classification, allowing for efficient parameter optimisation. Using a coarse to fine grid search, better model parameters were found. Typically the cost parameter fell in the $10^1 - 10^2$ range: far lower than reported. Regardless, the macro-F1 values failed to exceed the 0.600 level for all feature sets.

While replication frustratingly remained unsuccessful, a much better understanding of handling large data sets during supervised machine learning was gained. While even the 1-gram feature set (7,264 rows by 12,365 columns at 99.8% sparsity) took upwards of 50 minutes per cross-validation cycle at the beginning of the SVM investigation, eventually this was reduced to 60 seconds.

5.2.3 Feature Selection

Seeing as no difference was found between the R and WEKA implementations of LibSVM, an investigation into the pre-processing steps within WEKA followed. The data was loaded into WEKA as an arff file, processed using the `StringToWordVector` class. This function is one of many filters provided by WEKA and specifically belongs to the `filters.unsupervised.attribute` package. The `StringToWordVector` class takes 16 options, including those that specify TF-IDF transformation of term frequencies, tokenizing functions using regular expressions and stemming and stopword handling. Two options that caught the eye were regarding the number of word fields to create (-W: default 1000) and a toggle for pruning rates on a per class basis (-O: default TRUE). Within the class, a call is made to the `DictionaryBuilder` class and these options are taken as arguments. This class generates a unique list of words present within a data set, before reducing this to the desired number of word fields. An overview of the pruning methodology is provided by algorithm 1. It is suspected that `DictionaryBuilder` sorts the dictionary for each class and then uses the 1000-th (or another value for W) value as a minimum for all other features within that class.

Algorithm 1: WEKA's Pruning within the DictionaryBuilder class

```
input  : A full DFM
output: A pruned and consolidated DFM
for every class do
    dictionary ← uniquely occurring features within documents of class
    Sort dictionary by total feature frequency and then rank;
    if length(dictionary) < wordsToKeep then
        min_value ← min_frequency
    else
        min_value ← max(min_frequency, frequency at rank wordsToKeep)
    end
    for every feature do
        if feature frequency ≤ min_value then
            Add to consolidated list using HashMap
        end
    end
end
```

A similar function was created in R to emulate algorithm 1, also labelled `DictionaryBuilder`. Upon applying this pruning method to the Athar data set, on the n-gram and dependency triplets separately, 10-fold cross validation produced much better results, using a fraction of the time. By limiting the data set to roughly 1000 tokens per distinct feature, macro-F1 scores jumped up by 20%, providing results roughly similar to those reported.

While basing feature selection based on frequency of words per class is relatively primitive given the other options available, it does work. In figure 3 the likelihood ratio is used, and already shows some sentiment carrying tokens. Some informal experimentation with measures like point wise mutual information, chi square and TF-IDF weighted counts were conducted (for information, consult [33]), but these booked little to no gain over simple word frequency .

5.3 Comparison of Results

5.3.1 Model Comparisons

From the results it becomes clear that a very close, but partial replication has been achieved. Notably, even with all the same features implemented, and using the same feature selection method,

	NB				**SVM**			
	Athar		**Replication**		**Athar**		**Replication**	
Features	Macro	Micro	Macro	Micro	Macro	Micro	Macro	Micro
1-gram	0.482	0.776	0.617	0.842	0.581	0.863	0.601	0.888
1-3gram	0.474	0.764	0.579	0.819	0.597	0.862	0.596	0.886
FS1	0.469	0.755	**0.664**	**0.863**	0.760	0.897	0.721	0.909
FS2	**0.471**	**0.755**	0.664	0.862	**0.764**	**0.898**	**0.724**	**0.910**

Table 2: A comparison of results between Athar and the replication models. Note that for SVM, the macro-F1 scores remained a few percentage points below the benchmark while micro-F1 scores were slightly above (n-gram features excluded).

this R implementation of the SVM model remains slightly worse than those reported by Athar. Despite this, results clearly show the applicability of the methods to the data, resulting similar F1 scores. A spike in macro-F1 scores was expected when introducing the dependency features, evidenced by Athar's SVM implementation going from < 0.60 to 0.76. After feature selection this did indeed happen, indicating that the dependency features introduce a lot of noise to the data set. Therefore, the discrepancy in the results above might be entirely due to failing to select the right discriminating words prior to modelling.

