8-15-2014 12:00 AM

Identification of Informativeness in Text using Natural Language Stylometry

Rushdi Shams
The University of Western Ontario

Supervisor
Dr. Robert E. Mercer
The University of Western Ontario

Graduate Program in Computer Science
A thesis submitted in partial fulfillment of the requirements for the degree in Doctor of Philosophy

IDENTIFICATION OF INFORMATIVENESS IN TEXT USING NATURAL LANGUAGE STYLOMETRY

(Thesis format: Integrated Article)

by

Rushdi Shams

Graduate Program in Computer Science

A thesis submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

The School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

Abstract

In this age of information overload, one experiences a rapidly growing over-abundance of written text. To assist with handling this bounty, this plethora of texts is now widely used to develop and optimize statistical natural language processing (NLP) systems. Surprisingly, the use of more fragments of text to train these statistical NLP systems may not necessarily lead to improved performance. We hypothesize that those fragments that help the most with training are those that contain the desired information. Therefore, determining informativeness in text has become a central issue in our view of NLP. Recent developments in this field have spawned a number of solutions to identify informativeness in text. Nevertheless, a shortfall of most of these solutions is their dependency on the genre and domain of the text. In addition, most of them are not efficient regardless of the natural language processing problem areas. Therefore, we attempt to provide a more general solution to this NLP problem.

This thesis takes a different approach to this problem by considering the underlying theme of a linguistic theory known as the Code Quantity Principle. This theory suggests that humans codify information in text so that readers can retrieve this information more efficiently. During the codification process, humans usually change elements of their writing ranging from characters to sentences. Examples of such elements are the use of simple words, complex words, function words, content words, syllables, and so on. This theory suggests that these elements have reasonable discriminating strength and can play a key role in distinguishing informativeness in natural language text. In another vein, Stylometry is a modern method to analyze literary style and deals largely with the aforementioned elements of writing. With this as background, we model text using a set of stylometric attributes to characterize variations in writing style present in it. We explore their effectiveness to determine informativeness in text. To the best of our knowledge, this is the first use of stylometric attributes to determine informativeness in statistical NLP. In doing so, we use texts of different genres, viz., scientific papers, technical reports, emails and newspaper articles, that are selected from assorted domains like agriculture, physics, and biomedical science. The variety of NLP systems that have benefitted from incorporating these stylometric attributes somewhere in their computational realm dealing with this set of multifarious texts suggests that these attributes can be regarded as an effective solution to identify informativeness in text. In addition to the variety of text genres and domains, the potential of stylometric attributes is also explored in some NLP application areas—including biomedical relation mining, automatic keyphrase indexing, spam classification, and text summarization—where performance improvement is both important and challenging. The success of the attributes in all these areas further highlights their usefulness.

Keywords: Stylometry, The Code Quantity Principle, Text Analytics, Natural Language Processing, Machine Learning, Computational Linguistics, Biomedical Relation Mining, Keyphrase Indexing, Spam Classification, Text Summarization.

Co-Authorship Statement

This thesis is written in an integrated article format. All the papers related to this thesis—published, submitted, or to be submitted—have two authors. The author of this dissertation is the primary author and Robert E. Mercer is a co-author of all the publications. Rushdi Shams, the author of this dissertation, carried out the literature survey, developed the approach described in the paper, conceived the design of the study, performed the statistical analysis, and drafted the manuscripts for the publications. In the capacity of a supervisor of this research, Robert E. Mercer has participated in the design of the study, interpretation of its results and preparation of the manuscripts for publication.

Dedication

To my mother. And to my mother.

Acknowledgments

Above all, I thank Almighty Allah for His blessings and mercy upon me. Surely I belong to Him and to Him shall I return. He gave me the strength, knowledge, enthusiasm, perseverance, and patience to finish this work in time and with this much capacity.

I would like to express my sincere appreciation and thanks to my supervisor Dr. Robert E. Mercer. He is among the best mentors I have found so far. He was always enthusiastic about my ideas, appreciating for our achievements, and encouraging during all the research blocks. Most importantly, he was always smiling! His methods have guided me towards the completeness as an independent researcher in text analytics.

My appreciation goes to my lovely wife Zereen for her patience for four years and for her faith in me. Her encouragements, inspirations and constructive criticisms of my work were among many of the driving forces for my thesis. I thank her for our discussions on code optimization ideas, program bugs, developing re-usable software tools, and so on.

Special thanks to my sister Nishat and my brother Mehedi and their families. During my hard time of my PhD, they were truly beside me as they always have been. My brother was always my inspiration to study Computer Science and do programming. I am also in debt to my in laws Maa and Baba who were always considerate, highly encouraging, and passionate about my PhD.

My sincere thanks to my lab colleagues and friends Brian, Fadi, Hospice, Dr. Maryam, Mengshuo, Shifta, and Syeed. Their invaluable suggestions on my work during the lab meetings were very helpful. Especially, Syeed helped me a lot by modifying some of his software tools that made it possible for me to annotate a few datasets. Besides, I thank all of my friends in London ON for whom I never felt alone. Special thanks to Janice for providing me all the documents and help whenever I needed them. Thanks to all the "third-floor" ladies in the main office. You were simply superb!

I am very grateful to many people who helped me in many regards during my research. I am extremely grateful to my friend Dr. Sadia Afroz for giving me countless information about her studies on Stylometry. I thank Dr. Charles Ling for his "how to" speeches on conducting a PhD research, giving a killer presentation, and his remarks on "cost-sensitive learning". My thanks go to Dr. Alyona Medelyan for her time-to-time correspondence during the keyphrase indexing experiments. I am grateful to Dr. Ian Witten for his correspondence in evaluating some of the keyphrase indexers. Thanks to Dr. Robert Holte for his remarks on "cost-sensitive evaluation" and letting me use his "Cost-curve Tool". His comments on some of the Cost Curves were invaluable. I am indebted to Dr. Arzucan Özgür for his correspondence on some of the datasets. I appreciate Lance DaSilva and Lizhen Guo for their help in one my projects. In the very end of the thesis, the help of Dr. Constantin Orăsan came as a life saver.

I also acknowledge the insightful thoughts and valuable feedbacks from my thesis examiners: Dr. Sylvia Osborn, Dr. Charles Ling, Dr. Victoria Rubin, and Dr. Fred Popowich.

Finally, I thank my parents. My mother lost my father during his PhD and I "almost" lost my mother to dementia during my PhD. May they have mercy even as they cherished me in childhood.

This thesis is dedicated to my mother.

Contents

List of Figures

List of Tables

List of Appendices

Chapter 1

Introduction

1.1 Thesis Statement

Determining informativeness or importance in text is a big challenge in natural language processing. In this thesis, we explore some possibilities to find informativeness in text using a set of natural language attributes or features related to Stylometry—the linguistic analysis of writing style. The hypothesis underlying our exploration is based on a linguistic principle that authors apply linguistic techniques, deliberately or not, to codify important information in texts. The principle is widely known as the *Code Quantity Principle* [1]. Interestingly, the elements of this codification are also the key elements present in the writing styles of humans. In this study, these shared elements are represented as attributes—called the *stylometric attributes*—based on Stylometry. Using these attributes, we provide a more general solution to a problem that so far has been approached mostly with domain-dependent solutions.

1.2 Problem Statement

On a good note, most texts now-a-days are in digital form and publicly available. Because of this plethora of text, for the last few years, we have experienced a big increase in statistical natural language processing. The effectiveness of the methods from this domain depends mostly on the quantity and the quality of data. Although both aspects are equally important, the *quantity* of data almost always receives most of the attention. *More data beats the algorithm and the model* was almost the tone set by some of the state-of-the-art research in natural language processing (see for instance, [2] and [3]). However, in his talk at the Strata 2012 conference, Netflix's Xavier Amatriain said something very different—"Data is important but data without a sound approach becomes noise" [4] . He argued that language models usually have *high variance* and therefore adding more data helped the studies mentioned in [2] [3]. Surprisingly for data with *high bias*, adding more data simply does not help. Therefore, the quantity of data is not the sole end means to succeed in natural language processing.

A trivial way to insure data quality is *data preprocessing* such as removing function words, non-ASCII symbols, `html` tags, interpreting mathematical formulae, etc. [5]. And surprisingly, on many occasions, data preprocessing indeed boosts performance of natural language processing methods [6]. A better way to generate quality text data is to classify *important* or

informative text units like words, sentences, and paragraphs among a set of text units. The usual definition of importance or informativeness of data is the amount of detail contained in a text unit [7] or its potential information-carrying capacity [8]. In addition to the performance boost, selecting an important subset of texts should also substantially reduce extraneous data and remove unwanted work from various text processing routines. This is absolutely important because of today's *information overload* when everyone is trying to get rid of unnecessary data [9].

Unfortunately, until now, there are very few attributes to gauge informativeness of text data that do not depend on the genre (i.e., scientific documents, newswire articles, and emails) or domain (i.e., agriculture, economics, politics physics, sports, disaster and even biomedical science) of the data. For instance, sentence position is a good attribute for summarization methods but not for spam classification; it is good for newspaper article summarization but not for summarizing medical questionnaires. Over the last decades, a number of surface-level text attributes have been proposed by many studies (see, for example, [10–15]). However, much is discussed about their limitations including their excessive domain dependency. Some semantic graph-based [16] and *n*-gram text attributes [17], however, are promising in terms of performance on different texts. Still, only a few attributes can be named that perform well despite difference in genres and domains of the data.

1.3 Motivation

Text importance and human psychology are tied together by some interesting research. Duffy and Kabance [18], for instance, explored the reasons behind the easy-readings of kindergarten prose. Their scientific study was centered around human psychology of signaling information and its effect on writing. What they have found is interesting: the sentences of kindergarten prose are less informative and by being so, they contain fewer complex words and syllables making them *simple* to read. Interestingly, a linguistics theory known as the *Code Quantity Principle* [1] also proposed much the same thing. The theory states that humans codify important information in texts in such a way that it can signal the reader, and the codification process is based on conscious or unconscious changes made in the use of some linguistic elements such as syllables, long and short words, etc. This theory has been recently supported with substantial evidence that suggests that authors have a tendency to make sentences that have more coded information longer than others (see, for instance, [19]).

Stylometry, on the other hand, is a branch of linguistics that studies writing styles in texts. It assumes that combining words into syntactic structures through writing is both habitual and variable, and during this procedure authors create individualized and unconscious patterns that can distinguish one from another [20]. The patterns often can be represented with features that are quantifiable, distinctive, salient, frequent, and relatively immune from the conscious control of authors [21]. Besides, these features, known as *stylometric features*, help to count stylistic traits of a particular author and to distinguish genuine style from mere chance variation in usage [22]. Following are some features related to stylometry: *word length, number of syllables in a sentence, sentence length, distribution of parts-of-speech, number of function words in a sentence, vocabulary richness, vocabulary distributions, word frequency*, etc. (more detail of these features can be found in [22]). Note that these features or attributes are closely related to

those that humans consciously or unconsciously modify during their codification of important information in texts as described in the code quantity principle [23]. The two views captured by stylometry and the code quantity principle are therefore closely related. However, to the best of our knowledge, the link has not been established yet because the former hypothesizes that patterns in writing are mostly individualized, while the latter finds that in certain situations the patterns can be generalized.

1.4 Objectives

The thesis has three major objectives. As the thesis progresses, the reader will understand that we try to explore new ideas and possibilities but always keep the following three objectives in mind.

To define a novel set of natural language attributes that can identify informativeness. We are interested in defining a more general, non-domain-specific but still effective set of natural language attributes. In doing so, combining both the Code Quantity Principle and Stylometry we define a set of stylometric attributes to model unit of text present in documents or articles. We consider the informativeness determination problem as a binary classification problem: one class of text units will be informative while the rest will be in the non-informative class. Note that throughout the research we use the *sentence* as the text unit. The only exception is our work on spam classification where the body of *email* is considered as the text unit (see Chapters to).

To evaluate the attributes for different genres of text. This evaluation should test the scope of applicability of the attributes for different genres of texts. We limit this scope in this thesis to the texts collected from domains like biomedical science, agriculture, nuclear physics, email texts, and newspaper articles.

To apply the attributes for different natural language processing tasks. There are some natural language processing application areas for which we can apply the attributes and observe their possibilities. Some of the areas are classic like text classification and summarization while some are becoming crucial and important such as biomedical relation mining and automatic keyphrase indexing.

We have carefully selected the genres of texts and application areas so that we can compare our work with previous state-of-the-art techniques. These comparisons are required to allow us to remark on the accomplishments of our research. In addition, to achieve the goals of the thesis, we explore various machine-learning techniques like supervised and semi-supervised learning.

Note that this thesis explores the possibility of stylometric attributes to indentify informativeness in texts. We, however, do not attempt to explore the reasons behind the capacity to determine informativeness that they display. Although the reasons would be very interesting to know, a thorough investigation is required to answer those questions. Such an investigation can be seen as a possible extension to this thesis.

1.5 Contributions

In many aspects, the thesis has some major contributions. First, we define a novel set of natural language attributes related to stylometry to identify informativeness in texts. Much empirical evidence confirms that the attributes have potential strength to distinguish informativeness in texts. We explore the merits of the attributes for a wide array of domains like agriculture, physics, biomedial science, email, and newspaper articles. This finding is significant since we have only a few genre-independent natural language attributes. The other key contribution of the research is the strong evidence of the attributes' applicability in some state-of-the-art natural language processing areas that include biomedical relation mining, keyphrase indexing, spam classification, and single and multi-document text summarization. Last but not least, during this research, we have enhanced the usefulness of some benchmark datasets, that is we have provided an extra set of labellings for sentences, viz. the BioNLP [24], BioDRB [25], FetchProt [26], and the SemEval-2010 task 5 dataset [27].

1.6 Thesis Organization

The chapters in this thesis are designed in a way that should help the reader to understand the transitions that we had in our thinking and our approach. At the beginning of the research, our focus was only to find informativeness in biomedical texts. To do so, we used a single attribute known as the Fog Index [28]. The Fog Index is a yardstick measure to assess text readability. At this stage we did not consider the Fog Index to be a Stylometric attribute although the parameters of the index are all related to Stylometry. Also, as a sanity check to see whether this index had some possibilities, we did not develop any machine-learning techniques at this stage. Rather, we were relying on simple heuristic rules to classify the sentences of texts as informative and non-informative. We named this simple, single-attribute and rule-based idea *Text Denoising* since its main objective is to remove non-informative, redundant text called the *noise* text. The work in the first few chapters, namely from Chapter 2 to 5, are built upon this idea. However, at some point, we understood the method's limitations which motivated us to make two important methodological changes that are summarized in Chapter 6. First, we developed a machine-learning approach and second we explored more attributes. However, although the bulk of the attributes are stylometric, the success of the approach depended somewhat on some domain-specific attributes. Therefore, the reader might note that we did not use the term *stylometry* or *stylometric attributes* before Chapter 8. The objective of our research changed ever so slightly from that point on—we began to emphasize the evaluation of the attributes for texts of different genres and domains rather than simply denoising text by removing redundant text fragments from biomedical research articles. We started to see the problem of determining informativeness as a binary classification problem and applied machine-learning models generated from the stylometric attributes for text classification (Chapters 7 to 9), biomedical relation mining (Chapter 10), and text summarization (Chapter 11). Note that our approaches before and after the transitions are very closely related—the minor difference being how the output is presented.

The thesis is organized as follows:

Chapter 1 provides the introduction to the thesis. The chapter outlines the problem, motivation for doing this research, thesis objectives, and contributions to the scientific community.

Chapter 2 outlines a method for text reduction called *Text Denoising* that can be used as a first step for the effective mining of biomedical concepts containing disease and chemical relations from research articles. The method uses a single attribute related to readability known as the Fog Index [28] and heuristics to classify sentences into informative and non-informative categories.

Chapter 3 extends the work in Chapter 2 by using four more attributes related to readability.

In **Chapter 4**, we move our focus from biomedical relation mining task to aiding automatic keyphrase indexing. The chapter describes the use of *Text Denoising* as an intermediate step to enhance the performance of a supervised keyphrase indexing tool called Maui. The chapter is brief because its content is published as a short paper. For more details, a useful source can be found elsewhere[1].

Chapter 5 is built upon the work described in Chapter 4. It describes the application of *Text Denoising* for enhancing the performance of two supervised keyphrase indexers named KEA [29] and KEA++ [30]. Based on the idea presented in the following chapter, the work of Chapters 4 and 5 is extended and can be found in Appendix A.

Chapter 6 can be seen as the transition chapter between our rule-based and machine-learning based approaches. Additionally, this chapter briefly summarizes the work presented in the previous chapters.

Chapter 7 describes an anti-spam filter developed using the stylometric attributes and machine-learning models to classify both personalized and non-personalized spam emails.

Chapter 8 extends and improves the work described in Chapter 7 for personalized spam emails.

Chapter 9 combines and extends the work presented in Chapter Chapter 7 and 8. The contents of this chapter include more in-depth analysis of the performance of the anti-spam filter.

Chapter 10 outlines the performance of the stylometric attributes for protein relation bearing sentence classification in biomedical research articles. This work can be seen as an extension of the work presented in Chapter 6 but with more sophisticated machine-learning methods and with more attributes.

Chapter 11 illustrates how the stylometric attributes perform for single and multi-document summarization.

Finally, **Chapter 12** briefly summarizes the thesis and explores its limitations and possible avenues for extension.

Bibliography

[1] T. Givón, *Syntax: A functional typological introduction*. Amsterdam: John Benjamins, 1990.

[1] http://arxiv.org/abs/1204.2231

[2] A. Halevy, P. Norvig, and F. Pereira, "The unreasonable effectiveness of data," *IEEE Intelligent Systems*, vol. 24, pp. 8–12, Mar. 2009.

[3] E. Brill, "Processing natural language without natural language processing," in *Proceedings of the 4th International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing'03, (Berlin, Heidelberg), pp. 360–369, Springer-Verlag, 2003.

[4] X. Amatriain, "Netflix recommendations: beyond the 5 stars." O'Reilly Strata Conference 2012, 2012.

[5] D. Pyle, *Data preparation for data mining*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999.

[6] S. M. Weiss, N. Indurkhya, and T. Zhang, *Text Mining. Predictive Methods for Analyzing Unstructured Information*. Springer, Berlin, 1 ed., 2004.

[7] R. L. Daft and R. H. Lengel, "Information Richness: A New Approach to Managerial Behaviour and Organizational Design," *Research in Organizational Behaviour*, vol. 6, pp. 191–233, 1984.

[8] S. B. Sitkin, K. M. Sutcliffe, and J. R. Barrios-Choplin, "A dual-capacity model of communication media choice in organizations," *Human Communication Research*, vol. 18, no. 4, pp. 563–598, 1992.

[9] M. Eppler and J. Mengis, "The Concept of Information Overload: A Review of Literature from Organization Science, Accounting, Marketing, MIS, and Related Disciplines," *Kommunikationsmanagement im Wandel*, pp. 271–305, 2008.

[10] L. Plaza, M. Stevenson, and A. Daz, "Resolving ambiguity in biomedical text to improve summarization.," *Inf. Process. Manage.*, vol. 48, no. 4, pp. 755–766, 2012.

[11] L. P. Morales, A. D. Esteban, and P. Gervás, "Concept-graph based biomedical automatic summarization using ontologies," in *Proceedings of the 3rd Textgraphs Workshop on Graph-Based Algorithms for Natural Language Processing*, TextGraphs-3, (Stroudsburg, PA, USA), pp. 53–56, Association for Computational Linguistics, 2008.

[12] Y. Ko, J. Park, and J. Seo, "Improving text categorization using the importance of sentences," *Inf. Process. Manage.*, vol. 40, pp. 65–79, Jan. 2004.

[13] C. Nobata, S. Sekine, and H. Isahara, "Evaluation of features for sentence extraction on different types of corpora," in *Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering - Volume 12*, MultiSumQA '03, (Stroudsburg, PA, USA), pp. 29–36, Association for Computational Linguistics, 2003.

[14] T. Hirao, H. Isozaki, and E. Maeda, "Extracting important sentences with support vector machines," in *In Proc. 19th COLING*, pp. 342–348, 2002.

[15] W. T. Chuang and J. Yang, "Extracting sentence segments for text summarization: a machine learning approach," in *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '00, (New York, NY, USA), pp. 152–159, ACM, 2000.

[16] J. Leskovec, N. Milic-frayling, and M. Grobelnik, "Impact of linguistic analysis on the semantic graph coverage and learning of document extracts," in *Proceedings of the Twentieth National Conference on Artificial Intelligence(AAAI2005)*, pp. 1069–1074, 2005.

[17] M. Berker and T. Gngr, "Using genetic algorithms with lexical chains for automatic text summarization.," in *Proceedings of the 4th International Conference on Agents and Artificial Intelligence (ICAART2012)*, pp. 595–600, SciTePress, 2012.

[18] T. M. Duffy and P. Kabance, "Testing a readable writing approach to text revision," *Journal of Educational Psychology*, vol. 74, pp. 733–48, 1982.

[19] S. Ji, "A textual perspective on givn's quantity principle," *Journal of Pragmatics*, vol. 39, no. 2, pp. 292 – 304, 2007. Focus-on Issue: Discourse, Information, and Pragmatics.

[20] C. Chaski, "Variables and method for authorship attribution," Oct. 11 2007. US Patent App. 11/398,728.

[21] D. I. Holmes, "The Evolution of Stylometry in Humanities Scholarship," *Literary and Linguistic Computing*, vol. 13, pp. 111–117, Sept. 1998.

[22] D. I. Holmes, "The Analysis of Literary Style–A Review," 1985.

[23] E. Lloret and M. Palomar, "Challenging issues of automatic summarization: Relevance detection and quality-based evaluation.," *Informatica*, vol. 34, no. 1, pp. 29–35, 2010.

[24] J.-D. Kim, S. Pyysalo, T. Ohta, R. Bossy, N. Nguyen, and J. Tsujii, "Overview of bionlp shared task 2011," in *Proceedings of the BioNLP Shared Task 2011 Workshop*, (Portland, Oregon, USA), pp. 1–6, Association for Computational Linguistics, June 2011.

[25] R. Prasad, S. McRoy, N. Frid, A. Joshi, and H. Yu, "The biomedical discourse relation bank," *BMC Bioinformatics*, vol. 12, no. 1, pp. 188+, 2011.

[26] K. Franzën and D. Oppenheimer, "The fetchprot corpus: Documentation and annotation guidelines," tech. rep., Swidish Institute Of Computer Science Report, Sweden, 2007.

[27] S. N. Kim, O. Medelyan, M.-Y. Kan, and T. Baldwin, "Semeval-2010 Task 5: Automatic keyphrase extraction from scientific articles," in *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval 2010)*, (Uppsala, Sweden), 2010.

[28] R. Gunning, "Fog index after twenty years," *Journal of Business Communication*, vol. 6, no. 3, pp. 3–13, 1969.

[29] I. Witten, G. Paynter, E. Frank, C. Gutwin, and C. Nevill-Manning, "KEA: Practical automatic keyphrase extraction," in *Proceedings of the 4th ACM Conference on Digital Libraries*, (Berkeley, CA, USA), pp. 254–255, 1999.

[30] O. Medelyan and I. Witten, "Domain-independent automatic keyphrase indexing with small training sets," *Journal of the American Society for Information Science and Technology (JASIST)*, vol. 59, no. 7, pp. 1026–1040, 2008.

Chapter 2

Extracting Connected Concepts from Biomedical Texts using Fog Index

This chapter is based on the paper titled "Extracting Connected Concepts from Biomedical Texts using Fog Index" co-authored with Robert E. Mercer that appeared in the 12th Conference of the Pacific Association for Computational Linguistics (PACLING 2011).

In this paper, we establish Fog Index (FI) as a text filter to locate the sentences in texts that contain connected biomedical concepts of interest. To do so, we have used 24 random papers each containing any of the four pairs of connected concepts. For each pair, we categorize sentences based on whether they contain both, any or none of the concepts. We then use FI to measure the difficulty of the sentences of each category and find that sentences containing both of the concepts have low readability. We rank sentences of a text according to their FI and select 30 percent of the most difficult sentences. We use an association matrix to track the most frequent pairs of concepts in them. This matrix reports that the first filter produces some pairs that hold almost no connections. To remove these unwanted pairs, we use the Equally Weighted Harmonic Mean of their Positive Predictive Value (PPV) and Sensitivity as a second filter. Experimental results demonstrate the effectiveness of our method.

2.1 Introduction

In recent years, extraction of connected biomedical concepts (i.e., disease, treatment, genes) from texts has drawn the attention of scientists interested in finding functional similarity (i.e., identification of genes involved in human diseases) [1]. Although benchmark research has reported successful methods to extract biomedical concepts [2, 3], they have rarely followed simple procedures. For example, Perez-Iratxeta *et al.* [4] could not relate diseases with gene functions from biomedical texts forthwith- they needed to apply a twofold intermediary process of connecting disease with chemical components and chemical components with gene functions. The key reason for not applying simple methods to extract connected concepts from biomedical texts is manifold. While some researchers concentrated on the number of co-occurrence of concepts in the abstract of a paper [4, 5], others preferred to comb through either the full text [6] or pre-specified segments (i.e., Introduction, Methods, Results, and Discussion) [7]. Moreover, the connections can be either very general (i.e., biochemical connections)

or very specific (i.e., regulatory connections). Therefore, the demand of developing simple methods to identify and extract biomedical concepts from a scientific literature that maintain connections, general or specific, with one another is not met till to date. This situation suggests to use simple yet improved computational method to identify and extract important, explicit, and implicit connections from biomedical texts.

As text is highly structured by syntax and semantics of natural language, it is believed that any relation extraction method should involve these two features. However, several reports asserted their complexity [8, 9]. Apart from this, Sherman [10] proposed that scientific literature is subject to statistical analysis and zeroed in on the importance of average sentence length. Gunning [11] practically demonstrated this important measure along with the number of complex words (i.e., words with three or more syllables) to assess the readability of text known as the Fog Index (hereinafter, FI). It is now considered a yardstick for readability assessment of books, scientific literature and newspapers, and even to detect online chatting bots [12]. Besides, an interesting ascertainment that texts become relatively difficult to read when contain ideas and relations [18] can be motivational for using FI to find hidden relations in papers.

In this paper, we report a simple novel statistical method to extract connected biomedical concepts from biomedical texts using FI. We statistically established FI as a text filter, experimenting on 24 random papers that describe four pairs of concepts: *Ischemia-Glutamate, Ataxia-Dehydrogenase, Hypogonadism-Gonadotropin,* and *Epilepsy-GABA*. Besides FI, our method also uses the equally weighted harmonic mean of the connections' Positive Predictive Value (hereinafter, PPV) and sensitivity as a second filter. While the prior concentrates on the important part of the text where the connections are stated, the latter assesses their representativeness. We selected the sentences of a paper that are difficult to read and ranked the most frequent connected concepts present in them. With careful observations, we noticed that the first filter produces some *noisy* pairs of concepts that hold almost no connection. To exclude them, we re-ranked every pair of concepts based on the equally weighted harmonic mean of their PPV and sensitivity, and filtered them.

In the remainder of this paper, we describe related work, illustrate the methodology, report and discuss experimental results, and draw conclusions.

2.2 Related Work

New research trends in the biomedical field include the discovery of hidden connections in texts to form new hypotheses that can be explored further by conventional experimentation [6]. A series of investigations by Swanson [13, 14] showed that these hidden connections can lead us to new discoveries. He reported that fish oil leads to change in blood viscosity and red blood cell rigidity that helps prevent Raynaud's syndrome [13]. Later, investigative reports started to discover suggestions for clinical therapies and basic physiological linkages from bibliographically isolated texts. However, the working principles of Swanson's empirical research include the computational burden of full-text syntactic analysis and involve large literature databases like MEDLINE. Our work, though it does not generate hypotheses, can be a good means of finding implicit connections in texts using fewer computations (as it filters out texts according to their readability and does not operate syntactically) without involving literature repositories.

A handful of research work in semantic relation classification or extraction from bioscience

texts depends on the proper identification of connections. Rosario and Hearst [15] concentrated on discovering connections between "treatment" and "disease". They reported 79.6 percent accuracy in blindly identifying concepts that fall into either of the categories and are somehow connected with one another. They used a MEDLINE-based neural network that addresses it to be intriguing yet complicated. A similar machine learning technique was applied by Frunza and Inkpen [8] to extract disease-treatment connections from texts. Their reported accuracy surpassed the results of Rosario and Hearst although their interest was limited to MEDLINE 2001 titles and abstracts. Their paper, like many other prominent work [16,17], has a significant use of PPV and sensitivity to evaluate the mining technique. Contrary, we used these measures to evaluate the representativeness of the connected concepts.

Perez-Iratxeta *et al.* [4] proposed a massive framework to prioritize disease associated genes. Instead of looking into literature, they combined several isolated pieces of biomedical repositories like Medical Subject Heading (MeSH), Gene Ontology (GO), RefSeq database, and MEDLINE. They used both databases and ontology that have lack in communication with one another and thus experienced tedious and complex scoring methods and formidable number of intermediate stages. In our work, we decided to stick with texts only to remain simple yet capable of producing improved results.

Robert Gunning [11] first introduced Fog Index (FI) to measure semantic difficulties using average sentence length and polysyllabic words in his 1952 book *The Technique of Clear Writing*. We were motivated to apply FI as our text filter when we came across the results of an experiment carried out by Duffy and Kabance [18]. They converted a passage with no more than two phrases into primer prose and applied FI to test its readability. They found the score well below the readability index (i.e., it was excessively easy to read). Their investigation on this phenomenon suggested that easy articles (in this case the primer prose) obscure the relationships and ideas as they emphasize each of them equally. In other words, difficult articles possess relationships and ideas and emphasize them in particular that yields low readability. We believe that if biomedical texts display a similar attribute, then FI can be an appropriate measure to filter texts that bear associations of scientific interest.

2.3 Methodology

The work of Perez-Iratxeta *et al.* [4] lists pairs of connected concepts like disease-chemical components, chemical component-genes, and disease-genes. Among them, we considered four disease-chemical component pairs, namely *Ischemia-Glutamate, Ataxia-Dehydrogenase, Hypogonadism-Gonadotropin.*, and *Epilepsy-GABA*. We collected 24 scientific papers (six for each pair of concepts) at random from several biomedical literature repositories. To work with the text only, we removed the title, affiliations, keywords, footnotes, figures, tables, acknowledgements, and references from the paper.

We considered each pair of concepts and a paper related to them. We classified its sentences into three sets: sentences containing both of the concepts, none of the concepts and any of the concepts. For example, the sentence *"Glutamate, which is potentially excito-toxic to brain neurons, is released excessively during ischemia"*, will be put into the set of sentences containing both of the concepts *Ischemia* and *Glutamate*, as *Ischemia* and *Glutamate* are both present. Then, we applied Gunning's formula for FI (Eq. 2.1) to score the sentences of every

set. According to this formula, the lower the score of a sentence, the easier it is to read.

$$\text{Fog Index} = 0.4 \times \left(\left(\frac{\text{Words}}{\text{Sentence}} \right) + 100 \times \left(\frac{\text{Complex Words}}{\text{Words}} \right) \right) \tag{2.1}$$

It can be noted that according to Gunning, words that are polysyllabic (i.e., contain three or more syllables) are called *Complex Words*. Also, as we applied FI on every sentence, the value of Sentences is always 1. We normalized this score by the paper's average number of syllables per word because readability score of long and short sentences varies due to the total number of syllables [11]. Eq. 2.2 provides the normalized FI (FI') of the sentences in every set.

$$\text{Normalized Fog Index, FI}' = \frac{\text{Fog Index, FI}}{\text{Average No. of Syllables per Word}} \tag{2.2}$$

The FI' calculated for the three sets of sentences for the 24 papers in groups related to the four pairs of connected concepts are shown in Table 2.1.

Table 2.1 shows that, for every pair of connected concepts, while the set of sentences containing any or both of the concepts displays either *Low* or *Medium* readability, the set of sentences containing none of them consistently has *High* readability (i.e., Low FI'). This observation leads us to decide that those sentences that are easier to read contain fewer connected concepts and therefore, we should look into low-readable sentences for hidden connections.

Now that we have FI as a functioning text filter, we need to define a means to determine the number of low-readable sentences to be considered for concept extraction. To do this, we ranked every sentence in a paper based on their FI score and sorted them in descending order (i.e., the most difficult sentences are at the top of the list). From this sorted list, in five chunks, we selected the top 50 percent, 40 percent, 30 percent, 20 percent, and 10 percent of the sentences. For every chunk, we tagged these sentences with Genia Biomedical POS tagger [19], identified the nouns in them and used an association matrix to record the frequency of their co-occurrences (i.e., number of occurrences of one noun with the other). For instance, the connected concepts in the sentence *"Glutamate, which is potentially excito-toxic to brain neurons, is released excessively during ischemia"* are *glutamate-brain, glutamate-neurons, glutamate-ischemia, brain-neurons, brain-ischemia,* and *neurons-ischemia*. From the output of the association matrix, we kept the 20 most frequent connected concepts for our experiment. As we observed, some chunk i contains new connections that are absent in chunk $i-1$ and vice versa. To find a threshold, we tracked the number of connections revealed and missed by every chunk i with respect to its previous chunk $i-1$. From Figure 2.1, we see that for the first chunk (50 percent of the sentences), all of the 20 most frequent connections are new. The number of new connections remains steady up to the third chunk (30 percent of the sentences) but then reaches the extremes in the fourth and fifth. The results in Figure 2.1 are shown for six papers related to *Ischemia* and *Glutamate*. Similar experiments with the other connected concepts showed that

Category	Ischemia-Glutamate	Readability	Ataxia-Dehydrogenase	Readability	Epilepsy-GABA	Readability	Dehydrogenase-Gonadotropin	Readability
FI'_{none}	5.99	High	5.77	High	6.50	High	5.58	High
FI'_{both}	8.26	Low	7.23	Medium	7.24	Medium	10.29	Low
FI'_{any}	6.83	Medium	7.33	Low	7.58	Low	7.62	Medium

Table 2.1: Normalized FI for three sets of sentences from 24 papers

Rank	Connected Concepts	Frequency	Semantic Connection	Rank	Connected Concepts	Frequency	Semantic Connection
1	Levels-Glutamate	48	Yes	8	10min-Ischemia	17	Yes
2	Ischemia-Glutamate	37	Yes	9	Glutamate-Neurons	16	Yes
3	Levels-Ischemia	33	No	9	Levels-5min	16	No
4	Levels-10min	22	No	9	Glutamate-Experiment	16	Yes
4	10min-Glutamate	22	No	10	Levels-Neurons	15	No
5	Levels-Half	21	No	10	Glutamate-CA4	15	Yes
5	Levels-Increase	21	Yes	10	Levels-CA4	15	No
6	Increase-Glutamate	20	Yes	11	Ischemia-5min	14	Yes
6	Glutamate-5min	20	No	11	Levels-Pretreatment	14	No
7	Half-Glutamate	19	No	11	Levels-Experiment	14	No

Table 2.2: Most frequent connected concepts for a paper on Ischemia and Glutamate

if we take less than 30 percent of the ranked sentences, the number of new concepts reaches the extremes.

We recorded a similar behavior for the number of connections dropped by every chunk. Figure 2.2 shows that as we start with it, the first chunk (10 percent of the sentences) does not miss any connection but the number of dropped connections suddenly starts to reach the extremes in the fourth and fifth. Again, the results in Figure 2.2 are produced by six papers on *Ischemia* and *Glutamate*. Similar experiments carried out with the other connected concepts showed that if we take less than 30 percent of the ranked sentences, the number of dropped connections reach the extremes.

These two observations indicate that the degree of concepts connected with each other is conserved if we take 30 percent of the low-readable sentences. Similar results are obtained for the three other pairs of concepts.

Provided this threshold, Table 2.2 shows the 20 most frequent connected concepts found in a paper on *Ischemia-Glutamate* where the connections are ranked according to their frequency. For each pair shown in Table 2.2, we extracted those sentences from the paper that contain both of the concepts. These sentences are fed to the Unified Medical Language System (UMLS) semantic relation network [20] to find out if the concepts have any semantic connection. Surprisingly, we found that among the 20 connected concepts, only nine have textual seman-

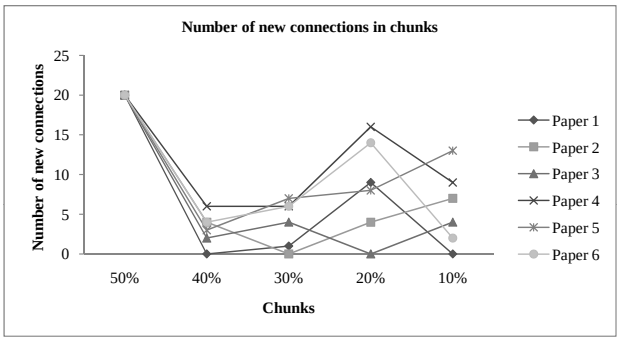


Figure 2.1: Number of new connections for six papers on Ischemia and Glutamate

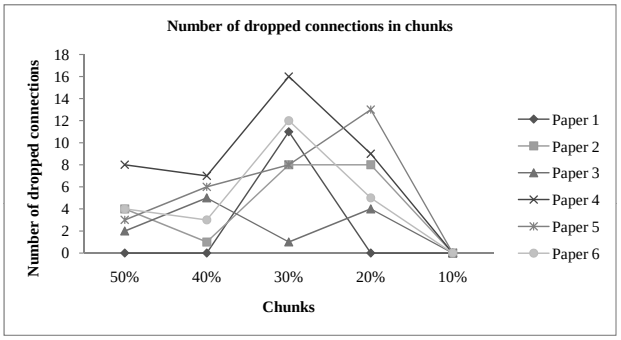


Figure 2.2: Number of dropped connections in five chunks for six papers on Ischemia and Glutamate

tic connections (Levels-Glutamate, Ischemia-Glutamate, Levels-Increase, Increase-Glutamate, 10min-Ischemia, Glutamate-Experiment, Glutamate-Neurons, Glutamate-CA4, and Ischemia-5min).

So, FI, as a text filter, brings in some text that contains most frequent connected concepts, some of which lack representativeness (i.e., they do not hold any connection). It urged us to provide a means to filter out these *noisy* pairs of concepts. As we collected texts at random, we observed that it is possible for the pairs to never co-occur in a sentence which indicates that our data set is imbalanced. So, we used the equally weighted harmonic mean of the PPV and sensitivity of the pairs of concepts provided by FI to evaluate their representativeness as it is a great evaluation metric for imbalanced dataset [8].

PPV[1] is the percentage of correctly predicted connections and sensitivity represents the percentage of connections identified as relevant by our method. To measure the PPV and sensitivity of every pair of concepts, we first considered the set of sentences filtered by FI and counted the number. This is the total number of results returned by our system (R) that comprises the number of True Positives (TP) and False Positives (FP). Then, we take a pair depicted in Table 2.2, searched the paper, and developed a second set of sentences that contain both of its concepts. The number of sentences in this set is the number of results that should have been returned by our system (S) and comprises the number of True Positives (TP) and False Negatives (FN). Finally, we counted the number of sentences that are present in both sets– which is the number of TPs by our system. Afterwards, FP is obtained by subtracting TP from R and FN is obtained by subtracting TP from S. So, the PPV of every pair of connected concepts is $\frac{TP}{TP+FP}$ and the sensitivity of every pair of connected concepts is $\frac{TP}{TP+FN}$. We then applied the formula in Eq. 2.3 to determine the equally weighted harmonic mean for the given pair of concepts. In this way, we measured this mean for every pair of concepts in Table 2.2.

$$\text{Harmonic Mean of PPV and Sensitivity} = \frac{2 \times \text{PPV} \times \text{Sensitivity}}{\text{PPV} + \text{Sensitivity}}. \tag{2.3}$$

We re-ranked the pairs of concepts in Table 2.2 according to their individual Harmonic Mean of PPV and Sensitivity, and considered the first 10 pairs of concepts. These 10 pairs of connected concepts are said to be the representative connected concepts of the paper. Similar

[1]Similar to F-Score, Precision, and Recall but their use in Information Retrieval and Classification is different. The terms PPV and Sensitivity have been used to avoid confusion with the evaluation terminology.

procedure is followed to evaluate the representativeness of the pairs of concepts for the rest of the connected concepts: *Ataxia-Dehydrogenase, Hypogonadism-Gonadotropin,* and *Epilepsy-GABA*.

We also measured the accuracy of every connected pair by using Eq. 2.4–

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}}. \tag{2.4}$$

where *TN* is the number of True Negatives and can be found by subtracting *TP*+*FP*+*FN* from the total number of sentences in a text, and ranked them accordingly. However, we found that in the case of accuracy, the ranked connections are not well distinguished.

2.4 Results and Discussions

In this section, the re-ranked connected concepts according to their individual Harmonic Mean of PPV and Sensitivity are reported. The results show that most of the biomedical connected concepts extracted by the proposed method are reported to be semantically connected by UMLS. This indicates that the use of the Harmonic Mean substantially decreased the number of *noisy* relations extracted by the FI.

Table 2.3 lists the 10 connected concepts for a paper on Ischemia and Glutamate among which seven pairs of concepts are reported as semantically connected by UMLS. It can be seen that the pairs of concepts in Table 2.2 that hold merely no relation between them are decreased significantly. However, the three pairs of concepts, not having any semantic connection, could not be filtered because of their high frequency of co-occurrence in the text.

Table 2.4 shows the 10 connected concepts for a paper on Ataxia and Dehydrogenase, seven of which are semantically connected in the UMLS semantic relation network. Our observation of this domain reveals that PDHC (Pyruvate Dehydrogenase Complex) is manifested in Ataxia patients, especially those suffering from Friedreich's Ataxia. So, the relations among Friedreich, Ataxia and PDHC are vividly represented in the list. *Pyruvate-Ataxia* is extracted as connected concepts because in the text, the elaboration of PDHC co-occurred with Ataxia many times.

Table 2.5 lists the 10 connected concepts for a paper on Hypogonadism and Gonadotropin. According to the UMLS semantic relation network, eight of these pairs are semantically connected. Steroids have significant effects on diseases like Hypogonadism, where release of testosterone plays an important role. Therefore, the connection between AAS (a shorthand for Anabolic Androgenic Steroid) that induces Hypogonadism and Testosterone is present in the list. The pairs of concepts not semantically connected are still reported by our method for their extremely high co-occurrences in the text.

Table 2.6 displays the connected concepts present in a paper on Epilepsy and GABA. Epilepsy is a neuronal disease that causes inhibition, significantly affects neuronal structure like the Hippocampus, and is caused by low levels of GABA. In the list, we find seven concepts that are semantically related according to UMLS. It should also be noted that the Harmonic Mean of the pairs of concepts are significantly lower than that found from other papers, meaning either the paper is small or the concepts co-occurred infrequently.

Rank	Connected Concepts	Harmonic Mean	Semantic Connection
1	Ischemia-Glutamate	51.85	Yes
2	Levels-Ischemia	43.47	No
3	Levels-Glutamate	41.66	Yes
4	Glutamate-Neurons	39.02	Yes
5	10min-Ischemia	37.50	Yes
6	Glutamate-CA4	35.89	Yes
7	Increase-Glutamate	32.55	Yes
8	10min-Glutamate	31.81	No
9	Ischemia-5min	31.57	Yes
9	Glutamate-5min	31.57	No

Table 2.3: Connected concepts for a paper on Ischemia and Glutamate

Rank	Connected Concepts	Harmonic Mean	Semantic Connection
1	Friedreich-Ataxia	59.25	Yes
2	PDHC-Ataxia	56.00	Yes
3	Activity-Friedreich	43.47	Yes
3	Patients-Ataxia	43.47	Yes
3	Activity-Ataxia	43.47	Yes
3	PDHC-Friedreich	43.47	Yes
4	Preparations-Ataxia	40.00	No
4	Preparations-Friedreich	40.00	No
5	Pyruvate-Ataxia	38.09	No
6	Patients-Friedreich	36.36	Yes

Table 2.4: Connected concepts for a paper on Ataxia and Dehydrogenase

2.5 Conclusions

In this paper, we report on the extraction of connected concepts from biomedical texts by assessing text readability. The readability of text is determined by a metric called Fog Index (FI). We curated 24 random papers by using four pairs of connected concepts as keywords and applied FI on them. Experimental results showed that sentences display low readability if they contain connected concepts. We selected 30 percent of the most difficult-to-read sentences, and used an association matrix to track the most frequent pairs of concepts in them. To remove those pairs of concepts that have a rather weak connection, we used the equally weighted harmonic mean of their positive predictive value and sensitivity as a second ranking filter. The results are supported by finding almost all of the extracted concepts semantically connected by the UMLS semantic relation network.

Rank	Connected Concepts	Harmonic Mean	Semantic Connection
1	AAS-Treatment	29.41	Yes
2	Use-AAS	21.62	No
3	AAS-Testosterone	18.46	Yes
4	Gonadotropin-Treatment	18.18	Yes
5	Testosterone-Treatment	14.92	Yes
6	Levels-Testosterone	14.49	Yes
7	AAS-Conditions	12.90	Yes
7	Treatment-HCG	12.90	Yes
7	Replacement-Therapy	12.90	No
7	Treatment-Therapy	12.90	Yes

Table 2.5: Connected concepts for a paper on Hypogonadism and Gonadotropin

Rank	Connected Concepts	Harmonic Mean	Semantic Connection
1	Inhibition-GABA	26.08	Yes
2	GABA-Synapse	20.25	Yes
3	Neurons-Synapse	14.70	Yes
4	Inhibition-Hippocampus	12.30	Yes
5	Synapse-Change	9.37	No
6	Neurons-GABA	8.00	Yes
7	Properties-GABA	6.45	Yes
7	GABA-Change	6.45	No
8	GABA-Number	6.34	No
9	Cl-Gradient	3.33	Yes

Table 2.6: Connected concepts for a paper on Epilepsy and GABA

Bibliography

[1] F. Yuan, R. Wang, M. Guan, and G. He, "A Novel Computational Method for Predicting Disease Genes Based on Functional Similarity", *Lecture Notes in Computer Science*, Vol. 6216/2010, 2010, pp. 42-51.

[2] S. Zhao, "Named Entity Recognition in Biomedical Texts using an HMM Model", *International Joint Workshop on Natural Language Processing in Biomedicine and Its Applications*, Geneva, Switzerland, 2004, pp. 84-87.

[3] J. Kazama, T. Makino, Y. Ohta, and J. Tsujii, "Tuning Support Vector Machines for Biomedical Named Entity Recognition", *ACL-02 Workshop on Natural Language Processing in the Biomedical Domain*, Vol. 3, Philadelphia, USA, 2002, pp. 1-8.

[4] C. Perez-Iratxeta, P. Bork, and M. Andrade, "Literature and Genome Data Mining for Prioritizing Disease-Associated Genes", *Discovering Bimolecular Mechanisms with Computational Biology (Molecular Biology Intelligence Unit II)*, 2006, pp. 74-81.

[5] H. Dai, Y. Chang, R. T. Tsai, and W. Hsu, "New Challenges for Biological Text-Mining in the Next Decade", *Journal of Computer Science and Technology*, Vol. 25, No. 1, 2010, pp. 169-179.

[6] R. Lindsay and M. Gordon, " Literature-Based Discovery by Lexical Statistics", *Journal of the American Society for Information Science*, Vol. 50, No. 7, 1999, pp. 574-587.

[7] S. Agarwal and H. Yu, "Automatically Classifying Sentences in Full-text Biomedical Articles into Introduction, Methods, Results and Discussion", *Bioinformatics*, Vol. 25, No. 23, 2009, pp. 3174-3180.

[8] O. Frunza and D. Inkpen, "Extraction of Disease-Treatment Semantic Relations from Biomedical Sentences", *2010 Workshop on Biomedical Natural Language Processing, ACL 2010*, Uppsala, Sweden, 2010, pp. 91-98.

[9] G. Leroy, H. Chen, and J. D. Martinez, "A Shallow Parser based on Closed-class Words to Capture Relations in Biomedical Text", *Journal of Biomedical Informatics*, Vol. 36, 2003, pp. 145-158.

[10] A. L. Sherman, "Analytics of literature: A Manual for the Objective Study of English Prose and Poetry", Boston: Ginn & Co, 1893.

[11] R. Gunning, "Fog Index after Twenty Years", *Journal of Business Communication*, Vol. 6, No. 3, 1969, pp. 3-13.

[12] O. S. Goh, C. C. Fung, A. Depickere, and K.W. Wong, "Using Gunning-Fog Index to Assess Instant Message Readability from ECAs", *3rd International Conference on Natural Computation (ICNC 2007)*, Hainan, China, 2007, pp. 480-486.

[13] D. R. Swanson, "Fish Oil, Raynaud's Syndrome, and Undiscovered Public Knowledge", *Perspectives in Biology and Medicine*, Vol. 30, 1986, pp. 7-18.

[14] D. R. Swanson, "A Second Example of Mutually Isolated Medical Literatures related by Implicit, Unnoticed Connections", *Journal of the American Society for Information Science*, Vol. 40, 1989, pp. 432-435.

[15] B. Rosario and M. A. Hearst, "Classifying Semantic Relations in Bioscience Texts", *42nd Annual Meeting of Association for Computational Linguistics*, Barcelona, Spain, 2004.

[16] L. Smith *et al.*, "Overview of BioCreative II Gene Mention Recognition", *Genome Biology*, Vol. 9 (Suppl.2): S2, 2008.

[17] M. Krallinger *et al.*, "Overview of the Protein-Protein Interaction Annotation Extraction Task of BioCreative II", *Genome Biology*, Vol. 9 (Suppl.2): S4, 2008.

[18] T. M. Duffy and P. Kabance, "Testing a Readable Writing Approach to Text Revision", *Journal of Educational Psychology*, Vol. 74, 1982, pp. 733-48.

[19] Y. Tsuruoka *et al.*, "Developing a Robust Part-of-Speech Tagger for Biomedical Text", *Lecture Notes on Computer Science*, Vol. 3746/2005, 2005, pp. 382-392.

[20] Unified Medical Language System (UMLS), “UMLS Terminology Services (UTS)”, URL: https://uts.nlm.nih.gov/home.html [March 10, 2011]

Chapter 3

Evaluating Core Measures of Text Denoising for Biomedical Relation Mining

This chapter is based on the paper titled "Evaluating Core Measures of Text Denoising for Biomedical Relation Mining" co-authored with Robert E. Mercer that appeared in the 3rd International Workshop on Global Collaboration of Information Schools (WIS 2012).

Text Denoising is a tool that reduces texts to their content-rich parts. It has been reported as an effective tool which improves biomedical relation mining as well as supervised keyphrase indexers for digital libraries. The idea behind text denoising is that the complexity of a sentence plays an important role for it being the content-rich part of the text. Therefore, the core measure of text denoising is a well-known readability formula called Fog Index (FI). However, the effect of using other readability formulas is yet to be explored. In this paper we plug in four other readability formulas—FRES, SMOG, FORCAST, and FKRI—with text denoising and report their performance on mining relations from a corpus of 24 biomedical texts. Experimental results show that FI outperforms all other formulas in terms of meaningful relation extraction. The results also show that besides FI, formulas like SMOG index and FKRI can be used as core measures of text denoising for biomedical relation mining.

3.1 Introduction

Readability formulas measure text difficulty in quantitative terms and thus are served as effective warning systems against complex writing style since the 1930's. Nowadays, their use is widespread—from the assessment of readability of newspaper articles and scientific literature to the identification of chatting bots [1]. Recently, Shams and Mercer [2] proposed textual noise reduction, similar to that in image processing, using FI—a technique to keep content-rich sentences of a scientific paper based on their reading difficulty. The idea was based on a hypothesis that texts yield complexity when they contain concepts and relations [3]. The tool with its core measure FI, as applied on a corpus of 24 biomedical texts, extracted meaningful biomedical relations. Later, it was applied on three keyphrase indexers KEA [4], KEA++ [5] and Maui [6] to reduce their training data [7] [8]. Experimental findings showed that even with reduced training data, the indexers induced better classifiers and achieved better F-score than their benchmarks. The authors concluded that low-readable sentences of a text are content-rich

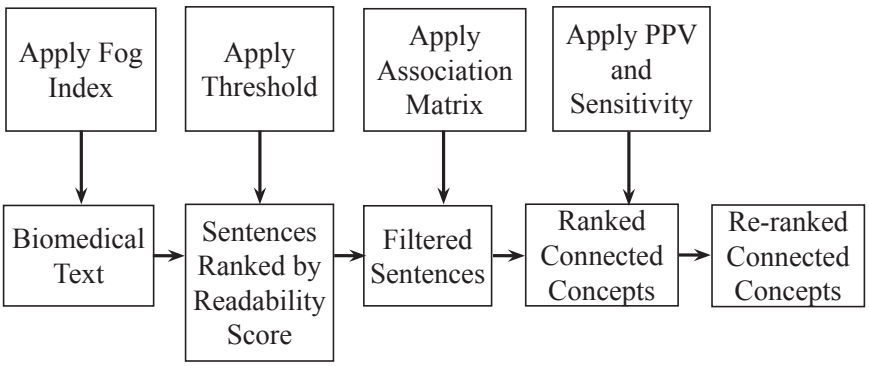


Figure 3.1: Text denoising and related concept extraction method described by Shams and Mercer [2]

when extracting relations or keyphrases.