Far more interestingly, the results for the Naive Bayes model are far higher than those reported. While it was expected that this relatively simplistic model would suffer when applied to the larger data sets, instead it performed admirably as a baseline comparison. A variety of reasons could be responsible for this improvement. Athar mentions the possibility for multicollinearity in features fed into the Naive Bayes model, violating its assumption of independence, evidenced by the poor results [6]. Perhaps due to re-implementation of feature selection methods, a large amount of similar features have been reduced to merely a few important ones.

Another reason might be the selected prior. Since this detail was not specified by the author, this replication made use of the document-class proportion prior. As such, the uninformed prior of word sentiment leaned heavily towards the 'objective' class (see figure 4). Therefore, the posterior would have resembled the sentiment class imbalance present within the data set.

Lastly, there is the slim possibility that Athar used a Gaussian Naive Bayes model as opposed to a multinomial or binary-multinomial recommended for text classification [33] [35]. There exist a variety of packages that make use of the Naive Bayes model in WEKA, and as such a mistake could have been made in selecting the right one. This issue is also common in R, with a packages like `e1071` and `quanteda` both having Naive Bayes implementations. The first assumes a Gaussian prior, whereas the second reproduces the model as described in the books cited above [10].

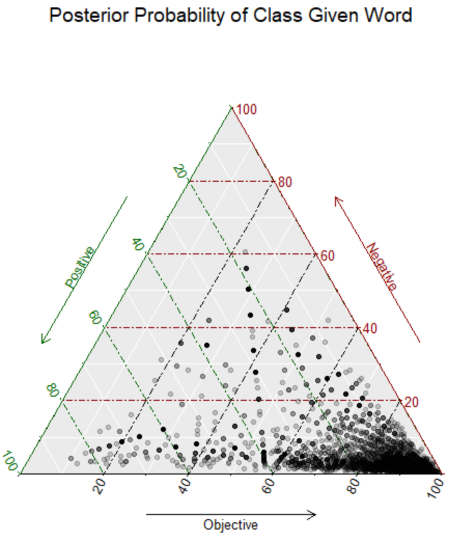


Figure 4: The sentiment of features as determined by the Naive Bayes model. Note that the majority of features lean towards objectivity, also this is expected given the use of the proportion of classes per document as a prior. Despite this, it can be seen that a few features lean towards the negative and positive corners, but are nowhere near as dominant as the objective features.

Athar	0	1	2	3	4	5
0.469	0.661	**0.684**	0.636	0.587	0.557	0.519

Table 3: Applying Laplacian smoothing of k=1 significantly increases the macro-F1 score of the Naive Bayes model.

6 Additional Tasks

6.1 Naive Bayes Extension

As mentioned already, the evaluation scores of the Naive Bayes model applied in the replication were far higher than those presented by Athar. Notably, while best-practises recommended in [26] were applied for the SVM classifiers, similar optimisation was not applied to the Naive Bayes model. One commonly recommended practise for sparse data is add-one or Laplacian smoothing [33]. This reduces the likelihood of 0 probabilities in the case of rare terms missing in the training data. As such, a number of constants were used for smoothing on the FS1 data set. An optimum was reached at k=1. With only 0.06 difference between the replicated SVM and Laplace smoothed NB models, it begs to question whether the additional classification accuracy is worth the greatly increased training time.

6.2 Feature Set Extension

6.2.1 Features

While many of the features employed by Jha et al. [2] [28] were already employed in some form by Athar, many were not. In this section an attempt is made to extend the analysis by expanding the feature set. Not all features could be replicated. These included the self citation presence, section and two lexica. Where minor alterations were made is described below, although the majority of features were replicated as closely as possible given the method descriptions in [2] and [28].

1. **Reference Processing**: while at this point of the investigation it is already blatantly apparent that not all citations are equal, many are not even syntactic or semantic constituents of the sentence. The following steps were performed using regular expressions.

   (a) **Reference Tagging**: all references are replaced by a single token. Target references (the subject of the sentiment) were replaced by *TREF* and all other references by *REF*.

   (b) **Reference Grouping**: all references occurring within proximity are grouped and replaced by a *GREF* token. Groups containing the target reference are replaced by *GTREF*. References within groups could be separated by punctuation or an and/or.

   (c) **Reference Filtering**: using an algorithm described in [3], all citations within the data set were reduced to those most likely syntactic, drastically reducing the number of errors in later processing steps. Indicators of syntatic importance include occurring at the start of the sentence or clause, or appearing after a preposition. The list of prepositions used came from [49], and included more complex 3-gram prepositional phrases.

2. **Reference count**: using a data set prior to grouping references (see above), the number of references that occurred within a citation sentence were counted.

3. **Separate Reference**: using the pre-processed dat set, checked whether the target reference occurred within a group or not.

4. **Closest Verb/Adjective/Adverb**: after pre-processing, all words were fed through the POS tagger provided by `CleanNLP` using the `udpipe` backend. All words were replaced by their lemma with their POS tag appended, i.e.

   these_DET problem_NOUN formulation_NOUN be_AUX similar_ADJ those_PRON study_VERB GTREFSYN_PROPN

   Universal POS tags were chosen as these had far less granularity, providing greater focus on syntactically relevant verbs/adjectives/adverbs. Unlike in [2], proximity to a citation was not selected based on shortest-path distance within the dependency tree, but rather using actual distance in word index. To reduce the number of tokens when merged with other features, the POS tags at the end of tokens were removed again, leaving only the lemmatized words.

5. **Lexicon Matching**: not all lexica used in the initial analysis were available. Notably the speculation cues and contrary expressions word lists came from outdated sources and were not accessible. The list of pronouns was taken from [48], with match presence being record (0 or 1). The negation cues were similar to those used by Athar, both coming from [37],

and matches were counted. Subjectivity cues were taken from [50], and were processed in a similar manner as the nearest verb/adjective/adverb.

6. **Dependency Parsing**: again using the `CleanNLP` framework, and using the `udpipe` backend, dependency tags were created. The lemmatized form of the governor and dependent were prepended with a relationship tag, and used as triples. Abu-Jbara et al. cite Athar [5], and as such, his methodology was copied.

6.2.2 Validation

	Jha et al.			**Replication**		
Metric	Objective	Positive	Negative	Objective	Positive	Negative
Precision	0.836	0.521	0.687	0.681	0.464	0.800
Recall	0.955	0.463	0.792	0.787	0.406	0.830
F1	0.891	0.513	0.687	0.730	0.430	0.812
	macro-F1: 0.686			macro-F1: 0.658		

Table 4: A comparison of results between Jha et al. and the replication for the SVM model, now 10-fold cross validation, SVM: Linear Kernel, c = 1.0.

To validate the accuracy of the extracted features, an SVM classifier was ran over the data set used used in [28], kindly provided by Rahul Jha. Unlike their classifier, the replication had separate data sets for each classification step, and the subjectivity results were not used for the polarity classifier. This means no propagation of errors, inflating the evaluation metrics for the positive and negative classes.

Despite this results were somewhat comparable. It is believed that the low comparative performance of the objective and positive sentiment classes occurred due to lacking features (the missing lexica and sectional location). One expects that the presence of self citations would be a strong predictor of positive sentiment, whereas the section location and lack of subjectivity cues would be strong predictors for the objective citation.

6.2.3 Extension

	NB		**SVM**	
Features	Macro	Micro	Macro	Micro
FS2	**0.684**	**0.862**	0.724	0.910
FS2 + Citation Count/Separate	0.681	0.881	**0.726**	**0.909**
FS2 + Nearest Verb/Adjective/Adverb	0.678	0.880	0.723	0.908
FS2 + Lexica	0.681	0.879	0.721	0.908
FS3*	0.681	0.882	0.723	0.909

Table 5: * Feature set 3 includes all Athar and the citation/nearest POS Jha et al. features.
A comparison of results between the replication with and without the extended feature set.

The features described in 6.2.1 were generated for the Athar data set and attached to FS2. These were used as input for a radial basis, three class SVM (c=100) and a multinomial Naive Bayes model with a document-class proportion prior. The results are displayed in Table 5. Besides a marginal increase for the citation count and separate features, no improvement was obtained.

		Predicted				
		Negative	Objective	Positive	Total	
Actual	Negative	143	68	33	244 (0.034)	F1: 0.72
	Objective	5	6138	134	6277 (0.86)	F1: 0.95
	Positive	6	434	303	743 (0.0.10)	F1: 0.50
	Total	154 (0.02)	6640 (0.91)	470 (0.07)	7260	

Table 6: The confusion matrix produced when using FS3. Notably, the distinction between O and P remains problematic. Here macro-F1 = 0.72 and micro-F1 = 0.92.