To assess readability, FI considers two core measures, namely sentence length and complex words. Currently, over 50 formulas have been proposed [9] as readability measures. They consider different features involved in readability like paragraph length, white spaces, use of headings, monosyllabic words, choice of sample size, and proper nouns [10]. Among these formulas, four are considered not only as yardsticks but also close to the popularity of FI—Flesch Reading Ease Score (FRES), SMOG Index, FORCAST Index, and Flesch-Kincaid Readability Index (FKRI). Like FI, the two formulas provided by Flesch use the sentence length but consider word length as their second core measure. However, the Flesch formulas use different weighting factors leading them to correlate almost inversely with FI. On the other hand, SMOG index is similar to FI except that it operates on some specific samples of the text. FORCAST, unlike most other formulas, uses only one vocabulary element—monosyllabic words—making it useful for texts without complete sentences. Despite the difference in their working principles, the formulas can estimate the difficulty of style; their intention is not to rate the content, organization, format, imagery or quality of readers [11].

In this paper, we evaluate the performance of four readability formulas—FRES, SMOG, FORCAST, and FKRI—on textual noise reduction and the mining of biomedical relations. We apply the two-step method proposed by Shams and Mercer [2]: (i) extract the most frequent related concepts in the 30% of the low-readable sentences of biomedical texts; (ii) and select the related concepts with higher equally weighted harmonic mean of their Positive Predictive Value (hereinafter, PPV) and sensitivity. To achieve this, we use the dataset of 24 full biomedical texts [2] that describe four pairs of concepts: *Ischemia-Glutamate*,*Ataxia-Dehydrogenase*, *Hypogonadism-Gonadotropin*, and *Epilepsy-GABA* [12]. Experimental results show that FI outperforms the other four readability formulas as the core measure by extracting more meaningful relations. The results also demonstrate that among the four formulas, SMOG index and FKRI are competitive to be used like FI as the core measure of text denoising for biomedical relation mining.

The organization of the paper is as follows. In the next section, we describe text denoising as well as the readability formulas used in this research. Following that, Section 3.3 describes the methodology. Section 3.4 shows the experimental findings. Section 3.5 draws the conclu-

sion of the paper.

3.2 Background

In this section, we briefly describe the text denoising technique that extracts more content-rich sentences from full biomedical texts based on their FI scores. This is followed by an overview of the four readability formulas, FRES, SMOG, FORCAST, and FKRI. A detailed description of FI can be found in [2].

3.2.1 Text Denoising

One key aspect of biomedical papers is that they contain hidden or explicit relations, especially among drugs, chemicals, diseases, genes and proteins. Most of the proposed automated relation miners attempt to extract these relations from paper abstracts because they are easier to access and they are believed to contain biomedical content information. However, it is unlikely that abstracts will contain all important relations because they are at best the concise summaries of texts. For this reason, a number of biomedical ontologies like OMIM (Online Mendelian Inheritance in Man) and GO (Gene Ontology) use human annotators to extract relations from full texts. This is a time-consuming as well as error-prone procedure. To overcome these shortcomings, Shams and Mercer [2] have proposed a method, *Text Denoising*, that identifies those sentences in a text, called the denoised text, where content information, such as biomedical relations, is more likely to occur. The rest of the text is called the noise text. The authors suggested that the describing of biomedical relations lengthens sentences and increases the use of polysyllabic words. Some readability indexes, the Fog Index in particular, are based on these two factors. They proceeded to use Fog Index to measure sentence readability and showed experimentally that 30% of the low-readability sentences, the denoised part of a text, contain the relations of interest. Figure 3.1 shows the text denoising method.

To evaluate Text Denoising, the method was applied on a dataset of 24 full texts that describe four related pairs of disease and chemical components. The method extracted pairs of biomedical concepts from the denoised part of the dataset of which about 75% are reported as related by the Unified Medical Language System's (UMLS) semantic relation network[1]. It was also noted that the noise text did not contain any related biomedical entities of interest. These experimental findings supported the hypothesis of the authors that sentences that are difficult to read have the content information of the full text.

3.2.2 Overview of Readability Formulas

A brief description of the readability formulas is as follows (in historical order).

Flesch Reading Ease Score (FRES)

Considered as one of the oldest and most accurate readability indexes, FRES was developed to advocate a return to the phonics [13]. The formula uses two core measures—average sentence

[1]Accessible through UMLS Terminology Services (UTS) at https://uts.nlm.nih.gov/home.html

length and word length. It was originally developed to assess the grade-level of a reader. Its use now extends to questionnaire formulation in the US Department of Defense and medical form content assessment. Mathematically, FRES can be written as Eq. 3.1.

$$FRES = 206.835 - 1.015 \times \left(\frac{Total\ Words}{Total\ Sentences}\right) - 84.6 \times \left(\frac{Total\ Syllables}{Total\ Words}\right) \tag{3.1}$$

The FRES score spans the range 0 to 100, where scores between 90 and 100 are considered easily understandable by an average 5th grader and scores between 0 and 30 are considered easily understandable by university graduates.

SMOG Index

SMOG index, when first published, was anticipated as a proper substitute for FI due to its accuracy and ease of use [14]. A recent study claims that SMOG index should be the preferred formula when evaluating medical materials [15]. The formula for SMOG index counts the complex words (i.e., words that are polysyllabic) in three 10-sentence samples from documents of n sentences, takes the square root of the sum of the count normalized by n and 30, and then adds 3.1219 (Eq. 3.2).

$$SMOG\ Index = 1.043 \times \sqrt{30 \times \frac{Complex\ Words}{n}} + 3.1219 \tag{3.2}$$

The meaning of SMOG index is similar to FI—the index indicates the year of education required by the reader to understand the sentence. For example, a passage with a SMOG index of 12 means that to understand it, the reader should have 12 years of academic education.

FORCAST Index

The FORCAST index was originally formulated to assess the reading skills required by different military jobs without focusing on running narratives [16]. However, the index, unlike many other formulas, uses a single yet significant vocabulary element—the count of simple words (i.e., monosyllabic words). Due to its relative ease of use, the index was applied to write understandable publications by the U.S. Air Force. Eq. 3.3 is used to give the FORCAST index of any document.

$$FORCAST\ Index = 20 - \frac{N}{10} \tag{3.3}$$

where N is the number of monosyllabic words in a 150-word sample of the text. The index, like many other formulas, indicates the grade level of the reader required to understand the content of the text.

Flesch-Kincaid Readability Index (FKRI)

A second instalment of a readability index proposed by Flesch and further investigated and modified by Kincaid [17] eventually took the form of Eq. 3.4.

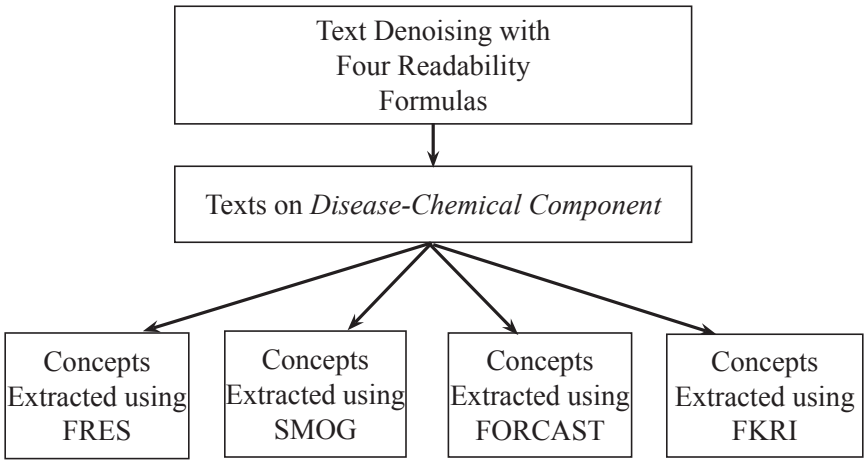


Figure 3.2: Experimental Procedure

$$FKRI = 0.39 \times \left(\frac{Total\ Words}{Total\ Sentences}\right) + 11.8 \times \left(\frac{Total\ Syllables}{Total\ Words}\right) - 15.59 \tag{3.4}$$

The score, like the other indexes, corresponds to a grade level. However, it correlates inversely with FRES due to different weighting factors. For example, a FKRI score of 10.1 would indicate that the text is anticipated to be understandable by any student studying in grade 10. Conversely, in the case of FRES, this score would indicate it as a low-readable text. Another key difference between them is that FKRI defines the lowest possible grade level score in theory, which is −3.40, although very few real-life passages comprise a single one-syllable word.

3.3 Methodology

In this section, we describe the dataset of 24 full texts, the experimental procedures and performance evaluation measures that we used in our experiment.

3.3.1 The Dataset

FI was successfully established as a core measure for text denoising in [2]. The trial used 24 biomedical texts as a test dataset divided into four sets. Each set describes one pair of concepts related with an explicit *disease-chemical component* relation reported by Perez-Iratxeta *et al.* [12]. The pairs of concepts are *Ischemia-Glutamate, Ataxia-Dehydrogenase, Hypogonadism-Gonadotropin,* and *Epilepsy-GABA*. In the experiment reported in this paper, we used the same dataset—providing the means to have a fair comparison. Key characteristics of the dataset are as follows:

- The texts have been randomly collected from PubMed paper repository[2] that can be described by the four concept pairs mentioned above.

[2]http://www.ncbi.nlm.nih.gov/pubmed/

- Texts have been preprocessed. For example, several sections of the texts like title, affiliations, tables, figures, acknowledgments, and references have been removed.
- Document size in terms of number of words varies.

Several annotation tasks for the dataset are still being carried out. However, for this experiment, we only needed the pre-processed full texts.

3.3.2 Procedure

The experimental procedure is shown in Figure 3.2. In our experiment, we followed the procedure described by Shams and Mercer [2] as shown in Figure 3.1, except that we used four different readability formulas other than the FI. The four formulas were applied one at a time on every sentence of the texts to provide each with a readability score. The sentences were then ranked based on this score. From these ranked sentences, 30% of the low-readable sentences were considered. Then, we used a co-occurrence frequency matrix (also known as association matrix) to find out the most frequently co-occurred concepts in the texts of which the 20 most frequent pairs were selected.

However, among these selected pairs of concepts, some lack representativeness (i.e., they do not hold any relation according to UMLS semantic relation network). These are called *noisy* pairs of concepts and needed to be removed. Because we randomly collected texts, we observed that it is possible for the pairs to never co-occur in a sentence which indicates that our data set is imbalanced. So, we used the equally weighted harmonic mean of the PPV and sensitivity of the pairs of concepts provided by FI to evaluate their representativeness as it is a great evaluation metric for imbalanced dataset [18].

PPV[3] is the proportion of correctly predicted relations and sensitivity is the proportion of relevant relations that are identified by our method. To measure these values, we considered the number of sentences extracted by the formulas which is the total number of results returned by the tool (R) that comprises the number of True Positives (TP) and False Positives (FP). Then, we took each pair in our co-occurrence frequency matrix and developed a second set of sentences that contain both the concepts. The number of sentences in this set is the number of results that should have been returned by our system (S) and comprises the number of True Positives (TP) and False Negatives (FN). The number of sentences that are present in both of these sets is the number of TP. Afterwards, FP is obtained by subtracting TP from R and FN is obtained by subtracting TP from S. So, the PPV of every pair of connected concepts is $\frac{TP}{TP+FP}$ and the sensitivity of every pair of connected concepts is $\frac{TP}{TP+FN}$. Eq. 3.5 is then used to determine the equally weighted harmonic mean for the given pair of concepts. In this way, we measured this mean for every pair of concepts in our co-occurrence matrix.

$$\textit{Harmonic Mean of PPV and Sensitivity} = 2 \times \left(\frac{\textit{PPV} \times \textit{Sensitivity}}{\textit{PPV} + \textit{Sensitivity}} \right) \tag{3.5}$$

These pairs are then re-ranked based on each of their PPV and sensitivity. From these re-ranked list, top 10 pairs of concepts were considered as the related concepts of the texts. As

[3]Similar to F-Score, Precision, and Recall but their use in Information Retrieval and Classification is different. The terms PPV and Sensitivity have been used to avoid confusion with the evaluation terminology.

Related Concepts	FI		SMOG		FKRI		FRES		FORCAST	
	Rank	Harmonic Mean	Rank	Harmonic Mean	Rank	Harmonic Mean	Rank	Harmonic Mean	Rank	Harmonic Mean
Ischemia-Glutamate	1	51.85	1	51.85	1	51.85	2	39.13	1	48.15
Levels-Glutamate	3	41.66	3	41.66	3	41.66			3	37.50
Glutamate-Neurons	4	39.02	4	39.02	4	39.02				
10Min-Ischemia	5	37.50	5	37.50	5	37.50	1	48.14	6	29.16
Glutamate-CA4	6	35.89								
Increase-Glutamate	7	32.55	6	32.55	6	32.55			4	32.55
Ischemia-5Min	9	31.57			8	31.57			5	31.57
Ischemia-DG					4	39.02				
Glutamate-Microdialysis							3	37.50		
Neurons-Ischemia							8	22.22		
CA1-Ischemia							9	15.78		
Glutamate-Release							6	27.77		
Levels-Ischemia	2	43.47	2	43.47	2	43.47			2	39.93
10Min-Glutamate	8	31.81	7	31.81	7	31.81	4	32.55	8	27.27
Glutamate-5Min	9	31.57	8	31.57					5	31.57
Ischemia-Release					6	32.55	10	15.00		
Experiment-Ischemia			9	27.77					7	27.77
Glutamate-Experiment			9	27.77					7	27.77
10Min-Release							5	31.57		
Increase-Ischemia							7	23.80		

Table 3.1: Relations extracted using the readability formulas from the papers on Ischemia and Glutamate

we have these 10 pairs of concepts per set of texts by each of the formulas, we divided them into two groups— (i) the first group contained the pairs of concepts that were reported to be related by UMLS semantic relation network; (ii) the second group was composed of pairs of concepts that do not have any semantic relation. The more pairs of concepts extracted using a readability formula in the first group, the better its performance is.

3.3.3 Evaluation Measures

Our evaluation of the readability forumlas is twofold. First, we are interested in knowing the number of meaningful pairs of concepts extracted using each of the formulas. The concepts in the tables 3.1–3.4 are divided into two segments. The upper segment of the table contains the first group of concepts while the lower segment of the table has concepts that belong to the second group.

Moreover, as we are motivated to see whether the four formulas can be used as core measures of text denoising, we evaluated the performances of the formulas against a gold standard. As FI is already proved to be an effective measure for text denoising, we considered the performance of FI as our gold standard. We considered the related concepts extracted using FI as the *positives* and examined the *true positives*, *false positives* and *false negatives* of a given formula, and calculated its *precision* and *recall*. In addition, we calculated both micro and macro average of precision and recall and hence the F-Score of every formula (Table 3.6). We calculated

Related Concepts	FI		SMOG		FKRI		FRES		FORCAST	
	Rank	Harmonic Mean	Rank	Harmonic Mean	Rank	Harmonic Mean	Rank	Harmonic Mean	Rank	Harmonic Mean
Friedreich-Ataxia	1	59.25	1	66.66	1	59.25	1	59.25	1	51.85
PDHC-Ataxia	2	56.00	2	56.00	2	56.00	3	48.00	3	39.99
Activity-Friedreich	3	43.47	6	43.47	3	43.47	7	34.78	2	43.47
Patients-Ataxia	3	43.47	3	52.17	3	43.47	2	52.17	5	34.78
Activity-Ataxia	3	43.47	6	43.47	3	43.47	7	34.78	2	43.47
PDHC-Friedreich	3	43.47	6	43.47	3	43.47	7	34.78	5	34.78
Patients-Friedreich	6	36.36	5	45.45			5	45.45		
Activity-PDHC					6	37.03			4	37.03
Preparations-Ataxia	4	40.00	7	40.00	4	40.00	6	40.00		
Preparations-Friedreich	4	40.00	7	40.00	4	40.00	6	40.00		
Pyruvate-Ataxia	5	38.09	4	47.61	5	38.09	4	47.61	6	28.57
Siblings-Ataxia									7	22.22
Disease-Pyruvate									8	21.05

Table 3.2: Relations extracted using the readability formulas from the papers on Ataxia and Dehydrogenase

the micro average as we have large number of sentences that differ from one set to the other as well as the macro average to see how the formulas performed across all sets [19]. To calculate the micro average, the *true positives, false positives* and *false negatives* were added up across every set first that are used to compute the statistics. On the other hand, the macro average was calculated by calculating the precision and recall for each instance first that is averaged over all instances in the reference standard.

3.4 Results and Discussion

From Table 3.1, interestingly both FI and FKRI as text denoising measures extracted seven pairs of relations, from texts on Ischemia and Glutamate. On the other hand, each of the rest of the formulas extracted five relations. Between FI and FKRI, the prior performed slightly better than the latter as most of its top-ranked pairs were semantically related. Similarly, on the other hand, SMOG performed marginally better than FRES and FORCAST as most of its meaningful relations had low harmonic mean. It is noteworthy that the ranks and harmonic means of the relations for the first three formulas were somewhat similar to each other—means that they extracted almost the same sentences.

In Table 3.2, the relations extracted using the formulas from the papers on Ataxia and Dehydrogenase are displayed. It is surprising that all of the formulas extracted exactly seven meaningful relations. In this case, the performance of FI and FKRI were almost identical. On the other hand, both SMOG and FRES extracted the same related concepts like FI but their harmonic means largely differed. Careful observations sustain that FI and SMOG performed best in this case followed by FKRI and FRES.

Table 3.3 displays the relations extracted by the readability formulas from the papers on Hypogonadism and Gonadotropin. Again, FI, FKRI, SMOG, and FORCAST each extracted eight pairs of concepts that are semantically related. However, FI, like the previous cases,

Related Concepts	FI		SMOG		FKRI		FRES		FORCAST	
	Rank	Harmonic Mean	Rank	Harmonic Mean	Rank	Harmonic Mean	Rank	Harmonic Mean	Rank	Harmonic Mean
AAS-Treatment	1	29.41	1	29.41	1	32.35	1	23.52	1	29.41
AAS-Testosterone	3	18.46	6	15.38	3	18.46			3	24.61
Gonadotropin-Treatment	4	18.18	2	18.18	4	18.18	2	21.21	8	12.12
Testosterone-Treatment	5	14.92	8	14.92	5	17.91	3	17.91	9	11.94
Levels-Testosterone	6	14.49	3	17.39	7	14.49			10	11.59
AAS-Conditions	7	12.90								
Treatment-HCG	7	12.90	9	12.90	8	12.90	6	12.90		
Treatment-Therapy	7	12.90	5	16.12	6	16.12	4	16.12	6	12.90
Gonadotropin-Testosterone			8	14.92	9	11.94	5	14.92		
Clomiphene-Citrate									4	24.32
Tamoxifen-Citrate									5	20.28
Use-AAS	2	21.62	4	16.21	2	18.91	8	10.81	2	27.02
Replacement-Therapy	7	12.90								
AAS-Conditions					8	12.90				
Testosterone-Production			7	15.15						
Function-Testosterone							7	12.30		
Therapy-Gonadotropin							9	10.00		
Function-Testosterone							10	9.67		
Therapy-Testosterone									7	12.69

Table 3.3: Relations extracted using the readability formulas from the papers on Hypogonadism and Gonadotropin

performed the best as it has most of its higher ranked pairs semantically related. FKRI and SMOG performed almost similar and better than FORCAST that gave many concepts a low-rank though they are semantically related. The performance of FRES is the poorest among the five as it extracted four pairs of concepts without semantic relations.

Table 3.4 shows the related concepts extracted using the formulas from the papers on Epilepsy and GABA. FI outperformed others by extracting seven semantically related concepts. FKRI, SMOG, and FRES extracted five related concepts each but the performance of FKRI is better than the other two. Between SMOG and FRES, most of the low-ranked pairs of concepts extracted using the prior lack meaning than the latter. On the other hand, FORCAST performed really poor in this case by extracting only two semantically related pairs.

Table 3.5 displays the performance of the four readability formulas—FKRI, SMOG, FRES, and FORCAST—against the gold standard. From the table, it can be seen that the performance of FKRI and SMOG was consistent throughout the four sets of papers. For all sets of papers, either of the formulas had the best F-Score. On the other hand, the F-scores of FRES and FORCAST largely varied for the papers and either of them had the lowest F-Score.

Table 3.6a displays that SMOG index had the best F-Score of 88.46% and it was more precise than the others. However, FKRI showed the best recall followed by SMOG index. On the other hand, FRES and FORCAST had identical precision and recall but their significanly poor recalls cost them lower F-Scores. Information on Table 3.6b shows similar results except that both SMOG index and FKRI achieved the best recall. From this analysis, it can be said

Related Concepts	FI		SMOG		FKRI		FRES		FORCAST	
	Rank	Harmonic Mean	Rank	Harmonic Mean	Rank	Harmonic Mean	Rank	Harmonic Mean	Rank	Harmonic Mean
Inhibition-GABA	1	26.08	1	28.16	1	34.78	1	34.78	1	23.91
GABA-Synapse	2	20.25	2	20.25	2	22.78	2	20.25		
Neurons-Synapse	3	14.17	4	14.70	3	14.70	5	11.76		
Inhibition-Hippocampus	4	12.30			4	10.66				
Neurons-GABA	6	8.00	3	16.00			6	10.66		
Properties-GABA	7	6.45	7	9.67	5	9.67	7	9.67		
Cl-Gradient	9	3.33								
Inhibition-Dentate Gyrus									6	9.37
Synapse-Change	5	9.37	8	9.37	7	9.37	4	12.50		
GABA-Change	7	6.45	7	9.67	9	6.45				
GABA-Number	8	6.34	5	12.69	6	9.52	8	9.52		
Synapse-Number					8	6.55				
Neuron-Input					9	6.45				
Animal-Models			6	11.26					3	14.08
Neurons-Inhibition			7	9.67						
GABA-Alteration							3	18.18		
Study-Tissue							9	9.37		
Study-Inhibition							10	8.95		
Rat-Inhibition									2	20.51
Slices-Inhibition									4	12.50
Animal-Rat									5	12.30
Cortex-slices									7	9.09
Epilepsy-Rat									8	8.95
Inhibition-Kindling									8	8.95
Number-Tissue									9	6.45

Table 3.4: Relations extracted using the readability formulas from the papers on Epilepsy and GABA

that SMOG index and FKRI are both performing similar to FI and thus can be used to reduce textual noise and extract related biomedical concepts.

3.5 Conclusions

While FI has been used in text denoising to make it a meaningful relations extraction tool for biomedical texts, we reported the performance of four other readability formulas, namely FKRI, SMOG, FRES, and FORCAST on this task. We applied the formulas to the sentences of 24 biomedical texts, ordered them according to their reading difficulty, and extracted frequently co-occurred concepts from the 30% of the low-readable sentences. These concepts were then re-ranked according to the harmonic mean of their PPV and sensitivity. A comparative result shows that FI outperformed the other formulas by extracting more meaningful relations according to UMLS semantic relation network. We also analyzed the performance of the formulas considering the performance of FI as a gold standard. It shows that SMOG index achieved the best F-Score followed by FKRI while FRES and FORCAST performed poorly. It can also be noted that SMOG index, like FI, uses the core measure of *complex words* and its performance

Readability Formula	Precision	Recall	F-Score
SMOG	100.00	71.43	83.33
FKRI	85.71	85.71	85.71
FRES	40.00	28.57	33.33
FORCAST	100.00	71.43	83.33

(a)

Readability Formula	Precision	Recall	F-Score
SMOG	100.00	100.00	100.00
FKRI	85.71	85.71	85.71
FRES	100.00	100.00	100.00
FORCAST	85.71	85.71	85.71

(b)

Readability Formula	Precision	Recall	F-Score
SMOG	87.50	87.50	87.50
FKRI	87.50	87.50	87.50
FRES	83.33	62.50	71.43
FORCAST	75.00	75.00	75.00

(c)

Readability Formula	Precision	Recall	F-Score
SMOG	100.00	71.43	83.33
FKRI	100.00	71.43	83.33
FRES	100.00	71.43	83.33
FORCAST	50.00	14.29	22.22

(d)

Table 3.5: Performance of the four readability formulas for the papers on (a) Ischemia and Glutamate, (b) Ataxia and Dehydrogenase, (c) Hypogonadism and Gonadotropin (d) Epilepsy and GABA

Readability Formula	Precision	Recall	F-Score
SMOG	95.83	82.14	88.46
FKRI	88.89	82.76	85.71
FRES	82.61	65.52	73.08
FORCAST	81.82	62.07	70.59

(a)

Readability Formula	Precision	Recall	F-Score
SMOG	96.88	82.60	89.16
FKRI	89.73	82.60	86.01
FRES	80.83	65.63	72.44
FORCAST	77.88	61.61	68.72

(b)

Table 3.6: Average precision, recall and F-Score of the formulas with (a) micro-average and (b) macro-average methods

is the best compared to the gold standard which reveals the fact that the measure of complex word fits best for text denoising and biomedical relation extraction.

As for relation mining we found at least two competitive measures for text denoising other than FI, their performances on training data reduction for keyphrase indexers can be of great interest. This task is left as future work.

Bibliography

[1] O. S. Goh, C. C. Fung, A. Depickere, and K. W. Wong, "Using gunnnig-fog index to assess instant messages readability from ecas," in *Proceedings of the Third International Conference on Natural Computation*, vol. 5 of *ICNC '07*, (Washington, DC, USA), pp. 480–486, 2007.

[2] R. Shams and R. E. Mercer, "Extracting connected concepts from biomedical texts using fog index," *Procedia - Social and Behavioral Sciences*, vol. 27, pp. 70–76, 2011.

[3] T. M. Duffy and P. Kabance, "Testing a readable writing approach to text revision," *Journal of Educational Psychology*, vol. 74, pp. 733–48, 1982.

[4] I. Witten, G. Paynter, E. Frank, C. Gutwin, and C. Nevill-Manning, "KEA: Practical automatic keyphrase extraction," in *Proceedings of the 4th ACM Conference on Digital Libraries*, (Berkeley, CA, USA), pp. 254–255, 1999.

[5] O. Medelyan and I. Witten, "Domain-independent automatic keyphrase indexing with small training sets," *Journal of the American Society for Information Science and Technology (JASIST)*, vol. 59, no. 7, pp. 1026–1040, 2008.

[6] O. Medelyan, *Human-competitive automatic topic indexing*. PhD thesis, University of Waikato, New Zealand, 2009.

[7] R. Shams and R. E. Mercer, "Investigating keyphrase indexing with text denoising," in *Proceedings of the 11th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2012)*, (Washington DC, USA), 2012.

[8] R. Shams and R. E. Mercer, "Improving supervised keyphrase indexer classification of keyphrases with text denoising," in *14th International Conference on Asia-Pacific Digital Libraries (ICADL 2012)*, (Taipei, Taiwan), 2012.

[9] J. Bogert, "In defense of the fog index," *Business Communication Quarterly*, vol. 48, pp. 9–12, 1985.

[10] E. Fry, "A readability formula that saves time," *Journal of Reading*, vol. 11, pp. 512–16 cont. 575–78, 1968.

[11] K. Koenke, "Another practical note on readability formulas," *Journal of Reading*, vol. 15, p. 205, 1971.

[12] C. Perez-Iratxeta, P. Bork, and M. Andrade, "Literature and genome data mining for prioritizing disease-associated genes," in *Discovering Biomolecular Mechanisms with Computational Biology* (F. Eisenhaber, ed.), Molecular Biology Intelligence Unit, pp. 74–81, Springer, 2006.

[13] R. Flesch, "A new readability yardstick," *Journal of Applied Psychology*, vol. 32, pp. 221–33, 1948.

[14] G. H. McLaughlin, "Smog grading – a new redability formula," *Journal of Reading*, vol. 12, no. 8, pp. 639–46, 1969.

[15] P. Fitzsimmons, B. Michael, J. Hulley, and G. Scott, "A readability assessment of online parkinson's disease information," *The Journal of the Royal College of Physicians of Edinburgh*, vol. 40, no. 4, p. 2926, 2010.

[16] J. S. Caylor, T. G. Stitch, L. C. Fox, and J. P. Ford, "Methodologies for determining reading requirements of military occupational specialities," Tech. Rep. 73-5, Human Resources Research Organization, Alexandria, VA, 1973.

[17] J. P. Kincaid, R. P. Fishburne, R. L. Rogers, and B. S. Chissom, "Derivation of new readability formulas (automated readability index, fog count, and flesch reading ease formula) for navy enlisted personnel," Research Branch Report 8-75, Chief of Naval Technical Writing: Naval Air Station Memphis, 1975.

[18] O. Frunza and D. Inkpen, "Extraction of disease-treatment semantic relations from biomedical sentences," in *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing*, BioNLP '10, (Stroudsburg, PA, USA), pp. 91–98, Association for Computational Linguistics, 2010.

[19] C. Manning and H. Shutze, *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press, 1999.

Chapter 4

Investigating Keyphrase Indexing with Text Denoising

This chapter is based on the paper titled "Investigating Keyphrase Indexing with Text Denoising" co-authored with Robert E. Mercer that appeared in the 12th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2012).

In this paper, we report on indexing performance by a state-of-the-art keyphrase indexer, Maui, when paired with a text extraction procedure called *text denoising*. Text denoising is a method that extracts the denoised text, comprising the content-rich sentences, from full texts. The performance of the keyphrase indexer is demonstrated on three standard corpora collected from three domains, namely food and agriculture, high energy physics, and biomedical science. Maui is trained using the full texts and denoised texts. The indexer, using its trained models, then extracts keyphrases from test sets comprising full texts, and their denoised and noise parts (i.e., the part of texts that remains after denoising). Experimental findings show that against a gold standard, the denoised-text-trained indexer indexing full texts, performs either better than or as good as its benchmark performance produced by a full-text-trained indexer indexing full texts.

4.1 Introduction

Today, most of the automatic indexers use supervised full-text classifiers to extract keyphrases from full texts [1] [2] [3] [4]. Maui [2] is the final successor of a legacy of keyphrase indexers and inherits from and builds upon both of its predecessors KEA [4] and KEA++ [3]. To extract keyphrases from test documents, Maui uses 13 features to develop its supervised classifier. However, a revealing experiment by Witten *et al.* [4] demonstrates that indexer performances depend not only on the set of features but also on document size. As they apply their full-text trained KEA on paper abstracts and compare against a gold standard, they find its performance on these reduced texts somewhat inferior and not competitive to that on full texts. Although Maui outperforms its predecessors to extract full-text keyphrases from food and agriculture, nuclear physics, and biomedical science texts [2], it has not been tested with a reduced set of texts till date.

Text Denoising is a method proposed by Shams and Mercer [5] which reduces the amount

of text in biomedical papers to 30% of the original. The authors suggested that the describing of biomedical relations lengthens sentences and increases the use of polysyllabic words. Some readability indexes, the Fog Index [6] in particular, are based on these two factors. They proceeded to use Fog Index to measure sentence readability and showed experimentally that the 30% of the sentences which had the lowest-readability, the denoised part of a text, contained the relations of interest. The rest is termed as *noise text* that are not as content-rich as denoised text.

In this paper, we report on the performance of a state-of-the-art keyphrase indexer named Maui [2] when paired with text denoising. We use three standard full-text corpora from the food and agriculture, nuclear physics, and biomedical science domains. From each corpus, we develop training sets comprising full texts and their denoised parts. The test sets are composed of full texts, and their denoised and noise parts. For training and testing each dataset, we use a standard 10-fold cross validation. We show experimentally that a 70% text denoising improves Maui's indexing performance. To evaluate Maui, we use quantitative measures like precision, recall and F-score, as well as qualitative measures like inter-indexer agreements. Experimental results show that Maui, trained with denoised texts, performs either better or comparably to its benchmark performance—those with full-text trained models to extract keyphrases from full-text test sets. Further details about this paper can be found in its full version at `arXiv:1204.2231v1 [cs.DL]`.

The remainder of this paper discusses the methods for training and testing the indexer (Section 4.2), an analysis of the results (Section 4.3), and ends with some concluding remarks (Section 4.4).

4.2 Methodology

In this section, we describe the datasets, training and testing procedure, performance measures, and the means to find the appropriate text denoising threshold for keyphrase indexing.

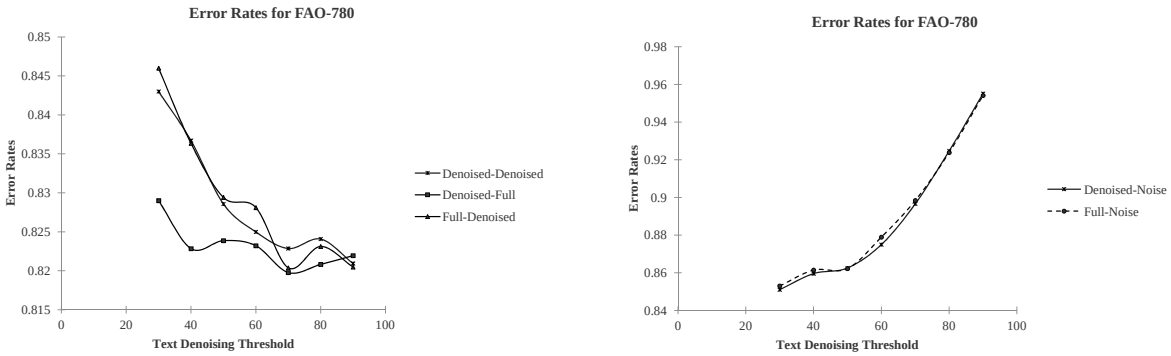
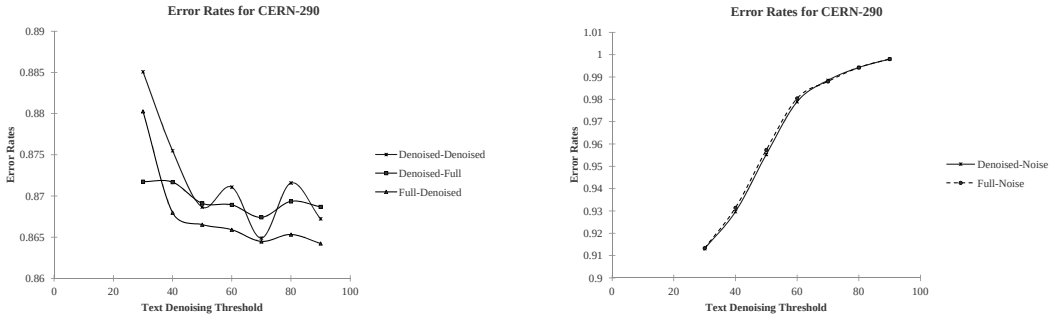
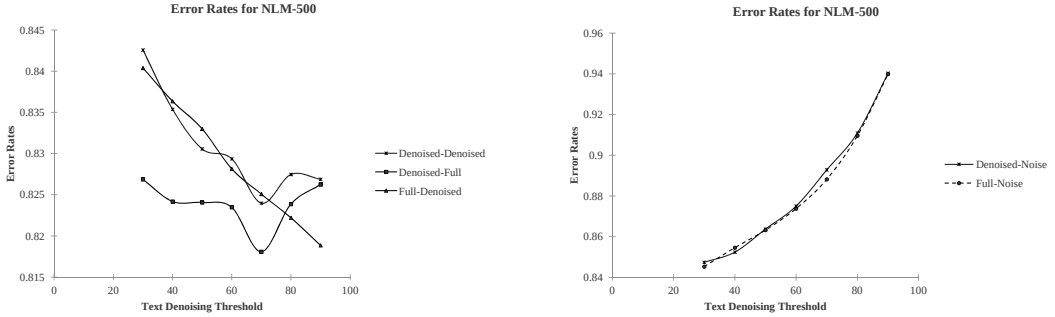


Figure 4.1: Error rates for different denoising thresholds with FAO-780 (a) denoised texts and (b) noise texts



(a) (b)

Figure 4.2: Error rates for different denoising thresholds with CERN-290 (a) denoised texts and (b) noise texts



(a) (b)

Figure 4.3: Error rates for different denoising thresholds with NLM-500 (a) denoised texts and (b) noise texts

4.2.1 Datasets

To train and test Maui, we use three standard corpora of full texts and keyphrases associated with them. Among the datasets are FAO-780 which comprises 780 full texts from food and agriculture; CERN-290 which is composed of 290 high energy physics documents; and NLM-500 which has 500 biomedical research articles. The details about these datasets can be found in the doctoral thesis by Medelyan [2].

4.2.2 Training and Testing

In this experiment, we use a conventional stratified k-fold experimental approach set by Medelyan [2] except that we train Maui not only on full texts but also on their denoised parts and test it on full texts as well as their denoised and noise parts. We consider the full texts along with their denoised and noise parts from each dataset and divide them randomly into 10 equal-sized folds without overlap. We train Maui on the training sets comprising full texts and denoised texts from each fold. The models that the indexers develop from full texts are called *full-text trained models* and those that are developed from denoised texts are called *denoised-text trained models*. The indexer then applies these models, k-th full-text trained model and k-th denoised text trained model, to extract keyphrases from the k-th test set composed of full texts, and their

					Benchmark Performance			
Trained Model	**Test Set**	**Precision**	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**	***t* value**
Denoised Text	Denoised Text	30.02	32.92	31.36	30.56	33.47	31.86	2.23
Denoised Text	**Full Text**	**30.49**	**33.50**	**31.87**				**2.76**
Full Text	Denoised Text	30.48	32.96	31.63				1.81

(a) Maui's performance on FAO-780 dataset with denoising level of 70% of each text

					Benchmark Performance			
Trained Model	**Test Set**	**Precision**	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**	***t* value**
Denoised Text	Denoised Text	24.38	25.33	24.79	24.58	25.56	24.99	2.16
Denoised Text	Full Text	23.99	24.95	24.42				2.26
Full Text	Denoised Text	24.38	25.40	24.82				1.31

(b) Maui's performance on CERN-290 dataset with denoising level of 70% of each text

					Benchmark Performance			
Trained Model	**Test Set**	**Precision**	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**	***t* value**
Denoised Text	Denoised Text	29.14	32.36	30.66	29.69	32.74	31.13	2.01
Denoised Text	**Full Text**	**29.96**	**33.22**	**31.50**				**3.52**
Full Text	Denoised Text	28.99	32.00	30.40				1.85

(c) Maui's performance on NLM-500 dataset with denoising level of 70% of each text

Table 4.1: Precision, recall and F-score of Maui with text denoising

denoised and noise parts. According to the average number of keyphrases in every document, we had the indexers extract 8 keyphrases, 7 keyphrases, and 15 keyphrases for each document in the FAO-780, CERN-290, and NLM-500 test sets, respectively. The extracted keyphrases are then compared against a gold standard which are the author-assigned keyphrases associated with the test documents.

4.2.3 Performance Measures

In this experiment, as well as conventional quantitative performance measures like precision, recall and F-score, we use three inter-indexing agreement measures popularly used for qualitative indexing assessment [7], namely Hooper's (H) [8], Rolling's (R) [9] and Cosine (C) inter-indexing agreements. The closer the agreement measures are to 1, the more the indexers agree on extracted keyphrases. We also calculate the error rates for each cross validation to find a text denoising threshold described in Section 4.2.4 and measure a *10-fold cross validated paired t-test* [10] to report statistical significance of Maui's indexing when paired with denoised texts. For any two given sets of results, we consider their error rates to calculate a paired t-value. If this calculated t-value lies outside ± 2.26 with a degree of freedom 9, then the difference between the set whose results have the lower error rate and the other set is said to be statistically significant at significance level $\alpha = 0.05$.

4.2.4 Text Denoising Threshold

To find the appropriate text denoising threshold for keyphrase indexing, we evaluate Maui's performance on each dataset by increasing the text denoising threshold in increments of 10% from 30% to 90%. As we vary the threshold, we plot the error rates of Maui on different test

					Benchmark Performance		
Trained Model	**Test Set**	**Hooper**	**Rolling**	**Cosine**	**Hooper**	**Rolling**	**Cosine**
Denoised Text	Denoised Text	0.18	0.29	0.30	0.18	0.30	0.31
Denoised Text	**Full Text**	**0.18**	**0.30**	**0.31**			
Full Text	Denoised Text	0.18	0.30	0.30			

(a) Maui's indexing agreements on FAO-780 dataset with denoising level of 70% of each text

					Benchmark Performance		
Trained Model	**Test Set**	**Hooper**	**Rolling**	**Cosine**	**Hooper**	**Rolling**	**Cosine**
Denoised Text	Denoised Text	0.14	0.24	0.24	0.14	0.24	0.24
Denoised Text	Full Text	0.14	0.24	0.24			
Full Text	Denoised Text	0.14	0.24	0.24			

(b) Maui's indexing agreements on CERN-290 dataset with denoising level of 70% of each text

					Benchmark Performance		
Trained Model	**Test Set**	**Hooper**	**Rolling**	**Cosine**	**Hooper**	**Rolling**	**Cosine**
Denoised Text	Denoised Text	0.18	0.30	0.30	0.18	0.30	0.31
Denoised Text	**Full Text**	**0.19**	**0.31**	**0.31**			
Full Text	Denoised Text	0.18	0.30	0.30			

(c) Maui's indexing agreements on NLM-500 dataset with denoising level of 70% of each text

Table 4.2: Inter-indexing agreements of Maui with text denoising

sets. Being a supervised indexer, Maui's best-fit model should be where the test error has its global minimum. That point eventually will also be the denoising threshold.

The error rates for different denoising thresholds for FAO-780 are plotted in Figure 4.1. It is notable that for both trained models, Maui has its global minimum at 70% denoising (Figure 4.1a). From this point on, the error rate increases and thus indicates an overfitting in Maui's models. Figure 4.1 shows that Maui's best performing pair is *Denoised-Full*—those models that are trained with denoised texts for keyphrase extraction from full texts. Similarly, Maui's best-fitted models with denoised texts for CERN-290 and NLM-500 are also at the 70% threshold (Figure 4.2a and 4.3a). Figure 4.1b, 4.2b, and 4.3b, on the other hand, show that the error rate for the noise test sets increases. This indicates that noise texts are less content-rich as Maui fails to extract a substantial number of keyphrases from them. These observations lead us to set the denoising threshold at 70%. At this threshold, Maui predicts keyphrases from unseen test examples most accurately.

4.3 Results and Discussions

In this section, we discuss the performance of Maui with text denoising and compare this with its benchmark performance—full-text trained model on full texts.

Table 4.1a compares the precision, recall and F-score of Maui with denoised texts and its benchmark performance on the FAO-780 dataset. Maui, using its denoised-text and full-text trained models on denoised text test sets, achieves F-scores of 31.36 and 31.63, respectively, compared to its benchmark F-score of 31.86. We see that for these two cases, the t-values

are 2.23 and 1.81, respectively, which means that the differences between the F-scores are not statistically significant. In other words, the benchmark performance of Maui cannot be said to be different than that with text denoising. On the other hand, Maui's F-score with its denoised text trained model on full-text keyphrase extraction is 31.87 with a t-value of 2.76. So, with 98% confidence we can say that the result is better than the benchmark performance. In addition, from Table 4.2a, we can see that Maui's agreements with the gold standards are as good as the benchmark agreements. This demonstrates that the indexing quality of Maui has not been compromised with text denoising.

For the CERN-290 dataset, although Maui could not outperform its benchmark F-score of 24.99, none of the t-values are significant at $\alpha = 0.05$. In other words, its performance with denoised texts cannot be said to be different than its benchmark performance with a 95% confidence level (Table 4.1b). Interestingly enough, although Maui agrees less with the gold standard for the CERN-290 dataset than that for FAO-780, its agreements on keyphrases with denoised texts are as good as its benchmark performance (Table 4.2b).

Maui's best performance for the NLM-500 corpus is with a denoised text trained model on full texts. Its F-score of 31.50 outperforms the benchmark F-score of 31.13 at the significance level of $\alpha = 0.05$. However, although its other two F-scores with denoised texts is somewhat lower than its benchmark F-score, they are not statistically significant with t-values of 2.01 and 1.85 (Table 4.1c). Maui's inter-indexing agreement on NLM-500 is somewhat similar to that on FAO-780 except that it agrees more with NLM-500 gold standards than its benchmark performance (Table 4.2c).

4.4 Conclusions

In this paper, we show that a 70% text denoising improves Maui's performance for the biomedical science texts, or it allows Maui to perform as good as its benchmark performance on the food and agriculture, and the physics texts. Although there are some cases where Maui, when paired with text denoising, experiences marginally lower F-score than its benchmark, indexing agreement measures show that its indexing quality has never been compromised; it extracts even better quality keyphrases from biomedical texts than its benchmark. From the experimental findings, we also can conclude that document size, per se, does not have the suggested effect on keyphrase indexing, rather it is the content richness that plays the key role in indexing. We leave the investigation on the effect of text denoising paired with other indexers as future work.

Bibliography

[1] K. Frantzi, S. Ananiadou, and J. Tsujii, "The c-value/nc-value method of automatic recognition for multi-word terms," in *Proceedings of the 2nd European Conference on Research and Advanced Technology for Digital Libraries (ECDL'98)*, (London, UK), pp. 585–604, 1998.

[2] O. Medelyan, *Human-competitive automatic topic indexing*. PhD thesis, University of Waikato, New Zealand, 2009.

[3] O. Medelyan and I. Witten, "Domain-independent automatic keyphrase indexing with small training sets," *Journal of the American Society for Information Science and Technology (JASIST)*, vol. 59, pp. 1026–1040, 2008.

[4] I. Witten, G. Paynter, E. Frank, C. Gutwin, and C. Nevill-Manning, "Kea: Practical automatic keyphrase extraction," in *Proceedings of the 4th ACM conference on Digital Libraries*, pp. 254–255, ACM Press, 1999.

[5] R. Shams and R. E. Mercer, "Extracting connected concepts from biomedical texts using fog index," *Procedia - Social and Behavioral Sciences*, vol. 27, pp. 70 – 76, 2011. Elsevier Science.

[6] R. Gunning, "Fog index after twenty years," *Journal of Business Communication*, vol. 6, pp. 3–13, 1969.

[7] O. Medelyan and I. Witten, "Measuring inter-indexer consistency using a thesaurus," in *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital libraries (JCDL-2006)*, (NC, USA), pp. 274–275, ACM Press, 2006.

[8] R. S. Hooper, "Indexer consistency tests: Origin, measurements, results and utilization," *IBM, Bethesda*, 1965.

[9] L. Rolling, "Indexing consistency, quality and efficiency," *Information Processing and Management*, vol. 17, pp. 69–76, 1981.

[10] E. Alpaydin, "Assessing and comparing classification algorithms," in *Introduction to Machine Learning*, pp. 342–343, Cambridge, UK: The MIT Press, 2004.

Chapter 5

Improving Supervised Keyphrase Indexer Classification of Keyphrases with Text Denoising

This chapter is based on the paper titled "Improving Supervised Keyphrase Indexer Classification of Keyphrases with Text Denoising" co-authored with Robert E. Mercer that appeared in the 14th International Conference on Asia-Pacific Digital Libraries (ICADL 2012).

In this paper, we show that two state-of-the-art supervised keyphrase indexers named KEA and KEA++, when paired with text denoising, induce improved keyphrase classifiers. The classifiers' performances are demonstrated on three standard full-text corpora collected from the food and agriculture, nuclear physics and biomedical domains. Using the denoised parts of the texts, the indexers induce keyphrase classifiers that are later used for full-text keyphrase extraction. Experimental results show that against a gold standard these classifiers perform better than those induced from full texts.

5.1 Introduction

The retrieval of appropriate documents from digital libraries becomes easier and more efficacious with the keyphrases assigned to them. In addition, when a document is properly meta-tagged with keyphrases, its accessibility increases because it is more likely to be cited by others. Often, authors assign keyphrases to their documents manually. This manual indexing has some drawbacks such as poor choice of keyphrases [1]. As a consequence, an array of automatic keyphrase indexers has been developed. Some examples are supervised indexers that induce classifiers from full texts and apply them to extract keyphrases from unseen documents [2] [3] [4] [5] [6], among many.

Besides the choice of learning algorithm and set of features, an indexer's predictive accuracy depends on the document size [6]. For instance, when the keyphrases extracted from paper abstracts by the Keyphrase Extraction Algorithm (hereinafter, KEA) with its classifier induced from full texts [6] are compared against a gold standard, its performance is found somewhat inferior and not competitive to that on full texts. Since this result, researchers have sidelined the task of improving indexing performance using a set of reduced texts. Recently, Shams and

Mercer [7] investigate the performance of a supervised indexer called Maui [3] by reducing its training data. Their experimental findings show that on biomedical texts, Maui generates a superior classifier with a text reduction method called *Text Denoising*. This work complements that investigation with two more indexers whose supervised classifiers work differently than that of Maui, and thus shows increased effectiveness of text denoising in this domain.

Text Denoising is a heuristic-based text extraction method. It reduces the amount of text in biomedical papers based on sentence readability [8]. The hypothesis behind this heuristic method is—the more difficult a sentence is to read, the more content-rich it is. The approach uses the Fog Index readability score [9] to rank the sentences of a text. Among these sentences, it then selects those that are the most difficult to read. This selected part of the text is called the *denoised text* and the remainder is called the *noise text*. This introductory work demonstrates that the denoised texts contain most of the biomedical relations of a corpus. Later, an investigation is pursued to observe the effect of text denoising on keyphrase indexing [7]. The authors conclude that text denoising can be used as an indicator of the content-rich parts of texts not only for relation mining but also keyphrase indexing.

In this paper, we show that rather than simply increasing the amount of training data, supervised keyphrase indexing can be improved by selectively reducing the training data to high quality sentences. We demonstrate that the keyphrase classifier performance of two state-of-the-art keyphrase indexers named KEA [1] [6] and KEA++ [4] improves if induced from texts reduced by the text denoising method. We use three standard full-text corpora from the food and agriculture, high energy physics, and biomedical science domains. The indexers induce their keyphrase classifiers from the denoised texts of each corpus. These classifiers are then used to extract keyphrases from full test documents. To evaluate the indexing performances, we use a standard 10-fold cross validation. As performance measures, we use standard quantitative measures like precision, recall, and F-score as well as quality indicators like indexing agreements. Experimental results show that the performances of the keyphrase classifiers induced from denoised texts are significantly better than their benchmarks achieved using keyphrase classifiers induced from full texts.

The remainder of this paper provides background on text denoising and the indexers (Section 5.2), discussions on the methods for training and testing the indexers (Section 5.3), an analysis of the results (Section 5.4), and ends with some concluding remarks (Section 5.5).

5.2 Background

In this section, we briefly discuss the text denoising method followed by KEA and KEA++.

5.2.1 Text Denoising

In their paper, Witten *et al.* [6] have demonstrated that the performance of KEA has been reduced when extracting keyphrases from paper abstracts. Similarly, the performance of biomedical relation miners that attempt to extract relations among drugs, chemicals, diseases, genes and proteins from paper abstracts is poor enough that a number of biomedical ontologies like OMIM (Online Mendelian Inheritance in Man) and GO (Gene Ontology) use human annotators to extract relations from full texts, a time-consuming and error-prone procedure. To overcome

these shortcomings, Shams and Mercer [8] proposed a method to identify those areas within a text, called denoised text, where content information, such as biomedical relations, is more likely to occur. The authors suggested that the describing of biomedical relations lengthens sentences and increases the use of polysyllabic words. Some readability indexes, the Fog Index [9] in particular, are based on these two factors. They proceeded to use the Fog Index to measure sentence readability and showed experimentally that the 30% of the sentences which had the lowest-readability, the denoised part of a text, contained the relations of interest.

Therefore, by setting the denoising threshold at 30% (i.e., by keeping the 30% of the most difficult-to-read sentences), text denoising has been evaluated with a corpus comprising 24 full texts that describe four related pairs of disease and chemical components. This method extracted pairs of biomedical concepts from the denoised part of the texts of which about 75 percent are reported as related according to the Unified Medical Language System's (UMLS) semantic relations network. It is noteworthy that the rest of the text, called noise text, did not contain any related biomedical concepts of interest.

5.2.2 The Keyphrase Indexers

KEA [6] is a supervised keyphrase indexer that induces Naïve Bayes classifiers using three features, namely *tf-idf*, first occurrence, and keyphraseness. When tested with the Computer Science Technical Reports (CSTR) corpus of NZDL, KEA performed reasonably well on domain-specific documents [1]. In their paper, Witten *et al.* [6] outlined some important aspects of keyphrase indexing, with or without a set of reduced texts: (i) keyphrases assigned by authors (also regarded as the gold standard) that are absent in the document affect indexer accuracy; (ii) machine learning algorithms perform as expected; (iii) the indexer performance is not on a par with human indexers; and (iv) reduced training data (e.g., that developed from abstracts) decreases accuracy for full-text indexing, etc.

KEA++ [4], which inherits from and builds upon its successor KEA, adds an advantage of using controlled vocabularies in SKOS (Simple Knowledge Organization System) format as an auxiliary source of keyphrases during the training phase. This minimized the effect of missing author assigned keyphrases in the documents. The authors, in addition, included the length of the keyphrases as a new feature. KEA++ was tested with corpora across different domains like food and agriculture, biomedical science, and physics. Although the performance is not comparable to that of English, it was also tested with French and Spanish texts. Experimental outcomes showed that a controlled vocabulary contributes to faster indexing and an improved accuracy over free-text indexing. The authors also credited the two new features for the noticeable improvement in precision but zeroed in on improved vocabulary construction for the better recall.

5.3 Methodology

In this section, we describe the datasets, training and testing procedure, performance measures, and the means to find the appropriate text denoising threshold for keyphrase indexing.

5.3.1 Datasets

To train and test KEA and KEA++, we used three standard corpora of full texts and keyphrases associated with them. Among the datasets are: FAO-780 which comprises 780 full texts from food and agriculture; CERN-290 which is composed of 290 high energy physics documents; and NLM-500 which has 500 biomedical research articles. The details about these datasets can be found in the doctoral thesis by Medelyan [3].

5.3.2 Training and Testing

In this experiment, we decided to use a conventional *k*-fold experimental approach (Fig. 5.1). We divided each dataset randomly into 10 equal-sized folds without any overlap. For each fold, we applied text denoising on the texts and kept both the full texts and their denoised parts. We then applied a standard 10-fold cross validation to train and test the indexers.