To discover where the misclassifications occur, a contigency table of a single 10-fold cross validation run of the FS3 data set trough the SVM model was produced. While the negative class achieves remarkably good recall (0.93) and sufficient precision (0.59), the positive class lags behind (recall: 0.64, precision: 0.41). Especially the distinction on the O-P axis seems difficult for the classifier to pick up on. This is reminiscent of the troubles in replicating the Jha et al feature set in Table 4.

6.3 Deliverables

Given the increased interest in SAoSC, it is understandable that some might want easily usable and reproducible code to build on for their analyses. For this reason, all data sets (both those of replicated studies and word lists), along with all R scripts will be made publicly available. Internally, these will be accessible in the UCR Moodle. Externally, it is the hope that these will be available on Github under the user IvoOVerhoeven and the repository Sentiment-Analysis-of-Scientific-Citations. This report will also be accessible there.

7 Discussion, Conclusions and Future Work

7.1 Discussion

Automatically identifying the sentiment of scientific citations is a non-trivial task. Common feature sets, model implementations and feature selection methods are necessary for generating a pipeline that hopefully generalises to other corpora. Despite increasing attention to this sub-domain, these pipelines remain incredibly sensitive to an author's interpretation and execution, as evidenced by the great difficulty experienced in replicating these studies.

In Athar's technical report [6], despite the greatly expanded descriptions compared to their standalone papers [5], reporting was insufficient to achieve the results presented. This replication was only partially successful due to making the data and code publicly available [7] well after the publication of their research. With the extensive documentation available for WEKA, similar functions as those used, could be emulated in R. Despite this, a significant discrepancy was found between the reported results for the naive Bayes model and those achieved when replicating. Somehow, after applying feature selection to the tokensets, macro-F1 scores were far higher than expected, suggesting that SVM did not manage to outperform the baseline as well as initially reported.

Similar problems with replication exist with the papers by Abu-Jbara, Ezra and Radev [2] and Jha et al. [28]. Despite this, a partial replication was achieved using a similar feature as described by these authors. Application of these feature sets along with Athar's did not result in better performance on the Athar data set. The models have great difficulty discriminating along the Objective-Positive domain, while being surprisingly good at identifying Negative sentiments given the low propensity of these within academic literature.

7.2 Conclusions

This report is the first successful replication of Athar's seminal work, despite two informal attempts [36] [21]. It is recommended to future researchers within the domain of citation sentiment analysis to greatly expand the methodology sections within their research publication. Furthermore, making used lexica, corpora and general data sets available would greatly improve the ease of replication and lower the barrier of entry. Considering the many potential applications, this goal is a necessary one, especially considering classification results have not improved dramatically in the decade and a half since the initial publication by Teufel [43].

7.3 Future Work

The corpora used in this replication studies have by no means been exhaustively analysed. The works of Athar and Teufel [8] [9], but also that by Abu-Jbara, Ezra and Radev [2] and Jha et al. [28], have proven the necessity for considering citation context. The considered corpora have the potential for such an analysis. Furthermore, future work seems inclined towards unsupervised machine learning architectures [30], which have not been attempted on the data sets employed in this report.

The detection of sentiment on its own is already a valuable tool for researchers, but linking this to significance of citations within a text [6], or the attitude of citing authors towards the cited author [28] are incredibly relevant applications of this technology.

8 Reflection

8.1 Technical Skills

Throughout this project, the most important developed technical skill is an understanding and general respect of research methodology. Using methods that were new to me (supervised machine learning), many new techniques had to be learned in an attempt to introduce rigour to the analyses and faithfully replicate the chosen study. Furthermore, having spent weeks attempting to emulate a methods description of another researcher, far more advanced than myself, the necessity for clear, yet comprehensive reporting of steps undertaken has been made abundantly evident.

Beyond these, many skills I expect to be prominent in future research had to be learned. These include writing clear and comprehensible code, handling medium to large data sets, applying statistical learning architectures, objectively evaluating results, function writing and quality assurance.