To generate each training-testing pair, we kept one of the 10 full-text folds out as our test set and combined the rest of the 9 folds as our training set. Doing this 10 times, each time leaving out a different fold from the 10 folds as a test set, we generated 10 pairs. KEA and KEA++ then induced classifiers from the training sets comprising denoised texts from each fold. In this way, they developed 10 classifiers for each dataset. The classifiers induced from denoised texts are called *denoised-text induced classifiers*. The indexers then used the *k*-th denoised text induced classifier to extract keyphrases from the *k*-th test set of full texts.

According to the average number of keyphrases in every document, we had the indexers extract 8 keyphrases, 7 keyphrases, and 15 keyphrases for each document in the FAO-780, CERN-290, and NLM-500 test sets, respectively. The extracted keyphrases were then compared against a gold standard which are the author assigned keyphrases associated with the test documents. The testing has been carried out for the rest of the folds and the performance measures described in section 5.3.3 were then averaged.

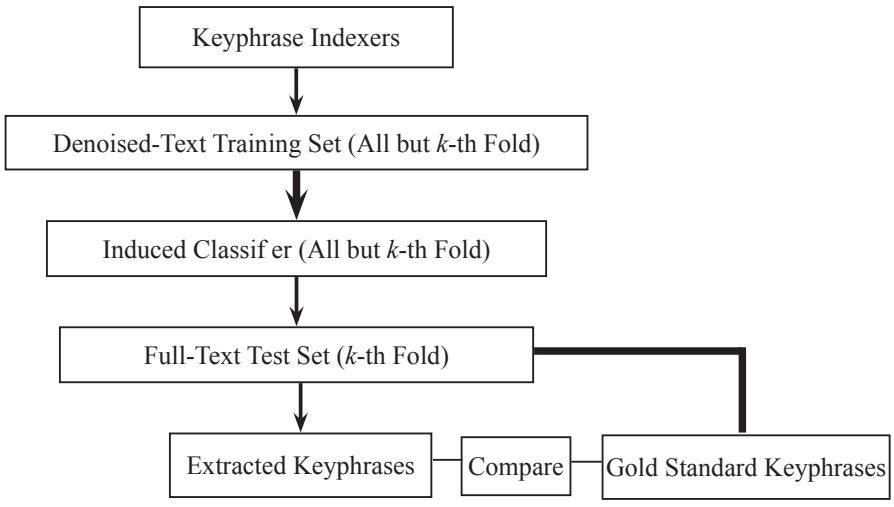


Figure 5.1: Keyphrase extraction from *k*-th fold

5.3.3 Performance Measures

In this experiment, we used the conventional quantitative measures for performance evaluation—precision, recall and F-score. In addition, we used three inter-indexing agreement measures popularly used for indexers' quality assessment [10]. The measures are called Hooper's (H), Rolling's (R) and Cosine (C) indexing agreements. The common property of these measures is that they measure the number of common keyphrases in relation to the size of the two sets of keyphrases being compared. We briefly summarize these agreement measures for the reader's convenience. If M and N are the number of idiosyncratic keyphrases assigned by two indexers and O is the number of phrases two indexers have in common, then Hooper's measure [11] is

$$H(indexer_1, indexer_2) = \frac{O}{M + N - O}.$$

Similarly, Rolling's measure [12] is defined as

$$R(indexer_1, indexer_2) = \frac{2{\cdot}O}{M + N}.$$

Cosine measure uses the geometric mean instead of Rolling's arithmetic mean and can be written as

$$C(indexer_1, indexer_2) = \frac{O}{\sqrt{M{\cdot}N}}.$$

The last two measures are almost identical unless the sets radically vary. It can be noted that Hooper's and Rolling's measures are identical to Jaccard's and the Dice coefficient, respectively, which are used to measure similarities between two sets. The closer the agreement measures are to 1, the more the indexers agree on extracted keyphrases.

We also calculated the error rates for every cross validation. The error rate is defined as

$$E = \frac{FP + FN}{N},$$

where FP and FN are the number of false positives and false negatives, respectively and N is the total number of test instances. The reasons for using the error rates are twofold: (i) to find a text denoising threshold from the classifiers' learning curves described in Section 5.3.4; and (ii) to measure a *10-fold cross validated paired t-test* [13] to report statistical significance of the results. For any two given sets of results, we considered their error rates to calculate a paired t-value. If this calculated t-value lies outside ±2.26 with a degree of freedom 9, then the difference of these two sets is said to be statistically significant at significance level $\alpha = 0.05$.

5.3.4 Text Denoising Threshold from Learning Curves

To find appropriate text denoising thresholds for keyphrase indexing, we plotted the learning curves of the keyphrase classifiers on each dataset by increasing the text denoising threshold in increments of 10% from 30% to 90%. Because both KEA and KEA++ induce Naïve Bayes classifiers, the best-fit classifier should be where the test error has its global minimum. Therefore, the objective is to discover the global minimum from the learning curves. Eventually, this global minimum is the denoising threshold for a given dataset.

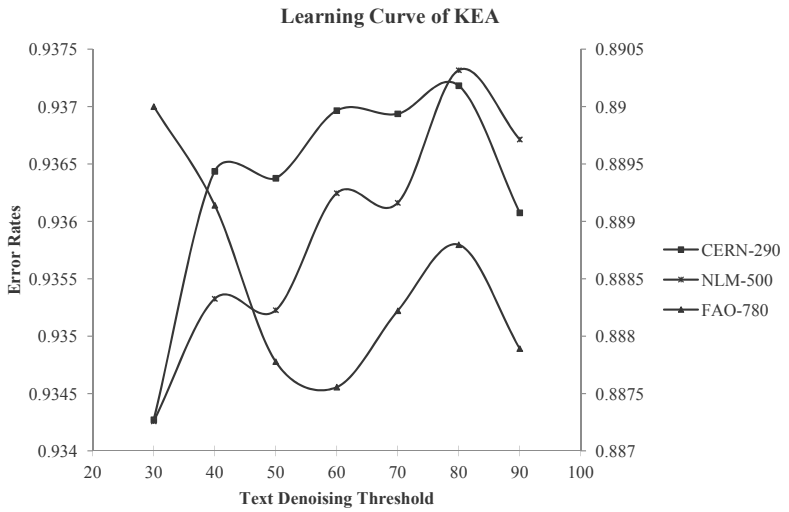


Figure 5.2: Text denoising thresholds of KEA for different datasets

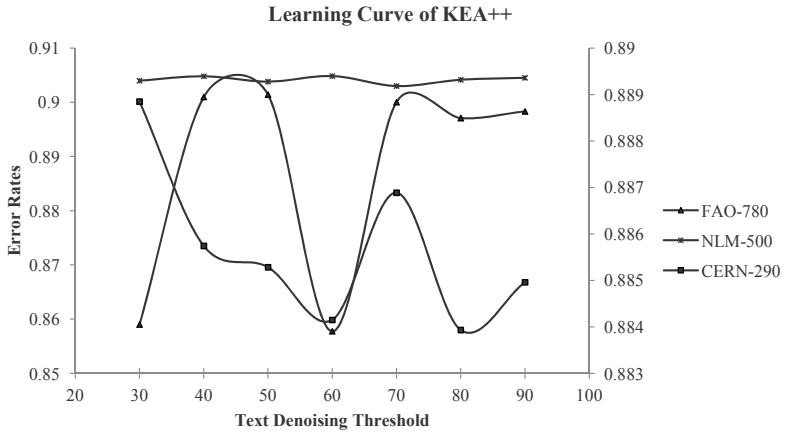


Figure 5.3: Text denoising thresholds of KEA++ for different datasets

For instance, Fig. 5.2 shows the learning curves of KEA on the three datasets for different denoising thresholds. In the case of FAO-780, the classifier of KEA induced at 60% denoising had the global minimum. From this point on, the error rate increased and thus indicated an overfitting in the learning process. However, both of KEA's best performing classifiers on CERN-290 and NLM-500 datasets were induced at 30% denoising.

On the other hand, the best-fit classifiers induced by KEA++ with denoised texts for both FAO-780 and CERN-290 dataset were at the 60% threshold (Fig. 5.3). In contrast, the best performing classifier of KEA++ for NLM-500 was induced at 70% text denoising.

The classifiers induced from different datasets at the aforementioned thresholds were the best performing and hence we applied these classifiers on test sets.

Classifier	Precision	Recall	F-Score	t-value	Hooper	Rolling	Cosine
With Text Denoising	22.17	23.96	23.03	5.07	0.12	0.20	0.21
Benchmark	19.96	21.70	20.76		0.11	0.19	0.20

(a) KEA on FAO-780 dataset with 60% of the texts

Classifier	Precision	Recall	F-Score	t-value	Hooper	Rolling	Cosine
With Text Denoising	14.66	14.80	14.73	3.42	0.07	0.13	0.13
Benchmark	12.26	12.38	12.29		0.06	0.12	0.12

(b) KEA on CERN-290 dataset with 30% of the texts

Classifier	Precision	Recall	F-Score	t-value	Hooper	Rolling	Cosine
With Text Denoising	14.01	15.22	14.60	4.14	0.07	0.12	0.12
Benchmark	11.65	12.85	12.21		0.06	0.11	0.11

(c) KEA on NLM-500 dataset with 30% of the texts

Table 5.1: Performance of KEA with text denoising

Classifier	Precision	Recall	F-Score	t-value	Hooper	Rolling	Cosine
With Text Denoising	26.85	29.21	27.98	3.78	0.15	0.24	0.25
Benchmark	24.08	26.49	25.19		0.14	0.23	0.23

(a) KEA++ on FAO-780 dataset with 60% of the texts

Classifier	Precision	Recall	F-Score	t-value	Hooper	Rolling	Cosine
With Text Denoising	23.24	23.32	23.28	2.40	0.12	0.20	0.21
Benchmark	21.04	21.15	21.04		0.10	0.18	0.18

(b) KEA++ on CERN-290 dataset with 60% of the texts

Classifier	Precision	Recall	F-Score	t-value	Hooper	Rolling	Cosine
With Text Denoising	19.23	21.17	20.15	6.38	0.10	0.17	0.18
Benchmark	17.02	18.92	17.91		0.08	0.15	0.15

(c) KEA++ on NLM-500 dataset with 70% of the texts

Table 5.2: Performance of KEA++ with text denoising

5.4 Results and Discussions

In this section, we discuss the performances of KEA and KEA++ classifiers induced from denoised texts and compare them with their benchmark performances.

Convincingly, on all three datasets KEA's denoised-text induced classifier outperformed its full-text induced classifier. For instance, Table 5.1a shows the precision, recall and F-score of the classifier generated by KEA with and without denoised texts on the FAO-780 dataset. KEA, with its denoised-text induced classifier applied on test sets, achieved an F-score of 23.03 compared to its benchmark F-score of 20.76. The difference of the scores was extremely significant at $\alpha = 0.05$ as we found the t-value 5.07 with a 10-fold cross validated t-test. KEA's convincing performance with denoised-text induced classifiers continued on the CERN-290 dataset as it achieved an F-score of 14.73 while the benchmark F-score is 12.29 (Table 5.1b). The difference, again, was significant with a t-value of 3.42. Finally, we also found KEA, when paired with text denoising, outperforming its benchmark performance on NLM-500 dataset.

For this dataset, KEA's F-score with denoised-text and full-text induced classifiers were 14.60 and 12.21, respectively. The t-value, for this case, was 4.14 showing that the improvement is statistically significant (Table 5.1c).

All of its agreement measures—Hooper, Rolling and Cosine—indicate that KEA's extracted keyphrases with this setup agree more with the gold standard than do the benchmark keyphrases (Table 5.1a–5.1c). Although the difference is small, the values show that KEA's quality keyphrase extraction has not been compromised on any of the datasets, rather it improved on every occasion.

KEA++ performed even better than KEA on the datasets with its reduced-data induced classifiers. For instance, Table 5.2a shows that KEA++, when paired with text denoising, outperformed its benchmark F-score of 25.19 on the FAO-780 dataset. A 3.78 t-value indicates this improvement as statistically significant at $\alpha = 0.05$. Convincing results are found on the CERN-290 dataset where KEA++ with our setup achieved an F-score of 23.28. Compared to that of its benchmark F-score of 21.04, this was significantly better with $t = 2.40$ (Table 5.2b). Finally, also on the NLM-500 dataset, text denoising helped KEA++ to achieve a significantly better performance with a 20.15 F-score compared to its benchmark F-score of 17.91 (Table 5.2c).

Interestingly, extracted keyphrases of KEA++'s denoised-text induced classifier agree more with the gold standard than do the key phrases extracted by its benchmark classifier. For instance, on FAO-780 dataset, its Hooper, Rolling and Cosine scores were 0.15, 0.24, and 0.25 respectively compared to the benchmark scores of 0.14, 0.23, and 0.23 (Table 5.2a). This trend of producing quality keyphrases by KEA++ with our setup can also be seen for the rest of the datasets as well (Table 5.2b and 5.2c).

It can be noted that although we got better results by reducing the training data, it is unlikely to achieve a similar result by reducing the test data. For instance, as we extracted keyphrases, using both the full-text and denoised-text induced classifiers, from denoised texts of the corpora, we did not find any global minima for them (as seen in Fig. 5.2 and Fig. 5.3) and thus found no improvement of the performances over the benchmark.

5.5 Conclusions

In this paper, we describe how reducing the content of the texts collected from different domain to high-quality sentences with text denoising can influence supervised keyphrase indexing. We use established techniques like 10-fold cross validation to measure the performances of two well known keyphrase indexers named KEA and KEA++. Using performance measures like precision, recall and F-score we evaluate their extracted keyphrases against the gold standards provided (author assigned keyphrases). Experimental outcomes show an improved performance with denoised text for extraction training and application to the full text. According to the different indexing agreement measures, the indexers improve on producing quality keyphrases as well.

From our experimental findings we can also conclude that– (i) document size, per se, does not have the suggested effect on keyphrase indexing—it is the content richness that plays the key role in indexing; (ii) text denoising is useful not only for biomedical relation mining but also for keyphrase indexing; and (iii) text denoising can be used for different domains other

than biomedical science.

Recalling the fact of varied denoising thresholds on different datasets and tasks, we are interested in applying machine learning techniques instead of heuristics for text denoising. In addition, we are interested in testing the effect of text denoising on indexers selected from those participating in Task 5 of SemEval-2010 [14]. These tasks are left as future work.

Bibliography

[1] E. Frank, G. Paynter, I. Witten, and C. Gutwin, "Domain-specific keyphrase extraction," in *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI'99)*, (Stockholm, Sweden), pp. 668–673, 1999.

[2] A. R. Aronson, J. G. Mork, C. W. Gay, S. M. Humphrey, and W. J. Rogers, "The NLM Indexing Initiative's Medical Text Indexer," in *Proceedings of the 11th World Congress on Medical Informatics (MEDINFO 2004)*, (San Francisco, USA), pp. 268–272, 2004.

[3] O. Medelyan, *Human-competitive automatic topic indexing*. PhD thesis, University of Waikato, New Zealand, 2009.

[4] O. Medelyan and I. Witten, "Domain-independent automatic keyphrase indexing with small training sets," *Journal of the American Society for Information Science and Technology (JASIST)*, vol. 59, no. 7, pp. 1026–1040, 2008.

[5] P. Turney, "Learning algorithms for keyphrase extraction," *Information Retreival*, vol. 2, pp. 303–336, 2000.

[6] I. Witten, G. Paynter, E. Frank, C. Gutwin, and C. Nevill-Manning, "KEA: Practical automatic keyphrase extraction," in *Proceedings of the 4th ACM Conference on Digital Libraries*, (Berkeley, CA, USA), pp. 254–255, 1999.

[7] R. Shams and R. E. Mercer, "Investigating keyphrase indexing with text denoising," in *Proceedings of the 11th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2012)*, (Washington DC, USA), 2012.

[8] R. Shams and R. E. Mercer, "Extracting connected concepts from biomedical texts using fog index," *Procedia - Social and Behavioral Sciences*, vol. 27, pp. 70–76, 2011.

[9] R. Gunning, "Fog index after twenty years," *Journal of Business Communication*, vol. 6, no. 3, pp. 3–13, 1969.

[10] O. Medelyan and I. Witten, "Measuring inter-indexer consistency using a thesaurus," in *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2006)*, (Chapel Hill, NC, USA), pp. 274–275, 2006.

[11] R. S. Hooper, "Indexer consistency tests: Origin, measurements, results and utilization," report, IBM Corporation, Bethesda, MD, 1965.

[12] L. Rolling, "Indexing consistency, quality and efficiency," *Information Processing and Management*, vol. 17, pp. 69–76, 1981.

[13] E. Alpaydin, "Assessing and comparing classification algorithms," in *Introduction to Machine Learning*, pp. 342–343, Cambridge, UK: The MIT Press, 2004.

[14] S. N. Kim, O. Medelyan, M.-Y. Kan, and T. Baldwin, "Semeval-2010 Task 5: Automatic keyphrase extraction from scientific articles," in *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval 2010)*, (Uppsala, Sweden), 2010.

Chapter 6

Extracting the Information-rich Part of Text using Text Denoising

This chapter is based on the paper titled "Extracting Information-rich Part of Texts using Text Denoising" that appeared in the 26th Canadian Conference on Artificial Intelligence (CAI-2013). It briefly summarizes the concepts presented in Chapter 2 to 5 as well as introduces the machine learning approach we consider in the following chapters. However, the work accomplished in this chapter was later investigated in more detail in Chapter 10. Note that we used the same three datasets (BioNLP [1], BioDRB [2], and FetchProt [3]) in both this chapter and in Chapter 10, and the results reported in this chapter are better. However, the work described in this chapter was presented at a Graduate Symposium and exhibited only a small improvement. A much more sophisticated machine-learning approach is presented in Chapter 10.

In this paper, we continue to report on a novel text reduction technique, called *Text Denoising*, that highlights information-rich content when processing a large volume of text data, especially from the biomedical domain. The core feature of the technique, the text readability index, embodies the hypothesis that complex text is more information-rich than the rest. When applied on tasks like biomedical relation bearing text extraction, keyphrase indexing and extracting sentences describing protein interactions, it is evident that the reduced set of text produced by text denoising is more information-rich than the rest.

6.1 Introduction

Often, to test a method's scalability as well as its performance across genres of texts, there is a need to process large volumes of text data in many disciplines of NLP, be it textual relation extraction, summarization or meta-tagging. It has been reported by many researchers [4] [5] that machine learning as well as rule-based approaches show improvements over their benchmarks with increased training data. However, the use of large volume of data can create several bottlenecks. One is technical—processing large data, like that from biomedical texts, slows down many algorithms; another is even more important—algorithms can exhibit a decreased accuracy because of the noise, which are irrelevant or redundant data for a given classification task, added by information-poor parts of texts.

There are several statistics, like word-level feature *tf–idf* and sentence-level feature *sentence position*, that help identify information richness. Although the degree of use shows their popularity, these features have some serious limitations. For example, *tf–idf* computes document similarity directly in the word-count space which may be slow for large vocabularies and sentence position is useful for summarization but is superficial in relation extraction. In other words, they are either task-specific and/or domain-specific measures.

Text readability has multivariate features that consider many attributes like length of paragraph, words and sentences, and number of polysyllabic and monosyllabic words. In this paper, I report a text reduction technique called *Text Denoising* that reduces text data based on text readability, especially from the biomedical domain, to that which is more information-rich by removing most of the noise. The reduced text is also expected to be task-independent and informative enough to improve accuracy of NLP tools across disciplines.

6.2 Proposed Method

Among text readability scores, the following five measures are considered as yardsticks— Fog Index (hereinafter, FI) [6], Flesch reading ease score (FRES) [7], Smog Index [8], Forcast Index [9], and Flesch-Kincaid readability index (FKRI) [10]. The choice of using text readability as an *information richness statistic* is motivated by the results of an experiment by Duff and Kabance [11]. In their experiment, a passage with no more than two phrases were converted into primer prose and FI was applied to test its readability. They found that the score was low (i.e., the prose was extremely easy to read). The authors concluded that easy texts obscure the relationships and ideas as they de-emphasize both. In contrast, difficult texts emphasize relationships and ideas yielding low readability. I suggest that the describing of biomedical relations, meta information, etc. lengthens sentences as well as increases the use of polysyllabic words which are the two principal components of many of the readability indexes.

Both rule-based and machine learning-based versions of *Text Denoising* are based on this principle that use text readability as a key feature and applies it at the sentence-level to identify those sentences within a text, called denoised text, where content information, such as biomedical relations, is more likely to occur. The rest of the text is called noise text. I am interested to observe the effect of using text denoising on different tasks and genres of text.

6.3 Text Denoising on Relation Extraction

I developed a corpus of 24 texts that describe four pairs of related MeSH C and MeSH D concepts reported by Perez *et al.* [12]. I applied the rule-based version of text denoising on these texts to extract related biomedical concepts. The only rule I set for this task was to extract 30% of the low-readability sentences from the texts according to their FI score. This threshold is termed as the *denoising threshold* and the texts extracted are called *denoised texts*; the rest is called *noise text*. This threshold point was set heurisitically considering the stability in the frequency of appearance of the related concepts in the corpus. Other than 30%, the results with different denoising thresholds ranging from 10% to 50%, however, was not satisfactory. I ranked the pairs of concepts present in the denoised texts using their frequency. Most of the

Rank	Related Concepts	Semantic Relation
1	Ischemia-Glutamate	Yes
2	Levels-Ischemia	No
3	Levels-Glutamate	Yes
4	Glutamate-Neurons	Yes
5	10min-Ischemia	Yes
6	Glutamate-CA4	Yes
7	Increase-Glutamate	Yes
8	10min-Glutamate	No
9	Ischemia-5min	Yes
9	Glutamate-5min	No

Table 6.1: Extracted related concepts for a paper on Ischemia and Glutamate

concept pairs with higher ranks, however, did not contain any semantic relations according to UMLS semantic relation network. Therefore, I re-ranked the pairs according to their positive predictive value (PPV) (similar to *precision* measure used in information retrieval evaluation tasks) and sensitivity. The pairs of concepts found from this re-ranking showed a convincing accuracy of 75% (ratio of semantically related concepts to total) against the output of the UMLS semantic relation network. Table 6.1 shows an output from a paper on one of the four pairs of concepts. Of note, I found that the noise texts did not have any related biomedical concepts. The detailled experimental setup and results are reported by Shams and Mercer [13].

Later, I performed an experiment with four other readability scores mentioned in Section 6.2 on the same corpus. A comparative result showed that FI outperformed the other indexes by extracting more meaningful relations [14]. I also analyzed the performance of the indexes considering the performance of FI as a benchmark. Table 6.2a and 6.2b show that the SMOG index is a close second to FI followed by FKRI, while FRES and FORCAST performed poorly. It can also be noted that the SMOG index, like FI, uses the core measure of *complex words* which reveals the fact that the measure of complex word fits best for text denoising and biomedical relation extraction.

Score	Precision	Recall	F-Score
SMOG	95.83	82.14	88.46
FKRI	88.89	82.76	85.71
FRES	82.61	65.52	73.08
FORCAST	81.82	62.07	70.59

(a)

Score	Precision	Recall	F-Score
SMOG	96.88	82.60	89.16
FKRI	89.73	82.60	86.01
FRES	80.83	65.63	72.44
FORCAST	77.88	61.61	68.72

(b)

Table 6.2: (a) Micro-average and (b) macro-average precision, recall and F-Score of the indexes on biomedical relation extraction

Classifier	FAO-780		CERN-290		NLM-500	
	F-Score	t-value	F-Score	t-value	F-Score	t-value
with Text Denoising	23.03	5.07	14.73	3.42	14.60	4.14
Benchmark	20.76		12.29		12.21	

(a) Performance of KEA

Classifier	FAO-780		CERN-290		NLM-500	
	F-Score	t-value	F-Score	t-value	F-Score	t-value
with Text Denoising	27.98	3.78	23.28	2.40	20.15	6.38
Benchmark	25.19		21.04		17.91	

(b) Performance of KEA++

Classifier	FAO-780		CERN-290		NLM-500	
	F-Score	t-value	F-Score	t-value	F-Score	t-value
with Text Denoising	31.87	2.76	24.42	2.26	31.50	3.52
Benchmark	31.86		24.92		31.13	

(c) Performance of Maui

Table 6.3: F-Scores of the keyphrase indexers with text denoising and its benchmark on three datasets

6.4 Text Denoising on Keyphrase Extraction

I investigated the usability of denoised texts as training data for machine learning-based keyphrase indexers called KEA [5], KEA++ [4] and Maui [15]. I applied the indexers with their classifiers induced from denoised training data on three datasets, namely FAO-780, CERN-290 and NLM-500. These datasets are composed of texts from the domains of agriculture, physics and biomedical science. I compared the result with their benchmark performances that were achieved by using the full-text training data. Convincingly, in a 10-fold cross validation experiment, both KEA and KEA++, with their classifiers induced from denoised training data, outperformed their respective benchmark F-scores [16]. Maui, on the other hand, had mixed results and its denoised text induced classifier performs comparably with its benchmark [17]. The F-Scores are listed in Table 6.3a, 6.3b and 6.3c where a *t-value* greater than or equal to 2.26 indicates the statistical significance of the results at 95% confidence. Of note, unlike the fixed denoising threshold of 30% for relation extraction, I found that to get bias-free classifiers for the indexers, the denoising threshold point needed to be varied (usually between 30%–70%) for different genres of texts. This outcome confirms that the rule to decide the amount of text to be extracted from texts substantially depends for different writing styles.

Dataset	Precision (%)	Recall (%)	F-Score (%)
BioNLP	82.5	87.8	85.1
BioDRB	84.7	91.1	87.8
FetchProt	90.8	89.2	90.0

Table 6.4: Performance of text denoising on extracting protein relations bearing sentences against the gold standard

6.5 Text Denoising on Extracting Protein Relations bearing Sentences

In an attempt to eliminate the denoising threshold which depends on writing style (Section 6.4), I decided to develop a machine learning version of text denoising. The classification task in hand was to annotate sentences of a set of texts with either *positive* or *negative* labels based on the presence of protein interactions. The feature set chosen is composed of 35 features like various parameters of readability indexes, term frequency, inverse sentence frequency, biomedical named entity, verbs and acronyms, stopwords, semantic words, and sentence positions. After applying a series of well known classifiers like Bayesian classifiers, Random Forest, SVM, AdaBoost, and Bagging, the classifier that performed best was chosen which is Bagging stacked with Random Forest. The corpora used for this experiment are BioNLP, BioDRB and FetchProt that contain over $85,000$ sentences. Two automated tools called RelEx [18] and WRelEx [19] are used to assign binary labels to each sentence of these corpora depending on the presence of any protein relations. Having realized after this assignment that the classes are negatively skewed (almost doubled the positive labels), synthetic positive samples are produced using SMOTE [20] where the minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k, which is five in our setup, minority class nearest neighbors. From initial results, I found that many features were highly correlated with each other but had low correlation with the class. Therefore, I used a *wrapper* method to select a set of bias-free features. However, I observed that this set of features varies for different corpora. Table 6.4 shows the precision, recall and F-Score of text denoising in a 10-fold cross validation setup, considering the highly agreed upon annotation of RelEx and WRelEx as the gold standard. It can be noted that the outcome of this experiment without using SMOTE was not satisfactory as the F-scores were under 80%.

6.6 Conclusion and Future Work

The proposed text denoising method performed much the same on several tasks and kinds of texts: the reduction of texts according to the readability improved relation mining, keyphrase indexing and extracting sentences that describe protein relations. This result strongly suggests that sentences that are difficult to read are more information-rich than the rest. The effect of text denoising is yet to be examined for text categorization and summarization. I am currently investigating the effect of readability on e-mail spam detection. The results so far are interesting as I am labelling spam and ham based on the readability of e-mail text content only (i.e., without

looking at the mail header). Also, I intend to train benchmark summarizers with denoised texts and see how they perform against gold standard summaries.

Bibliography

[1] J.-D. Kim, S. Pyysalo, T. Ohta, R. Bossy, N. Nguyen, and J. Tsujii, "Overview of bionlp shared task 2011," in *Proceedings of the BioNLP Shared Task 2011 Workshop*, (Portland, Oregon, USA), pp. 1–6, Association for Computational Linguistics, June 2011.

[2] R. Prasad, S. McRoy, N. Frid, A. Joshi, and H. Yu, "The biomedical discourse relation bank," *BMC Bioinformatics*, vol. 12, no. 1, pp. 188+, 2011.

[3] K. Franzën and D. Oppenheimer, "The fetchprot corpus: Documentation and annotation guidelines," tech. rep., Swidish Institute Of Computer Science Report, Sweden, 2007.

[4] O. Medelyan and I. Witten, "Domain-independent automatic keyphrase indexing with small training sets," *Journal of the American Society for Information Science and Technology (JASIST)*, vol. 59, no. 7, pp. 1026–1040, 2008.

[5] I. Witten, G. Paynter, E. Frank, C. Gutwin, and C. Nevill-Manning, "KEA: Practical automatic keyphrase extraction," in *Proceedings of the 4th ACM Conference on Digital Libraries*, (Berkeley, CA, USA), pp. 254–255, 1999.

[6] R. Gunning, "Fog index after twenty years," *Journal of Business Communication*, vol. 6, no. 3, pp. 3–13, 1969.

[7] R. Flesch, "A new readability yardstick," *Journal of Applied Psychology*, vol. 32, pp. 221–33, 1948.

[8] G. H. McLaughlin, "Smog grading – a new redability formula," *Journal of Reading*, vol. 12, no. 8, pp. 639–46, 1969.

[9] J. S. Caylor, T. G. Stitch, L. C. Fox, and J. P. Ford, "Methodologies for determining reading requirements of military occupational specialities," Tech. Rep. 73-5, Human Resources Research Organization, Alexandria, VA, 1973.

[10] J. P. Kincaid, R. P. Fishburne, R. L. Rogers, and B. S. Chissom, "Derivation of new readability formulas (automated readability index, fog count, and flesch reading ease formula) for navy enlisted personnel," Research Branch Report 8-75, Chief of Naval Technical Writing: Naval Air Station Memphis, 1975.

[11] T. M. Duffy and P. Kabance, "Testing a readable writing approach to text revision," *Journal of Educational Psychology*, vol. 74, pp. 733–48, 1982.

[12] C. Perez-Iratxeta, P. Bork, and M. Andrade, "Literature and genome data mining for prioritizing disease-associated genes," in *Discovering Biomolecular Mechanisms with Computational Biology* (F. Eisenhaber, ed.), Molecular Biology Intelligence Unit, pp. 74–81, Springer, 2006.

[13] R. Shams and R. E. Mercer, "Extracting connected concepts from biomedical texts using fog index," *Elsevier Procedia - Social and Behavioral Sciences*, vol. 27, pp. 70–76, 2011.

[14] R. Shams and R. E. Mercer, "Evaluating core measures of text denoising for biomedical relation mining," in *3rd International Workshop on Global Collaboration of Information Schools (WIS 2012)*, (Taipei, Taiwan), 2012.

[15] O. Medelyan, *Human-competitive automatic topic indexing*. PhD thesis, University of Waikato, New Zealand, 2009.

[16] R. Shams and R. E. Mercer, "Improving supervised keyphrase indexer classification of keyphrases with text denoising," *LNCS– The Outreach of Digital Libraries: A Globalized Resource Network*, vol. 27, pp. 77–86, 2012.

[17] R. Shams and R. E. Mercer, "Investigating keyphrase indexing with text denoising," in *12th ACM/IEEE Joint Conference on Digital Libraries (JCDL 2012)*, pp. 263–266, ACM, 2012.

[18] K. Fundel, R. Küffner, and R. Zimmer, "Relex - relation extraction using dependency parse trees," *BMC Bioinformatics*, vol. 23, no. 3, pp. 365–371, 2007.

[19] S. I. Faiz, "Discovering higher order relations from biomedical text," Master's thesis, Department of Computer Science, The University of Western Ontario, Canada, 2012.

[20] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

Chapter 7

Classifying Spam Emails using Text and Readability Features

This chapter is based on the paper titled "Classifying Spam Emails using Text and Readability Features" co-authored with Robert E. Mercer that appeared in the 13th IEEE International Conference on Data Mining (ICDM2013).

In this paper, we report a novel spam classification method that uses features based on email content-language and readability combined with the previously used content-based task features. The features are extracted from four benchmark datasets viz. CSDMC2010, SpamAssassin, LingSpam, and Enron-Spam. We use five well-known algorithms to induce our spam classifiers: Random Forest (RF), BAGGING, ADABOOSTM1, Support Vector Machine (SVM), and Naïve Bayes (NB). We evaluate the classifier performances and find that BAGGING performs the best. Moreover, its performance surpasses that of a number of state-of-the-art methods proposed in previous studies. Although applied only to English language emails, the results indicate that our method may be an excellent means to classify spam emails in other languages, as well.

7.1 Introduction

Since the publishing of *Spam!* in 1998 [1], the menacing onslaught of spam emails has grown exponentially. According to a recent survey, the number of spam emails sent out in March 2013 is about 100 billion [2]. This phenomenal quantity is 98% more than that from the end of the previous quarter. Among the drastic effects of spam emails are loss of individual productivity and financial loss of organizations. Anti-spammers, therefore, are putting forward efforts to prevent this potential threat to today's Internet.

Many studies agree that spam emails have attributable patterns [3] [4]. Recognizing this, spammers are constantly introducing new techniques to obfuscate these recognizable patterns to fend off supervised anti-spam filters. Therefore, spam filtering is, and is likely to remain, an interesting machine learning application. Typically, an email exhibits two types of features: (i) header features and (ii) content-language features. Using either set of features to detect spam email has its pros and cons. For instance, Zhang *et al.* [5] empirically showed that the performances of a handful of machine learning algorithms are not satisfactory with text

features. Lai and Tsai [6] found similar results, too—according to their study, header features have a lower *total cost ratio* than text features. Another typical argument to support the use of header features is that they are language independent (see, for example, [7] and [8]). However, Metsis *et al.* [9] exploited a language independent, content-based feature called *term frequency* (TF). Using this feature with different variants of the Naïve Bayes algorithm, they found results that are better than those found by a number of header-based methods. Likewise, studies are continuously reporting improved results with email-text features (see Prabhakar and Basavaraju [10], and Ma *et al.* [11])—most of which perform better than header features.

The proposed method utilizes text features that are long-established such as frequency of *spam words* and HTML tags as well as some that are new. The novelty of our work is that we introduce language-centric features such as grammar and spell errors, use of function words, presence of verbs and alpha-numerics, TF·IDF, and inverse sentence frequency. In addition, we use features related to message readability (e.g., reading difficulty indexes, complex and simple word frequency, document length, and word length). The features are extracted from four standard email datasets—CSDMC2010, SpamAssassin, LingSpam, and Enron-Spam. The features are then used by five well-known learning algorithms—Random Forest (RF), BAGGING, ADABOOSTM1, Support Vector Machine (SVM), and Naïve Bayes (NB)—to induce binary classifiers. From our extensive experiment, we find that the addition of text and readability features significantly improves classifier performance compared to that found with the commonplace features alone. Another notable finding is that the classifier induced by BAGGING performs the best on all of the datasets. In addition, its performance surpasses that of a number of state-of-the-art methods proposed in previous studies. Like Metsis *et al.* [9], our features are language independent. Whereas we derive the results solely from English emails, the proposed approach may be an excellent means to classify spam emails in any language.

The next section details the features used in this research. Following that, we describe the learning algorithms, performance evaluation measures, and datasets. In Section 7.6, we report the experimental findings, and in Section 7.7 we provide some discussion of these results, especially as compared with previous filtering methods. Finally, Section 7.8 concludes the paper.

Groups	**Features**			
Traditional Features	Spam Words	HTML Anchors	HTML Non-anchors	HTML Tags
Text Features	Alpha-numeric Words	Verbs	Function Words	TF·ISF
	TF·IDF	Grammar Errors	Spelling Errors	Language Errors
Readability Features	FI	FRES	SMOG	FORCAST
	FKRI	Simple Word FI	Inverse FI	Complex Words
	Simple Words	Document Length	Word Length	$\text{TF·IDF}_{complex}$
	TF·IDF_{simple}			

Table 7.1: The list of features used to classify spams in our experiment.

7.2 Feature Selection

Each email, in our experiment, is represented as $(\vec{x}, y)$, where $\vec{x} \in \mathbb{R}^n$ is a vector of n attributes and $y \in \{spam, ham\}$ is the label of the email. In our study, we explored 40 attributes to classify emails and therefore, $n = 40$. We have grouped these features into three subgroups: (i) traditional features, (ii) text features, and (iii) readability features. The list of features used in our experiment is shown in Table 7.1. The notation used in this section is "X_i : *name* = *feature value*", where X_i is used to label the graphs later in the paper, *name* is a descriptive name, and *feature value* indicates the feature value calculation. Some features are also normalized. These normalizations are discussed where warranted.

7.2.1 Traditional Features

Below are descriptions of the four typical features for spam classification that are used in our experiment.

Dictionary-based Features

The first feature in this group is the frequency of spam words. The selection of this feature is inspired by the interesting findings of Graham [12]. He showed that merely looking for the word *click* in the messages can detect 79.7% of spam emails in a dataset with only 1.2% false positives. To exploit this feature, we have developed a dictionary comprising 381 spam words[1] accumulated from various anti-spamming blogs. We treated each message as a bag-of-words and counted the frequency of spam words in them as follows:

X_1 : *Spam Words* = #*Spam Words*.

HTML Features

The frequency of HTML tags in emails is a common means to classify spams. We have subdivided this feature into three features: (i) frequency of anchor tags (i.e., number of close-ended tags <a> and), (ii) frequency of tags that are not anchors, *anchor′* (e.g., <p> or
), and (iii) total HTML tags in the emails (e.g., sum of (i) and (ii)). To identify HTML tags in the emails, we used a Java HTML parser called *jsoup*[2]. Each feature is normalized by the length of the email, N, which is the number of sentences in the message. The detailed HTML feature calculations are:

X_2 : $anchor = \frac{\#anchor\ tags}{N}$,
X_3 : $anchor' = \frac{\#anchor\ tags'}{N}$, and
X_4 : $total\ \text{HTML} = \frac{\#anchor\ tags + \#anchor\ tags'}{N}$.

7.2.2 Text Features

These novel (except for *Verbs*) features focus on various aspects of the email text. These features into divided into two subgroups: (i) Word-level Features and (ii) Error Features.

[1] http://cogenglab.csd.uwo.ca/sentinel/spam-term-list.html
[2] http://jsoup.org/download

Word-level Features

First, we considered the frequency of alpha-numeric words in the emails. We found reasonable evidence to select this feature during the development of the spam word dictionary (See Section 7.2.1)—many of the dictionary entries are alpha-numerics. The apparent reasons for this include that spammers often advertise menacing websites by replacing literals of the legitimate website with numerics and vice versa. This text feature is calculated as follows:

X_5 : *Alpha-numeric Words* = #*Alpha-numeric Words*.

In their study, Orasan and Krishnamurthy [13] showed that a *verb* Part-of-Speech (POS) feature can be a strong identifier of junk emails; nonetheless, very few works have followed up. We are, however, interested in this particular POS feature. We used the Stanford POS Tagger[3] to tag the POS of each word in the emails and considered the frequency of verbs in each:

X_6 : *Verbs* = #*Verbs*.

Third, we used the frequency of function words[4] in the emails as a feature. As the definition of function words varies from task to task, we define them as follows: function words are very frequent non-content-bearing words such as articles, prepositions, adverbs, etc. We developed a *stoplist function* to count the frequency of function words in the emails:

X_7 : *Function Words* = #*Function Words*.

Fourth, we included another novel feature: TF·ISF, a simple summation of the product of TF and ISF, where the prior is the frequency of a term t in a message and the latter is its inverse sentence frequency. TF of term t can be calculated as follows:

$\text{TF}_t = 1 + log(frequency_t)$, if $frequency_t > 0$, 0 otherwise,

while its ISF is:

$\text{ISF}_t = log\ \frac{N}{\text{SF}_t}$,

where N is the message length and SF_t is the number of sentences with term t. Inverse sentence frequency is a relative measure of whether the term is common or rare in a single message. The TF·ISF of a message is calculated as follows:

X_8 : $\text{TF·ISF} = \sum_t \text{TF}_t \times \text{ISF}_t$, *for all t in the message*.

Our next feature in this group is TF·IDF which is similar to TF·ISF except that the inverse document frequency, IDF, measures whether a given term t is common or rare in an entire dataset. The IDF of a term t in the message can be found as follows:

$\text{IDF}_t = log\ \frac{D}{\text{DF}_t}$,

where D is the total number of messages in the dataset and DF_t is the number of messages containing t. The TF·IDF feature is then calculated as follows:

X_9 : $\text{TF·IDF} = \sum_t \text{TF}_t \times \text{IDF}_t$.

The feature is then normalized by taking its square root.

Error Features

The next three features are concerned with the grammar and spelling errors present in the message. For each message, we counted the frequency of grammar and spelling errors using

[3] http://nlp.stanford.edu/software/tagger.shtml

[4] http://cogenglab.csd.uwo.ca/sentinel/function-word-list.html

a Java API called LanguageTool[5]. By summing up these two errors, we introduced another feature named *Language Errors*:

X_{10} : *Grammar Errors* = $\#N_{ge}$,

X_{11} : *Spelling Errors* = $\#N_{se}$, and

X_{12} : *Language Errors* = $\#N_{te}$,

where N_{ge} is the number of sentences with grammar errors, N_{se} is the number of sentences with spelling errors and N_{te} denotes the total number of sentences with language-based errors (i.e., $N_{te} = N_{ge} + N_{se}$). This group of features, to the best of our knowledge, is novel for spam classification. It is to be noted that for training emails, this feature has been normalized respectively for hams and spams; for the testing emails for which the class label is unknown, traditional attribute normalization is performed. Surprisingly, we get unsatisfactory results if we use traditional attribute normalization for the training emails.

7.2.3 Readability Features

The second group of novel features that we use are readability-based features. Readability deals with the difficulty of reading a sentence, a paragraph, or a document. Two important parameters to calculate readability are simple and complex words. Simple words are those that have at most two syllables while complex words contain three or more syllables. Based on these two factors and others, five scores, among many, are used as yardsticks to assess the readability of text. In this section, we divided readability features further into two subgroups: (i) Score-based Features and (ii) Frequency-based Features.

Score-based Features

We measured the readability of each message in the datasets using the five aforementioned scoring methods. First, we used the *Fog Index (FI)* [14], which is the most popular score to measure readability. The scores of each message can be found as follows:

X_{13} : $FI = 0.4 \times \left(\left(\frac{\sum Words}{N}\right) + 100 \times \left(\frac{\sum Complex\ Words}{\sum Words}\right)\right)$.

Second, we used the *Flesch Reading Ease Score (FRES)* [15], one of the oldest readability scores. The *FRES* of any given message can be found as follows:

X_{14} : $FRES = 206.835 - 1.015 \times \left(\frac{\sum Words}{N}\right) - 84.6 \times \left(\frac{\sum Syllables}{\sum Words}\right)$.

The *SMOG index*, when first published, was anticipated as a proper substitute for *FI* due to its accuracy and ease of use [16]. It is the third readability score used as a feature:

X_{15} : $SMOG\ Index = 1.043 \times \sqrt{30 \times \frac{\sum Complex\ Words}{N}} + 3.1219$.

The *FORCAST index* was originally formulated to assess the reading skills required by different military jobs without focusing on running narratives [17]. The index, unlike others, emphasizes the frequency of simple words. The following is the way to calculate the FORCAST index of a message:

X_{16} : $FORCAST\ Index = 20 - \frac{W}{10}$,

where W is the number of simple words in a 150-word sample of the text.

[5] `http://www.languagetool.org/java-api/`

A second instalment of a readability index proposed by Flesch and further investigated and modified by Kincaid [18] is known as the *Flesch-Kincaid Readability Index (FKRI)*. This feature can be calculated as follows:

X_{17} : $FKRI = 0.39 \times \left(\frac{\sum Words}{N}\right) + 11.8 \times \left(\frac{\sum Syllables}{\sum Words}\right) - 15.59$.

In addition, we have modified *FI* in two different ways to give two additional features. We chose to modify *FI* only, because empirical outcomes showed that these modifications of *FI* provide better results than those of other scores.

First, in X_{13}, in the place of complex words, we substituted the frequency of simple words:

X_{18} : $FI_{simple} = 0.4 \times \left(\left(\frac{\sum Words}{N}\right) + 100 \times \left(\frac{\sum Simple\ Words}{\sum Words}\right)\right)$.

Second, we took the arithmetic inverse of *FI* of the messages to get the feature *Inverse FI*:

X_{19} : $Inverse\ FI = \frac{1}{FI}$.

Frequency-based Features

We also considered the frequency of *Complex Words* and *Simple Words* in the messages as two more features:

X_{20} : *Complex Words* = #*Complex Words*, and

X_{21} : *Simple Words* = #*Simple Words*

Document Length, which has already been used in other features for normalizing, is also considered as a feature in our research. It simply denotes the number of sentences in a message:

X_{22} : *Document Length* = #*Sentences*.

Word Length of a message is simply the average number of syllables per word. This feature is calculated as follows:

X_{23} : $Word\ Length = \frac{\#Syllables}{\#Words}$.

Our last two features are the combination of TF·IDF, and frequency of simple and complex words. While X_9 deals with the TF·IDF of any term t, our second last feature $(\text{TF·IDF})_{complex}$ deals with that of complex words. Finally, our last feature $(\text{TF·IDF})_{simple}$ considers the TF·IDF of simple words. The formula to calculate these feature are given below:

X_{24} : $(\text{TF·IDF})_{complex} = \sum \text{TF}_{complex} \times \text{IDF}_{complex}$, and

X_{25} : $(\text{TF·IDF})_{simple} = \sum \text{TF}_{simple} \times \text{IDF}_{simple}$.

As well, we used 14 more features that are calculated by excluding stopwords from the features X_1, X_5, X_6, X_8, $X_{13} - X_{21}$, and X_{23}. In the graphs these features have labels ending with a dot. It is noteworthy that the features X_2, X_3, and X_4 could not be extracted from the LingSpam and Enron-Spam datasets as the related information was removed from the messages by the dataset curators (refer to Table 7.4).

Since we analyzed the feature vectors for all the messages in the datasets, we have found that the distribution of the most of the feature values is not normal rather they exhibit either positive or negative skewness. The effect of this skewness was eliminated by using a logarithmic transformation of all the feature values. Once log-transformed, the distribution becomes normal. The newly found log values of any given feature are then normalized by the highest transformed value of that feature in the dataset; the resulting values are therefore in [0,1].

Learning Algorithms	Parameters	
Random Forest (RF)	Maximum Depth: Unlimited	Number of Trees to be Generated: 10
	Random Seed: 1	
ADABOOSTM1	Number of Iterations: 10	Random Seed: 1
	Resampling: False	Weight Threshold: 100
BAGGING	Size of Bag (%): 100	Out of Bag Error: False
	Number of Iterations: 10	Random Seed: 1
Support Vector Machine (SVM)	SVM Type: C-SVC	Cost: 1.0
	Degree of Kernel: 3	EPS: 0.0010
	Gamma: 0.0	Kernel Type: Radial Basis
	Epsilon: 0.1	Probability Estimates: False
	Shrinking Heuristics: True	
Naïve Bayes (NB)	Use of Kernel Estimator: False	

Table 7.2: Parameters of the learning algorithms used in this experiment.

7.3 Learning Algorithms

We used the well-known learning algorithms Random Forest RF, Naïve Bayes (NB), Support Vector Machine (SVM), and two *meta-learning* algorithms viz. ADABOOSTM1 and BAGGING to induce binary classifiers. What follow are the reasons for choosing the algorithms in our experiment.

Among the learning algorithms, we have chosen Random Forest (RF) for three reasons. First, it has been used by a number of anti-spam filters because of its high spam-classification accuracy (see, for instance, [7] and [19]). Second, the algorithm runs efficiently on large data. Third and most importantly, the learning is fast—which is desirable for any *live* spam filter.

Our second algorithm, also an ensemble learning method, is called ADABOOSTM1. Although an ensemble method, ADABOOSTM1 is both simple and fast. The biggest advantage of this ensemble method is that it is less susceptible to training-data overfit. Last but not least, the algorithm has been reported to perform better than Naïve Bayes (NB) and Probabilistic TF·IDF for text categorization tasks [20]. The reasons to use *bagging* are primarily three. First, like ADABOOSTM1, BAGGING is simple and fast. Note that the speed of learning of this algorithm depends on the choice of the number of random samples of training data. Second, it is less susceptible to overfitting. Finally, it leads to improvements for unstable classification algorithms like trees [21]. In this experiment, RF has been chosen as the base algorithm for ADABOOSTM1 and BAGGING. Therefore, we name the classifiers generated by these two algorithms BOOSTED RF and BAGGED RF, respectively.

Support Vector Machine (SVM) is also a popular learning algorithm for spam detection. However, from the work of Zhang *et al.* [5], Lai and Tsai [6], Hu *et al.* [7], Qaroush *et al.* [19], and Ye *et al.* [22], it is evident that the performance of SVM is better with header features than with text features.

Like SVM, Naïve Bayes (NB) is a widely-used learning algorithm in the anti-spamming community. Benchmark anti-spam tools developed by Lai and Tsai [6], Hu *et al.* [7], Metsis *et*

		Actual	
		Spam	Ham
Prediction	Spam	$n_{s\to s}$	$n_{h\to s}$
	Ham	$n_{s\to h}$	$n_{h\to h}$

Table 7.3: Confusion matrix for spam classification problem.

al. [9], and Qaroush *et al.* [19] use NB to generate classifiers because the algorithm is simple yet powerful enough to detect spams effectively. For instance, on many occasions, with simple features like TF·IDF, NB even outperformed quality learning algorithms like SVM.

The parameter setup for the learning algorithms used in this experiment is presented in Table 7.2.

7.4 Evaluation Measures

The evaluation of spam classification differs from many other classification tasks. A significant number of previous works rely on performance measures like precision, recall, F-score, and accuracy [7] [8] [10] [11] [19]. However, even a poor classifier can achieve overly-optimistic results on a skewed dataset and *appropriate* performances on such datasets are not reflected by the aforementioned measures. Because of the presence of skewness in the datasets, *cost-sensitive* measures like ham misclassification rate (FPR), spam misclassification rate (FNR) and total cost ratio (TCR) [23] [24] are becoming more popular. In addition, being a balanced measure that combines both FPR and FNR, *area under the ROC curve* (hereinafter, AUC) is also preferred by many [24]. Considering these facts, we choose to report seven evaluation measures—FPR, FNR, accuracy, precision, recall (or simply spam recall), F-score, and AUC. The measures are explained below.

All of the measures reported in this paper depend on the confusion matrix given in Table 7.3. Precision is the fraction of spam predictions that are correct. Should the precision be bigger, the probability of misclassifying a legitimate mail as spam is smaller:

$Precision = \frac{n_{s\to s}}{n_{s\to s}+n_{h\to s}}$.

Spam recall examines the fraction of spam emails being recognized. A bigger spam recall points out that the probability of misclassifying a spam as legitimate mail is smaller:

$Recall = \frac{n_{s\to s}}{n_{s\to s}+n_{s\to h}}$.

F-score, simply, is the harmonic mean of precision and recall:

$F\text{-}score = \frac{2\times Precision\times Recall}{Precision+Recall}$.

Accuracy, on the other hand, is the percentage of correctly identified spams and hams:

$Accuracy = \frac{n_{h\to h}+n_{s\to s}}{n_{h\to h}+n_{h\to s}+n_{s\to h}+n_{s\to s}}$.

FPR denotes the fraction of all legitimate messages classified as spams:

$\text{FPR} = \frac{n_{h\to s}}{n_{h\to s}+n_{h\to h}}$.

In contrast, FNR is the fraction of all spams delivered to the user inbox:

$\text{FNR} = \frac{n_{s\to h}}{n_{s\to h}+n_{s\to s}}$.

Note that the lower the FPR and FNR, the better the performance. Considering the situation where users might accept spams to enter into their inbox but they do prefer their hams not to end up in the spam-traps, a higher FPR is more expensive than a higher FNR. To resolve this, we

Dataset	Total Messages	Spam Rate	Text Pre-processed?	Year of Curation
CSDMC2010	4,327	31.85%	No	2010
SpamAssassin	6,046	31.36%	No	2002
LingSpam	2,893	16.63%	Yes	2000
Enron-1	5,172	29.00%	Yes	2006
Enron-2	5,857	25.54%		
Enron-3	5,512	27.21%		
Enron-4	6,000	75.00%		
Enron-5	5,175	71.01%		
Enron-6	6,000	75.00%		

Table 7.4: Brief description of the email datasets.

need a balance between FPR and FNR, which is the AUC in this case. The AUC is measured using the *Receiver Operating Characteristics (ROC)* curves. The ROC curve is a 2-D graph whose *Y-axis* is the *true positive rate* (which is indeed 1−FNR) and *X-axis* is the FPR, and therefore depicts the trade-offs between the cost of $n_{s\to h}$ and $n_{h\to s}$.

7.5 Datasets

A number of standard email datasets are publicly available and widely used. In our experiment, we chose four of them: (i) CSDMC2010, (ii) SpamAssassin, (iii) LingSpam, and (iv) Enron-Spam. The reasons for choosing these datasets are manifold. Firstly, emails in these datasets have been sent out between 2000 and 2010. This provides an interesting test-bed that characterizes the change of language of emails spanning across a decade. Secondly, we are interested to evaluate our method with spam-skewed datasets. The reason behind this interest is because with ham-skewed datasets, even a poor classifier can achieve good FPR by classifying most of the emails, if not all, as hams (see, for example, Bratko *et al.* [24]). Therefore, we include Enron 4-6 in our experiment. Thirdly, we include LingSpam in our dataset because not only are its hams domain-specific but also its excerpts are of scholarly discussions on linguistics. As a result, we are expecting some features, *Error Features* (see Section 7.2.2) in particular, to be more useful for the LingSpam emails. Fourthly, we include datasets that are not explored by many (e.g., CSDMC2010 and Enron-Spam). Last but not least, the use of Enron-Spam gives us an opportunity to work with hams coming from a user's personal inbox.