8.2 Recommendations

While UCR offers a great array of courses, the majority of these are purely theoretical in nature. Practical assignments are given, but besides the Senior Project (Bachelor thesis equivalent), little opportunity exists to delve into purely applied fields, such as NLP. That said, for this particular project, a good theoretical underpinning is crucial before starting. The Mathematics course 'Theory of Statistics and Data Analysis' was very helpful for understanding the models used in NLP, along with the coding experience developed in such a modelling heavy course was. As mentioned earlier, the Methods Statistics 'Advanced Topics in Methods Statistics' introduced the R language and the Rstudio environment. This course focused on performing Meta-Analyses, and is therefore a great way to learn critically appraise research papers and incorporate methodologies in replication studies. Lastly, any number of courses in the Computer Science track was absolutely necessary for the extensive programming required.

Doing an academic internship is highly recommended for students at UCR. With its broad curriculum, I feared applying to more applied Masters programs. With a purely applied focus of this project, and the necessity to improve upon the work of others, I can more confidently express not just interest but also experience within domains for future development.

References

[1] Libsvm: A library for support vector machines. https://www.csie.ntu.edu.tw/~cjlin/libsvm/.

[2] Abu-Jbara, A., Ezra, J., and Radev, D. Purpose and polarity of citation: Towards nlp-based bibliometrics. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: Human language technologies* (2013), pp. 596–606.

[3] Abu-Jbara, A., and Radev, D. Reference scope identification in citing sentences. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (2012), Association for Computational Linguistics, pp. 80–90.

[4] Arnold, T. A tidy data model for natural language processing using cleannlp. *The R Journal 9*, 2 (2017), 1–20.

[5] Athar, A. Sentiment analysis of citations using sentence structure-based features. In *Proceedings of the ACL 2011 student session* (2011), Association for Computational Linguistics, pp. 81–87.

[6] Athar, A. Sentiment analysis of scientific citations. Tech. rep., University of Cambridge, Computer Laboratory, 2014.

[7] Athar, A. Citation sentiment classifier. https://github.com/awaisathar/CitationSentimentClassifier, 2017.

[8] Athar, A., and Teufel, S. Context-enhanced citation sentiment detection. In *Proceedings of the 2012 conference of the North American chapter of the Association for Computational Linguistics: Human language technologies* (2012), Association for Computational Linguistics, pp. 597–601.

[9] Athar, A., and Teufel, S. Detection of implicit citations for sentiment detection. In *Proceedings of the Workshop on Detecting Structure in Scholarly Discourse* (2012), Association for Computational Linguistics, pp. 18–26.

[10] Benoit, K. Naive bayes in quanteda vs caret: wildly different results. https://stackoverflow.com/questions/54427001/naive-bayes-in-quanteda-vs-caret-wildly-different-results, Jan 12, 2019.

[11] Benoit, K., Watanabe, K., Wang, H., Nulty, P., Obeng, A., Müller, S., and Matsuo, A. *Package 'quanteda'*, 2019. R package version 1.4.3.

[12] Benoit, K., Watanabe, K., Wang, H., Nulty, P., Obeng, A., Müller, S., and Matsuo, A. quanteda: An r package for the quantitative analysis of textual data. *Journal of Open Source Software 3*, 30 (2018), 774.

[13] Brooks, A., Daly, J., Miller, J., Roper, M., and Wood, M. Replication of experimental results in software engineering. *International Software Engineering Research Network (ISERN) Technical Report ISERN-96-10, University of Strathclyde 2* (1996).

[14] Calaway, R., and Weston, S. *Package 'foreach'*, 2017. R package version 1.4.4.

[15] Calaway, R., and Wetson, S. *Package 'doSNOW'*, 2017. R package version 1.0.16.

[16] Chang, C.-C., and Lin, C.-J. Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST) 2*, 3 (2011), 27.

[17] Cohen, J. A coefficient of agreement for nominal scales. *Educational and psychological measurement 20*, 1 (1960), 37–46.

[18] Collaboration, O. S., et al. Estimating the reproducibility of psychological science. *Science 349*, 6251 (2015), aac4716.

[19] De Marneffe, M.-C., and Manning, C. D. Stanford typed dependencies manual. Tech. rep., Technical report, Stanford University, 2008.

[20] Dimitriadou, E., Hornik, K., Leisch, F., Meyer, D., Weingessel, A., Chang, C.-C., and Lin, C.-C. *Package 'e1071'*, 2019. R package version 1.7-2.

[21] Ecer, D. Citation sentiment analysis. https://github.com/elifesciences/citation-sentiment-analysis, 2018.