7.5.1 Description

CSDMC2010[6], among the four, is the latest collection of emails. The spam rate of this dataset is reasonable—about 32%. Both hams and spams in this dataset are collected randomly (i.e., not from any particular inbox). CSDMC2010 is relatively new and except for the *ICONIP-2010*

[6] `http://csmining.org/index.php/spam-email-datasets-.html`

challenge participants, this dataset has not been explored by many. In contrast, SpamAssassin[7] is one of the most popular public datasets. Like CSDMC2010, the emails of SpamAssassin are also collected randomly. In addition, the spam rate of this dataset is almost equal to that of CSDMC2010. The LingSpam dataset [25] is both the smallest and oldest dataset used in this study. Moreover, its spam rate is smaller than the preceding datasets—only about 17%. It is, however, the odd one out of the four as the hams in this dataset are collected from the discussions of a linguistics forum; the spams, on the other hand, are collected randomly. Enron-Spam [9] is an email collection comprising six different datasets each of which contains ham messages from a single user of the Enron corpus. Of the six datasets, the bulk of the emails in Enron 1-3 are hams while the bulk of the emails in Enron 4-6 are spams. This collection is very different compared to the others as the hams of each dataset bear the characteristics of one individual. In this paper, we experiment on each of these six datasets and report the average performance. Table 7.4 briefly outlines the datasets used in this experiment.

7.5.2 Pre-processing

We noticed that spam emails have some features that can easily distinguish them from hams. Therefore, to provide a conservative estimate of our method's performance, we follow these pre-processing steps: First, symbols (like $ or ! signs) or spam words (like *porn*, *webcam*, or *lottery*) are excluded from the *subject* field of the emails. Second, *attachment* fields (if any) are excluded from the email texts. Third, non-ASCII characters present in the email texts not only are spam indicators but also the presence of such characters can change the readability of the entire message. As readability is one of our key features (see Section 7.2.3), we remove the non-ASCII characters from the messages as well.

7.6 Experimental Results

7.6.1 Feature Importance

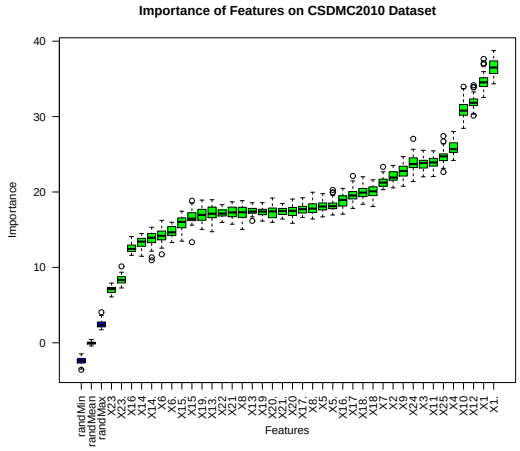
In machine learning, the identification of features relevant for classification is important, so is the observation of results when features work in groups. This identification and observation give us a set of features that are important [26] [27]. To measure feature importance, we use a *feature importance* measuring algorithm named *Boruta*[8] that uses a wrapper around Random Forest. The details of the *Boruta* algorithm are beyond the scope of this paper but can be found in the study by Kursa and Rudnicki [26]. We applied the algorithm on each of the datasets. Of note, due to limited space, the analysis of feature importance for the six Enron-Spam datasets are made available elsewhere[9].

Figure 7.1 exhibits the *boxplots* created with *Boruta* for the non-personalized email datasets. The *Y-axis* of each plot represents feature importance (a measure specific to the algorithm) while the *X-axis* represents feature labels (see Section 7.2). The features are sorted according to the ascending order of their importance in determining class labels. The key

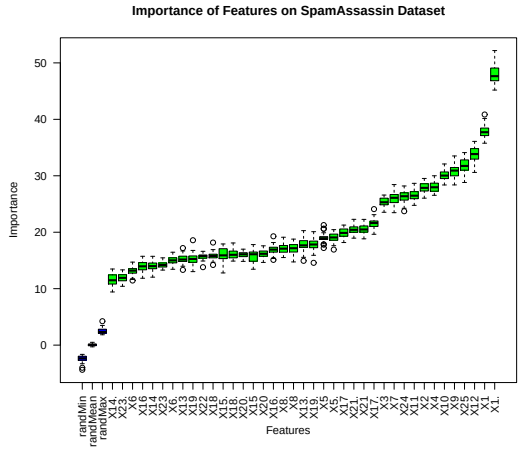
[7] `http://spamassassin.apache.org/publiccorpus/`

[8] http://cran.r-project.org/web/packages/Boruta/index.html

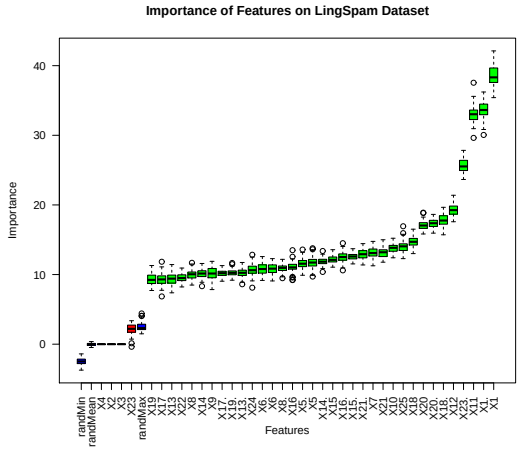
[9] `http://cogenglab.csd.uwo.ca/sentinel/feature-importance.html`



(a) The order of the features according to their importance for the CSDMC2010 dataset.



(b) The order of the features according to their importance for the SpamAssassin dataset.



(c) The order of the features according to their importance for the LingSpam dataset.

Figure 7.1: Feature importance on three datasets according to the *Boruta* algorithm.

finding in the plots is that for all the datasets, the most important feature is *Spam Words* (X_1). Another interesting finding is that the *error features* (see Section 7.2.2) are close in importance. In addition, the TF·IDF of simple words (X_{25}) and the HTML features (X_2 and X_3, in particular) are important features for CSDMC2010 and SpamAssassin emails (Figures 7.1a and 7.1b). The relative position according to the importance of *Alpha-numeric Words* (X_5) is similar in all three datasets as is the *Verbs* feature (X_6). Interestingly, the *function word* feature (X_7) performed reasonably well—to the best of our knowledge not many consider this as a spam detection feature. Except TF·IDF of simple and complex words, most of the *readability features* are on the left side of the graph—seeming to be less important. Note that the significance of the readability features becomes evident when we, next, investigate the feature performance in groups.

We are also interested in reporting the importance of the features in groups. To do this, we used the traditional features (Section 7.2.1) as our baseline. These features are first used by the

algorithms to classify the emails. Then, we added the text features and HTML features. As well, we added the readability features with the baseline. FPR is chosen as the measure to evaluate the classification. Figure 7.1 strongly suggests that the traditional features are important for spam classification. The interesting results shown in Table 7.5, however, indicate the effect of combining the features. One common attribute of these results is that the ham misclassification rate decreases as the baseline features are combined with text and readability features. Another significant finding is that combining the readability features with the baseline may not produce the optimal result but combining these two groups of features with text features (i.e., using the entire feature pool) improves the ham misclassification rate. However, in that case the two exceptions are CSDMC2010 and Enron-Spam. Also from the results, the lowest FPR is achieved by the RFS—BAGGED RF, in particular. Although the FPRS for SVM and NB seem promising, their significantly low AUC indicates a possible overfit, especially for the CSDMC2010 and LingSpam datasets. The results for SVM and NB, therefore, are in doubt. The lowest FPR in Table 7.5 is achieved by BAGGED RF using all features on the most language homogeneous dataset, the LingSpam dataset.

Note that the FPRS that are not statistically significant compared to the comparable baseline are written in *italics*.

7.6.2 Classification Performance Evaluation

Treating each dataset independently, the real-valued features are extracted from each email of each dataset. Then, using a conventional stratified 10-fold cross-validation approach, five classifiers are generated using the five algorithms. The classifiers are then evaluated. In a K-fold cross-validation, the original dataset is randomly partitioned into K equal-sized folds or subsets. Then each classifier is trained on $K - 1$ folds and evaluated on the remaining fold. Stratification means that the class (i.e., ham or spam) in each fold is represented in approximately the same proportions as in the full dataset. The cross-validation process is then repeated until each of the K folds is used exactly once as the validation data. The final estimation of the classifier is the average of the K results from the folds.

Table 7.6 shows the performance of the learned classifiers on spam email classification. The most striking attribute of this data is that the best ham misclassification rate is achieved by the BAGGED RF classifiers. In contrast, the best spam misclassification rate is attained by BOOSTED RF. To decide the best, we then refer to a balanced measure of the two: the AUC. According to the data shown in Table 7.6, BAGGED RF performs the best of all because it has the better AUC. A further indicator of the supremacy of BAGGED RF over the others can be found in Table 7.6c—its best FPR comes from LingSpam indicating that it performs best at identifying domain-specific hams. The interesting competition between BAGGED RF and BOOSTED RF continues on precision and recall. For all four datasets, BAGGED RF achieves the best precision while the best recall is scored by BOOSTED RF. For two datasets, SpamAssassin and LingSpam, BOOSTED RF achieves the best F-score. On the other hand, BAGGED RF ties with BOOSTED RF for CSDMC2010; for Enron-Spam it is BAGGED RF that attains the best F-score. Recall that, CSDMC2010 and SpamAssassin have similar characteristics as well as spam rates (see Section 7.5). This is complemented by the results in Tables 7.6a and 7.6b. The data show that except for the SVM classifiers, others have similar FPRS. When it comes to accuracy, BAGGED RF again outperforms the other algorithms—the only exception is the SpamAssassin dataset.

	Baseline	Baseline + Text Feature	Baseline + Readability Features	All
RF	0.060	0.034	0.038	0.040
BOOSTED RF	0.062	0.028	0.028	0.030
BAGGED RF	0.055	0.023	0.023	0.020
SVM	0.022	0.037	*0.023*	0.027
NB	0.031	0.065	0.093	0.101

(a) FPR of different groups of features combined with the baseline for the CSDMC2010 dataset.

	Baseline	Baseline + Text Feature	Baseline + Readability Features	All
RF	0.064	0.040	0.045	0.034
BOOSTED RF	0.071	0.029	0.034	0.026
BAGGED RF	0.061	0.026	0.028	0.022
SVM	0.055	*0.063*	*0.050*	*0.052*
NB	0.060	0.075	0.094	0.104

(b) FPR of different groups of features combined with the baseline for the SpamAssassin dataset.

	Baseline	Baseline + Text Feature	Baseline + Readability Features	All
RF	0.041	0.019	*0.023*	0.017
BOOSTED RF	0.044	0.018	0.016	0.017
BAGGED RF	0.040	0.011	0.015	0.009
SVM	0.0004	0.012	0.020	0.014
NB	0.063	0.091	0.214	0.218

(c) FPR of different groups of features combined with the baseline for the LingSpam dataset.

	Baseline	Baseline + Text Feature	Baseline + Readability Features	All
RF	0.404	0.164	0.245	0.174
BOOSTED RF	0.404	0.156	0.240	0.158
BAGGED RF	0.402	0.143	0.218	0.150
SVM	0.424	*0.371*	*0.442*	*0.416*
NB	0.408	0.304	0.376	0.336

(d) FPR of different groups of features combined with the baseline for the Enron-Spam dataset.

Table 7.5: The effect of incrementing groups of features on spam classification for (a) CSDMC2010, (b) SpamAssassin, (c) LingSpam, and (d) Enron-Spam Datasets. The FPRs that are not statistically significant compared to the comparable baseline are written in *italics*.

	FPR	FNR	Accuracy %	Precision	Recall	F-score	AUC
RF	0.040	0.092	94.338	0.914	0.908	0.911	0.980
BOOSTED RF	0.030	0.089	95.124	0.934	0.912	0.922	0.980
BAGGED RF	0.020	0.107	95.193	0.953	0.893	0.922	0.988
SVM	0.027	0.390	85.718	0.913	0.610	0.730	0.792
NB	0.101	0.396	80.471	0.737	0.604	0.662	0.855

(a) Evaluation measures of the full-featured spam classifiers for the CSDMC2010 dataset.

	FPR	FNR	Accuracy %	Precision	Recall	F-score	AUC
RF	0.034	0.093	94.707	0.923	0.907	0.915	0.979
BOOSTED RF	0.026	0.079	95.700	0.941	0.921	0.931	0.982
BAGGED RF	0.022	0.099	95.353	0.948	0.901	0.924	0.986
SVM	0.052	0.292	87.265	0.861	0.708	0.777	0.828
NB	0.104	0.558	75.373	0.660	0.443	0.529	0.847

(b) Evaluation measures of the full-featured spam classifiers for the SpamAssassin dataset.

	FPR	FNR	Accuracy %	Precision	Recall	F-score	AUC
RF	0.017	0.162	95.817	0.907	0.838	0.869	0.978
BOOSTED RF	0.017	0.162	95.886	0.910	0.838	0.871	0.977
BAGGED RF	0.009	0.193	95.956	0.944	0.807	0.868	0.986
SVM	0.014	0.341	93.156	0.907	0.659	0.760	0.822
NB	0.218	0.277	77.186	0.402	0.723	0.515	0.831

(c) Evaluation measures of the full-featured spam classifiers for the LingSpam dataset.

	FPR	FNR	Accuracy %	Precision	Recall	F-score	AUC
RF	0.174	0.072	91.681	0.887	0.927	0.906	0.960
BOOSTED RF	0.158	0.070	92.288	0.896	0.929	0.912	0.956
BAGGED RF	0.150	0.080	92.521	0.910	0.919	0.914	0.972
SVM	0.416	0.379	78.350	0.836	0.620	0.627	0.602
NB	0.336	0.302	73.344	0.680	0.697	0.686	0.750

(d) Average evaluation measures of the full-featured spam classifiers for the Enron-Spam dataset.

Table 7.6: Performances of the classifiers on (a) CSDMC2010, (b) SpamAssassin, (c) LingSpam, and (d) Enron-Spam Datasets.

Compared to the preceding results, the performances of SVM and NB are not satisfactory. Attaining very low FPR with a low AUC by SVM for CSDMC2010 and LingSpam indicates a possible training overfit. The reasons for this possible overfit are yet to be investigated. On the other hand, we examined the correlation among the features. It is evident that the features have inter-dependency[10]—which contradicts the independence assumption of the NB algorithm and

[10] `http://cogenglab.csd.uwo.ca/sentinel/sentinel-attribute-correlations.html`

	FPR	FNR	Accuracy %	Precision	Recall	F-score	AUC
RF	0.187	0.320	71.411	0.796	0.714	0.731	0.830
BOOSTED RF	0.159	0.329	71.510	0.805	0.712	0.732	0.843
BAGGED RF	0.125	0.369	69.473	0.809	0.695	0.713	0.855
SVM	0.059	0.578	55.772	0.799	0.558	0.570	0.681
NB	0.176	0.634	48.529	0.712	0.485	0.497	0.645

(a) Classifiers trained on ham skewed Enron 1-3 and tested on spam skewed Enron 4-6.

	FPR	FNR	Accuracy %	Precision	Recall	F-score	AUC
RF	0.458	0.077	64.494	0.808	0.645	0.661	0.865
BOOSTED RF	0.445	0.069	65.709	0.815	0.657	0.673	0.887
BAGGED RF	0.379	0.061	70.793	0.834	0.708	0.723	0.908
SVM	0.842	0.028	37.833	0.763	0.378	0.321	0.564
NB	0.734	0.089	44.048	0.732	0.440	0.425	0.717

(b) Classifiers trained on spam skewed Enron 4-6 and tested on ham skewed Enron 1-3.

Table 7.7: Performances of the classifiers on Enron-Spam.

can be a reason for the algorithm's poor classification performance.

The data in Table 7.6d describe the performance of our method on the Enron-Spam dataset. The most intriguing finding is that on this dataset, the classifiers' FNRs are remarkably low. Inevitably, the FPRs of the classifiers on the dataset are the poorest. Although the AUC is still reasonable, the results are not quite as good as we expected. This phenomenon confirms that for personalized email data, our approach misclassifies a lot of legitimate emails. Two other notable aspects of the reported data are that on the Enron-Spam collection the classifiers have (i) the best accuracy and (ii) the poorest recall.

From the results of Table 7.6, it can be seen that except for Enron-Spam, the FPRs of the classifiers are much better than their FNRs. The most likely cause for the classifiers labelling hams more correctly than spams is that CSDMC2010, SpamAssassin, and LingSpam are ham skewed (Table 7.5), To see whether the training on a skewed dataset affects the overall performance of the classifiers, we further tested with the Enron-Spam dataset. We first trained the classifiers with Enron 1-3 and tested them on Enron 4-6. And then, we trained the classifiers with Enron 4-6 and tested them on Enron 1-3. It is evident from the data in Table 7.7 that training on a ham skewed dataset results in improved FPR while training on a spam skewed dataset brings about better FNR. A significant change to spam recall has also been observed from the results as spam recalls listed in Table 7.7a are better than that in Table 7.7b. In other words, the results suggest that no matter how we train our classifiers, either with spam or ham skewed training data, we need our test set to contain more spams than hams to get better spam recall. This finding is significant because a low recall results in low F-score and AUC. Overall, from this particular experiment, we can say that to observe the true performance of spam classifiers, reported work should test their system on a balanced dataset.

7.7 Discussion and Related Work

In this section, our results are compared with some of the related work. Of the references mentioned throughout this paper, we only compare our results with the commensurate ones—those that used the same dataset. Moreover, the compared results are evaluated with a *student t-test* with a significance level set to 5% (i.e., $\alpha = 0.05$) to report if the differences are significant.

We have mixed results for the CSDMC2010 dataset. Qaroush *et al.* [19], for instance, investigated the performance of several learning algorithms on this dataset. They concluded that RF outperforms the rests. The reported spam recall in their paper is 0.958, which is significantly better than what we found (0.912) (Table 7.6a). Whereas their precision is similar to that of our approach, because of their high recall, their 0.958 F-score also outperforms our F-score of 0.922 (Table 7.6a). Surprisingly, we outperform them if we do a cost-sensitive analysis of our data. The AUC that we found for the dataset is 0.988 (Table 7.6a) which is better than what they found (0.981). An SVM-based spam filter developed by Yang *et al.* [28], on the other hand, reported 0.943 precision, 0.965 recall, and a promising AUC of 0.995. Among the three measures, we only obtained a better precision. Their second anti-spam filter uses a an NB classifier. This filter, interestingly, achieved 100% recall. Its precision of 0.935 and AUC of 0.976, however, was outperformed by our approach (Table 7.6a). Note that, the differences in the results are statistically significant.

By using 328 features, the filter developed by Ma *et al.* generates a Neural Network classifier. On the SpamAssassin dataset, they reported that both their precision and accuracy was 0.920. On the other hand, our approach achieved a 0.948 precision and 0.957 accuracy. Both of these results are statistically significant. Another Neural Network based filter developed by Srisanyalak and Sornil [29] uses immunity-based features from emails. The filter has been reported to be accurate 92.4% of the time. Our reported accuracy is better than this (Table 7.6b). The phenomenal FPR and FNR achieved by the filter developed by Bratko *et al.* (FPR=0.001 and FNR=0.012) indicates that our approach needs further improvement in these measures; our reported FPR and FNR are 0.023 and 0.079, respectively (Table 7.6b).

From previous studies, we found that the performance of the filters are relatively low on the LingSpam dataset. Prabhakar and Basavaraju [10], for instance, applied K-NNC and a data clustering algorithm called BIRCH on this dataset. Their filter achieved 0.698 precision, 0.637 recall, 0.828 specificity, and an accuracy of 0.755. In contrast, the data in Table 7.6c show that our approach has a precision of 0.944 with 0.838 recall, 0.990 specificity ($1 - \text{FPR}$), and 0.960 accuracy. Our reported AUC on LingSpam also outperformed that reported by Cormack and Bratko [30]; our AUC of 0.986 is significantly better than their AUC of 0.960. The recall we have on this dataset is much better than that reported by Yang *et al.* [28]; the precisions, however, are similar. Their NB-based filter achieved 0.943 precision and 0.820 recall. Surprisingly, the AUC of their filter (e.g., 0.992) significantly outperformed the AUC of our approach (Table 7.6c).

As mentioned in Section 7.6.2, our results with the Enron-Spam dataset are not satisfactory because of the *properly balanced* property of the dataset. The curators of the dataset, however, reported a spectacular spam recall of 0.975 [9] while our best spam recall on the dataset is 0.929 with BOOSTED RF. Moreover, their reported ham recall is 0.972; ours is a mere 0.842 (Table 7.6d). However, we have recently surpassed the results reported by Metsis *et al.* [9] using an anti-spam filter named SENTINEL [31] that we have developed using the ideas presented

in this paper.

7.8 Conclusions

To sum up, we consider the task of email classification as a supervised machine-learning problem. The novelty of this work is the use of a set of features related to the readability of email texts. Because the features are language-independent, the method reported in this paper is potentially able to classify emails written in any language. The aforementioned features as well as the traditional ones are used to generate binary classifiers by five well-known learning algorithms. We then evaluate the classifier performances on four benchmark email datasets. The evidence from this study suggests that although traditional features are individually more important than the other feature types, the combination of all of the features produces the optimal results. Extensive experiments also imply that classifiers generated using meta-learning algorithms perform better than trees, functions, and probabilistic methods. Finally, we compare the results of our method with that of many state-of-the-art anti-spam filters. Although the performance of our method is not always superior to other filter-dataset instances, we find that our approach surpasses a number of them. Taken together, the results suggest that the method described in this paper can be a good means to classify spam emails.

Because our results suggest that meta-learning algorithms perform the best, further tests should be carried out to see the performance of classifiers generated by stacking several algorithms.

Bibliography

[1] L. F. Cranor and B. A. LaMacchia, "Spam!," *Communication ACM*, vol. 41, pp. 74–83, Aug. 1998.

[2] Commtouch, "Internet threats trend report," tech. rep., Commtouch, USA, April 2013.

[3] J. Goodman, G. V. Cormack, and D. Heckerman, "Spam and the ongoing battle for the inbox," *Communications ACM*, vol. 50, pp. 24–33, Feb. 2007.

[4] E. Blanzieri and A. Bryl, "A survey of learning-based techniques of email spam filtering," *Artificial Intelligence*, vol. 29, pp. 63–92, Mar. 2008.

[5] L. Zhang, J. Zhu, and T. Yao, "An evaluation of statistical spam filtering techniques," *ACM Transactions on Asian Language Information Processing*, vol. 3, pp. 243–269, 2004.

[6] C.-C. Lai and M.-C. Tsai, "An empirical performance comparison of machine learning methods for spam e-mail categorization," in *Fourth International Conference on Hybrid Intelligent Systems (HIS '04)*, (USA), pp. 44–48, IEEE Computer Society, 2004.

[7] Y. Hu, C. Guo, E. W. T. Ngai, M. Liu, and S. Chen, "A scalable intelligent non-content-based spam-filtering framework," *Expert Systems Applications*, vol. 37, pp. 8557–8565, Dec. 2010.

[8] J.-J. Sheu, "An efficient two-phase spam filtering method based on e-mails categorization," *International Journal of Network Security*, vol. 9, no. 1, pp. 34–43, 2009.

[9] V. Metsis, I. Androutsopoulos, and G. Paliouras, "Spam filtering with Naive Bayes – Which Naive Bayes?," in *Third Conference on Email and Anti-Spam (CEAS 2006)*, (USA), 2006.

[10] R. Prabhakar and M. Basavaraju, "A novel method of spam mail detection using text based clustering approach," *International Journal of Computer Applications*, vol. 5, pp. 15–25, August 2010.

[11] Q. Ma, Z. Qin, F. Zhang, and Q. Liu, "Text spam neural network classification algorithm," in *2010 International Conference on Communications, Circuits and Systems*, (China), pp. 466–469, 2010.

[12] P. Graham, "A plan for spam." Available on: http://paulgraham.com/spam.html, Aug. 2003.

[13] C. Orăsan and R. Krishnamurthy, "A corpus-based investigation of junk emails," in *Third International Conference on Language Resources and Evaluation (LREC-2002)*, (Spain), May, 29 – 30 2002.

[14] R. Gunning, "Fog index after twenty years," *Journal of Business Communication*, vol. 6, no. 3, pp. 3–13, 1969.

[15] R. Flesch, "A new readability yardstick," *Journal of Applied Psychology*, vol. 32, pp. 221–33, 1948.

[16] G. H. McLaughlin, "Smog grading – a new readability formula," *Journal of Reading*, vol. 12, no. 8, pp. 639–46, 1969.

[17] J. S. Caylor, T. G. Stitch, L. C. Fox, and J. P. Ford, "Methodologies for determining reading requirements of military occupational specialities," Tech. Rep. 73-5, Human Resources Research Organization, Alexandria, VA, 1973.

[18] J. P. Kincaid, R. P. Fishburne, R. L. Rogers, and B. S. Chissom, "Derivation of new readability formulas (automated readability index, fog count, and Flesch reading ease formula) for navy enlisted personnel," Research Branch Report 8-75, Chief of Naval Technical Writing: Naval Air Station Memphis, 1975.

[19] A. Qaroush, I. M. Khater, and M. Washaha, "Identifying spam e-mail based-on statistical header features and sender behavior," in *CUBE International Information Technology Conference*, (USA), pp. 771–778, ACM, 2012.

[20] R. E. Schapire and Y. Singer, "Boostexter: A boosting-based system for text categorization," in *Machine Learning*, pp. 135–168, 2000.

[21] L. Breiman and L. Breiman, "Bagging predictors," in *Machine Learning*, pp. 123–140, 1996.

[22] M. Ye, T. Tao, F.-J. Mai, and X.-H. Cheng, "A spam discrimination based on mail header feature and svm," in *Fourth International Conference on Wireless Communications, Networking and Mobile Computing (WiCom08)*, pp. 1–4, 2008.

[23] A. Veloso, "Lazy associative classification for content-based spam detection," in *Proc. of the Latin American Web Congress*, pp. 154–161, IEEE Computer Society, 2006.

[24] A. Bratko, G. V. Cormack, D. R, B. Filipic, P. Chan, T. R. Lynam, and T. R. Lynam, "Spam filtering using statistical data compression models," *Journal of Machine Learning Research*, vol. 7, pp. 2673–2698, 2006.

[25] I. Androutsopoulos, J. Koutsias, K. Chandrinos, G. Paliouras, and C. Spyropoulos, "An evaluation of naive bayesian anti-spam filtering," in *Proc. of the Workshop on Machine Learning in the New Information Age*, 2000.

[26] M. B. Kursa and W. R. Rudnicki, "Feature selection with the Boruta package," *Journal of Statistical Software*, vol. 36, no. 11, pp. 1–13, 2010.

[27] R. Nilsson, J. M. Peña, J. Björkegren, and J. Tegnér, "Consistent feature selection for pattern recognition in polynomial time," *J. Mach. Learn. Res.*, vol. 8, pp. 589–612, May 2007.

[28] J. Yang, Y. Liu, Z. Liu, X. Zhu, and X. Zhang, "A new feature selection algorithm based on binomial hypothesis testing for spam filtering," *Knowledge-Based Systems*, vol. 24, pp. 904–914, Aug. 2011.

[29] B. Sirisanyalak and O. Sornil, "Artificial immunity-based feature extraction for spam detection," in *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, (SNPD 2007)*, vol. 3, pp. 359–364, 2007.

[30] G. V. Cormack and A. Bratko, "Batch and online spam filter comparison," in *Conference on Email and Anti-Spam, (CEAS 2006)*, (Mountain View, CA), July 2006.

[31] R. Shams and R. Mercer, "Personalized spam filtering using natural-language attributes," in *Proceedings of the 12th IEEE International Conference on Machine Learning Applications (ICMLA2013)*, (Miami, USA), IEEE, 2013.

Chapter 8

Personalized Spam Filtering with Natural Language Attributes

This chapter is based on the paper titled "Personalized Spam Filtering with Natural Language Attributes" co-authored with Robert E. Mercer that appeared in the 12th IEEE International Conference on Machine Learning Applications (ICMLA2013).

In this paper, we report the performance of an anti-spam filter named SENTINEL. In addition to some commonplace attributes, SENTINEL uses attributes related to natural language stylometry. The filter has been tested with six benchmark datasets in the Enron-Spam collection. Classifiers generated by well-known meta-learning algorithms like ADABOOSTM1 and BAGGING perform equally the best, while a Random Forest (RF) generated classifier performs almost as well. The performance of classifiers using Support Vector Machine (SVM) and Naïve Bayes (NB) are not satisfactory. Comparisons show that the performance of SENTINEL surpasses that of a number of state-of-the-art personalized filters proposed in previous studies.

8.1 Introduction

Today's spam emails are very different than what they were called in the 1980s. Simply, they are not just *"Usenet messages cross-posted to numerous newsgroups"* anymore [1]. Now, spams are in fact used to refer to unsolicited, massively posted commercial and non-commercial emails. The onslaught of spams has grown exponentially. In March 2013, for instance, 100 billion spams were sent out daily. This quantity is 98% more than that from the previous quarter [2]. The effects of spams include loss of individual productivity, financial loss of organizations, cluttering of user inboxes, and consumption of network bandwidth. Therefore, spam filtering is an important challenge.

Spam filtering is an interesting binary classification problem. Classifiers are trained on a reasonable quantity of spams and hams (i.e., legitimate emails), and then applied on unseen emails to classify them into one of these two categories. Classifiers induced by well-known algorithms like Naïve Bayes (NB) [3] [4] [5], Random Forest (RF) [6], Support Vector Machine (SVM) [7] and Neural Networks (NN) [8] are used in many operational anti-spam filters. As test beds for these classifiers, several email datasets have been used most of which contain emails that are collected from random sources. Enron-Spam [3], on the other hand, is a collection of

six benchmark datasets that is useful to test personalized filters, i.e., filters that are trained on incoming messages of a particular user that they are intended to protect.

Natural language attributes of email subject and body have a considerable ability to discern spams and hams [3] [8]. Most of these attributes are based on the importance of a term in the email (i.e., term frequency or TF) and its rarity in the email dataset (i.e., inverse document frequency or IDF). The TF·IDF attributes are very promising for personalized filters (see, for example, [3] and [9]), whereas Lai and Tsai [5] report that TF·IDF does not perform as well on randomly collected emails as on personalized emails. Furthermore, the calculation of TF·IDF is done on the word-count space. Therefore, for each newly arrived email, this attribute needs to be re-calculated. The re-calculation, if not done incrementally, can introduce latency. As well, the use of *terms* by spammers changes over time [10]. Therefore, the value of attributes based solely on *terms* and their *frequency* may decrease because of this time-sensitivity. However, there are natural language attributes that are still unexplored in this problem domain, viz. readability, grammar and spelling mistakes, and the use of function and content words. Similar attributes have been proposed by writer stylometry and used to detect fraud in documents [11].

In this paper, we report the performance of an anti-spam filter named SENTINEL on personalized emails. Besides conventional spam filtering attributes, SENTINEL uses many natural language attributes that are related to email readability, grammar, spelling, use of function and content words, TF·IDF, etc. The attributes are extracted from six datasets of the Enron-Spam collection. In this experiment, we test the classifiers generated by SENTINEL using Random Forest (RF), Support Vector Machine (SVM), and Naïve Bayes (NB). In addition, two more classifiers are generated using the meta-algorithms ADABOOSTM1 and Bootstrap Aggregating (BAGGING). We use RF as the base classifier for both and name these classifiers BOOSTED RF and BAGGED RF, respectively. Results show that BOOSTED RF and BAGGED RF perform almost equally the best while the performance of RF is a close second. Interestingly, the performance of SVM depends on the quantity of spams in the training set. NB has the poorest results of all—which is understandable as most of the attributes exhibit inter-dependency. Comparisons show that the performance of SENTINEL surpasses that of a number of state-of-the-art personalized filters proposed in previous studies. The natural language attributes of SENTINEL are similar to stylometric attributes and are therefore language independent. As a result, the filter may be an excellent means to classify spam emails in any language.

The next section details the related work. Following that, we describe the materials and methods in Section 8.3. In Section 8.4, we report the experimental findings. Finally, Section 8.5 concludes the paper.

8.2 Related Work

The earliest of the NB anti-spam filters were simple and computationally efficient. Likewise, they attained low misclassification rates on several datasets. These filters exploited the simple Bayesian framework, a set of rules, and both header and content attributes [12]. The initial success of these filters led several others to emerge by simply replacing the rules with predictive models (see, for example, [3] [5] [13] [14]). Several variations of these filters include *multivariate Bernoulli, Bag of Words (BoW), multinomial boolean*, etc. Metsis *et al.* [3], for instance, used 3000 multinomial boolean TF attributes—the results were very impressive. Two

years later, they achieved even better results by using the *transformed* TF attributes [15]. Nowadays, the performance of an NB filter is considered as the *de facto* standard to compare a newly developed method.

SVM filters are efficient for training in much the same way as NB filters nevertheless they need incremental training to reduce latency [16]. Furthermore, for this problem domain, SVM filters can handle large attribute sets [7]. Most of the benchmark SVM filters use the frequency-based *linear kernel*. As well, there are several SVM filters that use updatable supervised clustering algorithms [17]. However, the trade-off with SVM filters is that their misclassification rate for personalized emails is quite high.

The advantages of using *meta-learning* anti-spam filters are manifold. Firstly, when a base learner with a sufficient tree depth is used, they achieved low misclassification rates on many public datasets [18]. Secondly, these filters are resistant to the problem of overfitting and therefore they gain a more *appropriate* accuracy even on an unbalanced dataset[1] [19]. However, these filters have the weakness of *ensemble* learning—the interpretation of results is difficult. Studies show that meta-learning filters outperform many decision tree, NB, and SVM filters. Surprisingly, the use of this category of filters is still not as widespread as NB and SVM filters.

Over the last decade, Artificial Immune System based anti-spam filters have become a popular choice. These filters use *detectors* on the email for pattern matching. Detectors are in fact regular expressions that are defined *a priori*. Each detector is given a *weight* that is adjusted as the filters recognize a pattern in a given email. The weights of the matching detectors are then used (usually combined) to determine the email's class label. Notable immune system based filters are reported in [9], [20], and [21]. As the filters seek specific *signatures* in the emails, they are widely used in personalized email classification where similar patterns can be found in the writing style of the person the filter intends to protect. Also, many of these filters are able to deal with *Concept Drift*— the gradual or abrupt change of thematic context over time such as new advertisement themes in spam.

Ham + Spam	**Ham:Spam**	**Time-stamp (Ham, Spam Duration)**	
Enron 1 (`farmer-d` + GP)	3672:1500	[12/99, 1/02]	[12/03, 9/05]
Enron 2 (`kaminski-v` + SH)	4361:1496	[12/99, 5/01]	[5/01, 7/05]
Enron 3 (`kitchen-l` + BG)	4012:1500	[2/01, 2/02]	[8/04, 7/05]
Enron 4 (`williams-w3` + GP)	1500:4500	[4/01, 2/02]	12/03, 9/05]
Enron 5 (`beck-s` + SH)	1500:3675	[1/00, 5/01]	[5/01, 7/05]
Enron 6 (`lokay-m` + BG)	1500:4500	[6/00, 3/02]	[8/04, 7/05]

Table 8.1: The details of the Enron-Spam collection as described by Metsis *et al.* [3]: composition, ham-spam ratio, and time-stamps of the emails.

8.3 Materials and Methods

8.3.1 Email Datasets

For decades, several email datasets have been used to gauge the performance of anti-spam filters, viz. LingSpam, SpamAssassin, and CSDMC2010. Because the emails in these datasets are collected randomly and therefore do not represent personal inboxes, anti-spam filters aimed at protecting a particular user cannot be evaluated on them. Simply put, to evaluate personalized spam filtering, we need personalized email data. Enron-Spam[2] [3] is a collection of emails composed of six datasets each containing ham emails from a single person in the Enron corpus. These ham collections are dubbed as follows: `farmer-d`, `kaminski-v`, `kitchen-l`, `williams-w3`, `beck-s`, and `lokay-m`. The spams are collected from three different sources. First, a mix of spams that are collected from the SpamAssassin corpus and spam traps of the Honeypot project[3] are put together; these spams are dubbed as (SH). Second, BG spams are collected from the spam traps of Bruce Guenter[4]. Third, spams collected randomly from the mailbox of Georgios Paliouras [3] (GP). The foregoing six ham email collections are each paired with one of these three spam collections (SH, BG, and GP). Thereafter, the six collections are dubbed as Enron 1 to Enron 6. Of the six collections, Enron 1–3 are ham skewed (ham:spam is 3:1) while Enron 4–6 are spam skewed (ham:spam is 1:3). The summary of the characteristics of the Enron-Spam dataset can be found in Table 8.1.

For conservative estimates of the proposed filter, the following pre-processing steps are considered. First, the SUBJECT field of spams can contain symbols such as $ or !. As well, they contain spam words such as *porn*, *webcam*, or *lottery*. Therefore, such entries are excluded from the SUBJECT fields of the emails. Second, many emails in the collection can contain an ATTACHMENT field. This extraneous field is removed from the emails (if any). Third, non-ASCII characters in the email text are removed since the values of our natural language attributes are affected by their presence.

8.3.2 Attribute Selection

Each email in our experiment is represented as $(\vec{x}, y)$, where $\vec{x} \in \mathbb{R}^n$ is a vector of n attributes and $y \in \{spam, ham\}$ is the label of the email. In our study, we explored 37 attributes to classify spam emails and therefore, $n = 37$. Table 8.2 summarizes the attributes used in our experiment.

Word-level Attributes

To calculate the word-level attributes, we treated each email as a bag of words. We curated a dictionary that comprises 381 spam words[5]. Using this dictionary, we counted the frequency of spam words in the emails. This attribute is inspired by the interesting findings of Graham [13] who showed that by merely finding the word *click* in the emails can detect 79.7% of spam

[1]Most of the public email datasets are unbalanced [16].
[2]Downloadable at https://labs-repos.iit.demokritos.gr/skel/i-config/downloads/enron-spam
[3]Consult with http://www.projecthoneypot.org
[4]Overview at http://untroubled.org/spam
[5]Downloadable at `http://cogenglab.csd.uwo.ca/sentinel/spam-term-list.html`

emails in a dataset with only 1.2% ham misclassification rate. Our other word-level attributes include the frequency of alpha-numeric words, verbs, and function words. To identify the verbs in the emails, we used the Stanford part-of-speech Tagger[6]. On the other hand, to identify function words, we utilized the *stoplist function* of a topic indexer named *Maui*[7].

In our experiment, we included the term frequency (TF) attribute used by at least two NB filters [3] [15]. For each term, which appears in at least four training emails, *information gain* scores are computed according to previous work [22]. Of all the terms in the dataset, 3000 with the highest scores are considered as the TF *vector* for all the emails. Thereafter, should any of these 3000 terms be found in an email, the term's corresponding frequency value in the TF vector is first transformed (see [15] for details) and then normalized. Finally, the normalized value is used to calculate the probability score of the email according to Bayes' theorem. The overall method can be found in the study of Kosmopoulos *et al.* [15].

Furthermore, we measured the TF·ISF of each email. Here, the term frequency (TF) is the *square root* of the frequency of a term t in an email M while the inverse sentence frequency (ISF) is a relative measure of whether t is common or rare in M. TF·ISF, therefore, reflects how important each t is to an M. In addition, the measure controls the fact that in an email, some terms are generally more common than others. In contrast, we used TF·IDF of each email as our attribute, which is a numerical statistic that reflects how important a term t is to an email M in the dataset D to which M belongs. The TF·IDF value increases proportionally to the number of times t appears in the email M, but is offset by the frequency of t in the dataset D. The definition of TF is the same as for TF·ISF while inverse document frequency (IDF) measures whether a given term t is common or rare in the dataset D.

Error Attributes

Our next set of attributes is related to the grammar and spelling errors present in the emails. For each email, we simply counted the frequency of grammar and spelling errors using a Java API called LanguageTool[8]. By summing up the values of these two attributes, we introduced a third attribute in this category named *Language Errors*. Interestingly, the error attributes are not normalized since we have obtained better results without normalizing them.

Readability Attributes

Readability deals with the difficulty of reading a sentence, a paragraph or a document. At the heart of readability lies the notion of simple and complex words. Simple words are those that have at most two syllables while complex words contain three or more syllables. Since both types of words have a significant contribution for text readability, we have included them in our attribute list. Five standard scores use these two types of words to determine the readability of a given text: Fog index, Smog index, Flesch reading ease score, Forcast, and Flesch-Kincaid index. For each email, all five readability scores are computed and used as attributes. Among the scores, *Fog index* measures the relative use of complex words in a document. In addition, we modified the Fog index formula to measure the relative use of simple words in a document,

[6]Downloadable at `http://nlp.stanford.edu/software/tagger.shtml`

[7]Downloadable at `https://code.google.com/p/maui-indexer/`

[8]Available at `http://www.languagetool.org/java-api/`

Category	Quantity	Description
Word-level Attributes	11	Spam words, Alpha-numeric words, Function words, Verbs, TF [3] [15], TF·ISF, TF·IDF.
Error Attributes	3	Grammar and spelling mistakes.
Readability Attributes	23	Simple and complex words and their TF·IDF, Fog index, Simple and Inverse Fog index, Smog index, Flesch reading ease score, Forcast, Flesch-Kincaid score, Email length, Word length.

Table 8.2: Summary of the attributes: category, quantity, and description. The stylometric attributes are all Error Attributes, all Readability Attributes, and three Word-level Attributes: Alpha-numeric words, Function words, and Verbs.

and considered that as an attribute, too. Furthermore, we considered the arithmetic inverse of Fog index as another attribute. The other attributes in this category include email length (i.e., total sentences in an email), average word length (i.e., total syllables over total terms in an email), and TF·IDF of simple and complex words. The details of the attributes are discussed elsewhere [23].

Note that the aforementioned attributes are computed by both including and excluding the function words in the emails—the exceptions being the frequency of function words, TF, email length, and TF·IDF attributes. Analyzing the real valued attributes, we have found that their distribution is either left-tailed or right-tailed. This skewness of attribute values, which leads to poor classifiers, has been eliminated by using a *logarithmic transformation* of all the attribute values. Once log-transformed, the distribution of the attributes becomes normal. For each attribute, the transformed values are then normalized by the maximum; the resulting normalized values are therefore in [0, 1].

8.3.3 Learning Algorithms

In this experiment, well-known learning algorithms such as Random Forest RF, Naïve Bayes (NB), and Support Vector Machine (SVM) are used to induce binary classifiers. We also used two *meta-learning algorithms* viz. ADABOOSTM1 and BAGGING. Random Forest has been chosen as the *weak* learner for each, and are named BOOSTED RF and BAGGED RF, respectively. The reasons for choosing these algorithms follow.

As found in much recent research [4] [6], RF is able to produce highly accurate spam classifiers, which is desirable for any anti-spam filter. Overall, RF classifiers also have the reputation for being fast and efficient with large data [24]. On the other hand, although it generates complex models, SVM is a popular choice for anti-spam filters. The algorithm has notable performances with header features [4] [5] [6] [7]. NB is also a widely used learning algorithm for anti-spam filters. It is simple yet provides powerful spam detectors [3] [6]. In addition, on many occasions, NB using simple attributes such as TF·IDF has even outperformed quality learning algorithms, like SVM. Unlike the others, ADABOOSTM1 and BAGGING are meta-learners that improve a given *weak* learning algorithm most of the time [25] [26]. In the following experiments, we have considered RF as the weak learner for these two algorithms. Both ADABOOSTM1

		Actual	
		Spam	Ham
Prediction	Spam	$n_{s\to s}$	$n_{h\to s}$
	Ham	$n_{s\to h}$	$n_{h\to h}$

Table 8.3: Confusion matrix for spam classification problem.

and BAGGING are simple, fast, and above all, are less susceptible to overfitting the training data. And finally, their performances are better than algorithms like NB and *probabilistic* TF·IDF for text categorization tasks.

8.3.4 Evaluation Measures

All of the measures reported in this paper depend on the confusion matrix given in Table 8.3. Precision is the fraction of spam predictions that are correct and can be written as follows:

$\text{PREC} = \frac{n_{s\to s}}{n_{s\to s}+n_{h\to s}}$.

On the other hand, recall—also known as *spam recall*—examines the fraction of spam emails being retrieved:

$\text{REC} = \frac{n_{s\to s}}{n_{s\to s}+n_{s\to h}}$.

F-score (FM), simply, is the harmonic mean of precision and recall. Accuracy, on the other hand, is the percentage of correctly identified spams and hams:

$\text{ACC} = \frac{n_{h\to h}+n_{s\to s}}{n_{h\to h}+n_{h\to s}+n_{s\to h}+n_{s\to s}}$.

Noting that users might accept some spams to enter into their inbox but that they prefer their hams not end up in the spam-traps, email misclassification is cost-sensitive. The ham misclassification (false positive) rate denotes the fraction of all ham emails classified as spams:

$\text{FPR} = \frac{n_{h\to s}}{n_{h\to s}+n_{h\to h}}$.

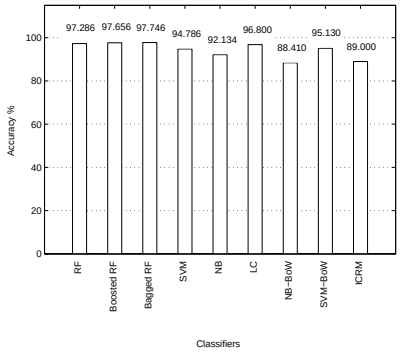
In contrast, the spam misclassification (false negative) rate is the fraction of all spams delivered to the user inbox:

$\text{FNR} = \frac{n_{s\to h}}{n_{s\to h}+n_{s\to s}}$.

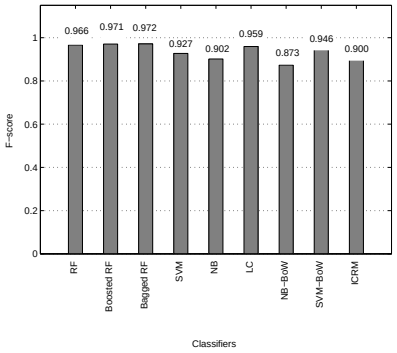
Another viable alternative for the cost-sensitive analysis of email misclassification is to report the *Area Under Curve* (hereinafter, AUC), measured using the *Receiver Operating Characteristics (ROC)* curves. The ROC curve is a 2D graph whose *Y-axis* represents $1 - \text{FNR}$ and whose *X-axis* represents FPR, thereby depicting the compromises between the cost of $n_{s\to h}$ and $n_{h\to s}$.

8.3.5 Experimental Procedure

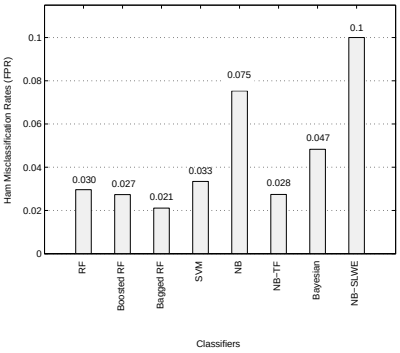
Treating each dataset independently, from each email of the six Enron-Spam datasets, SENTINEL extracts the real-valued attributes using its text processing unit. Using a conventional stratified 10-fold cross-validation approach, the filter then generates for each dataset five classifiers using the five algorithms described in Section 8.3.3. The classifiers are then evaluated. In a K-fold cross-validation, the original dataset is randomly partitioned into K equal-sized folds or subsets. Then each classifier is trained on $K - 1$ folds and evaluated on the remaining fold. Stratification means that the class (i.e., ham or spam) in each fold is represented in approximately the same proportion as in the full dataset. The cross-validation process is then repeated until each of the



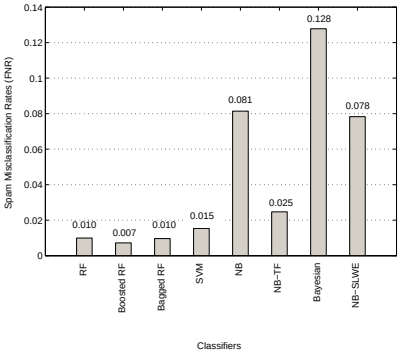
(a) Comparison of accuracy.



(b) Comparison of F-score.



(c) Comparison of ham misclassification rate (FPR).



(d) Comparison of spam misclassification rate (FNR).

Figure 8.1: Comparison of results of the classifiers generated by SENTINEL and other previous studies.

K folds is used exactly once as the validation data. The final estimation of the classifier is the average of the K results from the folds.

8.4 Results and Discussions

To evaluate anti-spam filters on Enron-Spam, scores described in Section 8.3.4 are almost always averaged across the six datasets [3]. The *Arithmetic Mean* is the standard averaging method used, regardless of the skewness present in the datasets (see Section 8.3.1), but we have found that due to the presence of extreme data points, the *Harmonic Mean* provides a more *appropriate* estimate than the arithmetic mean. That said, in this section, we report the *harmonic mean* of the results found from the six datasets.

Measures such as precision, recall, and AUC of the classifiers on Enron-Spam are presented in Table 8.4. The notable result from the data is that RF, BOOSTED RF, and BAGGED RF perform evenly good, which is confirmed by a *student t-test* with $\alpha = 0.05$. The tests further reveal that the first three classifiers in Table 8.4 are better than SVM and NB at $\alpha = 0.05$. The best scores achieved by SENTINEL are highlighted in the chart—except for recall, BAGGED RF has the best

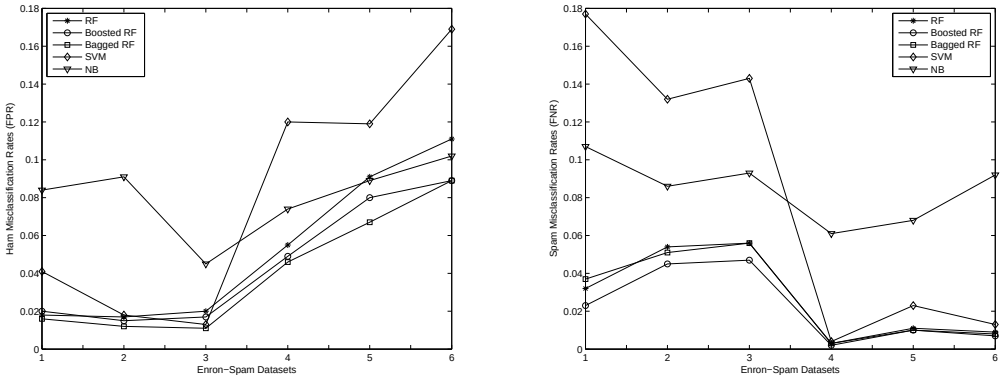
	PREC ± STD. DEV.	REC ± STD. DEV.	AUC ± STD. DEV.
RF	0.961 ± 0.009	0.972 ± 0.006	0.991 ± 0.004
BOOSTED RF	0.964 ± 0.009	**0.977 ± 0.006**	0.988 ± 0.005
BAGGED RF	**0.971 ± 0.009**	0.972 ± 0.007	**0.996 ± 0.002**
SVM	0.942 ± 0.013	0.913 ± 0.008	0.919 ± 0.014
NB	0.889 ± 0.013	0.915 ± 0.008	0.966 ± 0.009

Table 8.4: Precision (averaged), recall (averaged) and AUC of the five spam classifiers along with their standard deviation.

numbers. Interestingly, the NB classifier has a better AUC compared to that of SVM at $\alpha = 0.05$; for the remaining cases, however, the latter performs the better. The *near-perfect* AUCs of the first three classifiers strongly suggest that SENTINEL did not achieve the results by merely guessing everything either as spams (i.e., high $n_{h\to s}$ but low $n_{s\to h}$) or hams (i.e., high $n_{s\to h}$ but low $n_{h\to s}$)—the filter was properly balanced during its labeling. The accuracy and F-score of the classifiers can be found in Figures 8.1a and 8.1b, where they are compared to four state-of-the-art personalized filters: LC [9], NB-BOW [27], SVM-BOW [27], and ICRM [20]. Again, the first three classifiers of SENTINEL perform about equally well and surpass the rest. The filter that can claim to be a reasonably close second is the LC filter inspired by artificial immune systems [9]. For these two measures, the differences between SENTINEL's optimal classifier (BAGGED RF), and LC are significant at $\alpha = 0.05$.

The average FPR and FNR of SENTINEL can be found in Figures 8.1c and 8.1d, respectively. It comes as no surprise that BAGGED RF and BOOSTED RF perform the best—BAGGED RF misclassifies hams the least (FPR=2.1%, see Figure 8.1c) while BOOSTED RF misclassifies spams the least (FNR=0.7%, see Figure 8.1d). In addition, SENTINEL is compared to three personalized filters that are considered to be yardsticks: NB-TF [3], BAYESIAN [14], and NB-SLWE [28]. Except for the NB-TF filter, the three best classifiers of SENTINEL significantly outperform these yardsticks. Four of SENTINEL's classifiers (the exception is NB) surpass the FNR of NB-TF, but only BAGGED RF outperforms the FPR of NB-TF while our second best classifier—BOOSTED RF—closely ties with NB-TF; the FPR of BAGGED RF is lower than NB-TF but the difference is significant only at $\alpha = 0.10$.

We further investigated the FPR and FNR achieved by our classifiers on each of the six datasets (see Figure 8.2). Interestingly, RF exhibits strength similar to the meta-learners for spam classification (Figure 8.2b) but its ability to identify hams diminishes when more spams are included in its training data (Figure 8.2a)—specifically, for Enron 5 and Enron 6, even NB outperforms it. SVM's performance shows that this classifier's training data should be carefully selected—with spam skewed training data its ham misclassification rate is as high as 17%. This experiment also illustrates that the skewness of data has an effect on the training of anti-spam filters—except for the aberrant trend displayed by the NB classifier likely due to the presence of dependency among the attributes [23], the remaining classifiers misclassify fewer hams in Enron 1–3 (ham skewed) and fewer spams in Enron 4–6 (spam skewed).



(a) Ham misclassification rates (FPR) of the classifiers for the six datasets.

(b) Spam misclassification rates (FNR) of the classifiers for the six datasets.

Figure 8.2: Ham and Spam misclassification rates of the classifiers on the Enron-Spam collection.

8.5 Conclusions

In this paper we describe the development and evaluation of a prototype personalized anti-spam filter named SENTINEL. The filter uses natural language attributes, the majority being connected to stylometric aspects of writing. We evaluate the filter with a benchmark personalized email collection called Enron-Spam. Five well-known learning algorithms that induce binary classifiers using the real-valued natural language attributes of the emails are explored. The experimental outcomes show that SENTINEL performs best with two meta-learners: BOOSTED RF and BAGGED RF. As well, the performance of SENTINEL surpasses that of a number of state-of-the-art personalized filters proposed in previous studies. The evaluation therefore indicates that the filter can be utilized as a personalized anti-spam filter.

We still need to investigate several aspects of the filter viz. its real-time training and response latency. Moreover, its performance on *Concept Drift* can be observed by substituting the spams of Enron-Spam collection with the latest data. These investigations are left as future work.

Bibliography

[1] I. Androutsopoulos, G. Paliouras, and E. Michelakis, "Learning to filter unsolicited commercial e-mail (revised version)," Tech. Rep. 2, NCSR "Demokritos", 2004.

[2] Commtouch, "Internet threats trend report," tech. rep., Commtouch, USA, April 2013.

[3] V. Metsis, I. Androutsopoulos, and G. Paliouras, "Spam filtering with naive bayes – which naive bayes?," in *Third Conference on Email and Anti-Spam (CEAS 2006)*, (USA), 2006.

[4] Y. Hu, C. Guo, E. W. T. Ngai, M. Liu, and S. Chen, "A scalable intelligent non-content-based spam-filtering framework," *Expert Systems Applications*, vol. 37, pp. 8557–8565, Dec. 2010.

[5] C.-C. Lai and M.-C. Tsai, "An empirical performance comparison of machine learning methods for spam e-mail categorization," in *Fourth International Conference on Hybrid Intelligent Systems (HIS'04)*, (USA), pp. 44–48, IEEE Computer Society, 2004.

[6] A. Qaroush, I. M. Khater, and M. Washaha, "Identifying spam e-mail based-on statistical header features and sender behavior," in *CUBE International Information Technology Conference*, (USA), pp. 771–778, ACM, 2012.

[7] M. Ye, T. Tao, F.-J. Mai, and X.-H. Cheng, "A spam discrimination based on mail header feature and SVM," in *Fourth International Conference on Wireless Communications, Networking and Mobile Computing (WiCom08)*, pp. 1–4, 2008.

[8] Q. Ma, Z. Qin, F. Zhang, and Q. Liu, "Text spam neural network classification algorithm," in *2010 International Conference on Communications, Circuits and Systems*, (China), pp. 466–469, 2010.

[9] Y. Zhu and Y. Tan, "A local-concentration-based feature extraction approach for spam filtering," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 2, pp. 486–497, 2011.

[10] C. Orăsan and R. Krishnamurthy, "A corpus-based investigation of junk emails," in *Third International Conference on Language Resources and Evaluation (LREC-2002)*, (Spain), May, 29 – 30 2002.

[11] S. Afroz, M. Brennan, and R. Greenstadt, "Detecting hoaxes, frauds, and deception in writing style online," in *2012 IEEE Symposium on Security and Privacy*, (USA), pp. 461–475, 2012.

[12] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A Bayesian Approach to Filtering Junk E-Mail," in *Learning for Text Categorization: Papers from the 1998 Workshop*, (USA), AAAI Technical Report WS-98-05, 1998.

[13] P. Graham, "A plan for spam." Available on: http://paulgraham.com/spam.html, Aug. 2003.

[14] B. Issac, W. J. Jap, and J. H. Sutanto, "Improved bayesian anti-spam filter implementation and analysis on independent spam corpuses," in *2009 International Conference on Computer Engineering and Technology*, (USA), pp. 326–330, IEEE Computer Society, 2009.

[15] A. Kosmopoulos, G. Paliouras, and A. Androutsopoulos, "Adaptive spam filtering using only naive bayes text classifiers," in *Fifth Conference on Email and Anti-Spam (CEAS 2008)*, (USA), 2008.

[16] T. S. Guzella and W. M. Caminhas, "A review of machine learning approaches to spam filtering," *Expert Systems with Applications*, vol. 36, pp. 10206–10222, Sept. 2009.

[17] P. Haider, U. Brefeld, and T. Scheffer, "Supervised clustering of streaming data for email batch detection," in *24th International Conference on Machine Learning*, (USA), pp. 345–352, ACM, 2007.

[18] X. Carreras and L. Marquez, "Boosting Trees for Anti-Spam Email Filtering," in *RANLP-2001, 4th International Conference on Recent Advances in Natural Language Processing*, 2001.

[19] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer, 2001.

[20] A. Abi-Haidar and L. M. Rocha, "Adaptive spam detection inspired by a cross-regulation model of immune dynamics: A study of concept drift," in *Artificial Immune Systems*, pp. 36–47, Springer, 2008.

[21] A. Abi-Haidar and L. M. Rocha, "Adaptive spam detection inspired by the immune system," in *ALIFE*, pp. 1–8, 2008.

[22] I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, and C. D. Spyropoulos, "An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages," in *23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, (USA), pp. 160–167, ACM, 2000.

[23] R. Shams and R. Mercer, "Classifying spam emails using text and readability features," in *Proceedings of the 2013 IEEE International Conference on Data Mining (ICDM2013)*, (Texas, USA), IEEE, 2013.

[24] A. Liaw and M. Wiener, "Classification and Regression by randomForest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.

[25] R. E. Schapire, "A brief introduction to boosting," in *16th International Joint Conference on Artificial Intelligence (IJCAI'99)*, (USA), pp. 1401–1406, Morgan Kaufmann Publishers Inc., 1999.

[26] L. Breiman and L. Breiman, "Bagging predictors," in *Machine Learning*, pp. 123–140, 1996.

[27] M. Razmara, A. Razmara, and M. Narouei, "Textual spam detection: An iterative pattern mining approach," *World Applied Sciences Journal*, vol. 20, no. 2, pp. 198–204, 2012.

[28] J. Zhan, B. J. Oommen, and J. Crisostomo, "Anomaly detection in dynamic systems using weak estimators," *ACM Trans. Internet Technol.*, vol. 11, pp. 3:1–3:16, July 2011.

Chapter 9

Supervised Classification of Spam E-mails with Natural Language Stylometry

This chapter is based on the paper titled "Supervised Classification of Spam E-mails with Natural Language Stylometry" co-authored with Robert E. Mercer that has been submitted to the Elsevier Expert Systems with Application. The journal paper is under review. This paper combines and extends the ideas presented in Chapter 7 and 8.

In this paper, we report the development and evaluation of SENTINEL—an anti-spam filter based on natural-language and stylometry attributes. The performance of the filter is evaluated not only on non-personalized emails (i.e., emails collected randomly) but also on personalized emails (i.e., emails collected from particular individuals). Among the non-personalized datasets are CSDMC2010, SpamAssassin, and LingSpam, while the Enron-Spam collection comprises personalized emails. The proposed filter extracts natural-language attributes from email text that are closely related to writer stylometry and generate classifiers using multiple learning algorithms. Experimental outcomes show that classifiers generated by meta-learning algorithms like ADABOOSTM1 and BAGGING are the best, performing equally well and surpassing the performance of a number of filters proposed in previous studies, while a Random Forest (RF) generated classifier is a close second. On the other hand, the performance of classifiers using Support Vector Machine (SVM) and Naïve Bayes (NB) are not satisfactory. In addition, we find much improved results on personalized emails and mixed results on non-personalized emails.

9.1 Introduction

Spam emails are both unsolicited and massively posted commercial and non-commercial emails. The effects of spams include but are not limited to the loss of individual and organizational productivity, chaotic user inboxes, Internet speed degradation, and misappropriation of personal information. The onset of spams has grown exponentially. In March 2013, for instance, approximately 100 billion spams were received by users everyday which is 98% more than that from the previous quarter [1]. To date, most anti-spam filters are supervised tools that trap spam emails by detecting some mundane patterns learned during their training [2, 3]. Usually, these patterns are generated from attributes collected either from email headers [4, 5]

or from email text [6]. The filters generate classifiers from these attributes using algorithms like Naïve Bayes (NB) [4, 6, 7], Random Forest (RF) [8], Support Vector Machine (SVM) [9] and Neural Networks (NN) [10].

Spammers almost always introduce new techniques to bypass header-based anti-spam filters by originating their menacing emails from *white-list* sources. Spoofing text-based filters, on the other hand, requires substantial efforts by the spammers. For example, one of the easiest ways to detect spams is to search for *spam terms* in the email text. Spammers then simply change their vocabulary to circumvent these text-based filters, as pointed out by [11]. The authors reported that because of the cleverly removed spam terms, text-based filters tend to underperform over time. As a result, organization servers and personal inboxes are still overwhelmed by spam emails and therefore classifying spam emails remains a challenging machine-learning task.

[6] and [10] show that natural-language attributes of email subject and body have substantial ability to discern spams and hams. These attributes are based on the importance of a term in the email (i.e., term frequency or TF), its rarity in the email dataset (i.e., inverse document frequency or IDF), and their normalized form (TF·IDF). The advantage of exploiting these attributes is that they perform well on personalized filters (see [6, 12]). However, they are found to underperform on non-personalized emails [7]. Furthermore, the calculation of TF·IDF is done on the *word-count space*, so for each newly arrived email, this attribute needs to be re-calculated. The re-calculation, if not done incrementally, can introduce latency. The time-sensitivity of the data described earlier is also a reason not to rely on these *term-based* attributes. Therefore, along with these, we need to exploit other natural-language attributes that are more pervasive in nature. Among such pervasive attributes are those that are related to the writer's writing style. [13] and [14] showed that using writer stylometry and choice of words, authorship of documents can be identified. This finding can be aligned with the problem in our hand nicely because often, spammers write bulk emails to people by impersonating as a bank manager, business partner, etc. which they are not. These attributes present in the stylometry of emails are still unexplored in our problem domain. Among the stylometry attributes are text readability, grammar and spelling mistakes, the use of function and content words.

In this paper, we report the development and evaluation of an anti-spam filter named SENTINEL on both non-personalized and personalized emails. The bulk of the attributes used by SENTINEL are natural-language attributes related to writer stylometry. Three standard, non-personalized email datasets CSDMC2010, SpamAssassin and LingSpam, and six personalized datasets in the Enron-Spam collection are used to train and test the filter. SENTINEL generates classifiers using Random Forest (RF), Support Vector Machine (SVM), and Naïve Bayes (NB), and two meta-algorithms called ADABOOSTM1 and Bootstrap Aggregating (BAGGING) using RF as the base classifier. Results show that ADABOOSTM1 and BAGGING perform almost equally the best while the performance of RF is a close second. Interestingly, the performance of SVM depends on the quantity of spams in the training set. NB has the poorest results of all—which is understandable as most of the attributes are not independent. Comparisons show that the performance of SENTINEL surpasses that of a number of filters proposed in previous studies. We also find that the results on personalized emails are better to some extent compared to those on non-personalized emails. Because writer stylometry is language-independent, SENTINEL may be an excellent means to classify spam emails in any language.

The paper is organized as follows: in Section 9.2, we discuss some related work in the

domain. Section 9.3 outlines the attributes, learning algorithms, evaluation measures, experimental procedure, and describes the datasets. Results are discussed in Section 9.4. Finally, Section 9.5 concludes the paper with directions to possible future work.

9.2 Related Work

As we are interested to see how our filter performs on both non-personalized and personalized emails, in this section we describe work aimed at classifying emails from non-personalized and personalized email datasets.

9.2.1 Non-personalized Filters

Non-personalized filters are developed to safeguard an email server from spam emails. These filters are so called because their development and evaluation is done using randomly collected spam and ham emails. The spam emails are generally collected from multiple spam-traps while the ham emails are contributed by different individuals and thus do not characterize any particular user. Among the popular non-personalized datasets are CSDMC2010, SpamAssassin, and LingSpam.

[8] investigated the performance of their anti-spam filter on the CSDMC2010 dataset. Their filter is capable of generating multiple spam classifiers but unlike our approach, their predictive models are based on traditional attributes found in email headers. Among the classifiers, they found that the one generated using RF outperforms the rest namely NB, Bayesian Network, and SVM. The authors, like many, conceived the spam classification performance as a cost-sensitive analysis and used both conventional and cost-sensitive evaluation measures. Likewise, [15] developed two filters—one of which is based on SVM while the other is generated using NB . The filters extract attributes from email content using a novel attribute selection method based on a binomial distribution hypothesis testing called *Bi-Test*. When they compared the filter performances on the CSDMC2010 and LingSpam datasets, they found that the filters perform better on the former.

[10] developed an NN-based filter that uses 328 text attributes. On the SpamAssassin dataset, their reported accuracy of 0.920 was reasonably good. The filter developed by [16] also uses an NN classifier. The attributes chosen for their experiment are related to artificial immune systems (AIS). The filter has been reported to be accurate about 92% of the time on the SpamAssassin dataset. Prior to that, [17] measured their filter performance on the SpamAssassin dataset with a cost-sensitive evaluation. They found much improved results using data compression techniques.

From our literature survey, we found that the reported performances of filters are relatively low on the LingSpam dataset. [18], for instance, applied K-NNC and a data clustering algorithm called BIRCH on this dataset. Their TF·IDF based filter, compared to the filters tested on other datasets, achieved relatively low scores. [19] also found similar results on LingSpam when they experimented with four popular content-based anti-spam filters in the domain.

9.2.2 Personalized Filters

Unlike non-personalized filters, personalized anti-spam filters are aimed at protecting particular individuals. These filters learn through the choice of idioms and phrases, writing style, variation of sentences, etc of the legitimate users. Therefore, to train and test these supervised filters, personalized email data like those in the Enron-Spam collection are preferred.

The earliest of the personalized anti-spam filters were simple and computationally efficient as they used a simple NB classifier. These filters exploited the simple Bayesian framework, a set of rules, and both header and content attributes [20]. Likewise, they attained low misclassification rates on several datasets. The initial success of these filters led several others to emerge and they simply replaced the hand-crafted rules with predictive models (see, for example, [6, 7, 21, 22]). Several variations of these filters include *multivariate Bernoulli, Bag of Words (BoW), multinomial boolean*, etc. [6], for instance, used 3000 multinomial boolean TF attributes—the results reported by this study were highly impressive. Two years later, they achieved even better results by using the *transformed* TF attributes [23]. Nowadays, the performance of an NB filter is considered to be the *de facto* standard to compare newly developed personalized filters.

SVM filters are efficient for training in much the same way as NB filters, nevertheless it was found by [24] that the filters need incremental training to reduce latency. Furthermore, SVM filters can handle large attribute sets and in many cases, attribute selection is not necessary [9]. Most of the benchmark SVM filters use the frequency-based *linear kernel*. As well, there are several SVM filters that use updatable supervised clustering algorithms like the one reported by [25]. However, despite the aforementioned pros of using SVM-based filters, the trade-off is high misclassification rate, especially for personalized emails.

The advantages of using *meta-learning* anti-spam filters are manifold. Firstly, when a base learner with a sufficient tree depth is used, they achieve improved misclassification rates on many public datasets [26]. Secondly, these filters are resistant to the problem of overfitting and therefore they gain more *appropriate* accuracy even on an unbalanced dataset[1] [27]. However, these filters have the weakness of *ensemble* learning—the interpretation of results is difficult. Studies show that meta-learning filters outperform many decision tree, NB, and SVM filters. Surprisingly, the use of meta-learning filters is still not as widespread as NB and SVM filters.

Over the last decade, Artificial Immune System-based anti-spam filters have become a popular choice. These filters use *detectors* on the email for pattern matching. Detectors are in fact regular expressions that are defined *a priori*. Each detector is given a *weight* that is adjusted as the filters recognize a pattern in a given email. The weights of the matching detectors are then used (usually combined) to determine the email's class label. Notable immune system-based filters are reported by [12, 28, 29]. These filters seek specific *signatures* in the emails—this is why they are widely used in personalized email classification where similar patterns can be found in the writing style of the person the filter intends to protect. Also, many of these filters are able to deal with *Concept Drift*—the gradual or abrupt change of thematic context over time such as new advertisement themes in spam emails.

[1] Most of the public email datasets are unbalanced [24].

Category	Quantity	Description
Word-level Attributes	11	Spam words, Alpha-numeric words, Function words, Verbs, TF ([6, 23]), TF·ISF, TF·IDF.
Error Attributes	3	Grammar and spelling mistakes.
Readability Attributes	23	Simple and complex words, and their TF·IDF, Fog index, Simple and Inverse Fog index, Smog index, Flesch reading ease score, Forcast, Flesch-Kincaid score, Email length, Word length.
HTML Attributes	3	regular and anchor tags.

Table 9.1: The brief summary of the attributes used in our study: their categories, quantity, and description.

9.3 Methods and Materials

9.3.1 Attribute Selection

Each email in our experiment is represented as $(\vec{x}, y)$, where $\vec{x} \in \mathbb{R}^n$ is a vector of n attributes and $y \in \{spam, ham\}$ is the label of the email. In our study for the LingSpam dataset, we explored 36 attributes with one class attribute and therefore, $n = 37$. For the CSDMC2010 and SpamAssassin datasets, however, we considered three more attributes related to HTML tags because the datasets contain them since they are not pre-processed (Table 9.4). Therefore, for these two datasets, $n = 40$. Finally, for the Enron-Spam collection, in addition to the 37 attributes we used for the LingSpam dataset, we further explored a term frequency (TF) attribute previously utilized by [6, 23]. Therefore, in our study for the Enron-Spam collection, $n = 37$. Table 9.1 summarizes the attributes used in our experiment.

Word-level Attributes

We treated each email as a bag of words to calculate the word-level attributes. We curated a dictionary that comprises 381 spam words. Using this dictionary, we counted the frequency of spam words in the emails. This attribute is inspired by the interesting findings of [22] who showed that merely finding the word *click* in the emails can detect 79.7% of spam emails in a dataset with only 1.2% ham misclassification rate. Our other attributes in this category include the frequency of alpha-numeric words, verbs, and function words. To identify the verbs in the emails, we used the Stanford POS Tagger[2]. To identify function words, we utilized the *stoplist function* of a topic indexer named *Maui*[3].

In this category of attributes, we also included the term frequency (TF) attribute used by at least two NB filters developed by [6, 23]. For each term which appears in at least four training emails, *information gain* scores are computed according to the study conducted by [30]. Of all the terms in the dataset, the 3000 with the highest scores are considered as the TF *vector* for all the emails. Thereafter, should any of these 3000 terms be found in an email, the term's

[2]Downloadable at `http://nlp.stanford.edu/software/tagger.shtml`
[3]Downloadable at `https://code.google.com/p/maui-indexer/`

corresponding frequency value in the TF vector is first incremented and then normalized. Finally, the normalized value is used to calculate the probability score of the email according to Bayes' theorem. The overall method is well documented by [23]. Of note, the TF attributes work well on personalized emails according to [6] and are less important for non-personalized emails. Therefore, in this study, this attribute is only extracted from the personalized emails in the Enron-Spam collection.

Furthermore, we computed the TF·ISF of each email. Here, the term frequency (TF) is the commonly used *square root* of the frequency of a term t in a sentence in an email M while the inverse sentence frequency (ISF) is a relative measure of whether t is common or rare in the other sentences in M. This attribute is meant to reflect how important each t is to an M. In addition, the measure controls for the fact that in an email, some terms are generally more common than others. We also used the TF·IDF of each email as an attribute. It is a numerical statistic that reflects how important a term t is to an email M in the dataset D to which M belongs. The TF·IDF value increases proportionally to the number of times t appears in the email M, but is offset by the frequency of t in the dataset D. The definition of TF is the same above while inverse document frequency (IDF) measures whether a given term t is common or rare in the remaining emails in the dataset D.

Error Attributes

Our next set of attributes is related to the grammar and spelling errors present in the emails. For each email, we simply counted the frequency of grammar and spelling errors using a Java API called LanguageTool[4]. By summing up the values of these two attributes, we introduced a third attribute in this category named *Language Errors*. It is to be noted that these attributes are normalized separately for spams and hams. A traditional attribute normalization without considering the class labels of the instances is not followed because it is expected that the difference between the values of the attributes for spams and hams is large.

Readability Attributes

Readability is a measure of the difficulty of reading a sentence, a paragraph or a document. At the heart of readability lies the notion of simple and complex words. Simple words are those that have at most two syllables while complex words contain three or more syllables. Since both types of words have a significant contribution for text readability, we have included both in our attributes. Five standard scores use these two types of words to determine the readability of a given text: Fog index, Smog index, Flesch reading ease score, Forcast, and Flesch-Kincaid index. For each email, all five readability scores are computed and used as attributes. Also, among the scores, *Fog index* measures the relative use of complex words in a document. We modified the Fog index formula to measure the relative use of simple words in a document, and considered this as an attribute, too. Furthermore, we considered the arithmetic inverse of Fog index as another attribute. The other attributes in this category include email length (i.e., total sentences in an email), average word length (i.e., total syllables over total terms in an email), and TF·IDF of the set of simple and the set of complex words.

[4]Available at: `http://www.languagetool.org/java-api/`

Learning Algorithms	Parameters	
Random Forest	Maximum Depth: Unlimited	Number of Trees to be Generated: 10
	Random Seed: 1	
Boosted Random Forest	Number of Iterations: 10	Random Seed: 1
	Resampling: False	Weight Threshold: 100
Bagged Random Forest	Size of Bag (%): 100	Out of Bag Error: False
	Number of Iterations: 10	Random Seed: 1
Support Vector Machine	SVM Type: C-SVC	Cost: 1.0
	Degree of Kernel: 3	EPS: 0.0010
	Gamma: 0.0	Kernel Type: Radial Basis
	Epsilon: 0.1	Probability Estimates: False
	Shrinking Heuristics: True	
Naïve Bayes	Use of Kernel Estimator: False	

Table 9.2: Parameter setup for the learning algorithms used in this experiment to generate classifiers.

HTML Attributes

[6] suggested not to use the tracking of HTML tags to classify emails because examining a phishing `url` can sometimes lead to unfortunate results, such as the user inbox becoming compromised by spammers. Therefore, instead of tracking the `url`s in the HTML tags, we are interested in counting them. We exploit attributes related to HTML tags only for the CSDMC2010 dataset as out of the four datasets only this one contains these tags. We extracted three HTML based attributes from the emails of the CSDMC2010 dataset: (i) frequency of anchor tags (i.e., the number of close-ended tags <a> and), (ii) frequency of tags that are not anchors (e.g., <p> or
), and (iii) total HTML tags in the emails (e.g., sum of (i) and (ii)). Each of these features is normalized by the length of the email, N, which is the number of sentences in the message. To identify HTML tags in the emails, we used a Java HTML parser called *jsoup*[5].

Note that, the aforementioned attributes are computed by both including and excluding the function words in the emails—the exceptions being the frequency of function words, TF, email length, and TF·IDF attributes. Analyzing the real valued attributes, we have found that their distribution is either left-tailed or right-tailed. This skewness of attribute values, which leads to poor classifiers, has been eliminated by using a *logarithmic transformation* of all the attribute values. Once log-transformed, the distribution of the attributes becomes normal. Lastly, for each attribute, except the error attributes described in Section 9.3.1, we perform attribute normalization; the resulting normalized values are therefore in $[0, 1]$.

9.3.2 Learning Algorithms

As found in the recent research of [4, 8], RF is able to produce highly accurate spam classifiers. Overall, RF classifiers also have the reputation for being fast and efficient with large data [31]. On the other hand, although it generates complex models, SVM is a popular choice for anti-

[5]Downloadable at `http://jsoup.org/download`

spam filters. The algorithm has notable performances with header features [4, 7–9]. NB is also a widely used learning algorithm for anti-spam filters. It is simple yet provides powerful spam detectors [6, 8]. In addition, on many occasions, NB using simple attributes like TF·IDF has even outperformed quality learning algorithms, like SVM. Unlike the others, ADABOOSTM1 and BAGGING are meta-learners that improve a given *weak* learning algorithm most of the time [32,33]. In the following experiments, we have considered RF as the weak learner for these two algorithms. Both ADABOOSTM1 and BAGGING are simple, fast, and above all, are less susceptible to overfitting the training data. And finally, their performances are better than algorithms like NB and *probabilistic* TF·IDF for text categorization tasks.

The parameters that we set for the algorithms are described in Table 9.2.

9.3.3 Experimental Procedure

Treating each dataset independently, SENTINEL extracts the real-valued attributes from each email using its text processing unit. Using a conventional stratified 10-fold cross-validation approach, the filter then generates for each dataset five classifiers using the five algorithms described in Section 9.3.2. The classifiers are then evaluated. In a K-fold cross-validation, the original dataset is randomly partitioned into K equal-sized folds or subsets. Then each classifier is trained on $K - 1$ folds and evaluated on the remaining fold. Stratification means that the class (i.e., ham or spam) in each fold is represented in approximately the same proportion as in the full dataset. The cross-validation process is then repeated until each of the K folds is used exactly once as the validation data. The final evaluation measures of the classifiers are the averages of the K evaluation measures from the folds. These evaluation measures are described next.

9.3.4 Evaluation Measures

To evaluate anti-spam filters, a significant number of previous works rely on measures like precision, recall, F-score, and accuracy [24]. The reporting of these measures is done according to the confusion matrix given in Table 9.3. The measures are explained below.

Precision is the fraction of spam predictions that are correct and can be written as follows:

$$Precision = \frac{n_{s\to s}}{n_{s\to s} + n_{h\to s}}.$$

On the other hand, recall—also known as *spam recall*—examines the fraction of spam emails being retrieved:

$$Recall = \frac{n_{s\to s}}{n_{s\to s} + n_{s\to h}}.$$

F-score (FM), simply, is the harmonic mean of precision and recall and can be calculated as follows:

$$F - score = \frac{2 \times \text{PREC} \times \text{REC}}{\text{PREC} + \text{REC}}.$$

Accuracy, on the other hand, is the percentage of correctly identified spams and hams:

$$Accuracy = \frac{n_{h\to h} + n_{s\to s}}{n_{h\to h} + n_{h\to s} + n_{s\to h} + n_{s\to s}}.$$

		Actual	
		Spam	Ham
Prediction	Spam	$n_{s\to s}$	$n_{h\to s}$
	Ham	$n_{s\to h}$	$n_{h\to h}$

Table 9.3: Confusion matrix for the spam classification problem.

Some of the aforementioned measures, however, can be seriously flawed when working with datasets that have class imbalance problems. For instance, accuracy does not distribute weights rationally to the majority and minority class rather it places more weight on the majority class than on minority class. This makes it difficult for a classifier to perform well on the minority class. Moreover, email misclassification can be cost-sensitive considering that the users might accept some spams to enter into their inbox but that they prefer their hams not end up in the spam-traps. To overcome the problem with measures that cannot deal with the class imbalance problem (e.g., accuracy) ham misclassification rate (FPR) and spam misclassification rate (FNR) are also being used.

The ham misclassification (false positive) rate denotes the fraction of ham emails classified as spams:

$$\text{FPR} = \frac{n_{h\to s}}{n_{h\to s} + n_{h\to h}}.$$

In contrast, the spam misclassification (false negative) rate is the fraction of spams delivered to the user inbox:

$$\text{FNR} = \frac{n_{s\to h}}{n_{s\to h} + n_{s\to s}}.$$

The viable alternative to FPR and FNR is to report the *Area Under the Curve* (hereinafter, AUC), measured using the *Receiver Operating Characteristics (ROC)* curves. The ROC curve is a 2D graph whose *Y-axis* represents $1 - \text{FNR}$ and whose *X-axis* represents FPR, thereby depicting the compromises between the cost of $n_{s\to h}$ and $n_{h\to s}$.

Besides the aforementioned evaluation measures, we also used *cost curves* to report the classification performance. The cost curves, indeed, are the projection of the slopes in the ROC curves on the *X-axis* while the *Y-axis* is the expected misclassification cost. These curves provide visualizations of 2-class classifiers over the full range of possible class distributions and misclassification costs. That is, for skewed datasets, using these curves we can even say how good a given classifier would perform if the class distributions were equal—50% of the emails are spams and 50% of the emails are hams.

The details about the cost curves are described by [34, 35].

9.3.5 Datasets

For decades, several email datasets, viz., SpamAssassin, CSDMC2010, LingSpam and Enron-Spam, have been used to gauge the performance of anti-spam filters. The first three datasets are composed of randomly collected spam and ham emails over a given time period and therefore are suitable for developing and testing non-personalized anti-spam filters. Enron-Spam, on the other hand, is a collection of emails composed of six datasets each containing ham emails from a single person.

Dataset	Ham:Spam	Text Pre-processed?	Year of Curation
CSDMC2010	2949 : 1338	No	2010
SpamAssassin	4149 : 1884	No	2002
LingSpam	2414 : 481	Yes	2000

Table 9.4: Brief description of the non-personalized email datasets: ham-spam ratio, whether the texts are pre-processed, and the year of curation.

Ham + Spam	Ham:Spam	Time-stamp	
Enron 1 (`farmer-d` + GP)	3672 : 1500	[12/99, 1/02]	[12/03, 9/05]
Enron 2 (`kaminski-v` + SH)	4361 : 1496	[12/99, 5/01]	[5/01, 7/05]
Enron 3 (`kitchen-l` + BG)	4012 : 1500	[2/01, 2/02]	[8/04, 7/05]
Enron 4 (`williams-w3` + GP)	1500 : 4500	[4/01, 2/02]	[12/03, 9/05]
Enron 5 (`beck-s` + SH)	1500 : 3675	[1/00, 5/01]	[5/01, 7/05]
Enron 6 (`lokay-m` + BG)	1500 : 4500	[6/00, 3/02]	[8/04, 7/05]

Table 9.5: Brief descriptions of the Enron-Spam collection as described by [6]: composition of the datasets, ham-spam ratio, and time-stamp.

CSDMC2010, among the four, is the latest collection of emails and has not been used much. The spam rate of this dataset is reasonable—about 32%. In contrast, SpamAssassin is one of the most popular datasets. The spam rate of this dataset is almost equal to that of the CSDMC2010 dataset. The LingSpam dataset is both the smallest and oldest dataset that we consider. Its spam rate is also smaller than the others—only about 17%. It is the odd one out of the three non-personalized datasets because the hams in this dataset are collected from the discussions of a linguistics forum; the spams, on the other hand, are collected randomly. Table 9.4 summarizes these datasets.

The ham collections of Enron-Spam are named: `farmer-d`, `kaminski-v`, `kitchen-l`, `williams-w3`, `beck-s`, and `lokay-m`. The spams are collected from three different sources. First, a mix of spams that are collected from the SpamAssassin corpus and spam traps of the Honeypot project[6] are put together; these spams are dubbed as SH. Second, BG spams are collected from the spam traps of Bruce Guenter[7]. Third, spams are collected randomly from the mailbox of Georgios Paliouras (GP) [6]. The foregoing six ham email collections are each paired with one of these three spam collections (SH, BG, and GP). Thereafter, the six collections are dubbed as Enron 1 to Enron 6. Of the six collections, Enron 1–3 are ham skewed (ham:spam is 3:1) while Enron 4–6 are spam skewed (ham:spam is 1:3). The summary of the characteristics of the Enron-Spam collection can be found in Table 9.5.

During the development of the Enron-Spam dataset, [6] noticed that spam emails have some attributes that can too easily distinguish them from hams and therefore they should be pre-processed. To remove these attributes, the following pre-processing steps are considered. First, the SUBJECT field of spams can contain symbols such as $ or !. As well, they contain spam

[6]Consult with `http://www.projecthoneypot.org`

[7]Overview at `http://untroubled.org/spam`

words such as *porn*, *webcam*, or *lottery*. Therefore, such entries are excluded from the SUBJECT fields of the emails. Second, many emails in the datasets can contain an ATTACHMENT field. If it exists, this extraneous field is removed from an email. Third, non-ASCII characters in the email text are removed since the values of our natural language attributes are affected by their presence.

The reasons for choosing these datasets are manifold. Firstly, they contain emails sent out between 2000 and 2010. This provides an interesting test-bed that characterizes the change of language of both spams and hams spanning across a decade. Secondly, we include spam-skewed datasets in our experiment (i.e., Enron 4-6) because previously, many works reported that although an anti-spam filter can do well on ham classification using ham-skewed datasets like CSDMC2010, SpamAssassin, LingSpam, and Enron 1-3, its performance on spam classification can be seriously flawed (see, for example, [17]). Thirdly, we include LingSpam in our dataset because not only are its hams domain-specific but the hams are also excerpts of scholarly discussions on linguistics. We believe that the results of our stylometric approach with this dataset will be interesting to the anti-spam community. Fourthly, we include datasets that are popular and reported by many (SpamAssassin and LingSpam) as well as those that are not explored by as many (CSDMC2010 and Enron-Spam). Last but not least, it is one of our goals to investigate the performance of SENTINEL on both non-personalized and personalized emails.

9.4 Results and Discussions

9.4.1 Performance on Non-personalized Emails

In this section, we report the traditional as well as cost-sensitive evaluation of SENTINEL on the non-personalized email data. The most remarkable result to emerge from the data is that for all datasets, BAGGING generates classifiers that have the lowest ham misclassification rates (FPR) while the classifiers generated by ADABOOSTM1 have the lowest spam misclassification rates (FNR). These results can be found in Table 9.6. Given the results, it is difficult to decide which classifier is better. A good way to report the best-performing classifier is to refer to a balanced measure of the two misclassification rates: the AUC. As can be seen in Figures 9.1 and 9.3, of the two, the BAGGED RF classifiers have the better AUC. Table 9.6 also summarizes that of all the datasets, SENTINEL misses hams the least (FPR= 1%) on the LingSpam dataset. On the other hand, the lowest FNR achieved by the filter is about 7% on the SpamAssassin dataset. Besides, the results with SVM and NB are below expectations. The reasons to underperform

Classifiers	FPR	FNR
RF	0.040	0.092
ADABOOSTM1	0.030	0.089
BAGGING	0.021	0.107
SVM	0.028	0.390
NB	0.101	0.396

(a)

Classifiers	FPR	FNR
RF	0.035	0.093
ADABOOSTM1	0.027	0.079
BAGGING	0.023	0.099
SVM	0.052	0.292
NB	0.104	0.558

(b)

Classifiers	FPR	FNR
RF	0.018	0.162
ADABOOSTM1	0.017	0.162
BAGGING	0.010	0.193
SVM	0.014	0.341
NB	0.219	0.277

(c)

Table 9.6: Ham misclassification rates (FPR) and spam misclassification rates (FNR) of SENTINEL on (a) CSDMC2010, (b) SpamAssassin, and (c) LingSpam Dataset.

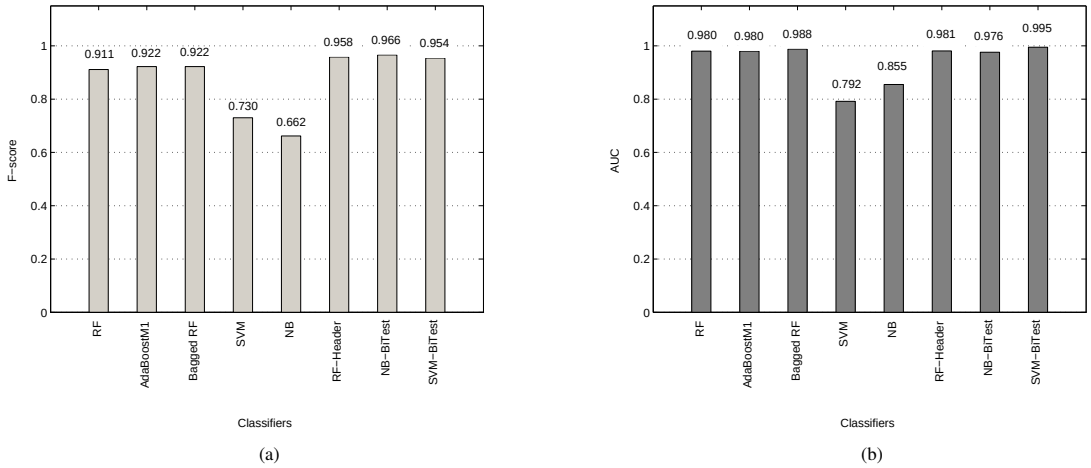


Figure 9.1: Comparison of (a) F-score and (b) AUC of the classifiers generated by SENTINEL and classifiers from previous studies on the CSDMC2010 dataset.

for a standard algorithm like SVM are further investigated in Section 9.4.2. In our analysis we found that the attributes are highly correlated which contradicts the naive assumption of the NB alogrithm. This apparent dependency among the attributes can be attributed to the algorithm's poor performance.

The F-score and AUC of the five classifiers of SENTINEL on CSDMC2010 dataset are compared to that found by three recently-proposed, cutting-edge filters: RF-HEADER [8], NB-BITEST, and SVM-BITEST [15]. Unexpectedly, SENTINEL is outperformed by all three filters in terms of F-score as shown in Figure 9.1a. Also, a *student t-test* with $\alpha = 0.05$ confirms that the differences are statistically significant. A possible explanation for this can be found in Table 9.6: the spam recall (1−FNR) of SENTINEL's classifiers are much higher than that achieved by the filters. However, when it comes to cost-sensitive analysis, SENTINEL performs more ideally—its BAGGED RF classifier's AUC outperforms RF-HEADER and NB-BITEST; SVM-BITEST with a reported AUC of 0.995 is the only exception. These comparisons are summarized in Figure 9.1b.

Similarly, the accuracies of the classifiers of SENTINEL, and two standard filters named TEXT-NN [10] and AIS-NN [16] are compared. Figure 9.2 shows that all of our ensemble-classifier accuracies are better than TEXT-NN and AIS-NN. The differences found in the accuracies are significant at $\alpha = 0.05$.

Finally, for the LingSpam dataset, the F-scores of SENTINEL are compared with two more filters called BIRCH [18] and NB-BITEST [15]. We found that all the F-scores of SENTINEL's classifiers, except that induced by NB, are better than the BIRCH filter. On the other hand, the F-score of NB-BITEST, however, is better than SENTINEL but only at $\alpha = 0.10$. The comparison is found in Figure 9.3a. In a similar way, the AUC of SENTINEL is compared with that reported by PPM [19] and NB-BITEST [15]. As shown in Figure 9.3b PPM performs the best in terms of AUC and the ensemble classifiers of SENTINEL are close seconds. Again, the differences are statistically significant at $\alpha = 0.05$.

The cost curves of the classifiers generated by SENTINEL for the three non-personalized datasets are shown in Figures 9.4, 9.5, and 9.6. The cost curves are a good means not only to explore the expected cost associated with the classifiers but also their performance for dif-

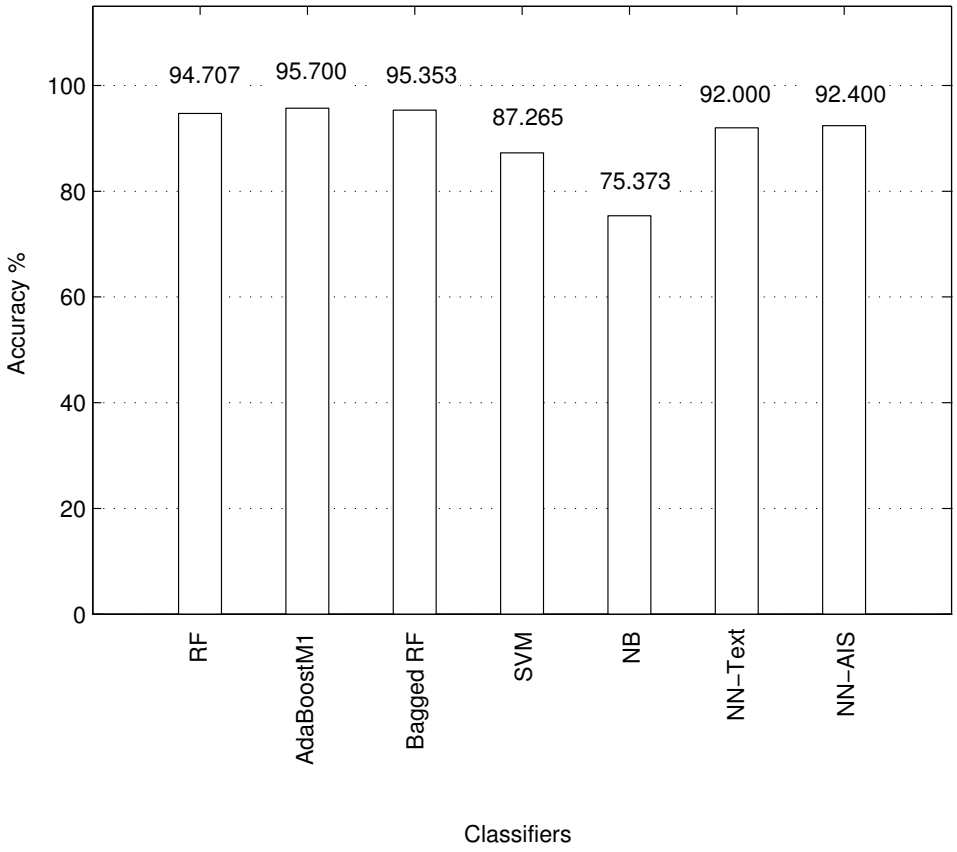


Figure 9.2: Comparison of accuracy of the classifiers generated by SENTINEL and classifiers from previous studies on the SpamAssassin dataset.

ferent ham-spam ratios in the datasets. Before we discuss further, please note that the *X-axis* of the curves denotes the *probability cost*, i.e., $x = 0.5$ illustrates the situation where the numbers of spams and hams in the dataset are equal and *trivial classifiers* are those that identify all emails either as hams or spams.

As anticipated, the poor performances of the SVM and NB classifiers are captured by the cost curves, too. For any ratio of ham and spam emails, the normalized expected costs associated with these two classifiers are very high (Figures 9.4, 9.5, and 9.6). According to the cost curves, the remaining classifiers perform much better for all ratios of ham and spam emails. However, we observe differences in their performances for different datasets. For instance, the ADABOOSTM1 classifier is expected to perform better than the BAGGED RF and RF classifiers if the majority of the emails are spams in the CSDMC2010 and SpamAssassin datasets (i.e., when *Probability Cost* > 0.5 in Figures 9.4 and 9.5). Interestingly, had these two datasets contained more spams, we could have expected that RF would have outperformed BAGGING. This observation is confirmed according to the Figures 9.4 and 9.5, where RF starts to perform better than BAGGING at some point after *Probability Cost* > 0.5. On the other hand, for the LingSpam dataset, the RF classifier can be expected, in most of the cases, to outperform both

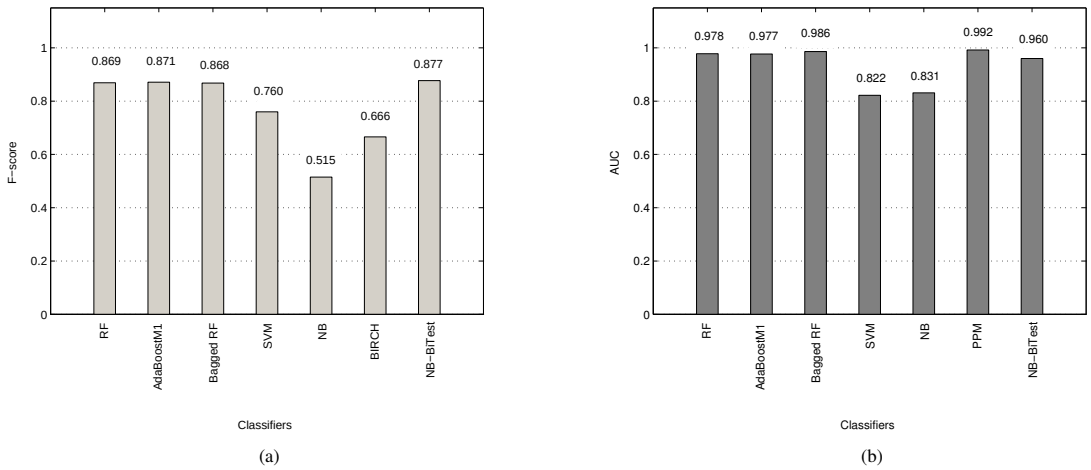


Figure 9.3: Comparison of (a) F-score and (b) AUC of the classifiers generated by SENTINEL and classifiers from previous studies on the LingSpam dataset.

the BAGGED RF and ADABOOSTM1 classifiers regardless of the skewness in the dataset (Figure 9.6).

9.4.2 Performance on Personalized Emails

Traditionally, the studies on Enron-Spam report the performances of their filters by averaging the scores described in Section 9.3.4 across the six datasets [6]. In doing so, they almost always have used the *arithmetic mean*. However, our tests reveal that like other cutting-edge filters [6,23], our proposed filters perform completely opposite for Enron 1-3 (hams are missed less often because more hams are in the training data) and Enron 4-6 (spams are missed less often because more spams are in the training data). These extreme trends in the performances are exhibited in Figure 9.7 where the reported values for the six datasets largely vary. Such extreme points affect the overall average if it is calculated using the arithmetic mean. The viable alternative for the averaging is then the *harmonic mean* which reduces the effect of outliers on the average. That said, in this section, we report the *harmonic mean* of the results found from the six datasets.

The accuracy and F-score of the classifiers can be found in Figures 9.8a and 9.8b, respectively. It can be noted that the values reported here for personalized emails are significantly better than the values reported in Section 9.4.1 for non-personalized emails. We also compared the accuracy and F-score to that found by four benchmark personalized email filters: LC [12], NB-BOW [36], SVM-BOW [36], and ICRM [28]. The two ensemble classifiers of SENTINEL perform about equally well and surpass the performances of the rest. The filter that can claim to be a reasonably close second is the LC filter inspired by artificial immune systems [12]. The differences between SENTINEL's optimal classifier (BAGGED RF), and LC are significant at $\alpha = 0.05$.

The average email misclassification rates of SENTINEL can be found in Figures 9.8c and 9.8d, respectively. BAGGING and ADABOOSTM1 perform the best—BAGGING misclassifies hams the least (FPR=2.1%, see Figure 9.8c) while ADABOOSTM1 misclassifies spams the least (FNR=0.7%, see Figure 9.8d). In addition, SENTINEL is compared to personalized filters that are used as yardsticks in the domain: NB-TF [6], BAYESIAN [21], and NB-SLWE [37]. Except for the NB-TF filter, the two

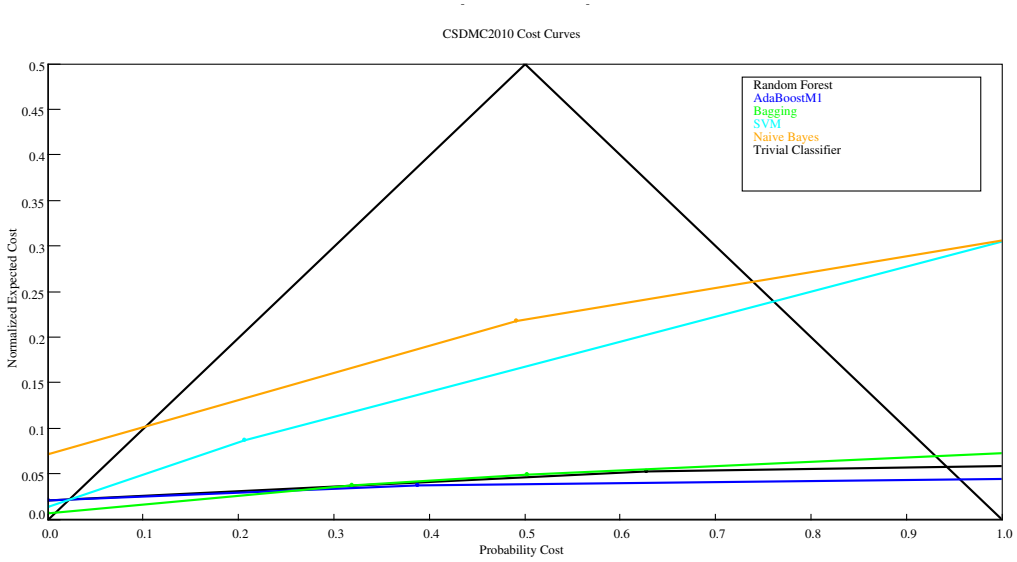


Figure 9.4: Cost curves of the five classifiers generated by SENTINEL on the CSDMC2010 dataset.

ensemble classifiers of SENTINEL outperform the rest. Four of SENTINEL's classifiers (except NB) surpassed the FNR of NB-TF but the FPR of our second best classifier—ADABOOSTM1—ties with NB-TF; the FPR of BAGGING is higher than NB-TF—the difference, however, is statistically significant only at $\alpha = 0.10$.

Further tests on each of the six datasets reveal that the skewness of data has a detrimental effect on the training of anti-spam filters. For instance, except for the aberrant trend displayed by the NB classifier, the remaining classifiers misclassify fewer hams in Enron 1-3 (ham skewed) and fewer spams in Enron 4-6 (spam skewed). This experiment also suggests that RF exhibits a similar ability to the meta-learners for spam classification (see Figure 9.7b). However, its ability to identify hams diminishes when more spams are included in its training data (see Figure 9.7a)—even so that for Enron 5 and 6, it is outperformed by NB. Overall, the results indicate that an SVM classifier is more sensitive to the skewness of the training data hence the training set should be carefully selected. With spam-skewed training data an SVM classifier's ham misclassification rate is as high as 17%. Similarly, with ham-skewed training data the spam misclassfication rate nears 18%.

We have investigated the expected performance of the classifiers on each dataset. Although we have produced cost curves for all the classifiers for the six datasets in the Enron-Spam collection, we present only the cost curves of the two best classifiers: ADABOOSTM1 and BAGGING. These cost curves can be found in Figures 9.9 and 9.10. From the curves, it is evident that we can expect that our classifier performances will vary for different ratios of hams and spams. What is interesting in these curves is that not only is the ratio of ham to spam playing a role in expected performances, but also the spams themselves. Note that the spam-skewed datasets display similar curves for both classifiers, and show similar spreads among themselves. Stylometric differences among the spams in these three datasets may be the cause for these observations. The ham-skewed datasets produce similar curves, but they are more tightly bundled. Even though the hams are from personal mailboxes, they are more likely to be stylometrically

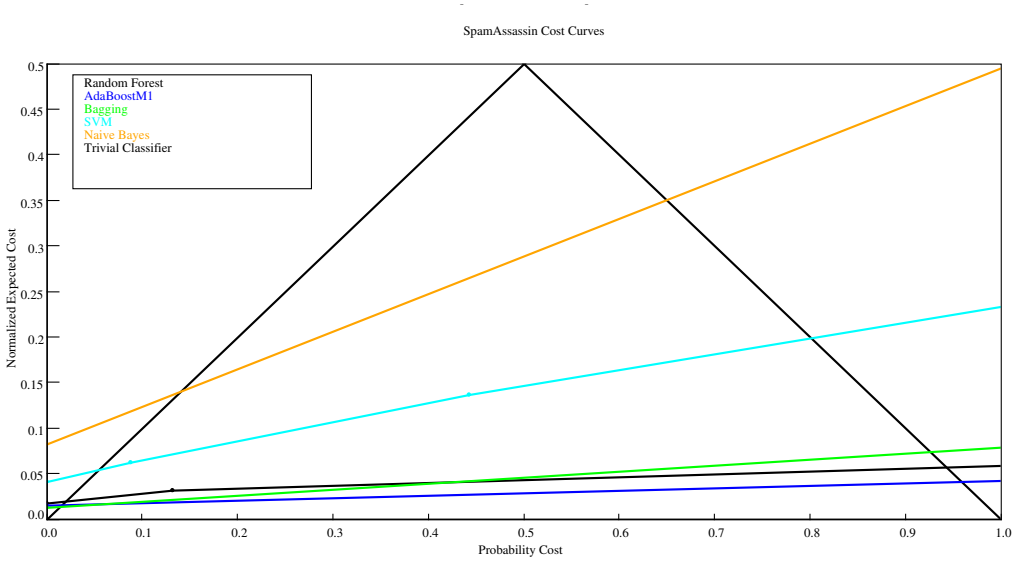


Figure 9.5: Cost curves of the five classifiers generated by SENTINEL on the SpamAssassin dataset.