[22] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. The weka data mining software: an update. *ACM SIGKDD explorations newsletter 11*, 1 (2009), 10–18.

[23] Helleputte, T., Gramme, P., and Paul, J. *Package 'LiblineaR'*, 2017. R package version 2.10-8.

[24] Hernández-Alvarez, M., and Gómez, J. M. Citation impact categorization: for scientific literature. In *2015 IEEE 18th International Conference on Computational Science and Engineering* (2015), IEEE, pp. 307–313.

[25] Hornik, K., Buchta, C., Hothorn, T., Karatzoglou, A., Meyer, D., and Zeileis, A. *Package 'rWeka'*, 2019. R package version 0.4-40.

[26] Hsu, C.-W., Chang, C.-C., Lin, C.-J., et al. A practical guide to support vector classification.

[27] James, G., Witten, D., Hastie, T., and Tibshirani, R. *An introduction to statistical learning*, vol. 112. Springer, 2013.

[28] Jha, R., Jbara, A.-A., Qazvinian, V., and Radev, D. R. Nlp-driven citation analysis for scientometrics. *Natural Language Engineering 23*, 1 (2017), 93–130.

[29] Ken Benoit. *Package 'readtext'*, 2019. R package version 0.75.

[30] Lauscher, A., Glavaš, G., Ponzetto, S. P., and Eckert, K. Investigating convolutional networks and domain-specific embeddings for semantic classification of citations. In *Proceedings of the 6th International Workshop on Mining Scientific Publications* (2017), ACM, pp. 24–28.

[31] Lewis, D. D. Evaluating text categorization i. In *Speech and Natural Language: Proceedings of a Workshop Held at Pacific Grove, California, February 19-22, 1991* (1991).

[32] Maechler, M. *Package 'Matrix'*, 2019. R package version 1.2-16.

[33] Manning, C., Raghavan, P., and Schütze, H. Introduction to information retrieval. *Natural Language Engineering 16*, 1 (2010), 100–103.

[34] Martin, J. H., and Jurafsky, D. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition.* Pearson/Prentice Hall Upper Saddle River, 2009.

[35] Martin, J. H., and Jurafsky, D. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*, 3 ed. 2018.

[36] Mondal, A. K. Sentiment analysis of citations. https://github.com/arnab39/Sentiment-Analysis-of-Citations, 2018.

[37] Morante, R., and Blanco, E. * sem 2012 shared task: Resolving the scope and focus of negation. In ** SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)* (2012), pp. 265–274.

[38] Popper, K. *The logic of scientific discovery.* Routledge, 2005.

[39] Roosevelt, U. C. Ucr 10 years. http://www.ucr.nl/about-ucr/history-UCR/Documents/UCR%2010%20Years.pdf.

[40] Roosevelt, U. C. University college roosevelt website. http://www.ucr.nl/Pages/default.aspx.

[41] Shepperd, M., Ajienka, N., and Counsell, S. The role and value of replication in empirical software engineering results. *Information and Software Technology 99* (2018), 120–132.

[42] Teufel, S., Siddharthan, A., and Tidhar, D. An annotation scheme for citation function. In *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue* (2006), pp. 80–87.

[43] Teufel, S., Siddharthan, A., and Tidhar, D. Automatic classification of citation function. In *Proceedings of the 2006 conference on empirical methods in natural language processing* (2006), Association for Computational Linguistics, pp. 103–110.

[44] Urbanek, S. *Package 'rJava'*, 2019. R package version 0.9-11.

[45] Watanabe, K., and Müller, S. Quanteda tutorials, 2019.

[46] Wickham, H. *tidyverse: Easily Install and Load the 'Tidyverse'*, 2017. R package version 1.2.1.

[47] Wijffels, J., Straka, M., and Straková, J. *Package 'udpipe'*, 2019. R package version 0.8.3.

[48] Wikipedia contributors. English personal pronouns — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=English_personal_pronouns&oldid=900481877, 2019. [Online; accessed 6-July-2019].

[49] Wikipedia contributors. List of english prepositions — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=List_of_English_prepositions&oldid=897751729, 2019. [Online; accessed 6-July-2019].

[50] Wilson et al. Opinionfinder. http://mpqa.cs.pitt.edu/lexicons/.

[51] Yan, E., Chen, Z., and Li, K. Authors' status and the perceived quality of their work: Measuring citation sentiment change in nobel articles. *Journal of the Association for Information Science and Technology* (2019).

[52] Yousif, A., Niu, Z., and Nyamawe, A. S. Citation classification using multitask convolutional neural network model. In *International Conference on Knowledge Science, Engineering and Management* (2018), Springer, pp. 232–243.

[53] Yousif, A., Niu, Z., Nyamawe, A. S., and Hu, Y. Improving citation sentiment and purpose classification using hybrid deep neural network model. In *International Conference on Advanced Intelligent Systems and Informatics* (2018), Springer, pp. 327–336.

[54] Yousif, A., Niu, Z., Tarus, J. K., and Ahmad, A. A survey on sentiment analysis of scientific citations. *Artificial Intelligence Review* (2017), 1–34.

A Word Lists

- **Negation** [37]: no, not, *n't, never, neither, nor, none, nobody, nowhere, nothing, cannot, can not, without,no one, no way
- **Prepositions** [49]: about, above, across, after, against, along, alongside, amid, among, around, as, at, atop, ontop, before, behind, below, beneath, beside, between, beyond, but, by, circa, come, despite, down, during, except, for, from, in, inside, into, less, like, minus, near, of, off, on, onto, opposite, out, outside, over, pace, past, per, plus, post, pre-, pro-, qua, save, short, since, than, through, throughout, till, to, toward, towards, under, underneath, unlike, until, unto, up, upon, versus, via, with, within, without, worth, according to, across from, adjacent to, ahead of, along with, apart from, as for, as of, as per, as regards, aside from, back to, because of, close to, counter to, down on, due to, except for, far from, inside of, instead of, left of, near to, next to, opposite of, opposite to, other than, out from, out of, outside of, owing to, prior to, pursuant to, rather than, regardless of, right of, subsequent to, such as, thanks to, up to, based on, as far as, as opposed to, as soon as, as well as, at the behest of, by means of, by virtue of, for the sake of, for lack of, for want of, in accordance with, in addition to, in case of, in front of, in place of, in point of, in spite of, on account of, on behalf of, on top of, with regard to, with respect to, with a view to
- **Pronouns** [48]:
  - First: I,me,my,mine,myself,we,us,our,ours,ourselves
  - Third: he,him,his,himself,she,her,hers,herself,it,its,itself, they,them,their,theirs,themselves,themself

B Model Evaluation

		Predicted				
		Negative	Objective	Positive	Total	
Actual	Negative	12	15	1	28 (0.04)	F1: 0.60
	Objective	0	629	9	638 (0.88)	F1: 0.96
	Positive	0	33	27	60 (0.08)	F1: 0.56
	Total	12 (0.02)	677 (0.93)	37 (0.05)	726	

Table 7: An example of a confusion matrix. This particular table was one test/train fold of the FS1 feature using the SVM model. Here macro-F1 = 0.70 and micro-F1 = 0.92.

Ultimately, both the Naive Bayes and Support Vector Machines models are binary classifiers. As such, their performance is evaluated by comparing the number of correctly classified to incorrectly classified instances. One intuitive method for displaying classifications is using a confusion or contingency table, for example Table 7. Here the main diagonal elements are the correctly classified instances, with off-diagonal vertical elements being false-positive and off-diagonal horizontal elements being false negatives. A popular evaluation metric within text classification is the F1 score; the harmonic mean between the relative frequency of true positives to predicted positives (precision) and to actual positives (recall) [31]. Otherwise, this is equivalent to the harmonic mean of the diagonal elements divided by the row and column sums. This measure is expected to be more indicative of model quality, as opposed to using just precision or recall, given highly unbalanced class proportions. In 7, for example, the model was 100% correct for the 12 citations classified as negative (precision), however these only represented 43% of the negative citation sentences provided (recall). With the objective class being dominant, classifications are naturally more likely to fall into this category over the others.

There exist a number of different methods for aggregating the F1 scores across classes. Micro-averaging treats all predictions as a single class, whereas macro-averaging is simply the mean over all classes [31]. As such, micro-averaging weighs classes by class proportion, while macro-averaging weighs each class equally. Given that model predictions tend to be more accurate for larger classes, the macro-average evaluation metrics tend to be much stricter (see caption of 7). Both are reported in this report, although Athar [6] mentions giving greater priority to macro-F1 as it is stricter for imbalanced data sets.