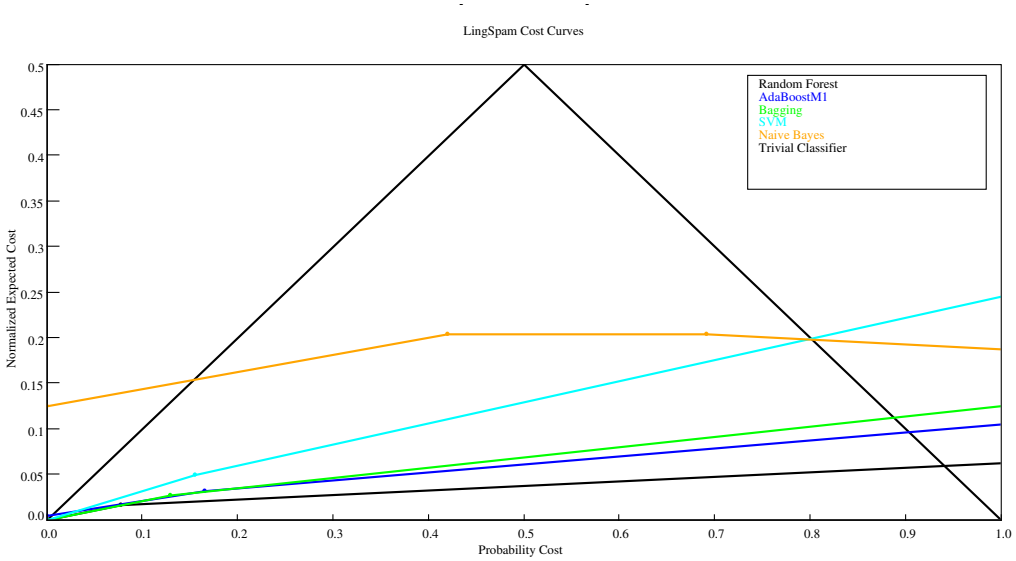
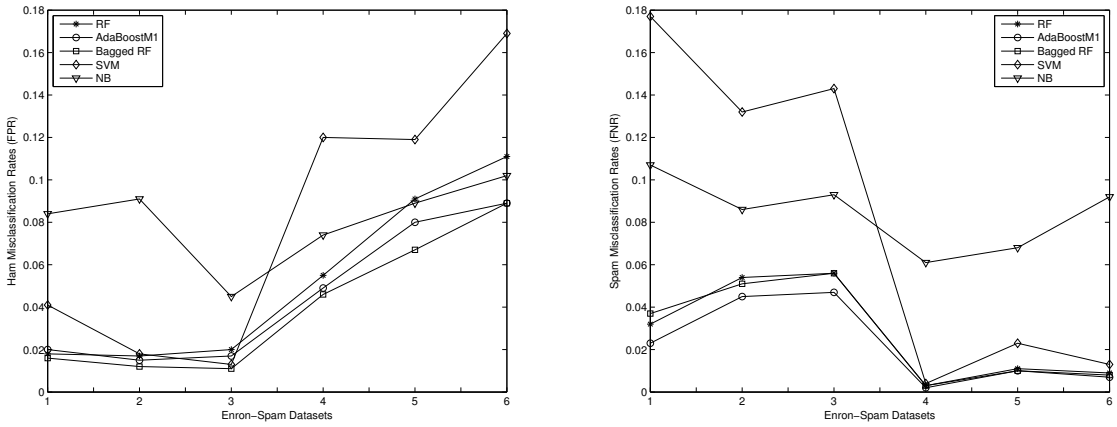
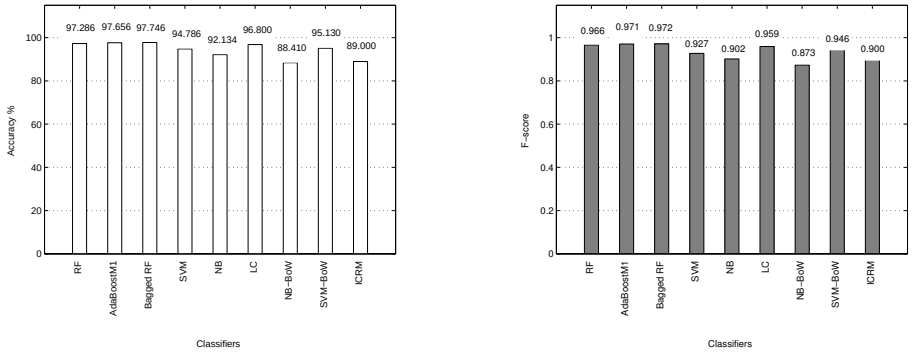


Figure 9.6: Cost curves of the five classifiers generated by SENTINEL on the LingSpam dataset.



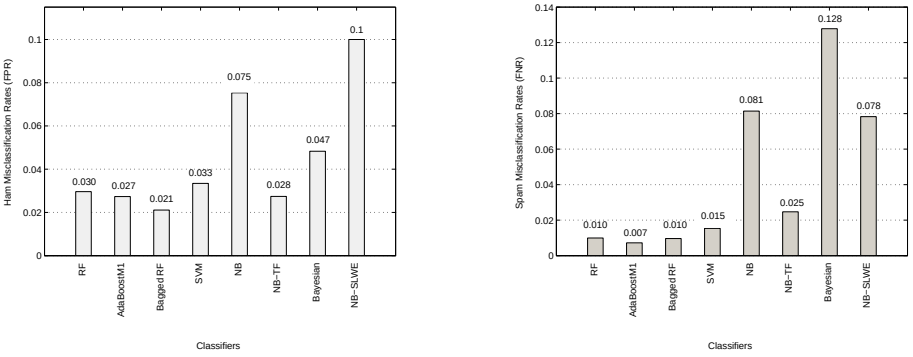
(a) (b)

Figure 9.7: (a) Ham misclassification rates and (b) spam misclassification rates of the five SENTINEL-generated classifiers on the Enron-Spam collection.



(a) Comparison of accuracies.

(b) Comparison of F-scores.



(c) Comparison of FPRs (ham misclassification rates).

(d) Comparison of FNRs (spam misclassification rates).

Figure 9.8: Comparison of the performances on the Enron-Spam collection of the five classifiers generated by SENTINEL and classifiers from previous studies.

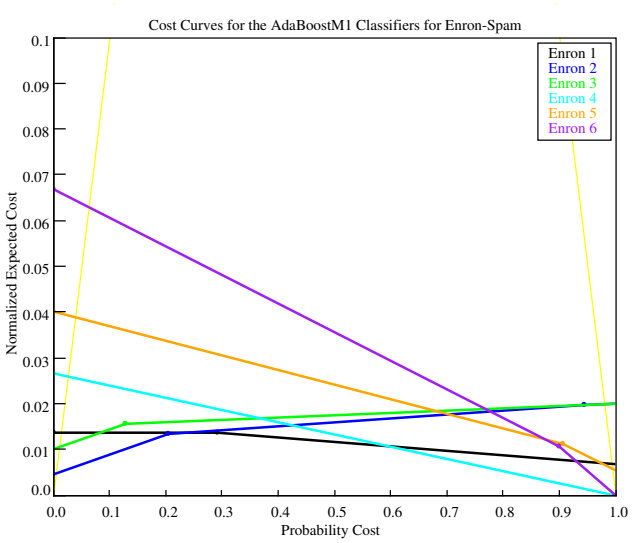


Figure 9.9: Cost curves for the ADABOOSTM1 classifiers generated for the six datasets in the Enron-Spam collection.

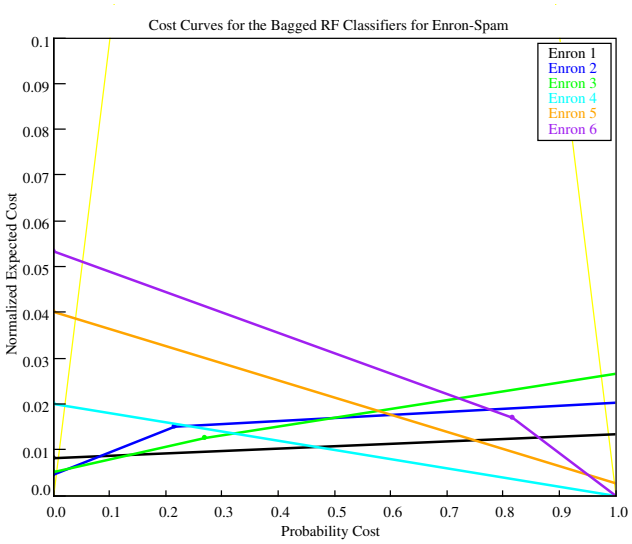
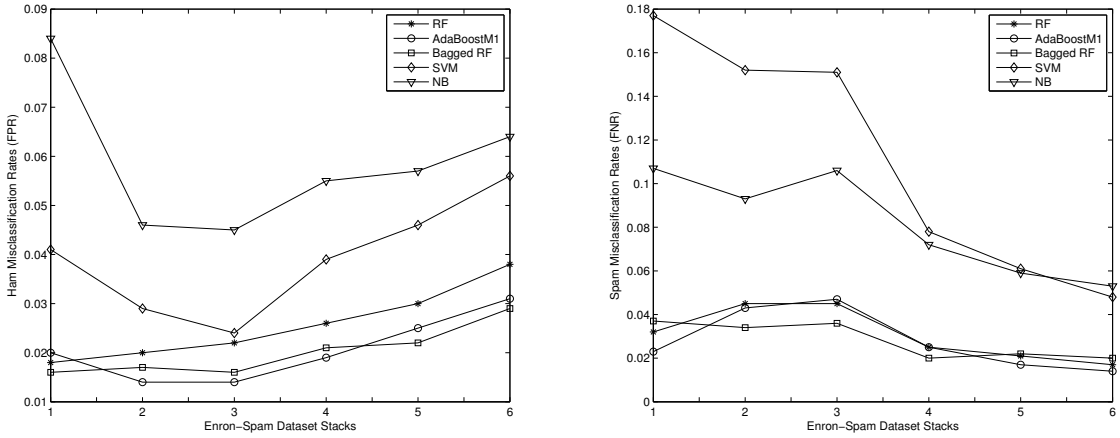


Figure 9.10: Cost curves for the BAGGED RF classifiers generated for the six datasets in the Enron-Spam collection.



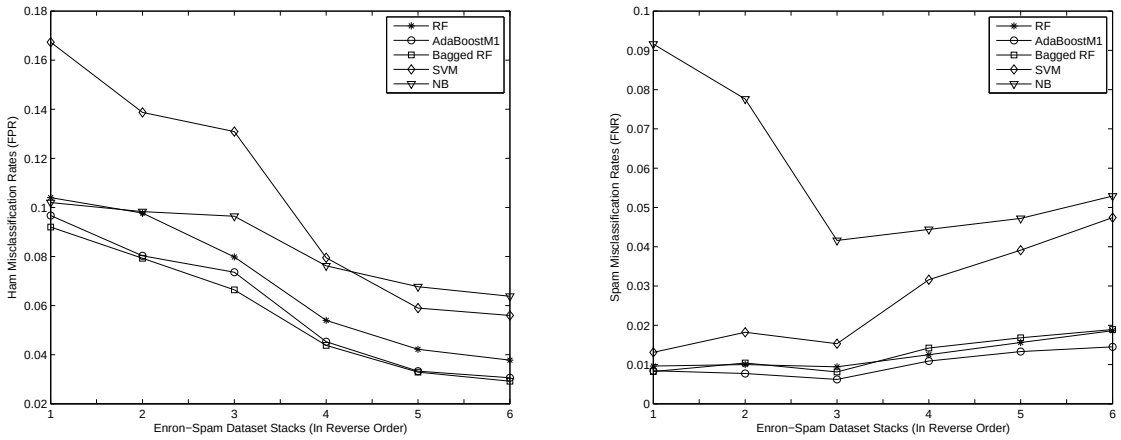
(a) (b)

Figure 9.11: The incremental (a) ham misclassification rates and (b) spam misclassification rates of the SENTINEL classifiers on the Enron-Spam collection.

similar because they are business-related. Some influence, albeit diminished because of the underrepresentation of spams, comes from the spams. Four of the curves are in a narrow band when *Probability Cost* = 0.5.

Lastly, we evaluated the performance of SENTINEL on the *stacks* of the six datasets. We generated two sets of stacks and following is the process of generating first set of stacks: starting with the stack composed of Enron 1, one dataset (in numerical order) is added to the previous stack. This process of creating the dataset stacks continues until Enron 6 is combined with Enron 1-5. Thus, the number of hams dominates up to the third stack and the ratio of spams and hams becomes closer to 1 after adding Enron 6 to the stacks of Enron 1-5. The second set of stacks is generated as follows: starting with the stack composed of Enron 6, one dataset (in reverse numerical order) is added to the previous stack. This process of creating the dataset stacks continues until Enron 1 is combined with Enron 6-2. Thus, the number of spams dominates up to the third stack and the ratio of spams and hams becomes closer to 1 after adding Enron 1 to the stacks of Enron 6-2.

As we evaluated SENTINEL on the stacks, it is evident that on a *numerically* balanced dataset, the best ham misclassification rate is achieved by BAGGED RF followed by ADABOOSTM1, RF, SVM, and NB while the best spam misclassification rate is achieved by ADABOOSTM1 followed by RF, BAGGED RF, SVM, and NB (see Stack 6 in Figures 9.11 and 9.12). Interestingly, we found that in addition to class imbalance in the datasets, the performance of SENTINEL depends on two other factors: the sources of the emails and the number of training emails for each class. A case in point, a notable change in ham misclassification rates can be observed for the first three reverse stacks (Figure 9.12a), even though the ratio of spams to hams in the three stacks is the same (i.e., 3:1). There are three differences among the stacks: the number of spam emails, the number of ham emails, and the email sources. This clearly suggests that besides the class imbalance problem, the email source and the number of training emails may have an influence on our filter's performance.



(a) (b)

Figure 9.12: The reverse incremental (a) ham misclassification rates and (b) spam misclassification rates of the SENTINEL classifiers on the Enron-Spam collection.

9.5 Conclusions

In this paper we describe the development and evaluation of an anti-spam filter named SENTINEL. The filter uses natural-language attributes, the majority being connected to stylometric aspects of writing. The real-valued, natural-language attributes extracted from the email texts are used to generate binary classifiers. The classifiers explored in this study are induced by five state-of-the-art learning algorithms. We evaluate the filter with benchmark non-personalized email datasets such as CSDMC2010, SpamAssassin, and LingSpam as well as standard personalized emails like those in the six datasets of the Enron-Spam collection. The evidence from extensive experiments implies that the classifiers that perform the best are of two ensemble methods: ADABOOSTM1 and BAGGING. In general, the performance of SENTINEL is mixed on non-personalized email data. This result is not unexpected because our findings demonstrate that the filter has limitations for non-personalized email data—mainly due to the absence of unique writing patterns in the randomly collected emails. Contrary to this, on personalized email data, SENTINEL surpasses the performances of a number of state-of-the-art personalized anti-spam filters. These outcomes imply that the attributes related to writer stylometry can better capture the imprinted patterns in personalized hams. One limitation of the filter is that its performance is affected by the extreme proportions of spams and hams in non-personalized datasets. On a good note, the filter is not affected at all by this factor on personalized datasets.

Our work clearly has some limitations. Firstly, several aspects of the filter, viz., its real-time training and response latency are not considered. It will require extensive tests to confirm SENTINEL's usability as an on-line filter. Secondly, personalized datasets share an interesting phenomenon called *concept drift* which is yet to be investigated. The reaction of the proposed filter with respect to this phenomenon can be tested by substituting the spams of the Enron-Spam collection with more recent data.

Because our results suggest that ensemble methods perform the best, further tests should be carried out to see the performance of the filter by stacking several algorithms to generate its

classifiers. Future studies can extend the work by replacing the supervised algorithms used in this study with semi-supervised learning algorithms.

Bibliography

[1] Commtouch, "Internet threats trend report," tech. rep., Commtouch, USA, April 2013.

[2] E. Blanzieri and A. Bryl, "A survey of learning-based techniques of email spam filtering," *Artificial Intelligence*, vol. 29, pp. 63–92, Mar. 2008.

[3] J. Goodman, G. V. Cormack, and D. Heckerman, "Spam and the ongoing battle for the inbox," *Communications ACM*, vol. 50, pp. 24–33, Feb. 2007.

[4] Y. Hu, C. Guo, E. W. T. Ngai, M. Liu, and S. Chen, "A scalable intelligent non-content-based spam-filtering framework," *Expert Systems with Applications*, vol. 37, pp. 8557–8565, Dec. 2010.

[5] J.-J. Sheu, "An efficient two-phase spam filtering method based on e-mails categorization," *International Journal of Network Security*, vol. 9, no. 1, pp. 34–43, 2009.

[6] V. Metsis, I. Androutsopoulos, and G. Paliouras, "Spam filtering with naive Bayes — Which naive Bayes?," in *Third Conference on Email and Anti-Spam (CEAS)*, (USA), p. 9pp, 2006.

[7] C.-C. Lai and M.-C. Tsai, "An empirical performance comparison of machine learning methods for spam e-mail categorization," in *Fourth International Conference on Hybrid Intelligent Systems*, HIS '04, (USA), pp. 44–48, IEEE Computer Society, 2004.

[8] A. Qaroush, I. M. Khater, and M. Washaha, "Identifying spam e-mail based-on statistical header features and sender behavior," in *CUBE International Information Technology Conference*, (USA), pp. 771–778, ACM, 2012.

[9] M. Ye, T. Tao, F.-J. Mai, and X.-H. Cheng, "A spam discrimination based on mail header feature and SVM," in *Fourth International Conference on Wireless Communications, Networking and Mobile Computing (WiCom08)*, pp. 1–4, 2008.

[10] Q. Ma, Z. Qin, F. Zhang, and Q. Liu, "Text spam neural network classification algorithm," in *2010 International Conference on Communications, Circuits and Systems (ICCCAS)*, (China), pp. 466–469, 2010.

[11] C. Orăsan and R. Krishnamurthy, "A corpus-based investigation of junk emails," in *Third International Conference on Language Resources and Evaluation (LREC-2002)*, (Spain), pp. 1773–1780, May, 29 – 30 2002.

[12] Y. Zhu and Y. Tan, "A local-concentration-based feature extraction approach for spam filtering," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 2, pp. 486–497, 2011.

[13] S. Afroz, M. Brennan, and R. Greenstadt, "Detecting hoaxes, frauds, and deception in writing style online," in *2012 IEEE Symposium on Security and Privacy (SP)*, (USA), pp. 461–475, 2012.

[14] F. Iqbal, L. A. Khan, B. C. M. Fung, and M. Debbabi, "E-mail authorship verification for forensic investigation," in *Proceedings of the 2010 ACM Symposium on Applied Computing*, SAC '10, (New York, NY, USA), pp. 1591–1598, ACM, 2010.

[15] J. Yang, Y. Liu, Z. Liu, X. Zhu, and X. Zhang, "A new feature selection algorithm based on binomial hypothesis testing for spam filtering," *Knowledge-Based Systems*, vol. 24, pp. 904–914, Aug. 2011.

[16] B. Sirisanyalak and O. Sornil, "Artificial immunity-based feature extraction for spam detection," in *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. SNPD 2007. Eighth ACIS International Conference on*, vol. 3, pp. 359–364, 2007.

[17] A. Bratko, G. V. Cormack, D. R, B. Filipic, P. Chan, T. R. Lynam, and T. R. Lynam, "Spam filtering using statistical data compression models," *Journal of Machine Learning Research*, vol. 7, pp. 2673–2698, 2006.

[18] R. Prabhakar and M. Basavaraju, "A novel method of spam mail detection using text based clustering approach," *International Journal of Computer Applications*, vol. 5, pp. 15–25, August 2010. Published By Foundation of Computer Science.

[19] G. V. Cormack and A. Bratko, "Batch and online spam filter comparison," in *Conference on Email and Anti-Spam, CEAS 2006*, (Mountain View, CA), p. 9pp, July 2006.

[20] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A Bayesian approach to filtering junk e-mail," in *Learning for Text Categorization: Papers from the 1998 Workshop*, (USA), pp. 55–62, AAAI Technical Report WS-98-05, 1998.

[21] B. Issac, W. J. Jap, and J. H. Sutanto, "Improved Bayesian anti-spam filter implementation and analysis on independent spam corpuses," in *2009 International Conference on Computer Engineering and Technology - Volume 02*, (USA), pp. 326–330, IEEE Computer Society, 2009.

[22] P. Graham, "A plan for spam," Aug. 2003.

[23] A. Kosmopoulos, G. Paliouras, and A. Androutsopoulos, "Adaptive spam filtering using only naive Bayes text classifiers," in *Fifth Conference on Email and Anti-Spam (CEAS 2008)*, (USA), p. 3pp, 2008.

[24] T. S. Guzella and W. M. Caminhas, "A review of machine learning approaches to spam filtering," *Expert Systems with Applications*, vol. 36, pp. 10206–10222, Sept. 2009.

[25] P. Haider, U. Brefeld, and T. Scheffer, "Supervised clustering of streaming data for email batch detection," in *24th International Conference on Machine Learning*, (USA), pp. 345–352, ACM, 2007.

[26] X. Carreras and L. Màrquez, "Boosting trees for anti-spam email filtering," in *RANLP-2001, 4th International Conference on Recent Advances in Natural Language Processing*, pp. 58–64, 2001.

[27] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer Series in Statistics, Springer, 2001.

[28] A. Abi-Haidar and L. M. Rocha, "Adaptive spam detection inspired by a cross-regulation model of immune dynamics: A study of concept drift," in *Artificial Immune Systems*, pp. 36–47, Springer, 2008.

[29] A. Abi-Haidar and L. M. Rocha, "Adaptive spam detection inspired by the immune system," in *ALIFE*, pp. 1–8, 2008.

[30] I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, and C. D. Spyropoulos, "An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages," in *23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, (USA), pp. 160–167, ACM, 2000.

[31] A. Liaw and M. Wiener, "Classification and regression by randomForest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.

[32] R. E. Schapire, "A brief introduction to boosting," in *16th international joint conference on Artificial intelligence - Volume 2*, IJCAI'99, (USA), pp. 1401–1406, Morgan Kaufmann Publishers Inc., 1999.

[33] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, pp. 123–140, 1996.

[34] R. C. Holte and C. Drummond, "Cost-sensitive classifier evaluation using cost curves.," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)* (T. Washio, E. Suzuki, K. M. Ting, and A. Inokuchi, eds.), vol. 5012 of *Lecture Notes in Computer Science*, pp. 26–29, Springer, 2008.

[35] C. Drummond and R. Holte, "Cost curves: An improved method for visualizing classifier performance," *Machine Learning*, vol. 65, no. 1, pp. 95–130, 2006.

[36] M. Razmara, A. Razmara, and M. Narouei, "Textual spam detection: An iterative pattern mining approach," *World Applied Sciences Journal*, vol. 20, no. 2, pp. 198–204, 2012.

[37] J. Zhan, B. J. Oommen, and J. Crisostomo, "Anomaly detection in dynamic systems using weak estimators," *ACM Transactions on Internet Technology*, vol. 11, pp. 3:1–3:16, July 2011.

Chapter 10

Protein interaction sentence classification with natural language stylometry

This chapter is based on the paper titled "Protein interaction sentence classification with natural language stylometry" that is submitted to the BMC Bioinformatics. The paper is still under review process.

Protein-protein interaction is secondary biomedical information pivotal to understanding critical biological processes like DNA replication, signalling pathways, and cell cycle control. Large numbers of these interactions can be found in biomedical research articles. However, the number of published articles is increasing so rapidly in this field that manual curation of protein interactions from them becomes difficult. This results in a significant amount of such information remaining hidden in the articles. Therefore, it is necessary to target automatic extraction of protein interaction information.

The success of automatic protein-protein interaction extraction depends on the identification of *candidate sentences*, that is, those sentences containing an interaction relation between two proteins. In this paper, we report a novel supervised approach that can classify candidate sentences with substantial recall. The reported approach is different than most others found in classic studies. Rather than using syntactic and semantic aspects of sentences, we exploit natural language stylometry which is the linguistic study of writing styles of authors. The writing styles are represented as attributes, called the *stylometric attributes*, that are extracted from the sentences of five standard biomedical datasets. With these attributes, supervised models are then induced by cascading 1-Nearest Neighbour with Naïve Bayes. These models classify sentences into one of two classes: *candidate* or *non-candidate* sentences. Evaluation of the classification made by the proposed approach with the gold standards shows that it performs better than the benchmarks set on some of the datasets.

This research has two major contributions. First, to the best of our knowledge, only two datasets have sentence-level annotations to denote the presence or absence of protein interactions. Three more datasets are successfully annotated and validated in this research. We believe that they can be used as test beds for many text mining tools. Second, we establish stylometric attributes as a means to identify protein interaction sentences in biomedical research articles. The key advantage of these attributes is that they are related to the writing style of authors and do not depend on syntax or semantics present in the text.

10.1 Background

Librarians, especially curators of databases, face the demanding task of identifying relevant secondary biomedical information from primary sources, viz., research articles. Such information includes but is not limited to genetic etiology of diseases, genotype-phenotype relations, protein interactions and sequencing, and biomolecular interactions. Databases that contain these various information types include MINT [1], BIND [2], and Swiss-Prot [3]. Interestingly, to bypass errors during data population, most of these databases are curated manually. However, the number of research articles published annually in the biomedical domain is already overwhelming and the number is increasing every year making this manual curation of information ever more difficult. Moreover, much important information stays hidden in the articles not because of curation error only but also due to the difficulty in extracting accurate information from long text documents [4].

The identification of protein interaction containing sentences in biomedical publications is considered to be one of the unresolved pattern recognition problems [5]. Throughout this paper, the sentences bearing a protein interaction are called the *candidate sentences*; the rest are called the *non-candidate sentences*. The success of this classification is salient to the detection of the interacting protein pairs in research articles. Therefore, a number of previous studies address the problem of candidate sentence classification. For instance, in their comparative study, Sugiyama *et al.* [6] construct binary classifiers using K-nearest neighbours [7], decision tree [8], neural network [9], and Support Vector Machine (SVM) [10]. These classifiers exploit attributes related to verbs, nouns, and the part-of-speech (POS) tags present within a 20-word radius. This work shows that SVM classifiers are highly effective for the classification task. The effectiveness of kernel-based classifiers like SVM for this task is also supported by the work of Mitsumori *et al.* [11] and Erkan *et al.* [12]. The former work identifies candidate sentences using word level attributes while the latter uses similarities of the paths present between two protein names found in the dependency parses of the sentences. Of note, Erkan *et al.* [12] is the first semi-supervised classification approach for this domain. Taking a different approach, Polajnar *et al.* [13] show that a non-parametric probabilistic classifier called the Gaussian Process [14] has strength similar to SVM but better than that of Naïve Bayes [15]. They use several bag-of-words and protein named entity attributes derived from biomedical abstracts. The key advantage of their method is the use of simple attributes that do not require the dependency parsing used in [12]. They also apply a semi-supervised approach [5] to enable the use of a large quantity of unlabelled data with the same set of attributes. However, in contrast to their earlier work [13], their modelling of the co-occurrence of words is achieved by applying two unique semantic spaces [16]. Yet another class of approaches target the syntactic properties of text with full or shallow parsing (see, for instance, Yakushiji *et al.* [17]). However, the performance of these syntactic-based approaches is limited by the choice of off-the-shelf software components like context-free grammar parsing tools.

In this paper, we aim to classify protein interaction sentences and not to detect interacting protein pairs in biomedical research articles. In doing so, we view the task as a supervised binary classification problem where the two classes are *candidate sentences* and *non-candidate sentences*, where the former contains a protein interaction while the latter does not. We choose 103 attributes that are related to natural language stylometry. Stylometry is the linguistic study

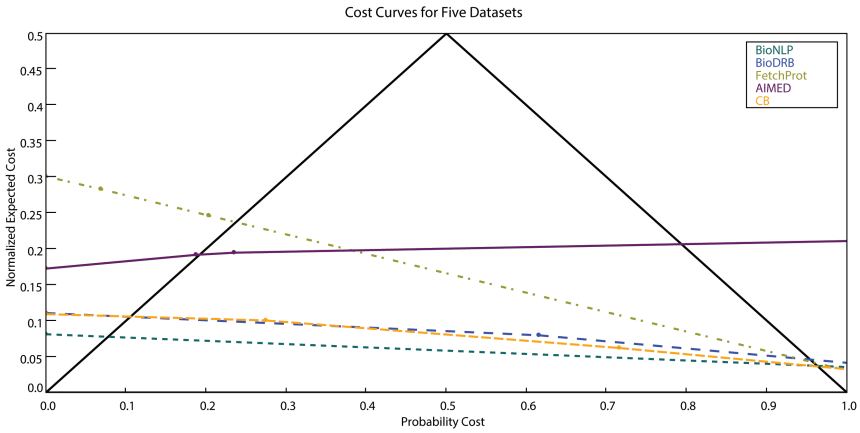


Figure 10.1: Cost curves of the classifiers for five datasets.

of authors' writing styles. Using stylometric attributes is a popular method to represent writing style patterns of article authors. The motivation for the use of these attributes is that when they put important information in their articles, writers change their writing style for these informative sentences [18]. Protein interactions are pivotal information and therefore the style present in sentences that describe them should be different than others. Moreover, our past studies [19] [20] [21] suggest that stylometric attributes have substantial advantages over syntactic and semantic attributes for biomedical relation mining and text classification. To the best of our knowledge, the set of attributes used in this research is novel for the classification task presented here. As test beds, we choose five standard biomedical datasets named BioNLP [22], BioDRB [23], FetchProt [24], AIMED [12], and Christin-Brun (CB) [12]. We annotate the first three datasets using two automated tools followed by human validation, while the last two datasets, already in use for the candidate sentence extraction task, are human annotated. Each sentence in the datasets is represented as a vector of the 103 stylometric attributes. For each dataset, we use a multi-stage classification, cascading 1-Nearest Neighbour with Naïve Bayes, to generate models with these attributes. The models are then used to classify a sentence as a candidate or non-candidate sentence. Using a 10-fold cross validation, we evaluate the performance of each model against the gold standard annotations. Experimental outcomes show that our approach has a remarkable recall. Overall, our approach outperforms the benchmarks set for the AIMED and CB datasets.

10.2 Results and Discussion

To report the performance of stylometric attributes for protein interaction candidate sentence classification, we carried out extensive tests. The results are described in this section by using both cost-insensitive measures like precision, recall, and F-score as well as cost-sensitive measures like Cost Curves and Receiver Operating Characteristics (ROC) curves.

Dataset	Classifier	Precision (%)	Recall (%)	F-score (%)
BioNLP	Multi-stage	76.9	92.6	84.0
BioDRB	Multi-stage	64.6	90.9	75.5
FetchProt	Multi-stage	39.0	96.1	55.5
AIMED	Multi-stage	53.3	74.7	62.2
AIMED	TSVM [12]	59.6	60.70	60.0
AIMED	Rule-based [17]	33.7	33.1	33.4
AIMED	SVM [11]	54.2	42.6	47.7
CB	Multi-stage	88.9	93.7	91.2
CB	TSVM [12]	85.6	84.9	85.2

Table 10.1: Precision, recall, and F-score of our approach for the five datasets. Comparisons for the AIMED and CB datasets are also illustrated.

10.2.1 Cost-insensitive Analysis

First, we discuss the performance of our approach for classifying protein interaction sentences from research articles using cost-insensitive measures like precision, recall, and F-score. The three datasets that we annotate for this experiment (BioNLP, BioDRB, and FetchProt) were used previously for tasks that are completely different than our task. Therefore, there is no benchmark set for these three datasets. However, observing the current state-of-the-art work of Erkan *et al.* [12] we can argue that our approach performs reasonably well on these three datasets. We compare our work directly with Erkan *et al.* [12] on the remaining datasets: AIMED and CB.

Table 10.1 strongly suggests that our approach has remarkably high recall maintaining competitive, albeit low, precision—for all datasets except AIMED, the recall is above 90%. Another interesting observation can be that for all but the AIMED dataset, our approach achieves better recall than precision. The anomalies regarding the AIMED dataset appear to be well substantiated by the findings of Faiz and Mercer [25] who have conducted a comparative study and show that the sentences of the AIMED dataset are more complex than the other benchmark datasets. The high recall values found in our tests are striking since they demonstrate the efficiency of our approach to classify a significant fraction of protein interaction sentences present in the datasets. The foremost cause of low precision is due to the class imbalance problem in the datasets (see Table 10.2). Note however that the recall is more important than the precision in tasks like protein interaction sentence classification. F-score is the harmonic mean of precision and recall. The low precision dominates the F-score in our case.

Interestingly, we get both the highest recall (96.1%) and the lowest precision (39.0%) for the FetchProt dataset. A reasonable explanation for this result can be that the FetchProt dataset has the smallest imbalance ratio (see Eqation 10.1 and Table 10.2). Due to the presence of a large number of non-candidate sentences in the dataset, our approach misclassifies many non-candidate sentences as candidates. Consequently, the lowest F-score of our approach is recorded on this dataset (55.5%). On the other hand, on the CB dataset our approach exhibits the best precision (88.9%) and it is noted that this dataset is more balanced than the rest. This precision and the second best recall (93.7%) result in our best F-score (91.2%) to be found on this dataset.

Dataset	Total Sentences	Candidate Sentences	Non-Candidate Sentences	Imbalance Ratio
BioNLP	6857	1735	5122	0.34
BioDRB	4704	975	3729	0.26
FetchProt	73954	12504	61450	0.20
AIMED [12]	4026	951	3075	0.31
CB [12]	4056	2202	1854	1.19
Total	93597	18367	75230	0.24

Table 10.2: Brief summary of the datasets used in this experiment.

We compare our results on the AIMED and CB datasets with that found by Erkan *et al.* [12] who apply a semi-supervised SVM (TSVM). The comparisons are made with a paired *t-test* with a significance level set to 5% (i.e., $\alpha = 0.05$). On the AIMED dataset, in contrast to their recall (60.7%) and F-score (60.0%), our recall (74.7%) and F-score (62.2%) are significantly better. The only case where their method outperforms our approach is the precision on the AIMED dataset (59.6% vs. 53.3%). Note that the F-scores on this dataset reported by Yakushiji *et al.* [17] and Mitsumori *et al.* [11] are 33.4% and 47.7%, respectively—these are significantly lower than our F-score reported in Table 10.1 (62.2%). Similar results are found on the CB dataset. In both precision and recall, our approach outperforms the results reported by Erkan *et al.* The F-score recorded on this dataset is 91.2% which is significantly better than their F-score of 85.2%.

10.2.2 Cost-sensitive Analysis

Table 10.2 clearly shows that the datasets used in this experiment are unbalanced and except for the CB dataset, their imbalance ratio is less than 1 (i.e., they contain more non-candidate sentences). A cost-sensitive analysis with Cost Curves [26] [27] can give us an idea whether results similar to those in Table 10.1 can be found if the datasets were balanced.

Figure 10.1 illustrates the cost curves of the classifiers we generated from five datasets. The *X-axis* of the plot denotes the probability cost while the *Y-axis* denotes the normalized expected cost associated with the classifiers. In other words, a value on the *Y-axis* at $X = 0.5$ reveals the expected misclassification cost of a classifier when presented with an equal number of candidate and non-candidate sentences. The results highlight that had the datasets contained more candidate sentences, the misclassification rate for candidate sentences by our classifiers would have decreased. What is surprising is that the situation is completely opposite for the AIMED dataset. The cost curves are almost stationary. This indicates that the misclassification rate of our approach will not be significantly affected by the presence of more candidate sentences. To confirm that rebalancing the classes will not change the misclassification rates, we have run a significance test on the curves with confidence interval set to 95% and found that this assumption is true for all the datasets except FetchProt and AIMED. This also explains the low precisions that we report on the FetchProt and AIMED datasets in the *Cost-insensitive Evaluation* section. Note that in terms of misclassification cost for a balanced dataset (i.e., X=0.5 on

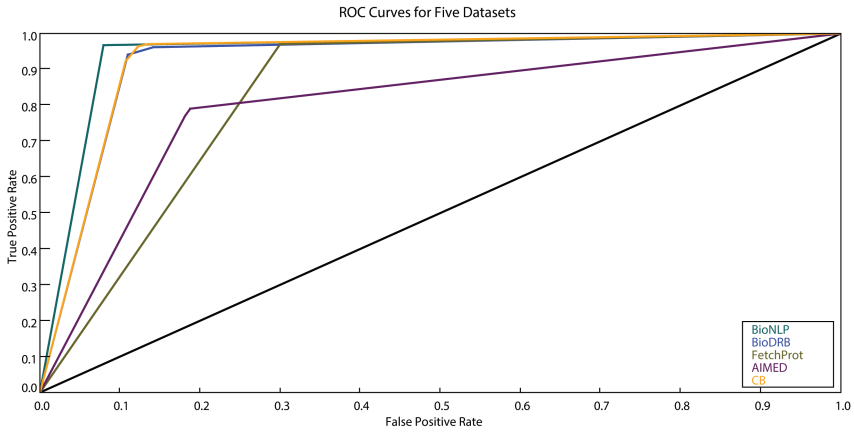


Figure 10.2: ROC curves of the classifiers for five datasets.

the plot), our approach performs the best on BioNLP followed by CB and BioDRB.

Similar results are found from the ROC curve analysis of our classifiers on the five datasets (see Figure 10.2). The plot strongly indicates that our approach gains the best recall with the lowest false positive rate (the rate that denotes the fraction of non-candidate sentences that are misclassified) on the BioNLP dataset. Among the rest, our approach misclassifies almost equally on the BioDRB and CB datasets. The recall on the FetchProt dataset becomes almost equal to the recalls found on the aforementioned datasets but at the expense of a larger false positive rate. The performance of our approach in terms of gaining recall at the expense of false positives on the AIMED dataset is not satisfactory.

10.3 Conclusions

An important initial step for successful protein-protein interaction extraction is first to classify *candidate sentences* that describe protein interaction relations. The evidence from this study strongly suggests that stylometric attributes are strong contenders for classifying candidate sentences. Our approach is different than most classical approaches that depend on the syntax or semantics of text. In this study, we use stylometric attributes that portray the writing styles of authors. Our experiments are extensive as to evaluate the attributes' performance, we use five standard biomedical datasets: BioNLP, BioDRB, FetchProt, AIMED, and CB. Among the datasets, however, only the last two had been used for the protein interaction sentence classification task. Therefore to start with, we prepared the first three datasets by annotating them with a rule-based and a supervised machine-learned annotator and validating the annotations with two human domain experts. Tests show that the level of annotation agreements is fair on the Landis and Koch scale [28].

Each sentence of the five datasets is represented as a vector of stylometric attributes. For each dataset, the attributes are used to train a multi-stage classifier that involves 1-Nearest Neighbour and Naïve Bayes. The models generated by the classifiers are thereafter used to classify the sentences of respective datasets into candidate and non-candidate sentences. Test

results are encouraging and show that our approach has a remarkable recall. We compare our approach to several contemporary supervised and semi-supervised methods, and we find that our approach is a clear improvement.

We are aware that our approach may have two limitations. Firstly, the precision of our approach is low, particularly for two datasets. Test results show that this is possibly a result of a high class imbalance present in the datasets and the level of difficulty involved in their sentence structure. Secondly, all the datasets used in this study have been annotated by automated tools and later validated by humans. Therefore, there is always the possibility that the tools misjudge the sentence labels. However, the number of sentences dealt with in this study is large and we accept this limitation of the tools when balanced against the workload required for human annotations.

Despite some minor limitations, our research can be a constructive method for classifying protein interaction sentences as a first step for protein-protein interaction extractors. In our view, the results reported constitute an excellent initial step toward the protein-protein interaction extraction task.

10.4 Methods

We use five standard biomedical datasets. A brief description of the datasets are given in this section. Two of the datasets are already annotated for the protein interaction sentence classification task. The rest have been curated for other purposes and do not contain annotations required for this task. The annotation of these three datasets and the validation process are also discussed. This section also lists and describes the stylometric attributes in this research. As well, we outline the design of the classifiers, the experimental procedure, and the performance measures.

10.4.1 Datasets

We use five standard biomedical datasets to classify candidate sentences that describe protein interactions. The datasets we use are BioNLP [22], BioDRB [23], FetchProt [24], AIMED [12], and Christine-Brun (CB) [12]. The first three datasets (BioNLP, BioDRB, and FetchProt) contain raw texts and do not have any annotations at the sentence level. We choose these datasets since they are used extensively for the protein-protein interaction extraction task and therefore contain a good number of candidate sentences. The last two datasets (AIMED and CB) have sentence-level annotations [12]. These two datasets have candidate sentence classification benchmarks set for them and therefore make it possible for us to gauge the performance of our approach.

Description

BioNLP is composed of 40 full-length articles that were used for five different shared tasks at the *BioNLP–2011 Challenge*. We collect the articles from the training, development and test sets of that task. This dataset contains approximately $6,857$ sentences with an average

of 171 sentences per article. BioDRB, on the other hand, is not, strictly speaking, a protein-protein interaction dataset. It is composed of 24 full biomedical articles that are annotated with discourse relations. However, we find that this dataset contains a reasonable number of protein interactions, making it a good source of candidate sentences. The BioDRB dataset contains approximately 4,704 sentences with 196 sentences per article on average. FetchProt comprises 190 articles of which 140 describe experimental evidence for *tyrosine kinase* activity for at least one protein. Out of the 190, we keep 169 articles and discard the rest as they have improper text formatting issues. This reduced dataset contains 73,954 sentences and each article on average contains about 438 sentences. Note that FetchProt is the largest of the five datasets that we use. The AIMED and CB datasets contain 4,026 and 4,056 sentences, respectively. The annotations for these two datasets were accomplished by using supervised machine learning approaches followed by a careful inspection by human curators [12]. A brief summary of the five datasets can be found in Table 10.2. The right-most column of the table shows the *imbalance ratio* in the datasets that can be defined as follows:

$$\textit{Imbalance Ratio} = \frac{\text{\# Candidate Sentences}}{\text{\# Non-Candidate Sentences}}. \tag{10.1}$$

Annotation

As we stated earlier, the BioNLP, BioDRB, and FetchProt datasets are not annotated at the sentence level, and we need to annotate each sentence of these datasets with one of the two labels: candidate and non-candidate. We use two automated tools for this annotation task and two human domain experts for validation. A description of the annotation task and the validation follows.

Annotation Tools We use a state-of-the-art rule-based relation miner called RelEx [29] to automatically annotate the BioNLP, BioDRB, and FetchProt datasets. It extracts candidate relations, which are later filtered, by applying three simple rules on the dependency parse trees [30] generated from each sentence. As well, we use a machine learning version of RelEx called WRelEx [25]. This tool extracts three types of attributes from the dependency trees of sentences: (i) dependency, (ii) syntactic and (iii) surface level attributes. With these attributes, WRelEx generates a binary *maximum entropy* classifier to identify relations in the sentences. We choose these tools for the annotation because of their very good performances on biomedical data. RelEx, for instance, was evaluated with a comprehensive dataset of 1 million MEDLINE abstracts and it achieved 80% precision and 80% recall. WRelEx, on the other hand, was evaluated with five protein interaction datasets: AIMED, BioInfer, HPRD50, IEPA, and LLL and in most cases, it outperformed RelEx (for details, see [25]).

The two annotating tools use different text processing resources. Therefore, we have tailored them both so that they use common resources. As a case in point, the LingPipe machine learning API [31] is integrated with RelEx and WRelEx so that both tools detect common sentence boundaries. This particular shared feature is necessary because a failure to detect common sentence boundaries can lead to inappropriate comparisons between the two tools. As another example, the Genia Tagger [32] is combined with the tools to detect proteins in the texts. WRelEx has adopted, like RelEx, to use the Stanford dependency parser [30] to gen-

Label	tab	Sentence	newline
Positive	\t	Like FAK, it lacks obvious protein interaction domains, and has also been implicated in the activation of ERK and JNK MAPKs	\n
Negative	\t	The signaling pathways activated by NaCl are less clear	\n

Table 10.3: Annotation format for the datasets. The tools annotate each sentence with its label followed by a tab, the sentence itself and a newline.

erate dependency parse trees for each sentence. The annotations made by the tools are based on their respective working principles—RelEx uses its rules and WRelEx uses its *maximum entropy* model. Table 10.3 shows the simple annotation format that the annotator tools follow with a positive and a negative example: the positive example refers to a candidate sentence and the negative example refers to a non-candidate sentence.

To judge the reliability of the automatic annotations, we use the standard reliability measures: *Cohen's Kappa* (κ) [33] and *Krippendorff's Alpha* (α) [34]. The κ and α values for the annotations done by the two tools are equal to 0.99, where a 1.00 indicates perfect agreement. The confusion matrix for their annotations can be found in Table 10.4. According to the annotation agreements, the tools make very few Type I and Type II errors, and disagree on only 6 sentences. These sentences are re-annotated individually by two human annotators—one is the author of WRelEx and the other is an author of this paper. From their double-blind reviews, for all six sentences, the human annotators agree with the annotation made by WRelEx. So, the annotations made by WRelEx are chosen as the automatic annotations for the three datasets.

Annotation Validation The annotation agreements in Table 10.4 show that the three datasets, BioNLP, BioDRB, and FetchProt, are negatively skewed (i.e., they contain more non-candidate sentences). For these skewed datasets, overly optimistic κ and α values can be found simply because the annotator tools can agree 82% of the time by tagging everything as non-candidate sentences. Therefore, we need humans to validate these machine annotations. To accomplish this, we engage two human experts (biochemistry graduate students who study protein-protein interactions) to re-annotate a statistically representative subsample of the datasets.

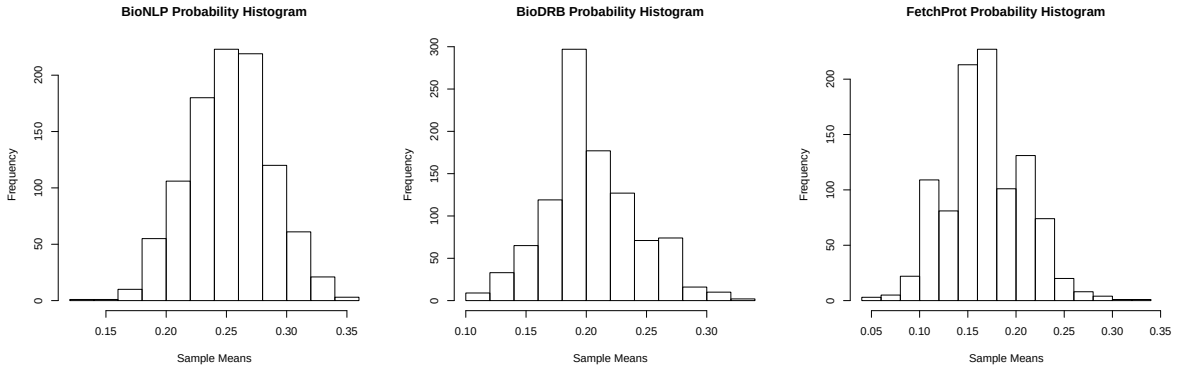
To calculate the size, n, of a statistically representative sample, we use the formula in Equation 10.2.

$$n = [\frac{z_{\alpha/2}\sigma}{E}]^2. \tag{10.2}$$

Here, n is the sample size; $z_{\alpha/2}$ is the critical value (i.e., the positive z value that is at the vertical

		RelEX		
		+	−	Total
WRelEx	+	15209	2	15211
	−	4	70300	70304
	Total	15213	70302	85515

Table 10.4: Confusion matrix for the dataset annotation by RelEx and WRelEx.



(a) (b) (c)

Figure 10.3: Probability Histogram for 1000 samples of (a) BioNLP, (b) BioDRB, and (c) FetchProt corpus. The histograms show the normality of the distributions.

boundary for the area of $\alpha/2$); σ is the population standard deviation; and E is the maximum allowed difference between the sample mean $\bar{x}$ and the population mean μ. For this study, we set $z_{\alpha/2} = 1.96$ for 95% confidence and a significance level of $\alpha = 0.05$, and $E = 0.03$. This gives us the sample sizes, n, for each of these datasets (see Table 10.5).

For each dataset, we then randomly generate $1,000$ samples of the appropriate size n and calculate the mean of their individual sample mean distributions, population means, sample standard deviations, and population standard deviations. Using these descriptive statistics and the three characteristics of the hypothetical distribution of sample means, we validate that the samples are representative of the populations as follows:

1. The mean of the distribution of sample means $\mu_{\bar{x}}$ should be close to the mean of the population of the individuals μ. Table 10.5 shows that the sampling meets this criterion as $\mu_{\bar{x}}$ and μ are almost equal for each corpus.

2. The standard deviation of the distribution of sample means $\sigma_{\bar{x}}$ should be less than the standard deviation in the population of individuals, σ. In other words, $\sigma_{\bar{x}} \approx \frac{\sigma}{\sqrt{n}}$. From Table 10.5, we can see that the sampling meets this criterion as well.

3. A plotted probability histogram (i.e., a histogram produced from the distribution of sample means) should be approximately normal for a sample size of $n >= 30$. Figures 10.3a, 10.3b, and 10.3c show the probability histograms for each dataset; their normality shows that this characteristic has been satisfied by the sample.

One of these $1,000$ samples of 351 sentences has been given to two human experts for re-annotation. For a fair re-annotation, we make sure that the experts (i) annotate the sentences in an uninterrupted environment; (ii) annotate blindly—one expert is not allowed to see the annotations made by the tools or by the other human expert; (iii) annotate the sentences in two different sessions for a test/retest evaluation; (iv) have a sufficient gap between the sessions so that they do not annotate from memory in the second session; (v) feel comfortable about the

Dataset	Mean of Sample Means, $\mu_{\bar{x}}$	Population Mean, μ	Samples *SD*, $\sigma_{\bar{x}}$	Population *SD*, σ	Sample Size, n
BioNLP	0.25	0.25	0.03	0.40	152
BioDRB	0.20	0.21	0.04	0.44	115
FetchProt	0.17	0.17	0.04	0.36	84
Total					**351**

Table 10.5: Statistics to validate the representativeness of the samples for the datasets.

time constraint so that they do not get fatigued; and (vi) get compensated for their participation since voluntary work sometimes does not meet the expected quality.

We measure the standard ϕ correlation coefficient of each human annotator for their annotations in Session 1 and Session 2. This gives their individual intra-annotator agreements for each dataset. Table 10.6 shows that for each dataset the humans have almost similar ϕ-values, except for the BioDRB dataset. A post-annotation discussion with Human 1 provides an explanation for this anomaly as the annotator admitted that his confidence level was low for his Session 1 annotation. Otherwise, the similarities between the ϕ-values show that the humans are reasonably consistent with their annotations. According to the scales proposed by Landis and Koch [28], the humans agree with themselves *moderately* with $\phi >= 0.40$.

To measure the inter-annotator agreements, we consider the ϕ coefficients between the annotations of each session by the humans and the tools, and average them. To find their average, we use *Fisher's Transformation* [35] to convert the ϕ's to the corresponding z's which are then averaged and converted back to give us ρ—the average of two correlation coefficients. We measure the ρ for inter-human annotations in a similar way. Table 10.7 illustrates the ρ's among the annotators. The inter-human measures show that the humans have *moderate* agreements (i.e., $\rho >= 0.40$) while the humans and machines have *fair* agreement (i.e., $0.20 < \rho < 0.40$ except for the aforementioned BioDRB dataset).

Confusion matrices to measure the correlations in Table 10.7 show that the tools correctly identify most of the non-candidate sentences as indicated by the human annotators. As the datasets contain more non-candidate sentences than candidate sentences, recognizing a good number of non-candidate sentences indicates the quality of the tools' annotations.

To understand better some of the results, we requested the human annotators to answer

Dataset	Human 1 vs. Human 1	Human 2 vs. Human 2
BioNLP	0.64	0.61
BioDRB	0.40*	0.65
FetchProt	0.81	0.80

∗From a post-annotation feedback, we came to know that human 1 was not fully confident with annotations for session 1, especially with the sample from BioDRB.

Table 10.6: Intra-annotator correlation measures. The measures show the moderate agreement between the human annotators.

Dataset	Human 1 vs. Machine	Human 2 vs. Machine	Human vs. Machine	Human 1 vs. Human 2
BioNLP	0.35	0.40	0.37	0.44
BioDRB	0.06	0.27	0.16	0.49
FetchProt	0.32	0.40	0.36	0.67

Table 10.7: Inter-annotator correlation measures. The measures show that the human-human agreement is moderate while the human-machine agreement is fair.

8 questions at the end of the sessions. The first 7 of these 8 questions are meant to report their confidence in the annotations, their familiarity with biomedical terms encountered in the sample, and their comfort level during the task. These questions require the annotators to provide a number between 1 and 5 inclusive, where a 1 denotes the lowest and 5 denotes the highest level of confidence/familiarity/comfort level. The last question seeks whether the annotators used any off-the-shelf software in the case of finding unfamiliar terms in the sample. Table 10.8 shows the summary of the feedback we received from the annotators. Interestingly, although the annotators were highly confident before the start of the sessions, as it began, they report a decrease in their confidence. The confidence of Human 1, especially, is the lowest for the first session and this has been reflected in the results in Tables 10.6 and 10.7. However, both annotators regain their confidence in session 2. Overall, Human 2 is more confident with her annotation. The skewness of the datasets do not affect the humans as they are almost equally confident with candidate and non-candidate sentences. Both annotators report that their familiarity with biomedical terms is high and even if they needed help to resolve unfamiliar terms in the sample, they did not use any off-the-shelf software (e.g., Google or biomedical named entity recognizers). As a consequence, the human-machine agreements reported in Table 10.7 are not as high as human-human agreements. Some of this low performance can be attributed to Genia Tagger, used by the annotator tools, has its own limitations, as well. Its errors in not finding protein names are propagated to the annotator tools.

10.4.2 Attribute Selection

Each sentence in the five datasets is represented as $(\vec{x}, y)$, where $\vec{x} \in \mathbb{R}^{103}$ is a vector of the 103 stylometric attributes and $y \in \{$*candidate, non-candidate*$\}$ is the label of the sentence. Table 10.9 summarizes the attributes used in our experiment. The details of the attributes are as follows.

Text Complexity Attributes

Text complexity attributes examine the readability level of sentences. Two of the major components to measure text complexity are the use of simple words and complex words. Simple words are those that have at most two syllables while complex words contain three or more syllables. These two types of words are widely used to measure standard scores to assess the reading difficulty level of sentences. Among these scores, five developed by psychologists and linguists are considered as attributes in our experiment. These scores are Fog index, Smog index, Flesch reading ease score, Forcast, and Flesch-Kincaid index. Fog index measures the

No.	Question	Human 1	Human 2
1.	Confidence at the start of annotation	4	5
2.	Confidence with your annotation for the first session	2	3
3.	Confidence with your annotation for the second session	5	4
4.	Confidence with your annotation for both sessions	3	4
5.	Familiarity with the biomedical terms in the sample	4	5
6.	Comfort level for labelling a sentence as "candidate"	4	4
7.	Comfort level for labelling a sentence as "non-candidate"	5	4
8.	Use of any third party tool (e.g., google, biomedical taggers) if the category (e.g., protein) of any term in the sample was unknown	no	no

Table 10.8: Post-annotation feedback from the annotators.

proportion of complex words in a unit of text. In addition, we modify the Fog index formula to measure the relative use of simple words in a document and consider that as an attribute, too. Furthermore, we consider the arithmetic inverse of the Fog index as another attribute. The details of the attributes related to readability level are beyond the scope of this paper and are discussed elsewhere [20]. The other attributes in this group are quite self explanatory. They include average word length (i.e., the ratio of total syllables to total words in a sentence) and number of syllables present in a sentence. As well, we consider two BOOLEAN attributes named long and short sentence where long sentences are called those that contain more than 20 words, short sentence otherwise. There are 28 attributes in this group.

Word-level Attributes

Our second group of 68 attributes are related to the words of a sentence that do not contribute to text complexity. To calculate the word-level attributes, each sentence is treated as a bag of words. Word count, as its name suggests, is the length of each sentence. Several software functions are developed to deal with word-level attributes like function word, content word, biomedical Named Entity (NE), biomedical verb, and semantic word. For instance, our *stoplist function* utilizes a dictionary of function words used in the English language to categorize a word into either function or content word. Secondly, each word is checked with the Genia Biomedical Tagger [32] to identify biomedical named entities. Of note, the protein names in the datasets are masked and do not count as a biomedical named entity since *a priori* knowledge like the presence of proteins in a particular sentence perhaps bias our experiments. Thirdly, a *verb-list function* uses a dictionary of verbs that are frequently present in biomedical articles. Fourthly, our *semantic word list function* identifies words that are nodes in the UMLS semantic

Group	Attribute Nos.	Attributes
Text Complexity	28	Fog Index, Monosyllabic Fog Index, Inverse Fog Index, FORCAST, SMOG, FKRI, Flesch, Complex Word, Simple Word, Word Length, Syllable Frequency, Long Sentence (boolean), Short Sentence (boolean)
Word-level	68	Word Frequency, Function Word, Content Word, Named Entity, Alpha-numeric, Biomedical Verb, Acronym, Semantic Word, All-alphabets, All-numeric, Lowercase, Uppercase, Long Word, Unique Word, Repeated Word, Conjunction, Number, Determiner, Preposition, Adjective, Noun, Pronoun, Adverb, Verb, Interjection, Foreign, List, Possessive, Particle, Symbol
Character-level	7	Character Frequency, Character per Word, Character without Whitespace, Special character

Table 10.9: Set of stylometric attributes used in the experiment.

relation network [36] (i.e., one of the 133 semantic types in the network). This software function requires WordNet [37] to find words with similar senses. Finally, to identify acronyms, a dictionary containing the biomedical acronyms is exploited using an *acronym-list function*. The other word-level attributes are self explanatory. They include alpha-numeric, all-alphabets, all-numeric, lowercase, uppercase, long word (i.e., words with more than five characters), unique word (used only once in a sentence), and repeated word (used multiple times in a sentence). As well, we also consider the POS of the words as attributes. In doing so, the Stanford POS tagger is used to tag words with their POS. The set of POS-based attributes includes conjunction, number, determiner, preposition, adjective, noun, pronoun, adverb, verb, interjection, foreign, list, possessive, particle, and symbol. The details about these POS tags can be found in the Penn Treebank's tag description [38]. Note that the bulk of the attributes used in this experiment come from this group.

Character-level Attributes

We also extract several character-level attributes such as total characters including and excluding whitespaces, characters per word, and special characters. The number of attributes that we consider from this group is 7.

10.4.3 Classifier Design

To generate classifiers for each of the datasets, we use an interesting idea of multi-stage classification or cascading classifiers [39] [40] which are available in the Weka machine learning toolkit [41]. Cascading classification is a special case of *ensemble learning*. Here, more than one classifier is concatenated by utilizing the information found from the output of any given classifier as additional information for the next classifier in the cascade. One key difference between cascading classifiers and voting or stacking ensembles is that the former group is multi-stage learning while the latter are multiexpert learning methods. This multi-stage de-

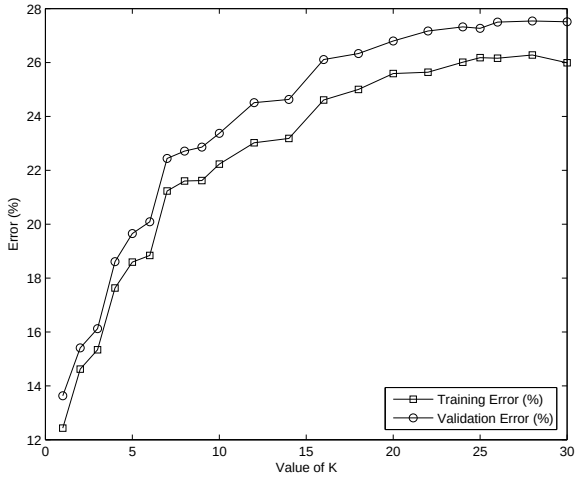


Figure 10.4: Learning curves to show the best value of κ for κ-Nearest Neighbour classifier.

sign of classifier generation is chosen since it is often faster as well as better-performing than ensemble learners and only requires simple learners in its cascade.

In this experiment, we design a two-stage classification. The first stage involves a κ-Nearest Neighbour classifier (with $\kappa = 1$) [7] that considers each vector of attributes representing the sentences in each of the datasets. Then, the classifier calculates the probabilities that the vector belongs to the candidate and non-candidate classes. These probabilities, which are normally used by the classifier model to predict the label of the vector by classifying the vector to the class with the highest probability, are then fed forward as extra attributes to the Naïve Bayes classifier [15] at the second stage. The choice of κ is made based on the learning curve. We vary the value of κ from 1 to 30 and plot the training and validation error rates made by the Naïve Bayes classifier. The training and validation error rates made by the Naïve Bayes classifier are calculated using the following equation.

$$\textit{Error Rate} = \frac{N_{x\to c} + N_{c\to x}}{N_{c\to c} + N_{x\to c} + N_{c\to x} + N_{x\to x}}. \tag{10.3}$$

From the learning curve in Figure 10.4, we find that with $\kappa = 1$ both the training and validation error are the lowest. Therefore, our first stage in the cascade becomes a 1-Nearest Neighbour classifier.

The 1-Nearest Neighbour classifier uses normalized Euclidean distance to find the training instance closest to the given test instance and predicts the same class as this training instance. If multiple instances have the same (smallest) distance to the test instance, the first one found is used. The Naïve Bayes classifier uses estimator classes. Weka chooses numeric estimator precision values based on an analysis of the training data and therefore the version of Naïve Bayes that we use in this experiment is not updatable. The parameters used in our experiment to design the multi-stage classifiers are provided in Table 10.10.

Classifier	Parameters
1-Nearest Neighbour	No. of neighbours: 1
	Best κ value selection: Using Training Data
	Distance weighting: None
	Nearest neighbour search algorithm: Linear NN Search (Distance function: Euclidean, Skip identical instances: False)
	Window size: 0
Naïve Bayes	Kernel estimator: None
	Supervised discritization: Yes

Table 10.10: Parameter setup for the learning algorithms used in this experiment to generate classifiers.

		Actual	
		Candidate	Non-candidate
Prediction	Candidate	$N_{c\to c}$	$N_{x\to c}$
	Non-candidate	$N_{c\to x}$	$N_{x\to x}$

Table 10.11: Confusion matrix for the protein interaction sentence classification problem.

10.4.4 Experimental Procedure

Treating each dataset independently, the values for the 103 stylometric attributes are computed for each sentence. These attributes are then used to generate a 1-Nearest Neighbour model. This model, when applied on the entire dataset, produces two probability distributions of candidate and non-candidate sentences. These distributions are then used as two additional attributes at the next stage, and altogether, the 105 attributes are then used to generate a Naïve Bayes model. Performance is tested using a stratified 10-fold cross-validation approach. That is, the sentences of each dataset are randomly divided into 10 equal-sized sets. Each one is used for evaluation and the rest for construction of a Naïve Bayes model. Stratification means that the classes (i.e., candidate and non-candidate) in each set are represented in approximately the same proportion as in the full dataset. This cross-validation process is then repeated until each of the 10 folds is used exactly once as the validation data. Finally, average evaluation values of the 10 tests are reported.

10.4.5 Evaluation Measures

To evaluate the performance of our approach, we report cost-insensitive measures like precision, recall, and F-score and cost-sensitive measures such as Cost Curves and Receiver Operator Characteristics (ROC) curves. All of the measures reported in this paper depend on the confusion matrix given in Table 10.11.

Cost-insensitive Measures

The binary classifiers predict the class of each sentence in the datasets as either a candidate or non-candidate sentence. Then its *precision* is calculated as the number of correct candidate sentence predictions divided by the total number of sentences it labels as candidate sentences. The *recall* of the classifier, on the other hand, is calculated as the number of correct candidate sentence predictions divided by the number of candidate sentences. To average the precision and recall, their harmonic mean is considered which is called the *F-score*. The calculations are as follows:

$$Precision = \frac{N_{c\to c}}{N_{c\to c} + N_{x\to c}}, \tag{10.4}$$

$$Recall = \frac{N_{c\to c}}{N_{c\to c} + N_{c\to x}}, \tag{10.5}$$

$$F\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall}. \tag{10.6}$$

Cost-sensitive Measures

Table 10.2 indicates that other than the CB dataset, all of the datasets have class imbalance problems: they have many more non-candidate sentences than candidate sentences. Besides precision, recall, and F-score, we report the performances using two other cost-sensitive evaluation measures, namely the *Cost Curves* [26] [27] and the *Receiver Operating Characteristics* (ROC) curves. They are related. Indeed, a cost curve is the projection of the slopes in the corresponding ROC curve on a plane's *X-axis* while the *Y-axis* represents the expected misclassification cost. Cost curves provide visualizations of binary classifier performance over the full range of possible class distributions and misclassification costs. That is, for imbalanced datasets, using these curves we can predict how good a given classifier would perform if the class distributions were equal—50% of the sentences are candidate and 50% of the sentences are non-candidate sentences. On the other hand, ROC curves plot the recall vs. the false positive rates (FPR). Recall can be found from Equation 10.5 and FPR denotes the total non-candidate sentences misclassified by the classifiers divided by the total non-candidate sentences. The latter can be calculated as follows:

$$False\ Positive\ Rate = \frac{N_{x\to c}}{N_{x\to c} + N_{x\to x}}. \tag{10.7}$$

ROC curves are particularly helpful to observe the gain in recall with the compromise of FPR of binary classifiers for imbalanced datasets (see [42] for details).

Bibliography

[1] A. Zanzoni, L. Montecchi-Palazzi, M. Quondam, G. Ausiello, M. Helmer-Citterich, and G. Cesareni, "MINT: A Molecular INTeraction database," *FEBS Letters*, vol. 513, no. 1, pp. 135–140, 2002.

[2] G. D. Bader, I. Donaldson, C. Wolting, B. F. F. Ouellette, T. Pawson, and C. W. V. Hogue, "BIND — The biomolecular interaction network database," *Nucleic Acids Research*, vol. 29, no. 1, pp. 242–245, 2001.

[3] A. Bairoch, B. Boeckmann, S. Ferro, and E. Gasteiger, "Swiss-Prot: Juggling between evolution and stability," *Briefings in Bioinformatics*, vol. 5, no. 1, pp. 39–55, 2004.

[4] M. E. Cusick, H. Yu, A. Smolyar, K. Venkatesan, A.-R. Carvunis, N. Simonis, J.-F. Rual, H. Borick, P. Braun, M. Dreze, J. Vandenhaute, M. Galli, J. Yazaki, D. E. Hill, J. R. Ecker, F. P. Roth, and M. Vidal, "Literature-curated protein interaction datasets," *Nature Methods*, vol. 6, pp. 39–46, Jan. 2009.

[5] T. Polajnar, T. Damoulas, and M. Girolami, "Protein interaction sentence detection using multiple semantic kernels," *Journal of Biomedical Semantics*, vol. 2, no. 1, pp. 1–18, 2011.

[6] K. Sugiyama, K. Hatano, M. Yoshikawa, and S. Uemura, "Extracting information on protein-protein interactions from biological literature based on machine learning approaches," *Genome Informatics*, vol. 14, pp. 699–700, 2003.

[7] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Machine Learning*, vol. 6, pp. 37–66, Jan. 1991.

[8] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[9] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2nd ed., 1998.

[10] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[11] T. Mitsumori, M. Murata, Y. Fukuda, K. Doi, and H. Doi, "Extracting protein-protein interaction information from biomedical text with SVM," *IEICE Transactions*, vol. 89-D, no. 8, pp. 2464–2466, 2006.

[12] G. Erkan, "Semi-supervised classification for extracting protein interaction sentences using dependency parsing," in *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 228–237, 2007.

[13] T. Polajnar, S. Rogers, and M. Girolami, "Classification of protein interaction sentences via Gaussian processes," in *Pattern Recognition in Bioinformatics* (V. Kadirkamanathan, G. Sanguinetti, M. Girolami, M. Niranjan, and J. Noirel, eds.), vol. 5780 of *Lecture Notes in Computer Science*, pp. 282–292, Berlin, Germany: Springer, 2009.

[14] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. Cambridge, MA, USA: The MIT Press, 2005.

[15] G. H. John and P. Langley, "Estimating continuous distributions in Bayesian classifiers," in *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, (San Francisco, CA, USA), pp. 338–345, Morgan Kaufmann Publishers Inc., 1995.

[16] T. Polajnar and M. A. Girolami, "Semi-supervised prediction of protein interaction sentences exploiting semantically encoded metrics," in *Pattern Recognition in Bioinformatics* (V. Kadirkamanathan, G. Sanguinetti, M. A. Girolami, M. Niranjan, and J. Noirel, eds.), vol. 5780 of *Lecture Notes in Computer Science*, (Berlin, Germany), pp. 270–281, Springer, 2009.

[17] A. Yakushiji, Y. Miyao, Y. Tateisi, and J. Tsujii, "Biomedical information extraction with predicate-argument structure patterns," in *Proceedings of the 11th Annual Meeting of the Association for Natural Language Processing*, pp. 60–69, 2005.

[18] T. M. Duffy and P. Kabance, "Testing a readable writing approach to text revision," *Journal of Educational Psychology*, vol. 74, no. 5, pp. 733–748, 1982.

[19] R. Shams and R. E. Mercer, "Extracting connected concepts from biomedical texts using fog index," *Procedia — Social and Behavioral Sciences*, vol. 27, pp. 70–76, 2011.

[20] R. Shams and R. E. Mercer, "Classifying spam emails using text and readability features," in *Proceedings of the 2013 IEEE International Conference on Data Mining (ICDM2013)*, (Texas, USA), pp. 657–666, IEEE, 2013.

[21] R. Shams and R. E. Mercer, "Personalized spam filtering using natural-language attributes," in *Proceedings of the 12th IEEE International Conference on Machine Learning Applications (ICMLA2013)*, (Miami, USA), pp. 127–132, IEEE, 2013.

[22] J.-D. Kim, S. Pyysalo, T. Ohta, R. Bossy, N. Nguyen, and J. Tsujii, "Overview of BioNLP Shared Task 2011," in *Proceedings of the BioNLP Shared Task 2011 Workshop*, (Portland, Oregon, USA), pp. 1–6, Association for Computational Linguistics, June 2011.

[23] R. Prasad, S. McRoy, N. Frid, A. Joshi, and H. Yu, "The biomedical discourse relation bank," *BMC Bioinformatics*, vol. 12, no. 1, pp. 188+, 2011.

[24] K. Franzën and D. Oppenheimer, "The FetchProt corpus: Documentation and annotation guidelines," tech. rep., Swedish Institute of Computer Science, Sweden, 2007.

[25] S. I. Faiz and R. E. Mercer, "Identifying explicit discourse connectives in text," in *26th Canadian Conference on Artificial Intelligence (CAI 2013)* (O. R. Zaïane and S. Zilles, eds.), vol. 7884 of *Lecture Notes in Computer Science*, (Regina, Canada), pp. 64–76, Springer, 2013.

[26] C. Drummond and R. C. Holte, "Cost curves: An improved method for visualizing classifier performance," *Machine Learning*, vol. 65, no. 1, pp. 95–130, 2006.

[27] R. C. Holte and C. Drummond, "Cost-sensitive classifier evaluation using cost curves," in *12th Pacific-Asia Conference (PAKDD)* (T. Washio, E. Suzuki, K. M. Ting, and A. Inokuchi, eds.), vol. 5012 of *Lecture Notes in Computer Science*, pp. 26–29, 2008.

[28] J. Landis and G. Koch, "Measurement of observer agreement for categorical data," *Biometrics*, vol. 33, no. 1, pp. 159–174, 1977.

[29] K. Fundel, R. Küffner, and R. Zimmer, "RelEx — relation extraction using dependency parse trees," *BMC Bioinformatics*, vol. 23, no. 3, pp. 365–371, 2007.

[30] M. catherine De Marneffe, B. Maccartney, and C. D. Manning, "Generating typed dependency parses from phrase structure parses," in *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pp. 449–454, 2006.

[31] "Alphabetical list of part-of-speech tags used in the penn treebank project." http://alias-i.com/lingpipe/demos/tutorial/sentences/read-me.html.

[32] Y. Tsuruoka, Y. Tateishi, J.-D. Kim, T. Ohta, J. McNaught, S. Ananiadou, and J. Tsujii, "Developing a robust part-of-speech tagger for biomedical text," in *Advances in Informatics* (P. Bozanis and E. N. Houstis, eds.), vol. 3746, pp. 382–392, 2005.

[33] J. Carletta, "Assessing agreement on classification tasks: The kappa statistic," *Computational Linguistics*, vol. 22, pp. 249–254, June 1996.

[34] K. Krippendorff, *Content Analysis: An Introduction to Its Methodology (second edition)*. 2004.

[35] G. S. Mudholkar, *Fisher's Z-Transformation*. New Haven: John Wiley and Sons, Inc., 2004.

[36] A. T. McCray, "Representing biomedical knowledge in the UMLS semantic network," in *High Performance Medical Libraries* (N. C. Broering, ed.), pp. 45–55, Westport, CT, USA: Meckler Corporation, 1993.

[37] G. A. Miller, "WordNet: A lexical database for English," *Communications of the ACM*, vol. 38, pp. 39–41, Nov. 1995.

[38] B. Santorini, "Part-Of-Speech tagging guidelines for the Penn Treebank project (3rd revision, 2nd printing)," tech. rep., Department of Linguistics, University of Pennsylvania, Philadelphia, PA, USA, 1990.

[39] P. A. Viola and M. J. Jones, "Rapid object detection using a boosted cascade of simple features.," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, vol. 1, pp. 511–518, 2001.

[40] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal on Computer Vision*, vol. 57, pp. 137–154, May 2004.

[41] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *SIGKDD Explorations Newsletter*, vol. 11, pp. 10–18, Nov. 2009.

[42] T. Fawcett, "ROC graphs: Notes and practical considerations for researchers," tech. rep., HP Laboratories Palo Alto, 2004.

Chapter 11

Summary Sentence Classification using Stylometry

This chapter is based on the paper titled "Summary Sentence Classification using Stylometry" co-authored with Robert E. Mercer. We are expecting to submit the paper in the 2015 North American Chapter of Association of Computational Linguistics (NAACL) Conference.

Summary sentence classification is an important step to generate document surrogates known as summary extracts. The quality of an extract depends much on the correctness of this step. We aim to classify summary sentences using a statistical learning method that models sentences according to a linguistic technique which examines writing styles, known as *Stylometry*. The sentences in documents are represented using a novel set of stylometric attributes. For learning, an innovative two-stage classification is set up that comprises two learners in subsequent steps: κ-Nearest Neighbour and Naïve Bayes. We train and test the learners with the newswire documents collected from two benchmark datasets, viz., the `CAST` and the `DUC2002` datasets. Extensive experimentation strongly suggests that our method has outstanding performance for the single document summarization task. However, its performance is mixed for classifying summary sentences from multiple documents. Finally, comparisons show that our method performs significantly better than most of the state-of-the-art summarization methods.

11.1 Introduction

One of the challenges to lessen the effect of today's *information overload* [1] is producing summaries automatically that are short in length, concise in nature, and rich in language properties [2]. In general, summaries are produced either from a single document or from multiple documents. *Extracts* are summaries that are constructed by using *copy and paste* of text units such as clauses, sentences, and paragraphs from source texts [3]. On the other hand, *abstracts* are summaries that are re-generated by relating these text units using textual cohesion and entailment. Overall, an extract is useful to understand the main idea of a source text while its abstract is better for conceptualizing the subject matter [4]. Empirically, the summarizers that simply extract salient sentences from source texts perform better for large-scale applications and therefore receive much of the attention [5]. Keeping this in mind, we put the focus of this paper on extracts—and not on abstracts—that are generated from both single and multiple

documents.

Summary sentences are sentences chosen as text units to be included in extracts [3]. The classification of summary sentences from source texts can be seen as a binary classification problem in statistical learning. Over the last decade, an array of methods have been proposed on this topic and on a good note, many are regarded as the domain's state-of-the-art [5]. These methods use classical attributes, viz., rhetorical structure [4], sentence position in the text [6], and the presence of keyphrases in sentences [7], as well as novel attributes, such as, lexical chains [8], *n*-gram models [9], and semantic graphs [10] [11].

In their study, Lloret and Palomar [12] showed that to classify summary sentences a linguistic theory named the *Code Quantity Principle* [13] can be considered. According to this principle, the codification of important information in text by humans so that this information gets more attention is a process that combines human cognition, psychology, and language. The principle states that the most important information within a text contains more lexical units such as syllables, words, and phrases [14]. These units, and many others, are also the key elements of *stylometry*—the application of the linguistic study to model writing styles in texts. Now, by definition summary sentences are more important than non-summary sentences. Therefore, modelling them using stylometry should result in a good classification.

In this paper, we report on a method to classify summary sentences using stylometry. We model the sentences of a set of documents using 87 stylometric attributes and use these attributes as features for a two-stage classification technique comprising the κ-Nearest Neighbour and Naïve Bayes classifiers. The documents are collected from two benchmark datasets for single and multi-document text summarization named the `CAST` datasets [15] and the `DUC2002` collection [16]. Results show that our method has outstanding performance for single document summarization; the performance for multi-document summarization is somewhat mixed. Comparisons show that the proposed method yields much better results than most of the state-of-the-art summarization methods.

The next section describes work related to this research. Following that, Section 11.3 discusses the datasets used and outlines the methods we followed. Results of this work can be found in Section 11.4. Finally, Section 11.5 draws conclusion to this paper.

11.2 Related Work

Many studies have reported state-of-the-art summary sentence classification methods. We, however, limit our discussions by relating our work to those that have used the same summarization data as ours.

Berker and Güngör [8] proposed a very simple yet powerful method to generate extracted summaries. They used lexical chains to represent lexical cohesion in a document as attributes. The attribute values were then weighted according to a genetic algorithm. These weights were then used to rank order the sentences in the document. Finally, the sentences with higher ranks were selected for the summary. To train and test the method, the `CAST` dataset was used. Overall, the method worked remarkably well. Importantly, the authors compared the performance of lexical chain attributes with that of classical summarization attributes such as *sentence location and length, similarity to title, synonymy, and co-occurrence*. The outcomes of this comparison were interesting and the study concluded that lexical chains can be an

effective means to classify summary sentences.

In their extensive study, Villatoro-Tello *et al.* [9] argued that most frequency-based attributes (i.e., sentence position, word frequency, or cue words) are domain-specific. Instead of these well-known attributes, they used an *n*-gram model to represent sentences. Note that *n*-gram models are widely used for text categorization tasks, and this work is the first that attempts to use them to generate text summaries. The results of the models were competitive on the CAST dataset. The authors also compared their model with a baseline and a *single word* model. In both cases, the *n*-gram model performed better.

Lescovek *et al.* [10] proposed an interesting method: to represent documents as semantic graphs. A semantic graph of a document, known as the *document graph*, is a combination of the logical forms of its sentences. The logical forms are merely the popular subject-predicate-object (*SPO*) representation of a text unit. For each document and its summary, the method generated two semantic graphs—one for the original document and one for its summary produced by human accessors. A Support Vector Machine (*SVM*) classifier was trained on the document graphs to identify the *SPO* triples in them that are also present in the corresponding summaries. Finally, the classifier was evaluated on test documents. To train and test this graph-based method, the CAST [15] and the DUC [16] were used. The outcomes of this study were remarkable and showed an overall improved summary sentence classification. Later, the authors re-generated the document graphs using the linguistic structures around the noun and verb phrases as additional components to the *SPO* triples (see [11]). Similar to their first experiment, these additional components also improved their method's overall performance.

The Columbia summarizer is a very well-known operational text summarizer [17]. It has been a regular participant in the Document Understanding Conferences (DUC) and is recognized as one of the benchmark summary tools. The summarizer has two major components: MultiGen [18] to generate summaries from single-event documents and Dissimilarity Engine for Multidocument Summarization [19] to generate summaries from documents on multiple events. It has been evaluated on the DUC summary sentence extraction tasks. On an average, its performance was in the top three among the 10 submitted systems.

11.3 Materials and Methods

11.3.1 Summarization Data

We have used two benchmark datasets in this experiment. One of them has summary sentences from single documents while the other contains summary sentences from multiple documents. Nevertheless, documents included in both of the datasets are written in English and are newswire articles on different topics like politics, economics, sports, and natural disasters.

The CAST dataset [15] was developed as a part of the Computer-Aided Summarisation Tool project to aid humans to generate *single document summaries* from newswire articles. Overall, the dataset contains 147 newswire articles from the Reuters Corpus [20]. Four human annotators were used to mark 15% of the document sentences as *essential* and an additional 15% as *important* for the summary. In this study, we call the first set of documents CAST-15 each of which contains 15% of their sentences marked as *essential* and the second set of documents CAST-30 each of which contains 30% of their sentences marked as *essential* and *important*.

Dataset	Total Sentences	Summary Sentences (+)	Regular Sentences (−)	Imbalance Ratio
CAST-15	2,579	315	2,264	7.2
CAST-30	2,579	667	1,912	2.9
DUC-200	20,800	491	20,309	41.4
DUC-400	20,800	667	1,912	20.8

Table 11.1: Brief summary of the datasets used in this experiment.

The documents were arbitrarily distributed to the annotators, therefore for some documents there were multiple sets (up to three) of selected sentences while for others there was only one. Considering this, we have selected the set of 89 documents that had a single annotator called *Annotator-1* and have modelled the sentences in the CAST-15 and CAST-30, separately. Out of 2,579 sentences, the CAST-15 and CAST-30 datasets contain 315 and 667 sentences labeled as *summary sentences*, respectively (Table 11.1).

Our second dataset is from one of the dataset collections prepared for the Document Understanding Conference 2002, widely known as the DUC2002 collection [16]. This collection is highly regarded as the benchmark for *multi-document summarization*. In this collection, one of the datasets contains 59 clusters of 533 newswire articles on 30 different topics. The sources of the articles include but are not limited to the Financial Times, the Wall Street Journal, and the Associated Press. The articles in each cluster belong to the same topic. Each cluster has a 200-word and a 400-word summary interpreted as *extracted summaries* since the sentences in the summaries are extracted directly from the articles in the cluster. Note that this dataset was not used in the official DUC evaluation since the primary focus of the conference was on abstracts rather than extracted summaries. In this experiment, DUC-200 and DUC-400 are the datasets with the 200-word and the 400-word summaries, respectively. Out of 20,800 sentences, the DUC-200 dataset has 491 sentences and the DUC-400 dataset has 956 sentences labeled as *summary sentences*. Note that the imbalance ratio of these two datasets is much higher than that of the CAST datasets (Table 11.1).

Treating each dataset in Table 11.1 independently, we have extracted all the sentences from its articles and put them in one single document. This process of *flattening* of the dataset makes the computing of the stylometric attributes comfortable.

11.3.2 Attributes for Summary Sentence Classification

We have represented each sentence in the datasets as $(\vec{x}, y)$, where $\vec{x} \in \mathbb{R}^{87}$ is a vector of the 87 stylometric attributes and $y \in \{+, -\}$ is the sentence label. A sentence has the label +, if it is a summary sentence, − otherwise. The labels are provided by the human annotators of the CAST and DUC datasets.

A brief list of the stylometric attributes used in this research can be found in Table 11.2. Details of these attributes are provided in the following discussion.

No.	Name	No.	Name	No.	Name	No.	Name
A1	Fog Index	A23	Word Length	A45	Numeric WSW %	A67	Number
A2	Fog Index WSW	A24	Word Length WSW	A46	Lowercase	A68	Determiner
A3	Simple Word Fog Index	A25	Syllable Count	A47	Lowercase %	A69	Preposition
A4	Simple Word Fog Index WSW	A26	Syllable Count WSW	A48	Lowercase WSW	A70	Adjective
A5	Inverse Fog Index	A27	Long Sentence	A49	Lowercase WSW %	A71	Noun
A6	Inverse Fog Index WSW	A28	Short Sentence	A50	Uppercase	A72	Pronoun
A7	FORCAST	A29	Word Count	A51	Uppercase %	A73	Adverb
A8	FORCAST WSW	A30	Function Word	A52	Uppercase WSW	A74	Verb
A9	SMOG Index	A31	Function Word %	A53	Uppercase WSW %	A75	Interjection
A10	SMOG Index WSW	A32	Content Word	A54	Long Word	A76	Foreign
A11	FKRI	A33	Content Word %	A55	Long Word %	A77	List
A12	FKRI WSW	A34	Alphanumeric	A56	Long Word WSW	A78	Possessive
A13	Flesch	A35	Alphanumeric %	A57	Long Word WSW %	A79	Particle
A14	Flesch WSW	A36	Alphanumeric WSW	A58	Unique Word	A80	Symbol
A15	Complex Word	A37	Alphanumeric WSW %	A59	Unique Word %	A81	Character
A16	Complex Word %	A38	Alphabets	A60	Unique Word WSW	A82	Character WSW
A17	Complex Word WSW	A39	Alphabets %	A61	Unique Word WSW %	A83	Character per Word
A18	Complex Word WSW %	A40	Alphabets WSW	A62	Repeated Word	A84	Character per Word WSW
A19	Simple Word	A41	Alphabets WSW %	A63	Repeated Word %	A85	Character without Space
A20	Simple Word %	A42	Numeric	A64	Repeated Word WSW	A86	Character without Space WSW
A21	Simple Word WSW	A43	Numeric %	A65	Repeated Word WSW %	A87	Special Character
A22	Simple Word WSW %	A44	Numeric WSW	A66	Conjuction	A88, A89	Positive/Negative Distribution

Table 11.2: The list of attributes used in this research. This list comprises stylometric attributes like text complexity attributes (A1–A28), word-level attributes (A29–A80), and character-level attributes (A81–A87). The attributes A88 and A89 are the positive and negative distributions provided by the κ-Nearest Neighbor classifier and used in the second stage of our two-stage classification approach. WSW means *without stopwords*.

Text Complexity Attributes

Every sentence has a level of reading complexity. To gauge it, there are many *de facto* standard scores such as Fog Index, Smog Index, Flesch Reading Ease Score, Forcast, and Flesch-Kincaid Index. Calculation of these scores is centred around the use of simple and complex words in a text unit. Note that in contrast to simple words that have at most two syllables, complex words contain three or more syllables. In this study, we have considered the aforementioned text complexity scores as well as the frequency of simple and complex words as attributes for summary sentence classification. In addition, contrary to the Fog Index that measures the proportion of complex words in a text unit, we have chosen the relative use of simple words in a sentence as an attribute (referred to as the *Simple Word Fog Index* attribute in Table 11.2). Another such modified attribute we have considered is the *Inverse Fog Index* which, as the name suggests, is the arithmetic inverse of the Fog Index. The details of the attributes—especially the readability scores—are beyond the scope of this paper; a description of the readability scores can be found in [21]. The rest of the attributes in this category are quite self explanatory: the average word length is the average of syllables present in the words of a sentence and the number of syllables is simply the count of syllables in a sentence. Finally, we have considered two `boolean` attributes: long and short sentence, where long sentences are composed of more than 20 words. In this category, we have selected 28 attributes that contribute to the complexity of text.

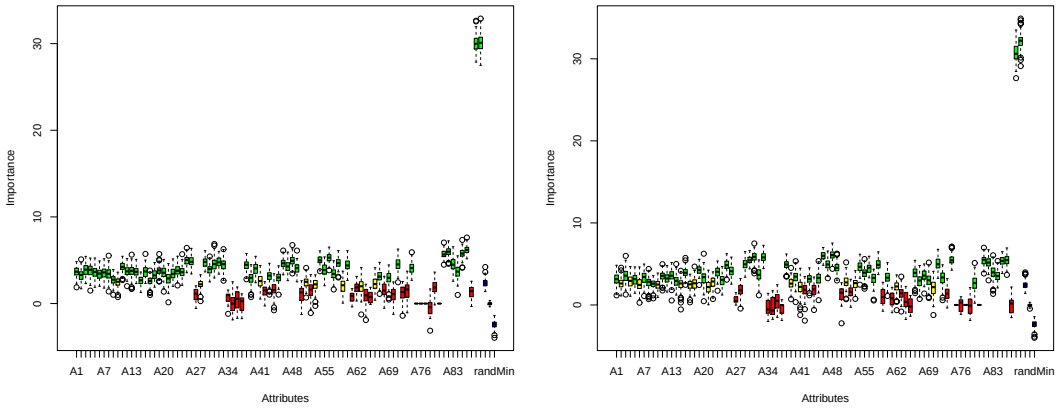
Word-level Attributes

This category contains the bulk of the attributes we have selected, 52 to be specific. During the calculation of these attributes, we have considered each sentence as a bag-of-words. Most of the attributes in this category are self explanatory. Word count, as its name suggests, is the length of a sentence. We have utilized a standard English *function word lexicon* to distinguish between function and content words. Other attributes in this category are alpha-numeric, all-alphabets, all-numeric, lowercase, uppercase, long word (i.e., the percentage of words with more than five characters), unique word (i.e., the percentage of words used only once in a sentence), and repeated word (i.e., the percentage of words used multiple times in a sentence). Note that, our notion of alpha-numeric word is not commonplace as we have specified a word as being alpha-numeric only if it contains both alphabetics and numerics. In addition, we have used the Stanford part-of-speech tagger [22] to tag the each word's part-of-speech. The set of part-of-speech based attributes includes conjunction, number, determiner, preposition, adjective, noun, pronoun, adverb, verb, interjection, foreign, list, possessive, particle, and symbol. Details regarding these part-of-speech tags can be found in the Penn Treebank's tag description [23].

Character-level Attributes

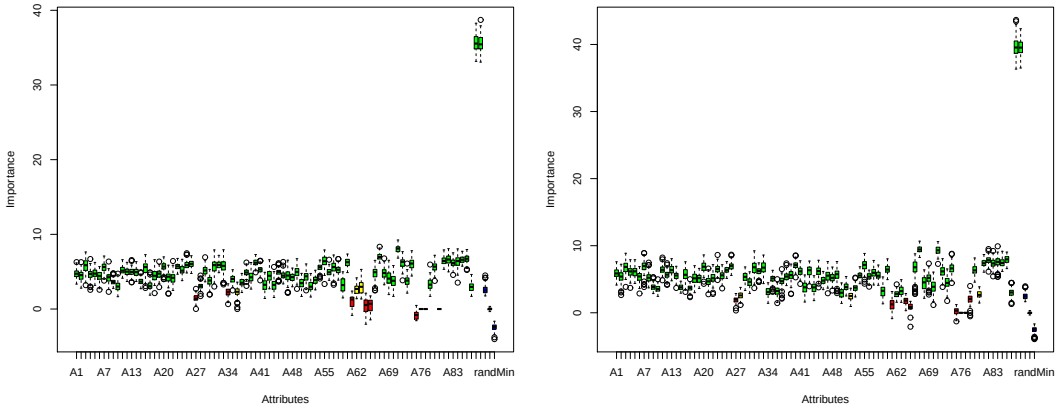
Our last category of attributes includes several character-level attributes such as total characters including and excluding whitespaces, characters per word, and special characters. The number of character-level attributes we have considered for this study is 7.

Note that we have calculated the attribute values with and without function words. The exceptions are the long/short sentence, word frequency, function word, content word, parts-



(a) Importance of attributes for the CAST-15 dataset.

(b) Importance of attributes for the CAST-30 dataset.



(c) Importance of attributes for the DUC-200 dataset.

(d) Importance of attributes for the DUC-400 dataset.

Figure 11.1: Attribute importance for the datasets using the Boruta algorithm.

of-speech, and special character attributes which are calculated without removing the function words.

Attribute Importance

Determining salient attributes to decide the class of data instances is highly recommended in machine learning methods [24] [25]. Therefore, to observe the importance of the stylometric attributes, we used a state-of-the-art algorithm named Boruta[1]. This unique algorithm uses a wrapper around the Random Forest classification algorithm and iteratively removes attributes which are proven to be less important than random probes according to a statistical test. The

[1] http://cran.r-project.org/web/packages/Boruta/index.html

selection process is both unbiased and stable, and its usefulness has been successfully demonstrated by Kursa and Rudnicki [24] on an artificial dataset.

One nice feature of the algorithm's implementation is its representation of the attributes and their importance as a 2−dimensional *box plot*. In the plot, a point on the `x`-axis represents an attribute and its corresponding point on the `y`-axis refers to its importance in quantitative terms. In addition, the *green* boxes represent the attributes that are *important* while the *red* boxes denote the attributes that are *unimportant*; the *yellow* boxes merely refer to attributes that are neither important nor unimportant.

We have applied `Boruta` on all of our datasets and the results are illustrated in Figure 11.1 (for the attribute numbers in the figure, refer to Table 11.2). Surprisingly, they demonstrate the strength of the character-level attributes. For all four datasets, most attributes from this category have the highest *importance*. The only exception is the *special character* attribute: although it has some importance for the `DUC` datasets, it is found unimportant for the `CAST` datasets. Interestingly, part-of-speech attributes like *interjection, foreign*, and *list* are unimportant for all the datasets. A careful examination shows that none of the datasets have any word from these categories. Additionally, there are other attributes that are found to be unimportant across all the datasets: *long sentence*, percentage of *unique words* without function words, and frequency and percentage of *repeated words* without function words. It was surprising to us since we had expected at least *long sentence* to be a strong attribute because of the length constraint put on the summary extracts during the dataset curation. Attributes related to *alpha-numerics, numerics, uppercase, adverbs*, and *symbols* are found to be unimportant for the `CAST` datasets. Interestingly, *nouns* are among the most important attributes for the `DUC` datasets and *verbs* are found to be strong for the `CAST-30` dataset.

Note that the most important attributes in Figure 11.1 (A88 and A89) are the two *data distributions* found from the κ-nearest neighbour classifier in our two-stage classification. Details regarding these two highly important attributes can be found in Section 11.3.3.

11.3.3 Classifier Design

Multi-stage or cascading learning [26] [27] is a special case of *ensemble learning*. As the name suggests, several classifiers $C_1, C_2, ..., C_{n-1}, C_n$ are staged serially so that C_n learns not only from the attributes of the training instances but also from the class distributions of these instances provided by C_{n-1} (e.g., if C_{n-1} is a decision tree, the class distributions are probability values of class membership for the training instances). In contrast to this multi-stage learning, voting or stacking ensembles are multi-expert learning methods. Multi-stage learning methods are often as fast and as good as ensemble learners but only require simple learners in their cascades.

To classify summary sentences, we have designed a two-stage learner: our first stage involves a κ-Nearest Neighbour learner [28] while our second stage is a Naïve Bayes learner [29]. There are several reasons for choosing these learners for our two-stage classification. First, both learners are *stable*: a small change in the training data rarely affects performance. Second, we are interested in exploiting the strengths of both discriminative (κ-Nearest Neighbour) and generative (Naïve Bayes) learners. Third, both learners are simple as their objective functions are based on probabilities. Last but not least, both learners—especially Naïve Bayes—perform well in text-based classification. We have used the implementations and the default parameter

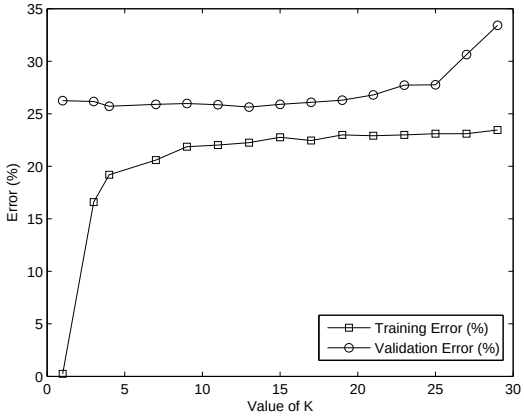


Figure 11.2: The learning curve of the κ-Nearest Neighbour classifier on the `CAST-30` dataset used to determine the optimum value of κ.

values of these learners found in the Weka machine learning toolkit [30].

The overall learning is evaluated using a stratified 10-fold cross validation. Treating each dataset independently, the values for the 87 stylometric attributes are computed for each sentence. Then, each dataset is randomly divided into 10 equal-sized stratified sets. Stratification means that the + and − classes in each set are represented in approximately the same proportion as in the full dataset. One set is used for evaluation and the remaining sets for construction of a κ-Nearest Neighbour classifier (stage 1 of 2). This classifier then generates two probability values (one for each class) for each instance in the evaluation set. This cross-validation process is then repeated until each of the 10 sets is used exactly once as the validation data for the κ-Nearest Neighbour classifier. Now, in addition to the 87 stylometric attributes, each instance in the dataset has been assigned two more attributes. So, each instance is now represented by 89 attributes and and each has the human-assigned class attribute. Using these attributes, a Naïve Bayes classifier then generates models in a stratified 10-fold cross validation (stage 2 of 2). We report the average values for the 10 folds of the measures described in Section 11.3.4.

Finding a good value of κ for a κ-Nearest Neighbour classifier is important since a too low κ value usually generates a *low bias-high variance* classifier which may experience an *overfit*. On the other hand, a too high κ value may generate a *high bias-low variance* classifier which perhaps *underfits* the data. To find the correct value of κ, we have examined the bias-variance tradeoff using a *learning curve* where training and validation error rates of the κ-Nearest Neighbour classifiers are plotted by varying the value of κ from 1 to 29; only the odd numbers in this range have been considered. As suspected, the learning curve in Figure 11.2 illustrates that low κ values generate *low bias-high variance* classifiers for the `CAST-30` dataset: the classifiers have very low training error but comparatively high validation error. However, according to the curve, we can expect to get a smooth decision boundary with $\kappa = 15$. The κ values for the remaining datasets are obtained in a similar way; we, however, have not included their learning curves that can be found elsewhere[2].

[2] `http://cogenglab.csd.uwo.ca/additionalmaterial/summary/text-summary-learning-curves-2014.zip`

		Actual	
		+	−
Prediction	+	True Positive (TP)	False Positive (FP)
	−	False Negative (FN)	True Negative (TN)

Table 11.3: Confusion matrix for summary sentence classification problem.

11.3.4 Evaluation Measures

To summarize the performances of the classifiers, we have used a wide variety of standard evaluation measures. The measures include precision, recall, F-score, accuracy, false positive rate, false negative rate, area under curve (AUC), and the Matthews correlation co-efficient. Noting that all of the datasets we have used suffer from a high class imbalance ratio (see Table 11.1), we have selected the measures because all except accuracy can deal with the class imbalance problem.

To understand the measures, refer to the confusion matrix shown in Table 11.3. The *precision* of classification is the fraction of instances correctly classified (into one of the two classes). Quantitatively, in our case it is the number of correct predictions for the summary sentence class divided by the total number of summary sentence predictions (Eq. 11.1). The *recall* (or true positive rate), on the other hand, is the fraction of relevant instances (for a class) that are correctly classified, which in our case is the number of correct predictions for the summary sentence class divided by the number of summary sentences in the dataset (Eq. 11.2). The *F-score* is the harmonic mean of the precision and the recall to represent their average (Eq. 11.3).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{11.1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{11.2}$$

$$\text{F-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{11.3}$$

The *accuracy* of a method represents the fraction of its overall classifications—both for + and − class in our case—that are correct (Eq. 11.4). However, for datasets with a high class imbalance ratio, this measure is not appropriate because it does not reflect misclassification costs and has a strong bias to favour the majority class [31]. Nevertheless, we have reported classification accuracy since it has been reported by some contemporary studies [9].

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}} \tag{11.4}$$

The *false positive rate (FPR)* is the fraction of negative instances that are misclassified. In our case, it is the number of sentences misclassified as a summary sentence divided by the total number of sentences that are not summary sentences (Eq. 11.5). Similarly, the *false negative rate (FNR)* is the fraction of positive instances that are misclassified. Interpreting Eq. 11.6 for our task, it is the number of misclassified summary sentences divided by the total number of summary sentences.

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{11.5}$$

$$\text{FNR} = \frac{\text{FN}}{\text{FN} + \text{TP}} \tag{11.6}$$

In addition, we report the area under curve (`auc`) which is a single scalar value representation of a method's receiver operating characteristic (`ROC`) performance. A `ROC` curve plots true positive rates (Eq. 11.2) and false positive rates (Eq. 11.5) for a binary classification method. The value of `AUC` will always be between 0 and 1.0. A random method will have an `AUC` of 0.50 and no realistic method should have an `AUC` less than this. In practice, this measure discriminates well and is often a good choice when a general measure of predictiveness is desired for data with a high imbalance ratio [32].

Our last measure to gauge the classification performance is the *Matthews correlation coefficient (MCC)* [33]. This correlation measure takes both *positives* and *negatives* into account (Eq. 11.7) and is a highly regarded measure when the class sizes vary significantly. In essence, MCC reports the correlation between the *actual* classes and the *predicted* classifications made by a method. Its value is always between -1 and $+1$, where $+1$ represents a perfect method, 0 represents a random method and -1 refers to a method that totally disagrees with the actual class membership.

$$\text{MCC} = \frac{(\text{TP} \times \text{TN}) - (\text{FP} \times \text{FN})}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \tag{11.7}$$

11.4 Results and Discussions

11.4.1 Performance of the Proposed Method

Table 11.4 summarizes the performance of our proposed method and shows comparable results from previous studies. The comparisons discussed below are made with a paired *t-test* by setting the significance level, α, to 0.05 (i.e., with 95% confidence interval).

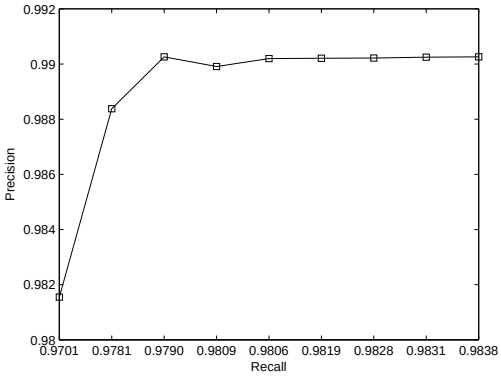
The precision, recall, and F-scores for our method for the two `CAST` datasets are really impressive (Rows 1 and 4). In addition, the high `AUC` values show that our method does not suffer from any serious *recall-FPR* tradeoff and the *near-perfect* MCC scores indicate that the class imbalance ratio present in these datasets has almost no effect on the learned model. The impressive *FNR* values for the datasets denote that our method misclassifies only 1% and 0.6% of the summary sentences for the `CAST-15` and `CAST-30` datasets, respectively. As expected, the method performs slightly better on the `CAST-30` dataset. For the `CAST-15` dataset, we have compared our results with that reported by Leskovec *et al.* [10] [11] (Rows 2 and 3). Our method performs significantly better than them in terms of precision, recall, and F-score. On the other hand for the `CAST-30` dataset, we have compared our work with Leskovec *et al.* [10] [11] (Rows 5 and 6), Berker and Güngör [8] (Row 7), and Villatoro-Tello *et al.* [9] (Rows 8 and 9). Again, the comparisons show that our method outperforms these state-of-the-art methods by a wide margin. The F-score that is the closest to ours is reported by Villatoro-Tello *et al.* [9] which is 90.1% with their *word sequence* model (Row 9). Note that in their study, Berker and

Dataset	Row	Method	Precision %	Recall %	F-score %	Accuracy %	FPR	FNR	AUC	MCC
CAST-15	1	**Stylometry Model**	**99.0**	**98.7**	**98.9**	**99.7**	**0.001**	**0.010**	**0.999**	**0.989**
	2	Position and Graph Model [10]	32.5	70.9	44.5					
	3	Head Noun and Verb Model [11]	48.0	47.0	48.0					
CAST-30	4	**Stylometry Model**	**99.1**	**99.4**	**99.3**	**99.6**	**0.003**	**0.006**	**0.997**	**0.989**
	5	Position, Graph and Linguistic Model [10]	44.5	65.6	53.0					
	6	Heads of Logical Form Triplets Model [11]	42.0	67.0	52.0					
	7	Lexical Chain Model [8]	46.0							
	8	Single Word Model [9]	88.7	84.4	86.5	79.8				
	9	Word Sequence Model [9]	96.5	84.5	90.1	84.5				
DUC-200	10	**Stylometry Model**	**18.0**	**98.4**	**30.4**	**89.4**	**0.108**	**0.017**	**0.983**	**0.397**
	11	Position and Graph Model [10]	33.7	64.4	44.2					
	12	Heads of Logical Form Triplets Model [11]	40.0	40.0	40.0					
	13	Columbia Summarizer [17]	19.0	14.7	16.6					
DUC-400	14	**Stylometry Model**	**42.1**	**98.6**	**59.0**	**93.7**	**0.065**	**0.014**	**0.987**	**0.623**
	15	Columbia Summarizer [17]	23.8	19.6	21.5					

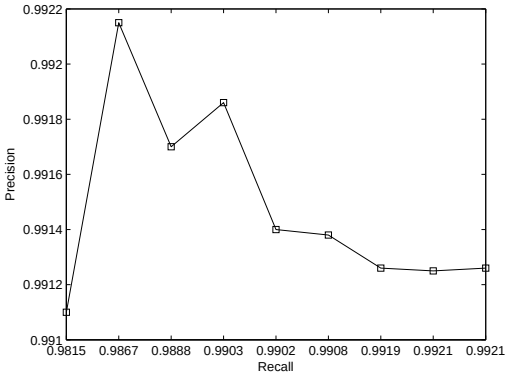
Table 11.4: Summary of the performance of the proposed two-stage method using stylometric attributes, as well as results from previous studies.

Güngör [8] only have reported the precision of their method (Row 7). Overall, the results on the CAST datasets suggest that our method is highly effective classifying summary sentences from single documents in the newswire article genre.

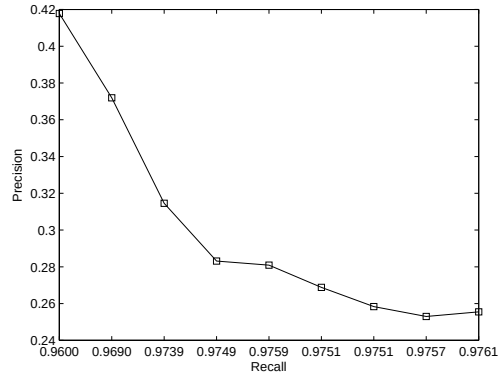
Interestingly, the results of our method on the DUC datasets are not as good as we had expected. Table 11.4 shows that for these datasets, our method has maintained outstanding recall but has lost ground on precision. As a result, we report comparatively low F-scores on these datasets (Rows 10 and 14). Also, contrary to the negligible difference in the precision, recall, and F-score between the two CAST datasets, the difference in these scores between the two DUC datasets is more substantial. Although the AUC scores are still high, low MCC scores suggest that our method has suffered from the high class imbalance ratio of the DUC datasets. In addition, the *FNR* values indicate that it misses about 2% and 1% of the summary sentence data from the two datasets. For the DUC-200 dataset, we have compared our results with Leskovec *et al.* [10] [11]. With their *position and graph* model, they reported an F-score of 44.2% [10] (Row 11) while the F-score with their *heads of logical form triplets* model is 40.0% [11] (Row 12). These results are much better than what we have obtained. However, our method outperforms the Columbia multi-document summarizer [17] for both the DUC-200 and DUC-400 datasets (Rows 13 and 15); the differences are statistically significant. Readers are encouraged to see paper [17] to find the results of the top five summarizer systems submitted in the DUC 2002 competition. The best of the summarizers (system code 19) on the DUC-200 dataset had an F-score of 18.4%. On the other hand, the best performance on the DUC-400 dataset was from a system (code 21) that had an F-score of 25.2%. Therefore, our results are much better than these submitted systems. To sum up, the results on the DUC datasets are mixed—the method still lacks the desired precision for multi-document summarization. One possible explanation for these low precision values is the following. When the human summaries are produced from multiple documents, from a set of similar sentences only one (or a few) is usually chosen. Since each cluster in the DUC2002 collection is composed of approximately 10 newswire articles on the same topic, the likelihood of choosing the wrong sentence(s) from a set of similar sentences increases when choosing fewer sentences (DUC-200 vs. DUC-400), thus increasing the number of *false positives*. This hypothesis is supported by



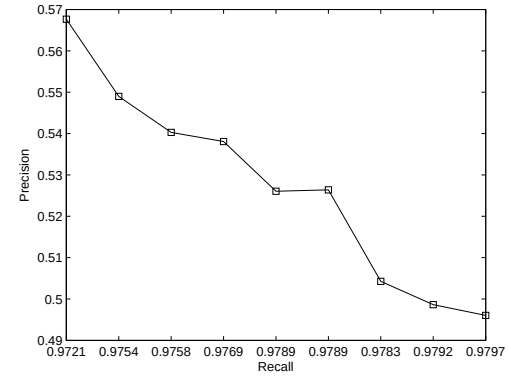
(a) Precision-Recall Curve for the CAST-15 dataset.



(b) Precision-Recall Curve for the CAST-30 dataset.



(c) Precision-Recall Curve for the DUC-200 dataset.



(d) Precision-Recall Curve for the DUC-400 dataset.

Figure 11.3: Precision-Recall Curves for the four datasets.

the *FPR* values in Table 11.4: the *FPR*s of the method on the DUC datasets (Rows 10 and 14) are significantly higher than those on the CAST datasets (Rows 1 and 4), and the *FPR* is higher for the DUC-200 dataset compared to the DUC-400 dataset.

The results we have found raises an important question—*why did the two-stage classification perform as well as this?* To answer this question, we need to examine the individual performance of the two learners—κ-Nearest Neighbour and Naïve Bayes—on the four datasets. Table 11.5 summarizes the *FPR* and *FNR* of the κ-Nearest Neighbour and Naïve Bayes learners. These measures have been obtained using a 10-fold cross validation. For the κ-Nearest Neighbour learners, the *low FPR-high FNR* in the data refers to their good ability to classify positive data points (summary sentences) and their poor ability to classify negative data points. This is completely opposite for the Naïve Bayes learners—they have *low FNR-high FPR*. In essence, the two learners used in our method are complementing each other thereby resulting in the reported classification results.

Dataset	Learner	FPR	FNR
CAST-15	κ-Nearest Neighbour	0.117	0.787
	Naïve Bayes	0.489	0.155
CAST-30	κ-Nearest Neighbour	0.239	0.606
	Naïve Bayes	0.503	0.147
DUC-200	κ-Nearest Neighbour	0.023	0.969
	Naïve Bayes	0.517	0.161
DUC-400	κ-Nearest Neighbour	0.044	0.930
	Naïve Bayes	0.530	0.143

Table 11.5: *FPR* and *FNR* of the κ-Nearest Neighbour and Naïve Bayes learners.

11.4.2 Effect of Data Size

In this section, we report the effect of data size on classification performance. To observe this effect, treating each dataset independently, we have generated nine equally sized sets. Each set contains $x\%$ of the original data with repetition and x has been varied from 10% to 90% with an increment of 10% per set. Therefore, our first set contains 10% and the ninth set contains 90% of the data. Then, on each original set we have run a 10 times 10-fold cross validation and recorded our method's precision and recall. Empirically, this should reduce the *variance* of our method across datasets with different size. With the data found from this experiment, we have plotted the *precision-recall curve* for each dataset.

The precision-recall curves in Figure 11.3 show that the data size has some effect on our method's performance. For the CAST-15 dataset, the curve is *ideal* and outlines the increase of both the precision and recall with the growth of the data size (Figure 11.3a). However, for the CAST-30 dataset, we get better recall with more data by compromising the precision ever so slightly (Figure 11.3b). On the other hand, the tradeoff is relatively high for the DUC datasets. Increasing data increases the recall slightly with a steep decrease in precision. Most importantly, however, the plots suggest that the precision-recall tradeoff stabilizes as we add more data.

11.5 Conclusions and Future Work

Summary sentence classification is regarded as a pivotal and reasonably complex step to generate summaries for text documents. Importantly, the quality of a summary depends much on the correctness of this step. Our aim is to provide a simple solution to this reasonably complex problem. We propose a statistical learning method to model sentences using a novel set of attributes related to stylometry—a popular linguistic study of writing styles. To learn the models, a discriminative and a generative learner are used in an interesting two-stage classification setup. The learners used in this experiment are the κ-Nearest Neighbour and the Naïve Bayes learners. Our extensive experiments with the summarization data collected from two benchmark datasets named the CAST dataset and the DUC2002 collection strongly suggest that the proposed method performs very well for the single document summarization task. On the other hand, its performance is mixed for classifying summary sentences from multiple documents. Finally, comparisons show that our method performs much better than most other

state-of-the-art summarization methods.

There is still room for improvement. First, we use only newswire articles in our experiment; results may vary for texts from domains like science and technology. Second, our performance on multi-document summarization is mixed. Although we have outlined possible reasons for this in Section 11.4.1, a thorough investigation is left as future work. Finally, more analysis should be done on anaphora, at least for the DUC datasets, since *pronouns* are regarded as a salient attribute for these datasets.

Bibliography

[1] M. Eppler and J. Mengis, "The Concept of Information Overload: A Review of Literature from Organization Science, Accounting, Marketing, MIS, and Related Disciplines," *Kommunikationsmanagement im Wandel*, pp. 271–305, 2008.

[2] D. R. Radev, E. Hovy, and K. McKeown, "Introduction to the special issue on summarization," *Computational Linguistics*, vol. 28, pp. 399–408, Dec. 2002.

[3] E. Hovy, "Text summarization," in *The Oxford Handbook of Computational Linguistics* (R. Mitkov, ed.), Oxford Handbooks in Linguistics, ch. 32, pp. 583–598, Oxford: Oxford University Press, 2003.

[4] W. T. Chuang and J. Yang, "Extracting sentence segments for text summarization: A machine learning approach," in *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, (New York, NY, USA), pp. 152–159, ACM, 2000.

[5] U. Hahn and I. Mani, "The challenges of automatic summarization," *Computer*, vol. 33, pp. 29–36, Nov. 2000.

[6] P. B. Baxendale, "Machine-made index for technical literature: An experiment," *IBM Journal of Research and Development*, vol. 2, pp. 354–361, Oct. 1958.

[7] H. P. Edmundson, "New methods in automatic extracting," *Journal of ACM*, vol. 16, pp. 264–285, Apr. 1969.

[8] M. Berker and T. Gngr, "Using genetic algorithms with lexical chains for automatic text summarization.," in *Proceedings of the 4th International Conference on Agents and Artificial Intelligence (ICAART2012)*, pp. 595–600, SciTePress, 2012.

[9] E. Villatoro-Tello, L. V. Pineda, and M. M. y Gomez, "Using word sequences for text summarization.," in *Text, Speech and Dialogue* (P. Sojka, I. Kopecek, and K. Pala, eds.), vol. 4188 of *Lecture Notes in Computer Science*, pp. 293–300, Springer, 2006.

[10] J. Leskovec, N. Milic-Frayling, and M. Grobelnik, "Extracting summary sentences based on the document semantic graph," Tech. Rep. MSR-TR-2005-07, Microsoft Research, January 2005.

[11] J. Leskovec, N. Milic-frayling, and M. Grobelnik, "Impact of linguistic analysis on the semantic graph coverage and learning of document extracts," in *Proceedings of the Twentieth National Conference on Artificial Intelligence(AAAI2005)*, pp. 1069–1074, 2005.

[12] E. Lloret and M. Palomar, "Challenging issues of automatic summarization: Relevance detection and quality-based evaluation.," *Informatica*, vol. 34, no. 1, pp. 29–35, 2010.

[13] T. Givón, *Syntax: A functional typological introduction*. Amsterdam: John Benjamins, 1990.

[14] S. Ji, "A textual perspective on givn's quantity principle," *Journal of Pragmatics*, vol. 39, no. 2, pp. 292 – 304, 2007. Focus-on Issue: Discourse, Information, and Pragmatics.

[15] L. Hasler, C. Orăsan, and R. Mitkov, "Building better corpora for summarisation," in *Proceedings of Corpus Linguistics 2003*, (Lancaster, UK), pp. 309 – 319, March 2003.

[16] P. Over and W. Liggett, "Introduction to DUC: An intrinsic evaluation of generic news text summarization systems," in *Proceedings of the Document Understanding Conference (DUC2002)*, 2002.

[17] K. M. David, D. Evans, A. Nenkova, R. Barzilay, V. Hatzivassiloglou, B. Schiffman, and J. Klavans, "The Columbia multi-document summarizer for DUC 2002," in *Proceedings of the ACL Workshop on Automatic Summarization/Document Understanding Conference (DUC2002)*, pp. 1–8, 2002.

[18] K. R. McKeown, V. Hatzivassiloglou, R. Barzilay, B. Schiffman, D. Evans, and S. Teufel, "Columbia multi-document summarization: Approach and evaluation," in *Proceedings of the Document Understanding Conference (DUC2001)*, 2001.

[19] B. Schiffman, A. Nenkova, and K. McKeown, "Experiments in multidocument summarization," in *Proceedings of the Second International Conference on Human Language Technology Research*, HLT '02, (San Francisco, CA, USA), pp. 52–58, Morgan Kaufmann Publishers Inc., 2002.

[20] T. Rose, M. Stevenson, and M. Whitehead, "The reuters corpus volume 1 - from yesterdays news to tomorrows language resources," in *Proceedings of the Third International Conference on Language Resources and Evaluation*, pp. 29–31, 2002.

[21] R. Shams and R. E. Mercer, "Classifying spam emails using text and readability features," in *Proceedings of the 2013 IEEE 13th International Conference on Data Mining (ICDM2013)*, (Texas, USA), pp. 657–666, IEEE, 2013.

[22] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, (Stroudsburg, PA, USA), pp. 173–180, Association for Computational Linguistics, 2003.

[23] B. Santorini, "Part-Of-Speech tagging guidelines for the Penn Treebank project (3rd revision, 2nd printing)," tech. rep., Department of Linguistics, University of Pennsylvania, Philadelphia, PA, USA, 1990.

[24] M. B. Kursa and W. R. Rudnicki, "Feature selection with the Boruta package," *Journal of Statistical Software*, vol. 36, no. 11, pp. 1–13, 2010.

[25] R. Nilsson, J. M. Peña, J. Björkegren, and J. Tegnér, "Consistent feature selection for pattern recognition in polynomial time," *Journal of Machine Learning Research*, vol. 8, pp. 589–612, May 2007.

[26] P. A. Viola and M. J. Jones, "Rapid object detection using a boosted cascade of simple features.," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, vol. 1, pp. 511–518, 2001.

[27] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal on Computer Vision*, vol. 57, pp. 137–154, May 2004.

[28] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Machine Learning*, vol. 6, pp. 37–66, Jan. 1991.

[29] G. H. John and P. Langley, "Estimating continuous distributions in Bayesian classifiers," in *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, (San Francisco, CA, USA), pp. 338–345, Morgan Kaufmann Publishers Inc., 1995.

[30] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *SIGKDD Explorations Newsletter*, vol. 11, pp. 10–18, Nov. 2009.

[31] S. Daskalaki, I. Kopanas, and N. Avouris, "Evaluation of classifiers for an uneven class distribution problem," *Applied Artificial Intelligence*, vol. 20, pp. 1–37, 2006.

[32] T. Fawcett, "An introduction to roc analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.

[33] B. W. Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme," *Biochimica Biophysica Acta (BBA)*, vol. 405, pp. 442–451, 1975.

.

Chapter 12

Conclusions

This study explores various possibilities to identify informativeness in natural language text utilizing a more general as well as a domain- and genre-independent set of attributes. We work with a set of attributes novel to this task that are related to a linguistic study called Stylometry which can be used to interpret writing styles in texts. Interestingly, we find that these stylometric attributes are somewhat closely related to another linguistic theory called the Code Quantity Principle. According to this principle, humans codify important information in their writing by changing certain linguistic elements that are also important for Stylometry. We combine the ideas from these two linguistic studies: that informativeness in natural language text can be found by analyzing the stylometric elements present in the writing styles of its authors.

There are many natural language attributes that attempt to find informativeness in text. However, the general theoretical and empirical literature on this subject discuss much about these attributes' weaknesses by being genre and domain dependent. Therefore, the biggest challenge of this study is to explore the possibilities of the stylometric attributes to find informativeness in texts by answering the following research questions:

1. Are the stylometric attributes an effective means to find informativeness in natural language text?

2. How do the stylometric attributes perform with different genres of texts?

3. How do the stylometric attributes perform for different natural language processing tasks?

12.1 Empirical Findings

This thesis is an example of how ideas, concepts, methodologies, and evaluations evolve over the research lifetime of a thesis. We have studied different methods and techniques to explore the possibilities of stylometric attributes to determine informativeness in texts and have provided substantial evidence to answer the aforementioned questions positively. The evidence is based mostly on empirical studies conducted on different genres of texts like biomedical, food and agriculture, physics, computer science, economics, conversational texts, and news for

	Biomedical	Food and Agriculture	Physics	Computer Science	Economics	Conversational Texts	News
Relation Mining	✓	×	×	×	×	×	×
Keyphrase Indexing	✓	✓	—	✓	—	×	×
Spam Classification	×	×	×	×	×	✓	×
Summarization	×	×	×	×	✓	×	✓

Table 12.1: The performance summary of the stylometric attributes for different genres of text (the columns) for different NLP areas (the rows). Items show where stylometric attributes were successful (cells with a '✓'), where there is room for improvement (cells with a '—'), and areas and genres that were not explored (cells with a '×').

different currently popular tasks such as biomedical relation mining, keyphrase indexing, spam classification, and text summarization. A brief summary of the performance of stylometry in determining informativeness in text is laid out in Table 12.1. The items show where stylometric attributes were successful (cells with a '✓'), where there is room for improvement (cells with a '—'), and areas and genres that were not explored (cells with a '×').

We find substantial evidence to support the the use of stylometric attributes for the biomedical relation mining task. To be precise, we are not interested in actually mining biomedical relations but rather to classify sentences in biomedical articles that contain them. Using only one of the attributes and a few simple rules, we attempt to find sentences describing the relations between human diseases and chemicals responsible for them (Chapter 2). The results are impressive as they show that this method correctly classifies approximately 75% of the sentences containing these relations in a dataset. In Chapter 3, we conduct a similar experiment but with a few more stylometric attributes, but the results are comparatively no better. Next, we outline a machine-learning apprach by modelling sentences of biomedical articles using 103 stylometric attributes and a two-stage classification. The results are better this time. The evidence from this last study strongly suggests that stylometric attributes are strong contenders for classifying sentences that contain protein-protein interactions.

As well, we explore the strengths of the attributes for automatic keyphrase indexing. However, this is one of our earlier experiments and therefore, the method we use for this task as described in Chapters 4 and 5 is similar to that reported in Chapters 2 and 3. This time, we use the attribute to generate subsets of informative text from full-length articles and train three keyphrase indexers. Results suggest that for all the indexers, the subsets treat the informative text as better training data. This finding indicates the strength of the attribute in classifying sentences that contain keyphrases in research articles. To test the attribute, we use texts from three different genres: agriculture, physics, and biomedical science. After further development of our ideas using machine learning, we return to this problem, replacing the rules with machine-learning models and increasing the number of attributes. The results with machine-learning techniques, again, are better than those with rules (Appendix A).

Spam detection is a classic natural language processing task. Our approach to this problem is somewhat different than previous approaches. And additionally, instead of our sentence-by-sentence approach, we model the full textual content of an email with stylometric attributes. Our experimentation is rigorous: we use nine email datasets, 37–40 stylometric attributes, and supervised learning with five different learning algorithms. The findings of the experiments are interesting. The attributes are not as strong as they seem for the other tasks but when used

together with certain other attributes particularly suited to this domain, the results are better than most of the benchmarks. Another interesting observation is that the stylometric attributes perform better on emails collected from a single source (or person) than on emails collected from multiple sources. Details can be found in Chapters 7 to 9.

We finally select another classic problem of single and multiple document summarization. However, we are interested in extracting summary sentences with our method rather than developing a full-functional text summarizer. To do so, we model sentences from four benchmark datasets using 87 stylometric attributes and two-stage classification. The outcomes of the experiment suggest that the method outperforms most of the latest summarizers. Interestingly, the method seems to work better for single document summarization. However, its performance on multi-document summarization is also competitive. The overall description of this method is outlined in Chapter 11.

12.2 Theoretical Implications

Identification of informativeness in text data is a challenge in natural language processing. This is mainly due to the presence of a large variety of texts and a wide array of problem areas in this domain. We propose a very simple solution for a fairly complex problem and explore its possibilities for the various types of texts and problem areas. Our work attempts to relate the elements of human codification of informativeness in text to the elements of human writing style. Interestingly, the linguistic theory to observe the codification process called the *Code Quantity Principle* and the study to observe writing styles called *Stylometry* have co-existed for some time. Surprisingly, even though the two are logically connected, this thesis is the first attempt to relate them and use writing styles to identify the codifications. We do discover that one of the code quantity principles that relate the degree of informativeness *linearly* with the number of lexical elements does not necessarily hold. Certainly, there is a relationship between these two, but our study finds that the relationship is complex and this complex relationship can be found by machine-learning models. Therefore, certain aspects of the principle should be revisited according to our findings.

12.3 Recommendations for Future Research

In some of the final sections of the Chapters 2 to 11, we suggested some work for future research. Much of the work mentioned in the early chapters has been accomplished and reported in the later chapters. In addition, due to time constraints, some of the suggestions could not be accomplished—they can be seen as possibilities to be explored by other researchers.

In Chapter 3, we left the work of pairing various readability measures with text denoising as future work. However, we later abandoned this idea completely since they do not produce competitive results compared to what is reported in Chapter 2. In Chapter 4, we left the possibilities of pairing text denoising with other keyphrase indexers as future work. We successfully completed the task with two indexers (described in Chapter 5). Unfortunately, the Medical Text Indexer (MTI) [1] of the National Library of Medicine (NLM) was too slow to do any serious ex-

[1] `http://ii.nlm.nih.gov/MTI/`

perimental work, since we were using its web interface. As a result, we did not further pursue working with this indexer. We believe that further experimentation could have been insightful since MTI is a specialized keyphrase indexer for biomedical documents. In Chapter 5, we left the testing of keyphrase indexing with a dataset for which performance gain is much more competitive. We completed this task and reported the outcome in Appendix A. The future work on spam classification indicated in Chapter 6 is accomplished in Chapters 7 to 9. However, as suggested in Chapter 7 and 9, we did not proceed with the *stacking* of machine-learning algorithms or semi-supervised algorithms for the anti-spam filters, nor tested the filters for *concept drifts* or real-time data as suggested in Chapter 8.

Besides what was left as future work and could not be accomplished in course of time, we have some general recommendations to extend the research. First and foremost, while doing the research we always kept in mind this question—*how do writing styles represent codification of informativeness?*. However, we have limited our goal to exploring the applicability of studying writing styles. Answering this question requires significant linguistic analysis and a comprehensive study to make the link among human psychology, human writing, and human codification of informativeness in text. Therefore, this avenue is still wide open for future research.

Except for the spam classification task, for all other tasks, our method is used as a preprocessing step for the original task, since this is what we set as our goal. Our method is not fully integrated in any of the existing natural language processing tools except the anti-spam filter that we developed. So, there is room to tightly couple our method of determining informative sentences into benchmark biomedical relation miners, keyphrase indexers, or text summarizers where the results could be stronger.

Keeping the future trends of biomedical science in mind, we suggest exploring the biomedical Named Entity Recognition (NER) area with the stylometric attributes. Some of our experiments show that the attributes are a good means to find human diseases and related chemical components from biomedical research texts. But the attributes are never tested for biomedical named entities in general. A few additional text components can make our method a competitive tool for NER.

We have some recommendations for future work from the machine-learning point of view. In most cases to report performance of our method on unbalanced datasets, we use cost-sensitive evaluations; interested researchers may want to use cost-sensitive learning instead.

12.4 Limitations of the Study

This study explores the stylometric attributes for at least four major natural language processing tasks, but has been constrained by time. As a consequence, the study encounters a number of limitations, which need to be considered.

First, we only consider stylometric attributes that work at the character level, word level, sentence level, and document level. There are some other attributes that work at the phrase level, the clause level, and the paragraph level. These attributes, however, are not considered in this study. Given their properties, we can assume that these attributes should provide good results at least for the text summarization task.

Second, much of our data is collected from research articles. In some cases, during the

data curation process, only the body text is retained. The rest, which includes figure and table captions, is cleaned. Had this other text also been retained during the curation process, it could have proved informative, too, at least for the protein interaction sentence classification task.

Third, we have annotated a dataset for keyphrase indexing using semi-supervised learning (see Appendix A). The initial labels for the data points are provided by the author of this thesis. However, the proper way to have the initial labels should be much more methodical. For instance, two human annotators can be applied to annotate the data points. Then if the agreements between the annotators are high, then the data points for which they disagree can be given to a third annotator. Then on the basis of a majority voting the final set of initial labels can be found.

12.5 Conclusions

A number of natural language attributes have been suggested as identifying textual informativeness or textual importance. In spite of what is often reported about their usefulness, in practice, most of these attributes depend on the genre and the topic domain of the text. Also, these attributes are rarely applicable across multiple standard and classic natural language processing tasks. Positional attributes, for instance, are good for generating summaries but are unsuited for classifying spam emails. Moreover, the importance of positional attributes is not the same to generate newspaper summaries and medical form summaries.

We, therefore, need natural language attributes that are capable of identifying informativeness for many texts of different kinds and domains. With extensive experimentation, this thesis shows that stylometric attributes are a competitive source of such attributes. These attributes attempt to represent certain aspects of human writing style. To the best of our knowledge, ours is the first attempt to relate human codification of information in texts with this type of representation of writing style.

Appendix A

Extended Work on Keyphrase Indexing

Our investigation on the *denoising threshold* reported in Chapters 4 and 5 points out that partly because the notion of what is informative is given by a threshold on a measure, the amount of text to include in the set of informative sentences changes for different genres of texts. The heuristics to set the threshold are computed manually. This is problematic if this method of text reduction is applied on an array of documents collected from different domains. To overcome this problem, we introduce a machine-learning approach where the sentences in a document are modeled with stylometric attributes and one or more statistical learners. Note, however, that the machine-learning approach for keyphrase indexing is developed much later, in the last two years of this thesis. Although we give a brief description of the stylometric attributes used for keyphrase indexing in this section, more details can be found in the chapters following Chapter 5. Also, the work described in this section can be a possible extension of the papers comprising Chapters 4 and 5. Therefore, our plan is to combine this work with the publications on keyphrase indexing for a possible journal submission.

The machine-learning approach to text donoising can be seen as a binary classification problem. The two classes, in this case, are sentences that may contain keyphrases and others that don't. We represent each sentence in a documents as $(\vec{x}, y)$, where $\vec{x} \in \mathbb{R}^{30}$ is a vector of the 30 stylometric attributes (see Table A.1 for a brief summary, details can be found in the chapters following Chapter 5) and $y \in \{+, -\}$ is the label of the sentence. A sentence has label +, if the sentence contains a keyphrase, −, otherwise. Our aim is to train a classifier on the stylometric attributes to label sentences from documents and then to provide only the sentences labelled as + to the indexer Maui [1]. Maui then treats these sentences as its training data and builds a model. Finally, the model can be evaluated on test documents. In other words, the machine-learning method we discuss here can be seen as an preprocessing step for Maui to build better models with reduced data, perhaps with more informative data.

To train a binary classifier on data, the first and foremost condition is to have them labeled as + and −. Unfortunately to the best of our knowledge, no dataset used to evaluate keyphrase indexing has this annotation. Therefore, annotating a dataset is the biggest challenge for this work. This is where semi-supervised learning becomes helpful. This type of statistical learning is very useful if the amount of labelled data is very small and often produces better results than supervised learning. The details of semi-supervised learning are beyond the scope of the thesis. However, interested readers can find the details about this novel statistical learning in [2].

As the dataset for this experiment, we have chosen the SemEval-2010 task 5 dataset [3].

Category	Quantity	Description
Word-level Attributes	5	Alpha-numeric words, Function words, TF·ISF.
Sentence Position Attributes	3	Position of a sentence in a document.
Readability Attributes	22	Simple and complex words, Fog index, Simple and Inverse Fog index, Sentence Length, Word length.

Table A.1: Summary of the attributes: category, quantity, and description.

Task 5 of the conference was to automatically assign keyphrases to scientific articles. The training data is composed of 144 scientific articles and the test data is compsed of 100 articles collected from the ACM Digital Library (conference and workshop papers). These articles are grouped into four sets according to the 1998 ACM classifications: C2.4 (Distributed Systems), H3.3 (Information Search and Retrieval), I2.11 (Distributed Artificial Intelligence—Multiagent Systems) and J4 (Social and Behavioral Sciences—Economics).

To provide the initial set of training data for our semi-supervised classification problem, one training article from each of the four ACM classification categories is randomly selected. Then, the sentences of these articles are annotated *by the author of this thesis* with a + label if the sentence seems to contain a keyphrase, with a − label, otherwise. These few labelled data are then used to annotate all of the unlabelled sentences in the remaining training articles using *self-training*. In self-training, using a set of initially labelled data L, a classifier, C_1 is generated. This classifier is then applied on a set of initially unlabelled data U. According to a pre-set *confidence threshold*, the classification of unlabelled data is observed. If the classifier's confidence reaches the threshold, the newly classified instances are concatenated with L to produce a set L_{new} and removed from U to produce U_{new}. A second classifier, C_2 is generated from L_{new} and then is applied on U_{new}. This cycle continues until the classifier converges—which means that either (a) all of the unlabelled data are confidently labelled by the classifier or (b) the classifier's confidence fails to reach the threshold for at least one unlabelled data item for several cycles, in which case, the unlabelled data is discarded. In our study, we use Naïve Bayes as the learning algorithm for the classifier and heuristically set its confidence threshold to 98%. If we compare these annotated data to what we have reported in Chapters 4 and 5, then the sentences labelled as + can be seen as the *denoised text* and those labelled as − as the *noise text*.

Once all the training data used in the SemEval-2010 are annotated, only the denoised text is used to train Maui. The default parameter values for Maui that it used in the conference are retained. Maui then generates a model and the model is applied on the test articles to generate keyphrases. Just like the conference guidelines, we configure Maui to provide the top 5, 10, and 15 keyphrases for each test article. The keyphrases produced by Maui are then compared with the gold standard keyphrases. Note that the gold standard is a combination of the keypharases indexed by the authors of the documents and a few human assessors. As evaluation measures, we select precision, recall, and F-score.

The performance of Maui trained with the denoised and the full text is summarized in Table A.2. For all of the test documents combined, Maui's precision, recall, and F-score when trained

Keyphrase Indexer	Top 5 Candidates		
	Precision%	**Recall%**	**F-score%**
Maui with Text Denoising	39.0	16.5	23.2
Maui in SemEval-2010 [3]	35.0	11.9	17.8
	Top 10 Candidates		
	Precision%	**Recall%**	**F-score%**
Maui with Text Denoising	28.6	21.4	24.5
Maui in SemEval-2010 [3]	25.2	17.2	20.4
	Top 15 Candidates		
	Precision%	**Recall%**	**F-score%**
Maui with Text Denoising	23.9	24.9	24.4
Maui in SemEval-2010 [3]	20.3	20.8	20.5

Table A.2: The performance of Maui with denoised and full text as its training data

with the denoised text are significantly better than that reported in [3], i.e., when trained on the full text. Convincingly, for each number of keyphrase candidates—top 5, 10, and 15—the denoised text training data produces a better performing Maui than the full-text data which was used to train the indexer in the conference. To test the difference in the scores, we use a paired *t-test* by setting the significance level $\alpha = 0.5$. We also compare the F-score of Maui using the denoised and the full text training data for each of the ACM categories (Table A.3). Interestingly, Maui performs much better with the denoised training data for the C, H, and I categories. However, the indexer performs more poorly for the J category. Our investigation shows that the following reasons might have affected the result for this category.

1. The articles from the J category are taken from economics domain. However, the annotator who put the initial labels for the sentences (i.e., the author of this thesis) is from the computer science domain. Therefore, the initial labels may not be as good as other categories. Note that the articles from other categories are from the computer science domain.

2. The human assessors who provided the gold standard keyphrases for the SemEval-2010 dataset were also computer science graduate students [3]. So, this might have affected the quality of the gold standard keyphrases for the economics articles.

3. We have also reported the amount of text reduced during the self-training in Table A.4. From the combined data, we see that our method overall reduces the original data to a

Keyphrase Indexer	ACM Categories			
	C	H	I	J
Maui with Text Denoising	23.0	24.4	20.1	18.0
Maui in SemEval-2010	19.3	23.9	17.6	21.3

Table A.3: The F-score (%) of Maui with and without denoised text as its training data on the individual ACM categories

Category	Total Sentence	Sentence Selected	Text Reduction (%)
C	11,464	5,256	54.2
H	11,466	1,930	83.2
I	10,494	3,291	68.6
J	14,903	197	98.7
Total	48,327	10,674	77.9

Table A.4: Total sentences in the training articles of the SemEval-2010 training data [3], number of sentences selected as training data for Maui and the rate of text reduction by the method.

remarkable 22% of its original size, but Maui, when trained with the reduced data, still performs much better than its original benchmark. Hiwever, among the four categories, only 3% (197 sentences out of 14,903) of the original sentences from the J category are selected by our method to train Maui. Such a small quantity of training data might be another reason for the comparatively low F-score of a denoised text-trained Maui on this genre of articles.

The data presented in this section confirms that the modelling of the sentences using stylometric attributes and a threshold determined by the semi-supervised learning produces a rich and reduced set of informative sentences. The burden of manually finding the *denoising threshold* is overcome as now the classification models take care of this automatically. The performance of Maui trained with the denoised text is much improved. In the SemEval-2010, the rank of the indexer was 9th (the ranking was done on the basis of the performances on the top 15 keyphrase extraction). The indexer's F-score when trained with the denoised text is 24.4% which is the fifth best score in the conference.

Bibliography

[1] O. Medelyan and I. Witten, "Domain-independent automatic keyphrase indexing with small training sets," *Journal of the American Society for Information Science and Technology (JASIST)*, vol. 59, no. 7, pp. 1026–1040, 2008.

[2] X. Zhu, A. B. Goldberg, R. Brachman, and T. Dietterich, *Introduction to Semi-Supervised Learning*. Morgan and Claypool Publishers, 2009.

[3] S. N. Kim, O. Medelyan, M.-Y. Kan, and T. Baldwin, "Semeval-2010 Task 5: Automatic keyphrase extraction from scientific articles," in *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval 2010)*, (Uppsala, Sweden), 2010.

Appendix B

Parameter Settings

B.1 Parameter Settings for the Algorithms in Chapter 8

The experiments illustrated in Chapter 8 are carried out using the Weka data mining tool[1]. The parameters set for the algorithms can be found in Table B.1.

Learning Algorithms	**Parameters**	
Random Forest (RF)	Maximum Depth: Unlimited	Number of Trees to be Generated: 10
	Random Seed: 1	
ADABOOSTM1	Number of Iterations: 10	Random Seed: 1
	Resampling: False	Weight Threshold: 100
BAGGING	Size of Bag (%): 100	Out of Bag Error: False
	Number of Iterations: 10	Random Seed: 1
Support Vector Machine (SVM)	SVM Type: C-SVC	Cost: 1.0
	Degree of Kernel: 3	EPS: 0.0010
	Gamma: 0.0	Kernel Type: Radial Basis
	Epsilon: 0.1	Probability Estimates: False
	Shrinking Heuristics: True	
Naïve Bayes (NB)	Use of Kernel Estimator: False	

Table B.1: Parameters of the learning algorithms used in the experiments of Chapter 8.

[1]Download at: `http://www.cs.waikato.ac.nz/ml/weka/`

B.2 Parameter Settings for the Algorithms in Chapter 11

The experiments illustrated in Chapter 11 are carried out using the Weka data mining tool[2]. The two algorithms used in the two-stage classification method described in that chapter are κ-Nearest Neighbour and Naïve Bayes. Except for the κ value for the former, other parameters set for this experiment are the default values in the Weka tool. The overall parameter settings can be found in Table B.2.

Classifier	Parameters
κ-Nearest Neighbour	No. of neighbours: κ, the choice of κ is different for different datasets. Justifications of this choice can be found in Chapter 11.
	Best κ value selection: Using Training Data
	Distance weighting: None
	Nearest neighbour search algorithm: Linear NN Search (Distance function: Euclidean, Skip identical instances: False)
	Window size: 0
Naïve Bayes	Kernel estimator: None
	Supervised discritization: No

Table B.2: Parameters of the learning algorithms used in the experiments of Chapter 11.

[2]Download at: `http://www.cs.waikato.ac.nz/ml/weka/`

Appendix C

Copyright Forms of the Papers

There are seven published papers included in this thesis. Two of the papers described in Chapter 2 and 3 are from Open Access sources. Therefore, the copyright releases for those to papers are not included herewith. For the papers published that are described in Chapter 4 to 8, the copyright release forms are included in this appendix.

ASSOCIATION FOR COMPUTING MACHINERY, INC. LICENSE TERMS AND CONDITIONS

Jul 15, 2014

This is a License Agreement between Rushdi Shams ("You") and Association for Computing Machinery, Inc. ("Association for Computing Machinery, Inc.") provided by Copyright Clearance Center ("CCC"). The license consists of your order details, the terms and conditions provided by Association for Computing Machinery, Inc., and the payment terms and conditions.

License Number	3430301499043
License date	Jul 15, 2014
Licensed content publisher	Association for Computing Machinery, Inc.
Licensed content publication	Proceedings of the 12th ACM/IEEE-CS joint conference on Digital Libraries
Licensed content title	Investigating keyphrase indexing with text denoising
Licensed content author	Rushdi Shams, et al
Licensed content date	Jun 10, 2012
Type of Use	Thesis/Dissertation
Requestor type	Author of this ACM article
Is reuse in the author's own new work?	Yes
Format	Print and electronic
Portion	Full article
Will you be translating?	No
Order reference number	None
Title of your thesis/dissertation	Identification of Information-richness in Text using Natural Language Stylometry
Expected completion date	Aug 2014
Estimated size (pages)	80
Billing Type	Credit Card
Credit card info	Master Card ending in 5415
Credit card expiration	11/2014
Total	8.00 USD
Terms and Conditions	

Rightslink Terms and Conditions for ACM Material

1. The publisher of this copyrighted material is Association for Computing Machinery, Inc. (ACM). By clicking "accept" in connection with completing this licensing transaction, you agree that the following terms and conditions apply to this transaction (along with the Billing and Payment terms

(either by CCC or ACM) as provided in CCC's Billing and Payment terms and conditions.

9. If full payment is not received within 90 days from the grant of license transaction, then any license preliminarily granted shall be deemed automatically revoked and shall be void as if never granted. Further, in the event that you breach any of these terms and conditions or any of CCC's Billing and Payment terms and conditions, the license is automatically revoked and shall be void as if never granted.

10. Use of materials as described in a revoked license, as well as any use of the materials beyond the scope of an unrevoked license, may constitute copyright infringement and publisher reserves the right to take any and all action to protect its copyright in the materials.

11. ACM makes no representations or warranties with respect to the licensed material and adopts on its own behalf the limitations and disclaimers established by CCC on its behalf in its Billing and Payment terms and conditions for this licensing transaction.

12. You hereby indemnify and agree to hold harmless ACM and CCC, and their respective officers, directors, employees and agents, from and against any and all claims arising out of your use of the licensed material other than as specifically authorized pursuant to this license.

13. This license is personal to the requestor and may not be sublicensed, assigned, or transferred by you to any other person without publisher's written permission.

14. This license may not be amended except in a writing signed by both parties (or, in the case of ACM, by CCC on its behalf).

15. ACM hereby objects to any terms contained in any purchase order, acknowledgment, check endorsement or other writing prepared by you, which terms are inconsistent with these terms and conditions or CCC's Billing and Payment terms and conditions. These terms and conditions, together with CCC's Billing and Payment terms and conditions (which are incorporated herein), comprise the entire agreement between you and ACM (and CCC) concerning this licensing transaction. In the event of any conflict between your obligations established by these terms and conditions and those established by CCC's Billing and Payment terms and conditions, these terms and conditions shall control.

16. This license transaction shall be governed by and construed in accordance with the laws of New York State. You hereby agree to submit to the jurisdiction of the federal and state courts located in New York for purposes of resolving any disputes that may arise in connection with this licensing transaction.

17. There are additional terms and conditions, established by Copyright Clearance Center, Inc. ("CCC") as the administrator of this licensing service that relate to billing and payment for licenses provided through this service. Those terms and conditions apply to each transaction as if they were restated here. As a user of this service, you agreed to those terms and conditions at the time that you established your account, and you may see them again at any time at http://myaccount.copyright.com

18. Thesis/Dissertation: This type of use requires only the minimum administrative fee. It is not a fee for permission. Further reuse of ACM content, by ProQuest/UMI or other document delivery providers, or in republication requires a separate permission license and fee. Commercial resellers of your dissertation containing this article must acquire a separate license.

Special Terms:

If you would like to pay for this license now, please remit this license along with your payment made payable to "COPYRIGHT CLEARANCE CENTER" otherwise you will be invoiced within 48 hours of the license date. Payment should be in the form of a check or money order referencing your account number and this invoice number 501352202. Once you receive your invoice for this order, you may pay your invoice by credit card. Please follow instructions provided at that time.

Make Payment To:
Copyright Clearance Center
Dept 001
P.O. Box 843006
Boston, MA 02284-3006

For suggestions or comments regarding this order, contact RightsLink Customer Support: customercare@copyright.com or +1-877-622-5543 (toll free in the US) or +1-978-646-2777.

Gratis licenses (referencing $0 in the Total field) are free. Please retain this printable license for your reference. No payment is required.

SPRINGER LICENSE
TERMS AND CONDITIONS

Jun 24, 2014

This is a License Agreement between Rushdi Shams ("You") and Springer ("Springer") provided by Copyright Clearance Center ("CCC"). The license consists of your order details, the terms and conditions provided by Springer, and the payment terms and conditions.

All payments must be made in full to CCC. For payment instructions, please see information listed at the bottom of this form.

License Number	3415390159732
License date	Jun 24, 2014
Licensed content publisher	Springer
Licensed content publication	Springer eBook
Licensed content title	Improving Supervised Keyphrase Indexer Classification of Keyphrases with Text Denoising
Licensed content author	Rushdi Shams
Licensed content date	Jan 1, 2012
Type of Use	Thesis/Dissertation
Portion	Full text
Number of copies	1
Author of this Springer article	Yes and you are the sole author of the new work
Order reference number	None
Title of your thesis / dissertation	Identification of Information-richness in Text using Natural Language Stylometry
Expected completion date	Aug 2014
Estimated size(pages)	80
Total	0.00 USD
Terms and Conditions	

Introduction

The publisher for this copyrighted material is Springer Science + Business Media. By clicking "accept" in connection with completing this licensing transaction, you agree that the following terms and conditions apply to this transaction (along with the Billing and Payment terms and conditions established by Copyright Clearance Center, Inc. ("CCC"), at the time that you opened your Rightslink account and that are available at any time at http://myaccount.copyright.com).

Limited License

With reference to your request to reprint in your thesis material on which Springer Science and

respect to the licensed material and adopts on its own behalf the limitations and disclaimers established by CCC on its behalf in its Billing and Payment terms and conditions for this licensing transaction.

Indemnity
You hereby indemnify and agree to hold harmless Springer Science + Business Media and CCC, and their respective officers, directors, employees and agents, from and against any and all claims arising out of your use of the licensed material other than as specifically authorized pursuant to this license.

No Transfer of License
This license is personal to you and may not be sublicensed, assigned, or transferred by you to any other person without Springer Science + Business Media's written permission.

No Amendment Except in Writing
This license may not be amended except in a writing signed by both parties (or, in the case of Springer Science + Business Media, by CCC on Springer Science + Business Media's behalf).

Objection to Contrary Terms
Springer Science + Business Media hereby objects to any terms contained in any purchase order, acknowledgment, check endorsement or other writing prepared by you, which terms are inconsistent with these terms and conditions or CCC's Billing and Payment terms and conditions. These terms and conditions, together with CCC's Billing and Payment terms and conditions (which are incorporated herein), comprise the entire agreement between you and Springer Science + Business Media (and CCC) concerning this licensing transaction. In the event of any conflict between your obligations established by these terms and conditions and those established by CCC's Billing and Payment terms and conditions, these terms and conditions shall control.

Jurisdiction
All disputes that may arise in connection with this present License, or the breach thereof, shall be settled exclusively by arbitration, to be held in The Netherlands, in accordance with Dutch law, and to be conducted under the Rules of the 'Netherlands Arbitrage Instituut' (Netherlands Institute of Arbitration).***OR:***

All disputes that may arise in connection with this present License, or the breach thereof, shall be settled exclusively by arbitration, to be held in the Federal Republic of Germany, in accordance with German law.

Other terms and conditions:

v1.3

If you would like to pay for this license now, please remit this license along with your payment made payable to "COPYRIGHT CLEARANCE CENTER" otherwise you will be invoiced within 48 hours of the license date. Payment should be in the form of a check or money order referencing your account number and this invoice number 501335616. Once you receive your invoice for this order, you may pay your invoice by credit card. Please follow instructions provided at that time.

Make Payment To:

SPRINGER LICENSE
TERMS AND CONDITIONS

Jun 24, 2014

This is a License Agreement between Rushdi Shams ("You") and Springer ("Springer") provided by Copyright Clearance Center ("CCC"). The license consists of your order details, the terms and conditions provided by Springer, and the payment terms and conditions.

All payments must be made in full to CCC. For payment instructions, please see information listed at the bottom of this form.

License Number	3415390358888
License date	Jun 24, 2014
Licensed content publisher	Springer
Licensed content publication	Springer eBook
Licensed content title	Extracting Information-Rich Part of Texts Using Text Denoising
Licensed content author	Rushdi Shams
Licensed content date	Jan 1, 2013
Type of Use	Thesis/Dissertation
Portion	Full text
Number of copies	1
Author of this Springer article	Yes and you are the sole author of the new work
Order reference number	None
Title of your thesis / dissertation	Identification of Information-richness in Text using Natural Language Stylometry
Expected completion date	Aug 2014
Estimated size(pages)	80
Total	0.00 USD
Terms and Conditions	

Introduction

The publisher for this copyrighted material is Springer Science + Business Media. By clicking "accept" in connection with completing this licensing transaction, you agree that the following terms and conditions apply to this transaction (along with the Billing and Payment terms and conditions established by Copyright Clearance Center, Inc. ("CCC"), at the time that you opened your Rightslink account and that are available at any time at http://myaccount.copyright.com).

Limited License

With reference to your request to reprint in your thesis material on which Springer Science and Business Media control the copyright, permission is granted, free of charge, for the use indicated in

established by CCC on its behalf in its Billing and Payment terms and conditions for this licensing transaction.

Indemnity
You hereby indemnify and agree to hold harmless Springer Science + Business Media and CCC, and their respective officers, directors, employees and agents, from and against any and all claims arising out of your use of the licensed material other than as specifically authorized pursuant to this license.

No Transfer of License
This license is personal to you and may not be sublicensed, assigned, or transferred by you to any other person without Springer Science + Business Media's written permission.

No Amendment Except in Writing
This license may not be amended except in a writing signed by both parties (or, in the case of Springer Science + Business Media, by CCC on Springer Science + Business Media's behalf).

Objection to Contrary Terms
Springer Science + Business Media hereby objects to any terms contained in any purchase order, acknowledgment, check endorsement or other writing prepared by you, which terms are inconsistent with these terms and conditions or CCC's Billing and Payment terms and conditions. These terms and conditions, together with CCC's Billing and Payment terms and conditions (which are incorporated herein), comprise the entire agreement between you and Springer Science + Business Media (and CCC) concerning this licensing transaction. In the event of any conflict between your obligations established by these terms and conditions and those established by CCC's Billing and Payment terms and conditions, these terms and conditions shall control.

Jurisdiction
All disputes that may arise in connection with this present License, or the breach thereof, shall be settled exclusively by arbitration, to be held in The Netherlands, in accordance with Dutch law, and to be conducted under the Rules of the 'Netherlands Arbitrage Instituut' (Netherlands Institute of Arbitration).***OR:***

All disputes that may arise in connection with this present License, or the breach thereof, shall be settled exclusively by arbitration, to be held in the Federal Republic of Germany, in accordance with German law.

Other terms and conditions:

v1.3

If you would like to pay for this license now, please remit this license along with your payment made payable to "COPYRIGHT CLEARANCE CENTER" otherwise you will be invoiced within 48 hours of the license date. Payment should be in the form of a check or money order referencing your account number and this invoice number 501335623. Once you receive your invoice for this order, you may pay your invoice by credit card. Please follow instructions provided at that time.

Make Payment To:
Copyright Clearance Center

Appendix D

Supporting Materials

Major Software Packages

1. The implementation of the summary sentence classifier (Chapter 11). Platform: Java and Weka. Download link:
   http://cogenglab.csd.uwo.ca/tools/text-summary-tool-2014.zip.

2. The software package for extracting protein interaction sentences from Biomedical Articles (Chapter 10). Platform: Java and Weka. Download link:
   http://cogenglab.csd.uwo.ca/tools/protein-interaction-tool-2014.zip.

3. Sentinel— the supervised anti-spam filter (Chapter 7). Platform: Java and Weka. Download link:
   http://cogenglab.csd.uwo.ca/tools/sentinel-tool-2013.zip.

4. Text Denoising—the supervised/semi-supervised tool to reduce insignificant data (Chapter 2 to 5). Platform: Java and WEKA. Download link:
   http://cogenglab.csd.uwo.ca/tools/TextDenoisingv2-2.zip.

5. BioConEx—the supervised biomedical concept extractor (Chapter 2). Platform: Java and Weka. Download link:
   http://cogenglab.csd.uwo.ca/tools/BioConEx.rar.

Minor Software Packages

1. SentEx—a supervised API using Java and LingPipe sentence boundary model for extracting sentences from free text. Download link:
   http://cogenglab.csd.uwo.ca/tools/SenTex.rar.

2. TRMS—a Text Readability Measuring Software using Java for measuring readability of texts. Download link:
   http://cogenglab.csd.uwo.ca/tools/TRMS.zip.

3. LDI—a Lexical Density Indexing tool using Java for measuring lexical densities of texts. Download link:
   http://cogenglab.csd.uwo.ca/tools/LDI.rar.

4. tf-idf—a tool using Java to provide the tf-idf measure of a given set of texts. Download link:
   http://cogenglab.csd.uwo.ca/tools/TF-IDF.zip.

Data Files

All the data files are in Attribute-Relation File Format (ARFF) and can be accessible using Weka (download from: http://www.cs.waikato.ac.nz/ml/weka/downloading.html).

1. The data files supporting the results of Chapter 11. Download link:
   http://cogenglab.csd.uwo.ca/datasets/text-summary-datafiles-2014.zip.

2. The data files supporting the results of Chapter 10. Download link:
   http://cogenglab.csd.uwo.ca/datasets/protein-interaction-datafiles-2014.zip.

3. The data files supporting the results of Chapters 7 to 9. Download link:
   http://cogenglab.csd.uwo.ca/datasets/sentinel-datafiles-2013.zip.

Datasets

1. The datasets that we develop and annotate described in Chapters 2 and 3. The files are in ASCII text format. Download link:
   http://cogenglab.csd.uwo.ca/datasets/discease-chemical-datasets-2011.zip.

2. The datasets that we annotate in this study are described in Chapter 10. The files are in ASCII text format. Download link:
   http://cogenglab.csd.uwo.ca/datasets/protein-interaction-datasets-2014.zip.

Curriculum Vitae

Name: **Rushdi Shams**

Education:

University of Western Ontario
London ON, Canada
Ph.D. Computer Science, August 2014.

University of Bolton
Bolton Lancs, United Kingdom
M.Sc. Information Technology, October 2007.

Khulna University of Engineering & Technology
Khulna, Bangladesh
B.Sc. Computer Science and Engineering February 2006.

Selected Awards:

Western Graduate Research Scholarship (WGRS), 2010—2014.
Western Graduate Thesis Research Award (GTRA), 2013.
Alberta Innovates Centre for Machine Learning (AICML) Grant, 2013.

Relevant Experience:

Graduate Research and Teaching Assistant
University of Western Ontario, London ON Canada
September 2010—August 2014.

Assistant Professor, Department of Computer Science and Engineering
Khulna University of Engineering & Technology, Bangladesh
March 2008—August 2010.

Researcher, Machine-mediated Multimodal Communications Lab
University of Bolton, United Kingdom
September 2007—January 2008.

Publications (in chronological order):

1. **Rushdi Shams** and Robert E. Mercer, Summary Sentence Classification using Stylometry, Unpublished Research Work, University of Western Ontario, 2014.

2. **Rushdi Shams** and Robert E. Mercer, Protein interaction sentence classification with natural language stylometry, *BMC Bioinformatics (submitted)*, 2014.

3. **Rushdi Shams** and Robert E. Mercer, Supervised Classification of Spam E-mails with Natural Language Stylometry, *Elsevier Expert Systems with Applications (submitted)*, 2014.

4. **Rushdi Shams** and Robert E. Mercer, Classifying Spam Emails with Text and Readability Features, *13th IEEE International Conference on Data Mining (ICDM2013)*, Texas, USA, December 7-10, 2013, pp. 657-666.

5. **Rushdi Shams** and Robert E. Mercer, Personalized Spam Filtering using Natural-Language Attributes, *12th IEEE International Conference on Machine Learning Applications (ICMLA2013)*, Florida, USA, December 4-7, 2013, pp. 127-132.

6. **Rushdi Shams**, Extracting Information-rich Part of Texts using Text Denoising, *26th Canadian Conference on Artificial Intelligence (CAI-2013)*, Regina, Canada, May 28-31, 2013, pp. 358-363.

7. **Rushdi Shams** and Robert E. Mercer. Evaluating Core Measures of Text Denoising for Biomedical Relation Mining. *3rd International Workshop on Global Collaboration of Information Schools (WIS 2012)*, Taipei, Taiwan, November 15, 2012.

8. **Rushdi Shams** and Robert E. Mercer. Improving Supervised Keyphrase Indexer Classification of Keyphrases with Text Denoising. *14th International Conference on Asia-Pacific Digital Libraries (ICADL 2012)*, Taipei, Taiwan, November 12-15, 2012, pp. 77-86.

9. **Rushdi Shams** and Robert E. Mercer. Investigating Keyphrase Indexing with Text Denoising. *2012 ACM/ IEEE- CS Joint Conference on Digital Libraries (JCDL2012)*. Washington DC, USA, 2012, pp. 263-266.

10. **Rushdi Shams** , M.S.A. Shahnawaz Chowdhury, and S.M. Abu Saleh Shawon. SenCept: A Domain-specific Textual Commonsense Concept Acquisition System. *International Journal of Computer and Information Technology (IJCIT)*, Vol. 1, No. 2, 2012.

11. **Rushdi Shams** and Robert E. Mercer. Extracting Connected Concepts from Biomedical Texts using Fog Index. *12th Conference of the Pacific Association for Computational Linguistics (PACLING 2011).* Kuala Lumpur, Malaysia, 2011, pp. 70-76.

12. **Rushdi Shams**, M.S.A. Shahnawaz Chowdhury, and S.M. Abu Saleh Shawon. Domain-specific Textual Commonsense Concept Acquisition using a Corpus. *1st IEEE International Conference on Communications, Computing and Control Applications (CCCA'11).* Hammamet, Tunisia, March 3-5, 2011. pp. 1-6.

13. **Rushdi Shams**, Adel Elsayed, and Quazi Mah- Zereen Akter. A Corpus-based Evaluation of a Domain-specific Text to Knowledge Mapping Prototype. *Journal of Computers*, Academy Publisher. vol. 5, no. 1, January 2010, pp. 69-80.

14. **Rushdi Shams**, M.M.A. Hashem, Afrina Hossain, Suraiya Rumana Akter, and Monika Gope. Corpus-based Text Summarization using Statistical and Linguistic Methods. *3rd IEEE International Conference on Computer and Communication Engineering (ICCCE'10)*. Malaysia, 2010, pp. 115-120.

15. Md. Amirul Islam, **Rushdi Shams**, and Md. Zahirul Islam, Minimization of Overlapping Coverage in Wireless Sensor Networks with Area-aware Coverage, *3rd IEEE International Conference on Computer and Communication Engineering (ICCCE'10)*, Kuala Lumpur, Malaysia, May 11-12, 2010, pp. 158-163.

16. M.M.A. Hashem, **Rushdi Shams**, Md. Abdul Kader, and Abu Sayed, Design and Development of Heart Rate Monitor using Fingertips, *3rd IEEE International Conference on Computer and Communication Engineering (ICCCE'10)*, Kuala Lumpur, Malaysia, May 11-12, 2010, pp. 197-201.

17. **Rushdi Shams** and Adel Elsayed. A Corpus-based Evaluation of Lexical Components of a Domain-specific Text to Knowledge Mapping Prototype. *11th IEEE International Conference on Computer and Information Technology (ICCIT 2008)*. Khulna, Bangladesh, 2008, pp. 242-247. (IEEE Catalog Number: CFP0817D, ISBN: 978-1-4244-2136-7, Library of Congress: 2008900703)

18. **Rushdi Shams** and Adel Elsayed. Development of a Conceptual Structure for a Domain-Specific Corpus. *3rd International Conference on Concept Mapping 2008 (CMC2008)*. Estonia and Finland, 2008.

19. Isnain Siddique, **Rushdi Shams**, and M.M.A. Hashem, Performance Enhancement of Ad Hoc Networks with Janitor Based Routing, *1st IEEE International Conference on Computer and Communication Engineering (ICCCE'06)*, Kuala Lumpur, Malaysia, May 9-11, 2006, vol. 1, pp. 368-373.